Multi-Hop Localization in Cluttered Environments

Muzammil Hussain

Wolfson College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity term 2013

To my late father, who could not live to see this day.

Acknowledgements

First and foremost, I am deeply grateful to my supervisor Niki Trigoni. Niki provided invaluable guidance over the past four years, making me mature as a researcher. Preparing for her meetings honed my critical thinking skills, presentations skills and eye for detail as well as abstract. She gave me immense support when I had to deal with the death of my new born son in 2011. I couldn't have asked for a better supervisor.

I am grateful to a large number of people for helping me through my years as a doctoral student at Oxford. I would like to thank Yusuf Aytar, who was a great friend, teacher and collaborator. I would like to acknowledge, among others, Andrew Markham, Andrew Symington, Paul Ward, Ricklef Wohlers, Sarfraz Nawaz, Zhuoling Xiao, Hongkai Wen, Mikael Wallman, Omer Gunes, Cornelius Namiluko, Tulio de Souza and Vijay D'Silva for their feedback. I would like to thank Eric Sommerlade, Georg Weissenbacher, William Stockland, Shamal Faily, Nassim Saghir, Said Alsarmi, Vehdat Gormez and Aadya Shukla for their encouragement and advice. I would like to thank the staff in the Engineering Science workshop and the Department of Computer Science, in particular Joe Atherton, Peter Turner, Julie Sheppard and Terry Brown, among others. I would like to acknowledge that work in this thesis was funded by the EPSRC project EP/F064209/1 entitled "Acoustic Actuated Sensor Networks for Industrial Processes (AASN4IP)".

I would like to thank my family for their prayers and encouragement throughout my studies at Oxford. In particular, I would like to acknowledge my mother, my brother Mudassir, and my uncle, Azeez Uncle. I fondly remember my late father and late uncle, Farooq Chacha, who would have been thrilled at this achievement and were a significant source of motivation for my pursuing graduate studies when they were alive.

A thesis can be as trying for a student's partner as for the student himself, and my case was no exception! I can't even begin to express my gratitude to my wonderful wife Faiza, who was endlessly patient, supportive and accommodating, even when my many delays affected her own life plans. Her complete faith in my capabilities and her complete devotion for my happiness play a big role in this milestone.

Abstract

Range-based localization is a widely used technique for position estimation where distances are measured to anchors, nodes with known positions, and the position is analytically estimated. It offers the benefits of providing high localization accuracy and involving simple operation over multiple deployments. Examples are the Global Positioning System (GPS) and network-based cellular handset localization. Range-based localization is promising for a range of applications, such as robot deployment in emergency scenarios or monitoring industrial processes. However, the presence of clutter in some of these environments leads to a severe degradation of the localization accuracy due to non-line-of-sight (NLOS) signal propagation. Moreover, current literature in NLOS-mitigation techniques requires that the NLOS distances constitute only a minority of the total number of distances to anchors.

The key ideas proposed in the dissertation are: 1) multi-hop localization offers significant advantages over single-hop localization in NLOS-prone environments; and 2) it is possible to further reduce position errors by carefully placing intermediate nodes among the clutter to minimize multi-hop distances between the anchors and the unlocalized node. We demonstrate that shortest path distance (SPD) based multi-hop localization algorithms, namely DV-Distance and MDS-MAP, perform the best among other competing techniques in NLOS-prone settings. However, with random node placement, these algorithms require large node densities to produce high localization accuracy.

To tackle this, we show that the strategic placement of a relatively small number of nodes in the clutter can offer significant benefits. We propose two algorithms for node placement: first, the Optimal Placement for DV-Distance (OPDV) focuses on obtaining the optimal positions of the nodes for a known clutter topology; and second, the Adaptive Placement for DV-Distance (APDV) offers a distributed control technique that carefully moves nodes in the monitored area to achieve localization accuracies close to those achieved by OPDV. We evaluate both algorithms via extensive simulations, as well as demonstrate the APDV algorithm on a real robotic hardware platform.

We finally demonstrate how the characteristics of the clutter topology influence single-hop and multi-hop distance errors, which in turn, impact the performance of the proposed algorithms.

Contents

1 Introduction **1**

1.1 Research Problem 2
1.2 Motivating Applications 5
1.3 Challenges 8
1.4 Proposed Approach 10
1.5 Contributions 11
1.6 Publications 11
1.7 Thesis Structure 13

2 Background **15**

2.1 Localization 15
2.1.1 Classification of Localization Techniques 18
2.1.2 Ranging Techniques 21
2.1.2.1 Signal Strength 21
2.1.2.2 Time of Flight 22
2.1.2.3 Radio Inferiometry 24
2.1.2.4 Angle of Arrival 24
2.1.3 Multi-Hop Localization 24
2.1.3.1 Iterative Localization 25
2.1.3.2 DV-Distance 27
2.1.3.3 DV-HOP 28
2.1.3.4 MDS-MAP 29
2.2 Range-based Localization in Clutter-Prone Environments 30
2.2.1 NLOS Detection/Elimination Techniques 31
2.2.2 NLOS Mitigation Techniques 32
2.2.3 Mitigation of Overestimate Error of Multi-Hop Distances 34
2.2.4 NLOS Characterization 35
2.3 SLAM-based Localization in Clutter-Prone Environments 36

2.4 Discussion 40

3 Multi-Hop Localization for NLOS-Prone Settings 41
3.1 Motivation 41
3.2 Methodology 45
3.2.1 Simulation Environment 45
3.2.2 Simulation Parameters 47
3.2.3 Algorithms 49
3.2.4 Performance Metrics 50
3.3 Evaluation 52
3.3.1 Effect of Communication Range 52
3.3.2 Effect of Number of Localizers 56
3.3.3 Effect of Anchor Placements and Clutter Topology . . . 58
3.4 Discussion 60

4 Localizer Placement in Cluttered Environments 63
4.1 Motivation 63
4.2 Optimal Placement for DV-Distance (OPDV) 65
4.2.1 Basic Algorithm 66
4.2.2 Constrained-localizer Algorithm 69
4.2.3 Optimality and Complexity Analysis 71
4.3 Adaptive Placement for DV-Distance (APDV) 72
4.3.1 Motivation 72
4.3.2 Algorithm Description 73
4.3.2.1 APDV Parameters 75
4.3.2.2 Alignment Variants 76
4.3.3 Termination Proof 79
4.4 Discussion 82

5 Simulation-Based Evaluation 83
5.1 Methodology 83
5.1.1 Simulation Setup 83
5.1.2 Simulation Parameters 84
5.1.2.1 OPDV Parameters 84
5.1.2.2 APDV Parameters 85
5.1.2.3 Parameters for Competing Techniques 87
5.1.3 Performance Metrics 87

5.1.4 Algorithms . . . 88
5.2 Sensitivity Analysis of OPDV . . . 89
5.2.1 Effect of OPDV Parameters . . . 89
5.2.1.1 Grid Granularity gs_{opdv} . . . 89
5.2.1.2 Communication range $\mathcal{R}_c$ . . . 91
5.2.1.3 Localizer limit N_{loc} . . . 92
5.2.2 Influence of Clutter Topology . . . 93
5.2.3 Comparison with Random Placement . . . 95
5.3 Sensitivity Analysis of APDV . . . 96
5.3.1 APDV Alignment Variants . . . 96
5.3.2 Effect of APDV parameters . . . 100
5.3.2.1 Step-Size Threshold $\mathcal{S}_T$ . . . 100
5.3.2.2 Number of Probing Directions N_{dir} . . . 103
5.3.2.3 Number of Localizers N_l . . . 104
5.3.3 Effect of Environment/Hardware Features . . . 105
5.3.3.1 Line-Of-Sight (LOS) Distance Error . . . 106
5.3.3.2 Robot Odometry Error . . . 109
5.3.3.3 Clutter Topology . . . 113
5.3.4 Comparison with Competing Techniques . . . 114
5.4 Discussion . . . 120

6 Robot-Based Evaluation 123
6.1 APDV Implementation . . . 123
6.1.1 Hardware Architecture . . . 124
6.1.1.1 Roomba Robot . . . 124
6.1.1.2 MIT Cricket Motes . . . 125
6.1.2 Software Architecture . . . 130
6.2 Evaluation . . . 132
6.2.1 Experimental Setup . . . 132
6.2.2 Results . . . 133
6.3 Discussion . . . 135

7 Influence of Clutter Topology on Distance Error 137
7.1 Motivation . . . 137
7.2 Characteristic Features of Clutter Topology . . . 141
7.2.1 Clutter Area Fraction . . . 141
7.2.2 Clutter Spacing Distribution . . . 141

7.2.3 Occupancy Grid . . . 142
7.2.4 Fourier Transformation . . . 144
7.2.5 GIST Characterization . . . 144
7.3 Effect of Clutter Topology on NLOS Bias . . . 146
7.3.1 Experimental Setup . . . 146
7.3.2 Estimation of NLOS Incidence Probability . . . 148
7.3.2.1 Support Vector Regressor . . . 149
7.3.2.2 Results . . . 150
7.3.3 Estimation of NLOS Bias Distribution . . . 152
7.3.3.1 k-Nearest-Neighbour Estimation . . . 153
7.3.3.2 Results . . . 154
7.3.3.3 Lognormal Fitting for NLOS Bias Distribution 157
7.4 Effect of Clutter Topology on Multi-Hop Error . . . 161
7.4.1 Experimental Setup . . . 161
7.4.2 Estimation of OPDV Multi-Hop Error . . . 162
7.5 Discussion . . . 166

8 Conclusion and Future Work 167
8.1 Summary of Contributions . . . 167
8.2 Limitations . . . 168
8.3 Directions for Future Work . . . 170

A OPDV Pseudocode 173

Bibliography 176

List of Figures

1.1 LOS and NLOS distances 4
1.2 Nuclear waste pond monitoring application 5
1.3 Fukushima Daiichi nuclear disaster recovery application 7

2.1 Anchor-based localization. 16
2.2 Classification of localization techniques. 18
2.3 Iterative localization in action. 25
2.4 DV-Distance in action. 27
2.5 DV-HOP in action. 28

3.1 Relationship between NLOS probability and actual distance. . . 42
3.2 Motivation for using localizers in cluttered environments. 44
3.3 2D raytracer simulator. 46
3.4 Clutter topologies used for multi-hop localization evaluation. . . 49
3.5 Effect of communication range. 53
3.6 Anchors and packet traffic for varying communication ranges. . 54
3.7 Empirical distributions of DV-Distance localization errors. . . . 56
3.8 Effect of number of localizers. 57
3.9 Anchor placement configurations. 58
3.10 Effect of anchor placement. 59
3.11 Effect of clutter topology. 59

4.1 Localizer placement for DV-Distance in cluttered environments. 64
4.2 Generation of input graph $\mathcal{G}$. 67
4.3 Sample output for the basic version of OPDV. 69
4.4 Sample output for the constrained-localizer version of OPDV. . 71
4.5 Aim of APDV. 74
4.6 APDV in action. 74
4.7 Step size threshold in APDV. 76

4.8 Single-step alignment variants. 77
4.9 Multi-step alignment variants. 78
4.10 Multi-hop distance formation between anchor and sensor robot. 79
4.11 Initial positions of localizers for alignment procedure. 80

5.1 Clutter topologies used for OPDV sensitivity analysis. 84
5.2 Clutter topologies used for APDV sensitivity analysis. 85
5.3 Sensor robot trajectories for APDV sensitivity analysis. 87
5.4 Influence of grid spacing of OPDV. 90
5.5 Influence of communication range on OPDV. 91
5.6 Influence of localizer limit on OPDV. 92
5.7 OPDV performance for clutter topology classes. 93
5.8 Constrained-localizer OPDV for clutter topology classes. 94
5.9 Comparison of OPDV and random localizer placement. 95
5.10 Setup for evaluation of APDV alignment variants. 97
5.11 APDV variants in error-free settings. 97
5.12 APDV variants in LOS error settings. 98
5.13 APDV variants in odometry error settings. 99
5.14 APDV variants in NLOS settings (preset positions). 100
5.15 APDV variants in NLOS settings (random positions). 101
5.16 Influence of step-size threshold on APDV (static case). 101
5.17 Influence of step-size threshold on APDV (dynamic case). 102
5.18 Influence of number of directions on APDV (static case). 103
5.19 Influence of number of directions on APDV (dynamic case). . . 104
5.20 Influence of number of localizers on APDV (static case). 104
5.21 Influence of number of localizers on APDV (dynamic case). . . . 105
5.22 Influence of LOS error on APDV (static case). 107
5.23 Influence of LOS error on APDV (dynamic case). 108
5.24 Influence of rotational odometry error on APDV (static case). . 110
5.25 Influence of rotational odometry error on APDV (dynamic case). 111
5.26 Influence of displace odometry error on APDV (static case). . . 111
5.27 Influence of displace odometry error on APDV (dynamic case). . 112
5.28 Influence of both odometry errors on APDV (static case). 113
5.29 Influence of both odometry errors on APDV (dynamic case). . . 113
5.30 Influence of clutter topology on APDV. 114
5.31 Performance traces for trajectory TR_1. 116

5.32 Performance comparison for trajectory TR_1. 118
5.33 Performance traces for trajectory TR_2. 119
5.34 APDV and basic OPDV for TR_2 120
5.35 Performance comparison for trajectory TR_2. 120

6.1 Hardware platform for anchor and localizer/sensor robots. . . . 124
6.2 Comparison of Cricket accuracy for different sound velocities. . . 126
6.3 Omni-directional distance measurement procedure. 128
6.4 Accuracy of omni-directional distance measurements. 129
6.5 Software architecture of the APDV implementation. 130
6.6 Experimental setup for Roomba experiments. 132
6.7 Performance evaluation of APDV on Roomba platform. 134

7.1 Sample clutter topologies for distance error characterization. . . 138
7.2 f and $\mathcal{P}$ for sample clutter topologies. 138
7.3 Multi-hop errors for sample clutter topologies. 140
7.4 Derivation of linear clutter spacing distribution, $\mathcal{CD}_l$. 142
7.5 Derivation of radial clutter spacing distribution, $\mathcal{CD}_r$. 143
7.6 Occupancy grid representation of clutter topology C2. 143
7.7 Fourier transform of clutter topologies C1, C2, C3. 144
7.8 GIST characterization of clutter topologies C1, C2 and C3. . . . 145
7.9 Generation of arbitrary clutter samples. 147
7.10 Graphical overview of SVR estimation of f. 149
7.11 Estimation of f using individual clutter features. 150
7.12 Effect of sample size for CD_l/CD_r on accuracy of f. 152
7.13 Graphical overview of k-NN-based estimation of $\mathcal{P}$. 153
7.14 Estimation of $\mathcal{P}$ using individual clutter features. 154
7.15 Box plots of estimation errors for k-NN based estimation of $\mathcal{P}$. . 156
7.16 Visual depiction of $\mathcal{P}$ estimations for sample clutter topology. . 157
7.17 Effect of k on k-NN estimation accuracy of $\mathcal{P}$. 158
7.18 Lognormal NLOS distribution fitting. 159
7.19 Visual depiction of $\mathcal{P}l$ for sample clutter topology. 161
7.20 Graphical overview of SVR-based estimation of multi-hop error. 163
7.21 Estimation of me with individual clutter features. 164

List of Tables

2.1 Existing NLOS/multi-hop error mitigation methods. 35

3.1 Parameter values for evaluation of multi-hop localization. 48

5.1 Parameter values for OPDV sensitivity analysis. 85
5.2 Parameter values for APDV sensitivity analysis. 86

7.1 Multi-hop distance errors for sample clutter topologies. 140
7.2 Measure of consistency level for f. 148
7.3 Measure of consistency level for $\mathcal{P}$. 149
7.4 Estimation of f using clutter topology features. 151
7.5 Estimation of $\mathcal{P}$ using clutter topology features. 155
7.6 Estimation of $\mathcal{P}l$ using clutter topology features. 160
7.7 Estimation of *me-mean* using clutter topology features. 163
7.8 Estimation of *me-median* using clutter topology features. 165
7.9 Estimation of *me-max* using clutter topology features. 165

Chapter 1

Introduction

Advances in micro-electromechanical systems (MEMS) technology have led to the proliferation of inexpensive miniature sensor and actuator systems that are also capable of wireless communication. Wireless sensor networks consist of such sensing/computing devices that are deployed to monitor an environment in a non-intrusive manner. Typically they are equipped with sensors to measure phenomena, like temperature, humidity or even radioactivity, while forming a cooperative multi-hop network. Certain applications, like habitat monitoring [104] or vineyard monitoring [11], require the deployment of a fixed sensor network whereas others [61, 89] require mobile sensors to perform the monitoring task.

Data is spatially indexed, hence it is critical that a sensor be aware of its own position at measurement time. In the case of static sensor networks, location information can either be hard-coded into the sensor, or in the case of large-scale, randomly deployed sensors, a number of multi-hop localization algorithms can be used [114, 69, 112] to enable the sensors to realize their positions. However, when the sensors are placed on mobile robots, localization can be a challenging task to achieve as a robot's position changes continuously with time. Here, localization is important for other reasons as well. If the task of the robots is to map unexplored environments, then it is all the more crucial for them to know their own positions in order to distinguish between the explored areas from those that are unexplored. The robots also need to know their own positions in order to travel to a pre-determined place for re-charging their batteries or for retrieval at the end of their exploration task.

Anchor-based localization is a widely used form of localization where anchors assist unlocalized nodes to realize their positions. Anchors are special purpose nodes with known positions in a given reference coordinate system.

The unlocalized node in effect derives its position with respect to the anchors' reference coordinate system. The unlocalized node typically infers information vis-a-vis the anchors in various forms. For example, in the Global Positioning System (GPS), a GPS sensor determines its position by estimating ranges to a number of satellites (anchors). In the case of multi-hop localization techniques such as DV-HOP [112] and MDS-MAP [118, 117], the unlocalized sensor nodes use hop-counts to anchors to localize themselves. A number of localization algorithms have been proposed where only proximity information to anchors is used for localization [35, 87]. Other anchor-based localization techniques [114, 99] use angle-of-arrival information from anchor beacons to estimate position.

In this dissertation, we focus on localization where the unlocalized nodes typically measure distances to the anchors. The presence of clutter in the midst of the anchors and the unlocalized node has a debilitating effect on the localization accuracy of range-based localization. The objective of this dissertation is to propose a novel approach to deal with this scenario, particularly when a majority of the ranges are obtained in the presence of clutter between the respective anchors and the unlocalized node. We propose using intermediate nodes to ameliorate the adverse effect of clutter on the measured ranges to the anchors.

The remainder of this chapter is organized in the following way. We discuss range-based localization and the effect of clutter on its accuracy in more detail in Section 1.1. We look at applications that motivate the need for range-based localization in cluttered environments in Section 1.2. We, then, discuss the shortcomings of current state-of-the-art research in addressing this issue in Section 1.3. In Section 1.4, we briefly discuss our proposed approach and highlight the key contributions of this thesis in Section 1.5. We list the author's publications related to this thesis in Section 1.6, and conclude with an outline of the dissertation in Section 1.7.

1.1 Research Problem

Range-based localization is a popular localization technique where the unlocalized node estimates its position by measuring ranges or distances to anchors. Each distance estimate to an anchor is used to constrain the location of the node with respect to the location of that anchor. When enough constraints

exist for the node, its location can be solved by using the analytical technique of trilateration.

The ranges/distances themselves can be estimated using a variety of mechanisms. For example, GPS sensors measure distances to satellites in the GPS constellation by determining the phase shift of the pseudo-random sequence (PRN) of microwave signals broadcast by the satellite. Each satellite sends out periodic signals that can be decoded by the GPS sensor using a corresponding code unique to the satellite. The phase shift of the PRN sequence is due to the travel time of the GPS satellite's signal.

In indoor environments, the MIT-Cricket mote uses the time difference of arrival between a radio signal and an acoustic signal to determine distance. During ranging, the Cricket mote sends a radio message and an acoustic chirp in quick succession. The recipient Cricket mote uses the radio message as a means for synchronization, given that radio waves travel much faster ($c = \sim 300$ million metres/second) compared to acoustic signals ($v = \sim 343$ metres/second). Thus, the recipient Cricket mote measures the time-of-flight of the acoustic signal from the transmitting Cricket mote without using an explicit synchronization protocol. Another ranging approach is to use received signal strength (RSS) and estimate distance based on the amount of decay in the signal strength of the radio signal as it travels in space [101, 137, 114].

In range-based localization, the primary mechanism for estimating distances over the wireless channel is to calculate the time of flight of the ranging signal. So the presence of obstacles in the environment between the ranging nodes gives rise to a subtle phenomenon of distance estimates obtained from reflected ranging signals. The distance measurement between the unlocalized node and the anchors ideally should represent the shortest path between the two. For example, an acoustic ranging signal would take the shortest path between the two communicating nodes if there is no obstacle in between the two. Distance estimates obtained from such signals are referred to as line of sight (LOS) distances. On the other hand, non-line of sight (NLOS) distances are obtained when the ranging signals bounce off surfaces before reaching their destinations. In the process, they travel longer distances, as shown in Fig. 1.1. The distance estimate is derived from the travel duration of the ranging signal, which obviously depends on the distance travelled by the signal. Thus, NLOS measurements introduce a large positive bias in the distance estimates, resulting in

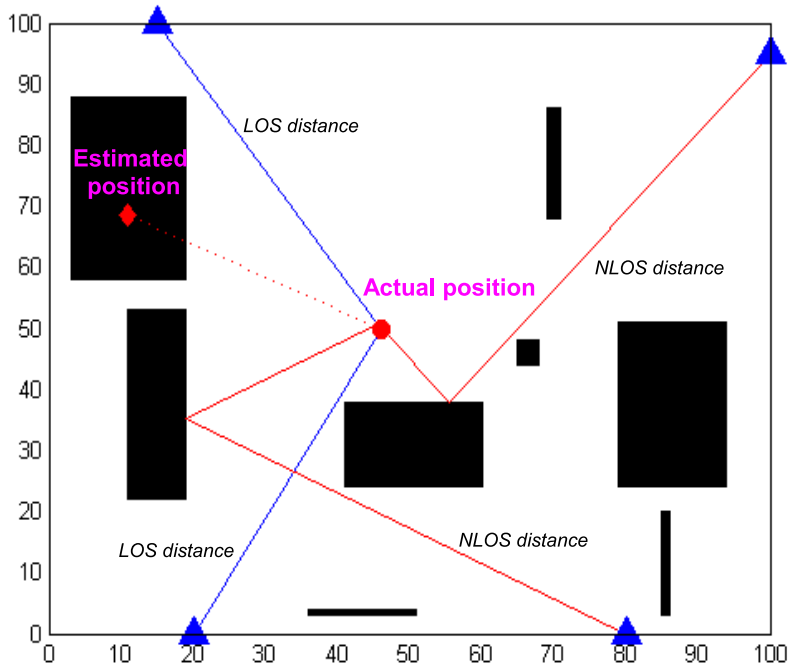
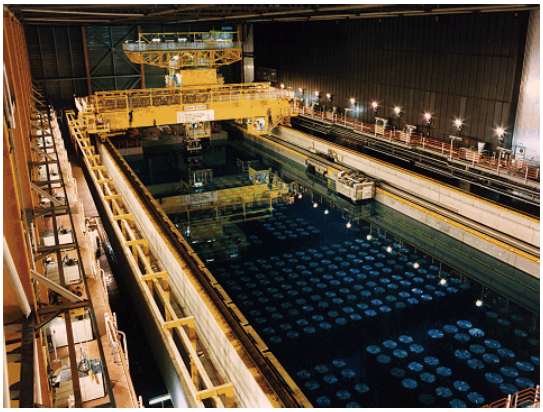


Figure 1.1: Line of Sight (LOS) and Non Line of Sight (NLOS) distance measurements in cluttered environments. Four anchors (denoted by large blue triangles) are used to localize the red dot. Two LOS distances and two NLOS distances are incident at the point. The estimated position is denoted by the red diamond. The localization error here is 39.61 units.

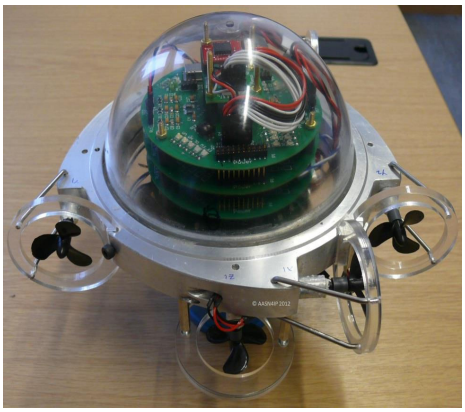
large errors in position estimation. For example, in Fig. 1.1, we see that NLOS distances cause a large localization error.

The effect of clutter on the range measurement errors is dependent on both the ranging signal and the clutter type. For example, acoustic signals reflect off even thin surfaces of wood, plastic and metal materials while radio signals reflect off metal surfaces but pass through dielectric materials like glass and concrete. This also depends on the frequency of the signals, with low frequency signals able to penetrate clutter much more easily than high frequency ones.

In the next section, we will look at two real-world applications that motivate us to consider the problem of range-based localization in cluttered environments.



(a) Nuclear Waste Storage Tank at Sellafield, Cumbria (Courtesy: National Nuclear Laboratory, Cumbria, UK)



(b) Proposed μAUV for the AASN4IP project (Courtesy: Simon Watson, University of Manchester, UK)

Figure 1.2: Mapping of radioactivity levels in nuclear waste storage ponds using micro-AUVs.

1.2 Motivating Applications

In this section, we will look at two applications that call for the deployment of robot swarms for exploration in cluttered environments. The localization of these sensor robots is both necessary and challenging. The first application is that of mapping radioactivity levels of a nuclear waste storage tank. The second application is that of exploring hazardous disaster zones using mobile robots. Both applications entail the operation of robots in GPS-denied indoor settings.

The management of nuclear waste is a pressing problem in our age as we look to move away from the more polluting forms of energy such as fossil fuels, and towards cleaner and sustainable alternatives like nuclear and solar energy. A nuclear power plant is a thermal power station powered by a nuclear reactor. Currently there are 437 nuclear power reactors in operation in 31 countries [1]. Spent nuclear reactor fuel has enough residual radioactivity to radiate considerable amounts of heat, making them unsuitable for deep underground storage immediately after removal from the nuclear reactor. Hence, the spent fuel is first cooled in large storage water tanks for long periods of 10 to 20 years as water is an excellent insulator of radioactivity. Modern nuclear waste disposal facilities require that the spent fuel be enclosed in metal canisters and stored in an orderly manner (as one can see in Fig. 1.2(a)), though previously it was merely dumped unshielded, in large metallic skips, into the storage tanks. In

order to manage such tanks, i.e., to remove parts of the debris which have radiation levels below a certain threshold, it is essential to have a map of the radioactivity of the entire tank. Such a survey would facilitate the removal of depleted nuclear fuel without disturbing, or bringing to the surface, other active nuclear material.

One possible solution is to use a robotic arm laden with radioactivity sensors that can be inserted in various sections of the tank to measure the radioactivity levels. However, the robotic arm might collide with the debris and get damaged when it is being moved around across various sections of the tank, or may even disturb debris as it is being inserted into the tank. The maintenance of the robotic arm may require human involvement, clearly undesirable given the hazardous working conditions. The other option is to deploy swarms of submersible robots or micro autonomous underwater vehicles (μAUVs) in the nuclear waste storage tank. The μAUVs can be equipped with temperature and radioactivity sensors to measure these phenomena in the environment. The μAUVs are able to manoeuvre in close spaces amidst the debris without disturbing it using proximity and obstacle-detection sensors, unlike a robotic arm. Moreover, the μAUVs offer the prospects of human-free maintenance, with a failed μAUV merely being abandoned at the bottom of the tank in the worst-case scenario. The μAUVs communicate wirelessly with base-stations on the tank's surface to obtain instructions for exploration missions as well as to report sensor measurements.

If a μAUV only records a sensor reading without recording its location, such information can only be used to deduce aggregate statistics like the average, maximum or minimum values. To construct a map of radioactivity and temperature levels across the tank, the μAUVs need to be aware of their positions in the tank at the time of recording sensor readings so that the measurements can be indexed according to their locations. For the purpose of localization of the μAUVs, anchors can be installed at the tank's surface. The μAUVs need to estimate their distances to the surface-anchors in order to estimate their positions. Since the μAUVs are exploring the environment among the debris, it is very likely that they will be occluded from the surface-anchors, especially as the μAUVs explore deeper into the storage tank. This results in μAUVs measuring NLOS distances to the anchors. The resulting localization errors preclude the sensor measurements from being used in the construction of the map of sensor readings.

(a) Fukushima nuclear meltdown site and the damaged reactor buildings. (Source: Baird Maritime)

(b) 'Road Runner' robot deployed by Tokyo Electric Power Company (TEPCO) to map the extent of the radioactivity leak. (Source: TOPY Industries, http://www.topy.co.jp)

Figure 1.3: Deployment of robots in disaster situations for mapping and exploration under conditions considered too hazardous for direct human involvement. Here, tethered robots were deployed at the Fukushima Daiichi nuclear reactor site, after the earthquake and subsequent meltdown, to map the extent of the radioactivity leak.

The deployment of robots to assist in damage control in disaster sites offers another application of the operation of robots in cluttered environments. Typically, the conditions are deemed too hazardous for direct human intervention or there is a need for small size robots to manoeuvre through close spaces in disaster areas. For example, during the 2011 Tohoku earthquake in Japan, the Fukushima Daiichi nuclear power plant near Tokyo city was severely damaged, as shown in Fig. 1.3(a). The resulting tsunami flooded the low-lying rooms housing the emergency generators, thus preventing the operation of the critical pumps required to continuously circulate coolant water through the nuclear reactors during emergency situations. The result was the meltdown of the reactors from being overheated by the high radioactive decay heat, releasing radioactive materials into the ocean. The Tokyo Electric Power Company (TEPCO), responsible for the reactor stabilization and cleanup operations, used remote-controlled robots to assess the damage and radioactivity levels in the damaged nuclear reactor. They deployed a tethered robot, the 'Road Runner', as shown in Fig. 1.3(b). However, it was reported that the 'Road Runner' was disabled due to the tether entanglement and was lost during the operation. While tethered robots have the advantageous capability of manual control by human tele-operators, their operation is severely restricted in cluttered environments due to the problem of tether entanglement and also because

their large sizes make them unsuitable for nimble manoeuvring in tight spaces amidst the debris.

Small autonomous vehicles have the advantage of being able to move through small spaces in the debris of such disaster zones. Instead of deploying a single large tethered robot, it might be more beneficial to deploy a large number of small robots that can form a multi-hop wireless network amongst themselves. Thus, even if some of the robots move deeper into the cluttered disaster site, the other robots can form the vital link back to the central base-station where humans can provide mission directives and also evaluate the measurements taken by the robots. Since numerous robots are used, the failure of a few robots can be handled elegantly. However, unlike in the case of the tethered robot where location can be determined by visual aids and robot odometry, the localization of mobile untethered robots is a challenge. Anchor-based localization offers a suitable alternative. Here, again, we find that the cluttered environment makes accurate localization of the robots a formidable task.

1.3 Challenges

Localization in NLOS situations has been widely studied over the past decade, particularly for cellular mobile terminal localization in urban settings. This is partly stimulated by the Federal Communications Commission (FCC) requirement for US cellular operators to be able to locate a mobile terminal within 100 metres for 67% of the time and 300 metres for 95% of the time [39]. Traditionally, there have been two approaches to address the problem: first, where distances are classified as LOS or NLOS and the NLOS distances are discarded; and second, where the NLOS distances are incorporated in the position estimation itself, albeit with reduced influence. The second approach, that of incorporating the statistical information of the NLOS error, has actually been shown to be beneficial in reducing localization error [58, 106, 45]. Robust estimators [45, 74] have been proposed to suppress NLOS distances which can be viewed as outliers. Other techniques [55] use certain statistics of the received ranging signal itself, such as kurtosis, mean excess delay and root mean square (RMS) delay spread of the received signal, to determine the weights for the estimator.

However, most existing NLOS mitigation approaches have been designed for scenarios where NLOS distances form the minority of all available distances to

anchors. Residual-based NLOS mitigation [23] will not work if a majority of the distances are NLOS or even if the number of LOS distances themselves are not sufficient for localization. Nawaz et al. [91] require that NLOS distances form a maximum of half the total distances when certain positivity constraints are met — the threshold is reduced to a third of all distances otherwise. Similarly, Kung et al. [66] propose a NLOS mitigation technique that breaks down when distances with large non-Gaussian errors constitute 30% or more of all anchor distances. Though Venkatesh et al. [129, 128] shows that their technique works even in the scenario where NLOS distances are the majority, they require a complete identification of LOS and NLOS distances is required. Hence, there is a need to address the scenario where a majority of distance measurements, or even all, are NLOS in nature, and classification of distances as LOS or NLOS is a formidable task.

SLAM-based localization techniques [17, 33] in obstacle-prone environments offer an alternative to range-based localization. Still, SLAM-based techniques have a number of disadvantages: loop closure [92] is a challenge for arbitrary deployment scenarios; expensive sensors, such as cameras or laser scanners, have to be mounted on the robots for using SLAM techniques. The requirement for high-quality cameras and lasers also precludes the robot from being lightweight and inexpensive as desired in the applications seen in the previous section. Moreover, most SLAM-based approaches assume that errors in measurements are Gaussian in nature, which is not the case when distance measurements to anchors [82, 34] are taken in cluttered environments. Particle filters [8] can deal with non-Gaussian error distributions, but not with the fact that these distributions depend on clutter [54, 59] and are unknown in NLOS-prone environments with arbitrary clutter topologies.

Inertial tracking, using inertial measurement units (IMUs), offers another alternative for the robot localization, especially when the robots are deployed with known initial positions. IMUs typically contain three orthogonal gyroscopes and three orthogonal accelerometers, measuring angular velocity and linear acceleration respectively. By processing signals from these devices it is possible to track the position and orientation of a robot. However, since the position is obtained via the double integration of accelerometer readings, the biases in the gyroscope and accelerometer sensors lead to a quadratic increase in the positional error. Hence, while inertial positional tracking works fine for

short distances travelled by the robots, it requires periodic corrections via information from other localization sources to deal with the growing error. Inertial measurements, thus, cannot be used independently to localize the robots over long periods of time.

1.4 Proposed Approach

The solution proposed in this dissertation for the aforementioned challenge is motivated by the observation that, given an obstacle-prone environment, the chance of obtaining an NLOS distance estimate between two nodes, in an obstacle-prone environment, is inversely proportional to the actual distance between the two nodes. In other words, the probability of an obstacle blocking the direct line-of-sight path between two nodes increases with the actual distance between the nodes. For instance, a mobile sensor robot deployed in a cluttered underwater tank is more likely to be occluded from the anchors, fixed at the surface, as it moves deeper into the tank. Since it is not feasible to move the anchors themselves into the cluttered tank, the other feasible option is to use intermediate nodes to shorten the overall inter-node distances. We would prefer the scenario where such intermediate nodes are mobile in nature and thus allowing a relatively small number of nodes to suffice, instead of having to deploy a large number of intermediate nodes in fixed positions in the cluttered environment. We use the term *localizers* to describe these intermediaries, as their primary purpose is to assist a mobile sensor robot that is doing the actual sensing/mapping task. While localizers have some similarities with anchors, in that they both assist an unlocalized node to estimate its position, there exist significant differences between the two in terminology. Anchors are self-sufficient permanent entities that are aware of their own (geographical) positions to a degree of high accuracy. Localizers, on the other hand, can be aware of their own positions, and also can be unlocalized themselves (as we will see later on in this dissertation), in the process of assisting another unlocalized node to estimate its position with better accuracy. Localizers can be 'created' in an ad-hoc manner depending on their current placements with respect to the cluttered environments. Multi-hop localization techniques can then be used with these localizers in order to localize the sensor robot. Multi-hop localization was initially introduced for localizing fixed sensor nodes in large area deployments where all the nodes could not gain direct access to a

sufficient number of anchors in order to localize themselves. However, here, we propose using multi-hop localization in scenarios where all the nodes have direct single-hop access to the required number of anchors, but the distance measurements are NLOS in nature due to obstacles between the anchors and the unlocalized sensor robot.

1.5 Contributions

The main contributions of this thesis are:

1. We perform an empirical study to show that using localizers and multi-hop localization is preferred to direct single-hop localization in NLOS-prone situations. We compare the performance of iterative localization, DV-Distance and MDS-MAP in NLOS-prone conditions and show that DV-Distance and MDS-MAP outperform single-hop localization as well as iterative localization.

2. We propose a centralized localizer placement algorithm, namely Optimal Placement for DV-Distance (OPDV), as an oracle-type algorithm that assumes knowledge of the complete clutter topology to generate optimal positions for placing localizers.

3. We propose a distributed localizer placement algorithm, namely Adaptive Placement for DV-Distance (APDV), as a practical algorithm that does not require the localizers to know the clutter topology or even to localize themselves. It carefully moves them to positions in the midst of the clutter that improve the localization accuracy of a target node.

4. We evaluate the optimal and adaptive placement techniques in both simulation and real robot testbeds.

5. We propose characteristic features for clutter topologies and study their impact on NLOS bias and multi-hop distance error.

1.6 Publications

The work in this thesis has been done in the context of the EPSRC project EP/F064209/1 entitled ‘Acoustic Actuated Sensor Networks for Industrial Pro-

cesses (AASN4IP)'. Parts of this thesis have been published in the following international conferences and journals:

1. **M. Hussain** and N. Trigoni, Adaptive node placement for improving localization accuracy in clutter-prone environments, in IEEE Wireless Communications and Networking Conference (WCNC), 2013.

2. **M. Hussain**, Y. Aytar, N. Trigoni, and A. Markham, Characterization of non-line-of-sight (NLOS) bias via analysis of clutter topology, in Proceedings of IEEE/ION Position Location and Navigation System (PLANS) Conference, 2012.

3. P. Green, P. Green, **M. Hussain**, S. Narwaz, A. Phasouliotis, Z. Qu, N. Trigoni, S. Watson, and T. York, Mapping legacy storage ponds, Nuclear Futures Journal, 2011.

4. S. Watson, T. York, P. Green, P. Green, **M. Hussain**, S. Nawaz, A. Phasouliotis, S. Qu, Z. Stanley, and N. Trigoni, Acoustic sensor networks for decommissioning, in Proceedings of Control & Instrumentation in Nuclear Installations (CANDI), 2011.

5. **M. Hussain** and N. Trigoni, Demo abstract: Dynamic node placement for multi-hop localization in cluttered environments, 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011.

6. **M. Hussain** and N. Trigoni, Distributed localization in cluttered underwater environments, Proceedings of the Fifth ACM International Workshop on UnderWater Networks (WUWNET), 2010.

7. S. Nawaz, **M. Hussain**, S. Watson, N. Trigoni, and P. Green, An underwater robotic network for monitoring nuclear waste storage pools, Sensor Systems and Software, 2010.

In papers 1, 2, 5 and 6, I have contributed towards the main ideas, generated all of the experimental results and have significantly contributed to the writing of the papers. For papers 3, 4, and 7, I contributed towards those sections of the paper that dealt with localization of a swarm of robot μAUVs in cluttered environments.

1.7 Thesis Structure

The remainder of the dissertation is organized as follows. In Chapter 2, we provide an overview of localization techniques and state-of-the-art work on localization in cluttered environments. In Chapter 3, we evaluate the performance of various multi-hop localization techniques in cluttered NLOS-prone environments. In Chapter 4, we describe the two localizer placement algorithms, namely OPDV and APDV. In Chapter 5, we perform an extensive sensitivity analysis of the OPDV and APDV algorithms and also compare their performances to various competing localization techniques. In Chapter 6, we discuss the implementation and evaluation of APDV on a realistic robotic platform. In Chapter 7, we investigate the role played by clutter topologies in determining single-hop NLOS error and multi-hop error. Finally in Chapter 8, we present concluding remarks and discuss ideas for future work.

Chapter 2

Background

Localization is a wide and active area of research that has applications in sensor networks, ubiquitous computing and autonomous navigation among other areas. Cluttered environments offer unique challenges for localization algorithms in terms of loss of accuracy and localization success-rate. In this chapter, we place our work within the context of the large body of literature in localization for cluttered environments. First, in Section 2.1, we define the problem of localization and look at a classification of localization techniques. Then we discuss the challenges of localization in cluttered environments, and discuss current approaches used to tackle the problem in Section 2.2. Finally, in Section 2.3, we look at the Simultaneous Localization and Mapping (SLAM) techniques, which are popular in the mobile robotics community, for clutter-prone environments.

2.1 Localization

Localization is the process of determining one's location. Knowledge of location is vital in wireless sensor and robotic networks (WSRN) where a large number of wireless sensors and a few robotic vehicles are deployed to achieve certain tasks such as mapping the environment or locating a region of interest. In other words, localization is important where it is necessary to *spatially index measurement data*. For example, a submersible robot, equipped with sensors to measure phenomena like temperature and turbidity and deployed in a underwater tank, needs to know its position at the time of recording the sensor readings for constructing a map of the tank for highlighting temperature/turbidity contours across various regions of the tank. Similarly, a robot involved in search-and-rescue operations needs to be able to inform its human operators of the locations of people and items it discovers on the way.

