Applications of Sensor Fusion in Classification, Localization and Mapping

Mahi Abdelbar

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

William H. Tranter, Co-Chair
R. Michael Buehrer, Co-Chair
A. A. (Louis) Beex
Jung-Min Park
Said M. Elnoubi
Michael J. Roan

February 16, 2018
Blacksburg, Virginia

Keywords: Sensor Fusion, Automatic Modulation Classification, Distributed Networks, Simultaneous Localization and Mapping, Indoor Localization, Pedestrian Dead Reckoning

Applications of Sensor Fusion in Classification, Localization and Mapping

Mahi Abdelbar

(ABSTRACT)

Sensor Fusion is an essential framework in many Engineering fields. It is a relatively new paradigm for integrating data from multiple sources to synthesize new information that in general would not have been feasible from the individual parts. Within the wireless communications fields, many emerging technologies such as Wireless Sensor Networks (WSN), the Internet of Things (IoT), and spectrum sharing schemes, depend on large numbers of distributed nodes working collaboratively and sharing information. In addition, there is a huge proliferation of smartphones in the world with a growing set of cheap powerful embedded sensors. Smartphone sensors can collectively monitor a diverse range of human activities and the surrounding environment far beyond the scale of what was possible before. Wireless communications open up great opportunities for the application of sensor fusion techniques at multiple levels.

In this dissertation, we identify two key problems in wireless communications that can greatly benefit from sensor fusion algorithms: Automatic Modulation Classification (AMC) and indoor localization and mapping based on smartphone sensors. Automatic Modulation Classification is a key technology in Cognitive Radio (CR) networks, spectrum sharing, and wireless military applications. Although extensively researched, performance of signal classification at a single node is largely bounded by channel conditions which can easily be unreliable. Applying sensor fusion techniques to the signal classification problem within a network of distributed nodes is presented as a means to overcome the detrimental channel effects faced by single nodes and provide more reliable classification performance.

Indoor localization and mapping has gained increasing interest in recent years. Currently-deployed positioning techniques, such as the widely successful Global Positioning System (GPS), are optimized for outdoor operation. Providing indoor location estimates with high accuracy up to the room or suite level is an ongoing challenge. Recently, smartphone sensors, specially accelerometers and gyroscopes, provided attractive solutions to the indoor localization problem through Pedestrian Dead-Reckoning (PDR) frameworks, although still suffering from several challenges. Sensor fusion algorithms can be applied to provide new and efficient solutions to the indoor localization problem at two different levels: fusion of measurements from different sensors in a smartphone, and fusion of measurements from several smartphones within a collaborative framework.

Applications of Sensor Fusion in Classification, Localization and Mapping

Mahi Abdelbar

(General Audience Abstract)

Sensor Fusion is an essential paradigm in many Engineering fields. Information from different nodes, sensing various phenomena, is integrated to produce a general synthesis of the individual data. Sensor fusion provides a better understanding of the sensed phenomenon, improves the application or system performance, and helps overcome noise in individual measurements. In this dissertation we study some sensor fusion applications in wireless communications: (i) cooperative modulation classification and (ii) indoor localization and mapping at different levels. In cooperative modulation classification, data from different wireless distributed nodes is combined to generate a decision about the modulation scheme of an unknown wireless signal. For indoor localization and mapping, measurement data from smartphone sensors are combined through Pedestrian Dead Reckoning (PDR) to re-create movement trajectories of indoor mobile users, thus providing high-accuracy estimates of user's locations. In addition, measurements from collaborating users inside buildings are combined to enhance the trajectories' estimates and overcome limitations in single users' system performance. The results presented in both parts of this dissertation in different frameworks show that combining data from different collaborative sources greatly enhances systems' performances, and open the door for new and smart applications of sensor fusion in various wireless communications areas.

Dedication

To Ahmed,
to my parents, Wafaa and Othman
to Maye
and
to Zeyad, Alya, and Marawan

Acknowledgments

It has been a long, exciting, and sometimes very exhausting, journey. Virginia Tech graduate school has been my home for years, whether it was the year I spent in Alexandria University in Egypt as part of the VT-MENA program, or the years I spent in Blacksburg. Throughout this time, I was blessed to meet and know many wonderful people that not only filled me with amazement and excitement and opened my eyes to new and different life prospective, but they were also constant sources of help and support that I would have never been able to write this dissertation without their presence in my life.

First, I'd express my deepest gratitude to Dr. R. Michael Buehrer. Dr. Buehrer was the instructor for the first class I attended when I arrived in Blacksburg in Fall 2012. I was amazed by his vast knowledge, his way of explaining the most complex ideas, and his continued support for his students. Through the years I spent at the Wireless@VT lab, I had a lot of encounters with him, where he was always present, supportive, engaged and his door was always open. I was privileged to have him as a co-chair on my PhD committee. His patient guidance, enthusiastic encouragement and useful critiques of my work had a huge impact on me as an engineer, a researcher and a future academic. He has a unique way of approaching problems and identifying solutions that has broaden my problem-solving skills. Thank you so much for always believing in me. You have given me a role model for how a mentor should be, that I will always look up to.

I am extremely grateful for both Dr. Tamal Bose and Dr. William H. Tranter. Dr. Bose is the person who has given me the opportunity to join Virginia Tech in the first place. He opened the door for me to this life-changing experience. He also provided me with the rare opportunity of joining his team in the University of Arizona for six months, where I was fortunate to work closely with him. Despite the distance, he has always been supportive and providing critical feedback for my work. Dr. Tranter was the advisor that welcomed me to Blacksburg. I was fortunate enough to have the opportunity to work closely with him for some time before his retirement. He always provided valuable insights, critical feedback and extreme attention to details. Despite his retirement, he had always been a great mentor and supporter every step of the journey. I am his last PhD student and I will always be so proud of that.

I am especially thankful to Dr. Louis Beex. He was always very generous with his time to provide insights, comments and feedback on my research. I'd like to also thank Dr. Paul Plassmann for providing great support in my hour of need. Also, I would like to thank Dr. Karen P. DePauw for her great insights about higher education, and providing a great role model for how a woman faculty leader should be. I am also thankful the VT-MENA family: Dr. Sedki Riad, Dr. Mohamed Rizk, Dr. Mostafa ElNainay and Dr. Yasser Hanafy. Dr. Sedki provided a home for all VT-MENA students in Blacksburg when we all needed one, and was always in our backs. Dr. Mostafa, Dr. Mohamed and Dr. Yasser, thank you so much for all the effort and support you put into the program.

A special thank you to Hilda Reynolds and Nancy Goad for the million things that they do for us, the students at Wireless@VT lab. You made my life so much easier by taking care of so many things, and providing me with a comfortable safe home in the lab. You were the shoulders I leaned on every day.

This journey wouldn't have been enjoyable without the people I met in the lab, who shared with me the happiest and the hardest moments. A huge thank you goes to Dhiraj, Aditya and Avik. I started the journey with Avik and Aditya in 2012, and through the years we have been through so much together. Dhiraj has been a constant support, in research and beyond, while he was in Blacksburg and after he left. I will always be grateful for being your desk neighbour for such a brief time. Avik, thank you for always having my back. Aditya, thank you so much for sharing the last part of this journey with me. Thank you for coffee, laughs, and listening to my endless rants. Daniel and Emily Jakubisin, knowing your family was like the cool refreshing air in the middle of the heat. Munawwar and Bushra, thank you for sharing your life's happiest moments with me. Thad, thank you for all the help, discussions and feedback. You were always there whenever we needed you. My girls, Nishta and Mehrnaz, thank you for being my lab-sisters and empowering me through the hard times. Finally, Yaman, although we didn't spend much time together, I will be forever grateful for your support during my last two months in Blacksburg.

Now, for the people out of the lab who were my family for years. Noha El-sherbiny, I won't ever be able to express my gratitude for knowing you and being a part of your life, and you being a part of mine. You have given me a home, a family, a sister and a friend when I needed them the most. I would have never been able to conclude this journey without your constant help and support. You have shown me what true friendship is and I'll be forever grateful. Marwa Abdel Latif and Hiba Assi, every day I was amazed by your incredible strengths. Thank you for allowing me to witness strong independent women in action. Shaimaa Osman, thank you for being a great friend. Eman Mostafa and Mai El-Shehaly, the companions to my journey, I am greatly thankful for the time I got to know you both. And Finally, Sarah Elhelw, thank you for welcoming me to Blacksburg six years ago, and for seeing me off, and for everything in between. These extraordinary high-achieving women were the driving force that kept me going.

Finally, away from Blacksburg, I am extremely thankful for my family that supported my dream through the years. My parents, thank you so much for the unconditional love and support. You have supported me through all my decisions, and for that I will be forever grateful. My only sister, Mai, the super woman that managed to provide me with a shoulder to lean on for years, while studying, taking care of her family and kids, and working. You took care of so many things so I would only focus on my PhD. You are my person. And finally, my beloved Ahmed, my husband, who shared every step of the way with me. Although we were far apart for years, I always knew that on the other side of the world, you had my back. That was the greatest support of all. Thank you for every rant you listened to, for every panic you had to calm down, and for every trip you took to share a moment with me. I am blessed to have you as my partner. This would not have been possible without you. This one is for you all!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Sensor Fusion is an essential framework in contemporary and future sensor networks and communications systems. It's a relatively new paradigm for integrating data from multiple sources to synthesize new information that in general would not have been feasible from the individual parts [2]. The deployment of large numbers of smart sensor nodes, equipped with multiple on-board sensors and processors, provides immense opportunities for applications in a wide variety of areas [3]. Applications of sensor fusion algorithms in different engineering fields include: automation systems [4], multi-sensor image fusion for remote sensing applications [5], target tracking [6], mechatronics [7], and robotics [8], to name a few.

Advances in wireless communications opened limitless possibilities for sensor fusion-dependent applications. Wireless Sensor Networks (WSN), consisting of large numbers of spatially distributed wireless-capable sensors, are widely deployed to collect and process information about different biological, medical and environmental phenomena [9]. The current and predicted emergence of the Internet of Things (IoT) implies billions of inter-connected devices forming heterogeneous systems, generating, collecting, and sharing data [10]. Dynamic Spectrum Access (DSA) and Cognitive Radio Networks (CRN) use collaborative spectrum sensing between cognitive mobile devices as one of its key underlying technologies [11]. Development of smart and effective sensor fusion algorithms in these different applications is an active and exciting ongoing research area.

Recently, the huge proliferation of smartphones in world markets provided the opportunity for utilizing smartphones as distributed sensors within crowdsourcing frameworks [12]. Smartphones are equipped with reasonably powerful processors, and a set of sensors such as accelerometers, digital compasses, proximity

sensors, gyroscopes, GPS, microphones, and cameras. These sensors can detect and monitor a large variety of biological activities and environmental phenomena [13]. Smartphones provide two different levels of sensor fusion: (1) fusion of data gathered by sensors in a single device (personal sensing) to detect a local activity or phenomenon and (2) fusion of data gathered by multiple smartphones, either through user input or through background data collection (group sensing). Personal sensor fusion applications usually include health and exercise monitoring. On the other hand, vehicular and pedestrian navigation, and traffic monitoring are usually group sensing applications. Localization applications can be considered as both a personal and a group sensing application depending on the level of sharing the user opts for [14].

In this work, we identify two key problems in wireless communications that can greatly benefit from sensor fusion algorithms: Automatic Modulation Classification (AMC) and Indoor Localization and Mapping based on smartphone sensors. Sensor Fusion is a multi-level process which includes detection, association, estimation and combination of data from single or multiple sources [15]. Sensor fusion algorithms proposed in this work refer to fusion at two different levels: combining measurement data from different sensors within the same node, either at the same time or different times, and combining measurement data from distributed nodes.

1.1.1 Automatic Modulation Classification (AMC)

Several emerging wireless communications standards include spectrum sharing [11] and Cognitive Radio (CR) [16] as two of their underlying technologies. Developing standards such as IEEE 802.11af WiFi using TV White spaces (White-Fi), IEEE 802.15.4 Low-Rate Wireless Personal Area Networks (LR-WPANs), and Public Safety (3GPP LTE FirstNet), all involve sharing the spectrum among a vast number of smart heterogeneous devices. Radio nodes in these networks need to perform much more than mere sensing of the signals in their vicinity. Nodes may want to find compatible devices they can communicate with, locate un-occupied frequencies, or avoid bands occupied by certain types of devices [17]. One of the key enabling components of unknown signal identification systems is the classification of the underlying modulation scheme. In addition, modulation classification plays a big role in military applications. Detection and classification of unknown wireless signals in warfare scenarios has several goals, including: to classify friendly and hostile transmissions, and to help design effective counter measures (i.e. jamming techniques [18]). Joint detection and signal classification techniques have long been part of the military Software Defined Radio (SDR) research [19].

Single-node modulation classification has been heavily researched over the last three decades [20]. However, as more technologies are moving towards large numbers of distributed wireless nodes, and the need increases for the nodes to identify wireless signals within their vicinity, applying sensor fusion algorithms is a natural evolution to the AMC problem. Data fusion from a set of geographically dispersed sensors, assum-

ing independent channel parameters, provides a better statistical description than any individual node [21], leading to more effective classification performance specially under degrading channel effects. In addition, combining data from multiple scattered nodes allows for using individual nodes with less strict computational requirements.

The idea of applying fusion algorithms to modulation classification and forming a *collaborative* or *cooperative* classification problem has been presented within the AMC literature, mainly in the works by W. Su et al. in [21–25], G. B. Markovic et al. in [26–30], and B. Dulek in [31–33]. B. Dulek and some of W. Su et al.'s work are based on the likelihood-based modulation classification approaches, which are mostly mathematically tractable but present their own high computational challenges that are believed to be not appropriate for small sensor nodes with low computational capabilities. Other works by W. Su et al. and G. B. Markovic et al., on the other hand, are based on variations of statistical feature-based modulation classification approaches which require less computations and are more suitable for distributed low power nodes. However, performance of feature-based cooperative algorithms has been presented mostly heuristically through simulations without tractable analysis.

Hence, the motivation of the first part of the dissertation is to present a theoretical framework for the cooperative modulation classification problem based on the fusion of feature data from distributed wireless sensor nodes. The tractability of the framework allows for better understanding of both the benefits and the challenges of the cooperation between nodes, and provides a reference for the efficiency of cooperation in signal classification scenarios. The approach to develop this framework follows these steps:

- Develop a framework for the performance of a single node AMC, and study the performance in different scenarios
- Develop a fusion algorithm for collaborative classification at the feature-level from multiple wireless sensor nodes
- Analyze the performance of the collaborative classification system under variable conditions

1.1.2 Indoor Localization and Mapping

With the introduction of many location-based services and location-guidance applications for smartphone users in modern wireless communications networks, indoor localization has gained increasing interest in recent years. Locating people with high accuracy within buildings is a key enabling technology for many intelligent and context-aware service provisioning scenarios. Indoor navigation applications which depend on indoor localization techniques include: guidance for patrons of museums and fitness centers, conference attendees, medical personnel and patients in health care facilities [34–36]. In addition, the Federal Communications Committee (FCC) is adapting new regulations for wireless Enhanced 911 (E911) caller

location accuracy. The report released by the FCC in February 2015 proposed that Commercial Mobile Radio Services (CMRS) providers should be able to provide to the Public Safety Answering Points (PSAPs) a dispatchable location or an x/y location within 50 meters for 70% of all 911 wireless calls within five years [37]. The FCC seeks to use technologies that would provide public safety personnel with caller location information up to a specific floor and room/suite number accuracy.

However, although currently-deployed positioning techniques (such as Cell ID, Base Station Time Difference of Arrival (BS-TDOA) and Global Positioning Systems (GPS)) work well for outdoor scenarios, they typically cannot provide the accuracy necessary for indoor localization up to the room/suite level due to the line-of-sight (LOS) problem [38]. The majority of indoor localization techniques adopt *fingerprinting* as the underlying scheme of location determination, such as the systems presented in [39, 40]. Fingerprinting is based on two stages: an off-line stage where site surveyors record Received Signal Strengths (RSS) from indoor Base Stations (BS), e.g. WiFi Access Points (AP), at specific locations inside a building creating a fingerprint map. In the operating stage, users query the fingerprint database to find the location best matching their current RSS. These fingerprinting techniques present their own set of challenges. The most pressing one is the non-stationarity of the radio map due to the dynamic nature of the environment. In addition, fingerprinting depends on the pre-existing infrastructure, knowledge of building floorplans, and the site surveying stage which is non-trivial [35, 41]. Novel methods for indoor localization of mobile users are required.

A new direction for indoor localization is the tracking of the smartphone user's movement trajectories inside buildings and exploiting these trajectories for more accurate localization, instead of applying localization algorithms that depend only on the most recent position. This idea has been presented in very few indoor localization algorithms within RF fingerprinting techniques such as the ones presented in [42–44]. Thus, the motivation for the second part of the dissertation is to investigate the use of movement trajectories for indoor localization within new frameworks that are independent of RF maps. The three main questions that motivated the research are:

- How can smartphones be utilized to track movement trajectories within buildings?
- How can smartphones within the same environment cooperate to provide better trajectory estimates for their users?
- How can movement trajectories be used for localization?

Sensor fusion presents itself as a major component in the solutions to these questions at different levels. First, measurements from smartphone sensors, such as accelerometers, gyroscopes and magnetometers, can be combined to provide estimates for the relative positions of the smartphone user and hence tracking the user movement trajectory through ***Pedestrian Dead-Reckoning (PDR)*** techniques [45]. Second, data fusion

between multiple smartphones located within the same area improves the accuracy of trajectory tracking, localization, and resulting maps.

Movement Trajectory Tracking

The problem of tracking mobile users' movement trajectories within buildings can be defined as a mapping problem as well, in two different contexts: (1) tracking a user's movement trajectory inside a building provides a *map* of the user's own movement. This map can be used for detecting the user's current position, for studying the user's motion pattern, among other applications. (2) Users' movement trajectories create topological maps of building floor plans, without *a priori* information about the building. These maps can be used to further enhance localization as well as in other applications such as building planning and navigation.

In parallel, ***Simultaneous Localization and Mapping (SLAM)*** is a heavily researched problem in the Robotics community [46]. SLAM is defined as the construction of a consistent map of a robot's previously unknown surrounding environment while simultaneously estimating its position within the map. Solving the SLAM problem can be categorized into two main approaches: filtering and smoothing [46–48]. Filtering approaches estimate a user's position recursively as new measurements become available, with the most popular algorithms based on Extended Kalman Filters (EKF) and Particle Filters (PF). Smoothing approaches, also referred to as GraphSLAM, estimate a sequence of user movements instead of just the current position in a batch-processing approach [48]. Adapting the SLAM problem to the pedestrian indoor localization problem provides a novel approach for tracking user movement trajectories.

In order to answer our first research question, we present an improved algorithm for indoor movement trajectory tracking using smartphone PDR within a GraphSLAM framework. This work is at the intersection of sensor fusion algorithms, indoor localization systems and pedestrian SLAM algorithms.

Collaborative Indoor Localization and Mapping

GraphSLAM is a probabilistic framework for the SLAM problem whose objective is to find the most probable sequence of user positions within an environment through minimizing the measurements errors from different sensors, thus forming an optimization problem [48]. The keywords for GraphSLAM are *loop closure* and *landmark*. Loop closure is the ability of the user to identify previously visited places, while landmarks are fixed-positions within the environment [49]. Loop closures and landmarks provide constraints to the GraphSLAM optimization problem. More loop closures/landmarks in the environment result in more constraints to the measurement error minimization problem which leads to improved trajectory tracking performance.

However, detecting loop closures and landmarks using smartphone sensors is not trivial. While for robots, images and range measurements processing techniques are used for data association and loop closure/landmark detection, this is not the case for smartphone users. Autonomous landmarks for smartphone users are mainly based on fixed wireless transmission points, for example WiFi AP [50], Bluetooth beacons [51] and RFID detectors [52], assuming pre-existing installations in the building. Other techniques for smartphone landmark detection suggest the detection and classification of user activities as landmarks, such as movement through stairs and elevators, which requires additional complex intelligent algorithms [53].

Hence, the second question in this part of the dissertation, is how to use cooperation between smartphone users to create virtual constraints to the GraphSLAM problem, without either the dependency on existing infrastructure, which cannot be guaranteed in unknown environments, nor the additional processing of detecting other activities or phenomena.

Indoor Localization through Movement Trajectories

Tracking mobile users' movement trajectories with high accuracy indoors opens the possibilities for novel indoor localization techniques. Using GraphSLAM for indoor pedestrian trajectory tracking, or other trajectory tracking algorithms, results in creating a group of high probability movement patterns within a building, or what we can refer to as a topological map of the building. With new users tracking their own movements for specific time duration, these new and possibly low accuracy trajectories can be used to locate, with high accuracy, where the users are inside the building through smart trajectory matching techniques. The idea is motivated through the successful implementation of snapping techniques in traffic and navigation applications where vehicles' movements are snapped to the closest match.

Thus, in the final part of the dissertation, we propose novel non-traditional techniques for indoor pedestrian localization through the application of pattern recognition and machine learning algorithms.

1.2 Contributions

The dissertation identifies two main applications for sensor and/or data fusion in wireless communications: Automatic Modulation Classification and Indoor Localization and Mapping, through the following contributions:

- Chapter 2: Analysis of the performance of a single-node cumulants-based modulation classifier.
- Chapter 3: Development of a collaborative cumulants-based modulation classification frame- work in distributed networks.

- Chapter 4: Development of an improved smartphones-based sensor fusion algorithm within the Pedestrian Dead-Reckoning (PDR) GraphSLAM framework for indoor mobile users' movement trajectory tracking.
- Chapter 5: Development of a new collaborative PDR GraphSLAM algorithm based on Bluetooth Low Energy (BLE) detection for indoor mobile users' movement trajectory tracking.
- Chapter 6: Development of non-traditional indoor localization techniques based on movement trajectory tracking.

Each of the problems investigated in the dissertation spans consecutive chapters: the automatic modulation classification problem is addressed in Chapters 2 and 3, and the indoor localization and mapping problem is addressed in Chapters 4, 5 and 6.

1.2.1 Single Node Cumulants-based Modulation Classification

In order to study the performance of cooperative feature-based modulation classification, the performance of a single-node classifier is studied first. Cumulants as classification features are a very popular choice for modulation classification. However, a complete analysis of the classifier performance is not available in the literature. In this chapter, we present a detailed analysis of the performance of a single node AMC, in terms of the average probability of correct classification, based on high-order cumulants and Maximum Likelihood (ML) Classification. The performance is studied under various conditions: (1) AWGN and flat Rayleigh fading channels and (2) classification of sets of two modulation schemes and sets of more than two modulation schemes. The contributions of the work presented in this chapter are:

- A comprehensive analysis of the performance of a fourth-order cumulants-based modulation classifier in terms of the probability of correct classification is presented under both AWGN and flat Rayleigh Fading Channels.
- A new hierarchical framework for classification of multiple modulation schemes is presented, also under both AWGN and Rayleigh Fading Channels.

1.2.2 Collaborative Cumulants-based Modulation Classification

In this chapter, we study the effect of collaboration on the performance of a group of distributed classifier nodes. We propose a centralized feature-level combining algorithm: each node calculates a set of features and shares this set with a Fusion Center (FC) where an overall decision is made. The FC applies a ML combining approach. The performance of the FC versus a single node is studied in terms of the average

probability of correct classification under various conditions: (1) AWGN and flat Rayleigh fading channels, (2) combining of features calculated at nodes receiving equal or non-equal average Signal-to-Noise Ratios (SNR), (3) classification of sets of two modulation schemes and sets of more than two modulation schemes, and (4) combining of features calculated from independent versus correlated signals. The main contributions of this chapter are:

- Development and analysis of a cumulants-based ML cooperative classification framework for digital modulation schemes under both AWGN and flat Rayleigh fading.
- Development of a new hierarchical framework for classification of multiple modulation schemes in cooperation scenarios.
- Performance analysis of the cooperative classification framework for the case of correlated received signals at the cooperating nodes.

1.2.3 Smartphones-based PDR for Indoor GraphSLAM

In this chapter, we develop an improved smartphones-based dead-reckoning algorithm for tracking the movement of indoor pedestrians. Measurements from smartphone sensors (accelerometer, magnetometer and gyroscope) are synthesized to generate an estimate of the user's movement trajectory within a building. The estimated trajectory is optimized within a GraphSLAM algorithm that estimates the most probable user positions through minimization of the measurement errors. Experiments were conducted inside an academic building at Virginia Tech with several pedestrians walking through trajectories of different lengths and duration while measurements are collected using an iPhone 7. The main contributions of this chapter are:

- Development of a *heading calibration* algorithm to overcome the drift errors in the smart- phone's heading angle measurements.
- Modeling of the displacement measurement errors resulting from the fusion of accelerometer and gyroscope measurements.
- Development of a *heading detection* pre-processing stage for GraphSLAM to adopt the algorithm to the pedestrian movement pattern, with minimal dependency on the building infrastructure.

1.2.4 Collaborative Pedestrian GraphSLAM based on BLE

GraphSLAM performance enhancement depends on detecting loop closures and/or landmarks in the environment which provide constraints to the measurement error minimization problem. In this chapter, we

propose a new collaborative GraphSLAM algorithm based on Bluetooth Low Energy (BLE) technology as a means to provide constraints to the optimization problem without the dependency on infrastructure landmarks. As part of measurements collection for PDR, smartphones scan and detect other BLE devices within their vicinity. By sharing the positions at which the phones were in close proximity with each other, these positions can provide virtual constraints to the joint optimization problem. Experiments were conducted within the academic building, with two users walking through different overlapping trajectories while holding two iPhone 7 devices collecting both odometry and BLE measurements. The main contributions in this chapter are:

- Modeling of the BLE Received Signal Strength (RSS) between smartphones (iPhone 7) in order to investigate its viability as a proximity detector.
- Development of a joint GraphSLAM problem where BLE RSS measurements between participating users provide non-equality constraints for the optimization problem.

1.2.5 Indoor Localization through Trajectory Tracking

In this chapter, we propose new non-traditional techniques for indoor localization through trajectory tracking. The idea of collecting movement trajectories within a building to further use as movement patterns suggests the application of pattern recognition techniques to the localization problem. Matching trajectories tracked through smartphone sensors or any other low-accuracy localization technique to a set of probable patterns within a building results in identifying the most recent position of the user with much higher accuracy up to the room level. In order to do that, we proposed using two different pattern recognition techniques: Neural Networks (NN) and trajectory similarity measures. The main contributions of this chapter are:

- Application of trajectory similarity measures as a new approach to the indoor localization problem at the room/suite level.
- Application of NN as a multi-class classifier to the indoor localization problem at the room level.

1.3 Relevant Publications

The work presented in this dissertation is primarily based on the following publications:

Cooperative Modulation Classification

- M. Abdelbar, B. Tranter, and T. Bose, "Cooperative Cumulants-Based Modulation Clas- sification in Distributed Networks", *IEEE Trans. Cogn. Commun. Netw., accepted for publication*. [54]
- M. Abdelbar, B. Tranter, and T. Bose, "Cooperative Cumulants-based Modulation Clas- sification under Flat Rayleigh Fading Channels," in *Proc. IEEE ICC*, Jun. 2015, pp. 7622- 7627. [55]
- M. Abdelbar, B. Tranter, and T. Bose, "Cooperative Combining of Cumulants-based Mod- ulation Classification in CR Networks," in *Proc. IEEE MILCOM*, Oct. 2014, pp. 434-439. [56]
- M. Abdelbar, B. Tranter, and T. Bose, "Cooperative Modulation Classification of Multiple Signals in Cognitive Radio Networks," in *Proc. IEEE ICC*, Jun. 2014, pp. 1483-1488. [57]

Indoor Localization and Mapping

- M. Abdelbar and R. M. Buehrer, "An Improved Technique for Indoor Pedestrian Graph- SLAM using Smartphones", *IEEE Sensors J., submitted for publication*. [58]
- M. Abdelbar and R. M. Buehrer, "Collaborative Pedestrian GraphSLAM based on Bluetooth Low Energy", *IEEE Trans. Mobile Comput., submitted for publication*. [59]
- M. Abdelbar and R. M. Buehrer,"Pedestrian GraphSLAM using Smartphones-based PDR in Indoor Environments", *IEEE ICC - Workshop on Advances in Network Localization and Navigation (ANLN), accepted for publication*, May 2018. [60]
- M. Abdelbar and R. M. Buehrer, "Indoor Localization through Trajectory Tracking using Neural Networks," in *Proc. IEEE MILCOM*, Oct. 2017, pp. 519-524. [61]
- M. Abdelbar and R. M. Buehrer, "Improving cellular Positioning Indoors through Trajectory Match- ing," in *IEEE/ION Position, Location and Navigation Symp. (PLANS)*, Apr. 2016, pp. 219-224. [62]

Chapter 2

Single-node Cumulants-based Modulation Classification Performance

Automatic Modulation Classification (AMC) of unknown wireless signals has been studied as a key technology in several Cognitive Radio (CR) and military applications. However, performance of AMC degrades severely under low Signal-to-Noise Ratios and fading channel scenarios. Cooperative classification is presented as a means to enhance the classification performance as well as to relax the computational constraints on individual nodes as presented in Chapter 3. In order to study the performance enhancement due to cooperative classification, the performance of single-node AMC is presented first under various conditions. The main contributions of this chapter are: (1) a comprehensive analysis of the performance of a fourth order cumulants-based modulation classifier in terms of the probability of correct classification is presented under both AWGN and Rayleigh Fading Channels, and (2) a new hierarchical framework for classification of larger sets of unknown modulation schemes is presented for the single-node classifier case.

2.1 Introduction

Signal classification has been a heavily researched area for the past three decades. Automatic Modulation Classification (AMC) was first introduced as a means to identify the underlying mod- ulation schemes of unknown signals. AMC has gained increasing interest with the introduction of Cognitive Radio (CR) networks [16] and Dynamic Spectrum Access (DSA) techniques [11]. In addition, unknown signal classification is an essential technology in electronic warfare scenarios. Recently, unknown signal classification has extended beyond AMC to include classification of advanced transmission technologies, such as OFDM and MIMO.

Signal classification algorithms are generally categorized into Likelihood-Based (LB) algorithms and Feature-based(FB) algorithms [20]. LB approaches formulate the modulation classification problem as a multiple composite hypothesis testing problem whose solution depends on the modeling of the unknown quantities, providing optimum classification performance with high computational requirements [63, 64]. On the other hand, FB algorithms provide sub-optimal performance with reasonable computational complexities. FB algorithms usually consist of two-stages: feature extraction and classification. Most FB classifiers rely on statistical signal features, which can generally be categorized into: (1) higher order cumulants and moments, used to classify digital modulation schemes [65–69], OFDM signals [70], and MIMO schemes [71–74], (2) cyclostationarity characteristics including cyclic cumulants, used also for digital modulation schemes classification [75–79], WiMAX, OFDM and BT-SCLD [80–82], and MIMO schemes [83–85], and recently (3) cross-correlation characteristics for MIMO classification [86–89].

In this work, we present a Maximum Likelihood (ML) feature-based AMC based on higher order cumulants. High-order cumulants are chosen as features for modulation classification because of their independence and noise immunity properties. The combination of ML and cumulants features allow for mathematical tractability of the algorithm performance. This chapter is organized as follows: an introduction to high-order cumulants is first presented in Section 2.2. Section 2.3 presents a theoretical framework for ML cumulants-based AMC in a single node under both AWGN channels and flat Rayleigh Fading channels. Section 2.4 presents a new hierarchical approach for modulation classification of more than two modulation schemes. Concluding Remarks are presented in Section 2.5.

2.2 High-Order Cumulants for Modulation Classification

Cumulants are high-order statistical features of random variables defined in terms of high-order moments. In this section, we first present an overview of the basic cumulants' definitions. Then large sample estimates of cumulants are presented with their respective asymptotic distributions as a basic tool for modeling cumulants' estimates.

2.2.1 Basic Cumulant Definitions

For random processes $y_1(n), y_2(n), ..., y_k(n)$, where $n = 1, ..., N$ and N is the length of the observation interval, the k^{th}-order joint moment and cumulant are defined as in [90]:

$$M\left(y_1, ..., y_k\right) = E\left[\prod_{j=1}^{k} y_j(n)\right], \tag{2.1}$$

Table 2.1: Theoretical Values of Second- and Fourth-order Cumulants for Different Modulation Schemes

	Real-Valued			Complex-Valued	
	BPSK	4ASK		QPSK	16QAM
C_2	1	1	C_{20}	0	0
			C_{21}	1	1
C_4	-2	-1.36	C_{40}	0	-0.68
			C_{42}	-1	-0.68

$$C\left(y_1,...,y_k\right)=\sum_{p}\left(-1\right)^{|p|-1}\left(|p|-1\right)!\prod_{\nu\in p}\left(E\left[\prod_{j\in\nu}y_j(n)\right]\right), \tag{2.2}$$

where p is the list of all possible partitions of $\{1,2,...,k\}$, $|p|$ is the number of parts (blocks) in each partition, ν is the list of all blocks within partition p. When $y_j = y, j = 1,...,k$, this gives the k^{th}-order moment M_k of the random process $y(n)$:

$$M_k = E\left[y(n)^k\right]. \tag{2.3}$$

For complex-valued random processes, mixed moments are defined according to the conjugation placements. The mixed moment of order k with m conjugations M_{km} is defined as in [65]:

$$M_{km} = E\left[y(n)^{k-m}\left(y(n)^*\right)^m\right]. \tag{2.4}$$

The k^{th}-order cumulants C_k and the k^{th}-order mixed cumulants with m conjugations C_{km} can be defined accordingly as in (2.2). Appendix A lists the equations relating high-order moments and cumulants up to the 8^{th}-order. The two basic properties of cumulants that make them attractive for AMC are: (a) the cumulants of Gaussian processes of order higher than two are equal to zero and (b) the cumulant of the sum of two statistically independent random processes equals the sum of the cumulants of the independent random processes [90, 91].

Unknown digital modulation schemes are modeled as random processes of length N, while high-order cumulants are selected as classification features. The cumulants of most interest are the 2^{nd}- and 4^{th}-order: $\{C_2, C_4, C_{20}, C_{21}, C_{40}, C_{42}\}$. Their theoretical noise-free values are presented in (A.1, A.3) and are calculated for some selected schemes in Table 2.1. For symmetric complex-valued modulations, $C_{20} = M_{20} = 0$. Throughout this work, modulation schemes are normalized to have unit energy, such that ideally $C_2 = C_{21} = 1$. The values are obtained by computing the ensemble averages over the noise-free constellations, under the constraint of unit energy [65].

2.2.2 Sample Estimates of Cumulants

For a random process $y(n)$, the values of high-order moments and cumulants can be estimated through sample averages [65]. For real-valued processes:

$$\hat{C}_2 = \hat{M}_2 = \frac{1}{N}\sum_{n=1}^{N} y^2(n), \quad \hat{C}_4 = \hat{M}_4 - 3\hat{M}_2^2 = \frac{1}{N}\sum_{n=1}^{N} y^4(n) - 3\left(\frac{1}{N}\sum_{n=1}^{N} y^2(n)\right)^2, \tag{2.5}$$

while for complex-valued processes:

$$\begin{aligned} &\hat{C}_{20} = \hat{M}_{20} = \frac{1}{N}\sum_{n=1}^{N} y^2(n), \quad \hat{C}_{21} = \hat{M}_{21} = \frac{1}{N}\sum_{n=1}^{N} |y(n)|^2, \\ &\hat{C}_{42} = \hat{M}_{42} - \left|\hat{M}_{20}\right|^2 - 2\hat{M}_{21}^2 = \frac{1}{N}\sum_{n=1}^{N} |y(n)|^4 - \left|\frac{1}{N}\sum_{n=1}^{N} y^2(n)\right|^2 - 2\left(\frac{1}{N}\sum_{n=1}^{N} |y(n)|^2\right)^2, \end{aligned} \tag{2.6}$$

where $\hat{M}_k$, $\hat{C}_k$, $\hat{M}_{km}$ and $\hat{C}_{km}$ are the sample estimates of moments and cumulants for real- and complex-valued processes respectively. From [92], invoking the central limit theorem, $\hat{M}_{km}$ is an unbiased estimator and asymptotically Gaussian, with:

$$E\left[\hat{M}_{km}\right] = M_{km}, \qquad Var\left[\hat{M}_{km}\right] = \left(M_{(2k)k} - |M_{km}|^2\right)/N, \tag{2.7}$$

where $E[x]$ is the expected value and $Var[x]$ is the variance of the random variable x respectively. From (2.7), high-order cumulants' estimates presented in (2.5) and (2.6) are also unbiased estimators and asymptotically Gaussian. In Appendix B, a detailed derivation of the expected value and the variance for each cumulant of interest is presented as functions of (1) theoretical noise-free values of C_2 or C_{21} which correspond to the average signal energy, (2) theoretical noise-free other high-order cumulants' values, and (3) the observation length N. Corresponding equations are presented in (B.1) for 2^{nd}-order, and in (B.4), (B.11), (B.15) and (B.19) for 4^{th}-order cumulants.

2.3 Single-node Performance of Cumulants-Based AMC

Swami and Sadler introduced in their seminal paper [65] high-order cumulants as robust classification features for AMC with low computational complexity. Since then, many feature-based modulation classification algorithms have proposed using high-order cumulants, for example: [66, 67, 93]. ML classification was initially introduced in [65], assuming equal variances for cumulants' estimates of different modulation schemes. The equal variance assumption results in threshold detection or nearest neighbor classifiers which were applied in most cumulants-based AMC. However, through investigation of the variance values of cu-

mulants' estimates for different modulation schemes, the equal variance assumption is not generally valid (the assumption is mostly valid only for modulation schemes within the same set, e.g. QAM set). In this work, ML classification is proposed in which different variance values for cumulants' estimates are taken into consideration. We propose a new theoretical analysis for the probability of correct classification for a cumulants- based modulation classifier for a single node.

2.3.1 Signal Models and Cumulants' Estimates

Throughout Chapters 2 and 3, a received unknown signal sequence $y(n)$ at each node is modeled according to one of two channel models: AWGN and flat Rayleigh fading channels. Each node calculates an estimate of a high-order cumulant of the received signal $\hat{C}_{km_y}$ (or $\hat{C}_{k_y}$), which is then used as a classification feature. In this section, we present each signal model and how it affects estimated cumulant values.