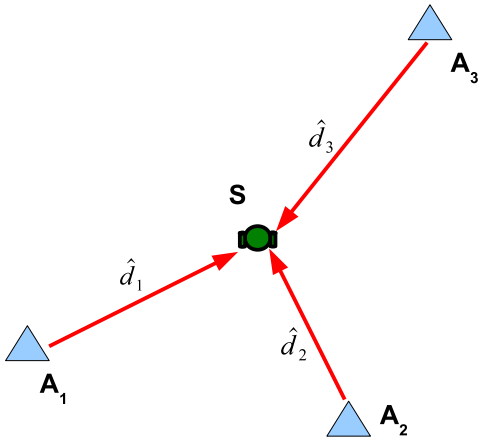


Figure 2.1: Anchor-based localization of robot S. Here the positions of anchors A_1, A_2 and A_3 together with the measured distances d_1, d_2 and d_3 are used to calculate the position of S.

Localization consists of two main steps [69, 116]: first, that of measuring geographical information from the ground truth of a pre-existing network deployment and, second, that of estimating location from the measured data. For simplicity and consistency purposes, we will consider the two dimensional scenario, since the extension to three dimensional scenario is straightforward. In the following paragraphs we take a closer look at the actual process of localization. We go through the various steps of a popular position estimation technique and thereafter discuss other methods for calculating location.

Fig. 2.1 shows a network setup where a sensor robot S is assisted in localization by anchors A_1, A_2 and A_3. Anchors, introduced in the previous chapter, are special purpose nodes which are aware of their own positions and are tasked with assisting unlocalized nodes/robots to localize. Here, S measures distances $\hat{d}_1$, $\hat{d}_2$ and $\hat{d}_3$ to each of the three anchors respectively. Using the distance information and the locations of the anchors themselves, tri-lateration can be used to calculate the position of the sensor robot. If (x_i,y_i) denotes the location of anchor A_i, we can obtain the following formulation:

$$\begin{aligned}(x-x_1)^2+(y-y_1)^2&=\hat{d}_1^2\\(x-x_2)^2+(y-y_2)^2&=\hat{d}_2^2\\(x-x_3)^2+(y-y_3)^2&=\hat{d}_3^2\end{aligned} \tag{2.1}$$

However, in real scenarios, the locations of the anchors may themselves be inaccurate or the measured distances can be noisy. In such circumstances, the

above equations are inconsistent and estimation techniques, like non-linear least squares [**?**], have to be used. A simple alternative method is to linearize the above equations and solve using linear least squares estimation. Subtracting the first equation from each of the other two equations would result in two linear equations which can be solved directly for x and y.

$$-2(x_2 - x_1)x - 2(y_2 - y_1)y = (\hat{d_2^2} - \hat{d_1})^2 - [(x_2^2 - x_1^2) + (y_2^2 - y_1^2)]$$
$$-2(x_3 - x_1)x - 2(y_3 - y_1)y = (\hat{d_3^2} - \hat{d_1})^2 - [(x_3^2 - x_1^2) + (y_3^2 - y_1^2)]$$

The above formulation has thus been represented as a linear system, given below in Equation 2.2. It has the convenience of being implementable on simple computation-constrained devices like wireless sensors.

$$AX = b \text{ where } A = -2. \begin{bmatrix} (x_2 - x_1) & (y_2 - y_1) \\ (x_3 - x_1) & (y_3 - y_1) \end{bmatrix} \tag{2.2}$$
$$X = \begin{bmatrix} x \\ y \end{bmatrix}$$
$$b = \begin{bmatrix} (\hat{d_2^2} - d_1^2) - [(x_2^2 - x_1^2) + (y_2^2 - y_1^2)] \\ (\hat{d_3^2} - d_1^2) - [(x_3^2 - x_1^2) + (y_3^2 - y_1^2)] \end{bmatrix}$$

There are a number of alternative techniques for calculating the position of the sensor robot. Triangulation is a technique which is based on angular measurements between the anchor and the unlocalized node. Angular measurements require specialized hardware which can detect the orientation of the incident communication signals, where the signals can be either radio or ultrasound. Medusa [114] and the more recent SpiderBat [99] are examples of ultrasound angle-of-arrival hardware equipped devices. Localization based on based measurements is shown to be inferior to that done with distance measurements [113]. Angular measurements are typically used to estimate distances to anchors, using the law of cosines and thereafter tri-lateration is applied [12]. Time Difference of Arrival (TDOA) based position estimation uses the differences in the arrival times of the emitted signal at various tightly-synchronized receivers. Correlation analysis provides a representation of the path differences of the signal to any pair of receivers which can be modelled as hyberboli. The point of intersection of these hyberboli yields the position of the signal-emitting node [124, 44, 36]. The main drawback of TDOA-based estimation is the prerequisite of synchronized receivers thus making it unviable for ad-hoc wireless sensor networks deployment. Finally, the Min-Max technique [115] offers a

simple, though coarse, method to estimate node position by calculating the intersection of bounding boxes representing each anchor position and the corresponding range. The center of this intersection area is approximated as the node position.

In the next section, we will give a broad overview and classification of localization methods.

2.1.1 Classification of Localization Techniques

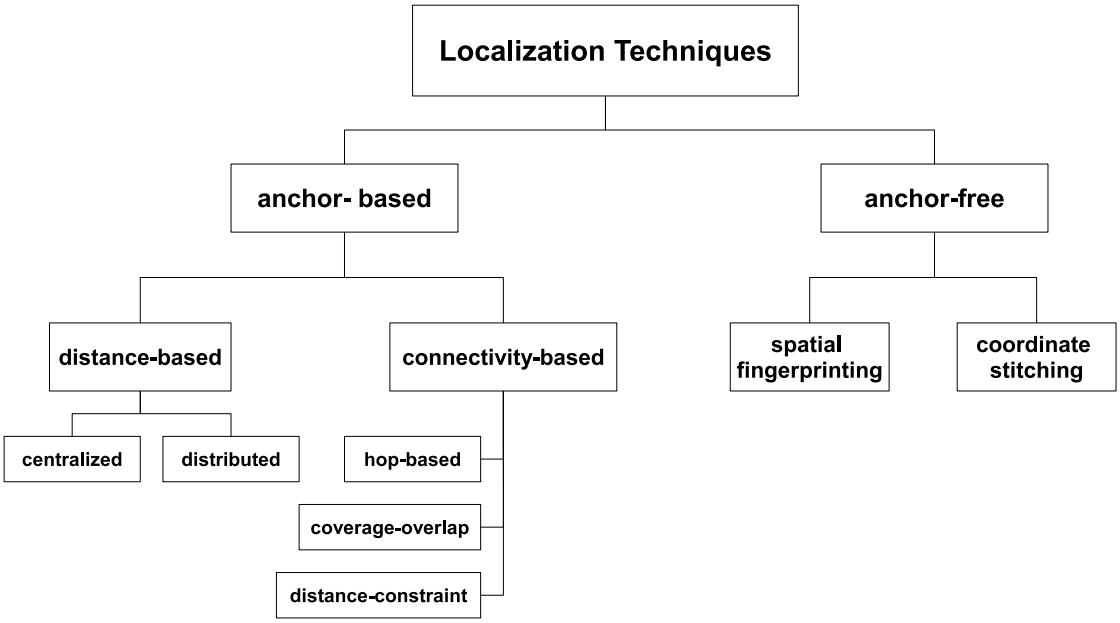


Figure 2.2: Classification of localization for wireless sensor and robotic networks

The primary classification we propose is that of *anchor-based* and *anchor-free* (alternative classifications can be found in [116, 78]. Anchors help define the geographical ground-truth information that needs to be collected by the unlocalized nodes before they calculate their positions. Typically, an unlocalized node estimates its distance to the anchors and uses the geographic positions of the anchors to estimate its own location in the environment. Anchors can range from multi-million dollar, highly-sophisticated satellites, as used in the Global Positioning System [50], to simpler wireless devices, equipped with acoustic and radio communication units, used in the Cricket location system [105]. Anchors are typically manually set up or use an external positioning system (like GPS) to obtain an accurate position themselves. Thus, they are generally more expensive than the nodes they are required to assist in localization. Hence, an

effort is made by all localization techniques to minimize the required number of anchors for a given network while maintaining a required localization accuracy throughout the network.

Anchor-based localization techniques are further divided into distance-based and connectivity-based methods. Distance-based methods can be further subdivided into centralized and distributed categories. Collaborative multilateration [115], semi-definite programming (SDP) based convex optimization [35], simulated annealing based optimization [62], and MDS-MAP [118] are examples of the centralized approach. Here, distance measurements from throughout the network are collected at a central location before being processed to estimate the positions of all nodes in the network at once. On the other hand, distributed methods allow a node to independently calculate its own position. The recursive iterative localization method [4, 114] and DV-Distance [97] are examples of such methods.

Connectivity-based localization techniques use proximity information to anchors to approximate distances to them used in the position estimation process. These are further divided into hop-based, coverage-overlap and distance-constraint categories. In hop-based approaches, the distances are represented by the number of hops an anchor advertisement travels in a multi-hop network topology. Here, a separate metric is later used to convert the hop-distances into actual distances. DV-Hop [96] and Hop-TERRAIN [112] as well as Nagpal et al. [87] and Shang et al. [118] are a few examples of hop-based methods. Finally, the coverage-overlap category methods use the positions of the anchors directly without estimating the distances to them at all. For example, Bulusu et al. [16] estimate the position of a node as the centroid of the polygon defined by the anchors it can reliably listen to. Similarly, He et al. [49] propose a technique whereby the unlocalized node determines if it is inside the triangle formed by arbitrary sets of three anchors taken from the set of all anchors it hears. The idea is to determine the overlap area of these triangles, with the center of gravity of the overlap area representing the position estimate. The primary issue with connectivity-based approaches is that they can only provide coarse-grained localization accuracy. The accuracy provided by such techniques suffice only for applications like geographic routing. Another issue with these methods is that they require isotropic or dense network topologies for good performance. In the distance-constraint category, the distance between two communicating nodes are constrained (and approximated) by the

communication range $\mathcal{R}$. Doherty et al. [35] formulate a convex optimization problem for localization of the entire network using distance approximated by communication ranges.

Anchor-free localization methods are attractive because they do not require the installation of anchors, which can be both costly and time-consuming. We further divide such techniques into the spatial fingerprinting and coordinate-stitching. In spatial fingerprinting, a detailed off-line characterization/profiling of communication signals (typically radio) is done for a confined area. The location is then identified based on a database of these signal signatures. This method assumes that it is possible to uniquely characterize every location in the environment based on the signal characteristics, which may not be possible for all cases. The main advantage of this approach is that it is immune to fixed environmental distortion of communication signals, for example by walls and obstacles. This is important, as we will see later in the dissertation, as errors introduced by environmental features like obstacles can cause large errors in location estimation. RADAR [9], ARIADNE [56] and LOCATOR [2] are examples of localization where spatial fingerprinting of radio signals is employed. Coordinate stitching techniques work by initially building a local coordinate system for each node in the network and then stitching them together, typically in a distributed manner, into a relative coordinate system for the entire network. Examples of this approach include the contributions of Capkun et al. [19], Moore et al. [86] and MDS-MAP(P) [117]. Capkun et al. [19] propose the GPS-Free algorithm where each node in the network initially builds a local coordinate system with itself as the origin, and positions of its immediate neighbours are fixed in this local coordinate system. Then, adjacent local coordinate systems are aligned to be the same by using common nodes (neighbors) between the adjacent nodes. Finally, linear translation is used to translate all the local coordinate systems into a single global coordinate system. GPS-Free suffers from error propagation of the distance errors and also the large overhead of translation of coordinate systems. Moore et al. [86] manages to restrict the propagation of error due to distance errors by using *robust-quads*. A robust quad is defined as a fully connected quadrilateral whose four sub-triangles are 'robust', in that their allowed angles have a lower bounded determined by the noise characteristics of the distance measurements. MDS-MAP(P) [117] is another example of the coordinate stitching approach and was developed as a solution to MDS-MAP not being able to perform well in non-isotropic

networks. Here, each node builds a local map of neighbour distances for a given communication hop-length. Since the shortest-path distances are small in hop-length, the non-isotropicity does not affect the accuracy as much as in MDS-MAP [118].

Besides the techniques mentioned above, Markov localization approaches [17] are popular for mobile robot localization in realistic environments. Markov localization is a probabilistic algorithm where instead of maintaining a single hypothesis as to where in the environment a robot might be, a probability distribution over the space of all such hypotheses is established. It assumes that if the robot's location is known at time t, future measurements at times $t+1$, $t+2$, $t+3$,.... are independent of past ones. Both anchor-based [123, 108] and anchor-free [75, 71, 92] versions exist of this localization paradigm. The Extended Kalman Filter (EKF) localization is a sub-optimal solution of Markov localization used in a number of scenarios [123, 108, 75, 71, 7]. Here, the position is estimated along with an associated position error (represented by the covariance matrix), given a feature-based map of the environment. The kinds of measurements (or observations) can vary from laser-scanner data of the robot's immediate vicinity [75], sonar measurements [71], or visual data [7], besides range measurements from anchors [123, 108]. EKF localization makes the critical assumption that the measurement error is Gaussian in nature. Simultaneous Localization and Mapping (SLAM) attempts to solve the robot localization problem without a pre-existing feature-based map [17, 10, 94]. It has been a very active area of research given the immense impact of being capable of deploying a self-localizing robot in an unfamiliar environment [29, 92, 34, 51]. We will discuss the SLAM-based approach in more detail in Section 2.3.

Next, we look at various ranging modalities that are found in the literature.

2.1.2 Ranging Techniques

In this section we will discuss the existing methodologies for determining distances between two nodes using wireless communication signals. We look at four different methodologies, namely signal strength, time of flight, radio inferiometry and angle of arrival, and discuss the pros and cons for each.

2.1.2.1 Signal Strength

Signal strength based distance estimation uses the fact that the signal strength of a communication signal decreases as it travels in space. The Friis trans-

mission model [40] is a well known expression for estimating the power of the received signal, where signal strength decreases by the square of the distance travelled. However it is valid for ideal conditions where the antennae are isotropic and RF power radiates in a spherical manner. The log-normal model proposed by Nakagami et al. [88, 107] is shown to be more realistic in modelling the path loss of radio signal propagation:

$$RSS(d) = P_t - PL(d_0) - 10\alpha log_{10}\left(\frac{d}{d_0}\right) + \chi_\sigma \tag{2.3}$$

where P_t is the transmission power, $PL(d_0)$ is the path loss for a reference distance d_0, α is the path loss exponent, and χ_σ is a Gaussian random variable $\mathcal{N}(0, \sigma^2)$, that models random variations in the RSS value. α is obtained from empirical data. Mao et al. [77] propose a method to estimate the path loss exponent using only RSS measurements without the actual distances.

Radio signal strength (RSS) has been used to determine distances between wireless sensors [101, 137, 114]. RSS-based distance estimation is well suited for wireless sensor and robotic networks since it does not require any additional hardware or computational costs. However it is shown to be unreliable even in simple test deployments. Patwari et al. [101] show that it is possible to achieve an accuracy of 1.8 metres using RSS-based localization but only under strict test conditions. Studies for outdoor deployments have shown that the best possible accuracy is around 50% of the communication range [114]. The reason for this is that radio propagates highly non-uniformly in real environments. The reflections and absorption of radio signals by obstacles in the environment make it difficult to predict the path loss. Signal strength based ranging with acoustic/ultrasound signals is not popular because of the high attenuation of acoustic signals over relatively small distances. Also, acoustic signals lose considerable energy when they collide with clutter, severely curtailing their transmission ranges.

2.1.2.2 Time of Flight

Estimating distance between nodes via signal propagation time, instead of signal attenuation, is a popular and proven way of distance measurement in wireless sensor networks. It is shown to be susceptible to less error than RSS-based

methods. However, the main requirement for this technique is the synchronization of the transmitter and the receiver to the same time base in order to measure the time of flight of the signal.

Radio: Time-of-flight of radio signals is a challenging task for short distances, as found in the case of wireless sensor networks, given the extremely high speed of light, $c = 300{,}000$ km/s. Thus, in order to measure time of flight for short distances, the transmitter and receiver have to be synchronized extremely tightly, as even a small clock drift can result in large errors in distance measurements. Radio time of flight is used in the case of GPS localization, where the GPS satellites are hundreds of miles away from the GPS receiver. Here, clock offset is measured as an additional parameter in the set of equations used for position estimation. 2.4 Ghz band radio ranging without explicit synchronization between two radios has been done using the symmetric double-sided two-way ranging (SDS-TWR) technique [123, 108]. Ultra-wideband (UWB) radio has shown promise in recent times [42, 111, 110]. Here, the nodes need not be time synchronized as round trip time (RTT) is used to measure time-of-flight. Lanzisera et al. [70] have shown that using RTT-based distance estimation with IEEE 802.15.4 Zigbee radios can yield RMS errors with a range of 1 metre to 3 metres in indoor environments.

Acoustic: Acoustic time-of-flight has been a popular method of estimating distances given the relatively inexpensive hardware required and also the high accuracy of the distance estimates. One reason for this is the slow speed of acoustic signal propagation when compared to that of radio. The vast difference between the speeds of the two types of signals is exploited to provide synchronization for nodes measuring distances using acoustic time of flight. Suppose two nodes, T and R, are involved in the ranging, where T is the transmitter and R is the receiver. T first sends a radio signal to R, upon reception of which R starts the clock to measure the time of arrival of the acoustic signal. This simple setup is shown to achieve accuracy of 1 cm even with a clock of only 32kHz [116]. Examples are Active Bats [135] and the MIT Cricket [105]. Acoustic signal attenuation near the ground is shown to be less than that of radio. However, the main disadvantage of acoustic signals is that they are easily affected by the environment as temperature gradients in the deployment environment can influence the speed of acoustic signals. Moreover, acoustic signals can be easily blocked and reflected by obstacles in the environment.

2.1.2.3 Radio Inferiometry

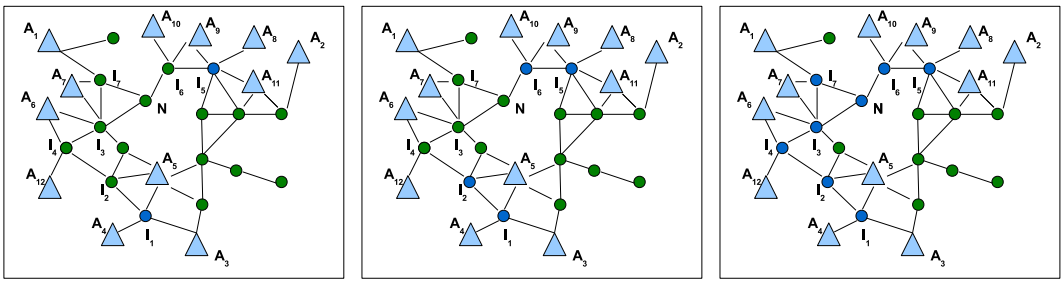
Radio inferiometry is a technique that exploits characteristics of the phase difference in radio signals to estimate distance between nodes. If two nodes T and R are transmitting at frequencies f_T and f_R respectively such that the difference between the frequencies is very small, a low beat frequency is created at two observer nodes O_1 and O_2. It is shown that the phase difference of the low-beat frequency between O_1 and O_2 is primarily dependent on the distances between the four nodes involved. Repeating this process with other sets of nodes can create an over-constrained system from which the distances between the nodes can be obtained. It is shown that this technique can achieve an accuracy of nearly 3 cm over ranges of up to 160 metres with the standard CC1000 radio in the mica2 platform [80]. However, the technique assumes precise time synchronization throughout the network to estimate the phase difference of the low-beat frequency between the observer nodes. It also assumes that there are no obstacles in the environment, which is shown to change both the observed phase difference as well as the beat frequency.

2.1.2.4 Angle of Arrival

The angle of arrival (AoA) technique works by using specialized hardware such as an array of antennae in the case of radio signals, or an array of microphones in the case of acoustic signals. By analysing the phase or time difference of the incident signal, the receiver can determine the angle from where the transmitted signal came. Medusa [114] and SpiderBat [99] are examples of microphone arrays developed for this purpose. Once an unlocalized node is able to estimate its bearing to two anchors, it is able to determine the distances to them using the law of cosines, since the distance between the two anchors is known [38]. The disadvantage of this approach is that the special-purpose hardware is more expensive and bulkier than the ones that are required for the RSS-based or time-of-flight based estimation.

2.1.3 Multi-Hop Localization

In this section we discuss multi-hop localization in more detail since it holds an important place in this dissertation. We look at four multi-hop localization algorithms, namely iterative localization, DV-Distance, DV-HOP and MDS-MAP, discussing their error characteristics and the current state of the art.



(a) Nodes I_1 and I_5 are localized first

(b) I_1 and I_5 thereafter assist I_2 and I_6, respectively, to localize

(c) N is localized with the help of I_3, I_7 and I_6

Figure 2.3: Iterative localization in action. 12 anchors are used for the localization for all nodes in the network. Nodes which have access to at least three anchors localize first. Thereafter, these nodes assist other nodes in their vicinities, in addition to the anchors, to localize themselves.

Multi-hop localization was originally envisaged in wireless sensor networks to address the problem of localization of a vast network on sensors when only a small number of anchors are available. Given the limited communication range of the anchors, they are not able to reach every sensor in the deployed network. More importantly, not every sensor has access to the required number of anchors for localization. Multi-hop localization has two main strains: those that uses previously localized nodes to localize other nodes further away from the anchors such as iterative localization [114, 4, 76]; and those that uses hop-based distances to anchors such as DV-Distance and DV-HOP [96, 97]. Other forms of multi-hop localization include collaborative multilateration [115] and coordinate stitching localization algorithms [19, 117, 86]. Collaborative multilateration is a centralized method where distance measurements are collected from throughout the network and are solved as an over-constrained system of equations for multiple-hops.

2.1.3.1 Iterative Localization

The key idea behind iterative localization is that recently localized non-anchor nodes can act as pseudo-anchors and help other nodes in their vicinities to localize, which would not have been possible if only anchors are used.

Fig.2.3 shows how iterative localization works in a number of steps. Node N cannot be localized even in the presence of 12 anchors since their communication range is not sufficient to reach N. Intermediate nodes I_1 – I_7 localize themselves recursively and thereafter N is localized. We can note two issues

with iterative localization. First, it requires a large number of anchors dispersed in the network. Second, the localization error of the intermediate nodes I_1 – I_7 can itself influence the localization accuracy of N.

In order to tackle the issue of error propagation in iterative localization, a number of solutions have been proposed. Liu et al. [76] propose a reliable least squares (RLS) pre-conditioner method to mitigate error in successive iterations.

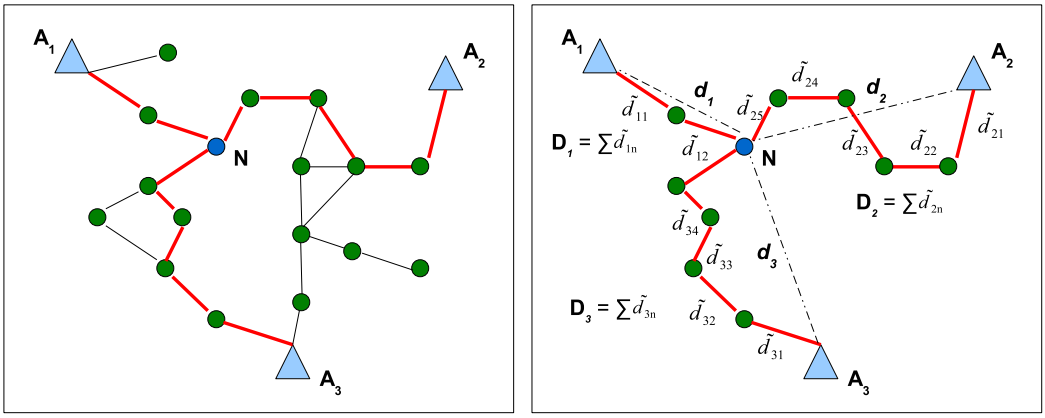
$$\hat{x_t} = \arg\min_{x} E\|(A + \Delta A)x - (b + \Delta b)\|^2 \tag{2.4}$$

Here, $\hat{x_t}$ is the estimated position, A and b are matrices computed from the linear least squares formulation (given in Eqn. 2.2) and ΔA and Δb denote the errors in matrices A and b respectively. The authors formulate an error metric obtained from the anchor error variances (represented recursively by the same metric) and Gaussian distance noise characteristics. The proposed technique is shown to outperform MDS-MAP and semi-definite programming for a variety of network topologies. However, the technique requires that distance errors be Gaussian — this will not work, for example, in case of the non-Gaussian distance errors found in cluttered environments. Albowicz et al. [4] propose a technique using least square (LS) residuals as an error metric. The residual of an estimated position x_t is given by

$$\sum_i (\sqrt{\|\hat{x_t} - x_i\|} - d_i) \tag{2.5}$$

where $\hat{x_t}$ is the estimated position, x_i is the location of the i^{th} anchor (or pseudo-anchor) and d_i is the distance estimate to the i^{th} anchor. The residual metric does not make any assumption regarding the underlying distribution of the distance error, and is shown to perform well with non-Gaussian distance errors especially when they are of large magnitude. However, it works only when the distances with non-Gaussian errors form a minority of all available distances.

Both of the above techniques use an error metric to control the iterative localization process. Typically in iterative localization, a localized node would not try to re-localize again after discovering new anchors/pseudo-anchors. If iterative localization itself is modified to allow re-localizations of previously localized nodes, it would lead to a vicious cycle of re-localizations among neighbouring nodes as a re-localized node would first cause its neighbour to re-localize (on virtue of the node broadcasting its new position to its neighbour) and then



(a) Node N using shortest-path multi-hop distances to anchors A_1, A_2 and A_3 to localize itself.

(b) Overestimating nature of the multi-hop distances $\sum_j d_{ij}$ when compared to the true distances d_i.

Figure 2.4: DV-Distance in action, where nodes estimates distances to anchors based on *shortest* multi-hop distances between them.

again re-localizing itself on receiving a new position from its neighbour. One can limit this behavior to an extent by introducing a threshold for change in the metric for accepting a new (re-localized) position. However this is found to be ineffective when the distance measurements are noisy and the anchor geometry is poor.

2.1.3.2 DV-Distance

Next, we look at DV-Distance [97], a multi-hop localization algorithm that uses multi-hop distances to anchors to overcome the paucity of anchors in the network. DV-Distance is inspired by well-known concepts: distance vector (DV) routing [103], in that localization information is forwarded hop-by-hop independently for each anchor, and anchor-based localization, such as GPS [50], where the position is calculated from the information obtained about a certain number of anchors. Anchors broadcast their information to their neighbouring (unlocalized) nodes. These neighbours will record the anchor information, namely the anchor positions and the distance to the anchors. Thereafter, these nodes will broadcast this information in their own vicinities. The nodes that receive these advertisements estimate their distance to an anchor as the sum of their distance to the advertising node and that node's distance to the anchor. This process continues until all nodes in the network estimate their distances to the available anchors. In order to restrict anchor advertisements being forwarded indefinitely in the network, each advertisement has a time-to-live (TTL) field

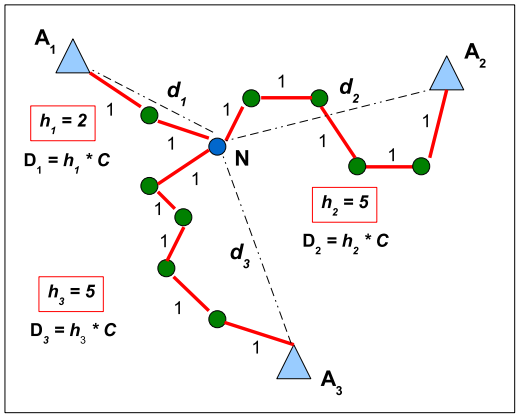


Figure 2.5: DV-HOP in action. Node N uses hop-count correction c to correct its hop-count based distances to actual distances.

which specifies the number of times an advertisement can be forwarded before it is ignored by all nodes in the network.

Fig. 2.4 illustrates how DV-Distance works in a small sensor network enabling node N to estimate multi-hop distances to the three anchors A_1, A_2 and A_3. Here, we see that none of the nodes have direct access to even two anchors, thus are unable to localize themselves. Fig. 2.4(a) shows node N with shortest-path multi-hop distances to the three anchors. DV-Distance has a drawback of overestimating the true distances to the anchors. This is evident, in Fig. 2.4(b), for the multi-hop distances between node N and anchors A_2 and A_3.

2.1.3.3 DV-HOP

We next look at the DV-HOP [97, 112] multi-hop localization algorithm. DV-HOP is different from DV–Distance in the estimation of distances to anchors since it considers the number of hops of the multi-hop path between the anchor and a node, instead of the cumulative sum of actual distances. In other words, it is an instance of DV-Distance where all the inter-node distances in the network are replaced by one. Here too, anchors would initially flood the network with advertisements of their location information. When an unlocalized node receives an advertisement, it records the hop-count to the anchor and then rebroadcasts after incrementing the hop-count. Fig. 2.5 shows an example of DV-HOP where node N measures hop-count distances h_1, h_2 and h_3 to anchors A_1, A_2 and A_3 respectively. Meanwhile, when other anchors receive such a forwarded advertisement, they will be able to calculate the average hop distance between the two anchors by virtue of knowing the actual location of

the other anchor (as it is included in the forwarded advertisement). In practice, the average hop count is calculated taking into account all other anchors the anchor may know of:

$$c = \frac{\sum_j (\sqrt{\|x_j - x\|})}{\sum_j h_j} \tag{2.6}$$

Here, c is the hop-count correction calculated by anchor using hop-count information to other anchors, x represents the position of anchor, and h_j represents the hop-count distance between the anchor and another anchor j. The anchor subsequently broadcasts this correction in its vicinity. When non-anchor nodes receive this correction, they can now calculate the approximate distances to each anchor on converting their hop-count based distances to real distances.

Evidently DV-HOP requires that the nodes in the network are uniformly distributed and would consequently perform poorly in anisotropic networks. Niculescu et al. [98] show that the Cramer Rao Bound (CRB) for DV-HOP location estimation is inversely proportional to the fraction of anchors in the network, the node density and the hop-count distance to the anchors. Both DV-Hop and DV-Distance suffer from the large traffic of messages in the network from the propagating anchor advertisements. The benefits of both DV-Hop and DV-Distance are that the anchors can be of high quality (unlike pseudo-anchors in the case of iterative localization) and they can also be placed in predefined geometries. They also typically require fewer anchors than iterative localization for similar network topologies.

2.1.3.4 MDS-MAP

The previous two techniques, DV-Distance and DV-HOP, use multi-hop distances between only the node and the anchors in order to calculate the node's position. In contrast, the technique we will discuss in this section, namely MDS-MAP [118], uses all distances between all nodes (single-hop, if it exists, and multi-hop) to calculate the positions of all nodes in the network at once. MDS-MAP is based on the mathematical technique Multidimensional Scaling (MDS), that renders n-dimensional data into a space which has fewer than n dimensions. The initial result of MDS-MAP, which has an arbitrary rotation and translation with respect to any externally defined global coordinate system, is brought to an absolute coordinate system using anchor positions and

Procrustes analysis. Since MDS-MAP is based upon linear regression, it requires a matrix (D) containing the distances (linear dissimilarities) between every node. It is possible that the distance matrix could be sparse since every node may not be able to reach every other node in the network due to limited communication range. To remedy this, the Floyd-Warshall all-pairs shortest paths algorithm is used to populate the distance matrix, whereby the shortest multi-hop path is used to denote the distance between two nodes which are not connected directly. MDS-MAP obtains the relative positions of all nodes (say n nodes) in the network as it tries to minimize the following formulation:

$$min \sum_{i<j<n} w_{ij}(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - d_{ij})^2 \tag{2.7}$$

where (x_i, y_i) and (x_j, y_j) are the coordinate positions produced by MDS for the i^{th} and j^{th} node respectively, and w_{ij} and d_{ij} are the corresponding weight and distance between the two nodes. The distances could suffer from either multi-hop overestimate errors or single-hop NLOS errors. Because MDS-MAP has to use all distances in D, a small number of erroneous distances can affect the localization accuracy for all the nodes in the network. As we can see, both MDS-MAP and DV-Distance use actual distances, except that in the case of MDS-MAP, the localization process is centralized.

Shang et al. [117] propose a distributed version of the MDS-MAP algorithm, called MDS-MAP(P). The algorithm runs in each node in the network, creating local maps of a node's n-hop neighbourhood in a local distance matrix. MDS is then performed on each node, and the solution is stitched together with other nodes' solutions using a coordinate space alignment. The benefits of this distributed version is seen only for very large networks where the cost of computing the global map scales linearly with the number of nodes in the network ($\mathrm{O}(n)$, rather than cubically ($\mathrm{O}(n^3)$ as with the original MDS-MAP algorithm.

2.2 Range-based Localization in Clutter-Prone Environments

We have previously seen in Section 1.1, the presence of clutter in the environment, where nodes are wirelessly measuring ranges to anchors, can have a

debilitating effect on the accuracy of the estimated distances and thus on the localization accuracy. The reason is that, in the presence of clutter, ranging signals take indirect paths between the node and the anchors, leading to large overestimates in the distance estimates. Thus, the non-line-of-sight (NLOS) distance measurements pose a serious challenge for the accurate localization of a node, say a sensor robot, in cluttered environments. NLOS measurements are shown to affect angular measurements in the Angle-of-Arrival (AoA) technique [**?**] and mobile handset localization in cellular networks [**?**]. The major challenge in identifying and mitigating the effect of NLOS measurements is that they, by nature, have an undefined distribution. In the following sections, we will discuss the state-of-the-art for localization in NLOS-prone environments. We look at the two main approaches to solving the problem: elimination-based methods, where the NLOS distance measurements are detected and eliminated entirely, and mitigation-based, where the NLOS distances are incorporated such that their debilitating effect is reduced. We then discuss techniques that seek to mitigate overestimate error of multi-hop distances in multi-hop localization techniques. Finally, we look at NLOS characterization in Section 2.2.4.

2.2.1 NLOS Detection/Elimination Techniques

This approach in dealing with NLOS distances assumes that it is possible to identify NLOS distances from a given set of distances to anchors. After the NLOS distances are identified, they are discarded and the position is estimated using the remaining distances. Wylie et al. [140] propose a time-based hypothesis test for determining whether a distance measurement from a moving node is NLOS or not, assuming knowledge of LOS and NLOS variance bounds. However, this technique does not work for stationary nodes. Borras [13] provides a decision-theoretic LOS/NLOS classification framework assuming that both LOS and NLOS distances follow Gaussian distributions and that the Gaussian characteristics as well as their prior error probabilities are known. Chan et al. [21] propose a residual testing (RT) method for identifying and discarding NLOS anchors, based on identifying the density function of the residuals. It assumes that only when all distances are LOS, the density function follows a centralized Chi-Square distribution. The residual weighting (Rgwh) algorithm proposed by Chen [23] shows that the least squares residual error is a good indicator of NLOS distances when they form a minority of distances. Ward et al. [135] use studentized residuals to identify NLOS distances. It must be noted

that most of the above techniques are feasible only when NLOS distance form a minority of the available distances to anchors, as discarding most distances will render localization impossible due to insufficient anchor information.

2.2.2 NLOS Mitigation Techniques

The alternative approach to eliminating detected NLOS distances is to incorporate the NLOS distances themselves in position calculation. In fact, Qi et al. [106] and Jourdan et al. [58] show that incorporating statistical information of NLOS biases actually benefit localization accuracy. There are three broad strains of this approach: first, NLOS distances are incorporated using a weighted least squares estimator; second, NLOS distances are incorporated using constrained least squares formulations; and third, robust estimators like M-estimators [53] are used to deal with NLOS distances.

Using the weighted least squares estimator to deal with NLOS distances works well if we are able to identify NLOS distances. Gezici et al. [42] and Caffery et al. [18] propose using the variances of the distance measurements as weights for the least squares position estimation. The assumption here is that variances of NLOS distances have been found to be much larger than those of LOS distances and taking an inverse of this variance can mitigate the effect of the NLOS distance to some extent. Güvenç et al. [47] propose a solution which uses certain statistics of the received ranging signal, such as kurtosis, mean excess delay and root mean square (RMS) delay spread, to determine the weights for the LS estimator. Chen and Feng [22] propose a method that incorporates the geometry of the anchors into the maximum likelihood position estimation by adding parameters derived from the intersection points of the circles portended by the anchors (the paper considers the case with 3 anchors). It uses weights based on the variances of the distances measurements, again assuming that NLOS distances will have larger variances and hence lower weights. Chen et al. [23] propose a residual weighting algorithm (Rwgh) that takes into account the residual of the LS estimation. The Rgwh algorithm is based on the key idea that using a subset of anchors where some are NLOS anchors yields a larger residual than one that consists of only LOS anchors. The final location estimate is calculated as the weighted sum of the intermediate position estimates obtained by minimization of the residuals for anchor sets of various sizes, with the corresponding normalized residual errors serving as the weights. However, as the number of anchors increase, the computational

complexity of the Rgwh algorithm becomes prohibitive. Li et al. [73] improve upon the Rgwh algorithm by initially discarding anchors with large residuals, thus making it less computationally expensive. Venkatraman [130] proposes a method wherein scale factors for NLOS biases are calculated based on the topology of the system. While this technique has the advantage of not requiring prior LOS and NLOS information and not even requiring a large number of anchors, it can be applicable only in well-defined topologies.

The second direction in NLOS mitigation techniques is that of using constrained optimization methods. Wang et al. [134], Chen and Feng [22], Venkatesh et al. [129, 128], Nawaz et al. [91] and Kim et al. [64] are a few examples. The key idea here is that the equality, $\|x - x_i\| = \hat{d}_i^2$, where x is the unknown position, x_i is the location of the i^{th} anchor, and $\hat{d}_i$ is the measured distance, is converted to $\|x - x_i\| \leq \hat{d}_i^2$ in order to accommodate the overestimated distances. Wang et al. [134] solves this constrained least squares (CLS) algorithm using quadratic programming techniques. Venkatesh et al. [129, 128] uses NLOS distances to construct a linear feasible region for the solution. However, it assumes perfect a-priori identification of LOS and NLOS distances. Linear programming is used to estimate the location using only the LOS distances with the constraint that the solution lies with this feasible region. A few papers propose solutions by first trying to estimate the NLOS bias itself prior to position calculation. Kim et al. [64] propose an interior point optimization (IPO) method to search for the optimal location estimate in the presence of NLOS biases in distance measurements. The paper first uses the IPO method to estimate the NLOS bias in the distance measurements, and then calculates the position after mitigating the biases. Nawaz and Trigoni [91] similarly estimate the biases prior to calculating the position, using l_1 norm-minimization.

Robust estimators have been used in a number of papers [45, 74, 90, 125] to suppress NLOS distances which can be viewed as *outliers*. The M-estimator developed by Huber [53] in 1964, which has been used by Sun et al. [125], shows an improved performance over the LS estimator when NLOS distances are present. The paper shows that using bootstrapping samples [142] can lead to further gains in location accuracy. Li et al. [74] and Casas et al. [20] use the Least Median Squares (LMS) technique to mitigate the effects of NLOS distances. Nawaz and Trigoni [90] use the Least Absolute Deviation (LAD) estimator instead of the LS estimator to calculate positions in the presence of NLOS distances. Güvenç and Chong [45] suggest using Least Trimmed

Squares (LTS), which is to minimize the sum of squares for the smallest set of the residuals. The paper also suggests using the S-estimator and the MM-estimator to deal with NLOS biases.

However, all the above techniques require that *NLOS distances form the minority of all available distances to anchors*. Venkatesh et al. [129, 128] show that their technique works for the scenario where NLOS distances are the majority only when a complete identification of LOS and NLOS distances is available. Nawaz et al. [91] require that NLOS distances form a maximum of half the total distances when positivity constraints are met — the threshold is reduced to a third of all distances otherwise. Residual-based NLOS mitigation will not work if a majority of the distances are NLOS or if the number of LOS distances themselves alone are insufficient for localization.

2.2.3 Mitigation of Overestimate Error of Multi-Hop Distances

In this section we look at a body of research in wireless sensor networks tackling localization in multi-hop networks where the distances suffer from large positive biases, similar to NLOS distances. Here the distance error is due to the overestimating nature of multi-hop distances which is characterized by the network connectivity and topology. Wang et al. [131, 132] propose an upper-bound least squares position estimation technique for dealing with overestimated (multi-hop) distances in concave network topologies. The authors assume that the distance measurements between nodes will be fairly accurate themselves, i.e., they will be LOS in nature. Shang et al. [119] suggest that using only the first 4 beacons with the shortest distances will reduce the effect of multi-hop overestimated distances. Xiao et al. [141] look at DV-HOP localization in cluttered environments where the multi-hop distances to the anchors can have larges biases from the large detours around obstacles. The authors propose a solution that is based on using the minimum hop-length from a pre-computed offline table to eliminate the unreliable distances received at the sensor node. The paper assumes that the network topology is uniformly distributed. Kung et al. [66] propose a centralized robust localization technique, the snap inducing shaped residuals (SISR), for handling large non Gaussian errors in dense network topologies. The SISR technique breaks down when more than 30% of distances have large errors. Shi et al. [120], using the residual formulation

Technique	Breakdown point
Wang et al. [131, 132]	50%
Nawaz and Trigoni [91]	50% (+vtity constraints met), 33% (otherwise)
Snap inducing shaped residuals (SISR) [66]	30%
Least Median Squares (LMedS) [20]	37.5%

Table 2.1: Performance of existing techniques for NLOS and multi-hop error mitigation. The breakdown point is the maximum percentage of NLOS distances among available distances.

in [66], propose a distributed two-phase algorithm for dealing with large overestimate errors. However, all of the aforesaid work assumes that good (or LOS) distances form the majority of the distances and that the overestimated distances can be eliminated from the location estimation process or have their effects ameliorated by using robust statistics or optimization formulations.

Table 2.1 tabulates current state-of-the-art NLOS mitigation techniques from the perspective of the maximum number of NLOS distances for optimal performance. These techniques do not require explicit identification of the NLOS distances. Clearly, no technique performs well when the proportion of NLOS distances exceeds 50%.

2.2.4 NLOS Characterization

Finally, we look at current work on the characterization of NLOS bias itself. This allows us to put our own characterization of NLOS bias, presented in Chapter 7, in perspective.

We have seen that Wylie et al. [140] attempted to identify NLOS distances by comparing the variance of the distances over time to a pre-defined threshold. Gezici et al. [41] compare the distribution of ranges to a known measurement error Gaussian distribution in order to classify the signal as NLOS or not. Guvenc et al. [46] use multipath channel statistics of UWB signals in the weighted least squares position estimation. The authors use the kurtosis and the mean excess delay of the multipath channel, to determine the weights to assign for each reference. Marano et al. [79] use various characteristics of the impulse response of the received UWB signal in order to detect and even estimate the approximate NLOS bias. The paper builds a database from an UWB distance

measurement campaign and then uses a Least-Squares Support Vector Machine (LS-SVM) to classify and estimate the NLOS bias.

A number of papers require knowledge of the NLOS bias distribution and make assumptions accordingly. Jourdan et al. [58] derive the Position Error Bound (PEB) using empirically derived NLOS bias distributions. A number of papers [42, 46] assume exponential error distribution for NLOS bias for UWB signals, while [128, 91, 57] assume the NLOS bias to be uniformly distributed within a predefined range. Borras et al. [13] assume that the NLOS bias follows a Gaussian distribution while defining a decision-theoretic framework for NLOS identification. Huang et al. [52] derive the Cramer Rao Bound for localization in NLOS environments where the NLOS bias distribution is obtained through a non-parametric Gaussian kernel density method.

The effect of clutter topology on the NLOS bias has also been previously studied [133, 5]. Wang et al. [133] conclude that the NLOS bias is strongly dependent on the clutter geometry and is, moreover, frequency-dependent for severe clutter. Alsindi et al. [5] conclude that the NLOS bias follows a lognormal distribution for a variety of radios and clutter-prone environment. Furthermore, the authors show that the parameters of this lognormal distribution is dependent on the clutter environment and system bandwidth [6, 3]. In Chapter 7 we will see that the lognormal-fitted NLOS bias distribution is indeed dependent on a number of features of the clutter topology.

2.3 SLAM-based Localization in Clutter-Prone Environments

In this section, we will look at an alternative approach for localization in cluttered environments — that of Simultaneous Localization and Mapping (SLAM). SLAM is a heavily researched field in mobile robotics as it aims to make possible a truly autonomous robot. SLAM has been implemented in a variety of domains from indoor robots to outdoor, underwater, and airborne systems. In the following paragraphs, we will first look at the basic intuition behind SLAM, and discuss major approaches to solving the problem. Then we look at the state-of-the-art of SLAM research and discuss existing limitations and challenges for its application to the problem of localization in cluttered environments.

The SLAM problem simply asks if it is possible for a mobile robot to be placed at an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this environment while simultaneously determining its location within this map [37]. In probabilistic form, the SLAM problem requires the estimation of the joint posterior probability of the landmark/map features' locations and robot location (at time k) given the observations and control inputs up to and including time k, together with the initial position of the robot p_0:

$$P(p_k, m|Z_{0:k}, U_{0:k}, p_0)$$

where p_k represents the current position, $m = m_1, m_2, ...m_n$ is the set of all landmarks or map features, $Z_{0:k} = z_1, z_2, ...z_k$ is the set of all landmark observations and $U_{0:k} = u_1, u_2, ...u_k$ is the history of control inputs. The joint posterior function is obtained through a recursive formulation, using Bayes theorem, with the distribution $P(p_{k-1}, m|Z_{0:k-1}, U_{0:k-1}, p_0$ at time $k-1$, the control u_k and observation z_k. A state transition model and an observation model are defined for modelling the effect of the control input on the next robot position and of the observation respectively. The state transition model $P(p_k|p_{k-1}, u_k$ is assumed as a Markov process where the next state p_k depends only on the immediately preceding state x_{k-1} and the applied control u_k. It is also assumed to be independent of the observations as well as the current map estimate itself. The observation model $P(z_k|x_k, m)$ represents the probability of observing z_k when the robot position and landmarks are known. The SLAM algorithm is implemented in a standard two-step recursive prediction (time-update) and correction (measurement-update) process. Thus, the joint posterior $P(p_k, m|Z_{0:k}, U_{0:k}, p_0$ for the robot position p_k and map m at a time k based on all observations $Z_{0:k}$ and all control inputs $U_{0:k}$ up to and including time k is given by:

TIME-UPDATE:

$$P(x_k, m|Z_{0:k-1}, U_{0:k}, x_0) = \int P(x_k|x_{k-1}, u_k) \times P(x_{k-1}, m|Z_{0:k-1}, U_{0:k-1})$$

MEASUREMENT-UPDATE:

$$P(x_k, m|Z_{0:k}, U_{0:k}, x_0) = \frac{P(z_k|x_k, m)P(x_k, m|Z_{0:k-1}, U_{0:k}, x_0)}{P(z_k|Z_{0:k-1}, U_{0:k-1})}$$

The SLAM problem is typically solved via computational methods, such as the extended Kalman filter (EKF-S LAM) [33] and the Rao-Blackwellized particle filter (FastSLAM) [84, 85]. The EKF-SLAM method makes the assumption that both the state transition model and the observation model are corrupted by white (uncorrelated) Gaussian errors. Moreover, the EKF-SLAM method uses a linearized version for the state transition model and the observation model. This simplification can lead to inconsistencies, and even non-convergence, of the solution. The FastSLAM approach attempts to address this problem by proposing recursive Monte-Carlo sampling to represent the non-linear state transition model as well as the non-Gaussian errors. The method uses the Rao-Blackwell technique to reduce the sample space of the problem to be computationally feasible.

SLAM research has matured in recent years and the SLAM problem is considered to be solved completely. Newman et al. [95] validated the non-divergence properties of EKF-SLAM by showing a robot returning to a precise stating point autonomously in an 'explore and return' experiment for an indoor setting. Guivant et al. [43] showed in their results that SLAM can deal with high-speed vehicle motion, non flat terrains, and dynamic clutter in an outdoor environment. Besides indoor [95, 14, 24, 28] and outdoor [43, 92] settings, SLAM-based robot navigation has been evaluated for airborne [63] and underwater [94, 139] settings as well. Various forms of sensors have been used for taking observations of the landmarks/map features including laser scanners [92, 95], ultrasonic/sonar transducers [24, 94] and monocular and stereo cameras [28, 29]. Different sensing modalities have also been used, such as range only [94], bearing only [30, 15], vision only [29, 83] as well as combinations of vision and laser scanning [7, 92]. The focus of current research has been on methods enabling large scale implementations in unstructured environments, particularly situations where infrastructure-based reference positioning systems, such as GPS, are unavailable or unreliable.

SLAM-based localization approaches have the inherent advantage over range-based localization, when operating in cluttered environments, of being immune to the presence of clutter. In fact the clutter aids in the localization process by providing a rich set of landmark/map features through which a robot can estimate its position in the environment using the SLAM algorithm. Typically, SLAM does not require distances to anchors, thus is not affected by the presence of clutter. In contrast, we have seen in the previous section that

range-based localization is adversely affected by incidence of NLOS distance measurements caused by the presence of clutter in the midst of the anchors and the robot. On the other hand, SLAM has a number of disadvantages when compared to range-based localization. One of the biggest disadvantage of using SLAM when compared to range-based localization is that inaccurate measurements can lead to an eventual catastrophic failure of the SLAM algorithm. Every time the robot takes measurements, these are first associated with the existing map, and thereafter the estimations are fused into the map. Thus, an association from an inaccurate observation cannot be revised later and affects future estimations given by the SLAM algorithm even if the future observations themselves are valid and accurate themselves. In contrast, range-based localization can be made to estimate the robot position afresh for each new set of observations. SLAM-based approaches require expensive and bulky sensors to be employed, something that may not be possible to fit on a small μAUV, like the one shown in Fig. 1.2(b). Similarly, SLAM-based approaches require relatively high computational and storage capabilities for computing the complex structures and mathematical operations in both EKF-SLAM and FastSLAM. In contrast, range-based localization offers a simpler and less computationally demanding alternative.

Another issue with SLAM is that it assumes that the world could reasonably be modelled as a set of simple discrete landmarks described by simple geometric primitives such as points, lines or circles — this may not hold in complex and unstructured environments. Typically, both range and bearing (direction) measurements of the landmarks are required. This may not be possible when only one type of sensor is used — for example, ranging sensors cannot give bearings of the landmarks while single camera sensors cannot give accurate information about the distance to the landmark. An important requirement for SLAM is that of loop closure — to recognise that a certain place has been revisited by the robot (and thus should have identical measurements as the previous visit) and propagate backwards the new information gained by closing the loop. However, it is a challenge to recognise loop closure in complex environments, and for arbitrarily large loops, the computational costs also are arbitrarily large, since the increased uncertainty increases the search-space for the location recognition [93, 10]. On the other hand, range-based localization is immune from the representational complexity of the environment, though it is affected by the presence of clutter in the direct view of the anchors.

2.4 Discussion

In this chapter, we looked at single-hop and multi-hop localization algorithms, in particular those methods that involve measuring distances to anchors. Existing approaches to identifying and mitigating NLOS errors are limited in that they assume NLOS distances form a minority of all distances to anchors. Techniques that do consider scenarios with majority NLOS distances require explicit classification of distances as LOS or NLOS. SLAM-based approaches in cluttered environments suffer from issues of loop closure, the need for expensive/bulky hardware, and the debilitating effect of erroneous measurements.

In the next chapter, we will evaluate various multi-hop localization techniques in cluttered NLOS-prone environments, and compare their performances with each other and with single-hop localization.

Chapter 3

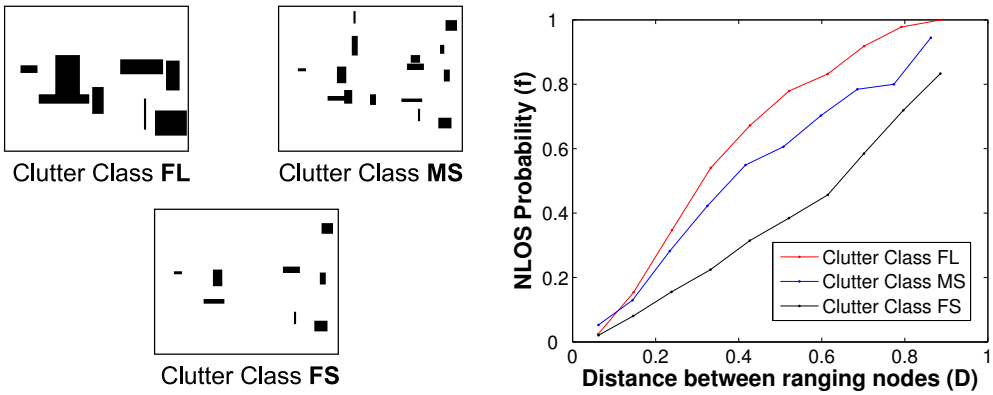
Multi-Hop Localization for NLOS-Prone Settings

In this chapter we investigate the application of multi-hop localization algorithms, described in Section 2.1.3, in cluttered NLOS-prone environments. First, we provide motivation for using multi-hop localization in cluttered environments in Section 3.1. Here, we introduce the concept of localizers, special purpose nodes that enable a target node/robot to localize with better accuracy in cluttered environments. We lay out the methodology for our evaluation in Section 3.2. We then evaluate the performance of the multi-hop localization algorithms under a variety of parameters in Section 3.3 and compare their performance to that of single-hop localization. Finally we discuss the need for localizer placement to further improve localization accuracy with practical numbers of localizers in Section 3.4.

3.1 Motivation

The primary motivation for using multi-hop localization in cluttered environments is the observation that, in such scenarios, the *actual* distance between two ranging nodes, for example an anchor and an unlocalized sensor robot, plays an significant part in determining the probability that the distance measurement is NLOS or not. In other words, the larger the distance the more likely that the distance measurement is NLOS in nature. To test this hypothesis we perform the following experiment.

We take random pairs of source and destination nodes in the clutter topologies and check if they are occluded from each other by obstacles. If they are occluded, we mark the resulting distance measurement as non-line-of-sight



(a) Clutter topology classes samples

(b) Relationship between ranging distance and NLOS probability

Figure 3.1: Relationship between NLOS distance probability and the actual distance between the ranging nodes. 100,000 node pairs were considered for each clutter class sample. D is the diameter of the enclosure.

(NLOS), else it is marked as line-of-sight (LOS). To evaluate the hypothesis in a variety of clutter topologies, we define three classes of clutter topologies:

1. **FL class**: This class is defined by *few* clutter objects of *large* size. The dimensions of the clutter objects are typically up to 40% of the enclosure dimensions.

2. **MS class**: This class is defined by *many* clutter objects of *small* size. The dimensions of the clutter objects are typically up to 15% of the enclosure dimensions.

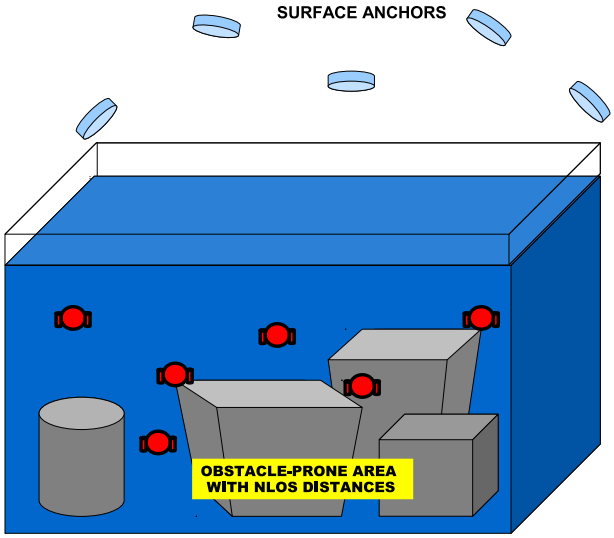
3. **FS class**: This class is defined by *few* clutter objects of *small* size. The dimensions of the clutter objects are typically up to 15% of the enclosure dimensions.

We will use this classification of clutter topologies extensively in this dissertation. We consider a sample from each of these three clutter classes as shown in Fig. 3.1(a). Fig. 3.1(b) shows the relationships between the NLOS probability and the actual distance between the source and destination nodes. We make two observations: first, for all clutter topology classes, the probability of a distance measurement being NLOS increases with the actual distance between the ranging nodes, and, second, different clutter classes vary significantly with respect to the NLOS probability. Class FL shows the largest rate of increase in NLOS probability reaching 50% NLOS probability for 0.28D (where D is the

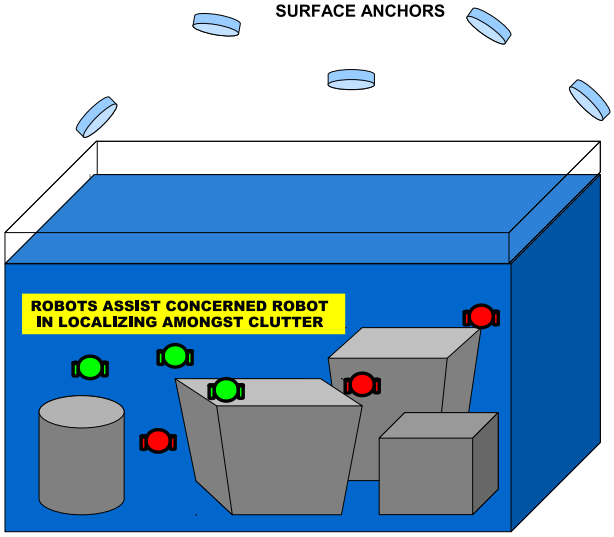
length of the diagonal of the enclosure) and achieving 100% NLOS probability when the actual distances between ranging nodes are greater than 0.85D. In contrast, in class FS, only distances greater than 0.66D have a 50% chance of being NLOS. To conclude, in a cluttered environment, we are more likely to get a NLOS distance if the ranging nodes are far apart, and there are many large-sized clutter objects.

In order to reduce the probability of a distance being NLOS in nature, thereby avoiding large localization errors, we need to reduce the actual distances between the ranging nodes. Since a robot exploring a cluttered environment needs to travel deeper into the environment in order to detect and report interesting phenomena, it is not possible to constrain the distance between the robot and the anchors without interfering with the exploration/sensing task. To tackle this situation, we introduce intermediate nodes, called localizers, between the robot and anchors, reducing the inter-node distances and thus reducing the probability of NLOS distance occurring. We can envision two scenarios where localizers can be employed. In the first scenario, mobile robots could be exploring a cluttered environment with only coarse-grained localization accuracy. Suppose one of the robots detects an unusual environmental phenomenon, such as high radioactivity or high temperature reading, and needs to be localized accurately to report the location of the reading, as shown in Fig. 3.2(a). Since numerous robots are exploring the environment at the same time, it is preferred to dynamically assign a neighbouring robot of the target robot as localizers in order to help it estimate its current location, as shown in Fig. 3.2(b). After the target robot reports its location, it, as well as the localizers, can continue with its previous exploration task. An alternative scenario involves a single robot that needs to traverse a pre-determined trajectory in a cluttered environment. For localization and navigation purposes, it is equipped with inertial sensors in addition to sensors for range-based localization. The inertial readings allow the robot to traverse only a short distance before the drift becomes large, thus necessitating position estimation via range-based localization at regular intervals to complement and correct the inertial readings. Unlike the first scenario, because the target robot needs to travel a trajectory, it might be necessary to assign dedicated localizers along its path. The target robot will henceforth be called the *sensor robot*.

The reasons for using multi-hop localization in cluttered environments are different to those in open-space uncluttered environments. In the latter, as



(a) Robot may detect an unusual event, but clutter may render localization difficult.



(b) Neighbouring robots can be used to assist the target robot to localize with high accuracy.

Figure 3.2: Motivation for using localizers for assisting in localization of robots in cluttered environments.

was previously discussed in Section 2.1.3, multi-hop localization was used to enable a large number of wireless sensor nodes to estimate their positions from a very small set of anchors. The limited communication range of the anchors prevented the wireless sensor nodes from being directly localized via one-hop range measurements to the anchors. In cluttered environments, however, the reason for using multi-hop localization is not the limited anchor communication range, but the large number of NLOS distances. It is possible for all distances

to anchors to be NLOS in nature. Our claim is that multi-hop localization offers a better alternative to direct single-hop localization in NLOS-prone situations. The goals of this chapter are to:

1. Show that multi-hop localization offers higher localizer accuracy than single-hop localization for various classes of clutter topologies.

2. Compare the performance of various multi-hop localization algorithms in different scenarios as we vary the communication range, number of localizers, anchor placements and clutter topologies.

3. Motivate the need for dynamic placement of localizers to further improve the accuracy of multi-hop localization algorithms.

3.2 Methodology

In this section we will describe the setup of the simulation environment and parameters for evaluating existing localization algorithms in NLOS-prone environments.

3.2.1 Simulation Environment

ProwlerSim [121] is a wireless sensor simulator tool written in MATLAB. It is a discrete event simulator (DES) at its core. An event queue is maintained and dequeued events are presented to an event handler. The simulator allows for timers to be set up so that they fire at set time periods. Node topology can either be set statically or be updated dynamically. Communication is modelled as a series of events. For example, the process of node A sending a message to node B would produce events 'Channel_Request', 'Packet_Transmit_Start', 'Packet_Transmit_End' and 'Packet_Sent' at the sender, and events 'Packet_Receive_Start', 'Packet_Receive_End' and 'Packet_Received' at the receiver. Events are classified as radio level events or application level events — with separate event handling code for each level. The main algorithmic logic is contained in the application level event handling code. For example, the above events 'Packet_Sent' and 'Packet_Received' are application level events that allow a node to take appropriate action after successfully sending a packet and successfully receiving a packet respectively. ProwlerSim allows the simulation of packet collisions, i.e., when two nodes attempt to send packets at the same

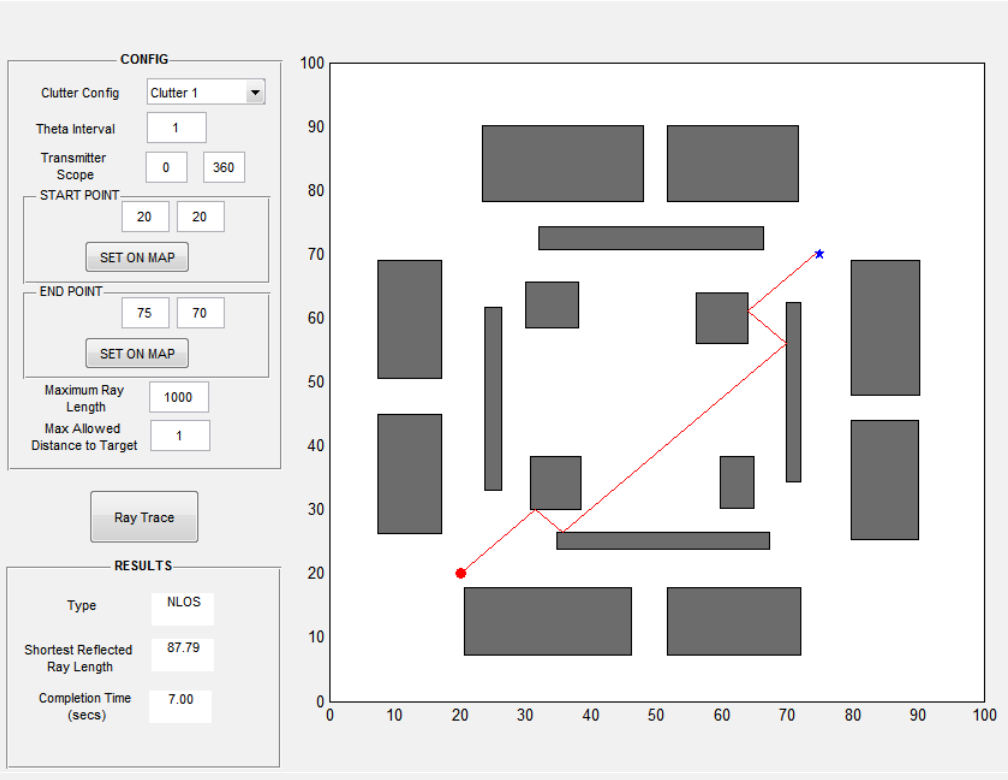


Figure 3.3: 2D raytracer simulator outputs the shortest indirect path between two points in the clutter topology when clutter obscures the direct path between the two.

time despite carrier sensing. The simulator allows for various signal propagation models to be used to simulate communication between wireless sensors such as the noisy disc model [136], the Rayleigh and Rician fading models [107], etc. Custom signal propagation models can also be incorporated. Simulation time can be bounded after which the simulation terminates irrespective of the current state of the simulation. We have developed a custom 2D ray tracer tool to simulate NLOS distances between two points, in a 2D enclosed environment, that are occluded from each other by clutter. The ray tracer works by shooting rays in various directions around the source point and traces the reflections across the clutter topology. While a number of such rays may eventually be incident on the destination point, we choose the shortest indirect ray as the NLOS distance. Note that in the case where there is no clutter between a pair of points, the distance between them is considered to be line-of-sight (LOS) and is simply the Euclidean distance between the two. Fig. 3.3 shows an example output of the ray tracing tool. Some of the key parameters of the ray tracer are:

1. **Direction Granularity (RT_{gran}).** It is the measure of the angle interval between each consecutive ray shot out from the source point. For example, if RT_{gran} is 0.1°, then rays will be shot out at angles 0°, 0.1°, 0.2°, etc.

2. **Maximum Ray Distance (RT_{mxdist}).** It denotes the maximum distance a ray can travel reflecting off edges in the clutter topology before it is deemed nonviable to continue. It can be used to represent the communication range of a ranging node.

3. **Maximum Distance from Destination (RT_{mxdest}).** It denotes the maximum distance allowed for a ray to pass close to the destination point and still be considered incident on the destination itself. For example, if we set RT_{mxdest} to 2 units, then during the ray tracing process if a ray passes within 2 units from the destination point, it is considered incident on the actual destination point.

We use a custom signal propagation model for our simulations which use the parameter RT_{mxdist} to determine reachability between two nodes in the clutter. In other words, two nodes can communicate with each other (and measure distance between each other) if the direct (LOS) or indirect (NLOS) path between them is at most RT_{mxdist} units.

3.2.2 Simulation Parameters

After describing the simulation environment, we will now summarize the various simulation parameters. We first describe the ProwlerSim-specific customizations. Thereafter we look at the various items we wish to evaluate for the multi-hop localization techniques.

For ProwlerSim simulations, we make the following customizations:

1. We disable packet collisions as we focus solely on the performance of the localization algorithms in terms of localization accuracy.

2. We keep the simulation time unbounded. Each simulation completes when there is no more messages being sent by any node.

3. We disable random packet loss.

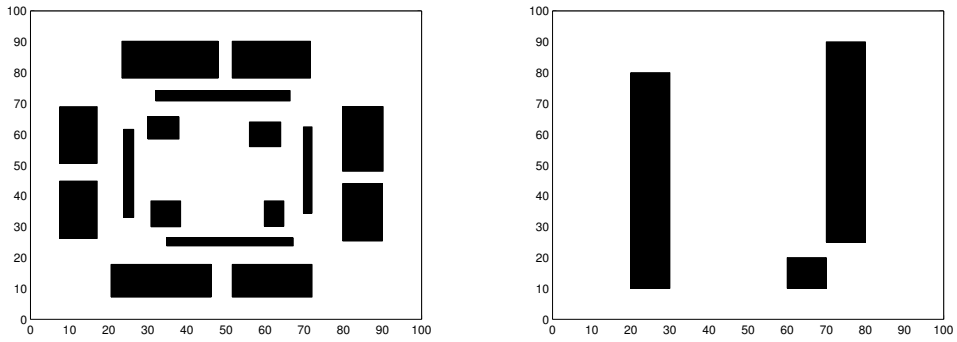
Parameter	Default	Range
Number of sensor robot samples (N_{sp})	30	-
Number of localizers (N_l)	10	[5 10 25 50]
Communication range ($\mathcal{R}$)	4D	[0.25D 0.5D 1D 2D 4D]
Anchor positions	{(10 0);(90 0);(80 100); (0 85)}	[{(10 0);(90 0);(80 100);(0 85)} {(50 0);(100 20);(90 100);(0 55)} {(80 0);(100 70);(40 100);(0 95)} {(70 0);(20 0);(50 100);(0 45)} {(30 0);(10 0);(70 100);(0 35)}]
Distance Error • LOS • NLOS	 Gaussian $\mathcal{N}(0, \sigma^2)$, $\sigma = 0$ Ray-tracer determined	
Clutter Topology	A	[A B]

Table 3.1: Parameter values for evaluation of single-hop and multi-hop localization algorithms in cluttered environments. D is the diameter of the enclosure. Clutter topologies A and B are shown in Fig. 3.4.

4. We use a custom signal propagation model where LOS distances are modelled with the noisy disc model [138] and NLOS distances are obtained using a 2D ray tracer. Under the noisy disc model, an erroneous distance estimate is obtained by sampling the Gaussian distribution $\mathcal{N}(d, \sigma^2$, where d is the true distance between the nodes and σ is the standard deviation of the Gaussian error. If $\mathcal{R}$ is the communication range, then a node can communicate and measure distances with its neighbouring nodes if:

   - Line-of-Sight exists: $d_{los} < \mathcal{R}$ where d_{los} represents the true distance between nodes.
   - Occluded by obstacles: $d_{nlos} < \mathcal{R}$ where d_{nlos} represents the length of the shortest indirect path obtained from the ray tracer.

In this chapter, we evaluate the performance of various multi-hop localization techniques under varying parameters, primarily, numbers of localizers, anchor placements, communication range and clutter topologies. Table 3.1 gives



(a) Clutter Topology A (b) Clutter Topology B

Figure 3.4: Clutter topologies used for the simulations. Clutter Topology A is representative of the MS class while Clutter Topology B represents the FL clutter class.

the ranges and the default values for the various parameters. For each distinct configuration of the parameters we evaluate the performance of the localization techniques on N_{sp} random positions of the sensor robot. The performance metrics are, thus, an average of N_{sp}(=30) values. The number of localizers ranges from 5 to 50. Again, localizers are assigned random positions in the clutter topology. Anchor placements are also varied with five different configurations. Fig. 3.4 shows the two clutter topologies used for our experiments. Clutter Topology A is representative of the MS clutter class and is the default clutter topology, while Clutter Topology B represents the FL clutter class. We use only square shapes for clutter pieces as well as the enclosing boundaries, though one can extend this work to circular shapes with some modifications to the ray tracer described in Section 3.2.1.

3.2.3 Algorithms

Next we list the localization algorithms we use for our evaluations. The multi-hop localization algorithms were introduced and discussed in detail in Section 2.1.3 of the previous chapter. Note, that the DV-HOP algorithm has been left out from our evaluation for reasons that will be discussed later in Section 3.4.

1. Single-hop localization. Here, only direct distance measurements to anchors are used. The linear least squares estimation (LLSE), outlined at the beginning of Chapter 2, is used for calculating the position.

2. Iterative localization. The iterative localization algorithm with reliable least squares (RLS) based error management [76] is used here. Localizers localize themselves and behave like pseudo-anchors for surrounding nodes (localizers and sensor robot). The error metric is also used to rank the quality of pseudo-anchors that a node has access to (anchors have zero error). Only the top four are used for localization.

3. DV-Distance. The DV-Distance [97] algorithm is used primarily for position estimation at the sensor robot. The localizers need not estimate their own positions in this scenario as they are employed only to assist the sensor robot to localize by forming multi-hop distances between the anchors and the sensor robot.

4. MDS-MAP. The MDS-MAP [118] algorithm is employed, in a centralized manner at the sensor robot, for the set of anchors, localizers and sensor robot. Each node measures its distances to all its neighbours in its communication range. Thereafter, all nodes (except the sensor robot) send their neighbour list and respective distances to the sensor robot. The anchors, besides sending their neighbour information, also send their respective positions to the sensor robot. The sensor robot then uses the MDS-MAP algorithm to calculate its position. Since the single-hop distances can have NLOS biases that are comparable to multi-hop overestimates errors, we set the weight w in the MDS-MAP estimation as 1 for all edges.

3.2.4 Performance Metrics

We use the following metrics to measure the performance of the localization algorithms:

1. Localization Error (LE). This is the average of the localization error over all sensor robot positions used in a simulation. The error in each position is measured as the Euclidean distance between the estimated position and the true position:

$$LE = \frac{1}{N_{sp}} \sum_{i=1}^{N_{sp}} \|\tilde{p_i} - p_i\|$$

where N_{sp} is the number of sensor robot sample positions, $\tilde{p_i}$ is the estimated position at the i^{th} sensor robot position, and p_i is the actual i^{th} sensor robot position.

2. Distance Error (DE). This is the average distance error over all the sensor robot positions evaluated in the simulation:

$$de_i = \frac{1}{N_a^i} \sum_{j=1}^{N_a^i} |\tilde{d_{ij}} - d_{ij}|$$

$$DE = \frac{1}{N_{sp}} \sum_{i=1}^{N_{sp}} de_i$$

where N_a^i is the number of anchors used for localizing the i^{th} sensor robot sample, $\tilde{d_{ij}}$ and d_{ij} represent the estimated and actual distances between the i^{th} sensor robot position and the j^{th} anchor respectively. The calculation of the distance error varies according to the localization algorithm used:

Single-hop localization. The distance error is calculated as the absolute difference between the true distance, between the sensor robot position and the anchor, and the measured distance, which could be either LOS or NLOS in nature.

Iterative localization. The distance error is calculated as in single-hop localization, but it is caused by distance anomalies between sensor robot positions and localizers that are used as pseudo-anchors.

DV-Distance. The distance error is calculated as the difference between the multi-hop cumulative distance of the localizer chain between the anchor and the sensor robot and the true distance between the two. Note that the distance can be single-hop LOS in nature, in which case the corresponding distance error is calculated as simply the LOS error (a multi-hop localizer chain is highly unlikely to exist between the anchor and the sensor robot in case there is no obstacle between the two).

MDS-MAP. In the case of MDS-MAP, unlike the previous three algorithms, the distance error is calculated as the (absolute) difference between the estimated and the actual distance between every pair of nodes of the connected graph formed by vertices consisting of anchors, localizers

and the sensor robot. Here, the distance can be LOS or NLOS in nature:

$$de_i = \frac{1}{N_{ng}} \sum_{j=1}^{N_{ng}} |\tilde{d}_i - d_i|$$

$$DE = \frac{1}{N_{sp}} \sum_{i=1}^{N_{sp}} de_i$$

where N_{ng} is the number of unique edges in the connected distance graph formulated for localizing the i^{th} sensor robot position, and $\tilde{d}_j$ and d_j represent the estimated and true distances for the j^{th} edge respectively.

3. Percentage of localized nodes/Localization success rate (SR_{loc}). This represents the fraction of sensor robot positions which were able to successfully localize using the given localization algorithm.

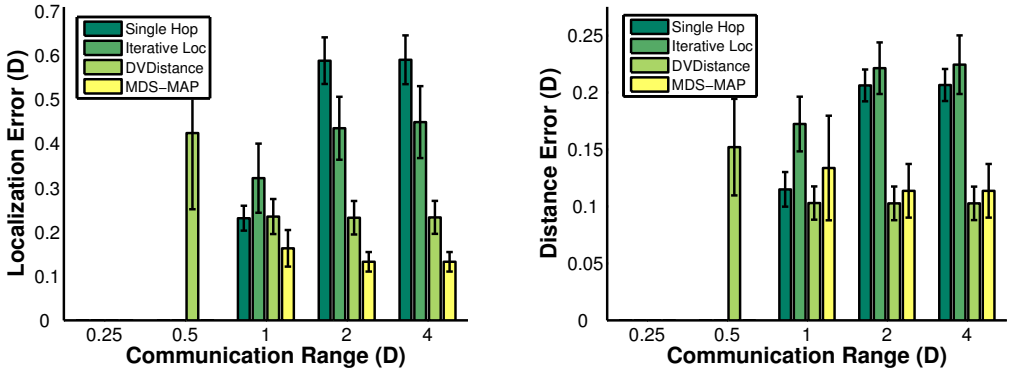
4. Packet traffic (T_{pkt}). This represents the total number of packets sent and received by all nodes in the simulation. This can be used to ascertain the energy requirements of a localization technique since communication has significant energy requirements and can quickly deplete the battery power of wireless sensors/robots. The metric can also be used together with localization accuracy to understand the performance tradeoffs of a localization algorithm.

3.3 Evaluation

Having laid out the methodology for our simulation-based evaluation in this chapter, we can now look at the simulation results. We first look at the effect of communication range on the localization error, assuming that both localizers and anchors have the same communication range. We then look at varying the number of localizers in the cluttered area and understand its effect on the localization accuracy of the multi-hop localization algorithms. Finally, we evaluate the influence of anchor placements and clutter topology on the localization error.

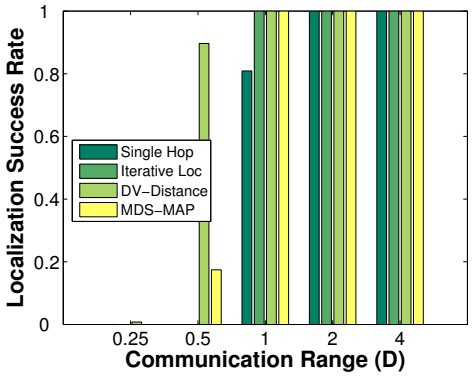
3.3.1 Effect of Communication Range

Communication range ($\mathcal{R}$) is a vital parameter for localization. Unlocalized nodes require access to a number of anchors proportional to the dimensionality, in order to localize themselves. In the ideal case, the communication



(a) Localization Error

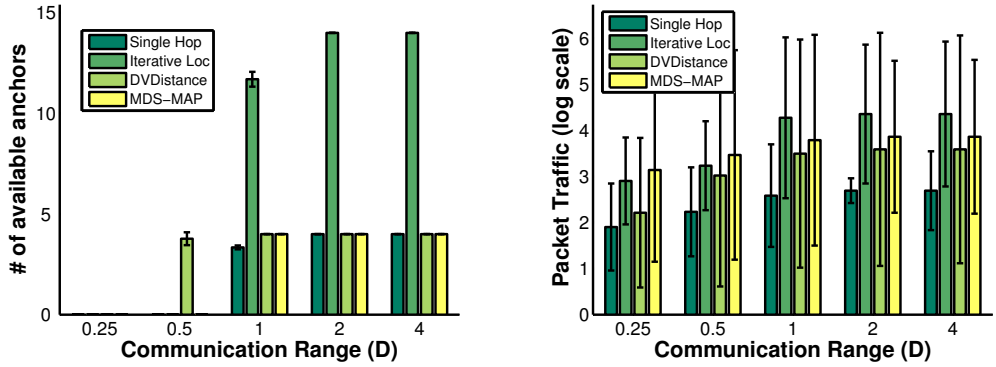
(b) Distance Error



(c) Localization Success Rate (1 means 100% success)

Figure 3.5: Effect of communication range ($\mathcal{R}$) on localization in cluttered environments. D is the diameter of the enclosure.

range of the anchors should be large enough to reach all unlocalized nodes. However this is not the case due to several reasons: wireless communication bandwidth may be limited, power consumption for large communication ranges may quickly deplete the anchors power supply, etc. In the case of cluttered environments, increased communication range increases the incidence of NLOS distance measurements at the unlocalized nodes. As we showed in Section 3.1, the probability of NLOS distances in a cluttered environment increases with the distance between two ranging nodes. However a small communication range may make the anchors inaccessible to the sensor robot in the midst of clutter. In the clutter-free scenario, multi-hop localization techniques help ameliorate the situation where all nodes do not have access to the required number of anchors. However, in cluttered environments, multi-hop localization can offer an alternative by providing smaller distances between nodes. We will now see how the communication range can affect the performance of the multi-hop



(a) Number of available anchors (b) Packet Traffic

Figure 3.6: Comparison of available anchors and packet traffic for the localization techniques. In case of iterative localization, Fig. 3.6(a) includes both anchors and pseudo-anchors.

localization techniques.

Fig. 3.5 shows the performance of the localization algorithms under varying communication ranges, represented in terms of the diagonal size (D) of the enclosed area. In the case of single-hop localization, no sensor robot position is able to localize for $\mathcal{R}$ values 0.25D and 0.5D. For $\mathcal{R} = D$, around 80% of sensor robot positions are able to localize with an average positional error of around $0.25D$. When the communication range is increased to $2D$ and $4D$, all sensor points are able to localize but with a higher average error of approximately 0.6D. Since all sensor robot positions have access to four anchors at $\mathcal{R}$ values of $2D$ and $4D$, with at least one anchor distance being NLOS, the resulting localization error is also large. The error at $\mathcal{R} =$ D is comparatively less than at $\mathcal{R} = 2D$, due to the fact that only a smaller number of sensor points are able to localize with a correspondingly smaller number of anchors, as shown in Fig. 3.6(a).

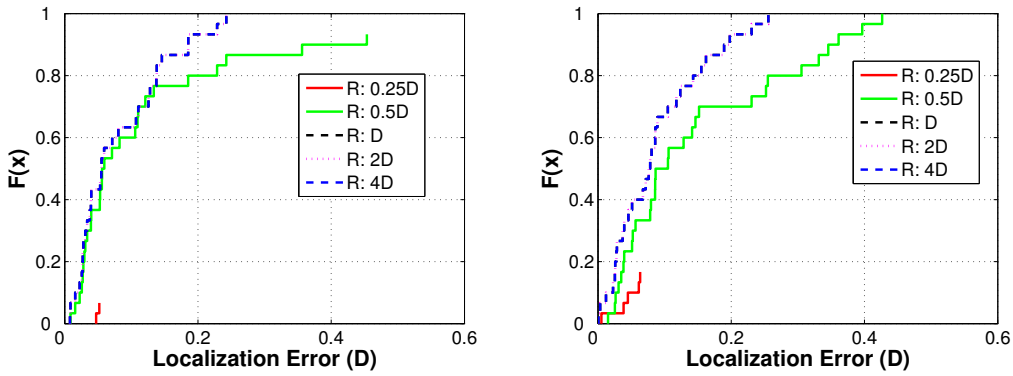
Next, we evaluate the performance of iterative localization. Unlike linear least-squares estimation (LLSE) used for single-hop localization, the employed version of iterative localization uses a regularization technique to handle pseudo-anchor and distance errors. Fig. 3.5(a) shows that no sensor robot position is able to localize for $\mathcal{R}$ values 0.25D and 0.5D as none has access to even three anchors (or pseudo-anchors). When $\mathcal{R}$ is D, all sensor robot positions can localize, but with an average error of 0.34D and a large variance. Compare this to single-hop localization and we see that the localizers helped all the sensor robot positions localize while only 80% could do so in the case where none were

used. When $\mathcal{R}$ is 2D and 4D, as also seen in the case of single-hop localization, all nodes are able to localize, albeit with increased localization error on average. We observe that iterative localization is able to obtain substantially better localization accuracy than single-hop localization despite having (slightly) larger distance errors. This is due to the regularization-based position estimation technique. We also see that the error metric does not necessarily choose references whose distance error is small since distance error is dependent only on the presence of clutter — something the metric cannot indicate. Iterative localization also incurs large packet traffic as nodes re-localize repeatedly.