AWGN Channel model

The received signal $y(n)$ at each node is modeled as:

$$y\left(n\right) = x(n) + w(n), \quad n = 1, 2, ..., N, \tag{2.8}$$

where $x(n)$ is the transmitted signal sequence and $w(n)$ is the additive noise sequence, $w \sim \mathcal{CN}(0, \sigma_g^2)$. Based on the cumulants' two basic properties stated in Section 2.2.1, all cumulants of $y(n)$ of order more than 2 ideally are not affected by AWGN and equal to the corresponding cumulants of $x(n)$, i.e. $C_{km_y} = C_{km_x}$ except 2^{nd}-order cumulants $\{C_{21}, C_2\}$, where:

$$C_{21y} = C_{21x} + {\sigma_g}^2 = C_{21x}\left(1 + 1/SNR\right), \quad SNR = C_{21x}/\sigma_g^2, \tag{2.9}$$

where SNR is the Signal-to-Noise Ratio at the receiving node (the same applies to C_2). Based on (2.9) and Appendix B, both the expected value and the variance of high-order cumulants' estimates of $y(n)$ can be represented as functions of the SNR. For example, for the 4^{th}-order cumulant $\hat{C}_{42}$ (B.15, B.19):

$$\begin{aligned} &E_{\hat{C}_{42y}}\left(SNR\right) = \left(\frac{N-2}{N}\right) C_{42x} - \left(\frac{2}{N}\right) C_{21x}^2 \left(1 + 1/SNR\right)^2, \\ &Var_{\hat{C}_{42y}}(SNR) = p_0 + p_1 C_{21x}\left(1 + 1/SNR\right) + \ p_2 C_{21x}^2 \left(1 + 1/SNR\right)^2 + p_4 C_{21x}^4 \left(1 + 1/SNR\right)^4. \end{aligned} \tag{2.10}$$

Fig. 2.1 shows the expected value and variance of $\hat{C}_{42}$ for a $16QAM$ modulation scheme at two different SNR values. Low SNR affects both the expected value and the variance of the estimated cumulant.

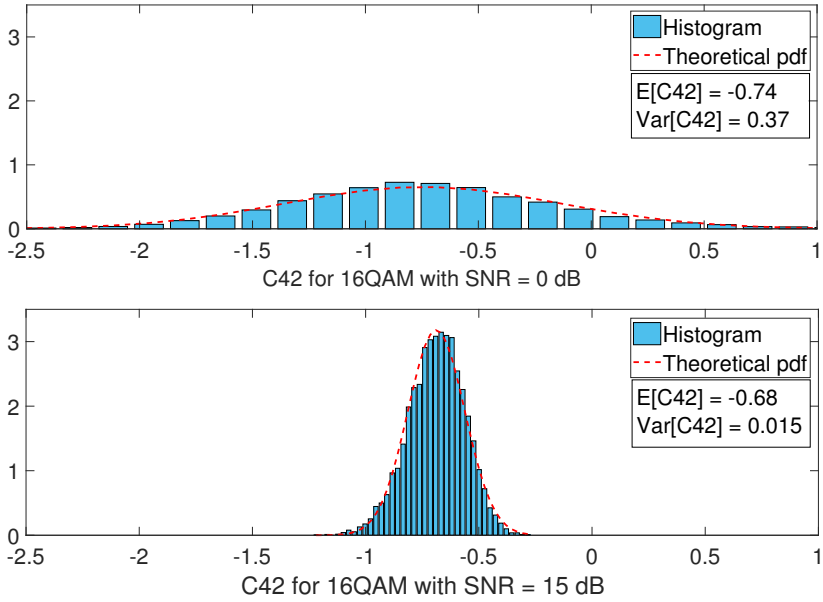

Figure 2.1: Histogram and theoretical pdf for $\hat{C}_{42}$ for 16QAM modulation scheme for different SNR values: 0 dB and 15 dB, N = 100.

Flat Rayleigh Fading Channel Model

The signal $y(n)$ at each node is modeled as:

$$y(n) = hx(n) + w(n), \quad n = 1, 2, ..., N, \tag{2.11}$$

where h is the channel coefficient assumed to be constant over the observation length N, $h \sim \mathcal{CN}(0, \sigma_h^2)$ such that the amplitude of h follows a Rayleigh distribution, i.e. $\nu = |h| \sim \mathcal{R}(\sigma_h\sqrt{\frac{\pi}{2}}, \sigma_h^2(2-\frac{\pi}{2}))$. Based on the cumulants properties [90], the estimates of high-order cumulants in this case are calculated as $C_{km_y} = |h|^k C_{km_x}$. Thus, to estimate the k^{th}-order cumulant of the transmitted sequence $x(n)$, the node needs to estimate the value of the channel coefficient amplitude $|h|$. Several algorithms for blind channel estimation were presented, for example in [66, 94–96]. Dividing $\hat{C}_{km_y}$ by $|h|^k$ results in a cumulant estimate with the same asymptotic parameters as in the AWGN channel case (for example in (2.10)), but with the SNR modeled as an exponential random variable with an instantaneous SNR that is a scaled version of the SNR under the AWGN channel:

$$SNR = |h|^2 C_{21x}/\sigma_g^2, \quad E\left[SNR\right] = E\left[|h|^2\right] C_{21x}/\sigma_g^2. \tag{2.12}$$

2.3.2 Maximum Likelihood Classifier

The node uses a ML classifier to decide the modulation scheme of $x(n)$ from a set of M possible modulation schemes, $M = \{M_1, M_2, ..., M_{N_M}\}$. Based on an estimated cumulant value and an estimation for the SNR: $s\hat{n}r$, the classifier generates a decision $D \in M$ that maximizes the probability that $\hat{C}_{km_x}$ is drawn from that modulation scheme M_j:

$$D = \underset{j}{argmax}\{p((\hat{C}_{km_x}, s\hat{n}r)|D = M_j)\}, \tag{2.13}$$

based on Bayes' rule and assuming a uniform distribution of the candidate modulation schemes. $\hat{C}_{km_x} = \hat{C}_{km_y}$ for the AWGN channel, $\hat{C}_{km_x} = \hat{C}_{km_y}/|h|^k$ for the fading channel, and $p((\hat{C}_{km_x}, s\hat{n}r)|D = M_j)$ is calculated based on the Gaussian pdf of $\hat{C}_{km}$ with mean $\mu = E_{\hat{C}_{km}}(s\hat{n}r)$ and variance $\sigma^2 = Var_{\hat{C}_{km}}(s\hat{n}r)$ for modulation scheme M_j. For brevity, $\hat{C}_{km_x}$ is referred to as $\hat{C}_{km}$ in the remaining of this chapter. The value of $s\hat{n}r$ in the case of Rayleigh fading reflects the effect of the fading coefficient as in (2.12).

2.3.3 Probability of Correct Classification

The ML classifier performance is mainly measured in terms of the probability of correct classification P_c. In this work, we present a new theoretical analysis for P_c within the two discussed channel models. For the set of modulation schemes M, P_c is defined as:

$$P_c = \frac{1}{N_M}\sum_{j=1}^{N_M} P_{c_{M_j}} = \frac{1}{N_M}\sum_{j=1}^{N_M} p((\hat{C}_{km} \in M_j)|D = M_j; s\hat{n}r). \tag{2.14}$$

In order to analyze P_c, the special case where the modulation set M only includes two modulation schemes is studied first, then later extended to include multiple modulation schemes.

Classification of Two Modulation Schemes in AWGN Channel

For $M = \{M_1, M_2\}$, P_c presented in (2.14) is redefined as:

$$P_{c_{AWGN}} = \frac{1}{2}p\left(\hat{C}_{km} \in M_1|M_1; s\hat{n}r\right) + \frac{1}{2}p\left(\hat{C}_{km} \in M_2|M_2; s\hat{n}r\right). \tag{2.15}$$

For two Gaussian processes, $(\hat{C}_{km}|M_1; s\hat{n}r) \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $(\hat{C}_{km}|M_2; s\hat{n}r) \sim \mathcal{N}(\mu_2, \sigma_2^2)$, the optimal test statistic is a quadratic function resulting in two thresholds:

$$\tau_{AWGN_{1,2}} = \frac{\mu_1 - \alpha\mu_2}{1-\alpha} \pm \sqrt{\frac{\alpha(\mu_1-\mu_2)^2}{(1-\alpha)^2} - \frac{\sigma_1^2 ln(\alpha)}{(1-\alpha)}}, \tag{2.16}$$

where $\alpha = \sigma_1^2/\sigma_2^2$, and $\{\mu_1, \sigma_1^2\}$ and $\{\mu_2, \sigma_2^2\}$ are the expected values and the variances of $\hat{C}_{km}$ for M_1 and M_2 respectively as functions of $s\hat{n}r$ and the observation length N as discussed in Section 2.3.1. In practice, only one τ_{AWGN} whose value is within the interval $[\mu_1, \mu_2]$ is considered. Thus, the probability of correct classification is approximated using Q-functions:

$$P_{c_{AWGN}} \simeq 0.5Q(\frac{\mu_1 - \tau_{AWGN}}{\sigma_1}) + 0.5Q(\frac{\tau_{AWGN} - \mu_2}{\sigma_2}). \tag{2.17}$$

Classification of Two Modulation Schemes in Fading Channel

For $M = \{M_1, M_2\}$ in a Rayleigh fading channel, the ML classifier uses the decision rule in (2.13). However, in order to calculate P_c, the probability of correct classification for each modulation scheme is averaged over the exponential distribution of the SNR, as follows:

$$\begin{aligned} P_{c_{RF}} &\simeq 0.5 \int_0^\infty Q(\frac{\mu_1 - \tau_{RF}}{\sigma_1}) \quad \frac{1}{\bar{\gamma}} e^{(-\gamma/\bar{\gamma})} d\gamma + 0.5 \int_0^\infty Q(\frac{\tau_{RF} - \mu_2}{\sigma_2}) \quad \frac{1}{\bar{\gamma}} e^{(-\gamma/\bar{\gamma})} d\gamma \\ &= \frac{1}{2\bar{\gamma}\sqrt{2\pi}} \int_0^\infty \int_{Z_1}^\infty exp(-\frac{z^2}{2} - \frac{\gamma}{\bar{\gamma}}) \quad dz d\gamma + \frac{1}{2\bar{\gamma}\sqrt{2\pi}} \int_0^\infty \int_{Z_2}^\infty exp(-\frac{z^2}{2} - \frac{\gamma}{\bar{\gamma}}) \quad dz d\gamma, \end{aligned} \tag{2.18}$$

where $z_1 = \frac{\mu_1 - \tau_{RF}}{\sigma_1}$, $z_2 = \frac{\tau_{RF} - \mu_2}{\sigma_2}$ and τ_{RF} is calculated in the same way as in (2.16). A closed form expression for the average probability of correct classification in (2.18) is very tedious. However, numerical evaluation of the double integrals is feasible.

2.3.4 Simulation Results

To study the performance of the ML classifier and validate the analysis of the probability of correct classification, an example is presented for classification of the set $M = \{QPSK, 16QAM\}$ using the 4^{th}-order cumulant $\hat{C}_{42}$. $(\hat{C}_{42}|QPSK; s\hat{n}r)$ and $(\hat{C}_{42}|16QAM; s\hat{n}r)$ are the Gaussian approximations of $\hat{C}_{42}$ with means $\{\mu_1, \mu_2\}$ and variances $\{\sigma_1^2, \sigma_2^2\}$ calculated for modulation schemes $\{QPSK, 16QAM\}$ respectively as functions of the estimated $s\hat{n}r$ as in (2.10).

In the simulation scenario, the node receives a signal $y(n)$ of variable length N, following both the AWGN and fading models, with average SNR in the range $[0-12dB]$. The node then estimates $\hat{C}_{42x}$ based on $\hat{C}_{42y}$

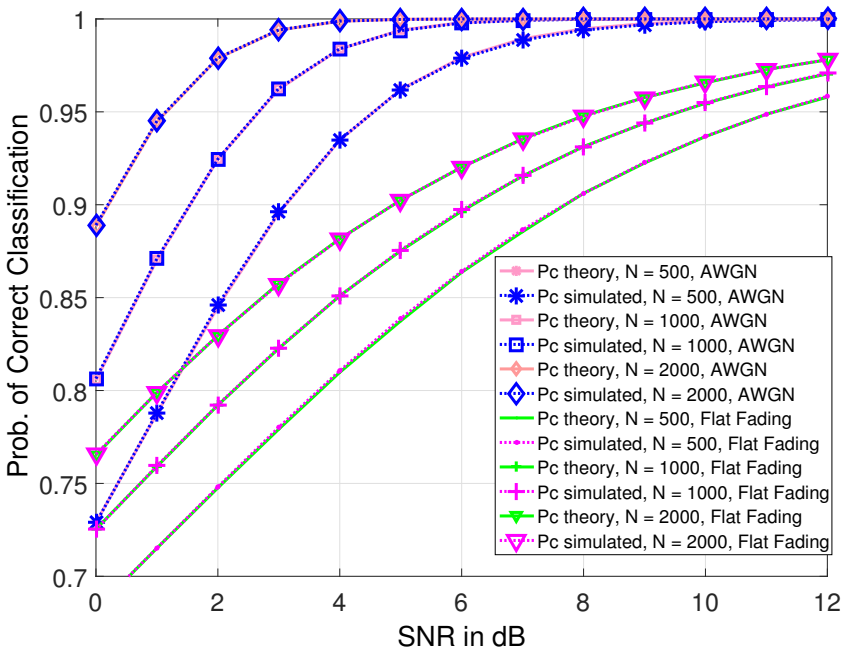

Figure 2.2: Theoretical and simulated average probabilities of correct classification for the set $\{QPSK,\ 16QAM\}$ under AWGN and flat Rayleigh fading, $N = \{500, 1000, 2000\}$, for a single node.

as explained in Section 2.3.2, calculates the probabilities $P(\hat{C}_{42x}|QPSK; \hat{snr})$ and $P(\hat{C}_{42x}|16QAM; \hat{snr})$, and then compares them according to the decision rule in (2.13). The average probability of correct classification is calculated according to (2.14). It's to be noted that the estimate of SNR is assumed to be perfect. The theoretical probabilities of correct classification are calculated according to (2.17) and (2.18). Fig. 2.2 shows the performance of the classifier for $N = \{500, 1000, 2000\}$ for both channel models. As shown, the performance of AMC degrades severely under fading channels as compared to AWGN channels. At an average SNR of 5dB, P_c reduces by at least 10%.

2.4 Hierarchical Classification for Multiple Modulation Schemes

Hierarchical approaches for modulation classification have been proposed starting in [65]. However, we propose a new hierarchical approach within the ML framework for multiple modulation classification in a new context as follows: (1) starting at the lowest-order cumulants (usually 2^{nd} order), a ML classifier generates a decision D_1 about which group of modulation schemes the received signal belongs to, (2) based on D_1, the classifier uses the true expected value of the low-order cumulants tested in step (1) to calculate the estimate of a higher order-cumulant, (3) another ML classifier generates a second decision D_2 about

which group of modulation schemes the received signal belongs to within the first group, (4) the steps (1-3) are repeated as needed. The main idea behind the hierarchical classifier is that high-order cumulants depend in their calculations on the lower-order cumulants and moments, as presented in Appendix A. Conditioning the calculations of high-order cumulants on the true expected values of lower-order cumulants results in smaller variances for the high-order cumulant estimates. The number of groups at each stage depends on the number of possible values for the cumulant tested at that step. Next, we present an example of a hierarchical classifier for four modulation schemes.

2.4.1 Classification of Four Modulation Schemes

To study the hierarchical classifier further, an example is presented to classify the set $M = \{M_1, ..., M_4\}$, where the set includes both real and complex-valued modulation schemes. The classification is performed in two stages, based on 2^{nd}- and 4^{th}-order cumulants respectively. The ML classification presented in (2.13) can be modeled as a multiple hypotheses testing problem. At each stage the classifier chooses hypothesis $\mathcal{H}_r$ that maximizes the probability $p(\mathcal{H}_r|\hat{C})$. This is equivalent to maximizing $p(\hat{C}|\mathcal{H}_r)$ given equally likely modulation schemes, where $\{\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_r\}$ is the set of possible hypotheses at this stage. The classifier node performs the following tasks:

- Stage 1: The classifier node calculates $\hat{C}_{20}$, which is equal to $\hat{C}_2$, of the received signal $y(n)$ as in (2.5, 2.6). At this stage, it's a binary hypothesis problem, with hypotheses:

$$\begin{aligned} &\mathcal{H}_0 : C_{20} = 0 \quad \{\text{Complex-baseband modulation scheme}\} \\ &\mathcal{H}_1 : C_{20} = 1 \quad \{\text{Real-baseband modulation scheme}\} \end{aligned} \tag{2.19}$$

  The classifier chooses a hypothesis such that:

$$\hat{\mathcal{H}} = \underset{H_r, r=\{0,1\}}{argmax} \left\{ p\left(\hat{C}_{20}|\mathcal{H}_r\right) \right\}, \tag{2.20}$$

  where $p(\hat{C}_{20}|\mathcal{H}_r)$ is calculated based on the asymptotic properties of $\hat{C}_{20}$.

- Stage 2: The classifier calculates the value of $\hat{C}_{42}$ as in (2.6) or $\hat{C}_4$ as in (2.5) based on the decision made in the first stage. The classifier then chooses the final hypothesis as:

$$\hat{\mathcal{H}} = \underset{H_r, r=\{0:4\}}{argmax} \left\{ p\left(\hat{C}_{42}, \hat{C}_{20}|\mathcal{H}_r\right) \right\} = \underset{H_r}{argmax} \left\{ p\left(\hat{C}_{20}|\mathcal{H}_r\right) p\left(\hat{C}_{42}|\mathcal{H}_r, \hat{C}_{20}\right) \right\}. \tag{2.21}$$

For real baseband modulation schemes, $\hat{C}_{42}$ should be replaced with $\hat{C}_4$ in (2.21).

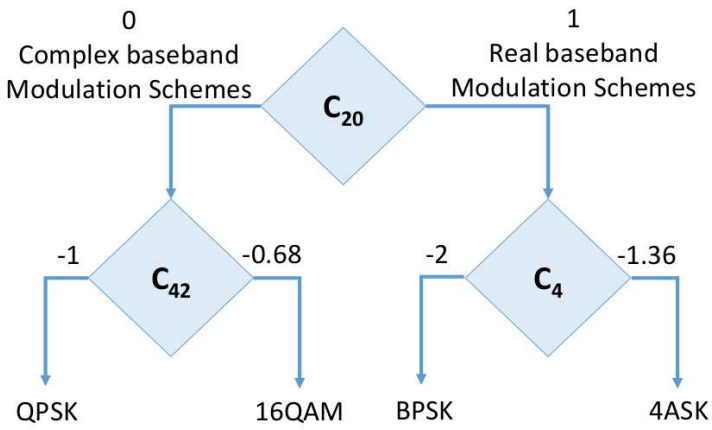

Figure 2.3: Hierarchical modulation classification system for the modulation set $M = \{BPSK, 4ASK, QPSK, 16QAM\}$. Classification is performed in two stages: $\hat{C}_{20}$ classifies real- versus complex-baseband modulation schemes, then $\hat{C}_4$ or $\hat{C}_{42}$ classifies individual schemes.

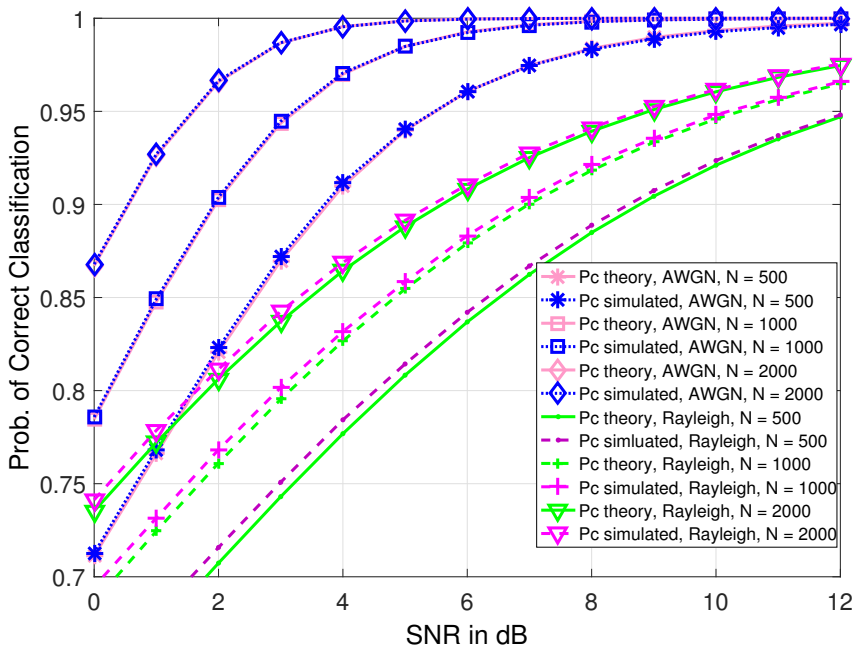

Figure 2.4: Theoretical and simulated average probabilities of correct classification for the set $\{BPSK, 4ASK, QPSK, 16QAM\}$ under AWGN and flat Rayleigh fading, $N = \{500, 1000, 2000\}$, for a single node.

2.4.2 Simulation Results

To study the performance of the hierarchical ML classifier, an example is presented for classification of the set $M = \{BPSK, 4ASK, QPSK, 16QAM\}$ under both AWGN and flat Rayleigh fading channels for different observation lengths $N = \{500, 1000, 2000\}$. The classifier architecture is shown in Fig. 2.3. Simulations are carried out following the steps presented in the previous section. The probability of correct classification is averaged over the four modulations schemes. The theoretical probabilities of correct classification are calculated depending on (2.17) and (2.18), as the classifier is considered to differentiate two modulation schemes at each stage. As shown in Fig. 2.4, the average probability of correct classification overall decreased as the number of modulation schemes increased with the same observation lengths.

2.5 Conclusions

In this chapter, we presented a comprehensive theoretical analysis of the performance of a fourth order cumulants-based modulation classifier in terms of the probability of correct classification. The classifier implemented a Maximum Likelihood (ML) approach where the non-equal variances of the estimated cumulants features were taken into consideration. The performance was studied under both AWGN and flat Rayleigh Fading Channels. In addition, a new hierarchical framework for classification of larger sets of unknown modulation schemes was presented for the single-node classifier case. Different simulation scenarios verified the proposed framework. Results showed that flat Rayleigh fading channels greatly degrades the classifier performance. The theoretical framework provides a reliable and tractable solution to the AMC problem, which can be extended to the cooperative classification in the following chapter.

Chapter 3

Cooperative Modulation Classification in Distributed Networks

Cooperative classification is presented as a means to enhance the classification performance of the single-node AMC as well as to relax the computational constraints on individual nodes. In this chapter we propose a centralized feature-level cooperative classification framework using Maximum Likelihood (ML) combining algorithm. Each node estimates a set of features, based on high-order cumulants, that are combined at a Fusion Center (FC) using ML to produce a final decision. We present new theoretical models for ML combining performance, as well as computer simulations, under both Additive White Gaussian Noise (AWGN) and flat Rayleigh fading channels. In addition, a hierarchical framework for cooperative modulation classification of larger sets of modulation schemes is also presented. Finally, the performance of the cooperative framework with correlated signals is studied. Analysis and simulations validate the theoretical models and prove the efficiency of the proposed cooperative framework under different scenarios.

3.1 Introduction

Cooperative classification is defined as sharing data and/or classification decisions between mutiple nodes to classify an unknown signal and reach an overall decision. Cooperative classification algorithms can be categorized according to different criteria, for example the categorization system suggested in [97]. In this work, we follow the data fusion algorithms categorization presented in [98] based on the cooperation architecture and the data abstraction level.

The cooperation architecture can be centralized or distributed. Centralized architectures assume the presence

of a Fusion Center (FC), to which each node transmits its data/decision and then the FC combines the shared data to reach a final decision. Most of the cooperative classification algorithms proposed in literature follow the centralized architecture, including both (1) LB approaches: Likelihood-Ratio Tests (LRT) [21, 99], Hybrid Maximum Likelihood (HML) [100, 101], using iterative optimization to search for the optimum decision rules [102], Markov Chain Monte Carlo (MCMC) sampling [103] and information theoretic measures [104] and (2) FB approaches: cyclostationarity [105], high order cumulants and moments [23, 25–30], and (3) comparisons of both [22]. Centralized approaches have also been chosen for unknown signals' parameter estimation [24]. Alternatively, in distributed architectures, the cooperating nodes reach the final decision independently through the iterative exchange of their data/decisions, thus imposing high processing power requirements on the individual nodes and increasing network overhead. Few cooperative algorithms implemented fully distributed architectures: in the HML approach presented in [31], which is later improved using online Expectation Maximization (EM) in [32, 33], and using cumulants in [106, 107].

The abstraction level of data during the fusion process can be signal, feature, or decision level. At the signal level, each sensor transmits the observed raw signal to the FC. In most signal-level combining approaches, the multiple raw signals are combined to form a single signal, and a regular single-variable AMC algorithm is applied to classify the unknown signal, as in the LB approaches in [22, 100, 108]. In [25], a different FB signal-level combining approach was presented where the signals are sent to the FC and then used within a multi-variable AMC to calculate joint high order statistical features for classification. Signal-level combining preserves the information in the signals and maintains minimal required processing at the individual nodes. However, exchanging the raw signals presents a huge overhead over the network, especially with an increasing number of cooperative nodes. On the highest level of abstraction, local decisions of individual nodes are transmitted and combined at the FC to obtain a global decision. This approach was implemented in both LB approaches presented in [21, 101, 102, 104] and FB approaches presented in [23, 26–30, 105]. Decision-level combining maintains the minimal overhead over the network, with the cost of losing all soft information in the raw data. In the intermediate level of abstraction, local features are calculated at the individual nodes and then transmitted to the FC for combining. Feature-level combining balances the trade-off between a reasonably low overhead over the network and preserving soft information from the raw signals. Few feature-level combining algorithms have been proposed in the literature, including centralized FB approaches in [26–30], distributed LB approaches in [31, 32] and distributed FB approaches in [106, 107].

In this chapter we present a cooperative cumulants-based AMC framework for distributed networks. It is a feature-level centralized algorithm; multiple nodes share high-order statistical features calculated for a received unknown signal with a FC. The FC then uses a Maximum Likelihood (ML) combining algorithm to combine the features calculated at each individual node and generate an overall decision about the modulation scheme of the unknown signal. This algorithm balances the trade-off between network overhead and preserving soft information about the unknown signal. High-order cumulants are chosen as classification

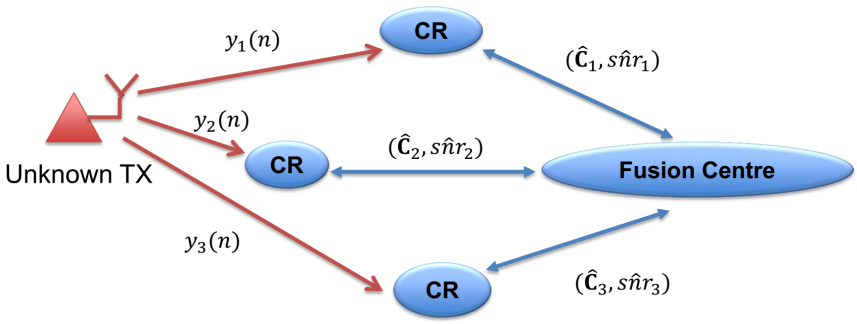

Figure 3.1: Cooperative classification system model

features due to their attractive noise immunity characteristics as discussed in Section 2.2.

The main contributions of this chapter are: (1) a theoretical analysis of the performance of cumulants-based ML cooperative classification framework for digital modulation schemes under both AWGN and flat Rayleigh fading, (2) a hierarchical framework for classification of larger sets of unknown modulation schemes in both single-node and cooperation scenarios; and (3) performance analysis of the cooperative classification framework for correlated signals. The aim of this chapter is to provide a theoretical reference for the efficiency of cooperation in signal classification scenarios.

3.2 ML Classification Combining

Cooperative classification recently has been proposed to improve the modulation classification performance, especially under fading conditions, without the need to increase the observation length N. In this section, we first present the system model adopted for cooperative cumulants-based classification based on ML. Next, the theoretical model for the average probability of correct classification is presented.

The system model adopted in this work consists of a set of N_s nodes, each receives a signal from an unknown transmitter as shown in Fig. 3.1. The received signal at each node, $y_i(n)$, is modeled as in (2.8) or (2.11) depending on the channel model. The links between the nodes and the FC are considered error free with sufficiently powerful error correcting codes. The unknown modulation formats are drawn from a set of N_M possible modulations, $M = \{M_1, M_2, ..., M_{N_M}\}$. Each node i is able to estimate the $s\hat{n}r_i$ of the received signal and the cumulants features vector $\hat{\mathbf{c}}_i$ required for classification, and then transmits the set of values of $(\hat{\mathbf{c}}_i, s\hat{n}r_i)$ to the FC, where the ML classification combining is performed to obtain the final decision.

The FC generates a decision D for the modulation scheme of the unknown transmitter, where $D \in M$, based

on the vectors of estimated features $(\hat{\mathbf{c}}, \hat{\mathbf{snr}}) = \{(\hat{\mathbf{c}}_1, \hat{snr}_1), (\hat{\mathbf{c}}_2, \hat{snr}_2), ..., (\hat{\mathbf{c}}_{Ns}, \hat{snr}_{Ns})\}$, as follows:

$$D = \underset{j}{argmax} \left\{ p\left((\hat{\mathbf{c}}, \hat{\mathbf{snr}}) \,|D = M_j \right) \right\}. \tag{3.1}$$

Assuming independence of the N_s nodes, the decision rule for ML classification combining is presented as a compound Bayesian decision problem, or what is called a Naive Bayes' Rule [109]. Equation (3.1) is extended to:

$$D = \underset{j}{argmax} \left\{ \prod_{i=1}^{N_s} p\left((\hat{\mathbf{c}}_i, \hat{snr}_i) \,|D = M_j \right) \right\}, \tag{3.2}$$

where $p((\hat{\mathbf{c}}_i, \hat{snr}_i)|D = M_j)$ is calculated based on the Gaussian pdf of each $\hat{C}_i$ with mean $\mu_{ji} = E_{\hat{C}_i}(\hat{snr}_i)$ and variance $\sigma^2_{ji} = Var_{\hat{C}_i}(\hat{snr}_i)$ for modulation scheme M_j at node i.

3.2.1 Probability of Correct Classification

As in the case of a single node, the cooperative ML classifier performance is measured in terms of the average probability of correct classification P_c. For a set of modulation schemes M, P_c is defined as:

$$P_c = \frac{1}{N_M} \sum_{j=1}^{N_M} P_{c_{M_j}} = \frac{1}{N_M} \sum_{j=1}^{N_M} p((\hat{\mathbf{c}} \in M_j)|D = M_j; \hat{\mathbf{snr}}). \tag{3.3}$$

In order to study the performance of the ML cooperative classifier, several cases are presented, starting with cooperative classification of two nodes for two modulation schemes.

3.3 Performance of Cooperative Cumulants-based AMC

3.3.1 Two Nodes, Two Modulation Schemes under AWGN, Equal SNR

For a candidate modulation set $M = \{M_1, M_2\}$, each node i receives an independent signal $y_i(n)$ following the model presented in (2.8), the decision rule in (3.2) is rewritten as:

$$p\left(\left(\hat{C}_1, \hat{snr}_1\right)|M_1\right) p\left(\left(\hat{C}_2, \hat{snr}_2\right)|M_1\right) \underset{M_2}{\overset{M_1}{\gtrless}} p\left(\left(\hat{C}_1, \hat{snr}_1\right)|M_2\right) p\left(\left(\hat{C}_2, \hat{snr}_2\right)|M_2\right), \tag{3.4}$$

where each side presents the joint probability distribution of the cumulant features $\{\hat{C}_1, \hat{C}_2\}$ calculated at nodes 1 and 2, modeled as a bi-variate independent Gaussian random variable. $\Gamma_{M_1} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{M_1}}, \boldsymbol{\Sigma}_{\mathbf{M_1}})$, and $\Gamma_{M_2} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{M_2}}, \boldsymbol{\Sigma}_{\mathbf{M_2}})$ are the joint probability distributions for each modulation scheme respectively, with

the mean vectors $\boldsymbol{\mu}_{\mathbf{M_1}} = [\mu_1, \mu_1]$ and $\boldsymbol{\mu}_{\mathbf{M_2}} = [\mu_2, \mu_2]$. For the case where both nodes are receiving the same SNR, the covariance matrices $\boldsymbol{\Sigma}_{\mathbf{M_1}}$ and $\boldsymbol{\Sigma}_{\mathbf{M_2}}$ are defined as:

$$\boldsymbol{\Sigma}_{M_1} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix}, \quad \boldsymbol{\Sigma}_{M_2} = \begin{bmatrix} \sigma_2^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}, \tag{3.5}$$

where μ_j and σ_j^2 are defined as in Section 2.3.3 for the corresponding modulation scheme. For $\sigma_1^2 < \sigma_2^2$, the probability of correct for each modulation scheme j can be approximated as:

$$P_{c_{M_j}} = \iint_A \frac{1}{2\pi |\boldsymbol{\Sigma}_{M_j}|^{1/2}} \exp\left[-\frac{[\boldsymbol{\Omega} - \boldsymbol{\mu}_{\mathbf{M_j}}]^T \boldsymbol{\Sigma}_{M_j}^{-1} [\boldsymbol{\Omega} - \boldsymbol{\mu}_{\mathbf{M_j}}]}{2} \right] d\boldsymbol{\Omega}, \tag{3.6}$$

where $\boldsymbol{\Omega} = [\omega_1 \quad \omega_2]^T$, $d\boldsymbol{\Omega} = d\omega_1 d\omega_2$. The integrals are calculated numerically over a circular region A with center at $\{r_1, r_2\}$ and radius R. The limits of integration are:

$$\omega_1 = [r_1 - R, r_1 + R], \quad \omega_2 = \left[r_2 - \sqrt{R^2 - (\omega_1 - r_1)^2}, r_2 + \sqrt{R^2 - (\omega_1 - r_1)^2} \right], \tag{3.7}$$

where

$$r_1 = r_2 = \frac{\mu_1 - \alpha\mu_2}{1 - \alpha}, \quad R^2 = \frac{2\alpha}{(1-\alpha)^2} (\mu_1 - \mu_2)^2 - 2\sigma_1^2 \frac{ln(\alpha)}{(1-\alpha)}, \quad \alpha = \sigma_1^2/\sigma_2^2. \tag{3.8}$$

An example of the pdf of $M = \{QPSK, 16QAM\}$ using the 4^{th}-order cumulant $\hat{C}_{42}$ and the decision boundary is presented in Fig. 3.2, with $snr_1 = snr_2 = 5dB$.

3.3.2 Two Nodes, Two Modulation Schemes under AWGN, Non-equal SNR

For more practical scenarios, the assumption of equal SNR is not valid. The classifier uses the same decision rule presented in (3.4). The distributions Γ_{M_1} and Γ_{M_2} are defined based on the following mean vectors and the covariance matrices :

$$\boldsymbol{\mu}_{M_1} = \begin{bmatrix} \mu_{11} \\ \mu_{12} \end{bmatrix}, \quad \boldsymbol{\Sigma}_{M_1} = \begin{bmatrix} \sigma_{11}^2 & 0 \\ 0 & \sigma_{12}^2 \end{bmatrix}, \quad \boldsymbol{\mu}_{M_2} = \begin{bmatrix} \mu_{21} \\ \mu_{22} \end{bmatrix}, \quad \boldsymbol{\Sigma}_{M_2} = \begin{bmatrix} \sigma_{21}^2 & 0 \\ 0 & \sigma_{22}^2 \end{bmatrix}, \tag{3.9}$$

where μ_{ji} and σ_{ji}^2 are the mean and the variance of the cumulant estimate for modulation scheme M_j at node i calculated as a function of $s\hat{n}r_i$. The probabilities of correct classification for each modulation scheme are defined as in (3.6). However, the integrals are calculated numerically over an elliptical region A with center at $\{r_1, r_2\}$ and radii $\{R_1, R_2\}$, where R_i is the radius in the direction of node i. The limits of integration for

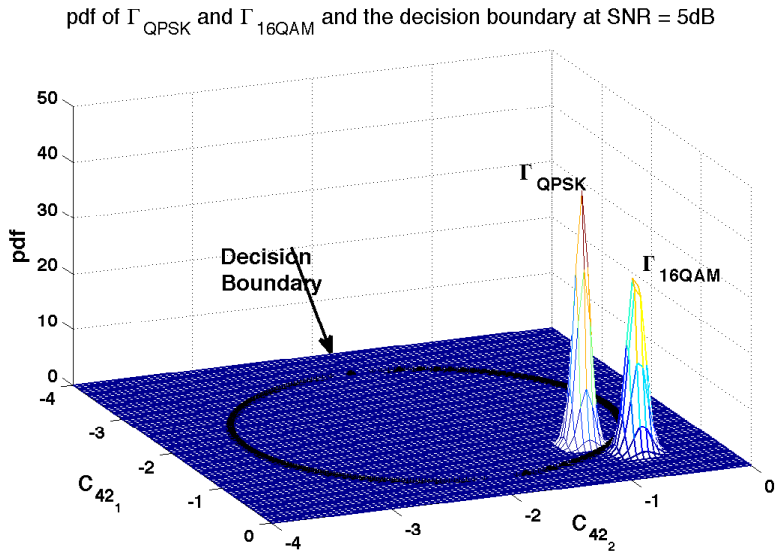

Figure 3.2: An example of the pdf of Γ_{QPAK} and Γ_{16QAM} and the circular decision boundary at snr of 5dB.

area A are:

$$\omega_1 = [r_1 - R_1, r_1 + R_1], \quad \omega_2 = \left[r_2 - \sqrt{R_2^2 - [\frac{R_2(r_1-\omega_1)}{R_1}]^2}, r_2 + \sqrt{R_2^2 - [\frac{R_2(r_1-\omega_1)}{R_1}]^2}\right], \tag{3.10}$$

where

$$\begin{aligned} r_1 &= \frac{\sigma_{21}^2\mu_{11} - \sigma_{11}^2\mu_{21}}{\sigma_{21}^2 - \sigma_{11}^2}, \quad r_2 = \frac{\sigma_{22}^2\mu_{12} - \sigma_{12}^2\mu_{22}}{\sigma_{22}^2 - \sigma_{12}^2} \\ R_1^2 &= \frac{\sigma_{21}^2\sigma_{11}^2}{\sigma_{21}^2 - \sigma_{11}^2}\rho, \quad R_2^2 = \frac{\sigma_{22}^2\sigma_{12}^2}{\sigma_{22}^2 - \sigma_{12}^2}\rho \\ \rho &= -2ln\left(\sqrt{\frac{\sigma_{11}^2\sigma_{12}^2}{\sigma_{21}^2\sigma_{22}^2}}\right) + \frac{(\mu_{11}-\mu_{21})^2}{(\sigma_{21}^2 - \sigma_{11}^2)} + \frac{(\mu_{12}-\mu_{22})^2}{(\sigma_{22}^2 - \sigma_{12}^2)}. \end{aligned} \tag{3.11}$$

An example of the pdf of $M = \{QPSK, 16QAM\}$ using the 4^{th}-order cumulant $\hat{C}_{42}$ and the decision boundary is presented in Fig. 3.3, with $snr_1 = 3dB$ and $snr_2 = 5dB$.

3.3.3 Two Nodes, Two Modulation Schemes under Flat Rayleigh Fading

For the set of two modulations schemes $M = \{M_1, M_2\}$ in a Rayleigh fading channel, the ML cooperative classifier uses the same decision rule presented in (3.4). The distributions Γ_{M_1} and Γ_{M_2} have mean vectors and covariance matrices as defined in (3.9). The average probability of correct classification for each modulation scheme is calculated through a numerical iterated integral: the joint Gaussian probability distribution

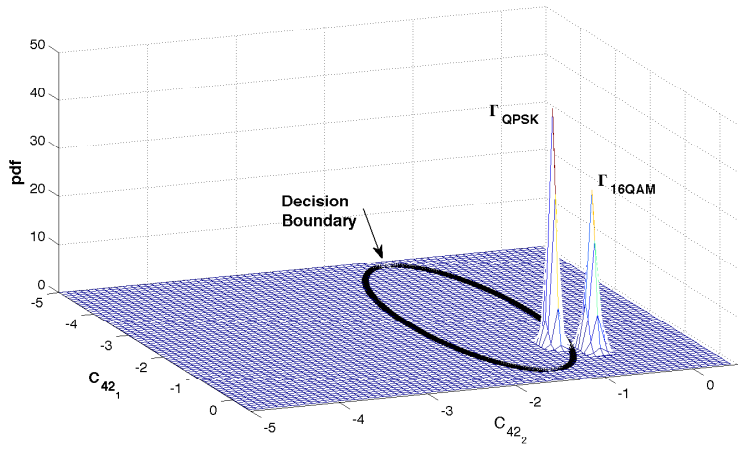

Figure 3.3: An example of the pdf of Γ_{QPAK} and Γ_{16QAM} and the elliptical decision boundary at $snr_1 = 3$dB and $snr_2 = 5$dB.

functions Γ_{M_1} and Γ_{M_2} are integrated over the exponential distributions of each node's SNR, then over the two-dimensional regions specified by the elliptical decision boundary in (3.10-3.11), as shown:

$$P_{c_{M_j}} = 1 - \iint_A \int_0^\infty \int_0^\infty \frac{1}{2\pi|\mathbf{\Sigma}_{M_j}|^{1/2}} \exp\left[-\frac{[\mathbf{\Omega} - \boldsymbol{\mu}_{M_j}]^T \mathbf{\Sigma}_{M_j}^{-1} [\mathbf{\Omega} - \boldsymbol{\mu}_{M_j}]}{2}\right] f_{\gamma_1}(\gamma_1) f_{\gamma_2}(\gamma_2) d\mathbf{\Omega} d\gamma_1 d\gamma_2, \tag{3.12}$$

where $f_{\gamma_i}(\gamma_i)$ is the exponential pdf of SNR at node i.

3.3.4 More than Two nodes

For cooperative classification from several scattered nodes, the classifier applies the general decision rule at the FC following (3.2) for N_s nodes. The joint probability distribution of the cumulants vector $\hat{\mathbf{c}}$ is modeled as a multi-variate Gaussian of N_s dimensions for each modulation scheme. The mean vector and the covariance matrix for modulation scheme j are defined as:

$$\boldsymbol{\mu}_{M_j} = \begin{bmatrix} \mu_{j1} \\ \mu_{j2} \\ \vdots \\ \mu_{jN_s} \end{bmatrix}, \quad \mathbf{\Sigma}_{M_j} = \begin{bmatrix} \sigma_{j1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{j2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{jN_s}^2 \end{bmatrix}. \tag{3.13}$$

The probability of correct classification is calculated by extending the numerical integrals of the joint Gaussian probability distribution functions to include N_s dimensions. Each added node to the network will add

one more nested integral for the case of AWGN and two more nested integrals for flat Rayleigh fading channels. The decision region A is a general hyperplane of order $(N_s - 1)$ [109]. The general form for the probability of correct classification for modulation scheme j:

$$P_{c_{M_j}} = \int_A \cdots \int \int_0^\infty \cdots \int_0^\infty \frac{1}{\sqrt{(2\pi)^{N_s}|\mathbf{\Sigma}_{M_j}|}} \exp\left[-\frac{[\mathbf{Z}-\boldsymbol{\mu}_{M_j}]^T \mathbf{\Sigma}_Q^{-1}[\mathbf{Z}-\boldsymbol{\mu}_{M_j}]}{2}\right] f_{\gamma_1}(\gamma_1)\cdots f_{\gamma_{N_s}}(\gamma_{N_s}) \, d\mathbf{Z}d\gamma_1\cdots d\gamma_{N_s}. \tag{3.14}$$

As specifying the decision region in higher-than-two dimensions Gaussian distributions is difficult, only simulated results are presented in this scenario for an increasing number of cooperative nodes under different observation lengths N.

3.3.5 Simulation Results

In order to study the performance of the ML cooperative classifier, we implemented the same example that was presented in Section 2.3.4: classification of the set $M = \{QPSK, 16QAM\}$ using the 4^{th}-order cumulant $\hat{C}_{42}$. In the simulation scenarios, each node i receives a signal $y_i(n)$ of the same length N following both the AWGN and fading models. Each node then estimates $\hat{C}_i$ and $\hat{snr}_i$, and transmits their values to the FC where the ML decision rule is applied. The average probability of correct classification is calculated according to (3.3). It's to be noted that the estimate of SNR is assumed to be perfect. The theoretical probabilities of correct classification are calculated according to the specific simulation scenario.

Fig. 3.4 shows the ML cooperative classifier performance for two independent nodes, each receiving a signal of length $N = 500$, versus a single node under both AWGN and flat Rayleigh fading. The figure shows both the theoretical probabilities of correct classification using numerical integration and simulation results. The performance of the FC with each node operating with an observation length of $N = 500$ is equivalent to (AWGN) or better than (fading) the performance of a single node receiving a signal of length $N = 1000$, as shown in Fig. 2.2 and 2.4.

Fig. 3.5 presents an example simulation for the non-equal SNR scenario. Simulations were carried out by fixing the SNR at a constant for one node (2dB), and varying the SNR for the second node over the whole range of $0 - 12$dB. Results show that ML combining always results in a better overall P_c for both nodes, although the percentage of improvement varies according to each node's SNR. While the node with the higher SNR does not obtain much improvement, the node with the lower SNR always benefits from the ML combining.

Fig. 3.6 presents the classifier performance with increasing number of nodes versus the average SNR at each node for an example of $N = 500$ under flat Rayleigh channels. With careful inspection of the simula-

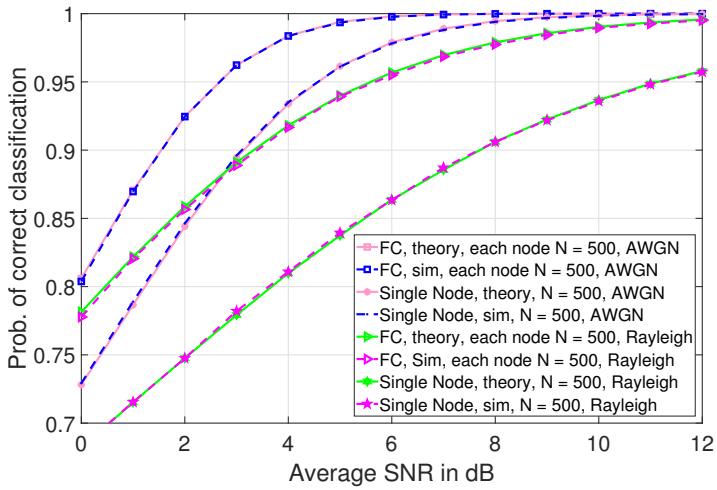

Figure 3.4: Average probability of correct classification for the set $\{QPSK, 16QAM\}$ under AWGN and flat Rayleigh fading, for a single node versus two cooperating nodes, each with $N = 500$ and equal average SNR.

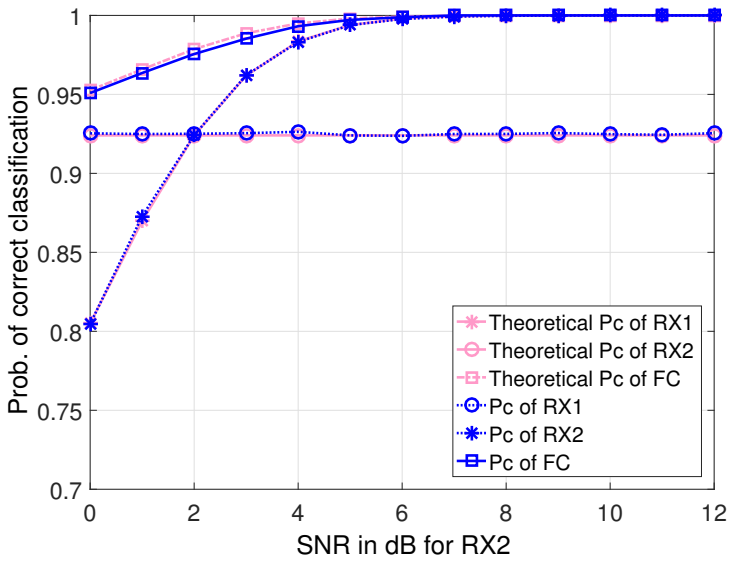

Figure 3.5: Average probability of correct classification for the set $\{QPSK, 16QAM\}$ under AWGN, $N = 1000$, with constant $snr_1 = 2dB$ and $snr_2 = 0 - 12dB$. The probability of correct for RX1 is constant.

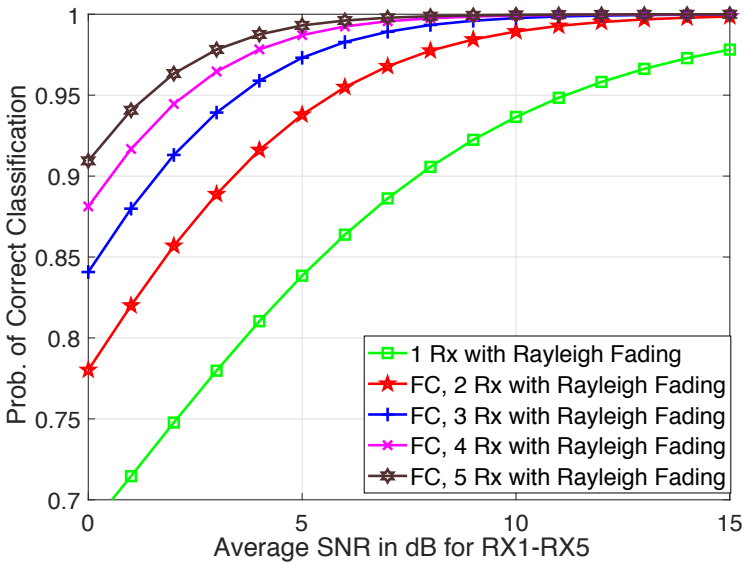

Figure 3.6: Average probability of correct classification for the set $\{QPSK, 16QAM\}$ under flat Rayleigh fading, $N = 500$, with 2, 3, 4 and 5 cooperating nodes.

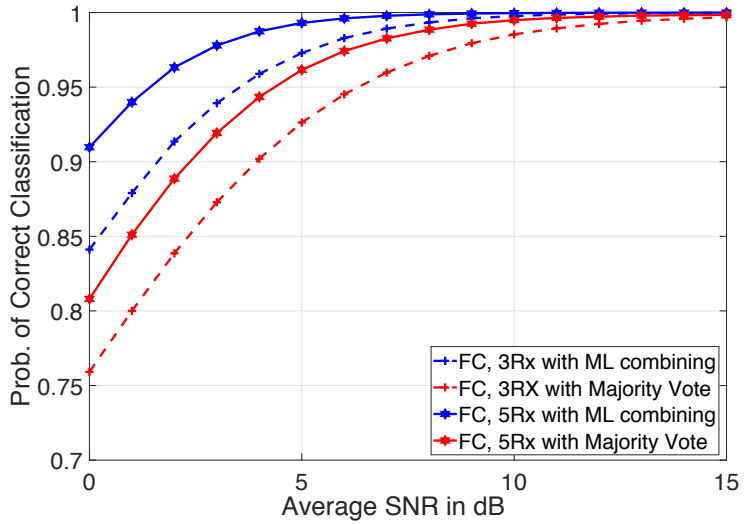

Figure 3.7: Average probability of correct classification for the set $\{QPSK, 16QAM\}$ under AWGN and flat Rayleigh fading, for 3 and 5 cooperating nodes each with $N = 500$ and equal average SNR implementing ML classification combining versus a decision majority vote.

tion results, ML cooperative classification from 5 nodes under flat Rayleigh fading outperforms the AWGN performance of a single node up to an average SNR of 15dB. To further illustrate the improvement provided by the proposed cooperative ML classifier, Fig. 3.7 presents a comparison of the proposed classifier's performance for 3 and 5 cooperating nodes versus a Majority Vote centralized classifier based on the independent decisions at each node [27]. Although hard-decision fusion presents the least network overhead, ML classification at the feature level preserves the features' soft information and improves the classification performance greatly, especially at low average SNR values.

3.3.6 Hierarchical Classification for Multiple Modulation Schemes

To enable classification of multiple modulation schemes, the FC performs the classification in a hierarchy of stages following the same approach presented in Section 2.4 with Naive Bayes' Rule combining of the feature sets from different nodes. Extending the same scenario presented in Section 2.4 into two cooperating nodes, the algorithm includes:

- Each node $\{J_1, J_2\}$ calculates the values of $\hat{C}_{20_i}$, $\hat{C}_{4_i}$ and $\hat{C}_{42_i}$, $i = 1, 2$, and transmits them to the FC.
- Stage 1: The FC uses the values of $\hat{C}_{20_i}$ to decide whether the modulation scheme is real- or complex-baseband. Based on the assumption that the two classifier nodes are independent, the combined decision for the first stage is:

$$\hat{\mathcal{H}} = \underset{H_r}{argmax} \left\{ p\left(\hat{C}_{20_1}, \hat{C}_{20_2} | \mathcal{H}_r\right) \right\} = \underset{H_r}{argmax} \left\{ p\left(\hat{C}_{20_1} | \mathcal{H}_r\right) p\left(\hat{C}_{20_2} | \mathcal{H}_r\right) \right\}. \tag{3.15}$$

- Stage 2: The FC uses the values of $\hat{C}_{42_i}$ or $\hat{C}_{4_i}$ based on the decision taken in stage 1 to decide on the final modulation scheme. For the case of complex-valued modulation schemes:

$$\begin{aligned} \hat{\mathcal{H}} &= \underset{H_r}{argmax} \left\{ p\left(\hat{C}_{42_1}, \hat{C}_{42_2}, \hat{C}_{20_1}, \hat{C}_{20_2} | \mathcal{H}_r\right) \right\} \\ &= \underset{H_r}{argmax} \left\{ p\left(\hat{C}_{42_1}, \hat{C}_{20_1}, \hat{C}_{20_2} | \mathcal{H}_r\right) p\left(\hat{C}_{42_2}, \hat{C}_{20_1}, \hat{C}_{20_2} | \mathcal{H}_r\right) \right\} \\ &= \underset{H_r}{argmax} \left\{ p\left(\hat{C}_{42_1} | \mathcal{H}_r, \hat{C}_{20_1}, \hat{C}_{20_2}\right) p\left(\hat{C}_{42_2} | \mathcal{H}_r, \hat{C}_{20_1}, \hat{C}_{20_2}\right) \left(p\left(\hat{C}_{20_1}, \hat{C}_{20_2} | \mathcal{H}_r\right)\right)^2 \right\}. \end{aligned} \tag{3.16}$$

Fig. 3.8 presents the performance of the hierarchical FC classifier for an observation length $N = 500$ under AWGN and flat Rayleigh fading channels versus a single node. The figure shows only simulation results as numerical integration is highly complex for this case. The performance of the FC is equivalent to (AWGN) or better than (fading) the performance of a single node receiving a signal of length $N = 1000$.

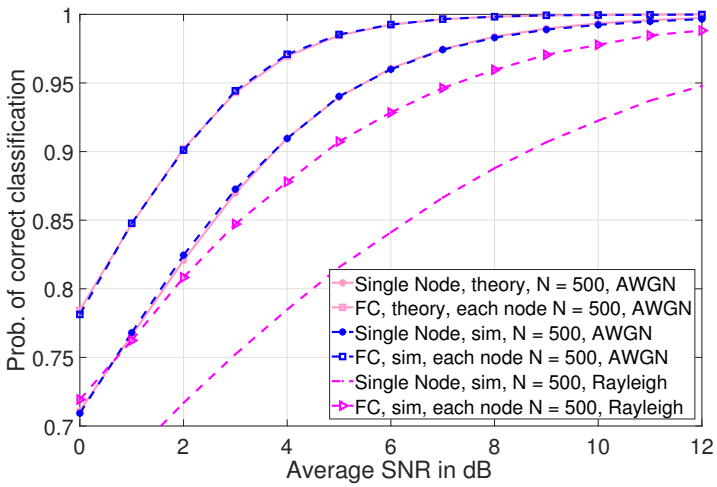

Figure 3.8: Average probability of correct classification for the set $\{BPSK, 4ASK, QPSK, 16QAM\}$ under AWGN only and AWGN plus flat Rayleigh fading, for a single node versus two cooperating nodes each with $N = 500$ and equal average SNR.

3.3.7 Remarks

In the proposed ML classification combining algorithm, the channels between the nodes and the FC are considered error free with sufficiently powerful error correcting codes which is a best-case scenario. In practical scenarios, the channels might suffer from high noise or severe fading such that the error correcting codes might not be sufficient. In this case, the FC will discard the erroneous feature values, thus degrading the performance by having one-less cooperating node.

In addition, Naive Bayes' Rule was proposed for classification combining with the assumption that the cooperating nodes are independent. This implies that the instantaneous received signal at each node $y_i(n)$ is different although originating from the same transmitter; sufficient spacing among nodes' locations implies independent noise and fading coefficients, as well as different N length frames of the transmitted data $x_i(n)$. Performance analysis proved that ML combining of cumulant features from independent nodes greatly improves overall classification performance. However, there is a probability that each of the cooperating nodes receives interleaving parts of N length frames of transmitted data. In this case, the independence assumption is not valid. In the following section, the effect of correlated received data on the performance of the cooperative classification framework is studied.

3.4 Performance of Correlated Signals in Cooperative AMC

For two nodes receiving signals from the same transmitter, a general model for the received signals under AWGN is:

$$\begin{aligned} y_1(n) &= x(n) + w_1(n), \quad n = 1, 2, ..., N, \\ y_2(n) &= x(n - \Upsilon) + w_2(n), \quad n = 1, 2, ..., N, \end{aligned} \tag{3.17}$$

where $y_1(n)$ and $y_2(n)$ are the received signals at nodes 1 and 2 respectively, and Υ is the time difference between arrivals of the same transmitted signal $x(n)$ at the two nodes. The cross-correlation function between the two received signals $\rho_{y_1y_2}$ is defined as:

$$\rho_{y_1y_2}(\tau) = E[y_1(n)y_2(n - \tau)]/\sigma_y^2. \tag{3.18}$$

At $\Upsilon = N$, the two received signals $y_1(n)$ and $y_2(n)$ are completely independent; $\rho_{y_1y_2}(\tau) \simeq 0$. In this case, the assumption of independent nodes is valid. On the other hand, at $\Upsilon = 0$, $\rho_{y_1y_2}(\tau) \simeq 1$ at $\tau = 0$. This case is referred to as perfect correlation. For any other value $\Upsilon = k$ where $0 < k < N$, $0 < \rho_{y_1y_2}(\tau) < 1$ at $\tau = k$. This case is referred to as partial correlation. For both latter cases, the independence assumption of the two nodes' received signals, as well as the application of Naive Bayes' Rule need to be investigated. This analysis is presented based on the two-nodes scenario, with a candidate set $M = \{QPSK, 16QAM\}$ at the same SNR.

3.4.1 Effect of Correlation on Joint Cumulants Probability Distributions

Taking the cross-correlation between the two received signals into consideration, the ML classification combining decision in (3.1) is valid. However, the joint probability of the estimated cumulants is a correlated bi-variate Gaussian random variable with correlation coefficient ρ. The FC decision rule is defined as:

$$p\left(\left(\hat{\mathbf{C}}_1, \hat{\mathbf{C}}_2, s\hat{n}r_1, s\hat{n}r_2\right)|QPSK\right) \underset{16QAM}{\overset{QPSK}{\gtrless}} p\left(\left(\hat{\mathbf{C}}_1, \hat{\mathbf{C}}_2, s\hat{n}r_1, s\hat{n}r_2\right)|16QAM\right), \tag{3.19}$$

where each side presents a correlated bi-variate Gaussian random variable. Let $\Gamma_{QPSK} \sim \mathcal{N}(\boldsymbol{\mu_Q}, \mathbf{\Sigma_Q})$, and $\Gamma_{16QAM} \sim \mathcal{N}(\boldsymbol{\mu_{16Q}}, \mathbf{\Sigma_{16Q}})$ be the joint Gaussian probability distribution functions of $\hat{\mathbf{C}}_{\mathbf{42}}$ for $QPSK$ and $16QAM$ modulation schemes respectively. The mean vectors of Γ_{QPSK} and Γ_{16QAM} are the theoretical values of C_{42} for $QPSK$ and $16QAM$ modulation schemes respectively; $\boldsymbol{\mu_Q} = [C_{42_Q}, C_{42_Q}]$ and $\boldsymbol{\mu_{16Q}} = [C_{42_{16Q}}, C_{42_{16Q}}]$. With correlated received signals, the covariance matrices are modeled as:

$$\mathbf{\Sigma}_Q = \begin{bmatrix} \sigma_Q^2 & \rho\sigma_{Q_0}^2 \\ \rho\sigma_{Q_0}^2 & \sigma_Q^2 \end{bmatrix}, \quad \mathbf{\Sigma}_{16Q} = \begin{bmatrix} \sigma_{16Q}^2 & \rho\sigma_{16Q_0}^2 \\ \rho\sigma_{16Q_0}^2 & \sigma_{16Q}^2 \end{bmatrix}, \tag{3.20}$$

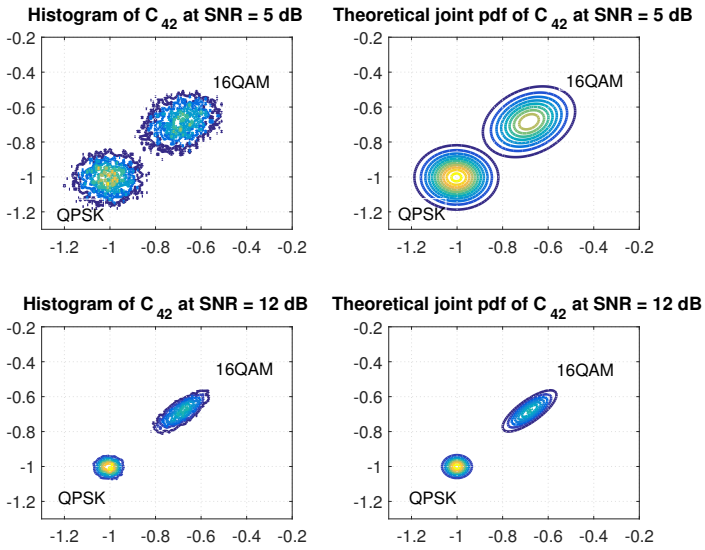

Figure 3.9: Contours of the theoretical joint pdf of $\hat{C}_{42}$ for $QPSK$ and $16QAM$ versus the simulated histogram at $\rho = 1$.

where σ^2_Q and σ^2_{16Q} are the variances of C_{42} for $QPSK$ and $16QAM$ modulation schemes under the given SNR, ρ is the cross-correlation coefficient between the two signals defined in (3.18), and $\sigma^2_{Q_0}$ and $\sigma^2_{16Q_0}$ are the variances of $\hat{C}_{42}$ for $QPSK$ and $16QAM$ modulation schemes respectively without AWGN, presented in (B.19). Note that for constant modulus modulation schemes the variance of $\hat{C}_{42}$ without AWGN, σ^2_0, equals zero, as shown in Appendix B.

Fig. 3.9 presents a comparison of the contours of the theoretical joint pdf of $\hat{C}_{42}$ for both $QPSK$ and $16QAM$ modulation schemes using the covariance model presented in (3.20), at two different values of SNR versus the histogram of $\hat{C}_{42}$ values resulting from simulations. The two received signals are perfectly correlated; $\rho = 1$. For constant modulus modulation schemes; i.e. $QPSK$, the joint pdf is not affected by correlation. For $16QAM$, the joint pdf ellipsoid's eccentricity increases with increasing SNR. As shown in the figure, simulations match the suggested model for correlated bi-variate Gaussian distribution for the cumulant estimates. Fig. 3.10 shows the eccentricity of the joint pdf ellipsoid versus increasing SNR for $16QAM$ modulation at different correlation values. With increasing correlation, the ellipsoid's eccentricity increases. The eccentricity equals zero; i.e a circle, when the signals are independent ($\rho = 0$). Eccentricity is calculated as $E = \sqrt{1-(b/a)^2}$, where (b/a) is the semiminor-to-semimajor axis ratio of the ellipse calculated as the ratio of the Eigen Values of the covariance matrix [110].

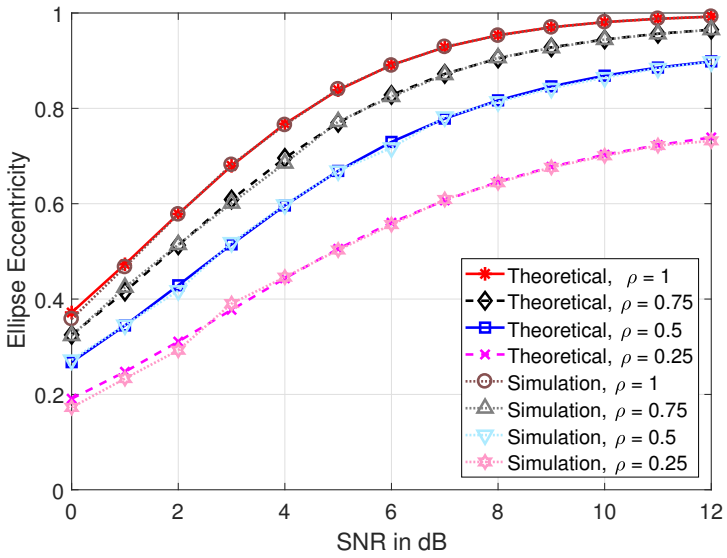

Figure 3.10: Eccentricity of the joint pdf ellipsoid of $\hat{C}_{42}$ of $16QAM$ versus SNR at different correlation values.

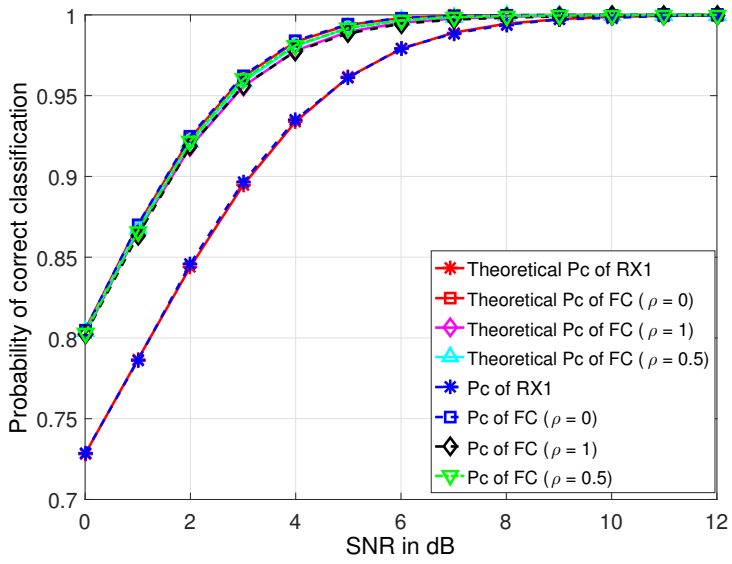

Figure 3.11: Average probabilities of correct classification of two nodes versus a single node for the set $\{QPSK, 16QAM\}$, $N = 500$, with variable correlation coefficient $\rho = [0, 0.5, 1]$.

3.4.2 Effect of Correlation on Average Probability of Correct Classification

The average probability of correct classification with correlated signals is theoretically calculated through numerical integration of the joint correlated bi-variate Gaussian distributions Γ_{QPSK} and Γ_{16QAM}. Since

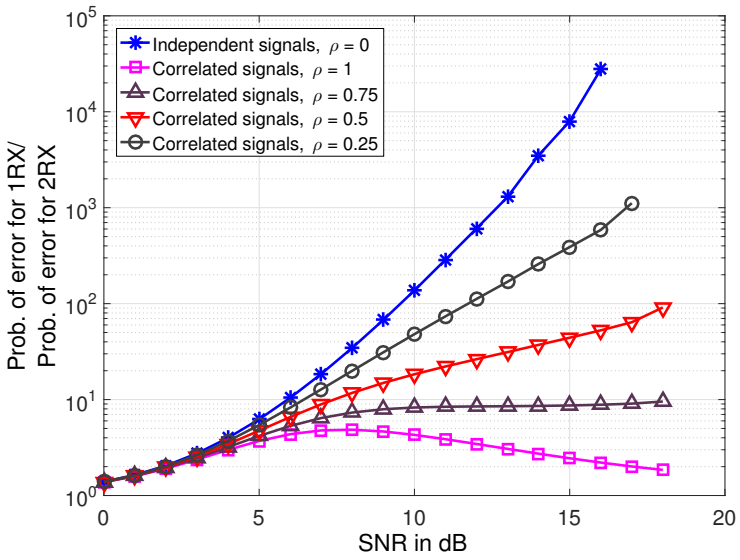

Figure 3.12: Improvement in probability of classification error for two nodes versus a single node in an AWGN channel for correlation coefficient $\rho = [0, 0.25, 0.5, 0.75, 1]$. Cooperative Classification always results in performance improvement.

the cooperating nodes only share the feature vectors, the FC will not be able to estimate the correlation coefficient in advance to adjust the decision boundary accordingly. Fig. 3.11 shows the average probability of correct classification for ML combining of 2 nodes with different correlation coefficients, assuming the independent decision boundary for both theory and simulations. Although the probability of correct classification is degraded when the two received signals are correlated while the FC assumes independence, still there is significant improvement in the overall classification performance, even at the worst-case scenario when the two nodes are receiving perfectly correlated signals.

To further investigate the performance of the ML classifier versus variable correlation scenarios, Fig. 3.12 shows the improvement in performance as a function of the probability of error for a single node divided by the probability of error for ML combining of two nodes versus SNR for different correlation values. The figure shows that up to a correlation of 0.75, there is a linear improvement in performance when implementing the ML combining framework. For the rare possibility of highly correlated signals, $\rho \approx 1$, the improvement is at best in intermediate SNR values ($SNR = 4 - 12dB$), where the ratio of the probability of error for a single node to two nodes is between 3.4 and 4.8. For very high SNR, the performance of the FC combining correlated signals from two nodes converges to the performance of a single node.

In flat Rayleigh fading scenarios, the effect of correlation is further reduced. Modeling the received signals at two nodes as in (2.11) with a time difference of arrival as in (3.17), the cross-correlation between the two signals, calculated as in (3.18), follows a Double Rayleigh Distribution [111]. According to the Cumulative

Distribution Function (CDF) of the Double Rayleigh distribution, with the same simulation parameters, the correlation between the two signals experiencing fading is less than the corresponding correlation due to overlapping signal frames for 72% of the time [112]. Having random independent fading coefficients mostly reduces the correlation between signals, pushing more towards signals independence and improving the ML classification performance.

3.5 Conclusions

In this chapter, we presented a cooperative framework for automatic modulation classification in distributed networks as a means to enhance the AMC performance, especially with weak detected signals and Rayleigh fading channels. The proposed framework is a cumulants-based centralized feature-level combining algorithm, in which cooperating nodes share locally calculated high-order cumulants features with a FC. The FC uses ML combining to generate a joint decision of the unknown modulation scheme. Feature-level combining balances the trade-off between reasonably low network overhead and preserving signal information. Both theoretical models and corresponding simulations were developed to study the performance of the proposed ML combining classifier in both AWGN and Rayleigh fading channels, for different numbers of cooperating nodes, and for multiple sets of unknown modulation schemes through a hierarchical approach. The framework provides a reliable solution to the AMC problem through sensor fusion techniques. ML classification combining results in better classification accuracy, even for extreme cases when some of the nodes are receiving signals with much lower SNR. In addition, although the benefits of ML classification combining degrades with correlated sensors, the joint classification decision accuracy still outperforms single-node performance for most practical correlation scenarios.

Chapter 4

Smartphones' Sensor Fusion for Pedestrian Indoor GraphSLAM

Simultaneous Localization and Mapping (SLAM) for pedestrians is a relatively new approach to the indoor localization problem. With the advancements in smartphone technology, pedestrian SLAM has transitioned towards fusion of smartphones' integrated sensors through Pedestrian Dead-Reckoning (PDR) techniques. In this chapter, we present a novel approach for indoor user localization, trajectory tracking and mapping through GraphSLAM: modeling the spatial structure of a user's positions as a graph optimization problem. The main contributions of this chapter are: (1) a new algorithm for calibrating the heading measurements acquired through the smartphone sensors for PDR, (2) a heading-detection stage as a pre-processing stage for GraphSLAM, and (3) the integration of these techniques into an overall GraphSLAM framework. Experiments were conducted using an iPhone 7 within an academic building with different users. The proposed algorithms were able to overcome the drift errors in heading measurements and provide accurate estimates for users' locations and movement trajectories.

4.1 Introduction

Simultaneous Localization and Mapping (SLAM) is the construction of a consistent map of a user's previously unknown surrounding environment while simultaneously estimating the user's current position within the map. It has been a heavily researched field in the robotics community for more than 3 decades [46]. The problem involves the operation of a robot with multiple sensors (specially cameras and laser scanners) within an unknown environment, where the robot performs multiple functions: position estimation, identifying environment landmarks, data association, map creation, and position and map updates. The popularity

of robotics SLAM is due to its various applications in exploring environments where it would not be possible or safe for humans to be, for example: planetary rovers, robots developed for mapping/detecting landmines and for military purposes. For more information about robotics SLAM, readers are referred to [46, 47, 113].

Adaption of SLAM techniques to pedestrian localization has gained interest over the last decade, primarily as a solution for the indoor localization problem where the Global Positioning System (GPS) and other cellular-based localization techniques fail to provide adequate accuracy. Generating building maps through Pedestrian SLAM is of importance in many applications, such as improved indoor position estimation, navigation assistance and path planning within public access buildings [114]. Another important application of pedestrian SLAM is localization and map creation for emergency responders when working through environments without *a priori* information [52, 115].

In general, the SLAM problem is defined in terms of estimating the most probable sequence of user poses (position and heading) and a building map in a predefined format. Solving the SLAM problem within a probabilistic framework can be categorized into two main approaches: filtering and smoothing [46–48]. Filtering approaches update the pose and map estimates recursively as a new measurement becomes available, with the most popular algorithms based on Extended Kalman Filters (EKF) and Particle Filters (PF). Smoothing approaches, also referred to as GraphSLAM, estimate a sequence of poses instead of just the current pose in a batch-processing approach [48].

With the increasing need for accurate indoor positioning algorithms up to the room or suite level using smartphones, tracking of pedestrian motion trajectories inside buildings has been presented in recent studies as a novel approach to improve indoor localization accuracy. In our recent work, we proposed using users' motion trajectories to detect the user's current position within a building (in terms of a room) through trajectory similarity measures [62], and Neural Networks [61]. In this work, we propose using pedestrian GraphSLAM based on smartphone sensors as a new approach for tracking mobile users' movement trajectories.

Although robotics GraphSLAM is a well-established research area, pedestrian GraphSLAM using only smartphone sensors faces its own distinct set of challenges:

1. The available sensors for pedestrian movement tracking using a smartphone are limited as compared to robots. Smartphones mainly use accelerometers, gyroscopes and magnetometers for Pedestrian Dead Reckoning (PDR) [116]. While accelerometers can provide relatively accurate estimates for step detection and estimation, heading estimation suffers from drift errors which result in trajectory rotation.
2. In order to detect the most probable movement trajectory, GraphSLAM usually exploits known assumptions of sensor measurement error distributions (mostly Gaussian distributed around the true values) [117], which does not hold for pedestrian heading measurements due to drift errors. Foot-mounted sensors provide higher sensitivity for detecting stationarity within pedestrians step cycles,

and hence overcome drift errors by applying Zero Velocity Updates (ZUPTs) as discussed in more detail in Section 4.2. However, for the less accurate and farther from the foot smartphone sensors, techniques still need to be developed to estimate heading direction measurement error distribution.

3. The infinite possibilities of smartphones' position and orientation, versus mounted sensors with structured positions, introduces even more ambiguities in heading estimation.
4. The variety of sensors in robotics-based SLAM allows for the automatic integration of loop closures and landmarks into the SLAM problem, usually through image processing and/or range measurements. However, using smartphones, novel methods for detecting loop closures and/or landmarks need to be developed, especially if detection without user interaction is required.

Few algorithms have considered the problem of pedestrian GraphSLAM using smartphones. In [118] and [119], pedestrian GraphSLAM algorithms were proposed utilizing Radio-Frequency Identification (RFID) nodes as landmarks in the former and combining WiFi Received Signal Strength (RSS) measurements with PDR in the latter. However, both algorithms used wearable devices that provide high accuracy measurements, avoiding the challenges associated with smartphones. On the other hand, other smartphones-based pedestrian SLAM algorithms have been presented, such as in [120–123], based on filtering approaches which usually require high computational capabilities to process the low-quality smartphone sensor measurements.

Smartphones-based GraphSLAM has been presented in [50, 124–126], where each proposed algorithm tried to overcome challenges (1)-(4) in different ways. In [124], a GraphSALM algorithm was proposed combining measurements from smartphone inertial sensors, WiFi, Bluetooth, LTE and Magnetic signals. The algorithm used QR codes and Near Field Communication (NFC) tags, in addition to GPS position fixes if available, as landmarks, and implemented PDR trajectory calibration based on landmarks to address the smartphone sensor challenges while maintaining the smartphone fixed inside a user's pocket. In [125], a 3D GraphSLAM algorithm was proposed for multi-floor localization in which measurements from the smartphone barometer were combined with accelerometer and gyroscope to classify a user's activity (walking, going up- or downstairs, going up or down through elevators) with different phone poses. The locations of detected stairs and elevators were used as landmarks. In [126], Gao and Harle proposed using sequences of Magnetic measurements as landmarks within a GraphSLAM algorithm based on smartphone PDR for indoor signal surveying. They also implemented a line filter to identify likely straight-line segments within the users' estimated PDR trajectories. A more efficient implementation of this algorithm was presented in [127]. Another smartphones-based GraphSLAM algorithm for signal surveying was proposed in [50] in which WiFi signal strengths received at certain locations are considered as virtual landmarks while holding the phone in a front position.

In this chapter we present a novel approach to overcome the drift in movement heading measurement error

and maintain a Gaussian error model for use in the GraphSLAM algorithm using smartphones [challenges (1) and (2)] with minimal dependency on loop closures and landmarks [challenge (4)]. The proposed algorithm is mainly different than the aforementioned ones in that: (1) it presents very good results with only one loop closure at the start and end points, which makes it practical for buildings without pre-existing infrastructure (NFC tags, WiFi APs, etc...). The single loop closure can be labeled by the user. (2) The algorithm calibrates heading measurements independently of landmarks. Other algorithms, except the one presented in [126], depend on landmarks and loop closures to overcome the drift in heading measurements, i.e. their performance is dependent on the abundance of landmarks. (3) The algorithm maintains the least squares error minimization as the underlying optimization technique for GraphSLAM, which can be solved very efficiently with minimal processing time.