In the case of DV-Distance, we find that increasing the anchor communication range has a more complex effect. We see that even for $\mathcal{R} = 0.5$D, 92% of the sensor points are able to localize, when none were able to in the case of single-hop localization and iterative localization. The reason is that, since DV-Distance is able to form multi-hop paths between the anchors and the sensor robot positions (with the help of localizers), anchors cannot be directly ranged at a large number of sensor robot positions. Since the individual links of these multi-hop paths are smaller in length and can be reached with the communication range of only 0.5D, DV-Distance is able to localize more sensor points than the previous two techniques. Since DV-Distance uses the shortest paths (single-hop or multi-hop) between the anchors and the sensor robot positions, the distance error is much smaller than that with single-hop NLOS distances. The localization error with DV-Distance decreases as $\mathcal{R}$ is increased from 0.5D to D. This is due to the fact that as $\mathcal{R}$ increases, the shortest path algorithm in DV-Distance is able to find tighter multi-hop chains along the clutter between the anchor and the sensor robot positions. However, beyond $\mathcal{R} = $ D, it actually remains the same as the shortest path distances cannot be further improved upon.

To get a closer look at the relationship between communication range and localization error in DV-Distance, we look at the empirical distributions of the localization error for two runs of DV-Distance in Fig. 3.7. We can see in both cases that the localization error decreases as the communication range $\mathcal{R}$ is increased since DV-Distance is able to find shorter paths along the clutter between the anchors and the sensor robot positions for large values of $\mathcal{R}$.

Finally, we look at the performance of MDS-MAP with increasing communication ranges in Fig. 3.5. We see that MDS-MAP performs substantially better than all the competing localization techniques we use in our evaluation. It is



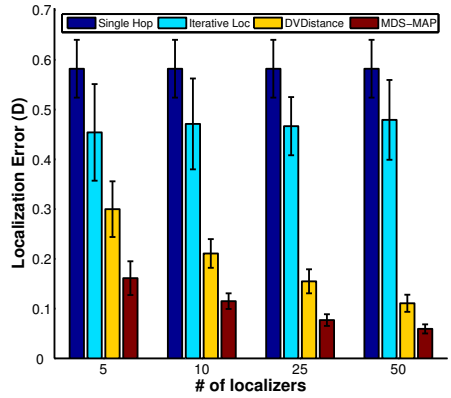
(a) Sample run. (b) Sample run.

Figure 3.7: Empirical distributions of DV-Distance localization errors of DV-Distance for two samples runs.

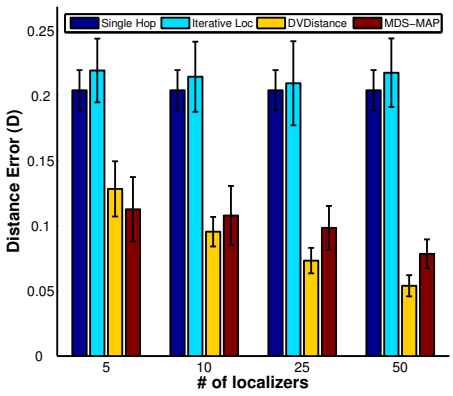
superior to DV-Distance by almost half. MDS-MAP performs better than DV-Distance since it uses more information about the network than DV-Distance. The information used by DV-Distance, namely the shortest paths between the anchors and the sensor robot, is already available in the distance matrix used by MDS-MAP. In addition to this, MDS-MAP uses information about distances between other localizers (and anchors) to constrain the positions of the sensor robot (as well as the localizers' positions). MDS-MAP incurs larger distance error, when compared to DV-Distance, since it uses distances between all nodes in the network. Despite this, it performs better than DV-Distance in terms of localization accuracy. In Fig. 3.5(b), in contrast to DV-Distance where the distance error remains the same for $\mathcal{R} \geq$ D, the distance error in MDS-MAP actually decreases when $\mathcal{R}$ increases from D to 2D. This is due to the fact that as $\mathcal{R}$ increases, more direct distances are possible between nodes (localizers and sensor robot) and some of these distances, despite being NLOS, may incur less error than the multi-hop distances used earlier. However, this advantage in localization accuracy comes at the cost of increased packet traffic, as shown in Fig. 3.6(b), as advertising nodes can reach a larger number of neighbouring nodes.

3.3.2 Effect of Number of Localizers

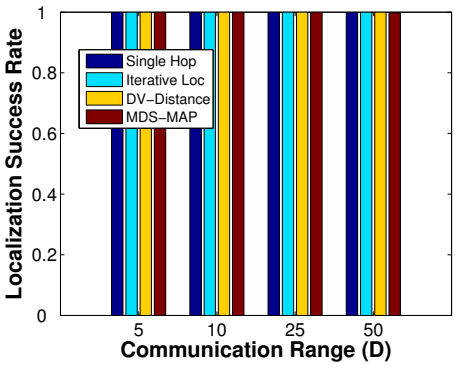
Next, we analyze the effect of the number of localizers on the localization accuracy of iterative localization, DV-Distance and MDS-MAP. We fix a random topology of sensor points and increase the number of localizers from 5 to 50.



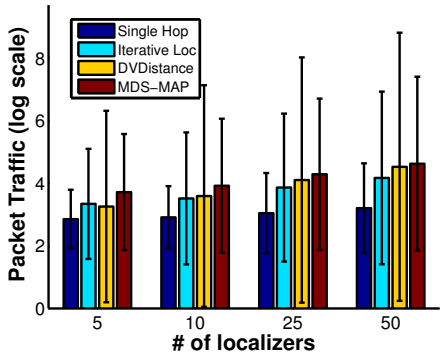
(a) Localization Error



(b) Distance Error



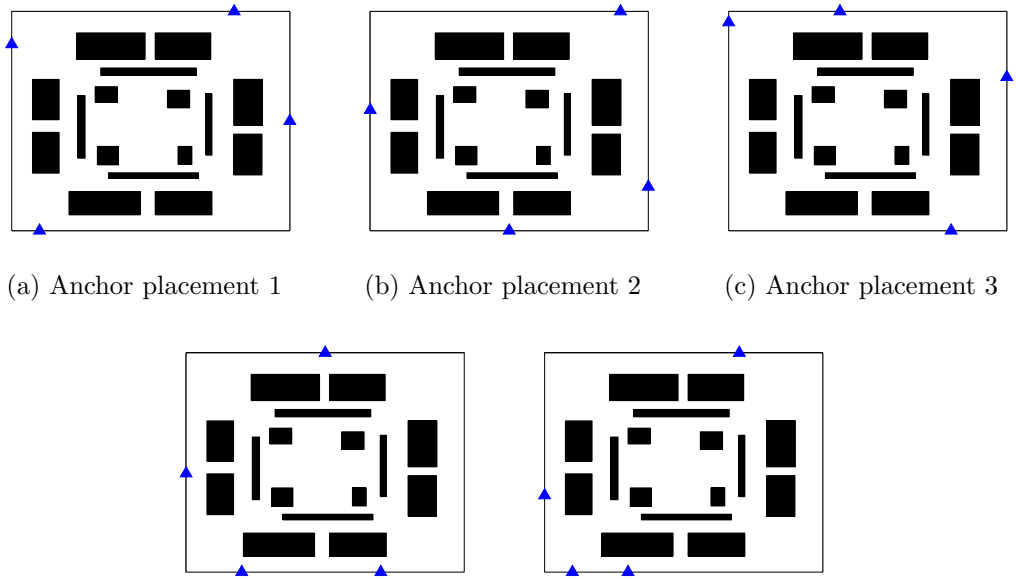
(c) Localization Success Rate (1 means 100% success)



(d) Packet Traffic

Figure 3.8: Effect of number of localizers on multi-hop localization in cluttered environments.

We repeat the simulation for N_{sp}(=30) distinct randomly generated topologies of sensor points. Fig. 3.8(a) shows the DV-Distance benefits from having more localizers in the environment, so much that the average localization error is reduced four-fold from 0.3D with only 5 localizers, to 0.077D for 50 localizers. When compared to DV-Distance, iterative localization remains at a very high localization error of 0.45D. Despite the availability of more references (anchors and pseudo-anchors) than just four anchors, NLOS distance errors are significant as shown in Fig. 3.8(b). MDS-MAP and DV-Distance perform well due to their use of multi-hop distances in lieu of NLOS distances with large biases. Again, MDS-MAP performs slightly better than DV-Distance, despite having a larger distance error in the process achieving a localization error of only 0.06D for 50 localizers. However, both techniques incur high packet traffic as the number of localizers increase, with MDS-MAP being worse than DV-Distance



(a) Anchor placement 1 (b) Anchor placement 2 (c) Anchor placement 3

(d) Anchor placement 4 (e) Anchor placement 5

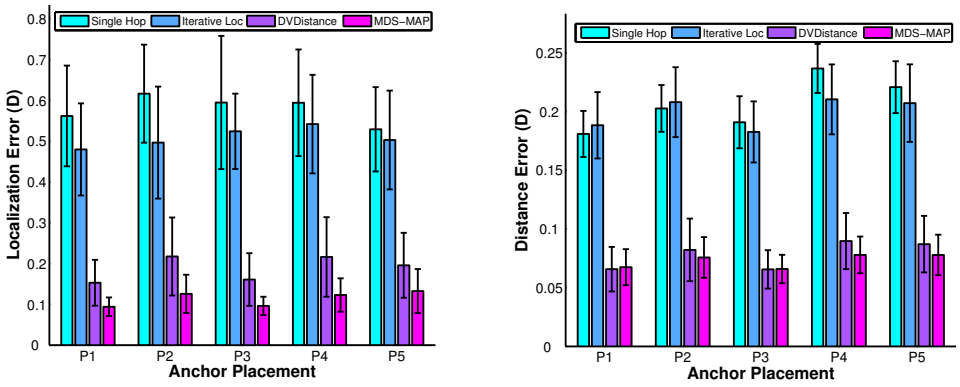
Figure 3.9: Anchor placement configurations used for multi-hop localization evaluation.

in this respect, as shown in Fig. 3.8(d).

3.3.3 Effect of Anchor Placements and Clutter Topology

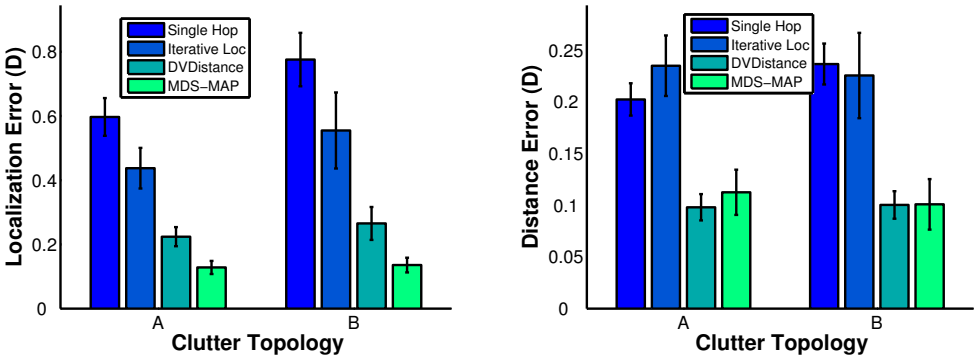
Finally, in this section we study the effect of anchor placements and clutter topology on the localization error of the localization algorithms. Analyzing localization error in cluttered environments is complex given the arbitrary nature of clutter topologies. The bias in NLOS distance measurement is ultimately determined by the actual placement of the anchor and the sensor robot in the midst of the clutter, assuming a sufficiently large communication range for the anchor.

First, we consider the variation of anchor placement for a given clutter topology. A random topology of sensor points and localizers is fixed and the four anchors are placed in five distinct anchor placements as shown in Fig. 3.9. We repeat the simulation for $N_{sp}(= 30)$ distinct randomly generated sensor robot and localizers positions. The results are shown in Fig. 3.10. We see that the localization error varies for all four localization techniques. In the case of single-hop and iterative localization, the variation can be ascribed to the changing NLOS distances due to different anchor placements. In the case



(a) Localization Error (b) Distance Error

Figure 3.10: Effect of anchor placement on localization algorithms in cluttered environments.



(a) Localization Error (b) Distance Error

Figure 3.11: Effect of clutter topology on multi-hop localization in cluttered environments.

of DV-Distance, it is due to the variation in the cumulative distances of the multi-hop localizer chains between the anchors and sensor robot positions. In the case of MDS-MAP, the variation is not only due to the change in the multi-hop distances between anchors and the sensor robot positions, but between anchors and the localizers as well. We find that DV-Distance and MDS-MAP consistently outperform single-hop localization and iterative localization for all anchor placements.

Next, we look at the effect of the clutter topology on the performance of the multi-hop localization algorithms. We use the two clutter topologies shown in Fig. 3.4 for our evaluation. For each of the N_{sp} runs, we choose a random topology of localizer and sensor robot positions that are the same for both

clutter topologies. We find that the clutter topology does have an impact on the localization error of all algorithms. However, the main pattern observed before, i.e., that DV-Distance and MDS-MAP significantly outperform single-hop and iterative localization, is repeated consistently here for both clutter topologies.

3.4 Discussion

In this chapter, we have evaluated the performance of various multi-hop localization algorithms in NLOS-prone conditions. We observed that localization in such settings is a complex problem, and its accuracy depends on factors such as communication range, clutter topology and anchor placements. In the case of multi-hop localization, the localizer density also plays a key role in localization accuracy. The communication range affects the performance of the various localization techniques in distinct ways. We found that single-hop localization suffers from the incidence of NLOS distances. Iterative localization with error management performs better than single-hop localization with the help of localizers, but, again, suffers from its inability to differentiate between LOS and NLOS distances. Furthermore, we can conclude that the error metric proposed by Liu et al. [76] is not a good indicator of position error in NLOS conditions. DV-Distance and MDS-MAP perform much better than the previous two techniques, even for smaller values of communication range. When clutter exists between two nodes, these algorithms use shorter multi-hop distances between nodes in lieu of the NLOS distances with larger errors. As expected, MDS-MAP performs better than DV-Distance since it uses more information about the network for embedding the position of the sensor robot. Increasing the communication range, in fact improves the localization accuracy since tighter multi-hop path chains (hence smaller distance error) are formed around the obstacles. Similarly, the number of (randomly deployed) localizers in the cluttered environment affects the performance of the multi-hop localization techniques, particularly DV-Distance and MDS-MAP. The reason for this is that as the localizer density increases, shorter/tighter multi-hop localizer chains can be formed around the clutter.

For the scenarios we considered in our evaluation, DV-HOP will not work properly since the communication range is large enough to reach all nodes in the environment. In DV-HOP, nodes accept those multi-hop paths to anchors

that offer the smallest number of hops — in the case of full connectivity, this reduces to a single hop! DV-Distance, on the other hand, uses actual distances and thus a node is able to accept a multi-hop path in lieu of a direct path to an anchor, provided the cumulative multi-hop distance is smaller than the single-hop NLOS distance.

Finally, the question we would like to ask at the end of this chapter is that whether it is possible to further improve on the performances of the multi-hop localization algorithms DV-Distance and MDS-MAP with a practical number of localizers? While they both show superior performance in cluttered environments, they do so in the presence of an impractical number of (randomly placed) localizers in the environment. We can foresee a reduction in the number of localizers if we can only replace random placements with carefully selected localizer placements. But what do we need to optimize in order to determine these placements? The answer lies in the multi-hop distances themselves. Since multi-hop distances are overestimates of the true distances, we can assume that reducing the multi-hop distances to the minimum possible value actually minimizes the distance error. The multi-hop distance between an anchor and a sensor robot can be reduced by varying the placement of the localizers. In case clutter exists between an anchor and a sensor robot, the multi-hop localizer chain can encompass it to offer the minimum multi-hop distance between the two. In case of DV-Distance, this minimization of the multi-hop distance is straightforward as multi-hop distances are measured only between the sensor robot and each anchor. In the case of MDS-MAP, however, the localization accuracy of the sensor robot depends on minimizing the errors (multi-hop distances) between all nodes at the same time (nodes include anchors, localizers and the sensor robot). Thus DV-Distance offers a practical alternative towards optimizing the localization error in NLOS-prone situations with a relatively small and practical number of localizers.

In the next chapter we propose algorithms for the careful placement of a small number of localizers in the cluttered environment that achieve similar accuracies to random placements when much larger numbers of localizers are used. We make use of the fact that localizers are mobile robots that can be directed to move to a desired position. Instead of moving localizers to predetermined positions, which would require them to localize themselves in the cluttered environment, we propose a solution that makes localizers adjust their positions by only measuring distances to their neighbours.

Chapter 4

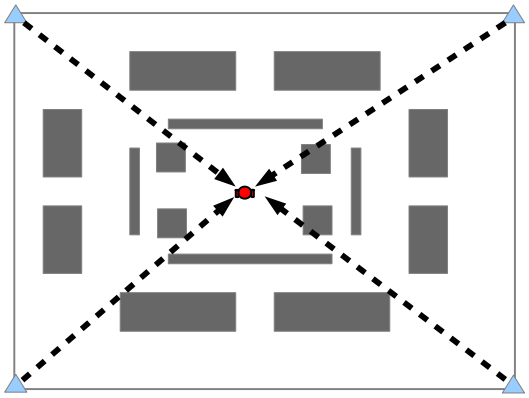
Localizer Placement in Cluttered Environments

In this chapter we introduce the concept of localizer placement in a cluttered environment to improve the localization accuracy of a single sensor robot when DV-Distance is used. We have seen in the previous chapter that increasing the localizer density improves the localization accuracy of DV-Distance. However, instead of using an impractically large number of localizers, one could instead place a relatively small number of localizers at strategic positions in the cluttered environment. To this end, we introduce two distinct localizer placement algorithms. The first one is an oracle-type algorithm which assumes the complete knowledge of the clutter topology, and assumes that localizers can be placed easily in the selected positions. The second algorithm is more practical since it does not require knowledge of the clutter topology. It works by moving localizers in a distributed fashion to reduce the distance error in the multi-hop localizer chains between the anchors and the sensor robot.

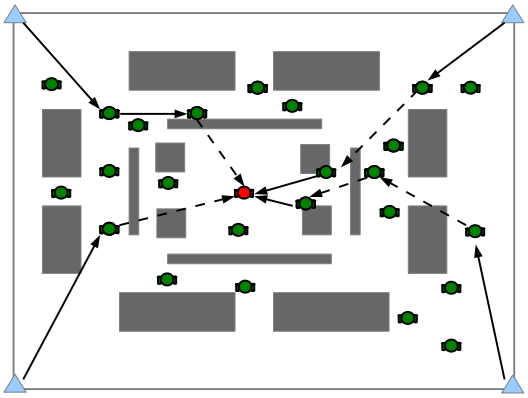
4.1 Motivation

As demonstrated in the previous chapter (Fig. 3.8), increasing the density of the localizers in the deployment area has a straightforward effect of improving the localization accuracy of DV-Distance. The reason is that, as the localizer density increases, DV-Distance can form 'less crooked' paths between the anchors and the sensor robot.

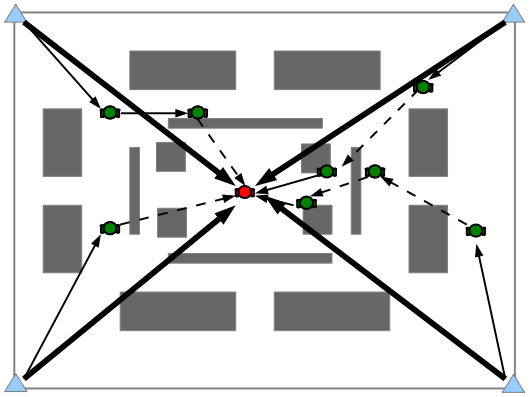
In the case of random placements, even if a very large number of localizers is deployed, in reality, only a small fraction of these localizers are used, with the remaining localizers rendered unnecessary. Fig. 4.1 demonstrates a scenario



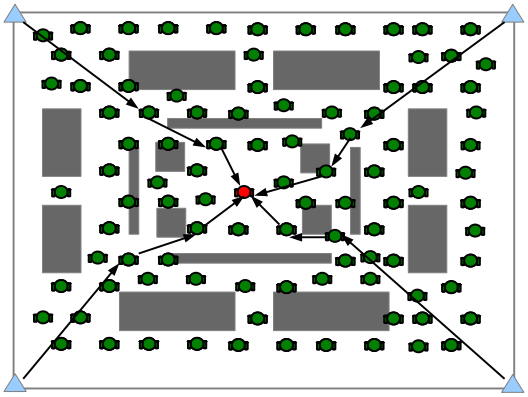
(a) Initial scenario with only NLOS single-hop distances to all anchors.



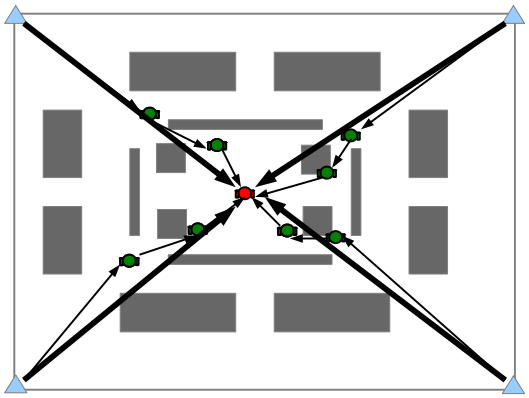
(b) Randomly scattered localizers form multi-hop distances between sensor robot and anchors using DV-Distance.



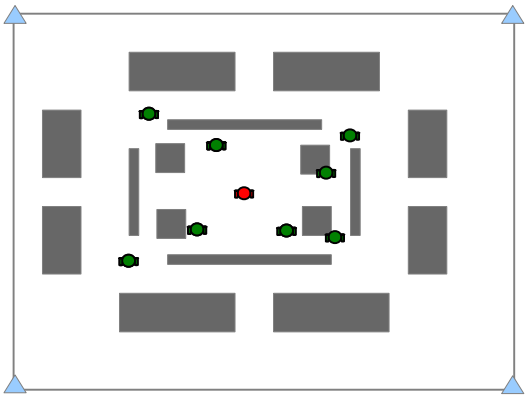
(c) Multi-hop distances reduce the distance errors between the anchors and the sensor robot when compared to single-hop NLOS distances.



(d) Localizers deployed at four times the density as in Fig. 4.1(b).



(e) Distance errors from the multi-hop chain of localizers is minimized for the given clutter topology.



(f) Aim of localizer placement is to find such placements without deploying large numbers of localizers.

Figure 4.1: Motivation for localizer placement for DV-Distance algorithm in cluttered environments.

where carefully placing localizers in the cluttered environment can have the same effect as having a large number of randomly placed localizers. In the initial scenario, shown in Fig. 4.1(a), no localizers are used and the sensor robot estimates its position using direct NLOS distances to the anchors. The localization error is large in this case because of the large errors in NLOS measurements. Next, consider the scenario shown in Fig. 4.1(b), where a sufficiently large number of localizers are placed at random positions in the cluttered environment. These localizers are used by the DV-Distance algorithm to form multi-hop paths between the anchors and the sensor robot. Of all the localizers deployed in the environment, only a small subset of the localizers are actually used to form the shortest path multi-hop distances, as shown in Fig. 4.1(c). Ideally, we would want the measured distances between the anchors and sensor robot to represent the true Euclidean distances between them, as shown by the dark thick arrows. Thus, we want these multi-hop chains to be as tight as possible around the clutter in order to minimize the distance error. With a four-fold increase in the number of localizers (Fig. 4.1(d)), we get tight chains around the clutter. In this particular case, we find that all the inter-localizer distances are LOS. Fig. 4.1(e) shows how close the multi-hop distances are to the actual Euclidean distances.

Deploying a large number of robots is clearly impractical. It is not cost-effective besides introducing other issues like increased interference in the wireless channel. If we are able to arrange localizers directly into the positions that minimize the multi-hop distances we can achieve good localization accuracy with a small number of localizers. Evidently, the efficacy of such localizer placements is influenced by the clutter topology itself. In Chapter 7, we will look at the role specific features of a clutter topology play in the estimation of multi-hop error. The communication ranges of the anchors and the localizers also affect the localizer placements.

4.2 Optimal Placement for DV-Distance (OPDV)

The **O**ptimal **P**lacement for **DV**-Distance (OPDV) algorithm aims to find the positions of localizers in a cluttered environment such that the multi-hop distances between the anchors and the sensor robot are minimized, thereby minimizing the localization error of the DV-Distance algorithm.

The OPDV algorithm makes the following assumptions:

1. The clutter topology is known.
2. The localizers can be placed accurately in desired positions in the cluttered environment.
3. The communication range of the anchors and the localizers follows a simplified disc model.
4. The sensor robot position is known a priori.

We develop two versions of the OPDV algorithm, one for the scenario where there is an unrestricted number of localizers available (Section 4.2.1), and the other for the case where the number of localizers is limited (Section 4.2.2). Finally, we discuss the complexity and optimality of OPDV in Section 4.2.3.

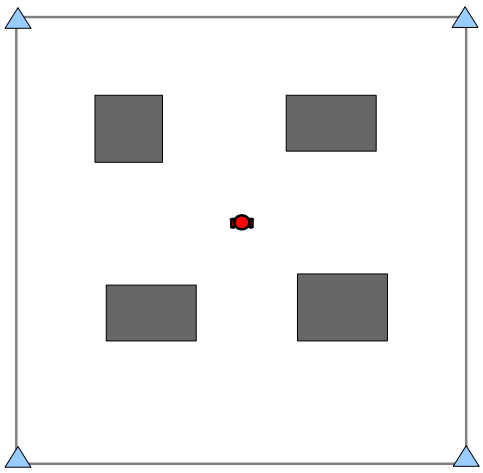
4.2.1 Basic Algorithm

In this section we describe the OPDV algorithm that assumes that there is no restriction on the number of deployable localizers. We first describe the inputs and outputs of the basic OPDV algorithm. The same can be used for the localizer-constrained version, described in Section 4.2.2.

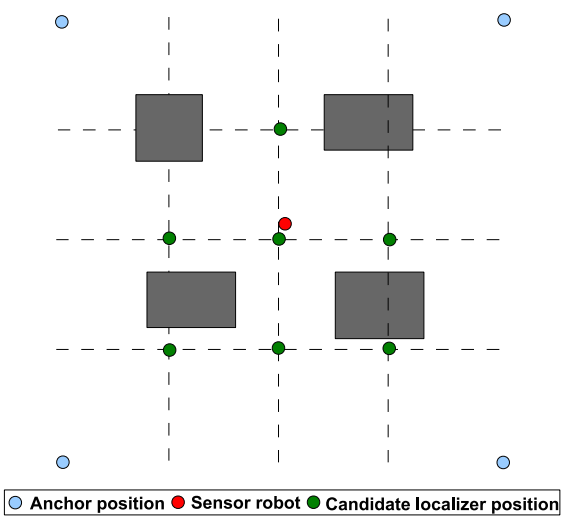
The OPDV algorithm is implemented as a graph search method. The input graph $\mathcal{G}$ is generated taking into account the following factors:

1. clutter topology C_{cl}
2. locations of the l anchors (A_1, A_1, ... A_l)
3. location of the sensor robot S
4. anchor communication range $\mathcal{C}_a$
5. localizer communication range $\mathcal{C}_l$

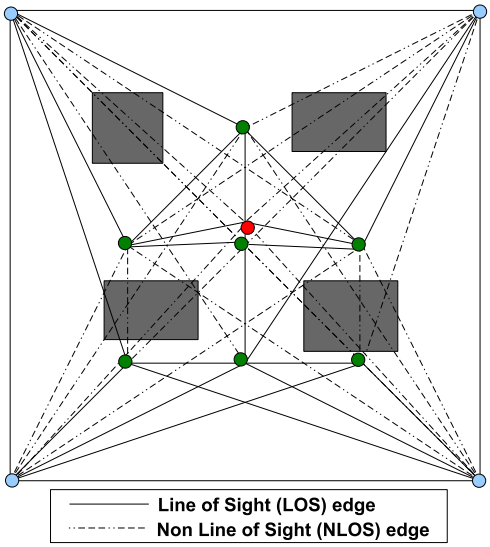
A sample clutter topology with four anchors and a sensor robot is shown in Fig. 4.2(a). First, the cluttered environment is divided into a grid (with grid resolution/spacing gs_{opdv}) and each point of the grid that is not covered by an obstacle is treated as a candidate localizer position, as shown in Fig. 4.2(b). This set of points, together with the points representing the positions of anchors and the sensor robot, form the vertices of $\mathcal{G}$. The edges of $\mathcal{G}$ are established based on the communication ranges of the anchors and localizers. For example, we draw an edge between an anchor and a potential localizer position, if their distance is smaller than the anchor communication range, $\mathcal{C}_a$. Similarly, we



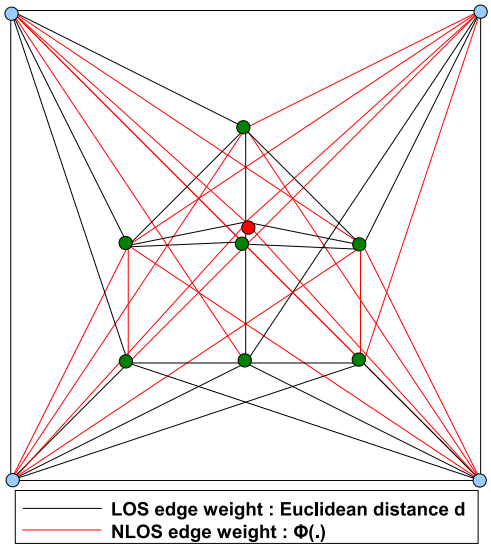
(a) Clutter topology with positions of anchors and sensor robot.



(b) Allocation of (viable) grid points for candidate localizer positions.



(c) Definition and classification of graph edges as LOS or NLOS edges.



(d) Resultant graph input $\mathcal{G}$ for OPDV algorithm.

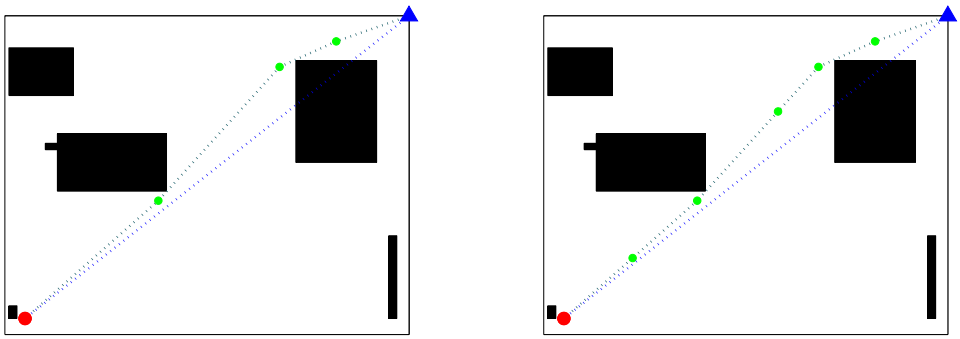
Figure 4.2: Both versions of OPDV, basic and constrained-localizer, are implemented as a graph search algorithm that finds the shortest paths in the weighted graph $\mathcal{G}$ between sensor robot (red vertex) and each of the anchors (blue vertices).

add an edge between two localizer positions if their distance is smaller than the localizer communication range, $\mathcal{C}_l$. The edges can represent two types of links: line-of-sight (LOS) links that do not intersect with obstacles, and non-line-of-sight (NLOS) links, which do intersect with one or more obstacles. The weight of the edge representing a LOS link is set to its Euclidean length d while

the weight of the edge representing a NLOS link is obtained using a function $\Phi(.)$. One version of the NLOS weight function $\Phi(.)$ could simply scale the Euclidean distance d with a NLOS penalty coefficient c_{np} such that $\Phi(.) = c_{np}d$. An alternative version could use the NLOS distance obtained from a ray-tracer (Section 3.2.1). Finally, we get the weighted graph $\mathcal{G}$ which serves as the input for OPDV. The output of OPDV are the positions of localizers between an anchor and the sensor robot.

The basic version of the OPDV algorithm uses A* graph search [31, 109] to find minimum cost multi-hop paths around the clutter. A* search uses greedy best-first search to find a least-cost path between the source and the destination. There may be other alternative paths of the same cost. The path is split into component edges and the A* search, at each step, selects the component edge that offers the minimum cost ($f(.)$) to the destination. In other words, if s represents the source vertex, d represents the destination vertex, $g(x)$ represents the cost of reaching vertex x from s and $h(x)$ represents the heuristic cost of reaching d from x, the A* algorithm selects the next edge component that minimizes $f(x) = g(x) + h(x)$. The heuristic $h(.)$ must be admissible, in that it does not overestimate the cost of reaching the destination. We implemented $h(.)$ for the OPDV algorithm as the Euclidean distance between the considered vertex and the destination vertex d, in order to ensure the admissibility of the heuristic. Furthermore, in our case the heuristic $h(.)$ is also *monotone*, i.e., $h(.)$ satisfies the condition $h(x) \leq \omega(x, y) + h(y)$, where $\omega(x, y)$ represents the weight of the edge between vertices x and y. This enables us to implement a more efficient version of the A* search algorithm where no vertex is evaluated more than once. We maintain a *closed set* to keep track of evaluated vertices in order to prevent them from being evaluated repeatedly.

The A* search is performed on the input graph $\mathcal{G}$ where the source vertex s is an anchor position and the sensor robot position is the destination vertex d. We start by placing the source vertex in a *priority queue* which pops the element (vertex) that has the least cost $f(.)$. When a vertex v is popped from the priority queue, it is added to the closed set, and each of its neighbours (that is not already in the closed set) is evaluated. For each neighbour vertex, we check if v offers a better route to the source s than its currently assigned path. In other words, we check if $g(v) + \omega(v, n_v^i) < g(n_v^i)$, where n_v^i is the i^{th} neighbour of v. If the above condition is true, we assign v as the 'parent' of n_v^i



(a) Communication range = 0.25D, Multi-hop error = 0.0188D

(b) Communication range = D, Multi-hop error = 0.0188D

Figure 4.3: Sample output for the basic version of the OPDV algorithm for different communication ranges. Multi-hop error is expressed in terms of the enclosure diameter D.

and update $g(n_v^i)$ and $h(n_v^i)$ as

$$g(n_v^i) = g(v) + d(v, n_v^i)$$
$$h(n_v^i) = g(n_v^i) + \|n_v^i, d\|$$

where $\|(.)\|$ represents the Euclidean distance. The A* search will terminate under one of the two conditions: 1) the node popped from the priority queue is the destination node d, or, 2) the priority queue becomes empty. A* search will get a negative result in the case of the latter condition, signifying that no path exists between the source and destination. In Fig. 4.3, we see an example of OPDV's output for two values of communication range, 0.25D and D (D is the diameter of the enclosure). Intuitively, the smaller the communication range, the more localizers are needed for the optimal placement.

4.2.2 Constrained-localizer Algorithm

In a practical deployment scenario, it is vital to be able to work with a limited supply of localizers. In the previous version of OPDV that assumed an unlimited supply of localizers, the grid granularity (gs_{opdv}) is the only factor that constrains the number of localizers in the path between an anchor and the sensor robot — the smaller gs_{opdv}, the more localizers can be accommodated. The A* search does not attempt to limit the number of localizers as it chooses the shortest path based on only the edge weights in input graph $\mathcal{G}$. Thus we

need an algorithm that also considers a hop-count constraint while searching $\mathcal{G}$ for the shortest path.

We use the Bellman-Ford shortest path algorithm [26] for computing the shortest paths for a given maximal hop-count H. This implementation is based on the property of the Bellman-Ford algorithm that, at its k^{th} iteration, it identifies the optimal path between a given source and each destination among paths of at most k hops. However, unlike A* search, it inherently finds the shortest paths from the source to all other vertices in the input graph $\mathcal{G}$. In contrast, A* search works towards finding a specific destination, thus being sufficient for our requirements since the single sensor robot position is always the intended destination of the graph search.

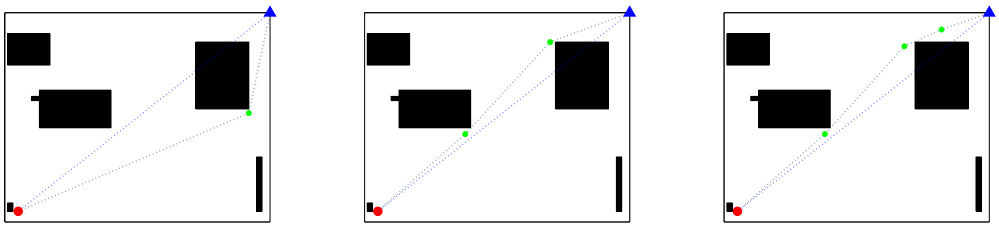
Bellman-Ford is based on the dynamic programming approach. While its basic structure is similar to Dijkstra's algorithm, it relaxes all the edges repeatedly for N-1 times (N being the number of vertices in the input graph), allowing minimum distances to propagate accurately throughout the graph. In contrast, Dijkstra's algorithm uses the greedy approach and selects the minimum weight unvisited vertex, where weight is defined as the cost to the source vertex, to relax. Consequently, given that Dijkstra's algorithm assumes that the shortest path has been obtained for each visited vertex, it cannot be used for the scenario where edges can have negative weights and subsequent paths can be found that violate this assumption. The Bellman-Ford algorithm, in contrast, can handle cases where there are edges that have negative weights since it makes no such assumption. For our application, however, the edges will always be positively weighted.

The key operation of the Bellman-Ford algorithm is the relaxation process. In this process, we check if the current shortest path to vertex u is shorter than an alternate path through vertex v. In other words, we check if

$$C(u) > C(v) + w(u, v)$$

where $C(u)$ and $C(v)$ represent the costs of the *current* minimum-cost paths from the source to u and v respectively, and $w(u, v)$ denotes the weight of the edge (u, v). If true, the minimum-cost path of u is updated to the current minimum-cost path of v, augmented with the edge (u, v).

In order to obtain the minimum cost path between the source and all other vertices for a given maximal hop-count H, we repeatedly relax all edges of the input graph at most H times instead of original N-1 times, where N is the



(a) N_{loc}: 1, Multi-hop error: 0.093D (b) N_{loc}: 2, Multi-hop error: 0.0194D (c) N_{loc}: 4, Multi-hop error: 0.0188D

Figure 4.4: Example output of constrained-localizer version of the OPDV algorithm. The multi-hop error is expressed in terms of the diameter of the enclosure D. The communication range of both anchors and localizers is set to D.

number of vertices in the input graph. We use an updated data structure for recording shortest-path costs: $C(i, h)$ denotes the cost of the (current) shortest-path for vertex i that is at most h hops from the source. It is possible that there is no path between the destination and the source after H iterations. In that case, the OPDV algorithm reports that no route exists between the source and the destination that uses at most H localizers.

Fig. 4.4 shows an example output of the localizer-constrained OPDV algorithm for a number of localizer constraints $N_{loc} = 1$, $N_{loc} = 2$ and $N_{loc} = 4$. We see that increasing the localizer limit from one to two offers the most improvement in multi-hop error. The pseudo-code of both versions of the OPDV algorithm is presented in Appendix A.

4.2.3 Optimality and Complexity Analysis

We claim that both versions of the OPDV algorithm, the basic and the constrained localizer versions, produce optimal paths between the anchor and the sensor robot for a given grid granularity. This is due to the fact that both A* search and the Bellman-Ford algorithm are complete and will always find a solution (in our case the shortest path between the anchor and the sensor robot) if one exists. Additionally, A* search is an optimal-efficient shortest path search algorithm as it considers fewer nodes than any other search algorithm which also uses an optimistic heuristic [31, 48]. An optimistic heuristic (or an admissible) is one that never overestimates the cost of reaching the goal/destination node.

The time complexity of the OPDV algorithm is dependent on two components: the input graph construction procedure and the graph search algorithm. The input graph construction procedure has a quadratic time complexity of $\mathrm{O}(N^2)$, where N is the number of vertices in the input graph. In the case of the basic version of OPDV, the time complexity of A* search is exponential in the length of the shortest length multi-hop path between the anchor and the sensor robot, $\mathrm{O}(e^L)$, $(L < N)$. In the case of the constrained-localizer version of OPDV, the Bellman-Ford algorithm has a time complexity of $\mathrm{O}(N_{loc}.|E|)$, where N_{loc} is the localizer constraint and $|E|$ represents the number of edges in the input graph $\mathcal{G}$. The resulting time complexity is $\mathrm{O}(N_{loc}N^2)$.

The output of OPDV depends on the selection of the grid size gs_{opdv}. A larger grid size would lead to 'looser' multi-hop paths around the clutter due to the sparser distribution of grid points during the construction of the input graph $\mathcal{G}$ by the OPDV algorithm. This in turn results in larger overestimate errors in the shortest length multi-hop paths found by OPDV.