The proposed approach is based on the following assumptions: (1) most building structures are rectangular, hallways and rooms are parallel and/or perpendicular, hence pedestrians' overall heading transitions inside buildings can be approximated as multiples of $\pi/2$, and (2) given the range of pedestrian position change per one step versus the regular sizes of rooms within buildings, the heading change per step can be quantized into 1 of 4 main directions, or 1 of 8 if more detail is preferred. The loss of smaller changes in angles (less than $\pi/4$) can be tolerated as they will have minimal effect on the user's location at the room level.

For challenge (3), studies investigating the effects of common smartphone placements on sensor measurements and movement tracking were presented in [128–130]. This challenge is out of the scope of this work. Throughout experiments, we assume that the pedestrian is holding the phone in front of him/her in a tilted position towards the direction of his/her movement. Other phone positions can also be tolerated as long as the phone remains in the same position during the user's movement. As for challenge (4), at least a single loop closure is considered throughout this work: the pedestrian will return to the same starting point. Some experiments included landmark positions, assumed to be known either through user input or through physical proximity to a fixed wireless-enabled device, such as a Bluetooth enabled printer in an office building. Consideration of additional loop closures and landmarks will be presented in future work.

In Section 4.2, we introduce PDR measurement models for smartphones, and propose a new algorithm for heading estimation calibration to overcome drift error. In Section 4.3, we propose a new heading-detection pre-processing stage for GraphSLAM. In Section 4.4, details about data collection and implementation of the proposed GraphSLAM are presented along with measured performance. Finally, conclusions and suggested future works are discussed in Section 4.5.

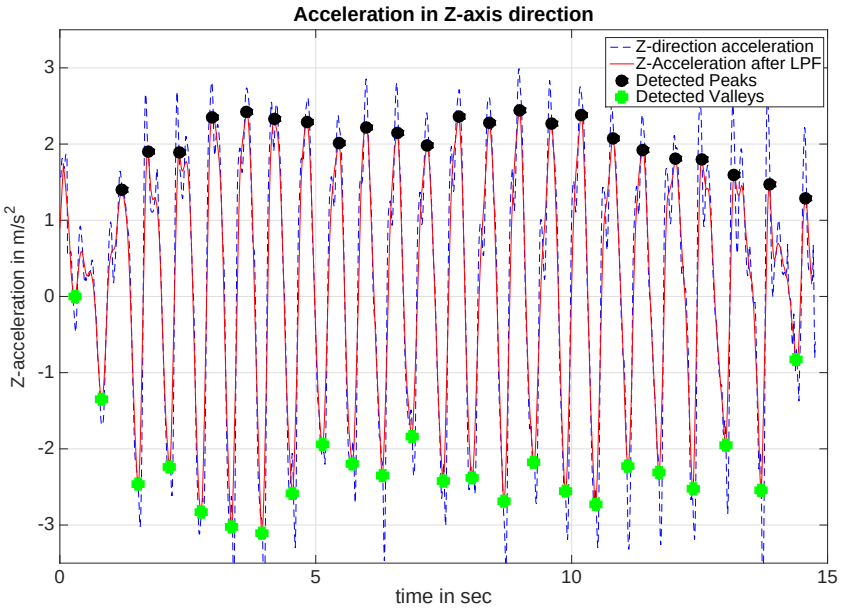

Figure 4.1: Z-axis acceleration signal for a 15 s straight walk and a sampling rate $F_s = 100$ Hz, after rotation and removing earth's gravity component, acquired from an iPhone 7.

4.2 Smartphones-based PDR

Dead-Reckoning (DR) is a relative position estimation technique. Instead of estimating a user's absolute location, DR estimates the displacement from a known starting point through Inertial Navigation Systems (INS). In 2D Pedestrian Dead-Reckoning (PDR), the displacement estimates usually include the changes in the x- and y-coordinates, or the pedestrian's step (or stride) length and the change in heading direction which is a special category of Inertial Measurement Units (IMUs) referred to as Step-and-Heading Systems (SHS) [116, 131, 132]. General IMUs use tri-axial accelerometers and gyroscopes for position and orientation estimation through a triple integration: double integration of the accelerometer measurements estimates the position change and integration of the gyroscope's angular velocity measurements estimates orientation [116]. This causes drift errors in position estimates that grow cubically in time. To regulate the drift errors, a common approach is to introduce absolute measurements into the system to close the integration loop. This has been done through the incorporation of measurements from other sensors, such as Magnetometers, or through applying Zero Velocity Updates (ZUPTs) and Zero Angular Rate Updates (ZARU) in foot-mounted sensors [133].

SHS provide an alternative to estimating the 2D displacement avoiding the triple integration. SHS provide algorithms for step detection and estimation directly from accelerometer measurements, thus avoiding the

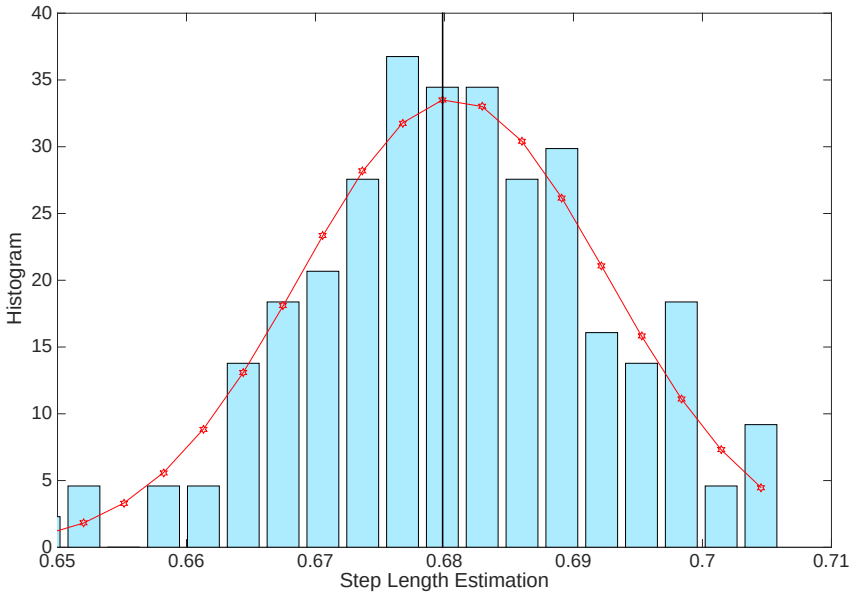

Figure 4.2: Histogram of step-length estimation for a true step size of 0.68 m using Kim algorithm [1].

double integrals required for displacement estimation, and resulting in linearly growing drift errors instead. SHS systems involve two main functions: (1) step detection and step length estimation, and (2) heading direction estimation. These functions can be performed efficiently on most of the recent off-the-shelf smartphones that contain accelerometers, gyroscopes and magnetometers.

4.2.1 Step Detection and Estimation

For step detection and step length estimation, the repetitive nature of human walking (or running) generates periodic patterns in the accelerometer measurements, most strongly in the vertical axis direction. Peaks in the vertical acceleration correspond to individual steps [134]. The signals acquired from the smartphone accelerometer are processed as presented in [45]: (1) acceleration signals in the x-, y- and z-direction are rotated through a rotation matrix based on the acquired {yaw, pitch, roll} angles to obtain the acceleration component in the true vertical axis (referred to as z-axis hereafter), (2) the earth's gravity acceleration component is removed from the z-axis acceleration, and (3) the signal is passed through a Low Pass Filter (LPF) to reduce random noise. Fig. 4.1 shows an example of the z-axis acceleration after processing using an iPhone 7. A moving average filter is used with window size w of 15 samples. Each peak corresponds to a step event.

Next, the step length l_k is estimated from the z-axis acceleration data at each time instant t_k where a step event was detected, $k = \{0, 1, 2, ..., K\}$. If a step event is not detected for a specific time duration, the user will be considered at a stop. Step length varies up to $\pm 40\%$ between different pedestrians, mainly because of height differences at a given walking speed [135]. In addition, a pedestrian's own step-length can vary up to $\pm 50\%$ according to walking speed, thus the need for accurate step-length estimation techniques. Several SHS systems suggested different algorithms, based on: the 4^{th} root of peak-to-peak acceleration [45, 135], the 3^{rd} root of average acceleration [1], average-to-peak acceleration [136], step frequency [130], and average speed [137].

In this work, experiments were conducted with different users applying each of the step length estimation algorithms mentioned above. The objective was to determine the algorithm that minimized the error in step length estimation, and at the same time maintained a Gaussian error distribution. The algorithms presented in Kim [1] and Scarlet [136] provided the best results. Fig. 4.2 shows an example of the step length estimation error using Kim [1] for a true step length of 0.68 m. This error can be modeled as Gaussian, $\epsilon_l \sim \mathcal{N}\{0, \sigma_l^2\}$, and $\sigma_l \approx 0.012$ m. To further test for normality, the kurtosis and skewness tests were performed [138]. Both values (kurtosis = 0.17 and skewness = -0.23) are accepted to represent a normal distribution with 95% confidence.

4.2.2 Heading Direction Estimation

After step detection and step length estimation, the heading direction is estimated. Smartphones can usually provide heading measurements through both magnetometers and gyroscopes [116, 139]. Magnetometers provide absolute headings (angle relative to earth's magnetic north), while gyroscopes provide a set of {yaw, pitch, roll} angles relative to the phone's axes. Some heading estimation algorithms, such as in [45], propose combining measurements from both.

However, magnetometer measurements inside buildings are greatly impacted by the presence of magnetic materials, such as electronic devices, mechanical and electronic infrastructures and different building materials [140]. There is usually a magnetic offset that is a function of a given location [141]. Measurements in [141] showed a magnetic field offset of up to $\pm 15^\circ$ in 90% of the locations inside the test building, sometimes going as high as 30°. Through our measurements using an iPhone 7 inside our office building, there was a magnetic offset of up to $\pm 30^\circ$. Thus, for heading direction estimation we followed the approach presented in [45] where the magnetometer measurements were only used when they presented high correlation with the gyroscope measurements, and only the relative changes in angle measurements were used, not the absolute heading values. Heading values θ_k are calculated between consecutive steps and relative to the starting forward movement direction, i.e. the average heading measurements for the first recorded second is

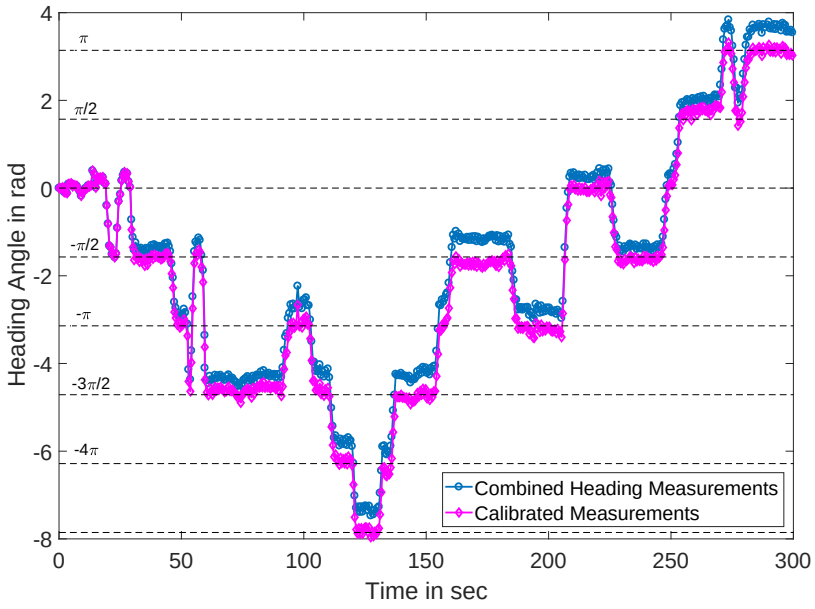

Figure 4.3: Heading direction estimation for a 5 min walk inside an office building from smartphone sensors before (blue) and after calibration every 30 s (magenta).

considered 0° heading [1]. At each detected step time instant k, there is a step length l_k and a corresponding heading direction θ_k.

An example of the resulting heading measurements, as a combination of measurements from both the magnetometer and gyroscope for a 5-minute walk is shown in Fig. 4.3 [blue dots]. The figure shows how the linear drift causes an increasing (or decreasing) average of heading measurements even for constant true heading direction. Fig. 4.4 (left side) shows the effect of the drift error on the heading direction estimation for the same walk by clustering the estimated heading measurements. The true heading values which are multiples of $\{0, \pi/2, -\pi/2, \pi\}$ are rotated as shown in the detected cluster centers. This results in rotated PDR trajectories as shown in the example in Fig. 4.6.b. In addition, modeling the heading measurement error as Gaussian with the true value as the mean is not possible with the drift error.

To overcome the linear drift in heading measurements, we propose a calibration approach for heading measurements inside rectangular architecture buildings. Through inspection of heading direction transitions, as in Fig. 4.3, and the fact that a high percentage of building architectures adhere to the same rectangular shapes (a part of an academic building floor plan is presented in Fig. 4.6 and 4.7), heading measurements with minimal changes over n successive steps, representing forward movement, can be modeled as multiples of $\pi/2$. Accordingly, a calibration approach is proposed where the heading measurements are calibrated

[1] The heading angle in this work is measured clockwise starting from the direction of movement, i.e. going straight means a heading angle of 0^o, while turning to the right means a heading angle of 90^o.

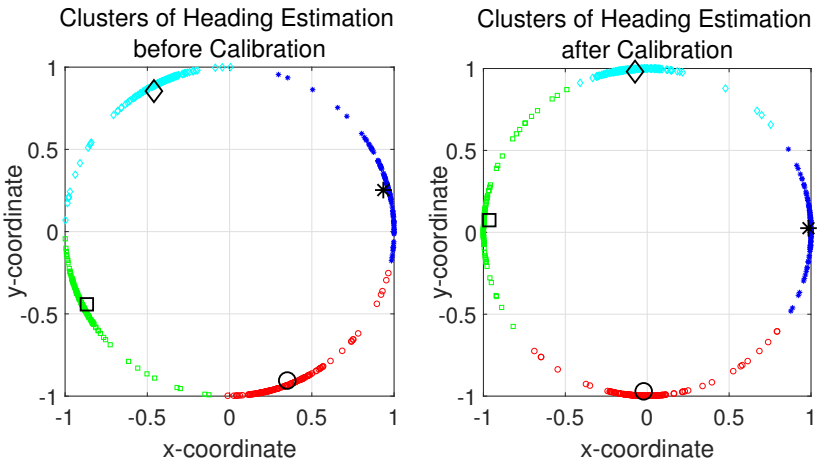

Figure 4.4: Clusters of Heading Direction estimation before and after calibration every 30 s for a 5 min walk.

every W seconds by snapping to the closest multiple of $\pi/2$. The algorithm is presented as Algorithm 1. For a window size of $W = 30$ s, and a number of successive steps $n = 5$, calibrated heading measurements are shown in Fig. 4.3 for the same 5 min walk [magenta crosses]. Fig. 4.4 (right side) shows the effect of calibration on the heading direction estimated angles, where the measurements are now centered around the values of $\{0, \pi/2, -\pi/2, \pi\}$. The calibration results in an error distribution around values of $\{0, \pi/2, -\pi/2, \pi\}$ that can be reasonably modeled as Gaussian as will be discussed in more detail in the following section. The values of kurtosis (0.63) and skewness (-0.32) pass the normality test, although they indicate more mean-concentrated values and left skewness which is expected as the drift error is reduced but not eliminated.

4.3 Pedestrian GraphSLAM using Smartphones-based PDR

GraphSLAM algorithms are based on modeling the spatial structure of the user's position measurements and the building landmarks (if any existed) in a graph format, followed by an optimization technique to detect the most probable sequence of positions under some constraints [48]. In this work, at each time instant t_k, a user's position vector $\mathbf{p}_k$ is defined as:

$$\mathbf{p}_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} + \begin{bmatrix} l_k sin(\theta_k) \\ l_k cos(\theta_k) \end{bmatrix} \tag{4.1}$$

Algorithm 1 Basic Heading Calibration Algorithm

Input
$\theta_{1:K}$ vector of heading measurements
$\mu_\theta = \{0, \pi/2, -\pi/2, \pi\}$ calibrated heading directions
Output $\theta_{1:K}$ calibrated heading measurements

noFrames = $\lceil \frac{K}{W} \rceil$
for $i = 2$ to noFrames **do**
    $F_{i-1} = \theta_{(i-2)W+1:(i-1)W}$ ▷ Previous Frame
    ▷ Check direction of movement of last n steps
    if abs(diff($F_{i-1}(end-(n-1):end)$)) < 0.2 **then**
        $\mu =$ mean$(F_{i-1}(end-(n-1):end))$
        ▷ find closest heading calibration
        $\mu^*_\theta = \arg\min(\mu - \mu_\theta)$
        ▷ Update next frames based on calibration value
        $\theta_{(i-1)W+1:end} = \theta_{(i-1)W+1:end} - \mu^*_\theta$
    end if
end for

where l_k is the estimated step length, and θ_k is the estimated heading angle. Each user position $\mathbf{p}_k$ and map landmark position $\mathbf{l}_i$ of a set of landmarks $\mathcal{L}$ is modeled as a node in the graph, while measurements between the user's positions and landmarks are presented as edges, as shown in Fig. 4.5. An edge between any two nodes models the spatial constraint relating the two nodes based on the probability distribution of their relative transformation.

For each edge in the graph connecting two nodes, corresponding to two positions $\mathbf{p}_i$ and $\mathbf{p}_j$, $\mathbf{d}_{ij}$ represents the measured displacement between the two nodes using the smartphone PDR, while $\hat{\mathbf{d}}_{ij}$ is the true displacement. For two successive nodes $\mathbf{p}_{k-1}$ and $\mathbf{p}_k$, the displacement vector between the two nodes is defined as:

$$\mathbf{d}_{k-1,k} = \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} l_k sin(\theta_k) \\ l_k cos(\theta_k) \end{bmatrix} \tag{4.2}$$

The displacement error representing the spatial constraint between each two nodes is defined as:

$$\mathbf{e}_{ij}(\mathbf{p}_i, \mathbf{p}_j) = \hat{\mathbf{d}}_{ij} - \mathbf{d}_{ij} = (\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j) - \mathbf{d}_{ij}, \tag{4.3}$$

where $\hat{\mathbf{p}}_i$ is the true position at time instant t_i.

The objective of GraphSLAM is to find the most probable sequence of positions $\mathbf{p}^* = \{\mathbf{p}^*_1, ..., \mathbf{p}^*_K\}$ that fits the PDR measurements while minimizing the displacement error over all edges. In a probabilistic formulation, GraphSLAM is considered a Maximum Likelihood (ML) estimation problem: the objective is to estimate the sequence of positions $\mathbf{p}^*$ that maximize the posterior probability $p(\mathbf{p}|\mathbf{d})$, where $\mathbf{p}$ is

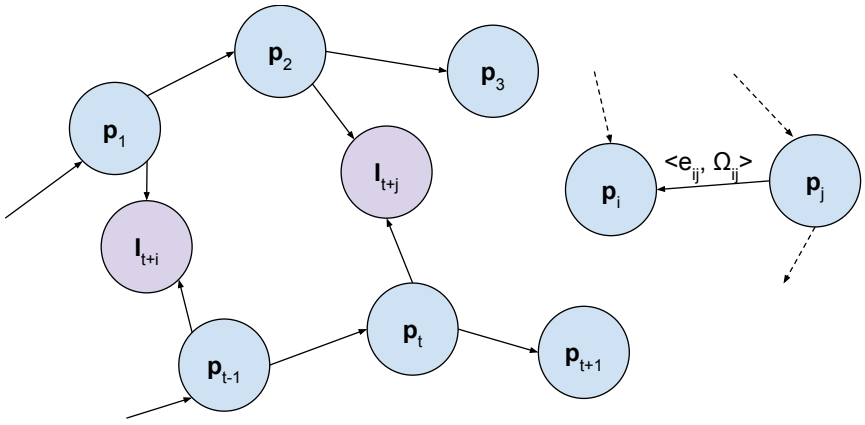

Figure 4.5: Pose-graph presentation of SLAM problem. Blue nodes correspond to user's positions, pink nodes correspond to landmark positions, and edges model the spatial constraints between nodes.

the position vector $[\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_K]$, and $\mathbf{d}$ is the measurement vector which is the concatenation of all corresponding measurement values $\mathbf{d}_{ij}$. Following a log-likelihood approach, with the assumption that the measured displacement $\mathbf{d}_{ij}$ is independent and Gaussian distributed with mean equal to the true value $\hat{\mathbf{d}}_{ij}$ and co-variance matrix $\mathbf{\Sigma_{ij}}$, the GraphSLAM ML problem is formulated as a linear Least Squares (LS) error minimization problem [142]:

$$\mathbf{p}^* = \underset{p}{argmin}\ F(\mathbf{p}) = \underset{p}{argmin} \sum_{i,j} \mathbf{e}_{ij}^T \mathbf{\Sigma}_{ij}^{-1} \mathbf{e}_{ij}, \tag{4.4}$$

where $\mathbf{e}_{ij}$ is defined as in (4.3) and is normally distributed with zero mean and co-variance $\mathbf{\Sigma_{ij}}$, and $[.]^T$ refers to matrix transpose. The solution to the LS optimization problem in (4.4) can be reduced to solving a set of linear equations [143]. The cost function in (4.4) is represented in matrix form as:

$$F(\mathbf{p}) = [\mathbf{Hp} - \mathbf{d}]^T \mathbf{\Sigma}^{-1} [\mathbf{Hp} - \mathbf{d}] \tag{4.5}$$

where $\mathbf{H}$ is the incidence matrix with all entries being 1, -1 or 0 based on the nodes' connections. The optimum value of $\mathbf{p}^*$ that minimizes the function $F(\mathbf{p})$ is given by [143, 144]:

$$\mathbf{p}^* = (\mathbf{H}^T \mathbf{\Sigma}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{\Sigma}^{-1} \mathbf{d}. \tag{4.6}$$

Computationally efficient algorithms for calculating $\mathbf{p}^*$ are widely available, especially with the exploitation of the sparsity of matrix $\mathbf{H}$ [143]. However, the proposed LS optimization formulation is based on the Gaussian assumption of the displacement measurements around the true values, which, as presented in (4.2), is dependent on both step length and heading measurements models.

In Subsection 4.2.1, the step length measurement was modeled as Gaussian, that is:

$$l_k = \dot{l} + \epsilon_l, \quad \text{where} \quad \epsilon_l \sim \mathcal{N}\{0, \sigma_l^2\}, \tag{4.7}$$

where $\dot{l}$ is the true step length. However, in Subsection 4.2.2, it was shown that the heading direction measurement error was modeled as Gaussian only after calibration relative to the set $\{0, \pi/2, -\pi/2, \pi\}$. In order to define a model for the measurement displacement, we propose a heading detection pre-processing stage for the GraphSLAM algorithm.

4.3.1 Heading detection

As discussed in Subsection 4.2.2, the majority of heading directions inside buildings can be modeled as multiples of $\pi/2$. Thus, we propose a heading detection pre-processing stage to detect the most probable true value for every estimated heading direction after calibration $\theta_{k_{cal}}$. The heading detection stage is modeled as a maximum likelihood detector. For the set of possible headings $\Theta = \{\phi_1, ..., \phi_4\} = \{0, \pi/2, -\pi/2, \pi\}$ the detector chooses the heading ϕ_n at each time instant k that has the highest probability of being the true value after calibration:

$$\theta_{opt} = \underset{n}{argmax}\{p(\theta_{k_{cal}} = \phi_n)\}. \tag{4.8}$$

Assuming that the probability of moving in any of the 4 directions is equal at every step, equation (4.8) results in a simple closest-distance detector. After detecting the optimum heading values, θ_{opt}, the heading measurement is modeled as:

$$\theta_k = \dot{\theta}_{opt} + \epsilon_\theta, \quad \text{where} \quad \epsilon_\theta \sim \mathcal{N}\{0, \sigma_\theta^2\}, \tag{4.9}$$

where $\dot{\theta}_{opt} = \{0, \pi/2, -\pi/2, \pi\}$ and $\sigma_\theta \approx 0.14$ according to our experiments. In other words, the algorithm assumes that the true heading direction at any time instant k can only be one of the four directions in the set Θ.

It's to be noted that implementing this stage will result in losing some details about movement transitions; i.e. any true heading value that is not in Θ will be rounded to the nearest value. However, experiments showed that the effect will be minimal on the overall movement trajectory and the location of the users at the room level. If more details about heading direction transitions need to be preserved, we suggest including multiples of $\pi/4$ directions as well in the set Θ. The distribution of measurements around the true values of multiples of $\pi/4$ was not calculated in this work. The value of σ_θ that was calculated for heading directions of multiples of $\pi/2$ was assumed to be also the standard deviation value for all other true heading directions. The experiments proved the validity of this assumption, as explained in Section IV. Experiments

Table 4.1: Assumed Displacement Distributions in terms of Step-length and Heading Angle Distribution Functions

Heading pdf	Approximations around θ_{opt}	Approximation pdf	Displacement pdf
$\dot{\theta}_{opt} = 0$	$sin(\theta_k) \approx \theta_k$	$f_{\theta_x}(\theta_k) \sim \mathcal{N}\{0, \sigma_\theta^2\}$	$D_x \sim \mathcal{N}\{0, \dot{l}^2\sigma_\theta^2\}$
$\theta_k \sim \mathcal{N}\{0, \sigma_\theta^2\}$	$cos(\theta_k) \approx 1 - \frac{\theta_k^2}{2}$	$f_{\theta_y}(\theta_k)$ is a scaled and shifted central χ^2 RV, $\theta_k^2 \sim \chi^2\{\sigma_\theta^2, 2\sigma_\theta^4\}$	Using Taylor Approx. $\mu_{D_y} = \dot{l}(1 - \frac{\sigma_\theta^2}{2})$, $\sigma_{D_y}^2 = (1 - \frac{\sigma_\theta^2}{2})^2\sigma_l^2 + \frac{\dot{l}^2\sigma_\theta^4}{2}$
$\dot{\theta}_{opt} = \frac{\pi}{2}$	$sin(\theta_k) \approx 1 - \frac{(\theta_k - \frac{\pi}{2})^2}{2}$	$f_{\theta_x}(\theta_k)$ is a scaled and shifted non-central χ^2 RV, $\Upsilon = (\theta_k - \frac{\pi}{2})^2 \sim \chi^2\{\sigma_\theta^2 + (\frac{\pi}{2})^2, 2\sigma_\theta^4 + 4\sigma_\theta^2(\frac{\pi}{2})^2\}$	Using Taylor Approx. $\mu_{D_x} = \dot{l}(1 - \frac{\mu_\Upsilon}{2})$, $\sigma_{D_x}^2 = (1 - \frac{\mu_\Upsilon}{2})^2\sigma_l^2 + \frac{\dot{l}\sigma_\Upsilon^2}{4}$
$\theta_k \sim \mathcal{N}\{\frac{\pi}{2}, \sigma_\theta^2\}$	$cos(\theta_k) \approx -\theta_k + \frac{\pi}{2}$	$f_{\theta_y}(\theta_k) \sim \mathcal{N}\{0, \sigma_\theta^2\}$	$D_y \sim \mathcal{N}\{0, \dot{l}^2\sigma_\theta^2\}$
$\dot{\theta}_{opt} = -\frac{\pi}{2}$	$sin(\theta_k) \approx -(1 - \frac{(\theta_k + \frac{\pi}{2})^2}{2})$	$f_{\theta_x}(\theta_k)$ is a scaled and shifted non-central χ^2 RV, $\Upsilon = (\theta_k + \frac{\pi}{2})^2 \sim \chi^2\{\sigma_\theta^2 + (\frac{\pi}{2})^2, 2\sigma_\theta^4 + 4\sigma_\theta^2(\frac{\pi}{2})^2\}$	Using Taylor Approx. $\mu_{D_x} = -\dot{l}(1 - \frac{\mu_\Upsilon}{2})$, $\sigma_{D_x}^2 = (1 - \frac{\mu_\Upsilon}{2})^2\sigma_l^2 + \frac{\dot{l}\sigma_\Upsilon^2}{4}$
$\theta_k \sim \mathcal{N}\{\frac{\pi}{2}, \sigma_\theta^2\}$	$cos(\theta_k) \approx -\theta_k + \frac{\pi}{2}$	$f_{\theta_y}(\theta_k) \sim \mathcal{N}\{0, \sigma_\theta^2\}$	$D_y \sim \mathcal{N}\{0, \dot{l}^2\sigma_\theta^2\}$
$\dot{\theta}_{opt} = \pi$	$sin(\theta_k) \approx -\theta_k + \pi$	$f_{\theta_x}(\theta_k) \sim \mathcal{N}\{0, \sigma_\theta^2\}$	$D_x \sim \mathcal{N}\{0, \dot{l}^2\sigma_\theta^2\}$
$\theta_k \sim \mathcal{N}\{\pi, \sigma_\theta^2\}$	$cos(\theta_k) \approx -(1 - \frac{(\theta_k - \pi)^2}{2})$	$f_{\theta_y}(\theta_k)$ is a scaled and shifted non-central χ^2 RV, $\Upsilon = (\theta_k - \pi)^2 \sim \chi^2\{\sigma_\theta^2 + \pi^2, 2\sigma_\theta^4 + 4\sigma_\theta^2\pi^2\}$	Using Taylor Approx. $\mu_{D_y} = \dot{l}(1 - \frac{\mu_\Upsilon}{2})$, $\sigma_{D_y}^2 = (1 - \frac{\mu_\Upsilon}{2})^2\sigma_l^2 + \frac{\dot{l}\sigma_\Upsilon^2}{4}$

also suggested that including more possibilities for heading direction will not gain further benefit.

4.3.2 Displacement Measurement Models

Assuming independence of the step length and heading angle measurements, the probabilities of measuring displacements d_x and d_y, defined in (4.2) can be modeled as functions of two random variables l_k and θ_k. Assuming that the error values in the heading angle measurement are sufficiently small $(\theta_k - \dot{\theta}_{opt})$, appropriate approximations could be used for $sin(\theta_k)$ and $cos(\theta_k)$.

Table 4.1 presents the displacement distributions $f_{D_x}(dx)$ and $f_{D_y}(dy)$ after the heading detection stage. Both are considered as the product of functions of two random variables: $f_{D_x}(dx) = f_L(l_k)f_{\theta_x}(\theta_k)$ and $f_{D_y}(dy) = f_L(l_k)f_{\theta_y}(\theta_k)$. From (4.7) and (4.9), $l_k \sim \mathcal{N}\{\dot{l}, \sigma_l^2\}$, while $\theta_k \sim \mathcal{N}\{\dot{\theta}_{opt}, \sigma_\theta^2\}$ respectively. Functions of θ_k depend on the detected value $\dot{\theta}_{opt}$ in the heading detection stage. Accordingly, modeling the displacements $f_{D_x}(dx)$ and $f_{D_y}(dy)$ is conditioned on the value of $\dot{\theta}_{opt}$.

The second column in Table 4.1 shows the corresponding approximations for small error values for the set Θ. For cases when $sin(\theta_k)$ or $cos(\theta_k)$ can be linearly approximated around θ_{opt}, $f_{\theta_x}(\theta_k)$ or $f_{\theta_y}(\theta_k)$ can be directly modeled as Gaussian, and the resulting displacement d_x or d_y can also be modeled as Gaussian as presented in [145]. On the other hand, when the approximation for $sin(\theta_k)$ or $cos(\theta_k)$ are 2^{nd}-order , $f_{\theta_x}(\theta_k)$

or $f_{\theta_y}(\theta_k)$ are scaled and shifted central (or non-central) χ^2 distributions. The resulting displacement d_x or d_y are the product of a Gaussian and a χ^2- random variable which cannot be approximated in closed form expressions [146]. In this case, Taylor series approximations were used to estimate the moments of the resulting distributions as in [147]. The last column in Table 4.1 presents the approximated distributions for the displacements d_x or d_y based on the heading values after detection θ_{opt}, which are used in implementing the GraphSLAM algorithm as discussed in (4.5) and (4.6).

4.4 Experiments and Results

Several experiments were conducted within an academic building on our campus for different users using an iPhone 7. Different users participated in the experiments, males and females, with heights in the range $1.6 - 1.9$ m. Users walked using different trajectories within the building ranging from 12 s for a 15 m walk, to 5 minutes for a 400 m walk. Users held their phones in a tilted front position. An iPhone app [148] was used to collect measurements from the accelerometer, gyroscope and magnetometer as the user was walking, save the data and upload to an online server. The algorithms were implemented offline on Matlab.

Two examples of a 150 s and a 300 s walk are presented in Fig. 4.6 and 4.7 respectively. In order to present the benefit of the proposed GraphSLAM algorithm, trajectories resulting from implementing different algorithms are shown. In Fig. 4.6(a) and 4.7(a), the true movement trajectories are shown. Trajectories resulting from implementing PDR only without and with heading calibration are presented in Fig. 4.6(b) and 4.7(b) as the red and green lines respectively. PDR without heading calibration suffers greatly from heading rotation as shown in the figures. Although heading calibration reduces the heading direction rotation, still the trajectory is far from the true one.

Fig. 4.6(c) and 4.7(c) show the resulting trajectory after applying heading calibration, heading detection pre-processing stage (including multiples of $\pi/2$ and $\pi/4$ heading directions) and finally the GraphSLAM optimization. The start and end positions were included as virtual nodes to constrain the optimization problem. As shown, the proposed algorithm greatly enhances the resulting trajectory, closely following the true one. Further improvement due to landmarks placed at different parts of the building are shown in Fig. 4.6(d) and 4.7(d), also included as virtual nodes in the optimization problem. The positions of landmarks were selected at the building entrances to simulate a more practical scenario where the details of the floorplan are not known in advance.

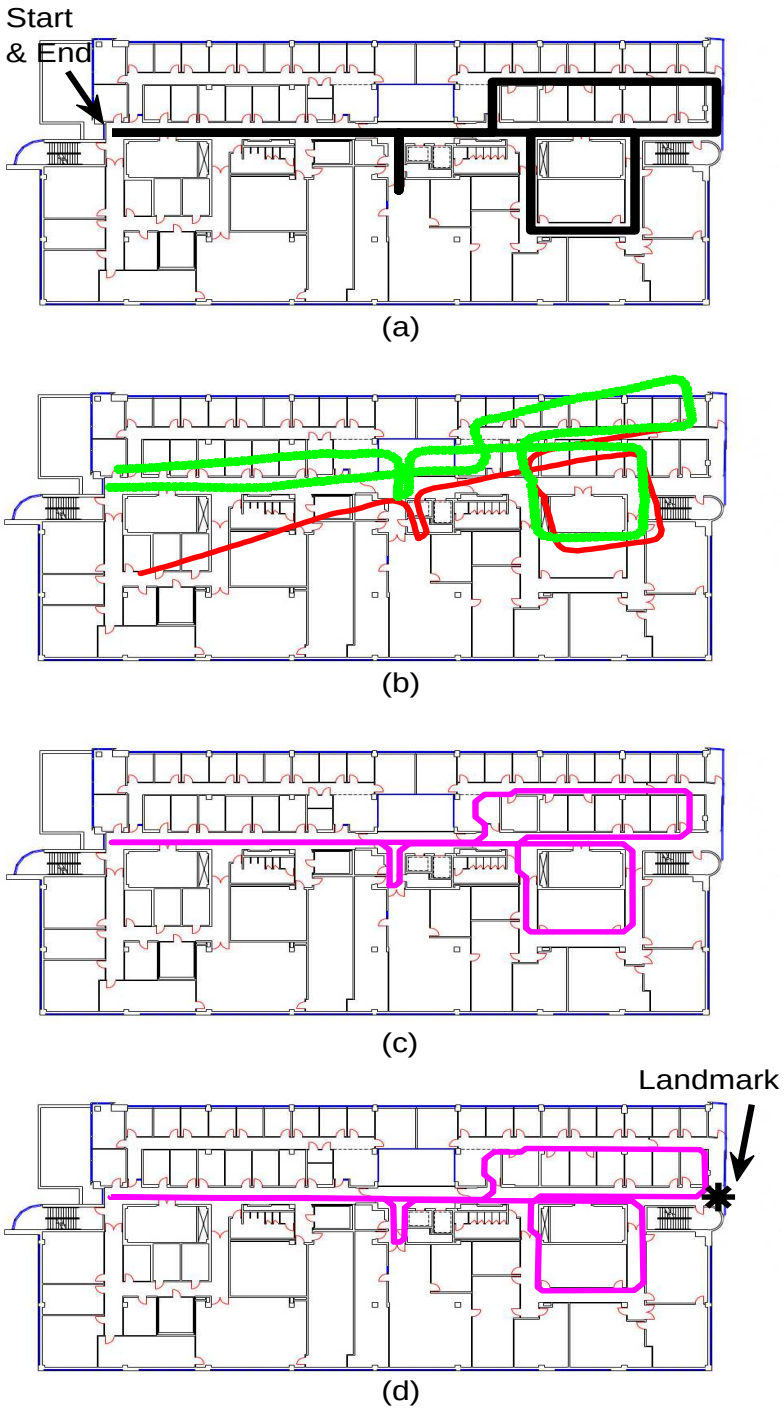

Figure 4.6: Movement trajectories for a user inside a part of an academic building. Duration: 2.5 mins. Total length $\approx$ 200 meters. (a) True trajectory, (b) PDR-based trajectories without (red) and with (green) heading calibration, (c) GraphSLAM trajectory with both heading calibration and pre-processing heading detection, and end point as a loop closure, and (d) GraphSLAM with loop closure and one additional landmark.

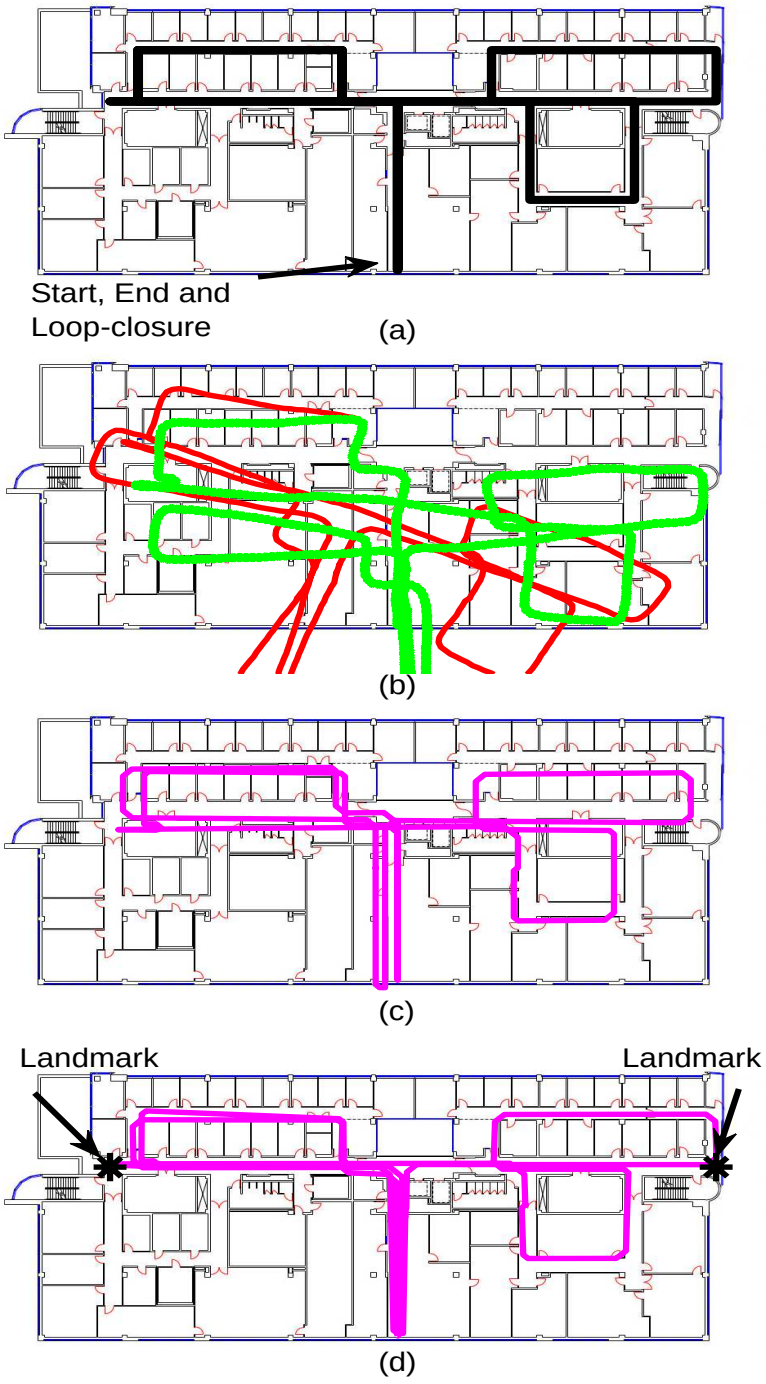

Figure 4.7: Movement trajectories for a user inside a part of an academic building. Duration: 5 mins. Total length $\approx$ 390 meters. (a) True trajectory, (b) PDR-based trajectories without (red) and with (green) heading calibration, (c) GraphSLAM trajectory with both heading calibration and pre-processing heading detection, and end point as a loop closure, and (d) GraphSLAM with loop closure and additional landmarks.

4.4.1 GraphSLAM Performance Evaluation

In order to provide a quantitative performance evaluation, ground truths for experiment trajectories were generated manually based on previous knowledge of the floor plan, as presented in Fig. 4.6(a) and 4.7(a), with an estimated average error of < 0.5 m between the true trajectory and the generated one. Two common trajectory similarity measures were implemented to evaluate the performance of the different algorithms [62]:

- Dynamic Time Warping (DTW): DTW calculates the cumulative sum of a distance measure between corresponding positions in two trajectories over a warping window [149]. The distance measure was chosen as the Euclidean distance.
- Longest Common Sub-Sequence (LCSS): LCSS measures the similarity between two trajectories by matching the common sequences in both the trajectories. The algorithm considers points in different trajectories a match if the distance between the two points is less than a threshold τ within a window [150].

Table 4.2 presents both DTW and LCSS measures for 4 different trajectories within the same building, resulting from implementing the four different algorithms: PDR, PDR with heading measurement calibration, GraphSLAM with heading direction calibration and heading detection, and GraphSLAM with the addition of landmarks. The cumulative sum resulting from DTW is averaged over the number of estimated position points in each trajectory, resulting in an average error per position in meters. LCSS is calculated for $\tau = \{1, 2\}$m, and presented as the percentage of estimated positions that lie within τ meters of the corresponding true positions.

From Table 4.2, it's shown that: (1) performance is variable depending on the trajectory length and the positions of the landmarks. However, implementing GraphSLAM with heading calibration and detection always improves the estimated trajectories compared to both cases of PDR. The average error per position improves by $66\% - 85\%$, while the percentage of estimated positions within 2 meters of the true positions increases by 23.4% in shorter trajectories to 63.2% in longer ones. (2) The addition of landmarks into the optimization problem usually results in improved performance, but it's situation dependent. For Traj. 2, adding a landmark results in overall smaller average error per position, although it results in fewer estimated positions within 1 m of the true values; i.e. the landmark provided better constraints for farther points but simultaneously shifted away some of the closer estimated positions. This confirms that implementing GraphSLAM with only known starting and ending positions is capable of greatly enhancing the accuracy of estimated trajectories within unknown buildings.

Table 4.2: Trajectory Similarity Measures for Performance Evaluation of GraphSLAM Proposed Algorithm

		PDR	PDR with heading calibration	GraphSLAM	GraphSLAM with landmarks
Traj. 1	DTW (Average error/position)	3.22 m	1.41 m	0.42 m	0.29 m
117m	LCSS (Estimated Positions% within 2 m)	44%	58.1%	92.9%	92.9%
107s	LCSS (Estimated Positions% within 1 m)	30.4%	43.5%	91.3%	91.8%
Traj. 2	DTW (Average error/position)	1.92 m	1.8 m	0.6 m	0.55 m
180m	LCSS (Estimated Positions% within 2 m)	53.6%	66.6%	90%	90%
169s	LCSS (Estimated Positions% within 1 m)	36.4%	22.3%	88.6%	82.8%
Traj. 3 (Fig. 4.6)	DTW (Average error/position)	4.47 m	4.64 m	0.71 m	0.59 m
200m	LCSS (Estimated Positions% within 2 m)	36.8%	35.7%	98.9%	100%
151s	LCSS (Estimated Positions% within 1 m)	19.9%	25.3%	85.2%	97.5%
Traj. 4 (Fig. 4.7)	DTW (Average error/position)	11.6 m	8.99 m	1.76 m	0.55 m
392m	LCSS (Estimated Positions% within 2 m)	24.6%	24.4%	71.7%	99.3%
301s	LCSS (Estimated Positions% within 1 m)	15%	17.4%	24.1%	97.2%

4.4.2 Using Trajectories for Improving Localization

Movement trajectories resulting from applying the proposed GraphSLAM approach create topological maps of unknown building floor plans. Topological maps use graphs to present possibilities of navigation through the environment [151]. The created maps can be used to improve indoor localization for the user, and any other future users moving inside the building, through trajectory matching approaches. Smartphones detecting and storing their movement trajectories for specific time durations can use trajectory matching techniques to estimate their most recent positions with much higher accuracy. Trajectory matching can be performed through iterative snapping approaches presented in [152–154] or through trajectory similarity measures which have been discussed in our previous work [62]. Fig. 4.8 shows two examples of trajectory matching for different users for trajectories of duration $\sim$ 30 s through the implementation of DTW similarity measure as discussed in [62]. Red lines are 30 sec trajectories detected through PDR with heading calibration. Black lines are the matched segments. By successfully matching trajectories to the topological map, the average localization error will depend on the performance of the GraphSLAM algorithm.

4.5 Conclusions

In this work, we presented a new algorithm for pedestrian GraphSLAM using smartphones-based PDR measurements and evaluated the algorithm using experimental data captured on an iPhone 7. Smartphones-

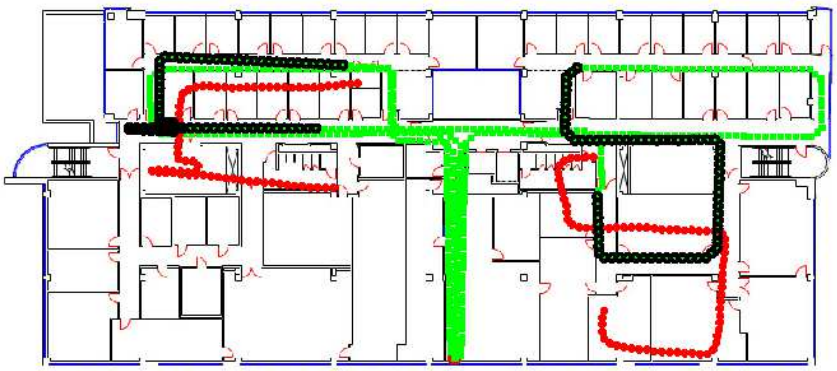

Figure 4.8: Two examples of matching movement trajectories to building topological maps.

based PDR measurements suffer greatly from the linear error drift in heading direction estimation. To overcome this type of error, we proposed a new approach, exploiting the fact that most buildings have rectangular architectures. We proposed a heading calibration algorithm to reset the drift error, in addition to a pre-processing stage for GraphSLAM that includes maximum likelihood detection for the heading direction. Implementing the new approach resulted in movement trajectories that overcame the trajectory rotation problem, and closely matched the true ones (typically within 1m of accuracy). The trajectories resulting from the proposed GraphSLAM algorithm create topological building maps that can be used for indoor localization through trajectory matching or snapping techniques.

Chapter 5

Multiple-User Collaboration for Pedestrian Indoor GraphSLAM using BLE

Pedestrian GraphSLAM is a new approach to the indoor localization and mapping problem. With the advancements in smartphone technology, pedestrian SLAM has transitioned towards utilizing smartphones' integrated sensors through Pedestrian Dead-Reckoning (PDR) techniques. However, the performance of SLAM systems greatly depends on the ability to detect loop closures and/or landmarks in the environment that are used to constrain the trajectory estimation problem. While in robotic SLAM systems, efficient loop closures and landmark detection techniques have been established through data association techniques (especially with the use of visual sensors and range scanners), the problem in pedestrian SLAM is very challenging. In this chapter, we present a novel approach for a collaborative pedestrian GraphSLAM system using Bluetooth Low Energy (BLE) detection between smartphones as a means to add constraints to the SLAM problem without the need for loop closures or landmark detection. The main contributions of this chapter are (1) analysis of the BLE Received Signal Strength (RSS) between smartphones with variable distances to determine its viability as a proximity detector, and (2) development of a new collaborative GraphSLAM framework as a jointly constrained optimization problem based on the BLE RSS between smartphones. Experiments were conducted using iPhone 7 devices within an academic building. The proposed collaborative technique was able to provide alternative constraints to the GraphSLAM optimization problem and provide accurate estimates for indoor users' locations and movement trajectories.

5.1 Introduction

The Simultaneous Localization and Mapping (SLAM) problem is defined as the construction of a consistent map of a user's previously unknown surrounding environment while simultaneously estimating the user's current position within the map. It has been a heavily researched field in the robotics community for more than 3 decades [46]. The problem involves the operation of a robot with multiple sensors (specially cameras and laser scanners) within an unknown environment, where the robot performs multiple functions: position estimation, identifying environment landmarks, data association, map creation, and position and map updates. The popularity of robotics SLAM is due to its various applications in exploring environments where it would not be possible or safe for humans to be, for example: planetary rovers, robots developed for mapping/detecting landmines and for military purposes. For more information about robotics SLAM, readers are referred to [46, 47, 113].

The keywords for GraphSLAM are *loop closure* and *landmark*. Loop closure is the ability of the user to identify previously visited places, while landmarks are fixed-positions within the environment [49]. Loop closures and landmarks provide constraints to the GraphSLAM optimization problem. More loop closures/landmarks in the environment result in more constraints to the measurement error minimization problem which leads to improved trajectory tracking performance.

However, detecting loop closures and landmarks using smartphone sensors is not trivial. While for robots, images and range measurements processing techniques are used for data association and loop closure/landmark detection, this is not the case for smartphone users. Autonomous landmarks for smartphone users are mainly based on fixed wireless transmission points, for example WiFi APs [50], Bluetooth beacons [51] and RFID detectors [52], assuming pre-existing installations in the building. Other techniques for smartphone landmark detection suggest the detection and classification of user activities as landmarks, such as movement through stairs and elevators, which requires additional complex intelligent algorithms [53].

In this chapter we present a novel technique for indoor pedestrian GraphSLAM based on Bluetooth Low Energy (BLE) technology as an alternative to the need for loop closures and landmark detection. BLE Received Signal Strength (RSS) measurements are used within a novel approach to provide distance constraints between collaborating smartphones within close proximity. A BLE *advertise* and *scan* application for iPhone devices was developed to collect BLE RSS measurements from participating smartphones. Mapping the maximum received RSS to an upper-bound on the distance between two smartphones allows the formulation of the GraphSLAM problem as a constrained optimization problem whose solution provides high accuracy estimates for users' movement trajectories without loop closures and landmark detection.

5.2 Bluetooth Low Energy (BLE)

Bluetooth is a wireless standard for short range, low power and low-cost communication systems [155]. Since releasing its first version in 1999, Bluetooth 1.0, Bluetooth has become an essential embedded component in billions of devices around the world, starting with mobile phones, and extending to thousands of other Bluetooth enabled devices and appliances. In 2014, the Bluetooth Special Interest Group (SIG) announced that Bluetooth achieves 90% penetration in all mobile phones [156]. Most smartphones in the market today are enabled with Bluetooth 4 (more recently Bluetooth 4.2).

Bluetooth Low Energy (BLE) is a new mode of operation introduced into the Bluetooth standard in version 4, in addition to the Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR) mode [157]. It is designed as a complementary technology to classic Bluetooth, optimized for short burst data transmission with very low power operation. The short burst-low power transmission mode introduced a new application for BLE-enabled devices, which is *broadcast*. BLE-enabled devices can *advertise* their presence: devices can periodically transmit their identity and a small amount of information. In addition, they can *scan* for other devices that are broadcasting within their vicinity. Utilizing the *advertise* and *scan* modes, BLE-enabled smartphones can detect the presence of other BLE-enabled devices within their vicinity, which is often referred to as proximity detection.

BLE operates in the 2.4 GHz band, using 40 channels with 2 MHz spacing, spanning the same bands used by WiFi. Out of the 40 channels, 3 are used for advertising, carefully positioned to minimize interference with common WiFi deployments [51]. The channels are labeled 37, 38 and 39, and operate at frequencies 2402 MHz, 2426 MHz and 2480 MHz, respectively.

5.2.1 Bluetooth for Indoor Localization

With Bluetooth modules embedded in almost every mobile phone in the market, there have been several proposed Bluetooth-based indoor positioning systems, such as [51, 158–163]. The underlying approaches for these proposals included: (1) range estimation and triangulation techniques, and (2) fingerprinting techniques. Range estimation techniques estimate the distance between the Bluetooth-enabled mobile device and multiple Base-Stations (BSs) within the environment and then combine the range measurements using triangulation [158, 162]. Most of the range estimation techniques depend on the relation between the Received Signal Strength (RSS) [1] and the associated distance for distance estimation, although Bluetooth pa-

[1] RSSI was an indicator that denotes whether the received power level is within a certain range in earlier Bluetooth releases and required an active connection to acquire. Starting in Bluetooth 2.1, RSSI was redefined such that it was not related to a certain power level anymore and that the RSS could be reported in dBm without the need for an established connection [164].

rameters other than RSS have been suggested also for range estimation, such as presented in [161]). On the other hand, fingerprinting techniques collect RSS measurements from multiple BSs in predefined positions within the environment in an off-line surveying stage, and then use the resulting RF map to find the closest match to the current position during the operating stage [159]. In order to estimate the distances between Bluetooth-enabled devices and BSs, several systems developed appropriate models for indoor Bluetooth signal propagation through measurements [158, 160, 162, 163].

Some challenges that the earlier proposed systems for Bluetooth-based indoor localization faced were resolved as newer versions of Bluetooth were released, for example: the requirement for an active Bluetooth connection in order to acquire the parameters required for range measurements was not necessary anymore starting in Bluetooth 2.1. However, the two main challenges for indoor positioning based on Bluetooth, which are still widely investigated, are: (1) the wide fluctuations of signal propagation in indoor areas mainly due to multipath interference which makes developing appropriate signal propagation models very complex, and at the same time leads to (2) high variations in RSS values measured at a given distance.

In addition, for new BLE deployments, BLE advertisements are broadcast on the three advertisement channels within a very fast cycle. The BLE-enabled receiver device scans for advertisements within a cycle of the three channels. For most of the receiver devices, received broadcast messages do not include information about which channel the transmitter used for broadcasting. This leads to even more variations in the RSS values received at a given distance from the same device, as the RSS values detected at each channel would be different due to their different frequency responses. The work by Faragher and Harle in [51] presented a comprehensive study of received BLE RSS using an iPhone 7 for both static and moving devices. For static devices, they confirmed variations in the RSS received from a BLE beacon of up $\pm$5dBm on a single advertisement channel at a constant distance of 3m, while the RSS mean values on the 3 channels had $\pm$2dBm difference. While for dynamic scenarios, multipath fading was evident where 30dB drops in power were recorded for movements just across 10cm.

In this work, we present a novel approach to include BLE RSS measurements within a collaborative GraphSLAM framework. Instead of using RSS values to estimate an absolute distance, the new collaborative algorithm proposes using received RSS parameter values between two BLE-enabled smartphones as a constraint for the GraphSLAM optimization problem. The following subsection presents BLE RSS measurements between two iPhone 7 devices, collected by a BLE scanning application developed specifically for this study. In the next section, we present the joint GraphSLAM optimization problem.

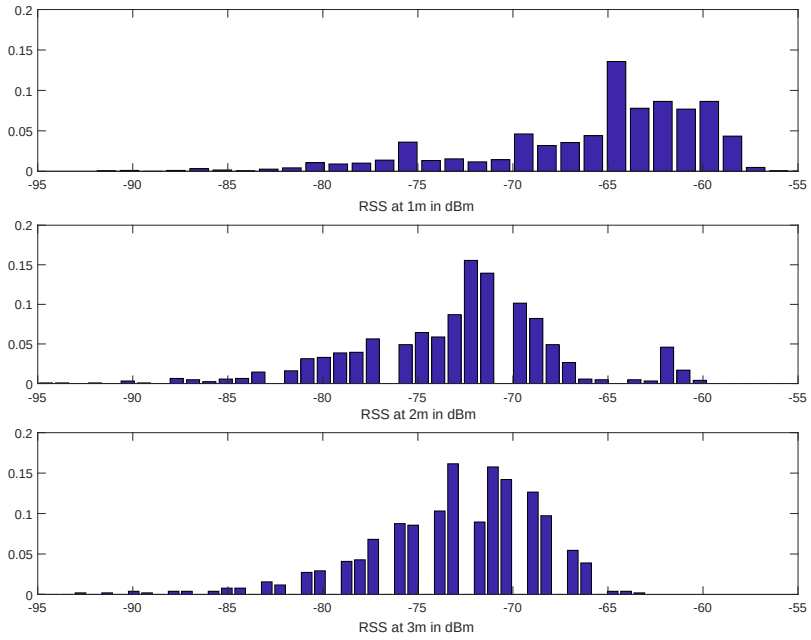

Figure 5.1: Histogram of RSS measurements received at two static BLE-enabled iPhone 7 devices at distances of 1-3m.

5.2.2 BLE RSS measurements

In order to model the RSS parameter values received at BLE-enabled smartphones, an application for BLE advertising and scanning was developed for iOS systems using SWIFT 4 in the Xcode Apple development environment. The application sets the iOS device to operate in both modes simultaneously: advertise and scan. The device broadcasts an advertisement message with a known ID so that another iOS device using the same application can detect its presence. One method for setting a unique ID for the device can be through setting a unique 128-bit universally unique identifier for the device called *UUID*. At the same time, the application scans for advertisement messages from BLE-enabled devices within its vicinity. When a message is detected, the application extracts from the advertisement message the unique *UUID* and *namekey*, along with RSS value and saves this data with the corresponding timestamp. More details about iOS development for BLE can be found on the Apple developers website [165].

The BLE advertising and scanning application was used to run a number of experiments to model the RSS measurements between two iPhone 7 devices. First, the RSS measurements between two static devices at pre-defined distances were collected. At every LOS distance of 1-3 m, RSS values were recorded for a 2-minute duration at different times of day inside a building corridor. Histograms of the recorded RSS values, combined from both phones, are shown in Fig. 5.1. RSS varies greatly at the same distance due to multipath fading and different frequency responses for the 3 advertisement channels. In addition, another set

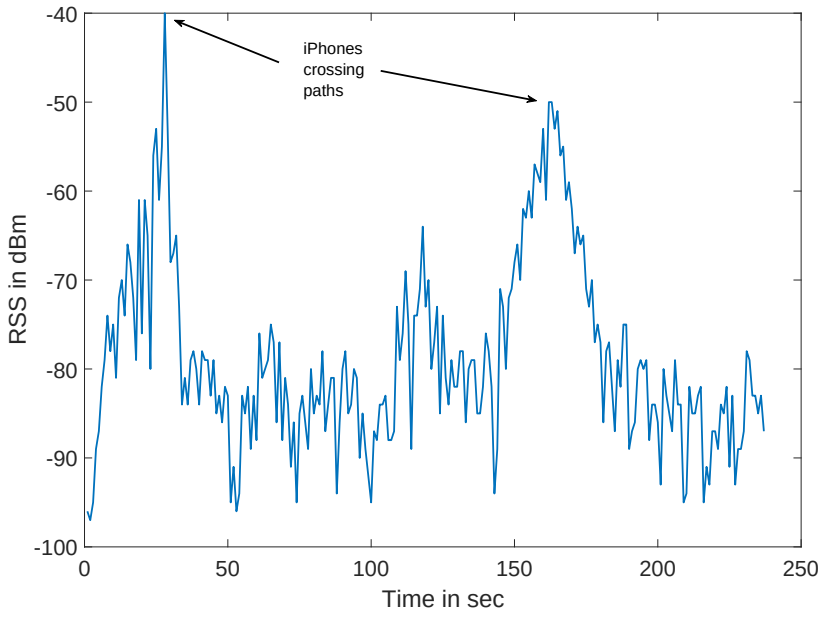

Figure 5.2: RSS in dBm received at a BLE-enabled iPhone 7 from a similar device walking towards each other and crossing paths twice within a ~ 4 minutes walk.

of experiments was conducted where the RSS measurements received at the 2 smartphones were collected and recorded as the two users holding the phones were moving towards each other in opposing directions. Fig. 5.2 shows an example of RSS received at one smartphone as a result of detecting the advertisement signal from the other phone while the two users are walking inside the building and crossing paths two times. RSS is maximum when the devices are at their closest proximity. Both Fig. 5.1 and 5.2 demonstrate how RSS varies greatly for a given distance.

Based on the measurements from the series of experiments, we propose a new approach for extracting distance information between BLE-enabled devices: instead of estimating an absolute value for the distance between the two devices, we use the measurements to extract constraint information at specific time instants. We propose a new approach for extracting distance constraints based on the following assumptions: (1) a RSS value from a device greater than a certain threshold τ_{RSS} can be used to assume that the distance between the two devices is less than a certain value d_{max}, and (2) detecting a peak in a sequence of RSS values at a time instant t_k, where the peak is greater than τ_{RSS}, implies that the distance between the two devices is minimal at that time instant. Based on our measurements, we adopted the following algorithm: detecting a peak in the sequence of RSS values where the maximum RSS value is greater than -55dB at t_k implies that the distance between the two devices is less than 1m at t_k. The assumption was validated through several experiments. Utilizing the distance constraints in the GraphSLAM problem provides an alternative to the constraints obtained through loop closures and landmarks within the environment.

5.3 Collaborative GraphSLAM using BLE

GraphSLAM is a probabilistic framework for the SLAM problem in which the problem is formulated in a pose-graph representation. Details about formulating the GraphSLAM problem for a single smartphone user were presented in Section 4.3. In this chapter, we extend the GraphSLAM problem into a joint optimization problem with distance constraints obtained from the BLE measurements between the cooperating users.

5.3.1 System Model

The proposed collaborative GraphSLAM algorithm assumes the presence of more than one user within the same building, as well as a Fusion Center (FC) where the measurements from the cooperating smartphones are combined. The FC is a general term referring to a centralized entity that collects measurements from the smartphones and performs the joint optimization problem. For example, the FC could be a local server within the building, or a remote server collecting data through a smartphone application. Each smartphone runs two simultaneous sensing processes for collecting measurements from: (1) PDR sensors (accelerometer, gyroscope, and magnetometer) for position estimation and (2) BLE sensor (advertising and scanning) for relative distance estimation to other users through RSS measurements. In the following analysis, we present the case for two users. The proposed system can be directly extended to more than two users.

Each smartphone collects a sequence of measurements from its sensors over a time duration where the user holding the phone is walking inside the building. Each smartphone performs the following operations:

- Based on the accelerometer measurements, the smartphone performs step detection and step length estimation as explained in detail in Subsection 4.2.1.
- Based on the measurements from the gyroscope and magnetometer, the smartphone performs heading angle estimation and calibration as discussed in detail in Subsection 4.2.2.
- Based on the estimates of the step length and the heading angle, each smartphone can generate a user's position vector using the PDR principle: $\mathbf{p}_1 = \{\mathbf{p}_{1,1}, \mathbf{p}_{1,2}, ..., \mathbf{p}_{1,K}\}$ for user 1 and $\mathbf{p}_2 = \{\mathbf{p}_{2,1}, \mathbf{p}_{2,2}, ..., \mathbf{p}_{2,N}\}$ for user 2 as in (4.1). Position vectors are not of the same length by default, $K \neq N$.
- Simultaneously, each smartphone uses the BLE advertising and scanning application to advertise its own information as well as scan for other collaborating smartphones within its vicinity based on their unique IDs. The BLE application generates a vector of RSS measurements with corresponding timestamps.

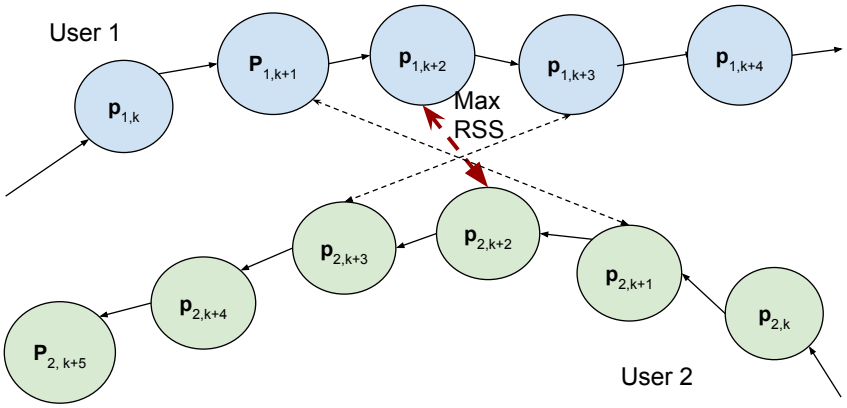

Figure 5.3: Pose-Graph representation of the collaborative GraphSLAM problem using BLE measurements.

- Each smartphone shares its position vector and its BLE RSS vector with the FC. The FC performs a joint GraphSLAM optimization based on measurements from the individual smartphones, as explained in the following subsection.

5.3.2 Joint GraphSLAM Optimization

Combining both estimated position and BLE RSS measurements from both smartphones results in a joint GraphSLAM constrained optimization problem. The FC formulates the joint optimization problem based on the position vectors $\mathbf{p}_1$ and $\mathbf{p}_2$ and the BLE RSS vector.

For each position vector $\mathbf{p}_1$ and $\mathbf{p}_2$, displacement measurement vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ are generated as in 4.2, as well as a concatenated measurement vector $\mathbf{d} = [\mathbf{d}_1^T \mathbf{d}_2^T]^T$. Measurement errors representing the difference between the true and the measured displacements for each user u, defined in (4.3) which is repeated here, are determined as:

$$\mathbf{e}_{u,ij}(\mathbf{p}_{u,i}, \mathbf{p}_{u,j}) = \hat{\mathbf{d}}_{u,ij} - \mathbf{d}_{u,ij} = (\hat{\mathbf{p}}_{u,i} - \hat{\mathbf{p}}_{u,j}) - \mathbf{d}_{u,ij}, \tag{5.1}$$

where $\hat{\mathbf{p}}_{u,i}$ is the true position of user u at time instant t_i. Displacement measurement errors represent the spatial constraints between successive positions for each user. They are modeled as presented in Table 4.1 based on the analysis presented in Subsection 4.3.2. Equation (5.1) can be represented in matrix form as:

$$\mathbf{e} = \mathbf{H}\mathbf{p} - \mathbf{d}, \tag{5.2}$$

where $\mathbf{H}$ is the incidence matrix with all entries being 1, -1 or 0 based on the nodes' connections and $\mathbf{p}$ is the concatenation of the true positions for both users.

From the BLE RSS vectors, the FC extracts the time-stamps $TS = \{TS_1, TS_2, ..., TS_R\}$ where peak RSS values are greater than τ_{RSS}. These time-stamps are matched with time-stamps in both position vectors $\mathbf{p}_1$ and $\mathbf{p}_2$ to detect the set of time instants $\mathbf{t}_1 = \{t_{1,TS_1}, t_{1,TS_2}, ..., t_{1,TS_R}\}$ and $\mathbf{t}_2 = \{t_{2,TS_1}, t_{2,TS_2}, ..., t_{2,TS_R}\}$ in both position vectors that correspond to positions at which the smartphones were within d_{max} distance from each other. In the example shown in Fig. 5.2, $R = 2$, where R is the number of detected RSS peaks greater than τ_{RSS}. Fig. 5.3 shows the joint GraphSLAM optimization problem, where $\mathbf{p}_{u,i}$ is the position of user u at time instant t_i, arrows present the transitions in each user's position, dashed arrows connect positions where the received RSS is greater than τ_{RSS}, and the red dashed arrow connects the two positions where the peak detected RSS is greater than τ_{RSS} which implies minimum distance between the two positions.

The objective of the joint GraphSLAM problem is to find the optimum sequence of positions $\mathbf{p}^*$ that will minimize the squared displacement error measurements presented in (5.2) under the constraint that positions corresponding to time instants $\mathbf{t}_1$ and $\mathbf{t}_2$ are within d_{max} distance of corresponding points. The joint constrained optimization problem can be formulated as:

$$\begin{aligned} \mathbf{p}^* &= \underset{\mathbf{p}}{argmin}\ F(\mathbf{P}) = \underset{\mathbf{p}}{argmin}\ [\mathbf{Hp} - \mathbf{d}]^T \mathbf{\Sigma}^{-1} [\mathbf{Hp} - \mathbf{d}] \\ s.t. \qquad & |\mathbf{p}_{1,t_{1,i}} - \mathbf{p}_{2,t_{2,i}}| \leq d_{max}, \qquad i = TS_1, TS_2, ..., TS_R, \end{aligned} \tag{5.3}$$

where $\mathbf{\Sigma}$ is the co-variance matrix of the displacement measurement error. The optimization constraints can also be represented in matrix form as:

$$\begin{aligned} \mathbf{p}^* &= \underset{\mathbf{p}}{argmin}\ F(\mathbf{p}) = \underset{\mathbf{p}}{argmin}\ [\mathbf{Hp} - \mathbf{d}]^T \mathbf{\Sigma}^{-1} [\mathbf{Hp} - \mathbf{d}] \\ & s.t. \qquad |\mathbf{A}\ \mathbf{p}_{TS}| \leq \mathbf{d}_{max} \end{aligned} \tag{5.4}$$

where $\mathbf{A}$ in an incidence matrix, $\mathbf{p}_{TS}$ is the vector of positions from both users corresponding to the peak BLE RSS values, and $\mathbf{d}_{max}$ is the vector of the corresponding distance constraints. The absolute distance constraints can be expressed in linear form as:

$$\begin{aligned} \mathbf{p}^* &= \underset{\mathbf{p}}{argmin}\ F(\mathbf{p}) = \underset{\mathbf{p}}{argmin}\ [\mathbf{Hp} - \mathbf{d}]^T \mathbf{\Sigma}^{-1} [\mathbf{Hp} - \mathbf{d}] \\ & s.t. \qquad \mathbf{A}\ \mathbf{p}_{TS} \leq \mathbf{d}_{max}, \\ & \qquad\quad -\mathbf{A}\ \mathbf{p}_{TS} \leq \mathbf{d}_{max} \end{aligned} \tag{5.5}$$

The problem in 5.5 is a constrained linear Least Squares (LS) optimization problem which is a well-defined optimization problem and can be solved efficiently through various techniques [143, 166].

Finally, we propose extending the distance constraints to include a couple of position points around the position points corresponding to the peak BLE RSS value. This results in an overall improvement of the

solution to the optimization problem as it better models the distance constraint variations over time. If the two points $p_{1,k}$ and $p_{2,k}$ are the positions of users 1 and 2 respectively at time instant t_k corresponding to the peak BLE RSS value such that $|p_{1,k} - p_{2,k}| \leq d_{max}$, it follows that $|p_{1,k+1} - p_{2,k+1}|$ is also constrained but within a longer distance which would be equivalent to the distances traveled by both users within a single step.

5.4 Experiments and Results

Several experiments were conducted within an academic building on our campus for different users using two iPhone 7 devices. Different users participated in the experiments, males and females, with heights in the range $1.6 - 1.9$ m. Users walked using different trajectories within the building such that trajectories intersected at different points. Users held their phones in a tilted front position. Both iPhone applications were used, the one in [148] was used to collect measurements from the accelerometer, gyroscope, and magnetometer and our own developed BLE application was used to advertise each phone's information and simultaneously scan for the other phone. Each application saved the collected data with corresponding time-stamps and uploaded to an online server. The algorithms were implemented offline using Matlab.

Two examples of ~ 210s and ~ 295s walks from two users simultaneously are presented in Fig. 5.4 and 5.5 respectively. In Fig. 5.4(a) and 5.5(a), the overlapping true movement trajectories are shown. Trajectories resulting from implementing PDR only for each phone without heading calibration are presented in Fig. 5.4(b) and 5.5(b) respectively. In both figures, the red trajectory presents user 1 movement, the blue trajectory presents user 2 movement, and the highlighted segments present the positions corresponding to peak BLE RSS values between the two devices. PDR without heading calibration suffers greatly from heading rotation as shown in the figures. Trajectories resulting from implementing PDR for each user phone with heading calibration are presented in Fig. 5.4(c) and 5.5(c) respectively. Although heading calibration reduces the heading direction rotation, the trajectories are still far from the true ones.

Fig. 5.4(d) and 5.5(d) show the resulting trajectories after applying heading calibration, a heading detection pre-processing stage and finally the joint GraphSLAM optimization including BLE RSS constraints. Only the start positions were included in the problem, corresponding to fixed entry points such as doors, elevators, or stairwells. As shown, the proposed joint GraphSLAM algorithm was able to efficiently estimate movement trajectories that are very close to the true ones. The estimated distances provided by BLE RSS detection were able to constrain the optimization problem, overcoming the need for loop closures and/or landmarks. Accordingly, the proposed algorithm is suitable for trajectory movement tracking and mapping for unknown buildings without the need to detect landmarks or loop closures through multi-user collaboration.

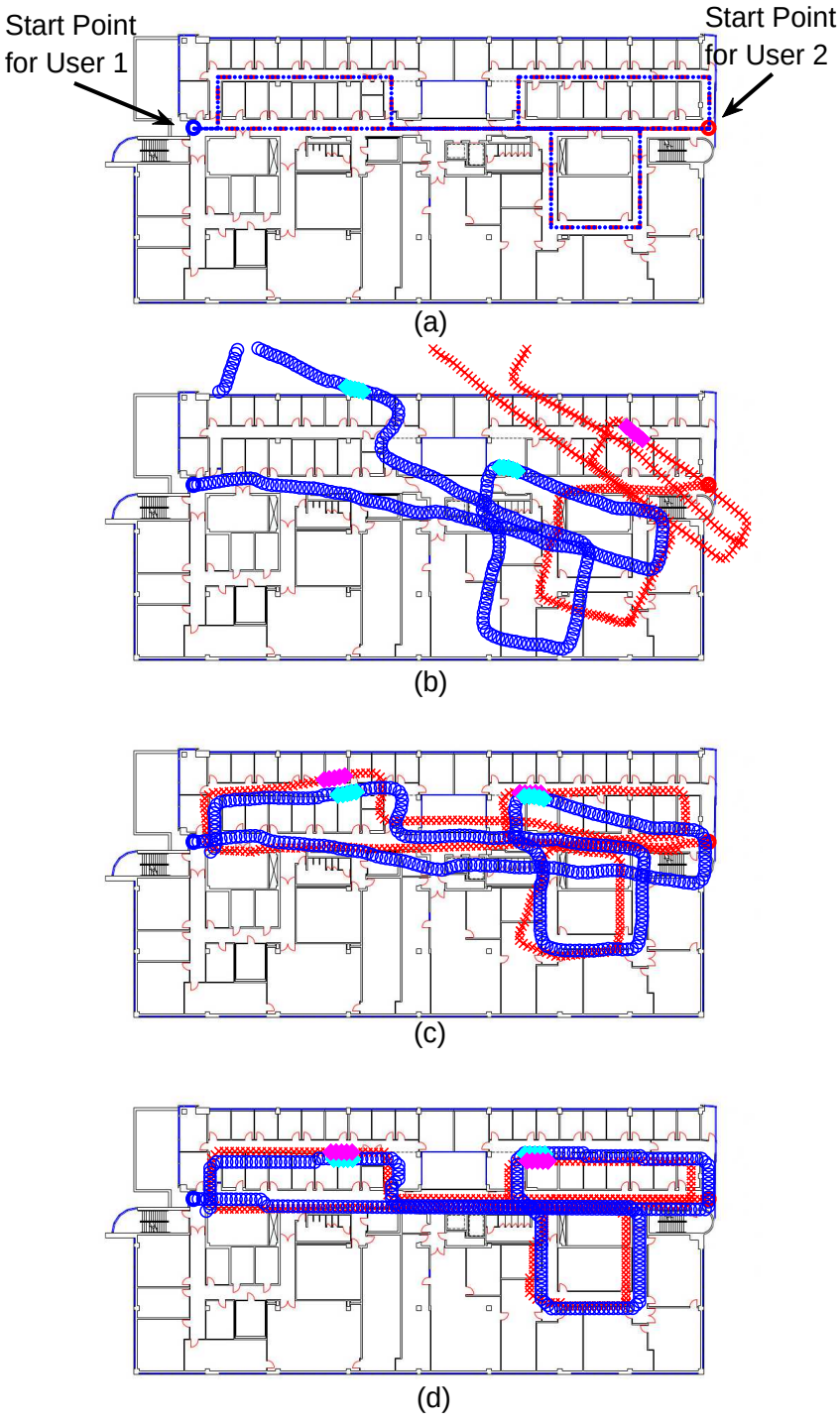

Figure 5.4: Collaborative GraphSLAM: Movement trajectories for two users inside a part of an academic building. Duration $\sim$ 210s. (a) True trajectories, (b) PDR-based trajectories without heading calibration, (c) PDR-based trajectories with heading calibration, (d) Joint GraphSLAM optimization with BLE RSS constraints. Red is user 1, blue is user 2, highlighted segments are positions corresponding to peak BLE RSS values.