4.3 Adaptive Placement for DV-Distance (APDV)

The OPDV algorithm is at best an oracle algorithm since it assumes prior knowledge of the clutter topology and easy deployment of localizers to their chosen locations. Hence OPDV cannot be deployed on a group of autonomous robots. In this section we propose an alternative algorithm, called **A**daptive **P**lacement for **DV**-Distance (**APDV**), which seeks to remedy the deficiencies of OPDV. In effect, OPDV represents the lower bound for the multi-hop overestimate error in a given localizer placement between anchors and the sensor robot. We will see, in the next chapter, that APDV fares in comparable terms with OPDV, while at the same time being practical and scalable enough to be deployed in actual robotic networks. We begin by discussing the need for APDV, pointing out those aspects of OPDV that prevent it from being deployed on real robots. We then describe the APDV algorithm in more detail in Section 4.3.2. We then present a termination proof in Section 4.3.3.

4.3.1 Motivation

OPDV requires the clutter topology to be known beforehand. This may not be possible in most scenarios where robots are usually deployed in an ad-hoc fashion in the cluttered environment for exploration and other purposes. Another

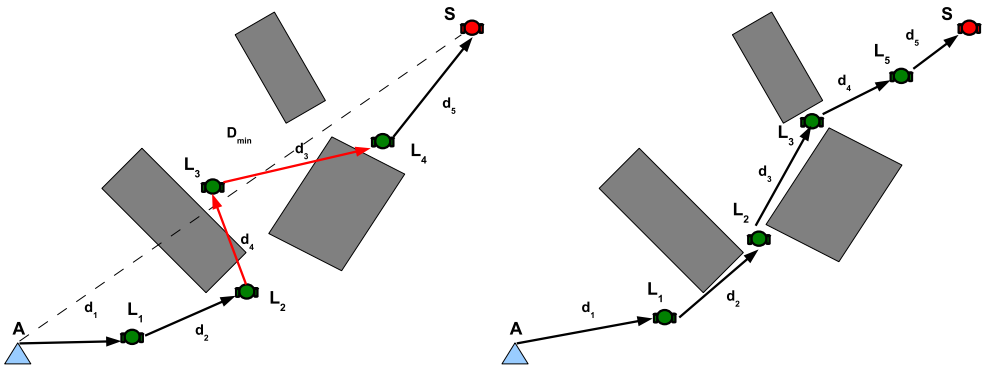
vital assumption made by OPDV is that the localizer can easily be placed in their selected positions. There are a number of issues with this assumption. Firstly, the localization error of localizers themselves will be non-trivial even in less cluttered areas. The localizers, in order to evaluate their instantaneous positions and move towards their target positions, have to be localized continuously. The localization error is expected to continually increase as a localizer moves deeper in the cluttered area near the sensor robot. Moreover, the placements selected by OPDV require the localizers to intricately arrange themselves in a chain amongst the clutter between an anchor and the sensor robot. If even a few localizers are unable to move to their exact placements, the result can be potentially that they are not in line-of-sight with adjacent neighbours. This would defeat the very purpose of the OPDV localizer placements. The centralized nature of OPDV makes it less scalable as the number of localizers increases since the placements have to be calculated at a central location and dissipated amongst the localizers. Lastly, OPDV finds a localizer placement given a known sensor robot position, which is unrealistic, and defeats the purpose of localization.

In order to tackle these issues of OPDV, we take note that it is beneficial to avoid localization of intermediate localizers. We propose an adaptive placement technique where localizers move, in a distributed manner, to minimize the cumulative distances of the multi-hop path, by only communicating with the adjacent neighbours. The proposed technique does not require explicit knowledge of the clutter topology. It does not require any identification of NLOS distances and works even when all or a majority of available distances to anchors are NLOS in nature.

Thus, the aim of the proposed APDV algorithm is to minimize the cumulative multi-hop distances between a sensor robot and the available anchors. Fig. 4.5 shows such an example with four localizers between a sensor S and an anchor A. In a clutter-free environment, the goal is to align the localizers in a straight line between the anchor and the sensor. In a cluttered environment, the goal is to move localizers so as to avoid NLOS distance measurements and minimize the multi-hop distance between anchor A and sensor robot S.

4.3.2 Algorithm Description

The APDV algorithm starts after an initial application of DV-Distance to identify the shortest chain of localizers between the sensor robot and anchor. The



(a) Initial positions of localizers between the anchor and the sensor in a cluttered environment.

(b) Desired positions of localizers between the anchor and the sensor in a cluttered environment.

Figure 4.5: APDV aims to align localizers between the sensor and the anchor node.

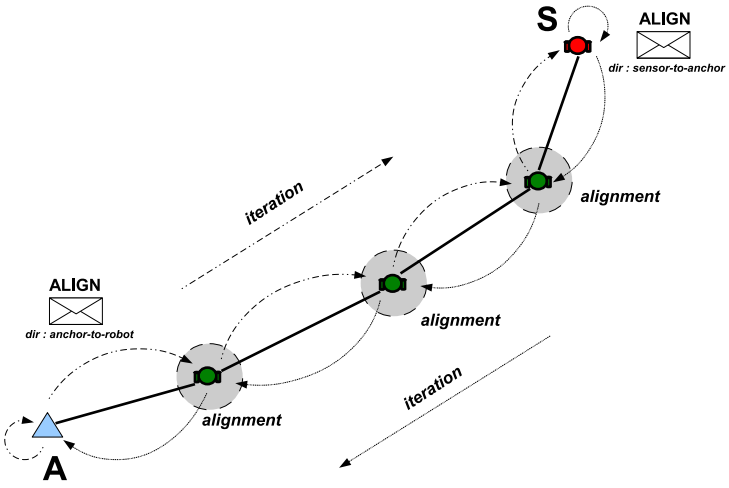


Figure 4.6: APDV algorithm in action.

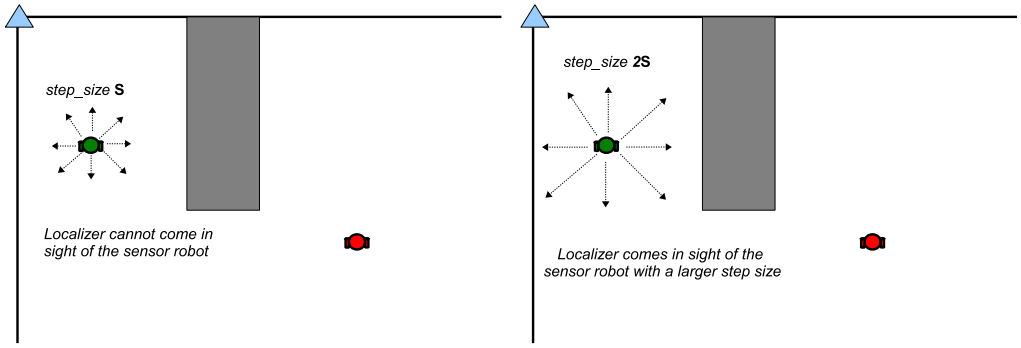
algorithm then proceeds with an iterative application of the *alignment* process in order to minimize the cumulative distances of the multi-hop path. The first iteration is triggered by the sensor robot, and results in a wave of localizer alignments from the sensor robot to the anchor. The second iteration is triggered by the anchor and moves in the opposite direction, and so on, as shown in Fig. 4.6. The iterations stop when either the anchor or sensor robot detects that no localizer was able to move to a new (better) position in the most recent iteration.

We will now describe an iteration in more detail. During the first iteration, the sensor robot S sends an **ALIGN** message to its neighbour localizer. The localizer reacts by sending a **DIST-REQ** message to request distance

measurements from its two neighbours. Its neighbours reply with **DIST-REP** messages that contain the requested distance measurements. The localizer then adjusts its position to better align itself with respect to its neighbours. We have defined four different variants of the *alignment* procedure, which are discussed in detail in Section 4.3.2.2. Once it moves to its new position, the localizer updates its distance to the anchor and broadcasts a DV-Distance update message with the new updated distance. It then passes the **ALIGN** message to the next localizer in the chain. Eventually the first localizer in the chain forwards the **ALIGN** message to the anchor A itself. The anchor decides whether to start a new iteration of the APDV algorithm based on whether at least one localizer was able to move to a new position. Thus, APDV repeatedly performs the alignment procedure in a series of iterations, and converges when none of the localizers in the chain moves to a new position.

4.3.2.1 APDV Parameters

The three key parameters of the APDV algorithm are the maximum number of directions (N_{dir}) to be probed by a localizer during the alignment process, the step size (s) taken by the localizer during the alignment procedure, and the step size threshold ($\mathcal{S}_T$). The first two parameters are tuned to take into consideration the type of robotic hardware, the locomotion speed of the robot, the energy constraints of the robot, the deployment clutter topology and the required localization accuracy. The line-of-sight (LOS) distance error is also an important factor. A high LOS ranging accuracy can allow us to employ a smaller N_{dir} with a smaller step size s as the robot is able to discern subtle changes in distances for relatively less probing. Similarly, a smaller N_{dir} with a large s is recommended in scenarios where distance error in LOS situations is comparable in magnitude with a small s. Having a large N_{dir} leads to better accuracy of APDV, but it comes at the cost of increased effort by the localizers. Since the localizer is typically a battery-powered robot, a large N_{dir} can have an adverse impact on the up-time of the robot. A large N_{dir} will also extend the completion time of APDV. This can be a critical issue when the sensor robot is measuring an ephemeral phenomenon and is required to determine its position within a certain period of detecting the phenomenon. It is vital to understand the tradeoff between accuracy, the energy constraints of the localizer robot, and the completion time of APDV.



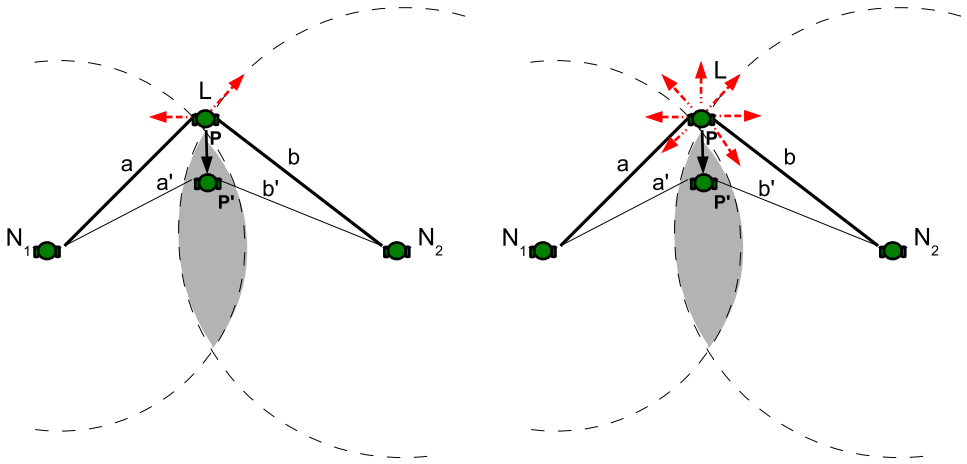
(a) Step Size = $\mathcal{S}$ (b) Step Size = $2\mathcal{S}$

Figure 4.7: Need for step size threshold $\mathcal{S}_T$ in APDV. The localizer can obtain better distances to its neighbours if it can break out of the clutter using a step size of $2s$.

The step size s represents the size of the step taken by the localizer during the alignment procedure. The smaller s is, the better the accuracy of APDV with respect to the multi-hop path between the anchor and sensor robot. On the other hand, we see that a smaller s will lead to a bigger overhead in terms of energy consumption for the robot as well as a larger completion time of APDV. The value of s allows us to make a tradeoff between localization accuracy and operational costs, similar to that with N_{dir}. The third key parameter of the APDV algorithm is the step size threshold ($\mathcal{S}_T$). The parameter adapts APDV to operate in cluttered environments. Here, instead of terminating the alignment procedure for the initial step size, say s, a localizer instead increases it and repeats the alignment procedure all over again. For example, in Fig. 4.7 we can see that the localizer will be able to proceed to a better position with a step size of $2s$ in the presence of clutter between the localizer and the sensor robot. If the alignment procedure converges again without moving to a new position, the step size is again incremented. These successive increments of the step size are bounded by the step size threshold, $\mathcal{S}_T$. When the localizer is not able to move to a fresh position (that offers more proximity to its neighbours) even when the $s = \mathcal{S}_T$, the localizer terminates the alignment procedure.

4.3.2.2 Alignment Variants

The alignment procedure plays a key role in the APDV algorithm. We have designed and tested four variants of this procedure: Random (R), Exhaustive (E), Random-with-Feedback (RF) and Exhaustive-with-Feedback (EF). The R and E variants are *single-step* procedures, in which a localizer takes at most



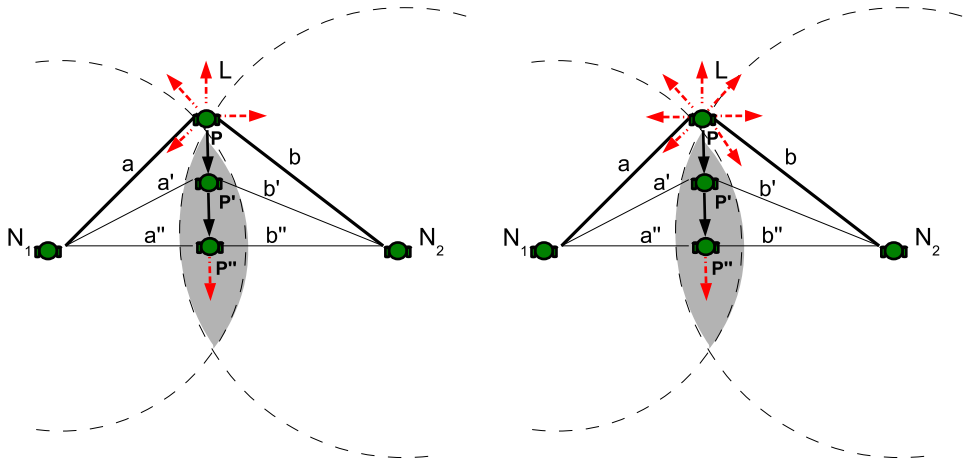
(a) R variant. Localizer L randomly chooses a direction (out of N_{dir} directions) and checks if both neighbour distances decrease. If yes, the alignment terminates.

(b) E variant. Localizer L evaluates all ($N_{dir} = 8$) directions and chooses the direction that reduces the sum of its neighbour distances (a+b) the most (here at position P').

Figure 4.8: Single-step alignment variants terminate when either, none of the N_{dir} directions give any improvement, or when the localizer moves to a new position P'.

one step to improve its position. The RF and EF variants are *multiple-step* procedures, in that a localizer continues to move in the direction where it obtained the first improvement. The alignment process terminates under different conditions for single-step and multiple-step procedures. For single-step procedures, the alignment terminates when the localizer is not able to move to a new position in any direction around its current position. For multiple-step procedures, the alignment procedure terminates under one of the two conditions: the localizer starts the alignment procedure but no direction yields an improvement over the current position; or when the previous optimal direction no longer yields an improvement.

We will now discuss the four variants in more detail. **Random (R) variant**. In the R variant, the localizer randomly chooses a direction (out of N_{dir} available directions) to take a single step. Let a and b represent the distances to its two adjacent neighbours at its original position, and a' and b' represent the respective distances after it moves to the new position. The localizer will accept the new position only if $a' < a$ and $b' < b$. In this case, the alignment process ends after the localizer moves to the new position. If it cannot accept the new position, it returns to its former position, marks the previously selected direction as inappropriate, and selects another random direction to probe. This



(a) RF variant. Localizer L evaluates random directions (red arrows) before choosing a direction (and position P') which reduces both distances its neighbours N_1 and N_2. Thereafter it evaluates a relaxed condition of $(a+b) < (a'+b')$ for each step in the previous successful direction. The RF variant terminates when the condition is violated.

(b) EF variant. Localizer L evaluates all ($N_{dir} = 8$) directions and chooses the direction that reduces the sum of its neighbour distances (a+b) the most (here at position P'). Thereafter it moves in the selected direction as long as the sum of neighbour distances decrease.

Figure 4.9: Multiple-step alignment variants, RF and EF variants.

is continued until it either accepts the new position, as shown in Fig. 4.8(a), or it runs out of directions to choose from.

Exhaustive (E) variant. In the E variant, the localizer moves into each of the N_{dir} available directions and thereafter selects to move in the direction that offers the maximum improvement $(a+b) - (a'+b')$, provided that $(a+b) > (a'+b')$. For sufficiently large N_{dir}, the localizer moves in the direction that is almost perpendicular to the imaginary line connecting its two neighbours, as shown in Fig. 4.8(b). If there is no direction that yields an improvement, i.e. for which $(a+b) > (a'+b')$, the localizer stays in its initial position and terminates the alignment procedure.

Random with Feedback (RF) variant. In the RF variant, initially, the localizer makes its first step as in the R variant. Recall that it only moves if it finds a direction that reduces the distances to both its neighbour, i.e. $a' < a$ and $b' < b$. In this case, it sets $a = a'$ and $b = b'$, and continues to move in the same direction and to take new measurements a' and b' to its two neighbours. The process continues until $(a'+b') > (a+b)$. When this happens, it stops. Notice that the criterion used to select the direction of the first step is stricter than the one used for the following steps. The first condition is strict to ensure

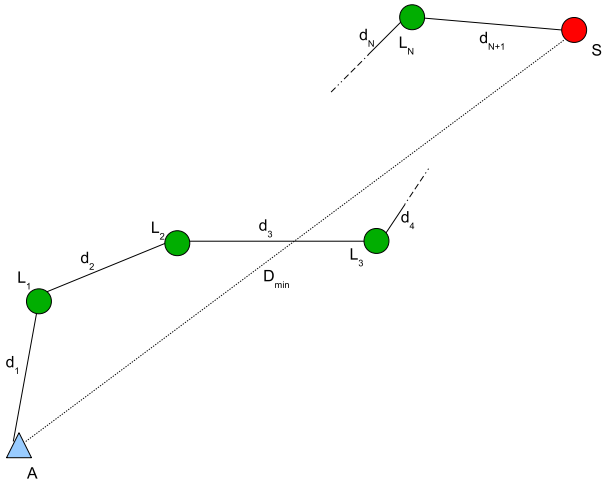


Figure 4.10: Multi-hop distance formation between the anchor A and sensor robot S, involving N intermediate localizers, $L_1 ... L_N$.

that the right direction that provides a fast route to alignment is found; the second is looser to avoid re-examining direction options frequently once the right direction is selected.

Exhaustive with Feedback (EF) variant. In the EF variant, initially, the localizer makes its first step as in the E variant. Recall that it checks all directions, and moves in the one that offers the maximum positive $(a+b)-(a'+b')$. It continues to move in the same direction and to take new measurements until $(a+b) < (a'+b')$. When this happens, it ends the alignment procedure.

4.3.3 Termination Proof

In this section we derive a termination proof for the APDV algorithm. We consider an anchor (A) and a sensor robot (S), with N localizers (L_1, L_2, L_3 ... L_N) in between them forming a multi-hop chain. The outline of the proof is as follows. We begin by showing that the alignment procedure is finite; we thereafter show that each subsequent alignment procedure during the execution of the APDV algorithm leads to a decrease in the cumulative multi-hop distance; finally we show that the sequence of iterations is finite and thus APDV is shown to terminate in finite time.

Lemma 4.3.1. *A monotone sequence $\{s_k\}$, which is bounded by L, is finite if and only if the difference between any two consecutive elements, s_k and s_{k+1}, has a lower bound ϵ, $\epsilon > 0$.*

Lemma 4.3.2. *The* alignment *procedure terminates in finite time.*

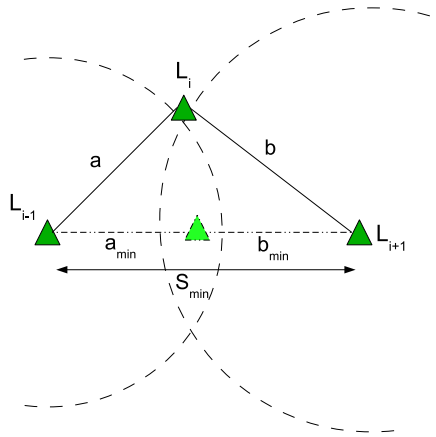


Figure 4.11: Localizer L_i with adjacent neighbours, L_{i-1} and L_{i+1}, at the start of an *alignment* procedure. The minimum sum of neighbour distances, S_{min} $(a_{min} + b_{min})$, is obtained when L_i moves onto the Euclidean line connecting L_{i-1} and L_{i+1}. Note that while S_{min} is fixed, the values of a_{min} and b_{min} can be arbitrary.

Proof. Consider localizer L_i with adjacent neighbours L_{i-1} and L_{i+1}, as shown in Fig. 4.11 (except L_1, which has A and L_2 as one of its adjacent neighbours, and similarly L_N has L_{N-1} and S, $N \geq 2$). Let a and b be its initial distances to L_{i-1} and L_{i+1} respectively. We proceed to show that each of the four variants of the *alignment* procedure is finite in time.

The R variant stops after L_i evaluates at most N_{dir} distinct directions for realizing a position, where the new neighbour distances (a' and b') satisfy the condition $a' < a$ and $b' < b$. Since N_{dir} is a finite number, the R variant stops after a finite period and both the new neighbour distances are smaller than the original ones.

In the E variant, the localizer L_i takes steps in all N_{dir} directions around its original position and evaluates the position for which the decrease in the sum of neighbour distances is maximum (and positive). Again, since N_{dir} is finite, the E variant will stop after the localizer evaluates all directions. Finally, $(a' + b') < (a + b)$.

The RF variant is a multi-step alignment procedure where the first evaluation is based on comparing individual distances, but subsequent evaluations are based on comparing the sum of distances. Suppose the localizer takes k positions before stopping. For the first new position, we obtain $a'_0 < a_0$ and $b'_0 < b_0$; for the next position, we obtain a'_1 and b'_1 such that $(a'_1 + b'_1) < (a_1 + b_1)$ where $a_1 = a'_0$ and $b_1 = b'_0$. If a_{sh} and b_{sh} denote the neighbour distances when

L_i lies on the shortest path between its neighbours (note that a_{sh} and b_{sh} are not necessarily unique), we can see that:

$$(a_0 + b_0) > (a_1 + b_1) > (a_2 + b_2)....(a_{k-1} + b_{k-1}) > (a_k + b_k) \geq (a_{sh} + b_{sh})$$

If we place a condition that, for every new position chosen, $(a + b) - (a' + b') \geq \epsilon_{ab}, \epsilon_{ab} > 0$, we can use Lemma 4.3.1 to prove that the RF variant is finitely bounded. Similarly, the EF variant can be proven to terminate in finite time. □

Lemma 4.3.3. *The cumulative sum of multi-hop distances between A and S decreases monotonically when a localizer completes its alignment procedure.*

Proof. Let D_0 denote the initial sum of multi-hop distances between A and S. We can classify the movements of a localizer during its alignment as *explorative*, when it moves in a given direction to evaluate the new distances to its neighbours and *corrective*, when it moves in a particular direction after inferring the given direction is (most, in the case of exhaustive variants) beneficial in reducing the sum of its neighbour distances. In the case of multi-step alignment variants, multiple corrective steps are made before the alignment procedure terminates. Let D_1 denote the multi-hop sum of distances when a localizer completes an alignment procedure after making corrective moves for the first time. Similarly, let D_i denote the sum of multi-hop distances after the i^{th} alignment procedure during the execution of APDV. From the proof of Lemma 4.3.2 we can deduce that $(D_{i-1} - D_i) \geq \epsilon_{ab}$. □

Theorem 4.3.4. *APDV terminates in a finite period.*

Proof. Let D_{sh} denote the shortest distance between anchor A and sensor robot S. In the case of a cluttered environment, it denotes the shortest distance obtained meandering through the clutter present between A and S. Lemma 4.3.3 states that the sequence S = (D_0, D_1, D_2,.....) is monotonically decreasing. Evidently, the sequence is bounded by D_{sh}.

$$D_0 > D_1 > D_2.... \geq D_{sh}$$

According to Lemma 4.3.1, sequence S has a finite number of elements. Thus we prove that APDV terminates in a finite time period. □

4.4 Discussion

This chapter proposed two localizer placement algorithms for cluttered environments, in particular OPDV and APDV. The purpose of these algorithms is to place localizers in certain positions amongst the obstacles such that the multi-hop distances between the anchors and the sensor robot are minimized (or, in the case of APDV, reduced to a large extent). The OPDV algorithm is centralized in nature and assumes that the clutter topology and the position of the sensor robot are known beforehand. The APDV algorithm, on the other hand, does not require knowledge of the clutter topology and can be deployed on real robots in obstacle-prone settings. The key idea behind APDV, that makes it mimic OPDV's output without the underlying assumptions, is that it is possible to move localizers to optimal positions in the clutter by merely measuring distances to neighbour nodes. This allows the localizers to move in the clutter without knowing their own locations. Since APDV works by seeking to minimize the distance between anchors and the sensor robot, it is robust to transient/temporal oscillations in distance measurements, that, at most, increases the settling time of the algorithm by a small extent.

For both OPDV and APDV algorithms, increasing the number of sensor robots being assisted by the same set of localizers simultaneously can lead to suboptimal solutions for all the sensor robots involved, given arbitrary clutter topologies.

In the next chapter, we will perform an extensive simulation-based evaluation of the OPDV and APDV algorithms for a variety of parameters and environmental settings.

Chapter 5

Simulation-Based Evaluation

In this chapter, we will evaluate and discuss the performance of the two localizer placement algorithms introduced in the previous chapter, namely Optimal Placement for DV-Distance (OPDV) and Adaptive Placement for DV-Distance (APDV). We begin by laying out the methodology for our evaluation in Section 5.1. Next, we evaluate OPDV in Section 5.2 where we look at the multi-hop distance error for a variety of OPDV parameters. In Section 5.3, we perform an extensive sensitivity analysis of APDV. Finally, we compare the performance of OPDV and APDV to other competing techniques in Section 5.3.4.

5.1 Methodology

In this section we will describe the methodology we use for the sensitivity analysis of the OPDV and APDV algorithms. We begin by briefly describing the simulation setup since it is similar to the one described in detail previously in Chapter 3. Thereafter we describe the simulation parameters and performance metrics. Then, we list the various competing algorithms used for comparison with OPDV and APDV.

5.1.1 Simulation Setup

The OPDV evaluation is done using an implementation of OPDV in vanilla C++ and MATLAB MEX. For the APDV experiments, we use the ProwlerSim simulator, introduced in Chapter 3, for our simulations. The APDV algorithm is implemented in ProwlerSim. We use similar settings of ProwlerSim for the evaluation of the APDV algorithm as we did for the multi-hop localization

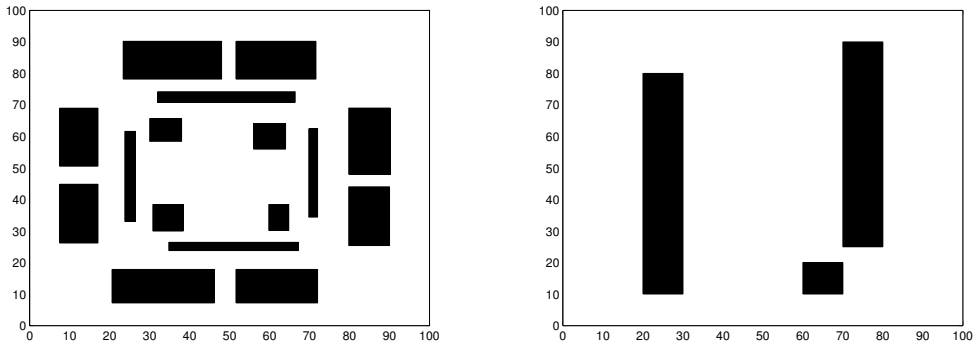
algorithm, outlined in Section 3.2.2. We also use the ray-tracer, introduced in Section 3.2.1, for simulating NLOS distances between nodes.

5.1.2 Simulation Parameters

In this section, we lay out the simulation parameters for the evaluation of the OPDV and APDV algorithms in three parts: sensitivity analysis for OPDV; sensitivity analysis for APDV; and competing algorithms comparison.

5.1.2.1 OPDV Parameters

We look at the performance of OPDV as we vary grid spacing (gs_{opdv}), communication range ($\mathcal{R}_c$), and localizer limit (N_{loc}). The communication range of anchors and localizers is set to the same value. The parameter values, including the default values, are presented in Table 5.1.



(a) Clutter topology A (b) Clutter topology B

Figure 5.1: Clutter topologies used for OPDV sensitivity analysis. Clutter topology A is representative of the MS class while Clutter topology B represents the FL clutter class.

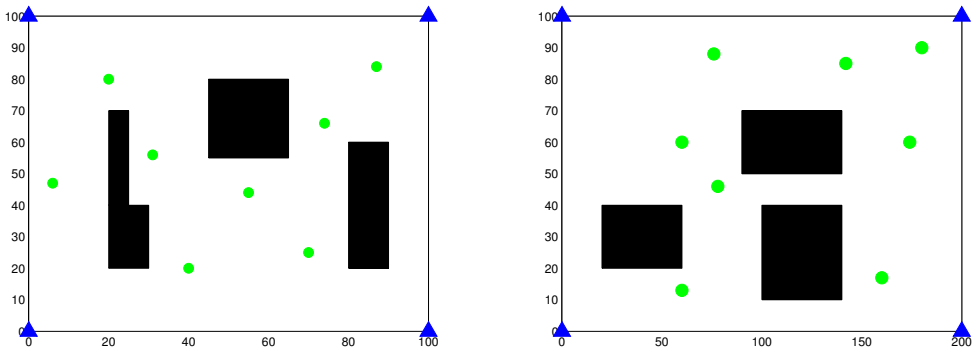
We evaluate the performance of OPDV for each OPDV parameter as an average over 30 distinct randomly generated sensor robot positions in the clutter topology. The grid spacing gs_{opdv}, which represents the granularity of the grid used to generate candidate localizer positions, is varied from 0.01S to 0.1S. The communication range $\mathcal{R}_c$ is varied from 0.05D to D, with a default value of D. The localizer limit N_{loc} for the constrained-localizer version of the OPDV algorithm is varied from 4 to 12. For all the OPDV parameters we use two clutter topologies, A and B, as shown in Fig. 5.1. The two clutter topologies represent the MS (many small obstacles) and FL (few large obstacles) clutter

Parameter	Default	Range
Number of sensor robot positions (N_{sp})	30	-
Grid spacing (gs_{opdv})	0.02S	[0.01S 0.02S 0.03S 0.05S 0.1S]
Communication range ($\mathcal{R}_c$)	D	[0.05D 0.01D 0.25D 0.5D 0.75D D]
Localizer limit (N_{loc})	Unrestricted	[4 8 12 16 20]
Clutter topology	[A B]	-

Table 5.1: Parameter values for ODPV sensitivity analysis. S is the length of the side of the square enclosure while D is its diameter. Clutter topologies A and B are shown in Fig. 5.1.

classes respectively. We use only rectilinear shapes for clutter pieces as well as the enclosing boundaries, though one can extend this work to include circular shapes with some modifications to the ray tracer described in Section 3.2.1.

5.1.2.2 APDV Parameters



(a) Clutter topology C with initial localizer positions (8 localizers).

(b) Clutter topology H with initial localizer positions (8 localizers).

Figure 5.2: Clutter topologies C and H used for the error sensitivity analysis of APDV algorithm. Both are representative of the FL clutter class.

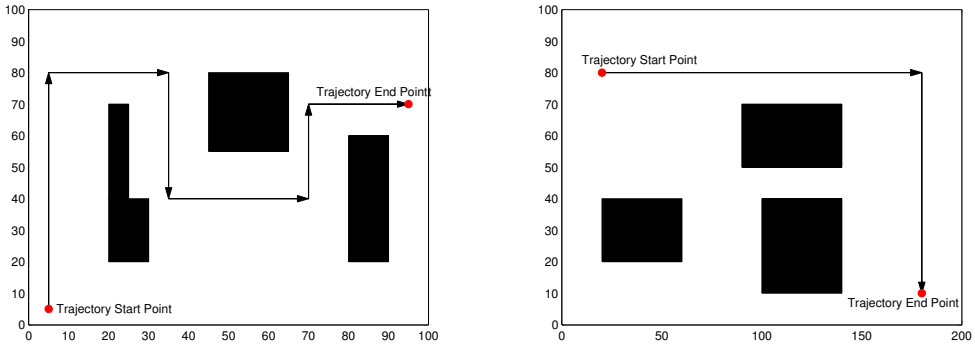
For the sensitivity analysis of APDV, we vary the APDV parameters step-size threshold ($\mathcal{S}_T$), the number of probing directions (N_{dir}), and the number of localizers N_l in the cluttered environment. We also evaluate APDV under varying conditions of environmental and hardware features such as clutter topology, LOS distance error and robot odometry error. Analysis is done for both the static case, where N_{sp} sensor robot positions are individually localized using APDV and the performance metrics are averaged over all positions, and

	Default	Range
Communication range ($\mathcal{R}$)	4D	-
Step size (s)	5	-
Step-size threshold ($\mathcal{S}_T$)	15	[5 10 15 20]
Number of probing directions (N_{dir})	12	[4 8 12]
Number of localizers (N_l)	8	[4 8 12 16]
LOS Gaussian error $\mathcal{N}(0, \sigma^2)$, σ	0	[0 0.5 1 2]
Robot odometry error (Rotational) $\mathcal{N}(0, (\theta * \epsilon_{rot}^2)$, ϵ_{rot}	0	[0.02 0.04 0.08 0.16]
Robot odometry error (Displacement) $\mathcal{N}(0, (d * \epsilon_{dsp}^2)$, ϵ_{dsp}	0	[0.02 0.04 0.08 0.16]
Clutter topology	C	[C H]
Sensor robot trajectory	TR_1	[TR_1, TR_2]

Table 5.2: Parameters for APDV sensitivity analysis. Clutter topologies C and H are shown in Fig. 5.2 and the trajectories TR_1 and TR_2 are shown in Fig. 5.3. d is the actual distance the robot is instructed to move. D is the diameter of the enclosure.

the dynamic case, where the sensor robot moves in an assigned trajectory and is localized at N_{spt} discrete positions along the trajectory. In this case, performance metrics are averaged over all discrete positions in the trajectory. For both static and dynamic scenarios we evaluate the effect of the various parameters over clutter topologies C and H (shown in Fig. 5.2), with clutter topology C being default. For the dynamic case, we considered two trajectories TR_1 and TR_2, shown in Fig. 5.3, that use clutter topologies C and H respectively. The number of discretized positions N_{spt} is dependent on the trajectory and the step size (s).

The communication range ($\mathcal{R}$) is set to 4D, where D is the diagonal length of the enclosure. The step-size threshold ($\mathcal{S}_T$) is varied from 5 to 20 units, with a default value of 15. The number of probing directions (N_{dir}) is varied from 4 to 12, with 12 being the default value. We evaluate the performance of APDV with 2, 4, 8, 12 and 16 localizers with 8 localizers used as the default case. The initial positions of the eight localizers for both clutter topologies C and D are given in Fig. 5.2. The LOS distance error is modelled as a Gaussian error $\mathcal{N}(0, \sigma^2)$ with the mean set to 0 and the standard deviation varying from $\sigma = 0.5$ to $\sigma = 2$. Two types of robot odometry errors are considered: rotational odometry error, which represents the error in turning θ° in any direction, and displacement odometry error, which represents the error



(a) Sensor robot trajectory TR_1 in clutter topology C.

(b) Sensor robot trajectory TR_2 in clutter topology H.

Figure 5.3: Sensor robot trajectories TR_1 and TR_2 moving in two clutter topologies C and H respectively.

a robot makes when moving a distance d. Both types of odometry errors are modelled as zero-mean Gaussian with variance being a function of the correct angle and correct displacement. In other words, rotational odometry error is $\mathcal{N}(0, (\theta * \epsilon_{rot}^2))$, where θ is the angle the robot is required to turn and ϵ_{rot} is the scale factor for rotational odometry error. Similarly, the displacement odometry error is $\mathcal{N}(0, (d * \epsilon_{dsp}^2))$, where d is the distance the robot is required to travel and ϵ_{dsp} is the scale factor for displacement odometry error. Both ϵ_{rot} and ϵ_{dsp} are varied from 2% to 16%. By default, we set both the LOS error and odometry error to zero.

5.1.2.3 Parameters for Competing Techniques

In the case of OPDV, the default values of grid spacing gs_{opdv}, communication range $\mathcal{R}$ and localizer limit N_{loc}, listed in Table 5.1, are used. Similarly, the default values for the APDV parameters listed in Table 5.2 are used for the APDV algorithm. In the case of the random-walk localizer placement techniques, the number of steps taken by each localizer N_{steps} is fixed to 10 and for each random move, a localizer chooses randomly from 12 directions.

5.1.3 Performance Metrics

We use the following metrics to measure the performance of the OPDV and APDV algorithms. The first two metrics are similar to the ones defined in Section 3.2.4.

1. Localization Error (LE). This is the average localization error over the N_{spt} discretized positions along the pre-determined trajectory of the sensor robot. The formulation is similar to the one given in Section 3.2.4. LE can also be calculated on the N_{sp} sensor robot positions in the static case.

2. Distance error (DE). This is the average distance error over the N_{spt} discretized positions along the pre-determined trajectory of the sensor robot. The formulation is similar to the one given in Section 3.2.4. DE can also be calculated on the N_{sp} sensor robot positions in the static case.

3. Total moves/unit-steps (T_{moves}). This represents the total number of moves or unit-steps, of size s, taken by all localizers during the entire simulation run. This does not include distance covered by the sensor robot over the assigned trajectory. This metric can be used to assess the energy and time requirements of APDV.

5.1.4 Algorithms

This section list the algorithms evaluated in this chapter; the first four represent existing approaches and the last two are approaches proposed in this dissertation (Chapter 4).

1. *No localizer (LLSE).* This refers to single-hop localization that does not make use of localizers. It takes as input single-hop distances to anchors, which could be either LOS or NLOS, and uses the linear least squares (LLSE) method for localization.

2. *No localizer (UBLSE).* This is similar to *No localizer (LLSE)* except that it uses an improved least squares estimator designed to tackle NLOS distances that takes into account upper-bound constraints [131].

3. *Static DV-Distance.* This is the existing DV-Distance algorithm, which makes use of localizers, but does not attempt to adjust their positions.

4. *Random Walk.* This method allows the localizers to take a certain number of steps in a random walk for each step taken by the sensor robot. The DV-Distance algorithm is used during their random walk to help the sensor robot to localize with more accurate distances to the anchors.

5. *OPDV*. This is an oracle algorithm, which assumes knowledge of the clutter topology and the sensor position, and finds localizer placements that minimize the multi-hop distances between the anchors and the sensor robot.

6. *APDV*. This is the distributed control algorithm that carefully moves localizers, forming multi-hop chains between the sensor robot and the anchors, in order to form as tight paths as possible around the clutter. APDV does not require knowledge of the clutter topology and requires localizers to communicate with only their adjacent neighbors while adjusting their positions.

5.2 Sensitivity Analysis of OPDV

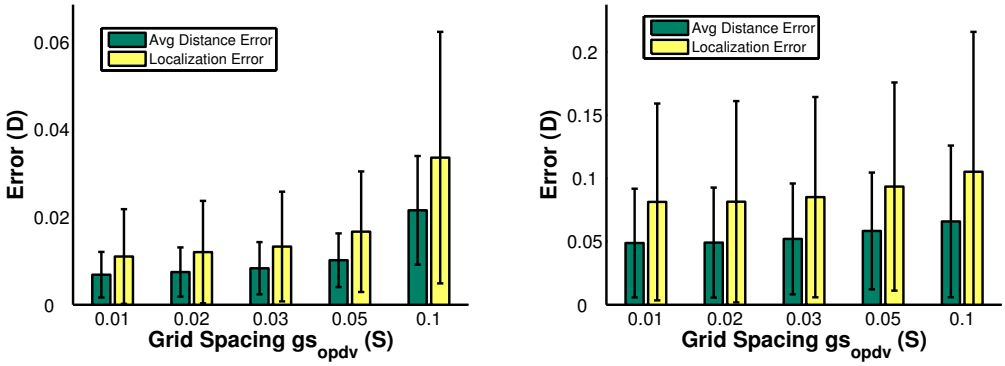
We are now in a position to perform the sensitivity analysis of the two localizer placement algorithms introduced in the previous chapter. In this section we will look at the performance of OPDV in a cluttered environment. We evaluate the performance of OPDV for distinct classes of clutter topologies and show that using OPDV-mandated localizer placements can lead to tremendous gains in localization accuracy when compared to other single-hop localization techniques. Finally we compare the performance of the DV-Distance algorithm when the localizers are placed randomly to that when they are placed using OPDV. Here, we see that OPDV achieves high localization accuracy that can only be achieved with a large (and almost impractical) number of localizers in case of random placements.

5.2.1 Effect of OPDV Parameters

We evaluate the three OPDV parameters, namely grid spacing (gs_{opdv}), localizer communication range ($\mathcal{R}_c$) and localizer limit (N_{loc}), on the two clutter topologies A and B shown in Fig. 5.1. We use N_{sp} (=30) randomly generated positions. The anchor positions are fixed at the four corners of the square enclosure.