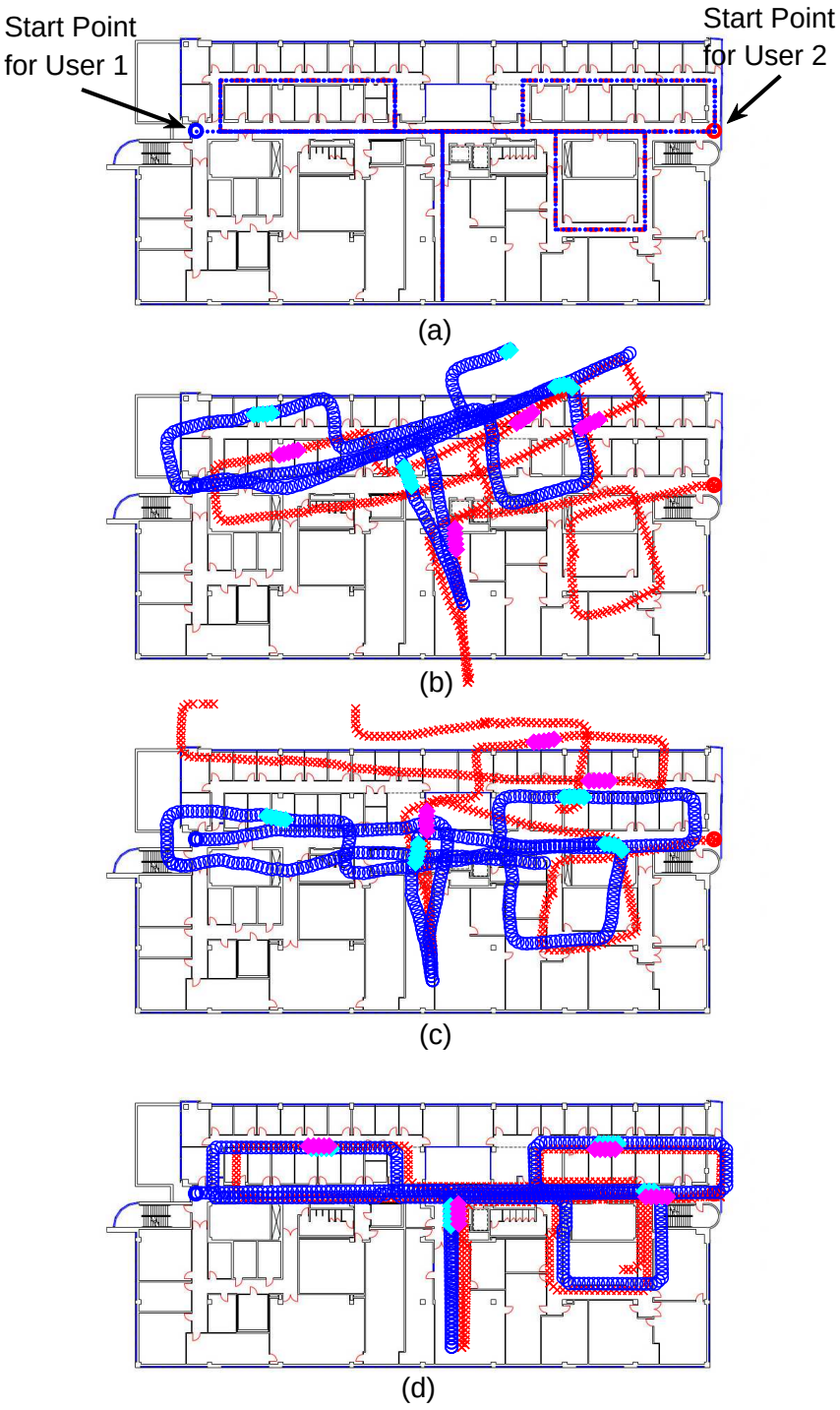

Figure 5.5: Collaborative GraphSLAM: Movement trajectories for a user inside a part of an academic building. Duration ~ 295s. (a) True trajectories, (b) PDR-based trajectories without heading calibration, (c) PDR-based trajectories with heading calibration, (d) Joint GraphSLAM optimization with BLE RSS constraints. Red is user 1, blue is user 2, highlighted segments are positions corresponding to peak BLE RSS values.

Table 5.1: Trajectory Similarity Measures for Performance Evaluation of Joint GraphSLAM Proposed Algorithm with BLE RSS Distance Constraints

		PDR	PDR with heading calibration	Joint GraphSLAM with BLE RSS constraints
User. 1 (Fig. 5.4)	DTW (Average error/position)	5.74 m	1.88 m	1.17 m
221m	LCSS (Estimated Positions% within 2 m)	28.8%	66.6.1%	89%
207s	LCSS (Estimated Positions% within 1 m)	17.2%	44.2%	30%
User. 2 (Fig 5.4)	DTW (Average error/position)	9.68 m	1.46 m	1.14 m
256 m	LCSS (Estimated Positions% within 2 m)	29.6%	75.4%	90.8%
220s	LCSS (Estimated Positions% within 1 m)	19.6%	51.8%	50.5%
User. 1 (Fig. 5.5)	DTW (Average error/position)	16.7 m	2.78 m	0.83 m
308m	LCSS (Estimated Positions% within 2 m)	19.7%	45%	83.8%
301s	LCSS (Estimated Positions% within 1 m)	11.6%	19.77%	60.6%
User. 2 (Fig. 5.5)	DTW (Average error/position)	4.77 m	7.96 m	1.17 m
328m	LCSS (Estimated Positions% within 2 m)	39.3%	24.4%	76.6%
291s	LCSS (Estimated Positions% within 1 m)	22.4%	9%	61.5%

5.4.1 Collaborative GraphSLAM Performance Evaluation

In order to provide a quantitative performance evaluation, we propose the use of trajectory similarity measures as previously presented in Subsection 4.4.1. Ground truths for experiment trajectories were generated manually based on previous knowledge of the floor plan, as presented in Fig. 5.4(a) and 5.5(a), with an estimated average error of < 0.5 m between the true trajectory and the generated one. Two common trajectory similarity measures were implemented to evaluate the resulting trajectories through collaborative GraphSLAM: Dynamic Time Warping (DTW) and Longest Common Sub-Sequence (LCSS). DTW calculates the cumulative sum of a distance measure between corresponding positions in two trajectories over a warping window [149]. The distance measure was chosen in this work as the Euclidean distance. On the other hand, LCSS measures the similarity between two trajectories by matching the common sequences in both the trajectories. LCSS considers points in different trajectories a match if the distance between the two points is less than a threshold τ within a window [150].

Table 5.1 presents both DTW and LCSS measures for the same 4 trajectories presented in Fig. 5.4 and 5.5 resulting from implementing the algorithms: PDR, PDR with heading direction calibration, and joint GraphSLAM with heading direction calibration, heading detection, and BLE RSS distance constraints. The cumulative sum resulting from DTW is averaged over the number of estimated position points in each trajectory, resulting in an average error per position in meters. LCSS is calculated for $\tau = \{1, 2\}$ m, and presented as the percentage of estimated positions that lie within τ meters of the corresponding true positions.

From Table 5.1, we come to the following conclusions:

- Performance is variable depending on the trajectories' length, the quality of individual smartphone measurements and the frequency and position of users' intersections while moving.
- Implementing the proposed collaborative GraphSLAM always improves the estimated trajectories in terms of the average error per position compared to both cases of PDR. The average error per position improves by $75\% - 95\%$ compared to PDR measurements, with the maximum improvement achieved for the case of the PDR trajectory suffering from maximum rotation (User 1, Fig. 5.5.b). The average error per position improves by $22\% - 85\%$ compared to calibrated PDR measurements.
- While calibrated PDR usually results in trajectory estimation improvement compared to PDR without calibration, this is not always the case as shown for User. 2 in Fig. 5.5.c where calibration helps in reducing trajectory rotation at the expense of increasing the overall position error. This demonstrates the need for the collaborative GraphSLAM algorithm to provide constraints to the trajectory estimation problem.
- Implementing the proposed collaborative GraphSLAM always improves the estimated trajectories in terms of the percentage of estimated positions within 2 m of the true positions. However, investigating the values in Table 5.1 (rows 2 and 3 for each case), we can conclude that collaborative GraphSLAM optimizes the overall trajectory such that sometimes more estimated positions lie within 2m of the true trajectory at the expense of fewer estimated positions lying within 1m, such as is the case for User 1 (Fig. 5.4).

5.5 Conclusions

In this chapter, we proposed another application of sensor fusion to the indoor localization and mapping area. A new algorithm for collaborative GraphSLAM using BLE RSS measurements from smartphones is presented, where collaboration between smartphones within close proximity provides constraints to the GraphSLAM problem overcoming the need for loop closures and landmark detection. A new approach was proposed to extract distance information from BLE RSS measurements between collaborating smartphones, providing distance constraints instead of absolute distance values. Incorporating the distance constraints within the GraphSLAM framework formulates a linear constrained least squares optimization problem that can be solved efficiently to provide estimates for users' movement trajectories.

The proposed collaborative GraphSLAM algorithm was evaluated using experimental data captured on two iPhone 7 devices. PDR measurements on both devices were collected to provide initial position estimates,

while a BLE *advertise* and *scan* application was used to collect RSS measurements between the phones. Implementing the new collaborative GraphSLAM algorithm resulted in movement trajectories that closely matched the true ones (typically within 1m of accuracy). Introducing BLE RSS measurements into the GraphSLAM problem provided a novel solution to the problem of loop closures and landmark detection for smartphone users.

Chapter 6

Improving Indoor Localization through Trajectory Tracking

Tracking mobile users' movement trajectories with high accuracy indoors opens the possibilities for novel indoor localization techniques. Using GraphSLAM for indoor pedestrian trajectory tracking, or other trajectory tracking algorithms, results in creating a group of high probability movement patterns within a building, or what we can refer to as a topological map of the building. With new users tracking their own movements for some specific time duration, these new and possibly low accuracy trajectories can be used to locate, with high accuracy, where the users are inside the building through smart trajectory matching techniques. In this chapter we present two different pattern recognition techniques: Neural Networks (NN) and trajectory similarity measures, as new non-traditional indoor localization frameworks based on users' movement trajectories.

6.1 Introduction

Estimating accurate position information in indoor environments is still an ongoing challenging problem. The currently deployed positioning techniques, such as the Cell ID, Base Station Time Difference of Arrival (BS-TDOA) and Global Positioning Systems (GPS), are optimized for outdoor operation. These technologies cannot provide the accuracy necessary for indoor localization scenarios up to the room/suite level where line-of-sight (LOS) communication is not available between the receivers and the satellites and/or BSs [38]. As the need for accurate indoor positioning is increasing rapidly, especially with the new E911 requirements, novel methods for indoor localization of mobile users are required.

A new direction for indoor localization is the tracking of the smartphone users' movement trajectories inside buildings and exploiting these trajectories for more accurate localization, instead of applying localization algorithms that depend only on the most recent position. In the previous chapters, we presented novel algorithms for tracking indoor mobile user's movement trajectories using smartphones sensors. In this chapter, we propose two novel systems for enhancing positioning of indoor mobile users exploiting users' movement trajectories within two different frameworks:

- Neural Networks (NN): The NN models indoor localization at the room level as a multi-class classification problem, where the input is a movement trajectory, and the output is which room the user is currently in. For a suitable training period, users' movement trajectories inside a building are tracked and recorded with corresponding room locations, and then used as training patterns for the Neural Network classifier.
- Trajectory Similarity Measures: Based on the assumption that movement trajectories of different users within a building can be tracked anonymously with high accuracy, a user's current position can be estimated through matching his own movement trajectory, tracked with much less accuracy, with the set of anonymous trajectories to detect his current location at the room level.

In order to simulate and study the performance of both frameworks, a large number of user movement trajectories needed to be generated within a simulation layout. Thus, Section 6.2 presents the approach adopted in both frameworks for simulating thousands of movement trajectories within buildings based on Human Mobility Modeling. In Section 6.3 and Section 6.4 we present the two frameworks for exploiting movement trajectories for indoor localization: NN and trajectory similarity measures respectively.

6.2 Human Mobility Modeling

The indoor human mobility model adopted in this work is based on the Steering Forces model presented in [167]. Human mobility is modeled based on a two-level approach: large-scale and small-scale descriptions. The large-scale description includes generating the movement path the user follows as a sequence of node numbers. The small-scale description includes the generation of 2-dimensional successive user positions.

6.2.1 Large-scale Mobility Description

In a building layout, every room center and every door are considered as movement nodes. Within the large-scale description of a user's movement trajectory, an initial room (A) and a destination room (B) are

selected randomly from the set of possible rooms $\mathcal{K} = \{1, 2, ..., K\}$. A sequence of nodes representing the movement path between rooms A and B is generated. The nodes are selected from the set of possible room centers and doors based on a shortest-distance measure. In this presented simulation, Dijkstra's algorithm is used to determine the movement sequence [168].

When the user reaches the final room destination (B), the user stays in the room for a random *waiting time* before another room is randomly selected as the next destination. Dijkstra's algorithm is applied again to determine the next movement sequence. Waiting times inside each room are generated with different criteria.

6.2.2 Small-scale Mobility Description

Steering behaviors were introduced as a modelling tool for the movement of autonomous agents (i.e. humans, animals, etc) in computer animations and interactive computer games [169]. The model presented here and used in simulations for generating point-by-point human movement trajectories is a 2D modified version of the model presented in [167, 169, 170].

Once the movement sequence is defined, each user moves towards his target destination, passing through doors and avoiding walls, other users and any other obstacles in the layout. The Steering Forces model considers each user as a positive pole and the next node target is considered as a negative pole, hence generating an attraction force steering the user towards his next target. Meanwhile, other users and obstacles, e.g. walls, are considered positive poles as well, thus generating repulsion forces steering the user away from them [170]. Each user is defined with a mass m (average 70kg), maximum velocity v_{max} (average human walking velocity is assumed 1.4m/s) and maximum acceleration a_{max} (average human walking acceleration is assumed 1m/s^2) [171].

At each time step n in the simulation duration, the forces affecting user j movement are calculated as follows, and then the acceleration, velocity, and position of the user are updated:

Attraction Force F_s: This is the force between the user and the next target node T_n in the movement sequence. The attraction force depends on the distance d_n (in m) between the latest position of the user P_{n-1} and the target node T_n:

$$d_n = T_n - P_{n-1}, \qquad \hat{d}_n = \frac{d_n}{|d_n|}, \tag{6.1}$$

where $\hat{d}_n$ is the normalization of d_n and $|d_n|$ is the norm of d_n. The desired velocity v_n^s (in m/s) at time step n is updated as:

$$v_n^s = min(v_{max}, \frac{|d_n|}{\delta t})\hat{d}_n, \tag{6.2}$$

where δt is the simulation step in seconds and the desired velocity is cut-off at v_{max}. The difference between the desired velocity v_n^s and the current velocity of the user v_{n-1} is defined as:

$$w_n = v_n^s - v_{n-1}, \qquad \hat{w}_n = \frac{w_n}{|w_n|}, \tag{6.3}$$

where $\hat{w}_n$ is the normalization of w_n and gives the direction of the steering force. Finally, the Attraction Force F_s (in N) is calculated as:

$$F_s = m\frac{|v_n^s|}{\delta t}\hat{w}_n \tag{6.4}$$

Wall Avoidance Repulsion Force F_w: This is the force between the user and every wall W_k within the vicinity of the current position of the user. In this model, only repulsion forces from walls within 2 meters are considered. For each wall W_k, the repulsion force is inversely proportional to the distance between the user and the wall d_{W_k} and in a direction orthogonal to the wall. The force is calculated as:

$$F_w|_{W_k} = m\frac{\alpha|v_{n-1}|}{\delta t}\frac{1}{|d_{W_k}|^2}\hat{W}_n \tag{6.5}$$

where α is a parameter set through simulations to control the users movement in the direction of the walls, and $\hat{W}_n$ is a normalized vector orthogonal to the wall. α is set to 0.16.

User Avoidance Repulsion Force F_u: This is the force between the user and any other user within the vicinity of his current position. In this model, only users within a circle of radius 1.2 meters are considered. Similar to the Wall Avoidance Repulsion Force, the User Avoidance Repulsion Force is inversely proportional to the distance between the two users d_{U_k} and in a direction orthogonal to the other user. The force is calculated as:

$$F_u|_{U_k} = m\frac{\beta|v_{n-1}|}{\delta t}\frac{1}{|d_{U_k}|^2}\hat{U}_n \tag{6.6}$$

where β is a parameter set through simulations to control the users movement in direction of each other, and $\hat{U}_n$ is a normalized vector in the opposite direction of the distance vector. β is set to 0.5.

Total Steering Force: This is the sum of the attraction and repulsion forces, defined as:

$$F = F_s + \sum_{k=1}^{W} F_w|_{W_k} + \sum_{k=1}^{U} F_u|_{U_k}, \tag{6.7}$$

where W and U are the number of walls and users within the vicinity of user j respectively.

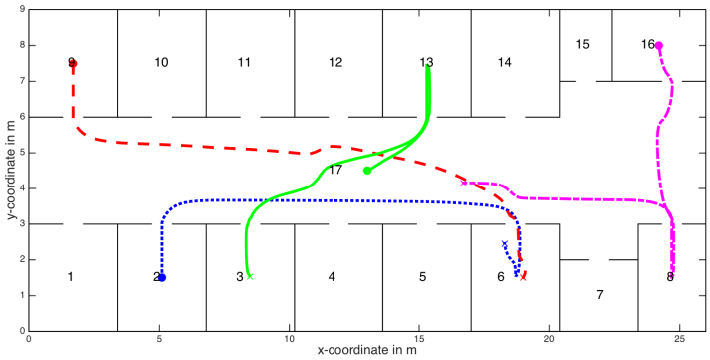

Figure 6.1: Example of four motion trajectories generated through the Human Mobility Model in a general office building layout.

Updating acceleration, velocity, and position: For each user j, the current acceleration, velocity and position are updated based on the total steering force. The updates are as follows:

$$\begin{aligned} a_n &= \frac{F}{m}, \\ v_n &= v_{n-1} + a_n \delta t, \\ p_n &= p_{n-1} + v_n \delta t. \end{aligned} \tag{6.8}$$

The acceleration and velocity will both be truncated at their maximum values as in (6.2). An example of four user movement trajectories generated using this model are shown in Fig. 6.1, where users 1-4 correspond to red, green, blue, and magenta lines. The trajectories are generated in a general office building layout with 12 equal-sized rooms (3.4m x 3m), 4 variable-sized rooms and a hallway.

6.3 Indoor Localization using Neural Networks

NNs have been proposed for indoor localization in different contexts, for example [172–175]. In [172, 173], NNs are used to map the measured Received Signal Strengths (RSSs) of a mobile user to the stored RSS fingerprints in a radio map and infer the user location accordingly, within a fingerprinting localization architecture. In [174], multiple NNs were used to incorporate the mobile user movement orientation into the RSS mapping in Bluetooth networks. In [175], features of users' past trajectories are used to predict the next locations using a Radial Basis Function (RBF) NN. Identifying users' locations up to the room level using NN's is a new approach for utilizing NNs in indoor localization. In this work, we propose using NNs, within a new framework, as a multi-class classifier for room identification in indoor scenarios.

6.3.1 System Model

The proposed framework for indoor localization at the room level consists of two components: (1) a point-based Positioning System (PS) that tracks and stores users' consecutive positions for a specified time duration generating multi-dimensional movement trajectories, and (2) a NN classifier that estimates the current room which a user is located in based on his movement trajectory.

There are several point-based PSs that could be implemented as the first component based on the users' handsets capabilities and the building infrastructure. The choice of a specific PS is beyond the scope of this work. Conventional outdoor PSs such as GPS or cellular positioning techniques could be utilized, usually providing low-accuracy position estimates. Depending on the building instrumentation, more accurate indoor PSs could also be implemented and combined with conventional GPS, such as RSS at WiFi APs, magnetic fields, mechanical energy, and atmospheric pressure [176]. More about indoor PSs could be found in [34–36].

The PS tracks consecutive mobile users' positions for a specific time duration inside a building generating movement trajectories as finite sequences of 2-D locations with time-stamps, i.e. two-dimensional time-series. The framework in this model is presented in a two-dimensional setting; identifying the room number on a single floor. Extending the system to include the vertical dimension is straightforward, although obtaining z-dimension position estimates is more challenging. At any given time t_N, each user j has a corresponding trajectory S_j defined as:

$$\begin{aligned} S_j &= [(s_{1,j}, t_{1,j}), (s_{2,j}, t_{2,j}), ..., (s_{N,j}, t_{N,j})] \\ &= [(x_{1,j}, y_{1,j}, t_{1,j}), (x_{2,j}, y_{2,j}, t_{2,j}), ..., (x_{N,j}, y_{N,j}, t_{N,j})] \\ &where \quad t_{i,j} < t_{i+1,j}, i = 1, 2, .., N \end{aligned} \tag{6.9}$$

where $t_{i,j}$ presents the temporal dimension and $s_{i,j} = (x_{i,j}, y_{i,j})$ presents the spatial dimensions. Each user's trajectory could be stored in the user's mobile device, or anonymously at a central server, or both.

The NN classifier is the second component in the framework. The NN considers indoor localization at the room level as a multi-class classification problem, where each user is assigned to one of k mutually exclusive classes. The input to the NN is the estimated movement trajectory of a queried user through the PS. The output of the NN is the room the user is currently in, where each room inside the building of interest is considered as one of the k output classes. For the NN training, crowdsourcing is suggested throughout a training period where participating users inside the building, periodically and anonymously, report their current location (room) to the system. Crowdsourcing has been presented as a viable distributed framework for problem solving in several commercial applications and academic research [177]. The set of movement trajectories and target rooms will comprise the training set for the NN. The NN could be further trained with

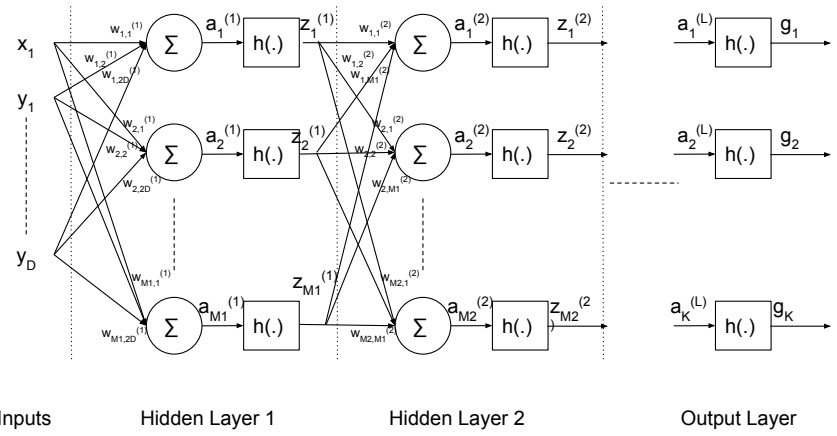

Figure 6.2: Multi-layer feedforward NN structure, with $2D$ inputs, K output nodes, L hidden layers each having M_l nodes

more trajectory data to improve performance.

The request for a user location inside a building could be initiated from the user's mobile device or from the CMRS provider. For example, when an E911 PSAP requests the location of a wireless caller from the CMRS provider. The framework is suggested to be implemented as a mobile application, where users can use the application to identify where they are inside a building (for example museum visitors). The framework could also be implemented as a background tracking application to be recalled when needed (for example a 911 call or for patients within medical facilities or potentially in military operations).

6.3.2 Neural Network Structure

In this work, feed-forward multi-layer perceptron NNs are proposed as multi-class classifiers. The NN has $2D$ inputs, K output nodes, and M_l nodes in each hidden layer l, where $l = 1, 2, ..., L$. $w_{j,i}^{(l)}$ is the weight coefficient connecting node i in layer $l-1$ to node j in layer l, while $w_{j,0}^{(l)}$ is the bias parameter of node j in layer l. Layer $l = 0$ presents the inputs. The NN structure is shown in Fig. 6.2.

The input to the NN, $\mathbf{s} = [x_1, y_1, ..., x_D, y_D]^T$ is the vector of successive x- and y- coordinates of a user for a specific time duration. For example, for a trajectory of duration 1 minute, where the user's position is estimated every 0.5 seconds, the vector length $2D$ will be equal to $1 * 60 * 2/(0.5) = 240$ inputs. The number of output nodes in the final layer is equal to the number of rooms K. The number of hidden layers L and the number of nodes in each layer M_l were chosen based on a series of simulations and are dependent on the specific building layout as will be presented in section IV. In general, using three hidden layers was found to be a good trade-off between complexity and performance.

For each node j in layer l, the output of the node is defined as:

$$z_j^{(l)} = h(a_j^{(l)}) = h\Big(\sum_{i=0}^{M_{l-1}} w_{j,i}^{(l)} z_i^{(l-1)}\Big) = h\Big(\mathbf{w_j^{(l)}} \mathbf{z^{(l-1)}}\Big) \tag{6.10}$$

where $j \in (1, M_l)$, $h(.)$ is the activation function, $\mathbf{w_j^{(l)}}$ is the weight vector at node j of layer l where each element is $w_{j,i}^{(l)}$, and $\mathbf{z^{(l-1)}}$ is the output vector of layer $l-1$. The activation functions in the hidden layers are generally chosen as the non-linear "sigmoid" or "tanh" functions [178]. For a multi-class classifier NN, the activation function selected for the output layer is the softmax function defined as:

$$g_k = \frac{exp(a_k^{(L)})}{\sum_{j=1}^{K} exp(a_j^{(L)})} \tag{6.11}$$

where g_k is the output of node k in the final layer. The softmax activation function satisfies $0 \leq g_k \leq 1$ and $\sum_k g_k = 1$.

Training for the NN is carried out using the Gradient Descent Optimization and Error Backpropagation algorithm. The target outputs for NN training take the binary form $t_k \in \{0, 1\}$, where $t_k = 1$ if the final position in the input vector is located in room k: $\{x_D, y_D\} \in Room\ k$. The NN outputs could be interpreted as $g_k = p(t_k = 1|\mathbf{s})$ [178]. The corresponding error function is the cross-entropy error function defined as:

$$E(\mathbf{w}) = -\sum_{n=1}^{N_t} \sum_{k=1}^{K} t_{nk} ln(g_k(\mathbf{s_n}, \mathbf{w})) \tag{6.12}$$

where $\mathbf{w}$ is the weight matrix and $\mathbf{s_n}$ is the n^{th} training input vector, where $n \in (1, N_t)$ and N_t is the number of training patterns.

6.3.3 Simulation Scenarios

An office building layout was used to perform different simulation scenarios. The layout presents a floor in a general office building with 12 rooms equal in size (3.4m x 3m), 4 other rooms of variable sizes (kitchen, bathroom, copy-room, and stairs) and a hallway. The layout is presented in Fig. 6.1. The purpose of these simulations is to present a proof of concept for the NN-based indoor localization framework.

As presented in Subsection 6.3.1, the proposed framework for indoor localization consists of two components: the PS and the NN. The output of the PS is a set of estimated users' movement trajectories. The performance of the PS, regardless of the specific positioning algorithm implemented, is simulated through first generating random trajectories using the Human Mobility Model presented in the previous subsection.

Table 6.1: Waiting Time τ Distribution and Prior Probabilities p_o for Each Room in the Office Building Layout in Different Simulation Scenarios

Room Number(s)	**Scenarios I**		**Scenarios II**		**Scenario III**		
	τ_{max} in sec	p_o	τ_{max} in sec	p_o	τ_{mean} in min	τ_{std} in min	p_o (average)
1,3,5,10,12,14	45	0.059	45	0.047	30	3.33	0.103
2,4,6,9,11,13	45	0.059	45	0.047	5	0.5	0.001
7,15	20	0.059	20	0.022	2	0.33	0.0054
8	60	0.059	60	0.062	10	1	0.016
16	60	0.059	60	0.062	60	5	0.267
17 (hallway)	10	0.059	10	0.26	1	0.33	0.029

Next, a positioning error is generated as a normal random variable, with zero mean and variable standard deviation $\sigma_\varepsilon \in [0, 35]$m, and added to the true trajectories.

Three different simulation scenarios were studied, based on how the training trajectories were generated, user activity and the prior probabilities of each room in the NN training set. In each simulation scenario, the PS update rate is set to 0.5 sec, and the time duration of trajectories is set to 2 minutes.

For the NN component of the system, the multi-layer feed-forward NN with back propagation learning algorithm, described in Subsection 6.3.2, is used to identify the room location of each user based on the movement trajectory tracked by the PS. For trajectories of duration 2 minutes, and an update rate of 0.5 sec, there are 480 nodes in the input layer. The number of nodes in the output layer is 17, one node representing each room. After careful investigation of several simulation scenarios, three hidden layers were chosen for the NN architecture, having 240, 120, and 60 nodes respectively. Training for each NN was performed with a set of 40,000 random trajectories.

Scenario I

Trajectories of 2 minutes duration were generated with high user activity, i.e. short waiting times in each room destination. The waiting time in each room τ is uniformly distributed, $\tau \sim \mathcal{U}(0, \tau_{max})$, where τ_{max} for each room is shown in Table 6.1. The trajectories for the NN training set are generated such that prior probabilities of NN outputs are equal; i.e. $p_o(k) = p(Room\ ID = k) = 1/K = 1/17$. For this purpose, trajectories for the training set were selected with target destinations.

Scenario II

In this Scenario, the trajectories are generated for a 2-minute duration with the same waiting time distribution as in Scenario I. However, the prior probabilities for NN outputs are not equal. Trajectories were generated with random destinations and cut off times. As seen in Table 6.1, rooms with longer waiting times have higher probabilities for being the user's current location. Although room 17 (the hallway) has the lowest maximum waiting time, because of random termination of the 2-minute trajectories there is higher probability, compared to other rooms, that it's the user's current location since all trajectories pass through the hallway.

Scenario III

In this scenario, slower user activity is studied. Trajectories of 8-hour duration were generated first, with slower and more realistic waiting times. Then, two minutes duration trajectories were extracted and used for the NN training and testing. Waiting time τ in each room is normally distributed, $\tau \sim \mathcal{N}(\tau_{mean}, \tau_{std}^2)$, where τ_{mean} and τ_{std} for each room are shown in Table 6.1. The waiting time distribution in each room was assumed based on the room type (office, copy-room, hallway, etc.). The prior probabilities for each NN output are also not equal in this scenario. The prior probabilities shown in the first three rows in Table 6.1 are the average probabilities for the respective room groups.

6.3.4 Simulation Results and Performance Analysis

The performance of the system is measured by two metrics: the average probability of correct room identification by the NN and the confusion matrix of the NN output. The average probability of correct room identification is calculated as:

$$\begin{gathered} P_{avg} = \sum_{k=1}^{K} p_o(k) P(Room\ ID = k | \{x_D, y_D\} \in Room\ k), \\ Room\ ID = \underset{k}{argmax}\{g_k\}, \end{gathered} \tag{6.13}$$

where g_k is defined as in (6.11). For comparison purposes, the NN performance in each scenario is compared to the performance of two other systems: (1) a basic single-point coordinate localization system where a user is localized in room k if the user's current position coordinates are within the room space, and (2) a single-point multi-layer feedforward NN where the input is only the user's current position (2 inputs). For this NN, there is only 1 hidden layer with 5 nodes. The error function, the output softmax activation function and the training algorithm are the same as the NN structure described in Subsection 6.3.2.

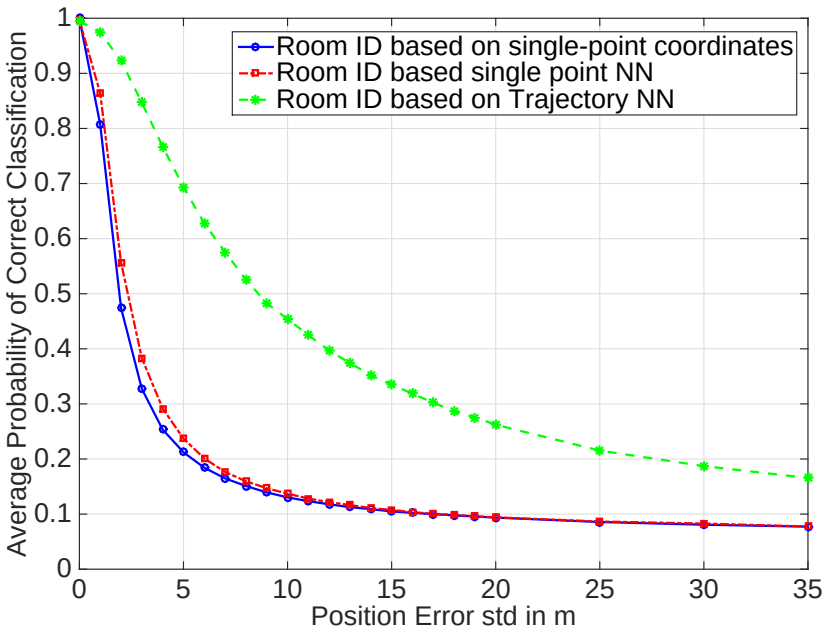

Figure 6.3: NN Scenario I: Average probability of correct room identification versus PS error for the proposed NN vs two single-point systems, with high user mobility and equal priors for NN training.

Scenario I

The average probability of correct room identification for the proposed NN localization system in this simulation scenario, versus the other two single-point systems, is shown in Fig. 6.3 for different values of PS error. As seen in the figure, for moderate positioning errors $\sigma_\varepsilon \in [0, 20]$m, with such high user mobility, the trajectory-based NN provides considerable improvement over both single-point systems. There is an improvement of 45% at $\sigma_\varepsilon = 5$m and of 32% at $\sigma_\varepsilon = 10$m. With equal prior probabilities, both single-point systems perform closely. In all scenarios, the average probability of correct classification of the single-point coordinate localization system can be estimated in a direct manner through calculating the 2-dimensional integrals of the PS error distribution over each room's area.

Scenario II

The system performance for this scenario is shown in Fig. 6.4. Again, the proposed trajectory-based NN provides considerable improvement over both single-points systems for errors up to 15m. However, with non-equal room priors, both NN-based systems converge to a probability equal to $max\{p_0(k)\}$ at high PS errors, i.e. NNs are biased towards predicting the more common classes [179].

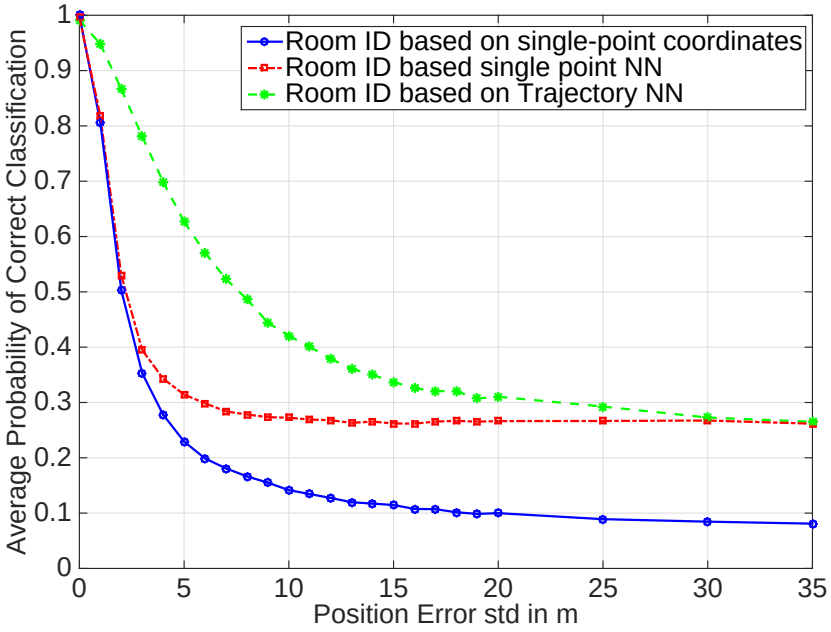

Figure 6.4: NN Scenario II: Performance comparison of the three systems with high user mobility and non-equal priors for NN training.

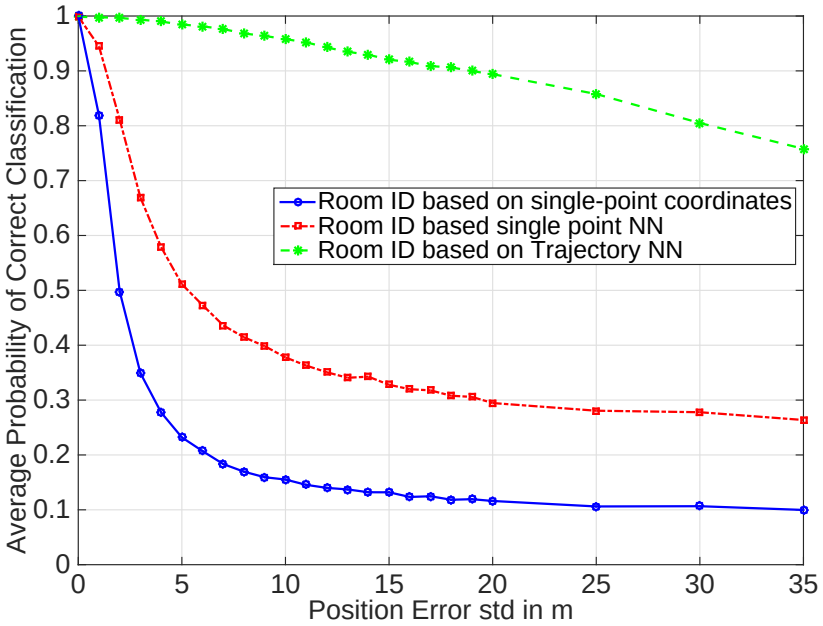

Figure 6.5: NN Scenario III: Performance comparison of the three systems with low user mobility and non-equal priors for NN training.

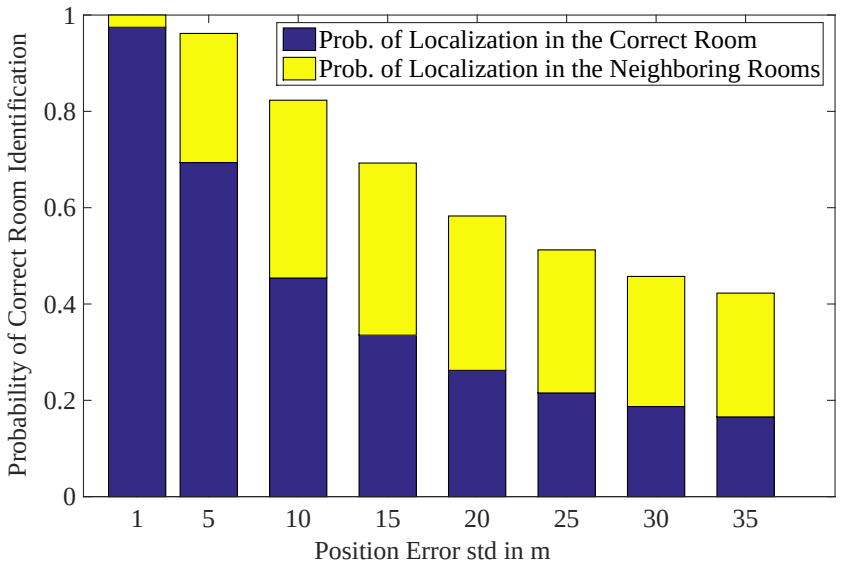

Figure 6.6: NN Scenario I: Probability of correct user localization in the target room, and user localization in the neighboring rooms of the target.

Scenario III

The system performance for this scenario is shown in Fig. 6.5. With slower user mobility, the proposed trajectory-based NN provides huge improvement over both single-points systems for errors up to 35m. At an error with $\sigma_\varepsilon = 35$m, the trajectory-based NN provides a 49% performance improvement as compared to the single-point NN.

NN Confusion Matrices - Identifying Neighboring Rooms

Fig. 6.6 shows the performance of the trajectory-based NN for Scenario I simulation in terms of the probability of correctly identifying the room the user is currently in, versus incorrectly selecting one of its neighboring rooms as the current position derived from the NN confusion matrix. Neighboring rooms are the immediate rooms in the vicinity of the target room. For example, for room 2, neighboring rooms are 1, 3 and 17. As presented in the figure, when the system miss-classifies the target room, it identifies one of the neighboring rooms instead with a probability ranging from 30% at $\sigma_\varepsilon = 35$m to 100% at $\sigma_\varepsilon = 1$m.

6.4 Indoor Localization through Trajectory Similarity Measures

In this section, we propose a new system for enhancing indoor positioning of mobile users exploiting advancements in movement trajectory tracking. Several systems and algorithms, including the GraphSLAM-based approaches we presented in Chapters 4 and 5, proposed novel ways for tracking movement trajectories within buildings. Matching a low accuracy user's movement trajectory estimated through a conventional point-based PS to a set of high-accuracy trajectories tracked through a state-of-the-art tracking system provides a much accurate estimate for the user's current location within a building up to the room level.

6.4.1 System Model

The improved indoor localization system we propose here is based on the assumption that the user is inside a building where the movement trajectories for all users can be tracked efficiently but anonymously through a FC. The FC refers to a general centralized entity within the building that can implement smart and collaborative approaches to track movement of all users inside the building while maintaining their privacy. On the other hand, each user in the building can track his own movement trajectory, while not necessarily with the same level of accuracy. In fact, a practical assumption would be that the user's tracked trajectories through a conventional point-based PS are at a much lower level of accuracy as compared to the trajectories tracked through the smart FC.

The system model in this work, such as the case with the NN classifier, is presented in a 2-dimensional setting; identifying the room number in a single floor. The building FC implements smart and efficient tracking algorithms that track and store the motion trajectories of all in-building personnel as a set of finite sequences of 2-D locations with timestamps, i.e. a set of two-dimensional time-series. At any given time t_N, the building FC will have a set $S = \{S_1, S_2, ..., S_M\}$ of possible motion trajectories of M persons for the duration of t_N time stamps. Each trajectory S_j is defined as in (6.9), represented here:

$$\begin{aligned} S_j &= [(s_{1j}, t_{1j}), (s_{2j}, t_{2j}), ..., (s_{Nj}, t_{Nj})] = [(x_{1j}, y_{1j}, t_{1j}), (x_{2j}, y_{2j}, t_{2j}), ..., (x_{Nj}, y_{Nj}, t_{Nj})] \\ &where \quad t_{ij} < t_{i+1,j}, i = 1, 2, .., N \end{aligned} \tag{6.14}$$

where t_{ij} presents the temporal dimension and $s_{ij} = (x_{ij}, y_{ij})$ presents the spatial dimensions. At time t_{Nu}, a mobile user will have a detected and stored motion trajectory through a point-based PS defined as:

$$\begin{aligned} R &= [(r_1, t_1), (r_2, t_2), ..., (r_{Nu}, t_{Nu})] = [(x_1, y_1, t_1), (x_2, y_2, t_2), ..., (x_{Nu}, y_{Nu}, t_{Nu})] \\ &where \quad t_i < t_{i+1}, i = 1, 2, .., Nu \end{aligned} \tag{6.15}$$

where N_u does not necessarily equal N. In fact, taking the limited resources of the point-based PS system

into consideration, Nu could be much smaller than N. There is no cooperation between the building FC and the point-based PS which implies that the set of trajectories S and the user trajectory R are not synchronized in time. Moreover, the location update rate for both will be different, i.e.:

$$[(t_{i+1,j} - t_{i,j}) \forall S_j \in S, i = 1, 2, .., N] \neq [(t_{i+1} - t_i), i = 1, 2, .., Nu]. \tag{6.16}$$

When it's required to locate a user inside a building, for example, when a E911 PSAP requests the location of a wireless caller from the CMRS provider, the user's PS will be able to provide a distorted version of the user's motion trajectory R for a specific time duration. In parallel, through contacting the building FC, the improved localization system will have access to the set of anonymous users' motion trajectories S. The system then will match the queried user's motion trajectory to the set of motion trajectories tracked by the building's smart network. Thus the system will be able to augment the user's trajectory with a more accurate version and thus providing more accurate location information up to the room level.

In order to match the queried user's trajectory to the closest trajectory tracked by the building FC, different trajectory similarity measures are implemented. A trajectory similarity measure tries to determine if two trajectories are similar based on a specific distance function [180, 181]. Trajectory similarity measures have been applied in a wide spectrum of computing and data management applications such as: urban and transit planning [177, 180], outdoor surveillance systems [181], tracking animal migration patterns, and analysis of top players' movement patterns in different sports [182]. In the following subsection, a discussion of different trajectory similarity measures and how they are applied in the proposed system is presented.

6.4.2 Trajectory Similarity Measures

The improved indoor localization system applies trajectory similarity measures to match the point-based PS tracked trajectory to one of the building FC tracked trajectories. Three different trajectory similarity measures have been investigated as will be briefly presented. These measures are all based on time-shifted matching, i.e. they do not need the two trajectories to have the same time positions [177]. In the following sections, two trajectories $S = [(s_1, t_1), ..., (s_N, t_N)]$ and $R = [(r_1, t_1), ..., (r_M, t_M)]$ of lengths N and M elements respectively are considered as input to each trajectory similarity measure.

Table 6.2 summarizes the different distance functions associated with each trajectory similarity measure and whether it's a metric or non-metric function. A metric distance function is defined as one that is strictly positive, symmetrical, and satisfies the triangle inequality [183]. For large trajectory databases, it is important to minimize the computation of the distance between trajectories in the database. The triangle inequality is an efficient way to apply pruning strategies [184]. Different metric and non-metric trajectory similarity measures have been tested in our proposed system. In the following sections we are only presenting the

Table 6.2: Summary of Distance Functions for Trajectory Similarity Measures

Distance Function	Metric/Non-Metric
$DTW(n,m) = \begin{cases} 0 & n = m = 0 \\ \infty & n = 0 \text{ or } m = 0 \\ dist_{DTW}(n,m) + min\{DTW(n-1,m-1), & \text{otherwise} \\ DTW(n-1,m), DTW(n,m-1)\} & \end{cases}$	Non-Metric
$LCSS(n,m) = \begin{cases} 0 & n = 0 \text{ or } m = 0 \\ LCSS(n-1,m-1)+1 & dist_{LCSS}(n,m) < \varepsilon \\ max\{LCSS(n-1,m), LCSS(n,m-1)\} & \text{otherwise} \end{cases}$	Non-Metric
$EDR(S,R) = \begin{cases} n & m = 0 \\ m & n = 0 \\ EDR(n-1,m-1) & dist_{EDR}(n,m) < \epsilon \\ min\{EDR(n-1,m-1)+1, & \text{otherwise} \\ EDR(n-1,m)+1, EDR(n,m-1)+1\} & \end{cases}$	Non-Metric

similarity measures that provided the best performance.

Dynamic Time Warping Measure (DTW)

The technique of Dynamic Time Warping (DTW) uses a recursive approach to align two trajectories so that some distance measure is minimized [149]. It allows finding a common pattern between the two trajectories when they are of different lengths. The distance is usually measured based on point-to-point L2-norms. For $n = 1, ..., N$ and $m = 1, ..., M$, the distance measure $DTW(n,m)$ is calculated as:

$$DTW(n,m) = dist_{DTW}(n,m) + min\{DTW(n-1,m), \\ DTW(n,m-1), DTW(n-1,m-1)\} \tag{6.17}$$

where the point-to-point distance is calculated as:

$$dist_{DTW}(n,m) = (s_{n,x} - r_{m,x})^2 + (s_{n,y} - r_{m,y})^2 \tag{6.18}$$

The DTW distance is the sum of the distance between current elements and the minimum of the cumulative distances of the neighboring points. The final distance measure between the two trajectories will be equal to $DTW(N,M)$.

DTW is relatively more sensitive to data outliers [180]. Also, for huge data sets, using the DTW measure could be very time consuming. Therefore, usually the recursive formula is limited to a window of size w. For a set of trajectories S_j and a queried trajectory R, a system deploying the DWT trajectory similarity measure will calculate $DTW(N_j, M)$ between each trajectory S_j and R. The trajectory corresponding to

the minimum $DTW(N_j, M)$ will be the closest to trajectory R.

Longest Common Sub-Sequence Measure (LCSS)

The Longest Common Sub-Sequence (LCSS) technique measures the similarity between two trajectories by matching the common sequences in both the trajectories. Any two points $\{s_n, r_m\}$ are considered a match if the distance between the two points is less than ϵ within a window size of δ, where ϵ and δ are application specific parameters that allow flexible matching in both the space domain and the time domain [177]. LCSS quantizes the distance between two points to be 1 or 0 (point match or not) which would remove the larger distance effects [184]. The distance measure $LCSS(n, m)$ is calculated as [150]:

$$LCSS(n,m) = \begin{cases} LCSS(n-1, m-1) + 1, \\ \quad \text{if } dist_{LCSS}(n,m) < \epsilon \text{ and} \\ \quad |n-m| < \delta \\ max\{LCSS(n-1,m), \\ \quad LCSS(n, m-1)\}, \\ \quad \text{otherwise} \end{cases} \tag{6.19}$$

where the point-to-point matching is calculated as:

$$\begin{aligned} &dist_{LCSS}(n,m) < \epsilon \implies |s_{n,x} - r_{m,x}| < \epsilon \text{ and} \\ &|s_{n,y} - r_{m,y}| < \epsilon \end{aligned} \tag{6.20}$$

LCSS is a non-metric measure. Although the binary distance makes the performance of the LCSS measure relatively better in the case of data outliers, it does not take into consideration the gap sizes between similar sub-sequences. For a system deploying the LCSS trajectory similarity measure, the trajectories with the maximum $LCSS(N, M)$ will be the most similar.

Edit Distance on Real Sequence Measure (EDR)

In computer science applications, Edit Distance (ED) is a way of quantifying how dissimilar two strings are by counting the minimum number of operations (insert, delete, and substitute) required to transform one string into the other [185]. Applying a similar approach in trajectory similarity, ED-based measures quantify the dissimilarity between two trajectories by measuring the mismatches among the trajectories. Contrary to LCSS, ED based measures assign penalties to the gaps between two matched sub-trajectories according to the lengths of gaps, which overcomes a main drawback in LCSS [182].

There are several trajectory similarity measures based on the ED concept. In this work, we will only present the Edit Distance on Real Sequence Measure (EDR). Similar to the LCSS measure, the EDR measure uses a threshold ϵ to detect point-by-point matching. In addition, it quantizes the cost assigned to a point mismatch to be 1 [182]. There is no penalty in the case of a point match. The distance measure $EDR(n, m)$ is calculated as:

$$EDR(n,m) = \begin{cases} EDR(n-1, m-1), \\ \quad \text{if } dist_{EDR}(n,m) < \epsilon \\ min\{EDR(n-1,m)+1, \\ \quad EDR(n,m-1)+1, \\ \quad EDR(n-1,m-1)+1\}, \\ \quad \text{otherwise} \end{cases} \tag{6.21}$$

where the point-to-point matching $dist_{EDR}(n, m) < \epsilon$ is calculated as in (6.19). For a system deploying the EDR trajectory similarity measure, the trajectories with the minimum $EDR(N, M)$ will be declared to be the similar ones.

6.4.3 Simulations and Results

In this section, we present different simulation scenarios and results and discuss the performance of the proposed indoor localization system. The purpose of the simulation is to study and analyze the performance of the proposed approach for indoor localization enhancement. Different scenarios were simulated in which a varying number of mobile users are moving inside a building, and it's required to locate one of those users inside the building at the room level. The main performance metrics applied are the probability of identifying the correct trajectory, and the probability of detecting the correct room in which the queried user is.

Two different building layouts were used in simulation scenarios. Layout 1 presents a floor in a general office building with most of the rooms equal in size (3m x 3m). Layout 2 presents a floor in a research lab facility, where room sizes vary from small (2m x 2m) rooms up to meeting and lecture rooms of (10m x 10m) as well as different hallways and lobbies. The layouts are shown in Fig. 6.7 and 6.8 respectively. In order to study and simulate the proposed system, large numbers of random movement trajectories were generated using the human mobility model presented in Section 6.2. These generated trajectories are considered as the true movement trajectories of users inside the building. They are used as a reference for localizing the true position of each user in terms of the final room ID.

For every user inside the building, the building FC is tracking a set of trajectories S_J. Each trajectory is

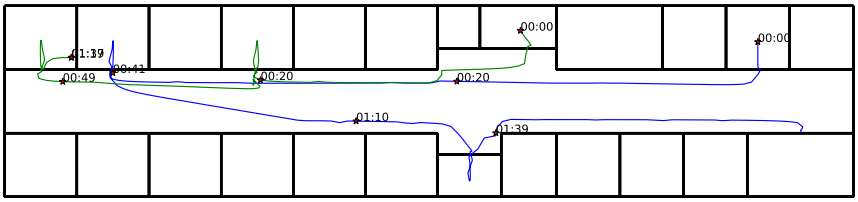

Figure 6.7: Two motion trajectories (user 1 in green and user 2 in blue) in an office building with mostly equal-sized rooms (3m x 3m).

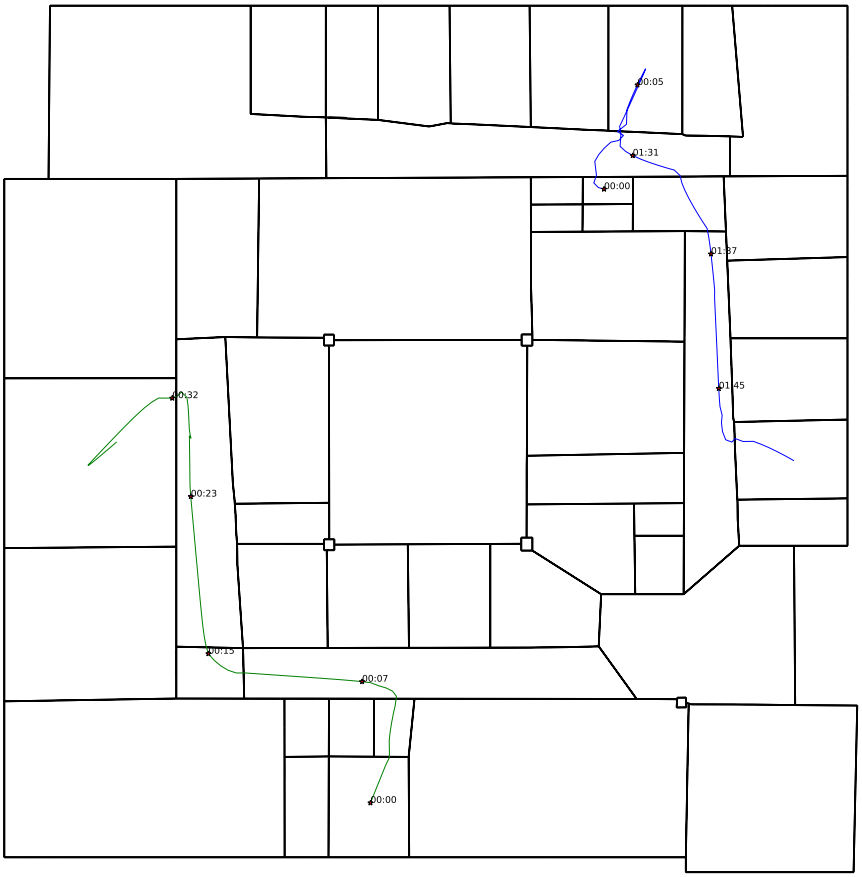

Figure 6.8: Two motion trajectories (user 1 in green and user 2 in blue) in a research lab facility. Room sizes vary from (2m x2m) up to (10m x 10m).

generated as a sequence of Gaussian random variables with the mean values equal to the true trajectory values in the space domain and the variances equal to the building localization error. Different error values were used in the range [0.5-2m] in different simulation scenarios. The time stamps for these trajectories are spaced at 0.5 second in time.

For the queried user, the point-based PS user location tracking system will have a much more distorted version of the true trajectory R, and usually with a lower update rate. These trajectories are generated in the same manner, but with higher values of localization errors. In each simulation scenario, the error for the PS location system was varied from [0-10m], while the time stamps were at 2 seconds apart.

Several simulations were conducted for the two different layouts, with varying number of users inside each building. The trajectories were compared over different time frames. The improved localization system compares the queried user trajectory R with the set of trajectories S_J through the trajectory similarity measures, and generates a decision of which trajectory in the set S_j is the closest to R. Next, the system will use the selected trajectory S_j to localize the user inside the building in terms of a room ID.

The performance of the improved system could be measured by two metrics: the probability of correct trajectory identification and the probability of correct room identification. Each poses limitation on the overall performance as will be presented in the following. In addition, several different parameters affect both the performance metrics. Here, we are only presenting two parameters: the trajectory length, and the smart building localization error.

Trajectory Similarity Measures Performance

Trajectory similarity measures performance is measured in terms of correct trajectory identification with variable cellular system localization error. One of the simulation scenarios for the trajectory similarity performance is presented in Fig. 6.9, for the office building layout with 20 users inside the building, a trajectory comparison time frame of 2 minutes and a smart building localization error of 1m. In this scenario all three measures were able to identify the correct trajectories with a probability of at least 70%. As shown in Fig. 6.9, the performance of the trajectory similarity measures is independent of the cellular system localization error, i.e. how much the queried trajectory is distorted, up to an error of 10m. This can be explained as follows: on average, the 20 users' motion trajectories are random such that even with a very highly distorted queried trajectory, the trajectory similarity measures are able to match the distorted trajectory to the correct one. In addition, the probability of correct trajectory identification never goes to 100%. This is due to the fact that with a small time frame such as 2 minutes, more than one user could be inside the same room for most of the 2-minute frame in which case the system would not be able to differentiate the two corresponding trajectories even with very low cellular system localization error.

Overall System Performance

The overall system performance is measured in terms of the current location of the user, i.e. the final room ID. As presented in Fig. 6.10 with the same simulation setup, the system performance will always exceed the

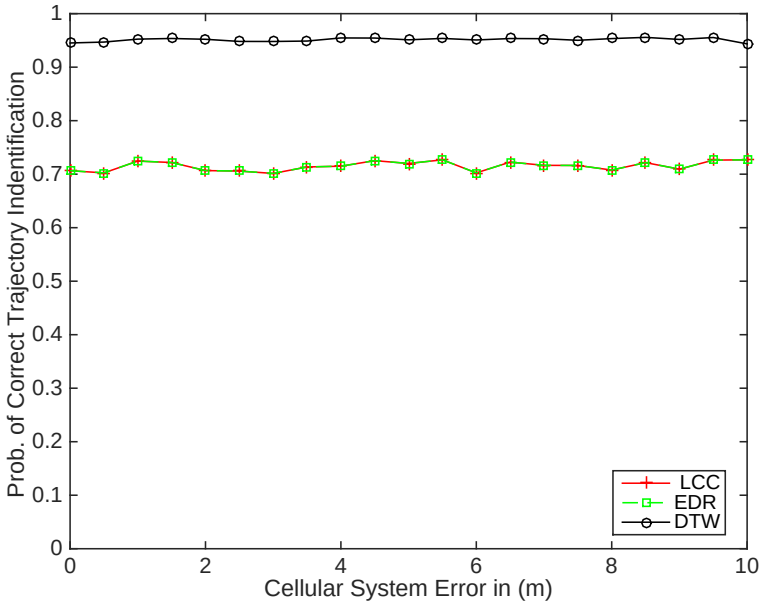

Figure 6.9: Probability of correct trajectory identification vs the point-based PS error up to 10m.

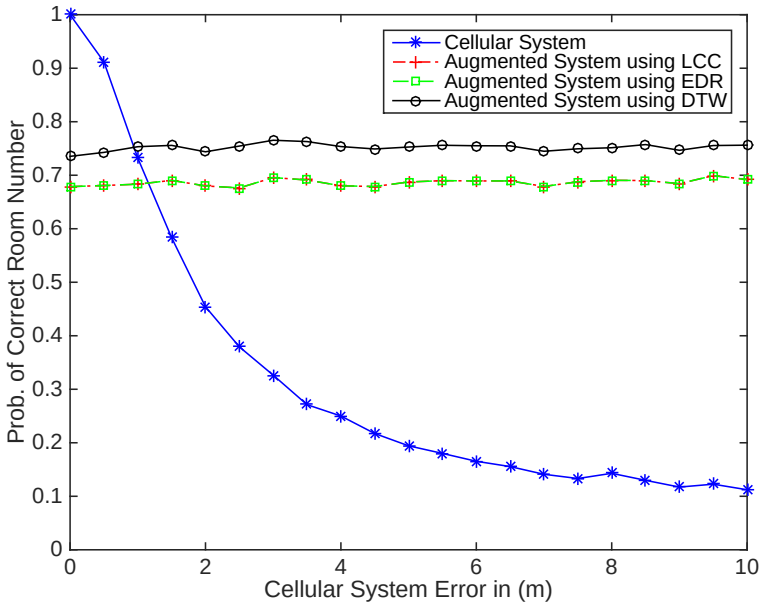

Figure 6.10: Probability of correct room number vs the point-based PS error up to 10m.

cellular system performance except for the unlikely case when the cellular localization error is smaller than the smart building localization error. Based on the performance analysis, identifying the correct trajectory

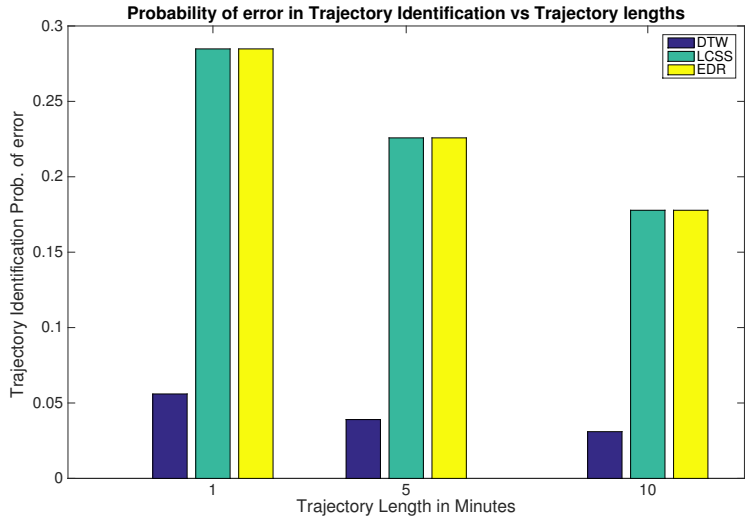

Figure 6.11: Trajectory Similarity Measures Performance in terms of probability of wrong identification with different trajectory duration times [1, 5, and 10 minutes].

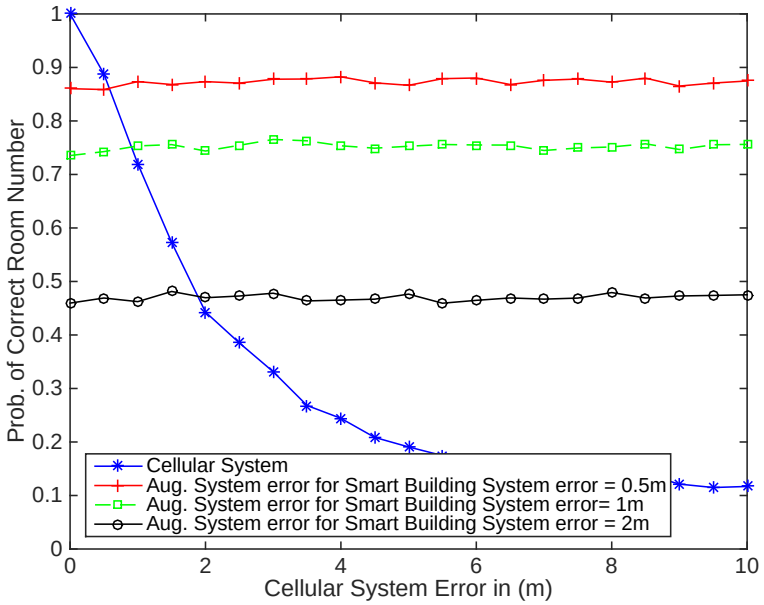

Figure 6.12: Overall system performance using DTW trajectory similarity measure for different values of the localization error of the smart building system.

does not mean necessarily identifying the correct room ID, due to the smart building localization error. With smaller room sizes, even with correct trajectory identification, the system could identify any of the

neighboring rooms as the current location instead. In addition, selecting a wrong trajectory does not mean necessarily identifying a wrong room ID, as the similarity measures could select a very similar trajectory that ends in the same room.

Trajectory Time Duration

The effect of the trajectory comparison time frame on the performance of the trajectory similarity measures has been investigated for trajectory time frames of 1, 5, and 10 minutes. The same simulation setup was used with a constant cellular localization error of 6m. Fig. 6.11 presents the average probability of error in trajectory identification using DTW, LCSS, and EDR measures respectively for the different trajectory lengths. The probability of detecting the correct trajectory is directly related to the capability of tracking and storing longer trajectories.

Smart Building Localization Error

The effect of the smart building localization error on the overall performance of the system has been investigated as well. The overall probability of correct room detection was calculated for different values of smart building localization error: 0.5, 1, and 2 meters using only the DTW similarity measure. As shown in Fig. 6.12, the error of the smart building localization is a limiting factor in determining the overall system performance. The effect of the building localization error will also be variable for different building layouts and room sizes.

6.5 Conclusions

In this chapter, we presented novel approaches for improving indoor localization accuracy based on two different approaches: NN and trajectory similarity measures. The objective of the two approaches is to correctly identify the indoor positions of mobile users at the room level. Both systems assume the mobile user equipment is able to track a low-accuracy version of its movement trajectory through a point-based PS. Then each system utilizes the low-accuracy movement trajectory to generate a much better estimate of the user's current location through pattern recognition techniques.

The NN system then applies a multi-layer feed-forward NN to the movement trajectory to identify the current room the user is located in. The proposed trajectory-based NN provides significant improvement in performance compared to single-point PSs. The correct room identification probability, using the proposed system, has an improvement of up to 49% with variable positioning system error variance in different sim-

ulation scenarios. The NN performance was studied with different user mobility levels and different prior room probabilities. In addition, when the NN miss-classifies the target room, it identifies one of the neighboring rooms instead with at least 30% probability.

The improved localization system based on trajectory similarity matches the low-accuracy version of the queried user movement trajectory to the set of anonymous trajectories tracked by the smart building FC and find the closest one. Identifying the corresponding high-accuracy trajectory, the improved system will be able to provide accurate user location information up to the room level. For this system, three different trajectory similarity measures were investigated in two different layouts. Through matching trajectories, the combined system is able to provide indoor location information at the room level with accuracy up to 75% for the PS system localization errors up to 10m for trajectory time frames of 2 minutes.

Chapter 7

Conclusions

This dissertation investigated the application of sensor fusion techniques to two main problems in wireless communications: Automatic Modulation Classification (AMC) and indoor localization and mapping. For both problems, we proposed novel algorithms for combining measurements from individual sensors, whether the sensors were at different locations, or within a single node. We conclude the dissertation by summarizing our findings.

Sensor Fusion for Automatic Modulation Classification

We proposed a collaborative Maximum Likelihood (ML) cumulants-based modulation classification framework as a means to improve the performance of a single node. In order to provide a comprehensive and a tractable analysis of the performance of the collaborative algorithm, we first re-visited the performance of a single-node ML cumulants-based modulation classifier and provided a comprehensive analysis of its performance in terms of the probability of correct classification in different scenarios. The analysis was next extended to the collaborative classification framework. We presented theoretical analysis and extensive simulations that demonstrated how the collaborative framework improves the overall performance, even for extreme cases when any of the cooperating nodes is receiving signals with much lower SNR. We also demonstrated the validity of the proposed collaborative algorithm for the case of correlated sensor signals.

Indoor Localization and Mapping

In the second part of the dissertation we sought to address three questions. Here are our findings for those questions:

How can smartphone sensor information be fused to track movement trajectories within buildings?

We proposed an improved algorithm for pedestrian GraphSLAM using smartphones-based PDR measurements. We proposed two techniques to overcome the drift errors in the heading direction measurements acquired through the smartphone gyroscope and magnetometer: (1) heading direction calibration algorithm, and (2) a pre-processing heading detection stage for the GraphSLAM algorithm. We presented a model for displacement measurement errors based on combining error models of the step-length estimation and the heading direction estimation. We used the proposed techniques in real-world experiments through iPhone 7 devices to collect measurements from smartphone sensors and estimate movement trajectories within an academic building. Experiments illustrated that the improved pedestrian GraphSLAM is able to provide high accuracy movement trajectory estimates that overcome the trajectory rotation problem, and closely match the true ones (typically within 1m of accuracy).

How can smartphones' sensor information within the same environment be fused to provide better trajectory estimates for their users?

We proposed a novel multiple-user GraphSLAM algorithm using BLE RSS measurements between smartphones within close proximity as an alternative to loop closures and landmark detection. In order to exploit BLE RSS measurements within the pedestrian GraphSLAM algorithm, we investigated the BLE RSS measurements between iPhone 7 devices collected though an iPhone app that we developed specifically for the purpose of this research. Hence, we proposed a new technique for extracting relative distance information between two smartphone devices based on BLE RSS, rather than absolute distance values. The relative distances between smartphones in close proximity are used as constraints to the GraphSLAM optimization problem. Experiments conducted within an academic building between two users walking through intersecting trajectories illustrated that the proposed collaborative algorithm provided high accurate estimates of movement trajectories without loop closures or landmark detection.

How can movement trajectories be used for localization?

In the final part of this dissertation we proposed novel indoor localization algorithms based on movement trajectories to provide high accuracy position estimates for indoor users up to the room level. Two frameworks were proposed: Neural Networks as multi-class classifiers and trajectory similarity measures. We presented analysis and simulations for both frameworks in different scenarios, where the efficiency of both was validated.

General Insights and Future directions

Through the work of this dissertation, we moved between two very interesting problems in wireless communications, and from a mainly theoretical framework of the modulation classification problem to a very practical treatment of the indoor localization problem. Each problem presented its own unique set of challenges that required different approaches to provide efficient solutions.

While modulation classification is a well-established field, application of sensor fusion algorithms is very promising in providing reliable and consistent classification performance. Implementing sensor fusion with advanced artificial intelligence and machine learning techniques will provide much powerful solutions to the problem. Machine learning approaches can also be an attractive approach to the indoor localization problem. Smart and novel ways to leverage the diversity of sensors embedded in smartphones are needed to provide more efficient and accurate solutions to the indoor localization problem. Extensions to the work presented in this dissertation in the field of tracking movement trajectories also include what we may refer to as map fusion. Topological maps resulting from tracking users' movement trajectories can be combined to build unified maps of building layout. Smart map fusion techniques are needed for matching, especially for overlapping maps.

Finally, the study of computational requirements for the proposed trajectory tracking algorithms to be implemented on smartphones in real-time is a promising area of future research. Development of real-time applications to run on smartphones for simultaneous localization and mapping is the next step in smartphones-based SLAM approaches.

Appendices

Appendix A

Equations for High-order moments and cumulants

Based on (2.1) and (2.2) presented in [65, 90], equations for calculating even-order moments and cumulants up to the 8^{th}-order are presented in the following. For symmetrically distributed random processes, as the digital modulation schemes of interest, odd-order moments and cumulants are equal to zero. For real-valued modulations:

$$\begin{aligned} C_2 &= M_2, & C_6 &= M_6 - 15M_4M_2 + 30M_2^3, \\ C_4 &= M_4 - 3M_2^2, & C_8 &= M_8 - 28M_6M_2 - 35M_4^2 + 420M_4M_2^2 - 630M_2^4, \end{aligned} \tag{A.1}$$

and the corresponding high-order moments:

$$\begin{aligned} M_2 &= C_2, & M_6 &= C_6 + 15C_2C_4 + 15C_2^3, \\ M_4 &= C_4 + 3C_2^2, & M_8 &= C_8 + 28C_2C_6 + 210C_4C_2^2 + 35C_4^2 + 105C_2^4. \end{aligned} \tag{A.2}$$

For complex-valued modulation schemes, i.e. QAM and PSK, $C_{20} = M_{20} = 0$. High-order cumulants are presented as:

$$\begin{aligned} C_{20} &= M_{20} = 0, & C_{40} &= M_{40}, \\ C_{21} &= M_{21}, & C_{42} &= M_{42} - 2M_{21}^2, \\ C_{60} &= M_{60}, & C_{80} &= M_{80} - 35M_{40}^2, \\ C_{61} &= M_{61} - 5M_{21}M_{40}, & C_{82} &= M_{82} - 12M_{61}M_{21} - 15M_{40}M_{42} + 60M_{40}M_{21}^2, \\ C_{62} &= M_{62} - 8M_{41}M_{21}, & C_{84} &= M_{84} - 16M_{63}M_{21} - |M_{40}|^2 - 18M_{42}^2 \\ C_{63} &= M_{63} - 9M_{42}M_{21} + 12M_{21}^3, & & \quad + 144M_{42}M_{21}^2 - 144M_{21}^4, \end{aligned} \tag{A.3}$$

and the corresponding high-order moments:

$$
\begin{aligned}
&M_{20} = C_{20} = 0, && M_{40} = C_{40},\\
&M_{21} = C_{21}, && M_{42} = C_{42} + 2C_{21}^2,\\
&M_{60} = C_{60}, && M_{80} = C_{80} + 35C_{40}^2,\\
&M_{61} = C_{61} + 5C_{21}C_{40}, && M_{82} = C_{82} + 12C_{61}C_{21} + 15C_{40}C_{42} + 30C_{40}C_{21}^2,\\
&M_{62} = C_{62} + 8C_{41}C_{21}, && M_{84} = C_{84} + 16C_{63}C_{21} + |C_{40}|^2 + 18C_{42}^2\\
&M_{63} = C_{63} + 9C_{42}C_{21} + 6C_{21}^3, && \qquad +72C_{42}C_{21}^2 + 24C_{21}^4.
\end{aligned}
\tag{A.4}
$$

Appendix B

Parameters of Large Sample Estimates of Cumulants

The following calculations follow the approach presented in Swami et al. [65].