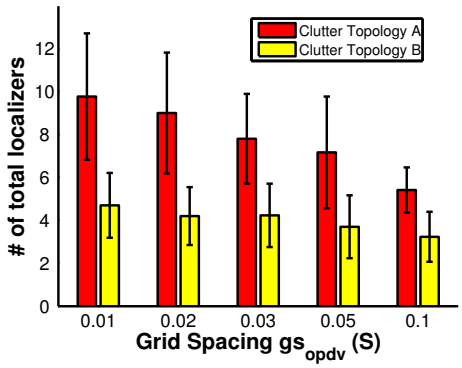
5.2.1.1 Grid Granularity gs_{opdv}

We evaluate the performance of OPDV for various values of gs_{opdv}. The performance is measured by the distance and localization errors over N_{sp} sensor robot



(a) Clutter topology A

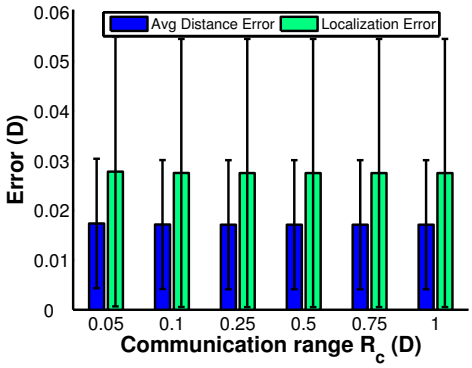
(b) Clutter topology B



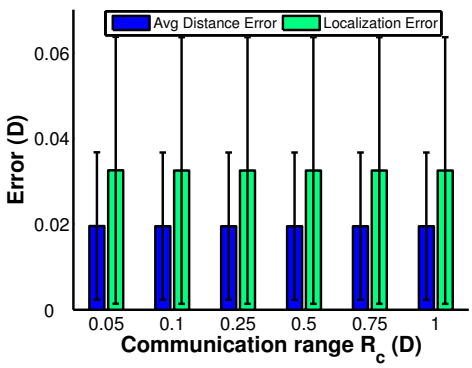
(c) Number of required localizers

Figure 5.4: Influence of grid spacing gs_{opdv} of OPDV. No limit is set on the number of localizers that can be used by OPDV.

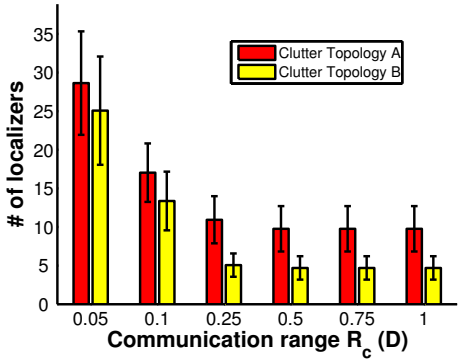
positions. Fig. 5.4 shows that both the distance and localization errors increase as we increase the grid spacing gs_{opdv}. The error magnitude depends on the clutter topology. For example, when $gs_{opdv} = 0.01S$ in clutter topology B, it yields a localization error of 0.08D, while in clutter topology A the same value of gs_{opdv} gives an error of 0.03D. The reason for this is that clutter topology B is more heavily cluttered, having large-size clutter objects, than clutter topology A. Fig. 5.4(c) shows the number of localizers required as gs_{opdv} increases. In general, the number of localizers decreases as the grid spacing increases as the number of candidate localizer positions decreases. Again, we see that the clutter topology has a marked influence on the number of required localizers with gs_{opdv}.



(a) Clutter topology A



(b) Clutter topology B

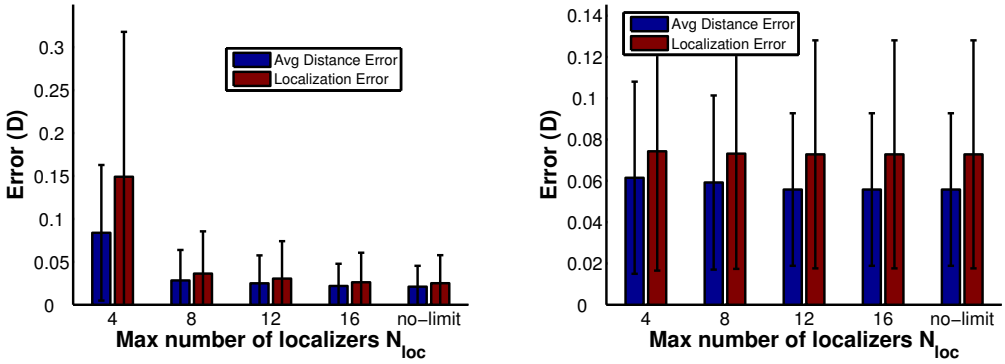


(c) Number of required localizers

Figure 5.5: Effect of communication range $\mathcal{R}_c$ on the performance of OPDV. Grid spacing gs_{opdv} is set to 0.02S, where S is the square side of the enclosure. No limit is set on the number of localizers that can be used by OPDV.

5.2.1.2 Communication range $\mathcal{R}_c$

Next, we evaluate the influence of communication range $\mathcal{R}_c$ on the performance of OPDV. Typically, a larger $\mathcal{R}_c$ should lead to improvements in the performance (distance/localization error) of OPDV due to the fact that candidate localizer positions (vertices) in the input graph $\mathcal{G}$ now have access to more vertices. Thus OPDV is able to output tighter localizer chains that weave in between the clutter objects. The results of our evaluation are shown in Fig. 5.5. Here, we find that $\mathcal{R}_c$ has little influence on the distance/localization error of OPDV. The reason for this is that since there is no limit on the number of localizers that can be used by OPDV, the algorithm is able to find a solution that is close to optimal for a given value of gs_{opdv}. However, in the case of a smaller $\mathcal{R}_c$, it comes at the cost of a larger number of required localizers as shown in Fig. 5.5(c). Here, we find that for both clutter topologies used in the



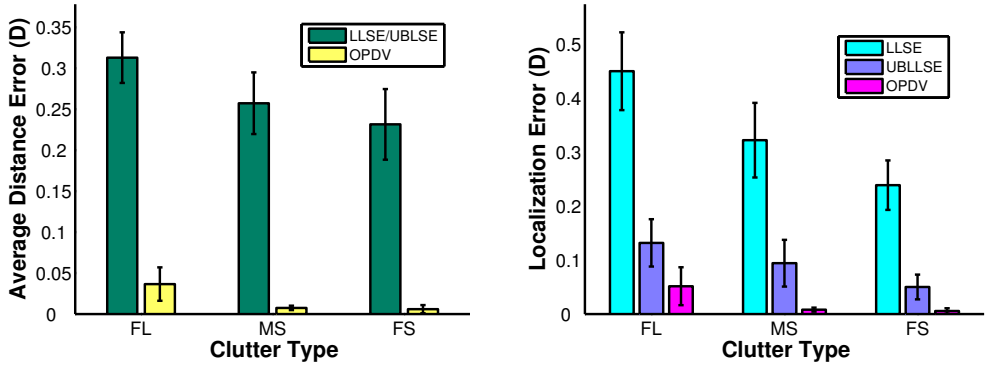
(a) Clutter Topology A (b) Clutter Topology B

Figure 5.6: Effect of localizer-limit N_{loc} on the performance of OPDV. Grid spacing gs_{opdv} is set to 0.02S, where S is the square side of the enclosure.

evaluation, the number of required localizers remains the same after a certain threshold (0.25D in both cases). The threshold marks the point where the candidate localizer positions are connected in the input graph $\mathcal{G}$ in a way that produces the best solution for a given gs_{opdv}. This threshold can be important in realistic deployments where it is necessary to limit the communication range of localizer robots in order to reduce the overall interference in the wireless channel.

5.2.1.3 Localizer limit N_{loc}

Finally, we look at the effect of the localizer limit (N_{loc}) on the performance of OPDV. This is an important parameter as it allows us to limit the number of localizers to a practical number. Fig. 5.6 shows the results for two clutter topologies. We make two observations: 1) the accuracy of OPDV increases with larger values of N_{loc}; 2) we see that the clutter topology has an effect on the variation of OPDV error with N_{loc}, as in the case of the communication range $\mathcal{R}_c$. In case of Clutter Topology B, the distance and localization errors are higher in magnitude and stabilize early (larger value of N_{loc} does not necessarily improve OPDV accuracy). However, in the case of Clutter Topology A, we do see a significant improvement in OPDV accuracy for larger values of N_{loc}. The reason behind this is that, since Clutter Topology B has large size clutter objects, the OPDV solution for a large portion of the sample sensor robot positions entails overcoming a large clutter object by placing a localizer at the corner. Since this arrangement is fairly inflexible and does not need more localizers, increasing the number of localizers does not lead to improved



(a) Average Distance Error (b) Localization Error

Figure 5.7: Comparison of the performance of OPDV with single-hop localization techniques for different classes of clutter topologies.

localization accuracy. In the case of topology A, which has numerous small sized clutter objects, it is possible to get tighter multi-hop paths when more localizers are supplied. Clutter Topology A actually performs worse than Clutter Topology B when the number of allowed localizers is small, i.e., $N_{loc} = 4$. However, when more localizers are available, the localization accuracy in Clutter Topology B rapidly increases. In Figs. 5.6(a) and 5.6(b), we can also compare the performance of the constrained-localizer OPDV to that of the basic (unrestricted-localizer) version. In both cases we see that it takes only $N_{loc} = 8$ to achieve a comparable localization accuracy to that of the basic version.

5.2.2 Influence of Clutter Topology

Next, we evaluate the performance of OPDV with respect to three clutter topology classes, namely the FL (few large clutter objects), MS (many small clutter objects) and FS (few small clutter objects), introduced in Chapter 3. The clutter classes signify the variations in sizes of the obstacles and the amount of clutter in the environment.

At the same time, we compare the performance of OPDV with that of single-hop localization techniques, linear least squares estimation (LLSE), and upper-bound least squares estimation (UBLSE) [131]. For evaluation, 30 clutter topologies are randomly chosen for each class, and for each clutter topology, 100 random positions for the sensor robot are generated. For each sensor robot position, we first evaluate whether it has LOS or NLOS distances with each of the four anchors. LOS distance error is set to 0 while NLOS distance error is

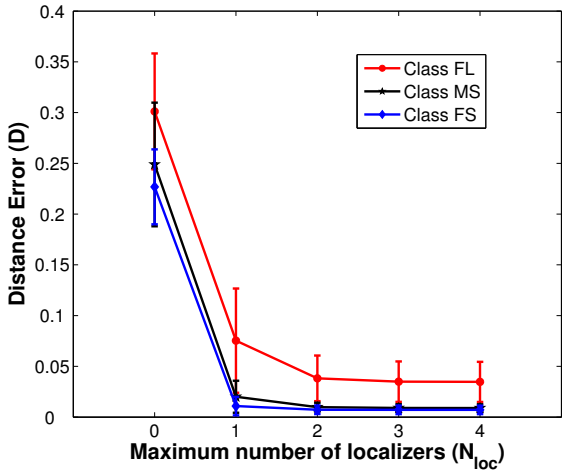
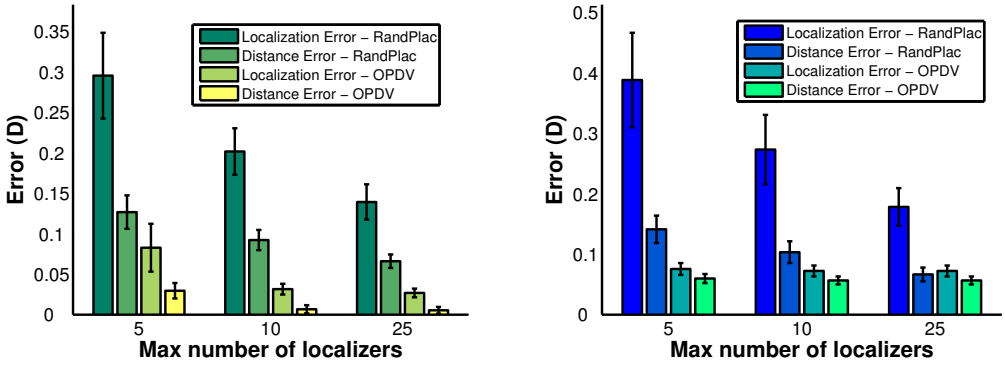


Figure 5.8: Influence of clutter topology on distance error of constrained-localizer version of OPDV. Please note that N_{loc} represents the maximum number of localizers allowed for *an* anchor; and not the total number of localizers allocated to all anchors.

obtained through ray tracing. In case of OPDV, the distance error, if any, is due to the overestimating nature of the multi-hop paths. The results are shown in Fig. 5.7. Both the NLOS error and the multi-hop error in OPDV-based DV-Distance decrease as the clutter size decreases in size. However the NLOS error (in LLSE/UBLSE) is many times greater than the multi-hop error, being 8.5 times in case of the FL clutter class, to as much as 39 times in case of the FS clutter class. Since distance error is a major source for localization error, one can see the immense benefits of using OPDV-based multi-hop localization in such scenarios. UBLSE is known to perform well in situations where the NLOS distances form a minority of all distances to anchors. Hence, on average, it performs much better than LLSE.

We now look at the influence of clutter topology on the distance error of the constrained-localizer version of OPDV. We consider 30 clutter topology samples from each clutter class. N_{sp} (=30) sensor robot and anchor positions are randomly selected, with the anchor positions being confined to the enclosure boundary. The constrained-localizer version of OPDV is applied to each pair of sensor robot and anchor position, with N_{loc} being varied from 0 to 4. The results are shown in Fig. 5.8. We find that clutter class FL shows the error with the largest magnitude and variance, while clutter classes MS and FS yield smaller distance errors. We find that the improvement in distance errors when a single localizer is added varies substantially between the various clutter classes, with



(a) Clutter Topology A (b) Clutter Topology B

Figure 5.9: Comparison of OPDV optimal localizer placement and random localizer placement for DV-Distance.

clutter class FL yielding a gain of 75%, clutter class MS yielding a gain of 91% and clutter class FS yielding a gain of 98%. We also find that two localizers are sufficient to get the maximum benefit in terms of error reduction for all clutter classes. This observations suggests that a small (practical) number of localizers may be allocated to obtain significant benefits in terms of localization accuracy.

5.2.3 Comparison with Random Placement

Finally, we compare OPDV-based localizer placement to random placement for DV-Distance. For evaluation purposes, we select $N_{sp} = 30$ randomly chosen positions for the sensor robot in both clutter topologies A and B. In the case of random localizer placement, we vary the number of localizers that are randomly placed from 5 to 25. In the case of OPDV, we vary N_{loc} from 5 to 25. The results are shown in Fig. 5.9. We observe that OPDV performs much better than random localizer placement for both clutter topologies. In both clutter topologies, the performance of OPDV with just 5 localizers exceeds the performance of random placements, even with 25 localizers, by a factor of at least two.

In this section, we looked at the evaluation of the OPDV algorithm. OPDV is at best an oracle-type algorithm since it assumes complete knowledge of the clutter topology and requires localizers to be placed in accurate positions deep in the cluttered environment — both being unrealistic assumptions. In the next section, we look at the performance evaluation of APDV, which has

been designed to overcome the shortcomings of OPDV in terms of real-world application.

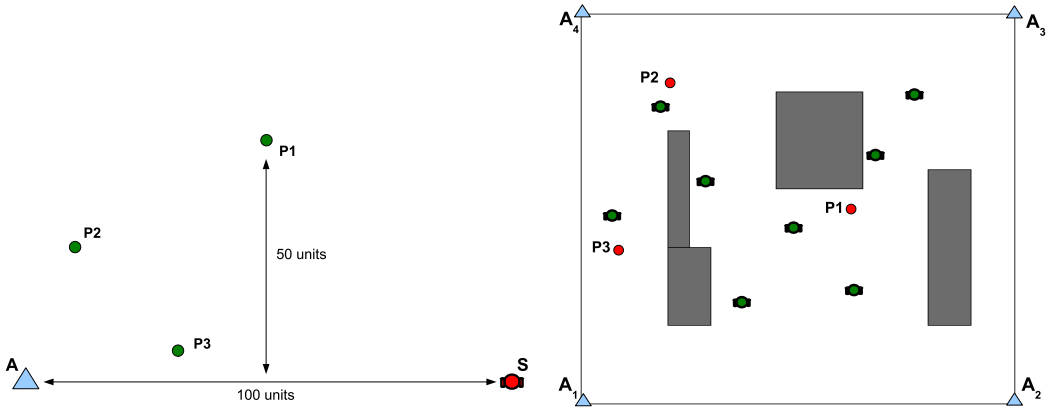
5.3 Sensitivity Analysis of APDV

In this section, we perform a simulation-based sensitivity analysis of APDV. We look at the effect of ADPV parameters, such as step-size threshold, number of probing directions and number of localizers, We also look at environmental/hardware factors such as clutter type, line-of-sight (LOS) distance errors and robot odometry errors. We begin by an extensive evaluation of the four variants of the APDV alignment procedure under various conditions. Thereafter, using the EF variant as default, we analyze the effect of various parameters on APDV, both in the static and dynamic cases. For the static scenario, APDV is applied to each position of the sensor robot independently. The dynamic scenario entails the application of APDV at discrete positions along a sensor robot's trajectory through the cluttered environment.

5.3.1 APDV Alignment Variants

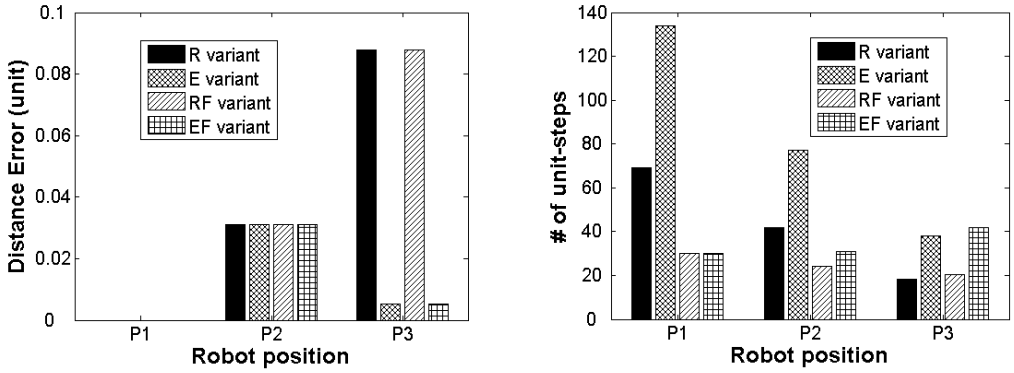
The alignment procedure is the basic building block of the APDV algorithm. It plays a key role in the distributed nature of APDV. APDV makes a localizer move to strategic positions in the clutter topology by merely measuring distances to its neighbours and avoiding clutter using obstacle-detection sensors. We evaluate four variants of the alignment procedure, introduced in Section 4.3.2.2. Two of them, the R and RF variants, use randomness to quickly find a viable direction to move the localizer such that its neighbour distances reduce. The other two, E and EF variants, use exhaustive search in N_{dir} directions to thoroughly search for the optimal direction to move the localizer such that its neighbour distances are reduced the most. On the other hand, the R and E are single-step variants, in that they terminate after moving to the first viable/optimal position. The RF and EF are multiple-step variants, in that a localizer continues to move in the first optimal direction assuming that viability/optimality of the direction holds.

We evaluate the four variants of the alignment process and compare their performance for clutter-free conditions as well in cluttered NLOS-prone conditions. Besides this, we evaluate the variant with varying distance and odometry errors.



(a) Clutter-free setting (b) Clutter-prone setting

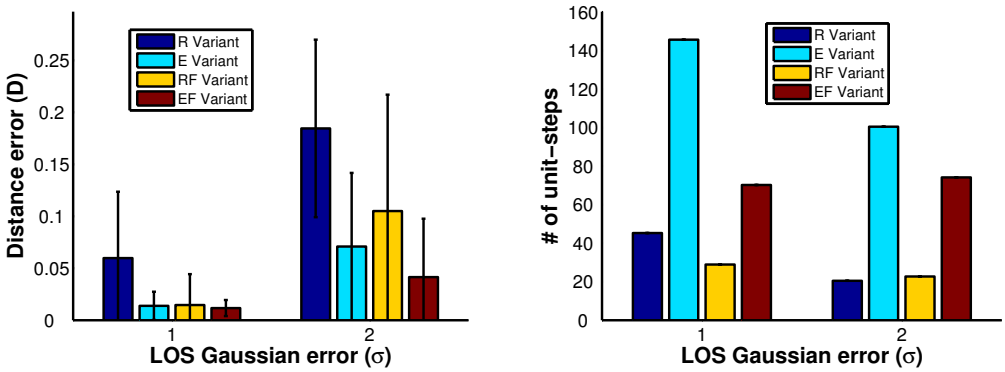
Figure 5.10: Evaluation setup for clutter-free and cluttered environments. In case of the clutter-free setup, a localizer is placed in the positions P_1, P_2 and P_3, while in case of the cluttered environment setup, a sensor robot is placed in the P_1, P_2 and P_3.



(a) Distance Error (b) Number of unit-steps

Figure 5.11: Comparison of variants in scenario where distance measurements are error-free.

Clutter-free environments with error-free conditions. We first evaluate the four variants in error-free conditions, i.e. when all distances are accurate and localizers have no odometry errors. We consider a scenario where a sensor S cannot directly communicate with an anchor A, but only through a localizer L, which initially is placed in one of the three positions P_1, P_2 or P_3, shown in Fig. 5.10(a). Fig. 5.11 shows that all four variants offer extremely high accuracy in error free conditions (given that the distance between sensor robot and anchor is 100 units). The exhaustive variant tends to be slower than the random variant, because it examines all directions before selecting one. The feedback-based variants (RF and EF) are typically much faster than the simple (R and

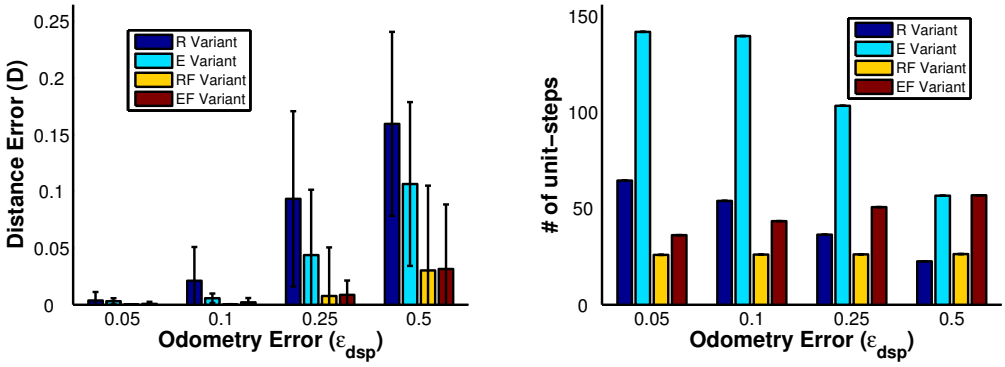


(a) Absolute Distance Error (b) Number of unit-steps

Figure 5.12: Comparison of variants in condition where the LOS distance measurements are corrupted by Gaussian noise.

E) variants because they leverage feedback about which direction works well, and continue moving in that direction as long as it is beneficial. The further away the localizer is from the anchor and the sensor, the greater the benefits of the feedback mechanism in terms of speed of completion of the alignment process.

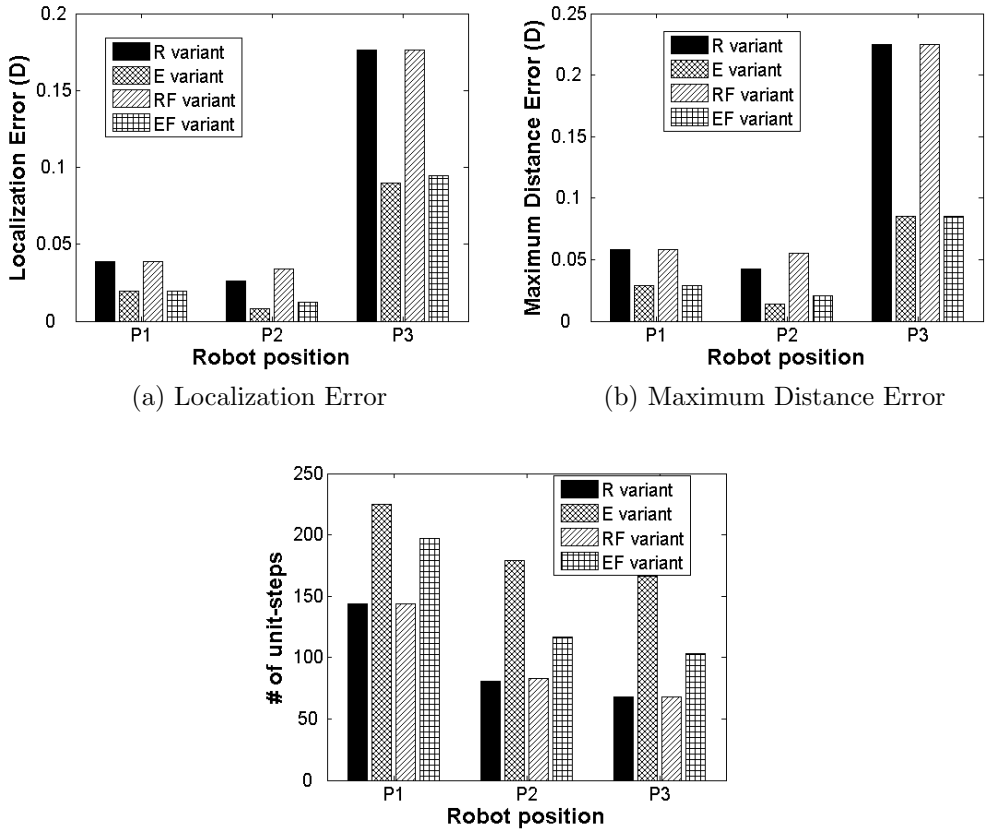
Clutter-free environments with LOS Gaussian noise. We next evaluate the variants in line-of-sight conditions but where the distance measurements are contaminated with Gaussian error. We assume that the localizer is initially in position $P1$, as shown in Fig. 5.10(a). Fig. 5.12 shows that as we increase the LOS error, the E and EF variants outperform the R and RF variants. The cause for this lies in the condition used by the localizers to select their next move. Recall that the exhaustive (E and EF) variants find a direction acceptable only if $(a' + b') < (a + b)$, whereas the random (R and RF) variants initially require that $a' < a$ and $b' < b$. The stricter condition set by the random variants is violated more frequently as we increase the Gaussian distance errors, which makes them give up and settle for sub-optimal localizer placements. The impact of distance measurement errors on the number of steps is more complex. On the one hand, the number of steps is increased because localizer decisions are affected by random errors and often make inconsistent decisions about the optimal direction. On the other hand, because of random errors, localizers typically end the alignment process earlier than they should, taking fewer steps and stopping at sub-optimal positions. Again, we find that the exhaustive variants, E and EF, take many more steps than the random variants, R and RF.



(a) Distance Error (b) Number of unit-steps

Figure 5.13: Comparison of alignment variants in the presence of odometry errors. The localizer starts at position $P1$ as shown in Fig. 5.10(a).

Clutter-free environments with odometry errors. We evaluate the four variants in the presence of odometry errors in the locomotion of the mobile localizers. We add to each localizer step a Gaussian error that is dependent on the distance traversed; if d is the intended distance, the actual distance traversed is $d + e_d$, where $e_d = \mathcal{N}(0, d * \epsilon_{dsp})$, where ϵ_{dsp} represents the scale factor of the odometry error. Fig. 5.13 shows that the feedback-based variants, RF and EF, are least affected by the odometry errors. The reason is that once the localizer selects a direction, it takes several steps in the same direction, until it aligns itself with its two neighbours. The errors accumulated have a relatively small effect on the final (converged) position. In contrast, the R and E variants attempt to select a 'fresh' direction every time, making them more susceptible to odometry errors. Likewise, we see in Fig. 5.13(b) that the localizer converges quickly (into an erroneous position) in the R and E variants.
Cluttered environments with NLOS distance errors: Finally, we evaluate the four variants in a cluttered NLOS-prone environment. Fig. 5.10b shows the clutter topology with the initial positions of 8 localizers scattered in the area. Four anchors are set in the corners of the square enclosure. We compare the four variants of APDV with static placement, i.e. the existing DV-Distance algorithm that does not attempt to adjust localizers from their initial positions. We find that the exhaustive variants, E and EF, deliver the most accuracy, but at the cost of increased number of localizer moves, as shown in Fig. 5.14. However, the EF variant requires substantially less number of localizer moves than the E variant. Fig. 5.15 shows that in this clutter setting,



(a) Localization Error

(b) Maximum Distance Error

(c) Total number of localizer moves

Figure 5.14: Comparison of the four variants in an NLOS-prone environment.

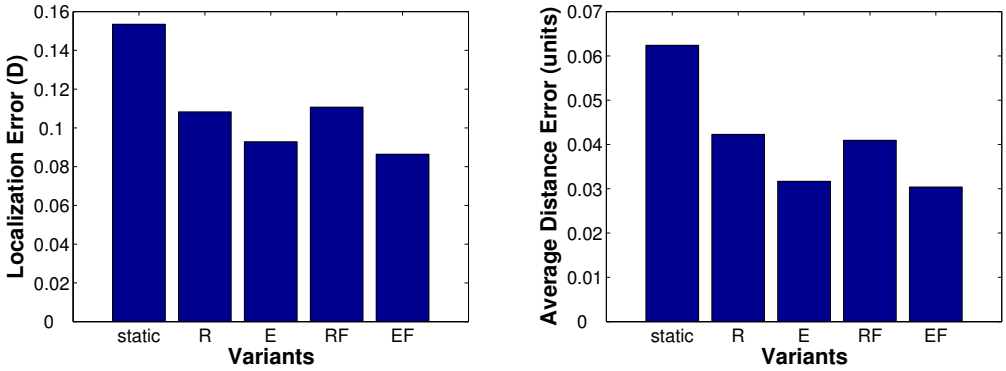
the EF variant performs the best and incurs only half the distance error of the static case.

5.3.2 Effect of APDV parameters

In the previous section, we evaluated the four variants of the alignment procedure and concluded that the EF variant offers the best performance. Henceforth we will use the EF variant as the default alignment procedure in APDV. In this section, we will look at the effect APDV parameters, namely the step-size threshold $\mathcal{S}_T$, the number of probing directions N_{dir} and the number of available localizers N_l in cluttered environments.

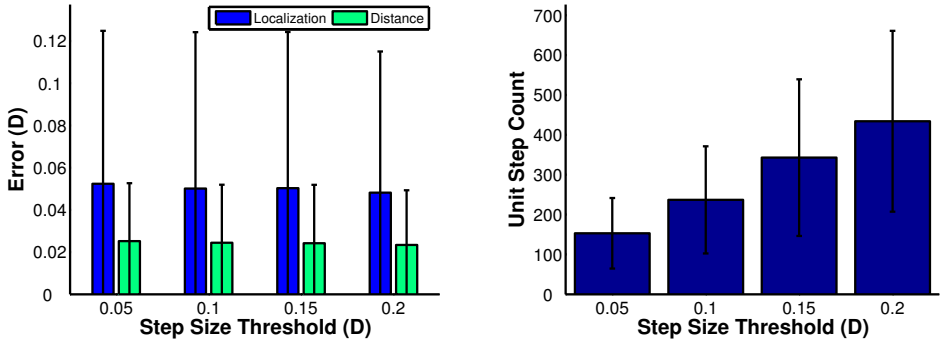
5.3.2.1 Step-Size Threshold $\mathcal{S}_T$

To recollect from Chapter 4, the step-size threshold is introduced to adapt APDV to operate in cluttered environments. The main motivation in using this



(a) Localization error (b) Average distance error

Figure 5.15: Comparison of alignment variants in a cluttered environment for random sensor robot and localizer positions.

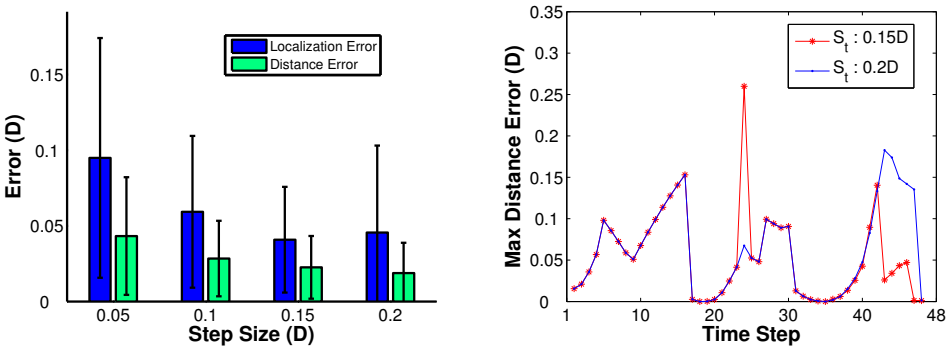


(a) Localization/Distance Error (b) Number of unit-steps

Figure 5.16: Evaluation of APDV for varying step-size thresholds given independent sensor robot positions. The clutter topology is shown in Fig. 5.2(a), while the positions of the eight localizers and sensor robot is randomly generated.

parameter is to increase the chances of the localizer breaking out of occlusion from behind clutter and gaining line-of-sight with its adjacent neighbours, thus improving the accuracy of the distance measurements.

First, we look at the influence of the step-size threshold on APDV for individual positions of the sensor robot. We choose $N_{sp} = 30$ random positions in the clutter topology C. For each sensor robot position, we place 8 localizers in random positions in the cluttered environment. We evaluate the performance improvement of APDV as the step-size threshold is increased from 0.05D to 0.2D, where D is the diameter of the enclosure. As we see in Fig. 5.16(a), the larger the step-size threshold, the smaller the multi-hop distance errors, even though the improvement for this particular setting is small. However, this



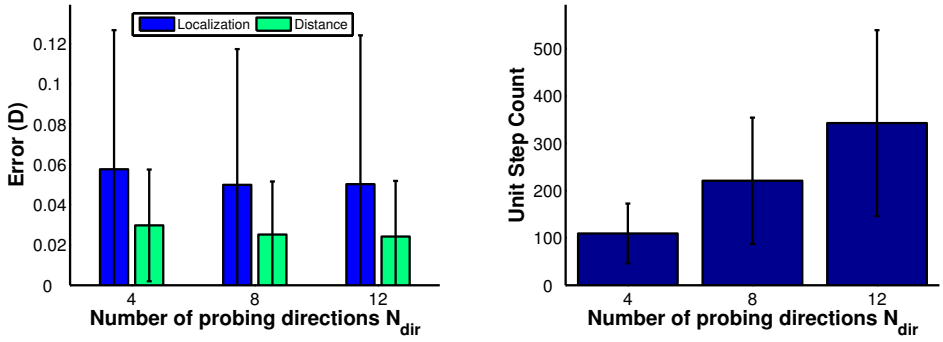
(a) Localization/Distance Error

(b) Comparison of $\mathcal{S}_T$ = 0.15D and $\mathcal{S}_T$ = 0.2D

Figure 5.17: Effect of step-size threshold on the performance of APDV. The clutter topology C with initial localizer positions and the mobile sensor robot trajectory TR_1 are shown in Figs. 5.2(a) and 5.3(a) respectively.

comes at the cost of increased number of unit-steps, as shown in Fig. 5.16(b). Here, a smaller number of unit-steps is not necessarily advantageous when the localizer has 'given up' earlier with a smaller step-size threshold. Thus, during realistic deployments one has to choose between localization accuracy and energy/time constraints of the available hardware and application.

Next, we look at how step-size threshold influences the performance of APDV over an entire trajectory of the sensor robot. We perform simulations of the sensor robot moving through the clutter topology C using the trajectory TR_1, given in Fig. 5.3(a), under varying step-size thresholds. The results are shown in Fig. 5.17. Unlike the static scenario, we do not see a concrete relationship between distance error and the step-size threshold particularly between $0.15D$ and $0.2D$. The reason for this is the complex role of the initial positions of the localizer at the start of APDV. The final positions of the localizers after the termination of any application of APDV become the initial positions for its next application. While a higher step-size threshold can give better results *initially*, the arbitrary positions of the localizers with respect to the clutter topology and the trajectory of the sensor robot may render their positions as sub-optimal initial positions for the next APDV application. Fig. 5.17(b) shows the comparison of the maximum distance error of the sensor robot for $\mathcal{S}_T$ = 0.15D and $\mathcal{S}_T$ = 0.2D. We see that initially APDV with $\mathcal{S}_T$ = 0.2D performs well especially for the 24^{th} step, but later on, these localizer positions cause a greater distance error in the later half of the trajectory. The step-size threshold value of 0.15D delivers better results here because the lo-



(a) Localization/Distance Error (b) Total number of unit steps

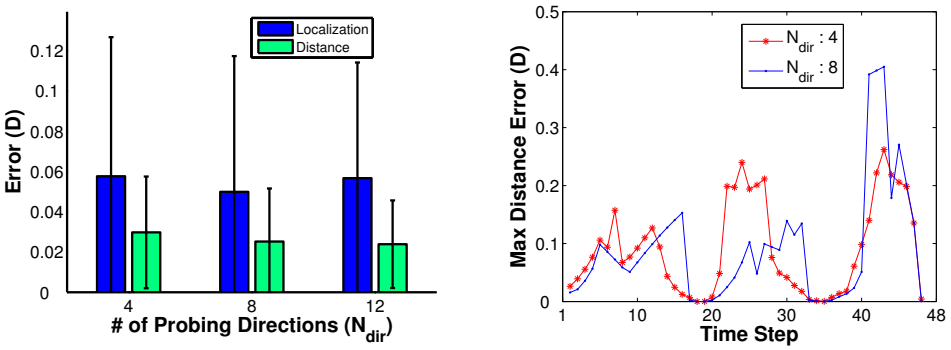
Figure 5.18: Evaluation of APDV for varying N_{dir} values given random independent sensor robot positions. The clutter topology is shown in Fig. 5.2(a), while the positions of the eight localizers and sensor robot are randomly generated.

calizers are situated in *better* positions vis-a-vis the clutter, anchor and sensor robot positions. The large variances in the graphs of Fig. (5.17) points to the large variations of the location error over different positions of the sensor robot with respect to the clutter topology, with the error being *relatively large* in a *small* number of sensor robot positions.

5.3.2.2 Number of Probing Directions N_{dir}

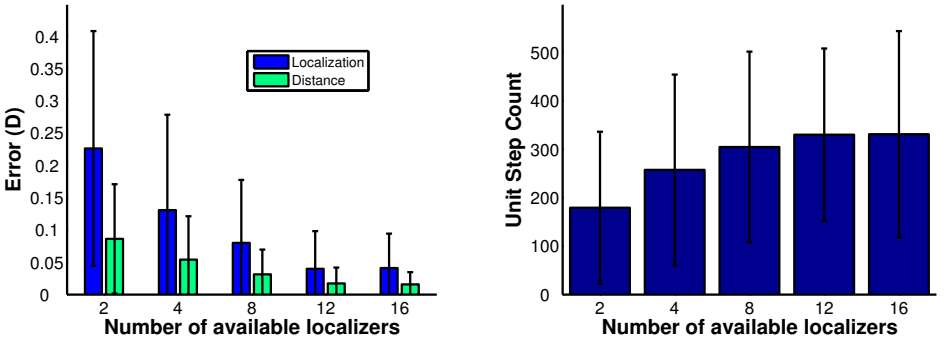
Next we look at the effect of the number of probing directions N_{dir} on the performance of APDV. We first evaluate the performance of APDV for random independent sensor robot positions with N_{dir} varied from 4 to 12. In Fig. 5.18, we find that increasing N_{dir} does improve the localization accuracy, but not significantly. However, the cost of APDV in terms of unit-steps of the localizers increases substantially for higher N_{dir} values.

Then, we look at the performance of APDV when the sensor robot follows the trajectory TR_1 and values of N_{dir} are varied. Again, we find that the performance fluctuates between various N_{dir} values instead of showing a definite trend as with the static sensor robot positions. The reason is similar to the one in the case of step-size threshold. For example, in Fig. 5.19(b), we see, initially, that APDV performs better with $N_{dir} = 8$ compared to when $N_{dir} = 4$. However, between steps 14 and 18, APDV with $N_{dir} = 4$ performs better due to the current positions of the localizers. Again, the large variances in the graphs of Fig. 5.19 alludes to the localization error being *relatively large* in a *small* number of sensor robot positions for an arbitrary clutter topology.



(a) Localization/Distance Error (b) Comparison of $N_{dir} = 4$ and $N_{dir} = 8$

Figure 5.19: Effect of N_{dir} on the performance of APDV for a sensor robot trajectory. The clutter topology with initial localizer positions and the mobile sensor robot trajectory are as in Fig. 5.2(a) and Fig. 5.3(a).



(a) Localization/Distance Error (b) Total number of unit steps

Figure 5.20: Evaluation of APDV for varying numbers of localizers given static sensor robot positions. The clutter topology C is shown in Fig. 5.2(a). For each sensor robot position, random positions are generated for the selected number of localizers.