B.1 Second-order Cumulants

Based on (2.7) and the relations presented in Appendix A, the expected values and the variances of the 2^{nd}-order cumulants, both real- and complex-valued, can be calculated as:

$$\begin{aligned} &E\left[\hat{C}_2\right] = C_2, \quad &&Var\left[\hat{C}_2\right] = \left(C_4 + 2C_2^2\right)/N, \\ &E\left[\hat{C}_{20}\right] = C_{20}, \quad &&Var\left[\hat{C}_{20}\right] = \left(C_{42} + 2C_{21}^2\right)/N, \\ &E\left[\hat{C}_{21}\right] = C_{21}, \quad &&Var\left[\hat{C}_{21}\right] = \left(C_{42} + C_{20}^2 + C_{21}^2\right)/N. \end{aligned} \tag{B.1}$$

B.2 Fourth-order Cumulants

B.2.1 Real-Valued Modulation Schemes

From (A.1), the expected value of the 4^{th}-order cumulant $\hat{C}_4$ of real-valued modulation schemes can be calculated as:

$$E[\hat{C}_4] = E[\hat{M}_4] - 3E[\hat{M}_2^2], \tag{B.2}$$

where,

$$\begin{aligned} E[\hat{M}_4] &= M_4, \\ E[\hat{M}_2^2] &= E[(\frac{1}{N}\sum_{i=1}^{N} y(i)^2)(\frac{1}{N}\sum_{j=1}^{N} y(j)^2)] \\ &= \frac{1}{N^2}E[\sum_{i=1}^{N} y(i)^4 + \sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N} y(i)^2 y(j)^2] \\ &= \frac{1}{N^2}[NE[\hat{M}_4] + N(N-1)(E[\hat{M}_2])^2] \\ &= \frac{1}{N}[M_4 + (N-1)M_2^2]. \end{aligned} \tag{B.3}$$

Through substituting from (B.3) in (B.2) and substituting the values of M_2 and M_4 from (A.2):

$$E[\hat{C}_4] = \left(\frac{N-3}{N}\right) C_4 - \frac{6}{N} C_2^2, \tag{B.4}$$

which is the asymptotic mean of $\hat{C}_4$. The variance of $\hat{C}_4$ is:

$$Var[\hat{C}_4] \;=\; Var[\hat{M}_4] + 9\,Var[\hat{M}_2^2] - 6\,Co\,Var[\hat{M}_4\hat{M}_2^2] \tag{B.5}$$

where

$$Var[\hat{M}_4] = \frac{1}{N}[M_8 - M_4^2], \tag{B.6}$$

$$\begin{aligned} Var[\hat{M}_2^2] &= E[\hat{M}_2^4] - (E[\hat{M}_2^2])^2, \\ E[\hat{M}_2^4] &= E[(\frac{1}{N}\sum_{i=1}^{N} y(i)^2)^4] \\ &= \frac{1}{N^4}E[\sum_{i=1}^{N} y(i)^8 + 3\sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N} y(i)^4 y(j)^4 + 4\sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N} y(i)^2 y(j)^6 \\ &+6\sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N}\sum_{\substack{k=1 \\ k\neq i \\ k\neq j}}^{N} y(i)^2 y(j)^2 y(k)^4 + \sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N}\sum_{\substack{k=1 \\ k\neq j}}^{N}\sum_{\substack{m=1 \\ m\neq i \\ m\neq j \\ m\neq k}}^{N} y(i)^2 y(j)^2 y(k)^2 y(m)^2] \\ &= \frac{1}{N^3}M_8 + \frac{3(N-1)}{N^3}M_4^2 + \frac{4(N-1)}{N^3}M_2 M_6 + \frac{6(N-1)(N-2)}{N^3}M_2^2 M_4 \\ &+\frac{(N-1)(N-2)(N-3)}{N^3}M_2^4, \\ E[\hat{M}_2^2] &= \frac{1}{N}[M_4 + (N-1)M_2^2] \quad \text{from (B.3)}, \\ Var[\hat{M}_2^2] &= \frac{1}{N^3}M_8 + \frac{2N-3}{N^3}M_2^2 + \frac{4N-4}{N^3}M_{21}M_6 + \frac{4N^2-16N+12}{N^3}M_4 M_2^2 \\ &+\frac{-4N^2+10N-6}{N^3}M_2^4 \end{aligned} \tag{B.7}$$

and

$$
\begin{aligned}
CoVar[\hat{M}_4\hat{M}_2^2] &= E[\hat{M}_4\hat{M}_2^2] - E[\hat{M}_4]E[\hat{M}_2^2] \\
E[\hat{M}_4\hat{M}_2^2] &= E[(\frac{1}{N}\sum_{i=1}^{N} y(i)^4)(\frac{1}{N}\sum_{j=1}^{N} y(j)^2)^2] \\
&= \frac{1}{N^3}E[\sum_{i=1}^{N} y(i)^8 + \sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N} y(i)^4 y(j)^4 \\
&+2\sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N} y(i)^6 y(j)^2 + \sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N}\sum_{\substack{k=1 \\ k\neq i \\ k\neq j}}^{N} y(i)^4 y(j)^2 y(j)^2] \\
&= \frac{1}{N^2}M_8 + \frac{(N-1)}{N^2}M_4^2 + \frac{2(N-1)}{N^2}M_6M_2 + \frac{(N-1)(N-2)}{N^2}M_4M_2^2, \\
E[\hat{M}_4] &= M_4, \\
E[\hat{M}_2^2] &= \frac{1}{N}[M_4 + (N-1)M_2^2] \quad \text{from (B.3)}, \\
CoVar[\hat{M}_4\hat{M}_2^2] &= \frac{1}{N^2}M_8 + \frac{-1}{N^2}M_4^2 + \frac{2(N-1)}{N^2}M_6M_2 + \frac{-2N+2}{N^2}M_4M_2^2
\end{aligned}
\tag{B.8}
$$

From (B.6-B.8) into (B.5), the variance of the sample estimate of C_4 can be expressed as:

$$Var[\hat{C}_4] = AM_8 + BM_4^2 + CM_2^4 + DM_4M_2^2 + EM_2M_6, \tag{B.9}$$

where

$$
\begin{aligned}
A &= \frac{N^2-6N+9}{N^3}, \\
B &= \frac{-N^2+24N-27}{N^3}, \\
C &= \frac{-36N^2+90N-54}{N^3}, \\
D &= \frac{48N^2-156N+108}{N^3}, \\
E &= \frac{-12N^2+48N-36}{N^3}.
\end{aligned}
\tag{B.10}
$$

Substituting for the values of M_2, M_4, M_6 and M_8 from (A.2) into (B.9), the variance of the sample estimate of C_4 can be expressed as a function of C_2 which corresponds to the average signal energy:

$$Var[\hat{C}_4] = m_0 + m_1C_2 + m_2C_2^2 + m_4C_2^4, \tag{B.11}$$

where

$$\begin{aligned} m_0 &= AC_8 + [35A + B]C_4^2, \\ m_1 &= [28A + E]C_6, \\ m_2 &= [210A + 6B + D + 15E]C_4, \\ m_4 &= 105A + 9B + C + 3D + 15E \end{aligned} \tag{B.12}$$

For constant modulus modulation schemes, i.e. BPSK, where $y(n) = \pm a$, all the moments in (B.9) are equal: $M_8 = M_4^2 = M_2^4 = M_4 M_2^2 = M_2 M_6 = a^8$, without noise added to the signal. In this case:

$$Var[\hat{C}_4] = M_8[A + B + C + D + E] = 0; \tag{B.13}$$

B.2.2 Complex-Valued Modulation Schemes

From (A.3), the expected value of $\hat{C}_{42}$ can be calculated as:

$$E[\hat{C}_{42}] = E[\hat{M}_{42}] - 2E[\hat{M}_{21}^2], \tag{B.14}$$

Following the same procedure as the parameters of the 4^{th}-order cumulant $\hat{C}_4$ of real-valued modulation schemes, the asymptotic mean of $\hat{C}_{42}$ is expressed as:

$$E[\hat{C}_{42}] = \left(\frac{N-2}{N}\right) C_{42} - \frac{2}{N} C_{21}^2, \tag{B.15}$$

The variance of $\hat{C}_{42}$ is:

$$Var[\hat{C}_{42}] = Var[\hat{M}_{42}] + 4\,Var[\hat{M}_{21}^2] - 4\,Co\,Var[\hat{M}_{42}\hat{M}_{21}^2] \tag{B.16}$$

Following the same approach in (B.6)-(B.8), the variance of the sample estimate of C_{42} is expressed as:

$$Var[\hat{C}_{42}] = FM_{84} + GM_{42}^2 + HM_{21}^4 + IM_{42}M_{21}^2 + JM_{21}M_{63} \tag{B.17}$$

where

$$\begin{aligned} F &= \frac{N^2 - 4N + 4}{N^3}, \\ G &= \frac{-N^2 + 12N - 12}{N^3}, \\ H &= \frac{-16N^2 + 40N - 24}{N^3}, \\ I &= \frac{24N^2 - 72N + 48}{N^3}, \\ J &= \frac{-8N^2 + 24N - 16}{N^3}. \end{aligned} \tag{B.18}$$

Substituting for the values of M_{84}, M_{42}, M_{21} and M_{63} from (A.4) into (B.17), the variance of the sample estimate of C_{42} can be expressed as a function of C_{21} which corresponds to the average signal energy:

$$Var[\hat{C}_{42}] = p_0 + p_1 C_{21} + p_2 C_{21}^2 + p_4 C_{21}^4 \tag{B.19}$$

where

$$\begin{aligned} p_0 &= FC_{84} + F|C_{40}|^2 + [18F + G]C_{42}^2, \\ p_1 &= [16F + J]C_{63}, \\ p_2 &= [72F + 4G + I + 9J]C_{42}, \\ p_4 &= 24F + 4G + H + 2I + 6J. \end{aligned} \tag{B.20}$$

For constant modulus modulation schemes, i.e. QPSK, where $y(n) = \pm a \pm ja$, all the moments terms in (B.17) are equal: $M_{84} = M_{42}^2 = M_{21}^4 = M_{42}M_{21}^2 = M_{21}M_{63} = (2a)^8$, without noise added to the signal. In this case:

$$Var[\hat{C}_{42}] = M_{84}[F + G + H + I + J] = 0; \tag{B.21}$$

Bibliography

[1] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, "A Step, Stride and Heading Determination for the Pedestrian Navigation System," *Journal of Global Positioning Systems*, vol. 3, no. 1&2, pp. 273–279, Dec. 2004.

[2] R. C. Luo, C. C. Chang, and C. C. Lai, "Multisensor fusion and integration: Theories, applications, and its perspectives," *IEEE Sensors Journal*, vol. 11, no. 12, pp. 3122–3138, 2011.

[3] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[4] S. S. Iyengar, S. Sastry, and N. Balakrishnan, "Foundations of data fusion for automation," *IEEE Instrumentation and Measurement Magazine*, vol. 6, no. 4, pp. 35–41, 2003.

[5] C. Pohl and J. L. Van Genderen, "Review article Multisensor image fusion in remote sensing: Concepts, methods and applications," *Int. J. Remote Sensing*, vol. 19, no. 5, pp. 823–854, 1998.

[6] D. Smith and S. Singh, "Approaches to multisensor data fusion in target tracking: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 12, pp. 1696–1710, 2006.

[7] R. C. Luo and C. C. Chang, "Multisensor fusion and integration: A review on approaches and its applications in Mechatronics," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 49–60, 2012.

[8] H. Chung, S. Oh, D. H. Shim, and S. S. Sastry, "Toward robotic sensor webs: Algorithms, systems, and experiments," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1562–1586, 2011.

[9] M. Ayaz, M. Ammadu Din, I. Baig, and E. H. M. Aggoune, "Wireless Sensor's Civil Applications, Prototypes and Future Integration Possibilities: A Review," *IEEE Sensors Journal*, vol. 18, no. 1, pp. 4–30, 2017.

[10] D. Evans, "The Internet of Things - How the Next Evolution of the Internet is Changing Everything," *CISCO white paper*, 2011. [Online]. Available: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[11] S. Dudley, W. Headley, M. Lichtman, E. Imana, X. Ma, M. Abdelbar, A. Padaki, A. Ullah, M. Sohul, T. Yang, and J. Reed, "Practical issues for spectrum management with cognitive radios," *Proc. IEEE*, vol. 102, no. 3, pp. 242–264, Mar. 2014.

[12] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021 [Visual Networking Index (VNI)]," *Cisco White paper*, 2012. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf

[13] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive Mechanisms for Crowdsensing: Crowdsourcing With Smartphones," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1732–1744, jun 2016.

[14] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Commun. Mag.*, vol. 48, no. 9, pp. 140–150, 2010.

[15] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[16] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201 – 220, Feb. 2005.

[17] T. Clancy, A. Khawar, and T. Newman, "Robust signal classification using unsupervised learning," *IEEE Trans. Wireless Commun.*, vol. 10, no. 4, pp. 1289–1299, Apr. 2011.

[18] S. Amuru and R. M. Buehrer, "Optimal jamming strategies in digital communications - impact of modulation," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec 2014, pp. 1619–1624.

[19] C. Bergstrom, S. Chuprun, S. Gifford, and G. Maalouli, "Software defined radio (SDR) special military applications," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, vol. 1, Oct. 2002, pp. 383–388 vol.1.

[20] O. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Commun.*, vol. 1, no. 2, pp. 137–156, Apr. 2007.

[21] J. Xu, W. Su, and M. Zhou, "Distributed automatic modulation classification with multiple sensors," *IEEE Sensors J.*, vol. 10, no. 11, pp. 1779 –1785, Nov. 2010.

[22] Y. Zhang, N. Ansari, and W. Su, "Multi-sensor signal fusion based modulation classification by using wireless sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2011, pp. 1–5.

[23] ——, "Optimal decision fusion based automatic modulation classification by using wireless sensor networks in multipath fading channel," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, 2011, pp. 1–5.

[24] W. Su, "Signal Sensing and Modulation Classification Using Pervasive Sensor Networks," in *1st Int. Workshop Cognitive Computing and Commun.*, 2013, pp. 441–446.

[25] ——, "Modulation Classification of Single-Input Multiple-Output Signals Using Asynchrounous Sensors," *IEEE Sensors J.*, vol. 15, no. 1, pp. 346–357, 2015.

[26] G. Markovic and M. Dukic, "Cooperative AMC schemes using cumulants with hard and soft decision fusion," in *20th Telecommun. Forum (TELFOR)*, 2012, pp. 400–403.

[27] ——, "Decision fusion methods for automatic modulation classification with multiple sensors in multipath fading channels," in *IEEE EUROCON*, Jul. 2013, pp. 105–112.

[28] ——, "Cooperative modulation classification with data fusion for multipath fading channels," *Electron. Lett.*, vol. 49, no. 23, pp. 1494–1496, 2013.

[29] G. B. Markovic, "Cooperative modulation classification by using multiple sensors in dispersive fading channels," in *22nd Telecommun. Forum (TELFOR)*, 2014, pp. 264–271.

[30] G. B. Markovic and M. L. Dukic, "Joint cumulant estimate correction and decision for cooperative modulation classification by using multiple sensors," *Ann. Telecommun.*, vol. 70, no. 5-6, pp. 197–206, 2015.

[31] B. Dulek, O. Ozdemir, P. K. Varshney, and W. Su, "Distributed maximum likelihood classification of linear modulations over nonidentical flat block-fading gaussian channels," *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 724–737, 2015.

[32] B. Dulek, "An Online and Distributed Approach for Modulation Classification Using Wireless Sensor Networks," *IEEE Sensors J.*, vol. 17, no. 6, pp. 1781–1787, 2017.

[33] ——, "Online Hybrid Likelihood based Modulation Classification using Multiple Sensors," *IEEE Trans. Wireless Commun.*, vol. PP, no. 99, p. 1, 2017.

[34] Y. Gu, A. Lo, and I. Niemegeers, "A Survey of Indoor Positioning Systems for Wireless Personal Networks," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 13–32, 2009.

[35] J. Torres-Solis, T. H. Falk, and T. Chau, "A Review of Indoor Localization Technologies: Towards Navigational Assistance for Topographical Disorientation," *Ambient Intelligence*, pp. 51–84, 2010.

[36] D. Dardari, P. Closas, and P. Djuric, "Indoor Tracking: Theory, Methods, and Technologies," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1263–1278, Apr. 2015.

[37] Federal Communications Commission, *Wireless E911 Location Accuracy Requirements : Fourth Report and Order*. PS Docket No. 07-114, Feb. 3 2015.

[38] M. P. Wylie and J. Holtzman, "The Non-Line of Sight Problem in Mobile Location Estimation," in *5th IEEE Int. Conf. Universal Personal Commun.*, vol. 2, 1996, pp. 827–831.

[39] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF based user location and tracking system," in *Proc. IEEE INFOCOM 2000*, vol. 2, 2000, pp. 775–784.

[40] M. A. A. A. Y. A. Rehim, "HORUS: A WLAN-based Indoor Location Determination System," Ph.D. dissertation, University of Maryland, 2004.

[41] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Trans. Syst. Man, Cybern.*, vol. 37, no. 6, pp. 1067–1080, 2007.

[42] H. Lee, M. Wicke, B. Kusy, and L. Guibas, "Localization of mobile users using trajectory matching," in *Proc. first ACM Int. workshop on Mobile entity localization and tracking in GPS-less environments*. ACM, 2008, pp. 123–128.

[43] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization," *IEEE Trans. Mobile Computing*, vol. 14, no. 2, pp. 444–457, 2015.

[44] L. Ma, Y. Fan, Y. Xu, and Y. Cui, "Pedestrian dead reckoning trajectory matching method for radio map crowdsourcing building in WiFi indoor positioning system," in *2017 IEEE Int. Conf. on Commun. (ICC)*. IEEE, May 2017, pp. 1–6.

[45] W. Kang and Y. Han, "SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization," *IEEE Sensors Journal*, vol. 15, no. 5, pp. 2906–2916, 2015.

[46] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec 2016.

[47] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part I," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[48] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-Based SLAM," *IEEE Intell. Transportation Syst. Mag.*, vol. 2, no. 4, pp. 31–43, winter 2010.

[49] S. Saeedi, M. Trentini, M. Seto, H. Li, and M. Seto, "Multiple-robot Simultaneous Localization and Mapping - A Review," *Journal of Field Robotics*, 2016.

[50] M. Zhang, L. Pei, and X. Deng, "GraphSLAM-based Crowdsourcing framework for indoor Wi-Fi fingerprinting," in *2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS)*, Nov 2016, pp. 61–67.

[51] R. Faragher and R. Harle, "Location Fingerprinting With Bluetooth Low Energy Beacons," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2418–2428, 2015.

[52] A. Kleiner, C. Dornhege, and S. Dali, "Mapping disaster areas jointly: RFID-coordinated SLAM by humans and robots," *SSRR2007 - IEEE International Workshop on Safety, Security and Rescue Robotics Proceedings*, pp. 1–6, 2007.

[53] M. Hardegger, D. Roggen, S. Mazilu, and G. Troster, "ActionSLAM: Using location-related actions as landmarks in pedestrian SLAM," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Nov 2012, pp. 1–10.

[54] M. Abdelbar, B. Tranter, and T. Bose, "Cooperative cumulants-based modulation classification in distributed networks," *IEEE Trans. Cogn. Commun. Netw., accepted for publication*, 2018.

[55] ——, "Cooperative cumulants-based modulation classification under flat rayleigh fading channels," in *2015 IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 7622–7627.

[56] ——, "Cooperative combining of cumulants-based modulation classification in cr networks," in *2014 IEEE Military Commun. Conf. (MILCOM)*, Oct 2014, pp. 434–439.

[57] ——, "Cooperative modulation classification of multiple signals in cognitive radio networks," in *2014 IEEE Int. Conf. on Commun. (ICC)*, Jun. 2014, pp. 1483–1488.

[58] M. Abdelbar and R. M. Buehrer, "An Improved Technique for Indoor Pedestrian GraphSLAM using Smartphones," *IEEE Sensors J., submitted for publication*, 2018.

[59] ——, "Collaborative Indoor Pedestrian GraphSLAM using Smartphone Sensors and BLE," *IEEE Trans. Mobile Comput., submitted for publication*, 2018.

[60] ——, "Pedestrian GraphSLAM using Smartphone-based PDR in Indoor Environments," in *IEEE ICC - Workshop on Advances in Network Localization and Navigation (ANLN), to be published*, Kansas City, MO, May 2018.

[61] ——, "Indoor localization through trajectory tracking using neural networks," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Oct. 2017, pp. 519–524.

[62] ——, "Improving Cellular Positioning Indoors through Trajectory Matching," in *IEEE/ION Position, Location and Navigation Symp. (PLANS)*, Apr. 2016, pp. 219–224.

[63] F. Hameed, O. Dobre, and D. Popescu, "On the likelihood-based approach to modulation classification," *IEEE Trans. Wireless Commun.*, vol. 8, no. 12, pp. 5884–5892, Dec. 2009.

[64] J. L. Xu, W. Su, and M. Zhou, "Likelihood-Ratio Approaches to Automatic Modulation Classification," *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 41, no. 4, pp. 455–469, 2011.

[65] A. Swami and B. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Trans. Commun.*, vol. 48, no. 3, pp. 416–429, Mar. 2000.

[66] H.-C. Wu, M. Saquib, and Z. Yun, "Novel automatic modulation classification using cumulant features for communications via multipath channels," *IEEE Trans. Wireless Commun.*, vol. 7, no. 8, pp. 3098–3105, 2008.

[67] V. Orlic and M. Dukic, "Automatic modulation classification algorithm using higher-order cumulants under real-world channel conditions," *IEEE Commun. Lett.*, vol. 13, no. 12, pp. 917–919, 2009.

[68] F. Wang and X. Wang, "Fast and robust modulation classification via Kolmogorov-Smirnov test," *IEEE Trans. Commun.*, vol. 58, no. 8, pp. 2324–2332, 2010.

[69] M. W. Aslam, Z. Zhu, and A. K. Nandi, "Automatic modulation classification using combination of genetic programming and KNN," *IEEE Trans. Wireless Commun.*, vol. 11, no. 8, pp. 2742 –2750, Aug. 2012.

[70] A. Gorcin and H. Arslan, "Identification of OFDM signals under multipath fading channels," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Oct. 2012, pp. 1–7.

[71] K. Hassan, I. Dayoub, W. Hamouda, C. N. Nzeza, and M. Berbineau, "Blind digital modulation identification for spatially-correlated MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 683–693, 2012.

[72] M. Muhlhaus, M. Oner, O. Dobre, and F. Jondral, "A low complexity modulation classification algorithm for MIMO systems," *IEEE Commun. Lett.*, vol. 17, no. 10, pp. 1881–1884, Oct. 2013.

[73] Y. Eldemerdash, M. Marey, O. Dobre, G. Karagiannidis, and R. Inkol, "Fourth-order statistics for blind classification of spatial multiplexing and Alamouti space-time block code signals," *IEEE Trans. Commun.*, vol. 61, no. 6, pp. 2420–2431, Jun. 2013.

[74] S. Kharbech, I. Dayoub, M. Zwingelstein-Colin, E. P. Simon, and K. Hassan, "Blind digital modulation identification for time-selective MIMO channels," *IEEE Wireless Commun. Lett.*, vol. 3, no. 4, pp. 373–376, 2014.

[75] A. Fehske, J. Gaeddert, and J. Reed, "A new approach to signal classification using spectral correlation and neural networks," in *IEEE Int. Symp. Dyn. Spectrum Access Netw. (DySPAN)*, Nov. 2005, pp. 144 –150.

[76] K. Kim, I. Akbar, K. Bae, J.-S. Urn, C. Spooner, and J. Reed, "Cyclostationary approaches to signal detection and classification in cognitive radio," in *IEEE Int. Symp. Dyn. Spectrum Access Netw. (DySPAN)*, Apr. 2007, pp. 212 –215.

[77] M. Öner and F. Jondral, "Air interface identification for software radio systems," *AEU Int. J. Electron. and Commun.*, vol. 61, no. 2, pp. 104–117, 2007.

[78] P. Sutton, K. Nolan, and L. Doyle, "Cyclostationary signatures in practical cognitive radio applications," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 13–24, Jan. 2008.

[79] B. Ramkumar, "Automatic modulation classification for cognitive radios using cyclic feature detection," *IEEE Circuits Syst. Mag.*, vol. 9, no. 2, pp. 27 –45, quarter 2009.

[80] A. Punchihewa, O. A. Dobre, S. Rajan, and R. Inkol, "Cyclostationarity-based algorithm for blind recognition of OFDM and single carrier linear digital modulations," in *IEEE Int. Symp. Personal, Indoor and Mobile Radio Commun. (PIMRC)*, 2007, pp. 1–5.

[81] A. Al-Habashna, O. Dobre, R. Venkatesan, and D. Popescu, "Second-order cyclostationarity of mobile WiMAX and LTE OFDM signals and application to spectrum awareness in cognitive radio systems," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 1, pp. 26–42, Feb. 2012.

[82] Q. Zhang, O. Dobre, Y. Eldemerdash, S. Rajan, and R. Inkol, "Second-order cyclostationarity of BT-SCLD signals: Theoretical developments and applications to signal classification and blind parameter estimation," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1501–1511, Apr. 2013.

[83] M. Shi, Y. Bar-Ness, and W. Su, "STC and BLAST MIMO modulation recognition," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov. 2007, pp. 3034–3039.

[84] M. Marey, O. Dobre, and R. Inkol, "Cyclostationarity-based blind classification of STBCs for cognitive radio systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2012, pp. 1715–1720.

[85] E. Karami and O. Dobre, "Identification of SM-OFDM and AL-OFDM signals based on their second-order cyclostationarity," *IEEE Trans. Veh. Technol.*, vol. 64, no. 3, pp. 942–953, Mar. 2015.

[86] M. Marey and O. A. Dobre, "Blind modulation classification algorithm for single and multiple-antenna systems over frequency-selective channels," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1098–1102, Sept. 2014.

[87] Y. Eldemerdash, O. Dobre, and B. Liao, "Blind identification of SM and Alamouti STBC-OFDM signals," *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 972–982, Feb. 2015.

[88] M. Marey, O. Dobre, and R. Inkol, "Blind STBC identification for multiple-antenna OFDM systems," *IEEE Trans. Commun.*, vol. 62, no. 5, pp. 1554–1567, May 2014.

[89] M. Marey, O. Dobre, and B. Liao, "Classification of STBC systems over frequency-selective channels," *IEEE Trans. Veh. Technol.*, vol. PP, no. 99, pp. 1–1, 2014.

[90] D. R. Brillinger, *Time series: data analysis and theory*. Society for Industrial and Applied Mathematics, 2001, vol. 36.

[91] J. Mendel, "Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications," *Proceedings of the IEEE*, vol. 79, no. 3, pp. 278–305, 1991.

[92] R. J. Serfling, *Approximation theorems of mathematical statistics*. John Wiley & Sons, 2009, vol. 162.

[93] M. Muhlhaus, M. Oner, O. A. Dobre, H. Jakel, and F. K. Jondral, "A novel algorithm for MIMO signal classification using higher-order cumulants," in *IEEE Radio and Wireless Symp. (RWS)*, 2013, pp. 7–9.

[94] J. K. Tugnait, "Blind equalization and estimation of digital communication FIR channels using cumulant matching," *IEEE Trans. Commun.*, vol. 43, no. 234, pp. 1240–1245, 1995.

[95] B. Ramkumar, T. Bose, and M. Radenkovic, "Robust multiuser automatic modulation classifier for multipath fading channels," in *IEEE Int. Symp. Dyn. Spectrum Access Netw. (DySPAN)*, 2010, pp. 1–10.

[96] S. Amuru and C. da Silva, "Cumulant-based channel estimation algorithm for modulation classification in frequency-selective fading channels," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, 2012.

[97] W. Su and J. Kosinski, "Framework of network centric signal sensing for automatic modulation classification," in *Int. Conf. Netw. Sensing and Control (ICNSC)*, Apr. 2010, pp. 534–539.

[98] A. Abdelgawad and M. Bayoumi, *Resource-Aware data fusion algorithms for wireless sensor networks*. Springer, 2012.

[99] J. L. Xu, W. Su, and M. Zhou, "Asynchronous and high-accuracy digital modulated signal detection by sensor networks," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, 2011, pp. 589–594.

[100] O. Ozdemir, R. Li, and P. K. Varshney, "Hybrid maximum likelihood modulation classification using multiple radios," *IEEE Commun. Lett.*, vol. 17, no. 10, pp. 1889–1892, 2013.

[101] T. Wimalajeewa, J. Jagannath, P. K. Varshney, A. Drozd, and W. Su, "Distributed Asynchronous Modulation Classification Based on Hybrid Maximum Likelihood Approach," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, 2015.

[102] H. He, S. Choi, P. K. Varshney, and W. Su, "Distributed Classification under Statistical Dependence with Application to Automatic Modulation Classification," in *2015 18th Int. Conf. Inform. Fusion (Fusion)*, 2015, pp. 1597–1602.

[103] O. Ozdemir, L. N. Theagarajan, M. Agarwal, T. Wimalajeewa, and P. K. Varshney, "An MCMC Approach to Multisensor Linear Modulation Classification," in *Proc. IEEE Wirless Commun. Netw. Conf. (WCNC)*, 2017.

[104] S. Hakimi and G. Abed Hodtani, "Optimized Distributed Automatic Modulation Classification in Wireless Sensor Networks Using Information Theoretic Measures," *IEEE Sensors J.*, vol. 17, no. 10, pp. 3079–3091, 2017.

[105] W. Headley, J. Reed, and C. da Silva, "Distributed cyclic spectrum feature-based modulation classification," in *Proc. IEEE Wirless Commun. Netw. Conf. (WCNC)*, Mar. 2008, pp. 1200–1204.

[106] P. Forero, A. Cano, and G. Giannakis, "Distributed feature-based modulation classification using wireless sensor networks," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, 2008, pp. 1–7.

[107] M. Sohul, B. Ramkumar, and T. Bose, "Multiuser automatic modulation classification for cognitive radios using distributed sensing in multipath fading channels," in *7th EAI Int. Conf. Cognitive Radio Oriented Wireless Netw. and Commun. (CROWNCOM)*, Jun. 2012, pp. 71–76.

[108] W. Su, "Signal modulation classification device using distributed sensors," Oct. 23 2012, US Patent 8,295,409.

[109] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[110] A. Leick, L. Rapoport, and D. Tatarnikov, *GPS Satellite Surveying*. John Wiley & Sons, Inc, 2015.

[111] J. Salo, P. Vainikainen, and H. M. El-Sallabi, "The Distribution of the Product of Independent Rayleigh Random Variables," *IEEE Trans. Antennas Propag.*, vol. 54, no. 2, pp. 639–643, 2006.

[112] I. Kovacs, P. Eggers, K. Olesen, and L. Petersen, "Investigations of outdoor-to-indoor mobile-to-mobile radio communication channels," in *Proc. IEEE 56th Veh. Technol. Conf.*, 2002, pp. 430–434.

[113] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," *IEEE Robotics Automation Magazine*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

[114] M. Angermann and P. Robertson, "FootSLAM: Pedestrian Simultaneous Localization and Mapping Without Exteroceptive Sensors - Hitchhiking on Human Perception and Cognition," *Proc. IEEE*, vol. 100, no. Special Centennial Issue, pp. 1840–1848, May 2012.

[115] C. Fischer and H. Gellersen, "Location and Navigation Support for Emergency Responders: A Survey," *IEEE Pervasive Computing*, vol. 9, no. 1, pp. 38–47, Jan 2010.

[116] R. Harle, "A Survey of Indoor Inertial Positioning Systems for Pedestrians," *IEEE Commun. Surveys & Tutorials*, vol. 15, no. 3, pp. 1281–1293, Third 2013.

[117] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[118] A. Kleiner and D. Sun, "Decentralized SLAM for pedestrians without direct communication," in *IEEE International Conference on Intelligent Robots and Systems*, 2007, pp. 1461–1466.

[119] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, "Efficient, generalized indoor WiFi GraphSLAM," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 1038–1043.

[120] H. Shin, Y. Chon, and H. Cha, "Unsupervised construction of an indoor floor plan using a smartphone," *IEEE Trans. Syst. Man, Cybern. Part C: Applications and Reviews*, vol. 42, no. 6, pp. 889–898, 2012.

[121] R. M. Faragher, C. Sarno, and M. Newman, "Opportunistic radio SLAM for indoor navigation using smartphone sensors," in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*. IEEE, apr 2012, pp. 120–128.

[122] R. Faragher and R. Harle, "SmartSLAM - an efficient smartphone indoor positioning system exploiting machine learning and opportunistic sensing," in *Proc. 26th Int. Tech. Meet. Satel. Div. Institute of Nav. (ION GNSS+ '13)*, 2013, pp. 1–14.

[123] A. Galov and A. Moschevikin, "Simultaneous localization and mapping in indoor positioning systems based on round trip time-of-flight measurements and inertial navigation," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, Oct. 2014, pp. 457–464.

[124] P. Mirowski, T. K. Ho, S. Yi, and M. MacDonald, "SignalSLAM: Simultaneous localization and mapping with mixed WiFi, Bluetooth, LTE and magnetic signals," in *Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Oct 2013, pp. 1–10.

[125] S. Yi, P. Mirowski, T. K. Ho, and V. Pavlovic, "Pose invariant activity classification for multi-floor indoor localization," in *Proceedings - International Conference on Pattern Recognition*, 2014, pp. 3505–3510.

[126] C. Gao and R. Harle, "Sequence-based magnetic loop closures for automated signal surveying," in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, Oct. 2015, pp. 1–12.

[127] ——, "MSGD: Scalable back-end for indoor magnetic field-based GraphSLAM," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2017, pp. 3855–3862.

[128] U. Steinhoff and B. Schiele, "Dead reckoning from the pocket - An experimental study," *IEEE Int. Conf. Pervasive Comput. Communi. (PerCom)*, pp. 162–170, 2010.

[129] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proc. ACM Int. Joint Conf. Pervasive and ubiquitous Comput. - UbiComp '13*, 2013, pp. 225–234.

[130] Q. Tian, Z. Salcic, K. I. K. Wang, and Y. Pan, "A Multi-Mode Dead Reckoning System for Pedestrian Tracking Using Smartphones," *IEEE Sensors Journal*, vol. 16, no. 7, pp. 2079–2093, 2016.

[131] O. Mezentsev, G. Lachapelle, and J. Collin, "Pedestrian dead reckoning - A solution to navigation in GPS signal degraded areas?" *Geomatica*, vol. 59, no. 2, pp. 175–182, 2005.

[132] S. Beauregard and H. Haas, "Pedestrian dead reckoning: A basis for personal positioning," *Positioning, Navigation and Communication*, pp. 27–35, 2006.

[133] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Comput. Graph. Appl.*, vol. 25, no. 6, pp. 38–46, 2005.

[134] R. W. Levei and T. Judd, "Dead reckoning navigational system using accelerometer to measure foot impacts," Patent US 5 583 776, Dec. 10, 1996.

[135] H. Weinberg, "Using the ADXL202 in pedometer and personal navigation applications," *Analog devices*, pp. 1–8, 2002. [Online]. Available: http://www.analog.com/media/en/technical-documentation/application-notes/513772624AN602.pdf

[136] J. Scarlett, "Enhancing the Performance of Pedometers Using a Single Accelerometer," *Analog Device*, pp. 1–16, 2007. [Online]. Available: http://www.analog.com/media/en/technical-documentation/application-notes/47076299220991AN_900.pdf

[137] N. H. Ho, P. H. Truong, and G. M. Jeong, "Step-detection and adaptive step-length estimation for pedestrian dead-reckoning at variouswalking speeds using a smartphone," *Sensors (Switzerland)*, vol. 16, no. 9, 2016.

[138] G. W. Corder and D. I. Foreman, *Testing Data for Normality*. John Wiley & Sons, Inc., 2009, pp. 12–37.

[139] A. Correa, M. Barcelo, A. Morell, and J. L. Vicario, "A Review of Pedestrian Indoor Positioning Systems for Mass Market Applications," *Sensors*, vol. 17, no. 8, 2017.

[140] M. H. Afzal, V. Renaudin, and G. Lachapelle, "Assessment of Indoor Magnetic Field Anomalies using Multiple Magnetometers," in *ION GNSS*, Sep. 2010.

[141] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-Effort Crowdsourcing for Indoor Localization," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Network. - Mobicom '12*, 2012.

[142] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[143] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[144] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, 1997.

[145] A. Oliveira and A. Seijas-Macias, "An Approach to Distribution of the Product of Two Normal Variables," *Discussiones Mathematicae Probability and Statistics*, vol. 32, no. 1-2, p. 87, 2012.

[146] S. Schoenecker and T. Luginbuhl, "Characteristic Functions of the Product of Two Gaussian Random Variables and the Product of a Gaussian and a Gamma Random Variable," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 644–647, 2016.

[147] O. C. Ibe, "Functions of Random Variables," *Fundamentals of Applied Probability and Random Processes*, pp. 185–223, 2014.

[148] P. Broder. (2017) Sensor play. [Online]. Available: https://itunes.apple.com/app/id921385514

[149] D. Berndt and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series," in *Workshop on Knowledge Discovery in Databases*, vol. 398, 1994, pp. 359–370.

[150] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering Similar Multidimensional Trajectories," in *Proc. 18th Int. Conf. Data Engineering*, Feb. 2002, pp. 673–684.

[151] W. H. Huang and K. R. Beevers, "Topological map merging," *Int. Journal of Robotics Research*, vol. 24, no. 8, pp. 601–613, 2005.

[152] R. Wang, M. Veloso, and S. Seshan, "Iterative snapping of odometry trajectories for path identification," *Lecture Notes in Computer Science*, vol. 8371 LNAI, pp. 432–443, 2014.

[153] ——, "O-Snap: Optimal snapping of odometry Trajectories for route identification," in *Proc. IEEE Int. Conf. Robotics. Automation*, 2014, pp. 5824–5829.

[154] R. Wang, R. Shroff, Y. Zha, S. Seshan, and M. Veloso, "Indoor trajectory identification: Snapping with uncertainty," in *IEEE Int. Conf. Intell. Robots Syst.*, 2015, pp. 4901–4906.

[155] P. McDermott-Wells, "What is Bluetooth?" *Potentials, IEEE*, vol. 23, no. 5, pp. 33–35, 2005.

[156] "Bluetooth Special Interest Group (SIG)." [Online]. Available: https://www.bluetooth.org/

[157] R. Heydon, *Bluetooth low energy: the developer's handbook*. Prentice Hall Upper Saddle River, 2013, vol. 1.

[158] S. Feldmann, K. Kyamakya, A. Zapater, and Z. Lue, "An indoor Bluetooth-based positioning system: concept , Implementation and experimental evaluation," in *Int. Conf. Wireless Network.*, Jan. 2003, pp. 109–113.

[159] U. Bandara, M. Hasegawa, M. Inoue, H. Morikawa, and T. Aoyama, "Design and implementation of a Bluetooth signal strength based location sensing system," in *IEEE Radio Wireless Conf. (IEEE Cat. No.04TH8746)*, Oct. 2004, pp. 3–4.

[160] S. Zhou and J. K. Pollard, "Position measurement using Bluetooth," *IEEE Trans. Consumer Electronics*, vol. 52, no. 2, pp. 555–558, 2006.

[161] A. K. M. M. Hossain and W. S. Soh, "A comprehensive study of bluetooth signal parameters for localization," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, 2007.

[162] Y. Wang, X. Yang, Y. Zhao, Y. Liu, and L. Cuthbert, "Bluetooth positioning using RSSI and triangulation methods," in *IEEE 10th Consumer Commun. Network. Conf. (CCNC)*, 2013, pp. 837–842.

[163] Z. Jianyong, L. Haiyong, C. Zili, and L. Zhaohui, "RSSI based Bluetooth low energy indoor positioning," in *IPIN 2014 - 2014 International Conference on Indoor Positioning and Indoor Navigation*, Oct. 2014, pp. 526–533.

[164] R. S. Campos, "Positioning in Bluetooth and UWB Networks," *Cadernos do IME-Série Informática*, vol. 38, pp. 35–54, 2017.

[165] "Apple Developer website - Bluetooth for developers." [Online]. Available: https://developer.apple.com/bluetooth/

[166] A. Altman and J. Gondzio, "Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization," *Optimization Methods and Software*, vol. 11, no. 1, pp. 275–302, 1999.

[167] N. Amiot, M. Laaraiedh, and B. Uguen, "Improved Mobility Modeling for Indoor Localization Applications," *2014 11th Workshop Positioning, Navigation and Commun. (WPNC)*, pp. 1–6, Mar 2014.

[168] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*. Addison Wesley, 2007.

[169] C. W. Reynolds, "Steering Behaviors for Autonomous Characters," in *Game Developers Conference*, 1999, pp. 763–782.

[170] N. Amiot, M. Laaraiedh, and B. Uguen, "Pylayers: An Open Source Dynamic Simulator for Indoor Propagation and Localization," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2013, pp. 84–88.

[171] B. J. Mohler, W. B. Thompson, S. H. Creem-Regehr, H. L. Pick, and W. H. Warren, "Visual Flow Influences Gait Transition Speed and Preferred Walking Speed," *Experimental Brain Research*, vol. 181, no. 2, pp. 221–228, 2007.

[172] S. H. Fang and T. N. Lin, "Indoor Location System Based on Discriminant-Adaptive Neural Network in IEEE 802.11 Environments," *IEEE Trans. Neural Netw.*, vol. 19, no. 11, pp. 1973–1978, Nov. 2008.

[173] C. Laoudias, D. G. Eliades, P. Kemppi, C. G. Panayiotou, and M. M. Polycarpou, "Indoor Localization using Neural Networks with Location Fingerprints," in *Int. Conf. Artificial Neural Netw.* Springer, 2009, pp. 954–963.

[174] M. Altini, D. Brunelli, E. Farella, and L. Benini, "Bluetooth Indoor Localization with Multiple Neural Networks," in *5th IEEE Int. Symp. on Wireless Pervasive Computing (ISWPC)*, 2010, pp. 295–300.

[175] C. Xiao and N. Yu, "A Predictive Localization Algorithm based on RBF Neural Network for Wireless Sensor Networks," in *Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2015, pp. 1–5.

[176] J. Schloemann, V. V. N. S. Malladi, A. G. Woolard, J. M. Hamilton, R. M. Buehrer, and P. A. Tarazaga, "Vibration Event Localization in an Instrumented Building," in *Experimental Tech., Rotating Mach. and Acoust.*, vol. 8. Springer, 2015, pp. 265–271.

[177] D. Zeinalipour-Yazti, C. Laoudias, C. Costa, M. Vlachos, M. I. Andreou, and D. Gunopulos, "Crowdsourced Trace Similarity with Smartphones," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1240–1253, 2013.

[178] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

[179] S. Lawrence, I. Burns, A. Back, A. C. Tsoi, and L. Giles, "Neural Network Classification and Prior Class Probabilities," *Network*, pp. 295–309, 1998.

[180] H. Wang, H. Su, K. Zheng, S. Sadiq, and X. Zhou, "An Effectiveness Study on Trajectory Similarity Measures," in *Proc. 24th Australasian Database Conf.*, vol. 137, Feb. 2013, pp. 13–22.

[181] Z. Zhang, K. Huang, and T. Tan, "Comparison of Similarity Measures for Trajectory Clustering in Outdoor Surveillance Scenes," in *18th Int. Conf. Pattern Recognition (ICPR'06)*. IEEE, 2006, pp. 1135–1138.

[182] L. Chen, M. T. Özsu, and V. Oria, "Robust and Fast Similarity Search for Moving Object Trajectories," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*. ACM, 2005, pp. 491–502.

[183] M. O'Searcoid, *Metric spaces*. Springer Science & Business Media, 2006.

[184] L. Chen and R. Ng, "On the Marriage of Lp-norms and Edit Distance," in *Proc. 30th Int. Conf. Very large Data Bases*, vol. 30, 2004, pp. 792–803.

[185] E. Ristad and P. Yianilos, "Learning String-Edit Distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 5, pp. 522–532, May 1998.