5.3.2.3 Number of Localizers N_l

Finally, we investigate the effect of the number of localizers N_l on the performance of APDV. We start with an evaluation of APDV for independent static sensor robot positions in clutter topology C. We find that the quantity of localizers has a straightforward impact on the performance of APDV, as shown in Fig. 5.20. When N_l is 2 or 4, the localization error is high. The reason is that the relatively few localizers are not sufficient for the four anchors and thus in such a situation some of the anchor distances would invariably be NLOS distances. The performance improves when N_l is 8 and stabilizes when 12 localizers are used. We don't see much improvement in APDV's performance

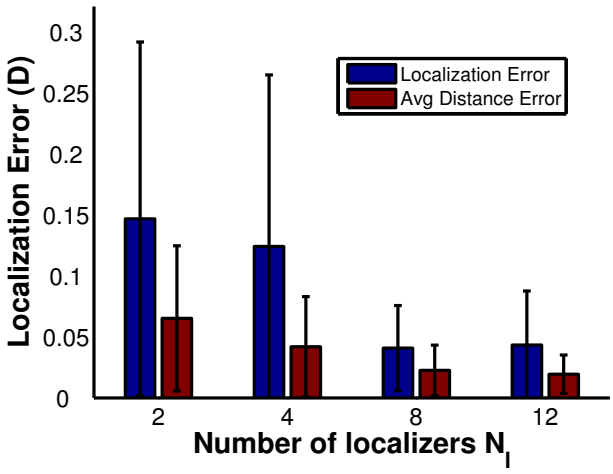


Figure 5.21: Effect of the number of localizers on the performance of APDV when the sensor robot moves in the pre-defined trajectory TR_1. The initial positions of the localizers are randomly generated for various values of N_l.

when $N_l = 16$ as we find that 12 localizers are sufficient for assisting a sensor robot in an arbitrary position in the given clutter topology.

Fig. 5.21 shows the effect of varying the number of localizers, from 2 to 12, on the performance of APDV algorithm on a sensor robot moving in the trajectory TR_1. As expected, the distance errors and the localization error both decrease as we introduce more localizers. The variance in the localization error along the entire sensor trajectory also decreases with increasing N_l. Unlike $\mathcal{S}_T$ and N_{dir}, we find that increasing N_l is the most consistent method to improve the performance of APDV even when the sensor robot is following a trajectory and previous positions of localizers can affect the performance of subsequent applications of APDV. The reason for this is that having more localizers in the clutter environment increases the chance that the localizers would be in the 'right' initial positions when the sensor robot initiates APDV.

5.3.3 Effect of Environment/Hardware Features

In the previous section, we looked at the effect of APDV parameters on the performance of APDV. In this section we explore some of the external factors that can influence the working and effectiveness of APDV in cluttered environments.

APDV makes two key assumptions: first, that LOS distances contain only small errors; and second, that the robot odometry errors are small and negligible. The first assumption is critical for APDV to enable a localizer to move in

the right direction that reduces its neighbour distances the most, thus reducing multi-hop error. LOS errors can also increase the completion time of APDV as localizers do not move into optimal positions. LOS errors also lead to large underestimates in the final multi-hop distances between the anchor and sensor robot. We find evidence for this in the APDV implementation on real robots, as will be discussed in the next chapter. The second assumption of APDV is important as it ensures that a localizer moves exactly into the position of the direction it evaluates as optimal. Even small positional mis-alignments can result in NLOS distances between neighbours, even though previously, during the exploratory phase of APDV, the localizer had LOS distances to them. This underlines the importance of a localizer being able to move back to its original position after all exploratory steps in an alignment procedure, or move to a new position in the optimal direction.

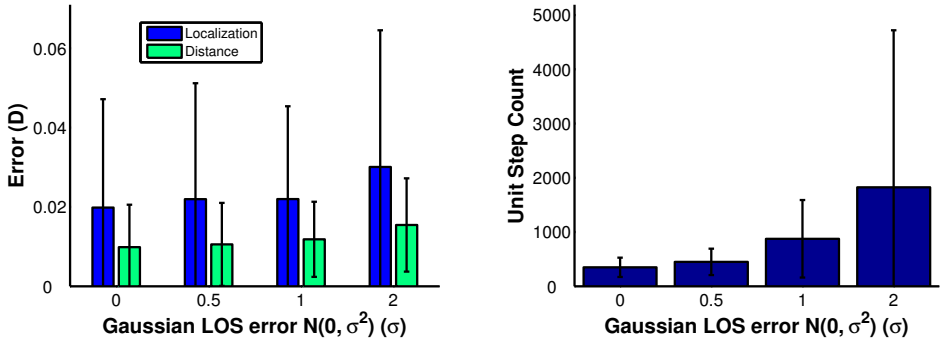
We begin by looking at the effect of LOS distance error, which is modelled as white Gaussian noise, on the localization accuracy and the cost of APDV. We then examine the effect of robot odometry errors, rotational and displacement, on the performance of APDV. Finally, we conclude the section by evaluating the effect of clutter topologies on the accuracy of APDV, comparing its performance to that of OPDV.

5.3.3.1 Line-Of-Sight (LOS) Distance Error

Line-of-sight (LOS) distance error is defined as the error in distances between nodes that are in line-of-sight with each other. They are typically small in value and are caused by measurement error, ephemeral equipment error, etc. These errors are usually modelled as a zero-mean Gaussian distribution $\mathcal{N}(0, \sigma^2)$, where σ is the noise standard deviation.

In order to reduce oscillations due to Gaussian LOS error, we introduce a parameter termed the *improvement threshold* t_{imprv}. The threshold t_{imprv} is defined as the minimum improvement that a new pair of neighbour distances should offer over the current pair of neighbour distances in order to replace them. For example, suppose $N_{dir} = 4$ and the EF variant is used and the localizer moves in N_{dir} directions measuring fresh neighbour distances (d_{ia}, d_{ib}) in the i^{th} direction ($1 \leq i \leq N_{dir}$) to neighbours a and b. For EF, we choose the optimal direction k such that

$$\arg\max_{k}((d_{ka} + d_{kb}) - (d_a + d_b)), \text{ where } ((d_{ka} + d_{kb}) - (d_a + d_b)) > 0$$



(a) Localization/Distance Error (b) Number of unit-steps

Figure 5.22: Evaluation of APDV for varying LOS distance errors given static sensor robot positions. The clutter topology C is shown in Fig. 5.2(a). For each sensor robot position, random positions are selected for the 8 localizers.

where d_a and d_b are the current distances to neighbours. However with this adaptation, we now choose the direction k such that

$$\arg\max_k((d_{ka} + d_{kb}) - (d_a + d_b)), \text{ where } ((d_{ka} + d_{kb}) - (d_a + d_b)) > t_{imprv}$$

It is important that the value of t_{imprv} reflects the LOS noise. We set the value of t_{imprv} as 6σ, where σ is the variance of the Gaussian LOS noise. Since a Gaussian distribution $\mathcal{N}(0, \sigma^2)$ draws a random value r such that $-3\sigma \leq r \leq +3\sigma$ with 99.7% probability, we can confidently assume that this overcomes the effect of the Gaussian noise in the sum of the two neighbour distances. The step-size s should also be set with LOS noise in mind. For practical reasons, we choose a step-size that is four times the maximum LOS Gaussian error, or $s = 4 * 3\sigma$. Thus, for $\sigma = 1$, we use a step-size of 12 units, such that 70% of LOS distances vary from 9 to 15 units, and for $\sigma = 2$, we use a step-size of 24 units, such that 70% of LOS distances vary from 18 to 30 units. On the other hand, this modification limits our selection of the step-size to large values which, in turn, can decrease the accuracy of APDV. A large step-size can increase the energy costs of the localizers.

Having discussed the implications of the LOS error on APDV, we can now evaluate the effect of LOS error for APDV. First, we look at the effect of LOS error on APDV in the static case, where APDV is run on single independent positions of the sensor robot. We randomly select $N_{sp} = 30$ positions for the sensor robot in the clutter topology C, and for each such sensor robot position, we further select 8 random (initial) positions for localizers. The APDV algorithm is thereafter deployed and the localization accuracy and cost in terms

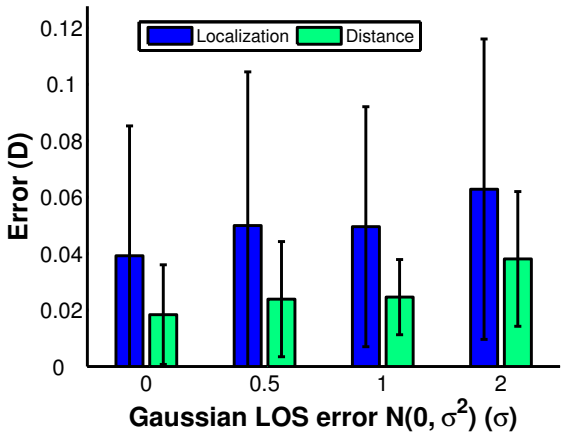


Figure 5.23: Effect of LOS distance error on APDV for the sensor robot trajectory TR_1. The clutter topology and the initial positions of the localizers are given in Fig. 5.2a.

of unit-steps is recorded. The results are shown in Fig. 5.22. We find that the LOS error does not have a significant impact on localization accuracy with average positional error only deteriorating to 0.03D even when LOS error σ is 2, compared to 0.02D when no LOS error is present. However, the cumulative number of unit-steps by all localizers increases by a factor of 16 for the same error when compared to the error-free case, as shown in Fig. 5.22(b). This underlines the debilitating effect LOS error has on the cost of APDV, despite the small decrease in localization accuracy. The increased localizer costs are due to the oscillations in the APDV iterations caused by the fluctuating distance errors. The localizers find it increasingly difficult to choose the optimal direction to move in, such that their neighbour distances are minimized. We find that though our values for t_{imprv} and s help in controlling localization error in adverse LOS conditions, they cannot prevent the localizer from taking many more steps than required in the error-free case.

Next we look at the influence of LOS distance errors on the performance of the ADPV algorithm for a sensor robot traversing the trajectory TR_1. The result is shown in Fig. 5.23. We see that, as expected, the distance and localization error increase as we increase the variance of LOS errors. However, we do find that, even with $\sigma = 2$, the average localization error increases by only 0.02D.

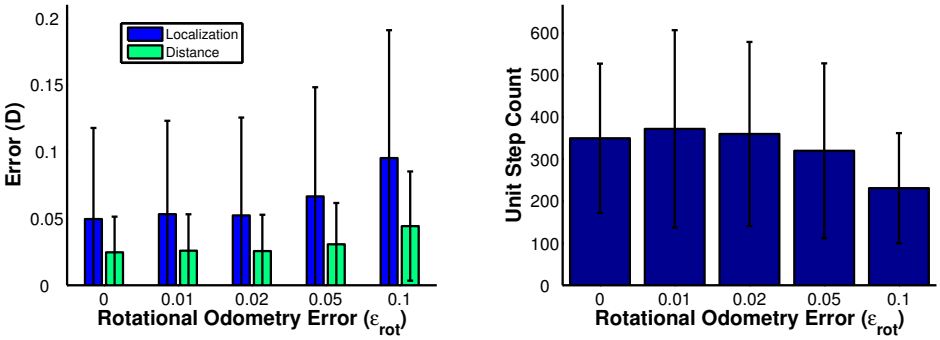
5.3.3.2 Robot Odometry Error

After looking at the effect of LOS distance errors, we now evaluate the effect of robot odometry errors on APDV. As pointed out earlier, robot odometry errors cause localizers to move to sub-optimal positions that can even result in NLOS distances to their neighbours. One possible solution is to record the neighbour distances of the original position connected to the optimal direction and check the distances again when the localizer finally (supposedly) moves in that direction. However, in this case, there is no feasible way to ensure that the original position is reached without incurring a large cost in terms of localizer locomotion.

We consider two types of robot odometry errors: rotational $\mathcal{N}(0, (\theta.\epsilon_{rot}^2))$, and displacement $\mathcal{N}(0, (d.\epsilon_{dsp}^2))$, where θ and d are the actual angle and displacement the robot is instructed to move respectively, and ϵ_{rot} and ϵ_{dsp} are the scale factors for the rotational and displacement odometry errors respectively. This offers a more realistic depiction of odometry errors than having a constant value for all rotations/displacements.

Since the alignment procedure entails the localizer to turn a certain angle and then move a certain displacement, the rotational odometry error has a higher impact on the robot's position and the effectiveness of APDV than the displacement odometry error. This is due to the fact that even small changes in the rotation angle lead to large variations in positions reached after the robot moves a certain distance in the selected direction. Typically this occurs when the robot has to turn multiple times in different directions and then select a certain direction and again move forward in it. Thus, it is possible that the position obtained in the latter displacement is not the same as the original (intended) position. This discrepancy worsens as the step-sizes of the alignment procedure increases. It is possible that the new position occludes the localizer from its two neighbours and results in NLOS distance errors in the multi-hop localizer chain. This may even lead to the sensor robot accepting the single-hop NLOS distance to the anchor eventually!

We now look at the effect of rotational and displacement odometry errors on APDV. As in the previous evaluation of APDV for LOS distance errors, we first evaluate APDV's performance of independent, static positions for the sensor robot. Thereafter we examine their effect on APDV algorithm for the scenario where the sensor robot moves along a trajectory.



(a) Localization/Distance Error (b) Number of unit-steps

Figure 5.24: Evaluation of APDV for varying rotational odometry error, given static sensor robot positions. The clutter topology C is shown in Fig. 5.2(a). For each sensor robot position, random positions are selected for the 8 localizers.

In our simulations, we confirm that the rotational odometry error does have a significant impact on the localization error of APDV. As shown in Fig. 5.24(a), the distance and positional error increase as the rotational odometry error increases, doubling in magnitude, when compared to the error-free case, for $\epsilon_{rot} = 0.1$. We find that the cumulative number of localizer unit-steps actually decreases as the rotational odometry error increases, as shown in Fig. 5.24(b). This is due to the fact that, as ϵ_{rot} increases, position mis-alignments increase and the localizers increasingly move to positions (in supposedly optimal directions) that result in NLOS distances to their adjacent nodes. Note that this is different from the scenario where a localizer is evaluating neighbour distances in various directions in an alignment procedure. Here, even if the localizer moves to a location that occludes it from its adjacent neighbours, it does return to its original position for a final consideration of all directions. With rotational odometry errors, the localizer moves into an incorrect position in the first place that may be very different from the position the localizer assumes it is moving into for the optimal direction.

Next, we see the effect of rotational odometry errors when the sensor robot is traversing trajectory TR_1 in the clutter topology C. We can see in Fig. 5.25 that the performance of APDV degrades substantially as the rotational odometry error increases to prohibitive levels. Thus, it is vital for us to rein in rotational odometry errors for APDV to achieve an acceptable performance. The large variances in Fig. 5.25 point to the localization error being relatively large in a small number of sensor robot positions for an arbitrary clutter topology.

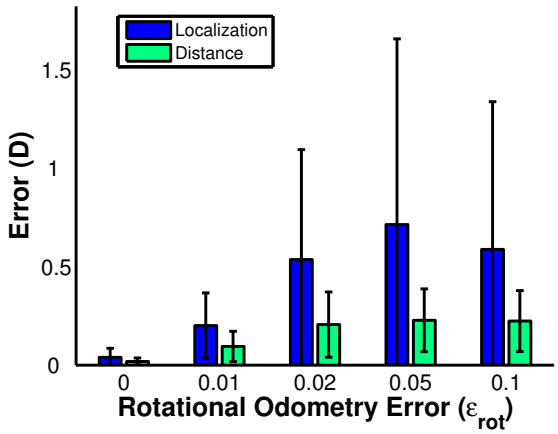
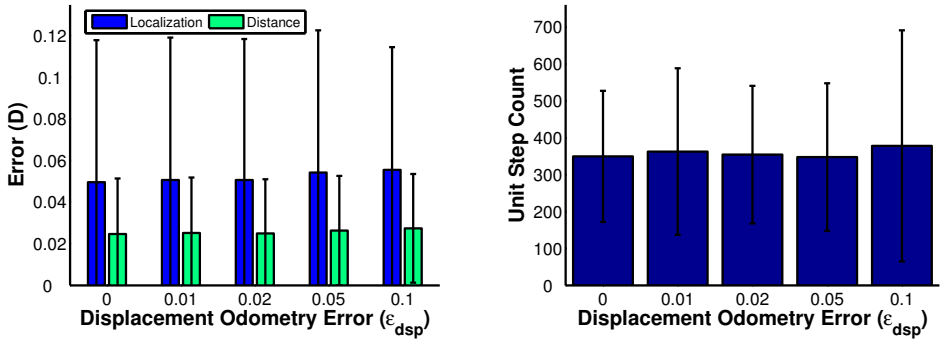


Figure 5.25: Effect of rotational odometry errors on APDV for a sensor robot moving through the pre-defined trajectory TR_1.



(a) Localization/Distance Error (b) Number of unit-steps

Figure 5.26: Evaluation of APDV for varying displacement odometry errors, given static sensor robot positions. The clutter topology C is shown in Fig. 5.2(a). For each sensor robot position, random positions are selected for the 8 localizers.

Unlike the rotational odometry errors, displacement odometry errors do not have a large impact on the performance of APDV, especially when the EF variant is used. The reason is that, since the error is based on displacement, it does not magnify the misalignment in positions to the extent seen in rotational odometry errors. For example, consider a localizer that has to move distance d from its current position to a new position P. In one scenario it has to turn an angle θ to the direction of P and move distance d. If P' represents the final position of the localizer after moving d, the position misalignment is $|P - P'|$, for a rotational odometry error e_{rot} for a turn of θ. For the second scenario, suppose the localizer has to move distance d in its current direction,

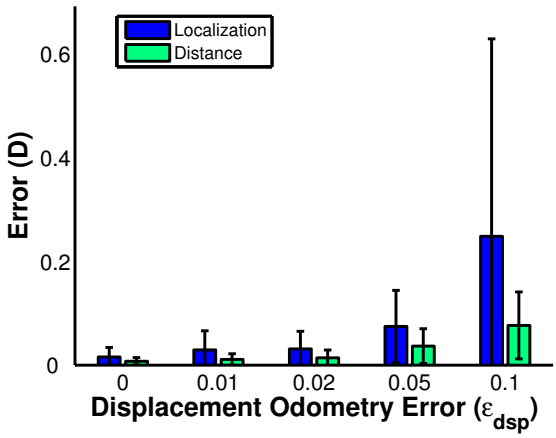
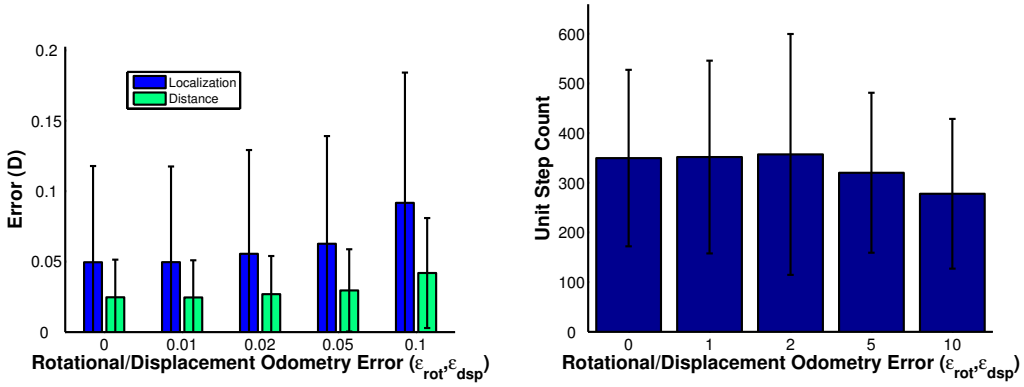


Figure 5.27: Effect of displacement odometry errors on APDV for a sensor robot moving through the pre-defined trajectory TR_1.

with P'' representing the erroneous position the localizer moves into and e_{dsp} the displacement odometry error. Clearly, $|P - P''| = e_{dsp}$ while $|P - P'| >> e_{dsp}$, even if $\epsilon_{rot} = \epsilon_{dsp}$. In this way, we can understand the variation of the performance of APDV for rotational and displacement odometry errors. Fig. 5.26 shows that the localization accuracy of APDV is not affected much by increasing displacement odometry errors. In fact, for individual sensor robot positions, the localization errors increase by as little as 0.01D for $\epsilon_{dsp} = 0.1$. The number of unit-steps taken by the localizers slightly increases since there may be cases where a localizer has not moved the entire distance to the optimal position between its neighbours and thus an additional APDV iteration is performed. A similar situation arises when the localizer overshoots the optimal position by a small distance. Fig. 5.27 shows that the displacement odometry error does not have a big impact on APDV for small values of ϵ_{dsp}, unlike rotational odometry errors. The localization error only increases from 0.04D to 0.08D for both $\epsilon_{dsp} = 0.01$ and $\epsilon_{dsp} = 0.02$, while in the case of rotational odometry errors, it increases to 0.2D for $\epsilon_{rot} = 0.01$ and to 0.54D for $\epsilon_{rot} = 0.02$.

Figs. 5.28 and 5.29 show the results for the scenario where robot locomotion suffers from both rotational and displacement odometry errors at the same time. Here, we consider equal values for both types of odometry errors, i.e., $\epsilon_{rot} = \epsilon_{dsp}$. In Fig. 5.28, we see that the effect of rotational odometry errors dominates that of displacement odometry errors and the localization accuracy of APDV is at least as bad as the case with only rotational odometry errors if not worse. This conclusion also applies to the dynamic scenario, as shown



(a) Localization/Distance Error (b) Number of unit-steps

Figure 5.28: Evaluation of APDV for varying rotational and displacement odometry errors, given static sensor robot positions. The clutter topology C is shown in Fig. 5.2a. For each sensor robot position, random positions are selected for the 8 localizers.

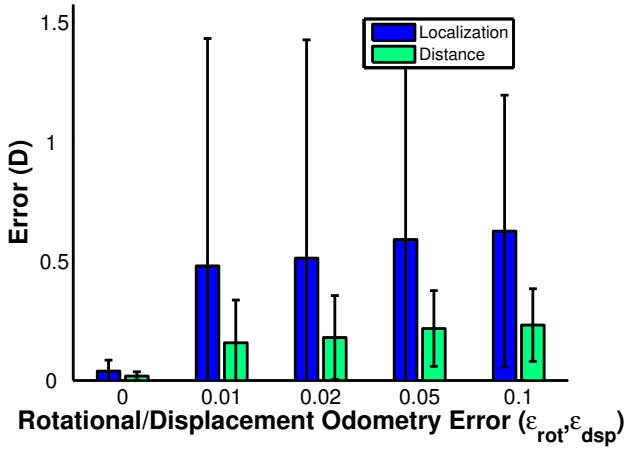
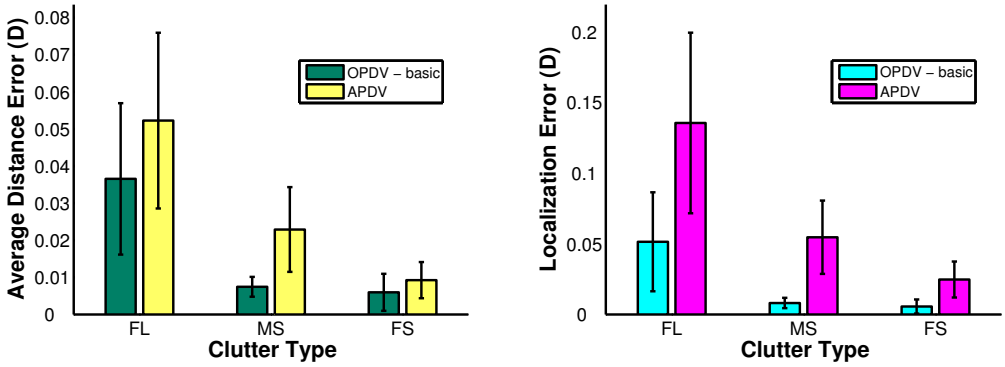


Figure 5.29: Effect of rotational and displacement odometry errors on APDV for a sensor robot moving through the pre-defined trajectory TR_1.

in Fig. 5.29. The large variances in Fig. (5.29) point to the localization error being relatively large in a small number of sensor robot positions for an arbitrary clutter topology.

5.3.3.3 Clutter Topology

Finally, we look at the effect of clutter topology on APDV. We evaluate its performance against the three clutter classes we defined in Chapter 3, namely the FL class, the MS class and the FS class. Recall that these classes are defined based on the size and number of clutter objects in the clutter topology.



(a) Distance Error (b) Localization Error

Figure 5.30: Comparison of APDV and OPDV for various clutter classes. OPDV represents the lower bound for the distance error when multi-hop localization is used in a clutter-prone environment.

Fig. 5.30 shows the performance of APDV for each of these three clutter topology classes and compares its performance with that of OPDV. OPDV can be said to represent the lower-bound for the multi-hop distance error. For the evaluation, 30 clutter topologies are used for each clutter class with 30 sensor robot positions being evaluated for each clutter topology instance. 8 localizers, with randomly generated positions, are used for the evaluation. OPDV and APDV are applied for each position of the sensor robot. We see that in general APDV follows the trend of OPDV, performing the worst with the FL class, due to the large sized clutter objects that force a larger overestimate in the multi-hop distances, and the best for the FS class which has small clutter objects with fairly open spaces between them. In the latter case, it is possible to achieve accurate multi-hop localizer chains, with minimal overestimate error, that weave through the clutter. We see that the performance of APDV is comparable to that of OPDV for all three clutter classes, having a distance error of at most 0.02D worse than that of OPDV. We see that APDV can outperform the single-hop localization techniques of LLSE and UBLSE, as shown in Fig. 5.7.

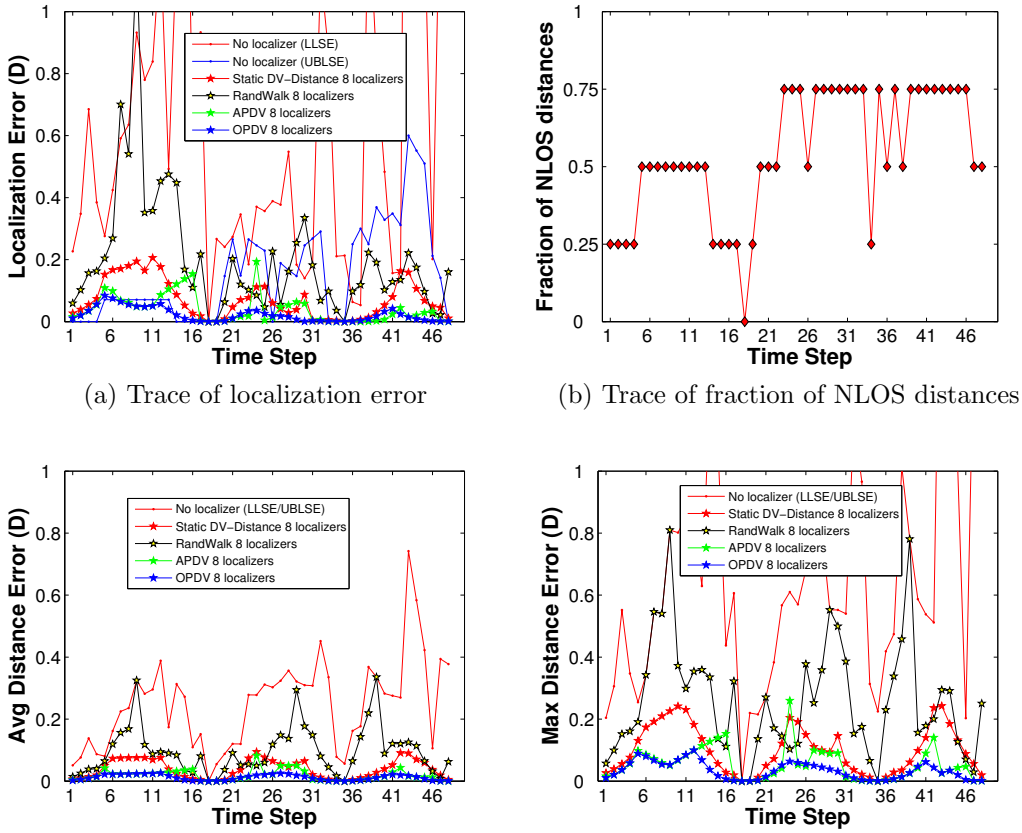
5.3.4 Comparison with Competing Techniques

In the previous section, we performed an extensive evaluation of APDV under varying values of APDV parameters as well as varying environmental and hardware factors. We saw that APDV is affected by external conditions such as LOS distance error, robot odometry error and clutter topology class. In

this section, we will compare APDV with various competing techniques enumerated in Section 5.1.4. We show that despite being a distributed algorithm with no requirement of global clutter topology information, APDV is able to perform well especially when the NLOS distances constitute the majority of all distances to anchors.

We compare the performance of APDV with two single-hop localization techniques (LLSE and UBLSE), with OPDV, and with DV-Distance without careful localizer placement (static DV-Distance and random walk). We use two different clutter topologies C and H as shown in Fig. 5.2, with the sensor robot following the pre-defined trajectories TR_1 and TR_2 respectively. The sensor robot trajectory is discretized such that the sensor robot localizes at distinct points on the trajectory (the sensor robot can be assumed to be stationary for a certain time period at those points in the trajectory). The two clutter topologies are representative of clutter class FL. The mobile trajectory and the placement of anchors are chosen so as to bring out the diversity of conditions regarding the placement of the sensor robot with respect to the anchors in the cluttered environment. Thus, in clutter topology C with TR_1, the sensor robot is less likely to have access to a majority LOS distances to anchors than in clutter topology H with TR_2. The algorithms' performances are evaluated using three metrics: localization error, average distance error and maximum distance error. We look at the traces of these metrics as the sensor robot moves in its assigned trajectory.

First, let us look at the performance evaluation in clutter topology C with trajectory TR_1. Four anchors are placed at the four corners of the square enclosure with eight localizers being placed at initial positions shown in Fig. 5.2(a). As the sensor robot moves through the clutter, it is occluded from a different number of anchors by the clutter at various points in its trajectory. For instance, initially it is occluded from just one anchor, but as the sensor robot moves ahead in the pre-defined trajectory an increasing number of anchors get occluded from the sensor robot. Only at a single point ([35, 75]) do all anchors become line-of-sight with the sensor robot. When APDV is used, the localizers move accordingly to assist the sensor robot to localize with better localization accuracy. When the sensor robot moves to a new position, the localizers move from their previous positions to assist the sensor robot. Localizers only move if they participate in a multi-hop chain between an anchor and the sensor robot.



(a) Trace of localization error

(b) Trace of fraction of NLOS distances

(c) Trace of average distance error

(d) Trace of maximum distance error

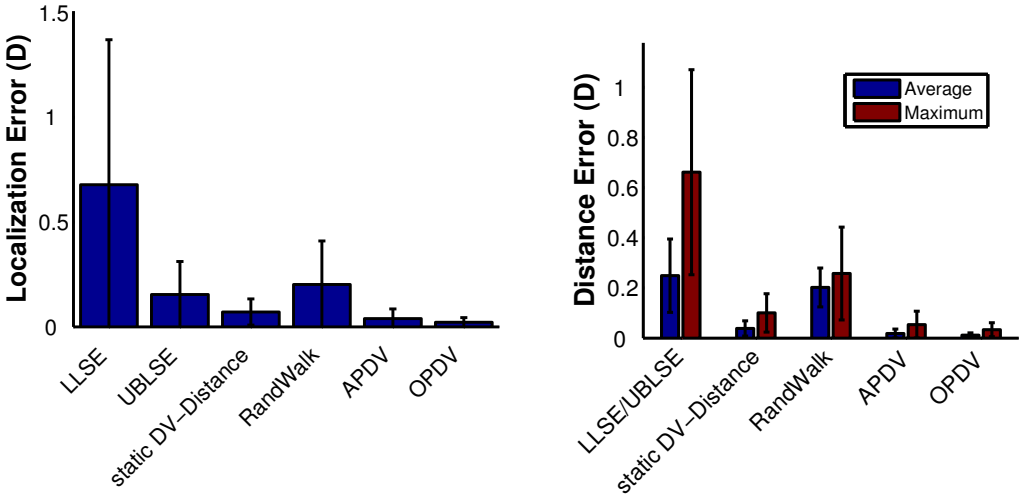
Figure 5.31: Trace of various performance metrics for the sensor robot along the trajectory TR_1 in clutter topology C.

A localizer is allowed to assist the sensor robot with respect to only a single anchor at a given time.

Fig. 5.31 shows the results of localizing the sensor robot along the trajectory TR_1 in clutter topology C. The x-axis shows time as the mobile sensor moves along its trajectory. The y-axis measures the localization error of the various techniques as a proportion of the length of the diagonal D of the square enclosure. Fig. 5.31(a) clearly shows that the LLSE approach is not practical in cluttered environments, since reflected NLOS distance measurements lead to enormous localization errors. UBLSE, however, performs very well during the early phase while degrading in performance later on. This is due to the fact that initially the sensor robot had only one NLOS distance, while later on the robot has at least two NLOS distances and thus LOS distances no longer form the majority of the available distances. In general, UBLSE performs well when NLOS distances form at most half of all distances to anchors. The localiza-

tion accuracy is significantly improved when fixed localizers are used in *Static DV-Distance*. However, this algorithm suffers when localizers do not happen to form tight multi-hop chains around the clutter, or use NLOS distance measurements among themselves. We use the constrained-localizer version of the OPDV algorithm with $N_{loc} = 8$ to establish the error lower bound for localizer placement when 8 localizers are used. We can see that using localizers placed at optimal positions in the clutter topology significantly reduces localization error. OPDV's performance benefits are most pronounced when NLOS distances constitute most or all distances to anchors. Finally, we look at the performance of APDV and find that it performs well in *most* cases by dynamically moving localizers to form tight localizer multi-hop chains between the sensor robot and the anchors. Thus APDV, by adjusting the positions of localizers amidst the clutter to address the zigzag overestimate and NLOS distance problems, outperforms Static DV-Distance by up to an order of magnitude. Using localizer placement based on simple random walk does not offer much benefit, even over static DV-Distance. This proves that only by carefully placing localizers using the metric of multi-hop distance, as used in OPDV and APDV, yields significant improvements in localization accuracy. Multi-hop localizer chains are too complex to optimize for random walk-based localizer placement especially when more than one localizer is involved in the multi-hop chain. Besides, random-walk placements require increased numbers of anchor adverts and DV-Distance control messages, since we would typically require at least one anchor advert for *every* random move of a localizer.

Fig. 5.32 allows us to compare the average performance of algorithms for the entire trajectory. While LLSE is clearly unsuitable to be used in a NLOS situation, UBLSE performs much better as a single-hop localization technique giving positional error of 0.15D on average even though the distance errors are of magnitude 0.35D. Static DV-Distance performs well compared to single-hop localization giving a localization error of 0.07D and a distance error of 0.05D on average. However, it must be pointed that these results of Static DV-Distance are valid for only the scenario where the localizer positions are fixed as shown in Fig. 5.10(b). They may vary for other random (static) localizer placements. We see that the performance of APDV is the best among practical algorithms and not far from the oracle-type OPDV algorithm. The average localization error in APDV is 0.04D compared to 0.02D for OPDV, while the average distance error in APDV is 0.03D compared to 0.02D in case of OPDV.

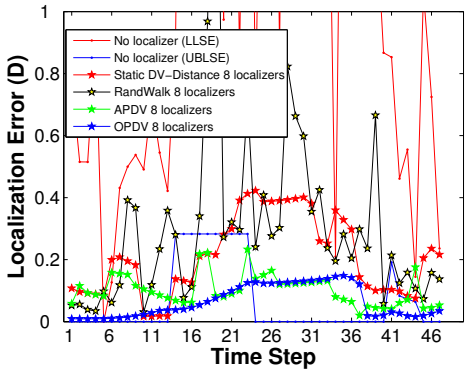


(a) Localization Error (b) Distance Error

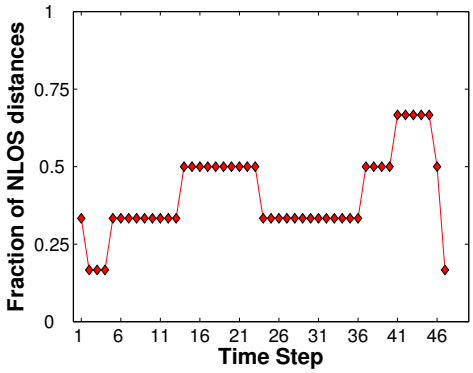
Figure 5.32: Comparison of localization and distance error between single-hop techniques, static DV-Distance, OPDV, APDV and random-walk placement techniques for trajectory TR_1 in clutter topology C.

Next, we consider the performance of all the aforementioned localization techniques for a sensor robot moving in a trajectory TR_2 in the clutter topology D. The results are shown in Fig. 5.3(b). This setup differs from that with clutter topology C and TR_1, in that here a larger number of the sensor robot positions have majority LOS distances to the anchors. This is shown in Fig. 5.33(b). Thus, we see that UBLSE performs very well, yielding a position estimate with zero error for a large number of sensor robot positions. Its performance degrades when the fraction of NLOS distances are 0.5 or more. We find that the performance of DV-Distance with the static positions of the localizers, as shown in Fig. 5.2(b), is poor compared to the previous scenario with clutter topology C and TR_1. The reason is that the (random) localizer placement does not yield similar benefits in localization accuracy as with the previous scenario. The same can be said when the localizers take random walks in hope of good placements.

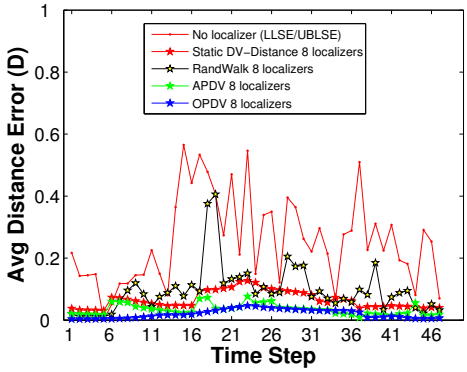
When looking at the trace of average distance error of the sensor robot along trajectory TR_2, we find that there are instances when OPDV shows more errors than APDV, even though we consider OPDV to be a lower bound for the localizer placement error. This can be explained by the manner in which we define the constrained-localizer version of the OPDV algorithm. Here, the total number of localizers (in this case 8) are divided equally among the anchors,



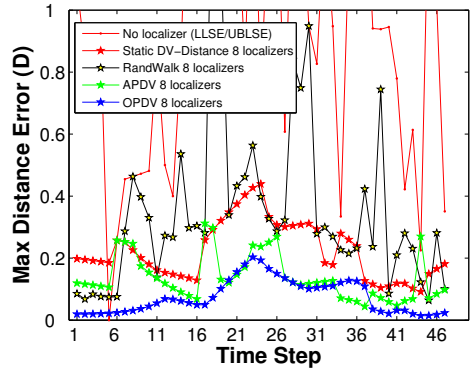
(a) Trace of localization error



(b) Trace of fraction of NLOS distances



(c) Trace of average distance error



(d) Trace of maximum distance error

Figure 5.33: Trace of the distance error at the mobile sensor robot for each step along the trajectory TR_2. In case of APDV, it is the distance error at the robot at the conclusion of APDV for each step.

which in turn serves as the localizer limit for the individual OPDV localizer chains. This may lead to sub-optimal solutions when considered as a complete problem, though individually the outputted localizer chains are optimal for their respective localizer limits. This problem, however, does not arise when the basic (unlimited-localizer) version of OPDV is used, as shown in Fig. 5.34. Here, we find that OPDV always provides the lower bound for the localizer placement error.

We compare, in Fig. 5.35, the various localization techniques in terms of average localization and distance errors for trajectory TR_2 in clutter topology H. We see the LLSE performs poorly in the NLOS-prone environment making it unsuitable for such conditions. UBLSE performs very well since a large number of points along the sensor robot trajectory get LOS distance measurements. As with trajectory TR_1 in Clutter Topology C, we find that OPDV and APDV

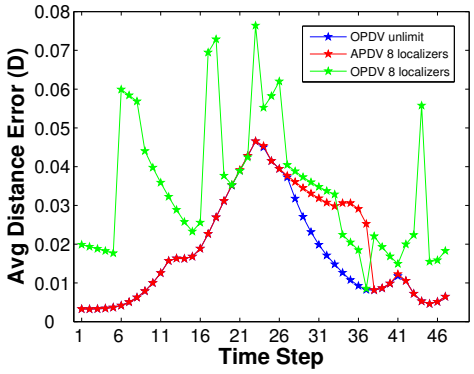
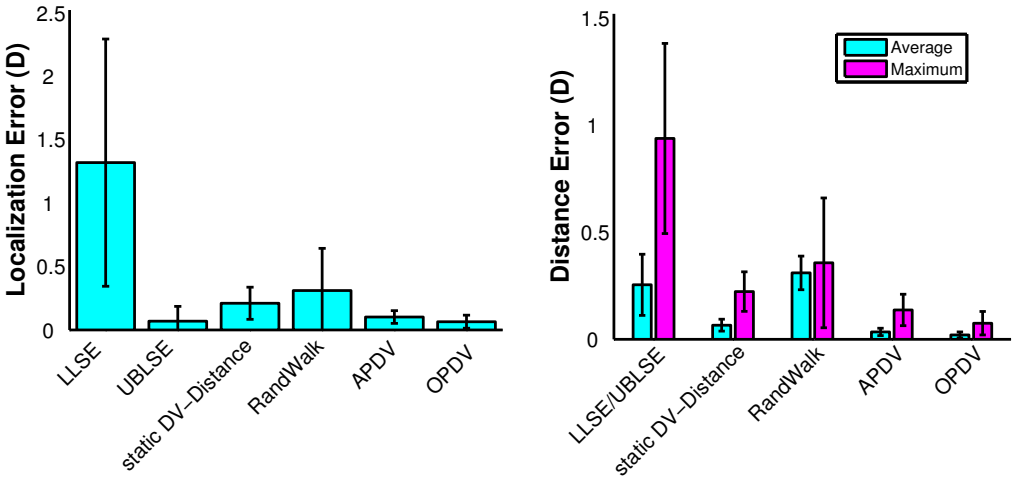


Figure 5.34: Comparison of unlimited localizer version of the OPDV algorithm and the localizer-constrained version of the OPDV algorithm with $N_{loc} = 8$. Here we find that only the unlimited localizer version of OPDV does provide the true lower bound for APDV.



(a) Localization Error (b) Distance Error

Figure 5.35: Comparison of localization and distance error between single-hop techniques, static DV-Distance, OPDV, APDV and random-walk placement techniques for trajectory TR_2 in clutter topology H.

outperform both static DV-Distance and random walk placement by a factor of two. We also find APDV's performance comparable to that of OPDV.

5.4 Discussion

In this chapter we performed an extensive study to analyze the performance of the proposed localizer placement algorithms, namely OPDV and APDV. We saw that, while OPDV can be impractical to implement in realistic environ-

ments because of its strong assumptions, it nevertheless serves as a useful tool to evaluate the lower bound for the multi-hop error in localizer placement. We looked at the performance of OPDV as we varied parameters such as grid spacing, communication range and number of available localizers. We concluded that clutter topology plays an important role in determining the localization accuracy of OPDV. We compared the performance of OPDV with random placements for DV-Distance and showed that OPDV can achieve substantial benefits in localization accuracy with only a fraction of the localizers required in the case of random placements. Thereafter, we looked at the performance of APDV for various APDV parameters as well as external factors like LOS error and robot odometry error. We concluded that, while the step-size threshold $\mathcal{S}_T$ and the number of probing directions N_{dir} do not significantly/consistently impact the localization accuracy of APDV, the number of localizers N_l in the cluttered environment has a definite, positive impact on the performance of APDV. We evaluated the performance of APDV under differing conditions of LOS distance error and robot odometry errors. We observed that it is vital to have an improvement threshold attuned to the LOS noise in order to overcome the oscillations in the APDV algorithm due to LOS error. We showed that rotational odometry errors have a much more debilitating effect on APDV when compared to displacement odometry errors. Finally, we compared the performances of APDV with a number of competing localization algorithms, and showed that APDV performs remarkably well compared to others, especially in situations where NLOS distances form the majority of distances to anchors.

In the next chapter, we will describe the implementation and evaluation of APDV on a real robotic platform. We use MIT-Cricket motes for distance measurements and the Roomba robot platform for demonstrating the efficacy of the APDV algorithm in realistic settings.

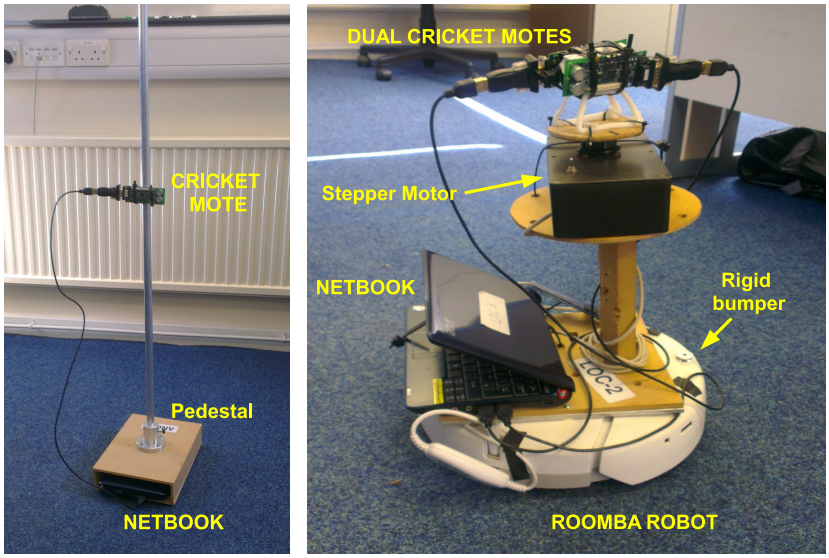
Chapter 6

Robot-Based Evaluation

In the previous chapter, we performed an extensive simulation-based evaluation of the APDV algorithm under a variety of APDV parameter values and external environmental/hardware conditions. In this chapter we evaluate an implementation of the APDV algorithm on real robots. We use the iRobot Roomba robotic platform for implementing the mobile localizers, and employ MIT-Cricket motes for the purpose of distance measurements between the anchors, localizers and a sensor robot. We describe the hardware and software implementation in Section 6.1. Here we discuss the various components of distance measurements, control communication and robot locomotion. Then, in Section 6.2, we report on the experimental results obtained using the robotic platform with a variety of clutter topologies and anchor-localizer placements.

6.1 APDV Implementation

In this section, we will describe the APDV implementation on a real robotic platform. We begin by describing the Roomba robotic platform in Section 6.1.1.1. We then discuss the distance measurement process using the MIT-Cricket motes, looking at both the basic Cricket mote operation and the omni-directional setup constructed for our experiments, in Section 6.1.1.2. Finally, in Section 6.1.2, we go over the software architecture of the APDV implementation, and look at how various software components communicate with each other and their respective hardware units.



(a) Anchor hardware (b) Localizer hardware

Figure 6.1: Hardware platform for anchor and localizer/sensor robots.

6.1.1 Hardware Architecture

Two types of nodes have been constructed for our experiments: anchors and robots. We have built a mobile robotic platform using iRobot Create robots which are equipped with MIT Cricket motes [105] for the purposes of distance measurement between nodes, as shown in Fig. 6.1(b). The robot can represent either a localizer or a sensor robot. Fig. 6.1(a) shows an example of an anchor, constructed using a single Cricket mote mounted on a pedestal.

6.1.1.1 Roomba Robot

We have used the hobbyist Create version of the Roomba for our experiments, which has been explicitly designed for robotics development. In place of the vacuum hardware of the original Roomba, the Create instead has a cargo bay with a 25 pin port that can be used for digital and analog input and output. The purpose of the cargo bay is to enable robotic accessories like robotic arms be placed on the Create. Alternatively, the Create possesses a serial port through which sensor data can be read and motor commands can be issued using the iRobot Roomba Open Interface protocol. For our experiments we mount a netbook on the cargo bay which runs the APDV algorithm and interfaces with the Create through the serial port. The Create possesses a number of sensors like the wheel encoders, bump sensors, an infra-red wall sensor and cliff sensors. The Create has two drive wheels (Left and Right). By using

various combinations of these two wheels it can move forward, backward or spin clockwise/counter-clockwise in its current position. The odometry errors of the Roomba family are found to be, in the case of naïve odometry, 11.5% for displacement odometry error (σ_{dsp}) and 12.4% for rotational odometry error (σ_{rot}). Tribelhorn et al. [127] propose a correction to the raw odometry readings which improves σ_{dsp} to 2.9% and σ_{rot} to 6.1%.

Obstacle detection is done using bump sensors. The two bump sensors are situated on the left and right sides of the single rigid bumper attached to the front of the Create. Fusing the data from the two bump sensors allows the Create to determine if it has bumped into an obstacle at the front of the bumper, at the left side of the bumper or at the right side of the bumper. The collision angle that will produce a front bump varies with the sampling rate — a 4Hz rate yields a range of 60° for the front bump [127]. Note that obstacles can be detected via 'bumps' only in the forward direction and not in the backward direction. Hence, if the Create robot encounters an obstacle while moving backwards, it merely attempts to continue moving unsuccessfully (leading it to rotate instead at the point of collision with the obstacle), which can lead to discrepancies in the odometry readings. For our experiments, this is not an issue since the clutter topology is static and the robot typically moves backwards only for retracing its forward displacement.

6.1.1.2 MIT Cricket Motes

The MIT Cricket motes use the principle of time-difference of arrival (TDOA) between a radio message and an acoustic chirp to calculate the distance. When the Cricket motes are face to face with each other, the maximum measurable range is around 12m. However, we found that the maximum range quickly decreases to 4-8m in the presence of obstacles, where the clutter topology and obstacle material play a vital role in determining the range. We chose MIT Cricket motes for our implementation for three reasons:

1. Line-of-sight (LOS) distances are accurate with a standard deviation of only 6 cm.
2. Ultrasound signals bounce off obstacles like wood and metal, to emulate reflected non-line-of-sight (NLOS) distance measurements, especially in closed small areas, instead of passing through the clutter. On the other

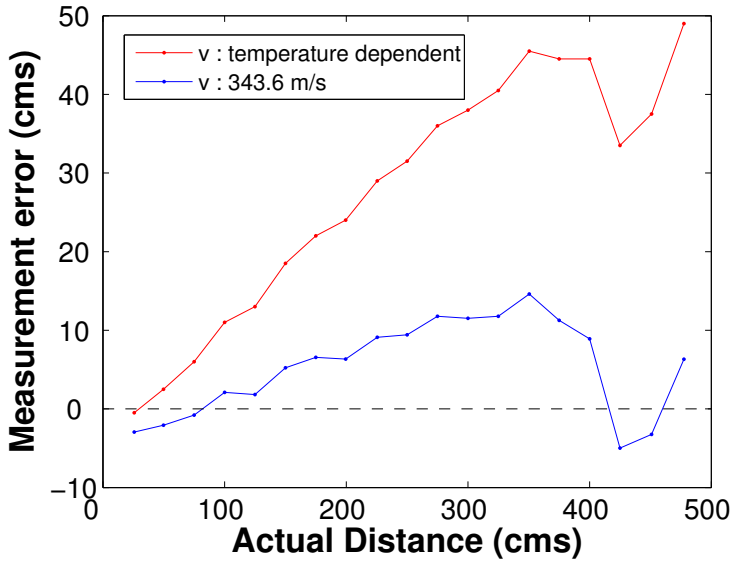


Figure 6.2: Comparison of accuracy of basic distance measurements between temperature-dependent sound velocity and a constant velocity of 343.6 m/s. The two Cricket motes are placed face-to-face.

hand, radio signals are able to pass through a number of materials which makes it much harder to emulate NLOS distances.

3. Cricket motes do not require to be time synchronized with each other. This saves the costly overhead of using a time synchronization protocol.

Basic distance measurements. A Cricket mote can be in two modes: a *beacon* mode, where it transmits a combination of radio and acoustic signals periodically; and a *listener* mode, where the Cricket mote listens for radio and acoustic signals and estimates ranges. Multiple listener Cricket motes can estimate their distance to a single transmitting beacon Cricket mote simultaneously. The Cricket mote has a radio transceiver operating in the 418 MHz unlicensed band and an ultrasound transceiver operating at 40 kHz. The radio on the Cricket mote is omni-directional while the ultrasound transceiver is directional. The signal strength of the ultrasound signal is maximum in the direction where the ultrasound transmitter is facing (0°), and the signal strength drops to 1% (-20 dB) of the maximum value of the signal strength at ±50° from the 0° direction. The velocity of sound in air is used to measure distances from the beacons to the receivers. The velocity of sound (v) depends on environmental factors such as the ambient temperature and humidity. The instantaneous velocity of sound is calculated using the formula

$v = (331.4 + 0.60 * T)/1000$ (m/s), where T is the ambient temperature. The temperature is periodically polled by the Cricket mote. The distance is then calculated as $t * v$, where t is the estimated travel time of the acoustic signal. In our experiments, we found that using the temperature dependent velocity for sound gives us larger errors than when a constant value of 343.6 m/s is used. Fig. 6.2 shows that the temperature dependent sound velocity is generally an overestimate of the true velocity of sound. For our experiments, we use $v =$ 343.6 m/s.

In the beacon mode, the Cricket mote repeatedly transmits a pair of acoustic and radio messages every 2000ms. In the receiver mode, the Cricket mote receives the radio message and the subsequent acoustic (chirp) message from a beacon and calculates the distance to the beacon based on the time difference of arrival between the radio and acoustic messages. The radio message contains beacon information such as beacon-ID and beacon coordinates. However, unlike the radio message, the acoustic chirp does not contain any information identifying its source. Every time a receiver Cricket mote receives a beacon radio message, it enables the ultrasound detector and starts a timer. Thus, it is possible that when two beacons are broadcasting, a receiver can receive a radio message from one beacon and then, immediately afterward, an acoustic chirp of another beacon. This typically leads to underestimations in the measured distances. Other reasons for erroneous distances could be environmental noise and non-line-of-sight propagation of the acoustic signals.

Omnidirectional distance measurements. Since the ultrasound transducers on the Cricket motes are not omnidirectional, we use a formation of two Cricket motes, arranged back to back, that can be rotated 360° using a stepper motor (shown in Fig. 6.1(b). If robot A wants to measure the distance to robot B, both robots together rotate the formation by 360°, with robot B being the Cricket beacon and robot A being the Cricket listener. This arrangement is devised so that a localizer can measure distances to both its neighbours simultaneously by being the beacon only once. Robot B then collects all the distance measurements from its two Cricket motes, and processes them to estimate the most likely straight-line distance between robots A and B. To do so, it sorts the distance measurements in ascending order and clusters them. It, then, takes the median of the first sufficiently large cluster, of cardinality greater or equal to parameter n_{sf}. The value of n_{sf} is determined by the number of collected distance estimates — for samples sizes < 3, n_{sf} is set to 1, otherwise n_{sf} is

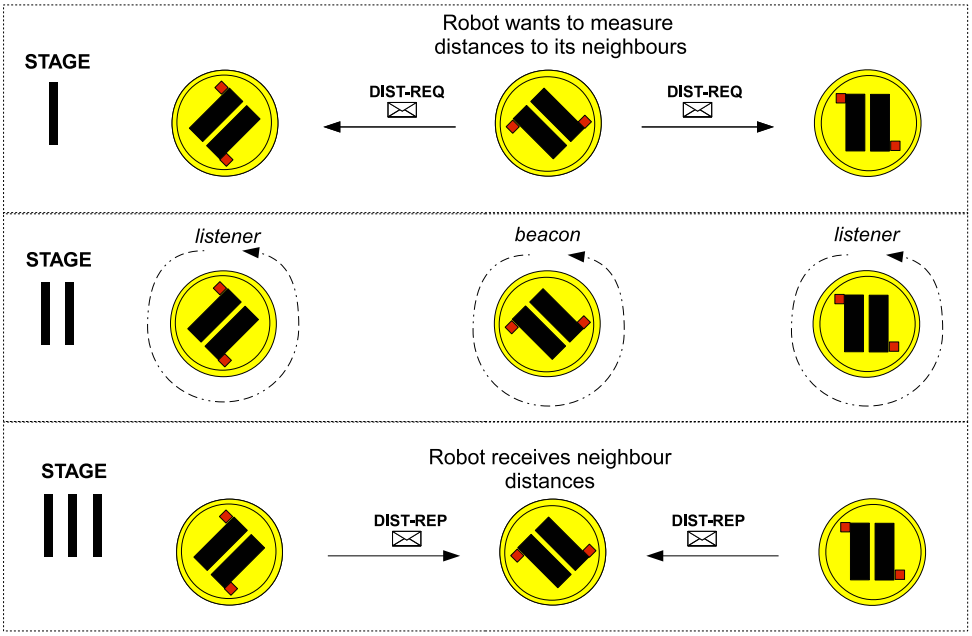
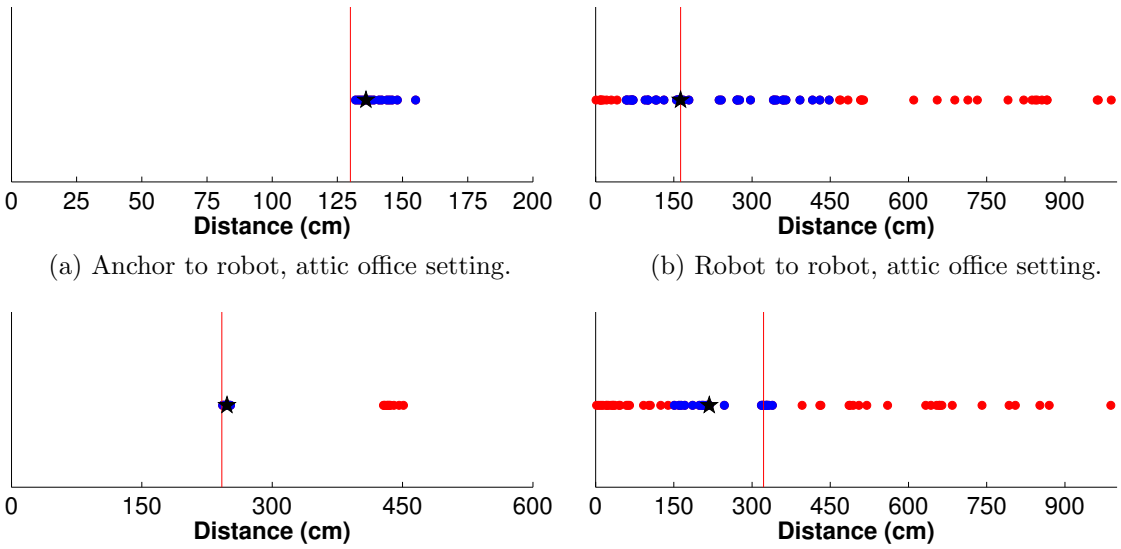


Figure 6.3: Robot measures distances to its neighbours simultaneously using a single operation. The robot becomes a Cricket beacon with its neighbours being Cricket listeners. The robot and its neighbours make a complete 360° so that at some point of the rotation the ultrasound transceivers of the robots are in line with each other to obtain the shortest straight-line distance.

set to 5. This is done to accommodate distance measurements where there is a large distance (500cm) between the ranging nodes and we are able to collect few distance measurements.

Fig. 6.3 shows an example of a robot measuring distances to its two neighbouring robots. The robot initiates the distance measurement procedure by sending a **DIST-REQ** message to its two neighbours. Then, the robot sets its Cricket motes to beacon mode and the neighbours to listener mode. Thereafter all three robots will rotate their Cricket motes a full 360° rotation. Finally, each of the two neighbours will estimate the most likely straight-line distance from the collected distances and inform this distance estimate to the robot using the **DIST-REP** message. Note that if the robot's neighbour is an anchor, the anchor remains stationary when taking distance measurements to the robot.

We now evaluate the accuracy of the omni-directional distance measurement mechanism. We performed a set of experiments in two settings at the Department of Computer Science, University of Oxford: one in the attic office room and the other one in the atrium. We evaluate the accuracy of distance measurements when a robot tries to estimate its distance to either an anchor



(a) Anchor to robot, attic office setting.

(b) Robot to robot, attic office setting.

(c) Anchor to robot, atrium setting.

(d) Robot to robot, atrium setting.

Figure 6.4: Evaluation of accuracy of omni-directional distance measurements in two settings: atrium and attic office space. The red vertical line represents the true distance. The entire set of distances collected during the omni-directional measurement process is shown with red dots, while the cluster chosen by the heuristic is represented with blue dots. The final estimate (median of the chosen cluster) is shown as a black star.

or another robot. The results are shown in Fig. 6.4. We see that in both settings the anchor-robot distances measurements are much more accurate than the robot-robot distance measurements, with the robot-robot ranging giving an error of almost 110 cm in Fig. 6.4(d). Robot-robot ranging is more complex than anchor-robot ranging since it involves two pairs of rotating Cricket mote formations. Since the Cricket motes measure distances using time-difference-of-arrival between radio and acoustic signals, the receiving Cricket motes receive a large variety of distances, both correct and incorrect, from the transmitting Cricket motes. This spread of distances can be seen in both Figs. 6.4(b) and 6.4(d). These figures also show that the environment has an influence on the spread and composition of distances. This, in turn, can have a significant impact on the performance of the heuristic used to select an estimate from the collection of distance measures. For example, we see that in the attic office setting, the distance collections between both robot-robot and robot-anchor have less spread, and thus the heuristic gives fairly accurate results. In the case of the atrium setting, we see, in Fig. 6.4(c), that another cluster of erroneous distances appear in the collection of distances between the robot and its anchor

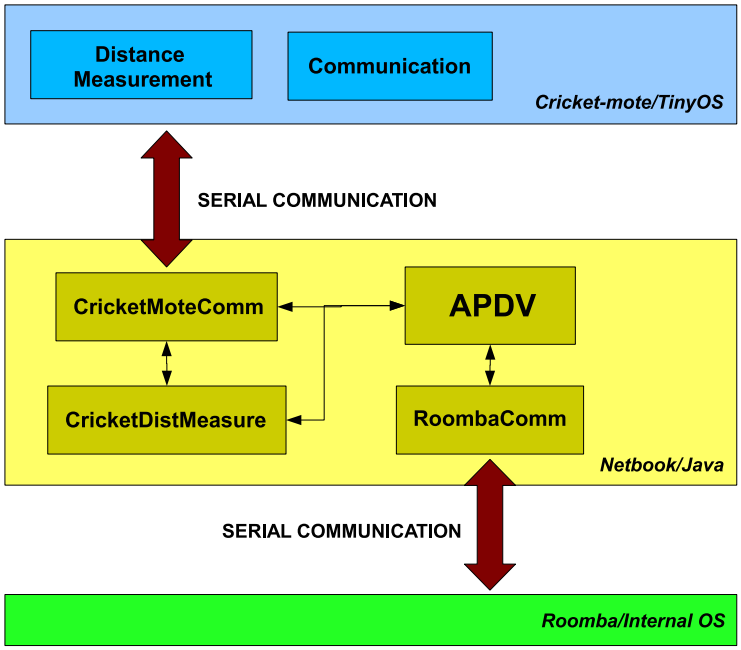


Figure 6.5: Software architecture of the APDV implementation. The three hardware blocks, namely the Cricket mote running Tiny-OS, the netbook running Java/Windows 7, and the Roomba robot are represented along with the interactions between these modules.

neighbour. This can be attributed to the position of the robot with respect to the anchor. Comparing the collections of distance measures for the robot-robot ranging in the two settings, we see that atrium setting yields more incorrect distance measures than the attic office setting.

6.1.2 Software Architecture

In this section, we describe the software architecture of APDV on the robotic platform. We will outline interactions between the three hardware units involved, namely the Cricket mote running the Tiny-OS [72] operating system, the netbook running Microsoft Windows and the Roomba robot, as shown in Fig. 6.5.

The Cricket mote is used for the purpose of measuring distances between the anchors and the robots (localizer and the sensor robot) as well as communication of control messages between the robots and the anchors. The Cricket mote was originally designed to operate in only the *beacon* and *receiver* modes. We modified the Tiny-OS code to enable the *standby* mode in which the Cricket mote can simply send and receive messages. To enable more beacon messages

to be sent during the omni-directional distance ranging process, we reduced the interval for a beacon to send the combination of radio and acoustic messages from 2000ms to $(664 + \Delta_c)$ ms, where Δ_c is a randomly chosen natural number between 0 and 668. Using serial computer communication over the UART port of the Cricket mote, messages can be transferred in both directions with the netbook. Thus, the Java program on the netbook can instruct the Cricket mote to send a message over the radio or move into any of the three modes of operation mentioned above. Similarly, the Cricket mote can report received radio messages as well as Cricket distance ranging measurements to the netbook. The stepper motor, on which the dual Cricket mote formation is mounted, is controlled directly from the netbook.

The main APDV algorithm is implemented as a multi-threaded Java program. One thread is used to monitor a message queue for messages from the Cricket mote and to process messages from it in a FIFO manner. Specific event handling code is provided for each message. The Cricket mote and Roomba communication are handled on distinct threads. The *CricketMoteComm* class deals with the serial communication with the Cricket mote using the Java SerialReader and SerialWriter classes. Separate instantiations of these classes are done for each of the two Cricket motes. The *CricketMoteComm* class is also responsible for collating the multiple distance measures received from the Cricket mote. The *CricketDistMeasure* class deals with the various procedures for distance measurement. It contains the heuristic code, mentioned in the previous section, for obtaining a distance from the collection of distance estimates received during the omni-directional ranging process. Yet another thread is used to operate the stepper-motor in the case of robots. In order to prevent inconsistency upon simultaneous updates and accesses of the program data structures by various threads, locks (namely the Java *ReentrantLock*) are used. The Java TimerTask class is used to provide timer functionality.

The Roomba robot is controlled from the Java program running on the netbook using the RoombaComm API [68], that provides a convenient encapsulation for the serial communication with the Roomba robot. The *RoombaOdometryControl* class is responsible for providing the functionalities of rotating and moving the Roomba robot. It allows for high-level operations such as making the Roomba robot retrace its last successful trip, moving the robot forward by x distance and returning it to its original position if it meets an obstacle mid-way, and rotating the robot by a certain angle relative to its current orientation.

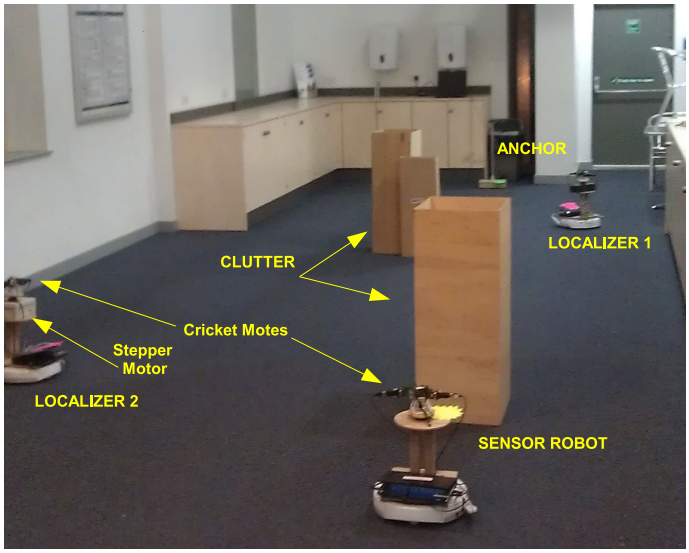


Figure 6.6: Experimental setup for two localizer configuration. The localizers (with red markers) form a multi-hop distance chain between the sensor robot (yellow marker) and the anchor (in the far end of the picture).

6.2 Evaluation

In this section, we evaluate the performance of the APDV algorithm in a real-world testbed setting. First we describe the experimental setup in Section 6.2.1. We then provide, in Section 6.2.2, an extensive evaluation of the APDV algorithm with three different clutter topologies and varying initial positions of the localizers.

6.2.1 Experimental Setup

The experiments were performed in the closed corner of the atrium in the Department of Computer Science, Oxford University, shown in Fig. 6.6. The experimental area is around 10m x 5m. We used custom-built plywood structures as obstacles since we found them to reflect ultrasound signal rather then letting them pass through. We evaluated the performance of only the EF variant of the alignment procedure, which was previous described in Section 4.3.2.2. We performed the experiments primarily to demonstrate that APDV successfully reduces the distance overestimates of the initial multi-hop distances overestimates.

6.2.2 Results

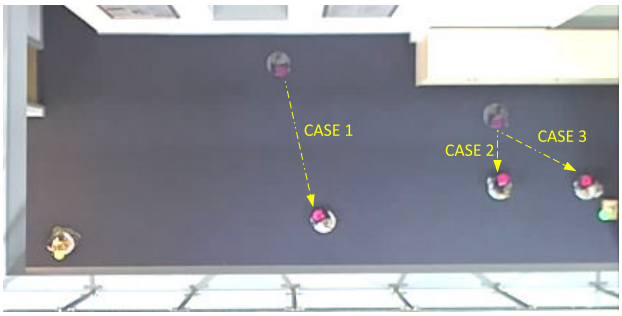
We evaluated APDV in four settings: no-clutter (Fig. 6.7(a)), clutter-1 (Fig. 6.7(c)), clutter-2 (Fig. 6.7(e)) and clutter-3 (Fig. 6.7(g)). Due to distance measurement errors, we notice that being close to the Euclidean line itself does not necessarily mean a lower distance error. Conversely, even though the localizer is further from the Euclidean line, we often find that the multi-hop distance is an accurate estimation of the Euclidean distance itself. We fix the number of probing directions N_{dir} to 4 throughout our evaluation. Thus, during a single APDV alignment procedure, a localizer moves 400cm when the step-size s is set to 50cm and 800cm when the step-size s is set to 100cm.

The results for the evaluation are shown in Fig. 6.7. We find that APDV is able to deliver reductions in the multi-hop error by almost 95% for some cases. We find that using a step size of 100 cm offers better accuracy (and faster completion times) than a step size of 50 cm, partly due to the LOS errors being closer in magnitude to the smaller step size.

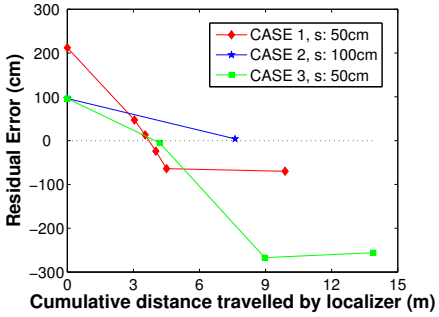
LOS errors. We find that LOS errors have a debilitating effect on the performance of APDV, especially when they lead to underestimated distances. For example, for CASE-3 in the no-clutter setting (Fig. 6.7(a)), the localizer settles at a position with a large underestimate error of 260 cms even though it is aligned closely with its neighbours. When compared to CASE-2, which uses a step size of 100 cm, the localizer moves almost twice the distance over multiple APDV iterations, only to settle with a large multi-hop error. One can see the effect of LOS errors for CASE-1 of clutter-3 settings in Fig. 6.7(h).

Interplay between multi-hop and LOS errors. We find that LOS underestimates and overestimating multi-hop distances have the net effect of reducing the overall multi-hop distance error. Thus, we can see that even if the localizers settle in a position that seems to have considerable multi-hop overestimate errors, the underestimate LOS errors help reduce the overall distance error. For example, for CASE-1 and CASE-2 of the clutter-1 settings, we find that the multi-hop distances are actually more accurate than what the final (physical) placements of the localizers would suggest.

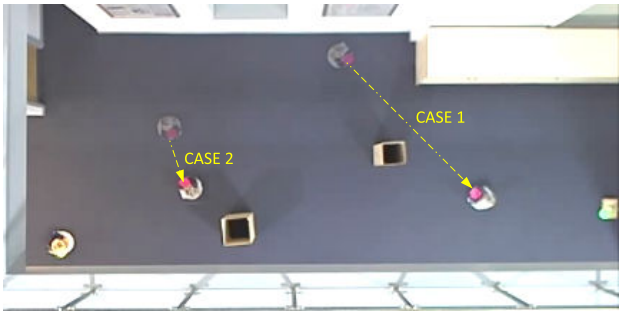
Temporal variation in LOS distances. We find that temporal variations in LOS error can affect the working on APDV. For example, for CASE-1 of the clutter-3 setting, we find that fluctuations in the multi-hop distances between the anchor and sensor robot lead to a longer completion time for APDV. The reason for this also lies in our implementation of APDV. When a localizer gets



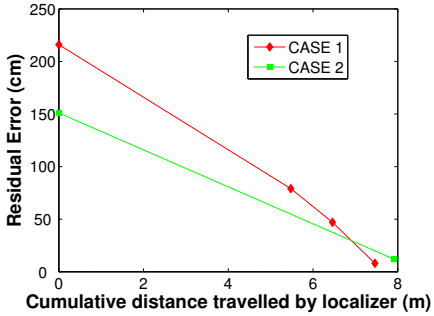
(a) 3 cases for no-clutter setting.



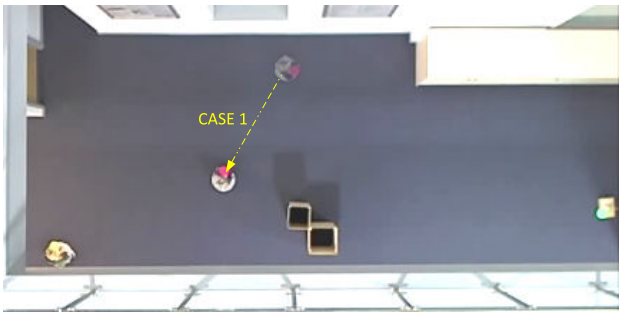
(b) Performance in no-clutter setting.



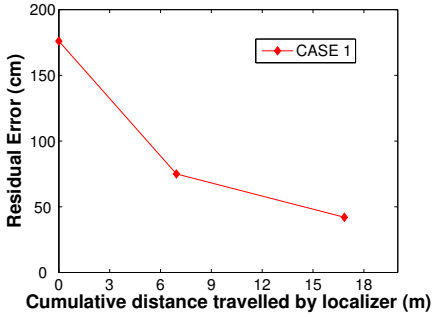
(c) 2 cases for clutter-1 setting.



(d) Performance in clutter-1 setting.

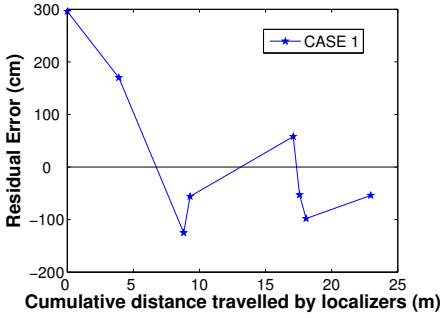


(e) Single case for clutter-2 setting.



(f) Performance in clutter-2 setting.

(g) Single case for clutter-3 setting.



(h) Performance in clutter-3 setting.

Figure 6.7: Performance evaluation of APDV under no-clutter, clutter-1, clutter-2 and clutter-3 settings.

an **ALIGN** message, it first measures its distances with its two neighbours to use as a reference for the subsequent alignment process. Thus, it accepts the neighbour distances irrespective of them being either larger or smaller than its current neighbour distances. Thus, we can get the scenario where the cumulative distances for the entire localizer chain can also increase in successive iterations. This does not happen when the localizer has the sensor robot and the anchor as its two neighbours. Nevertheless, the localizers settle with a multi-hop error of only 50 cms.

6.3 Discussion

In this chapter, we described a real test-bed implementation of the APDV algorithm with MIT Cricket motes and the Roomba robot. We discussed the hardware and software aspects of the implementation and evaluated the working of the APDV algorithm in a number of clutter topologies. Since the operating range of the Cricket motes is restricted to 400-500cm when the ultrasound signal bounces off clutter, we restrict our evaluation to small areas. We see that the APDV algorithm is able to provide improvements of up to 95% on the initial multi-hop distance errors. However, we do find that underestimate errors in the measured distances affect the completion time and the accuracy of the APDV algorithm. A key assumption of the APDV algorithm is that it has to deal with only two types of errors: NLOS distance errors and multi-hop overestimate errors. Since both of these errors are overestimating the actual distance, the APDV algorithm works by seeking a minimal sum of cumulative distances. However, our experiments in a real testbed show that negative distance errors also affect the working of APDV.

In the next chapter, we will look more closely at the relationship between the clutter topology and distance errors, both single-hop NLOS bias as well as multi-hop distance error.

Chapter 7

Influence of Clutter Topology on Distance Error

In the previous chapters, we looked at localizer placement algorithms for cluttered environments. We found that clutter topology plays an important role in determining the accuracy of these algorithms. Clearly, the localizer placement algorithms cannot provide high accuracy in all types of clutter topologies. In this chapter, we try to understand the influence of clutter topology on distance error itself. We consider two forms of distance errors: single-hop NLOS distance error obtained when a node estimates distances directly to anchors, and multi-hop error obtained from the distance overestimation in the cumulative multi-hop distances between a node and the anchors. First, we define a number of characteristic features for clutter topologies in Section 7.2. Next, in Section 7.3, we look at the effect of clutter topologies on NLOS bias via the perspective of these characteristic features. We show that, while it is challenging to obtain a direct relationship between any of these features and NLOS bias, using machine learning tools we can predict NLOS error when a combination of these characteristic features are taken into account. Finally, we apply the same technique in the determination of multi-hop error from the features of the clutter topologies in Section 7.4.

7.1 Motivation

The motivation for a characterization of distance error based on clutter topology is based on the observation that empirically we obtain distinct distance error characterizations for different clutter topologies. For example, consider

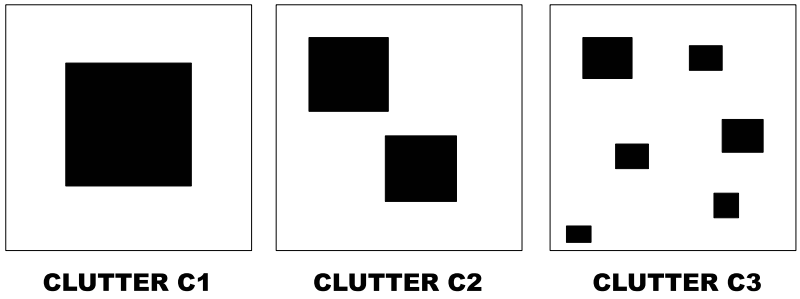
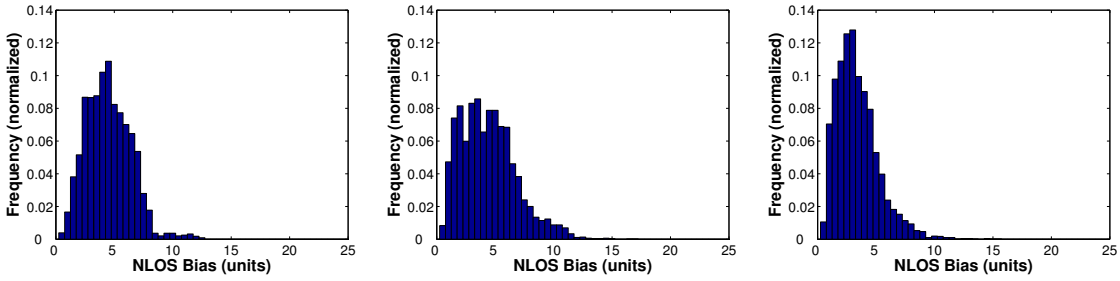


Figure 7.1: Sample clutter topologies C1, C2 and C3 or clutter classes FL (few large-size clutter), MS (many small-size clutter) and FS (few small-size clutter) respectively. These clutter classes were introduced in Chapter 3.



(a) $\mathcal{P}_{C1}$ ($f_{C1} = 0.4321$) (b) $\mathcal{P}_{C2}$ ($f_{C2} = 0.4441$) (c) $\mathcal{P}_{C3}$ ($f_{C3} = 0.5098$)

Figure 7.2: NLOS incidence probabilities and bias distributions for clutter topologies C1, C2 and C3. f and $\mathcal{P}$ obtained with 10000 samples.

the three clutter topologies C1, C2 and C3 in Fig. 7.1. Here, C1 is representative of the FL clutter class, C2 is representative of the MS clutter class and C3 is representative of the FS clutter class. We will characterize the NLOS distance error, obtained for a single-hop distance measurement taken in the midst of obstacles between the two ranging nodes, using two components:

1. NLOS Incidence Probability (f). This is the probability of a distance measurement being NLOS, as opposed to LOS, for an arbitrary pair of points in the clutter topology.

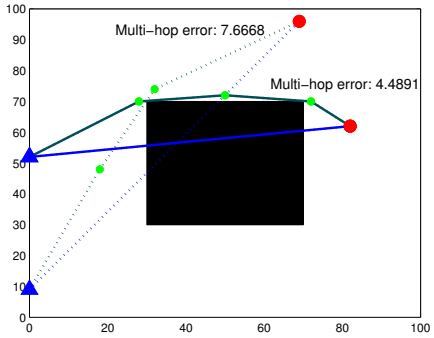
2. NLOS Bias Distribution ($\mathcal{P}$). This denotes the distribution of NLOS biases in the clutter topology. We use this as the error distribution for the NLOS distance between an arbitrary pair of points in the clutter topology.

Given a set of sample distances, some being LOS and others being NLOS, for arbitrary pairs of points in the clutter topology, the NLOS incidence probability

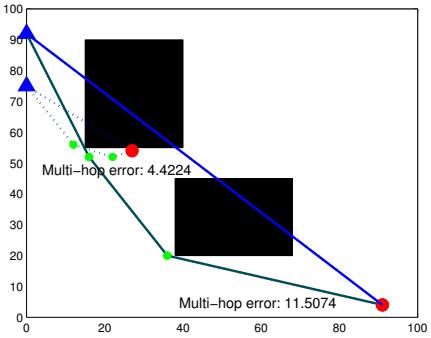
f can be obtained as the fraction of NLOS distances to the total number of distances. We can calculate the NLOS bias distribution $\mathcal{P}$ using the non-parametric histogram model method. The NLOS bias is calculated from the reflected NLOS distance obtained from the ray-tracer introduced in Chapter 3. Fig. 7.2 shows the variation of f and $\mathcal{P}$ for various clutter topologies. Fig. 7.2 shows that each of the three clutter topologies gives distinct values for f and $\mathcal{P}$. We see that f is determined by the spread of obstacles, even if they are smaller, in the environment. Notice, for example, that f is smaller in topology C1 than in topology C3, even though C1 has a clutter object much larger than all the clutter objects in topology C3. However, in the case of NLOS bias distributions, the size of the clutter objects does matter. As shown in Fig. 7.2, $\mathcal{P}_{C1}$ and $\mathcal{P}_{C2}$ have biases of larger magnitudes than in case of $\mathcal{P}_{C3}$. Given the complex interplay between the positions of the two ranging nodes and the (arbitrary) underlying clutter topology, it is not feasible to obtain analytical formulations for f and $\mathcal{P}$. One of the questions, then, we would like to address in this chapter is whether we can accurately predict f and $\mathcal{P}$ based on the characteristics of the clutter topology.

Next, we look at the impact of clutter topology on multi-hop distance error when shortest path distance-based multi-hop localization algorithms, such as DV-Distance [97] and MDS-MAP [118], are employed. The motivation for using multi-hop localization in cluttered NLOS-prone environment is outlined in Chapter 3. In the same chapter, the improvements in localization accuracy delivered by DV-Distance and MDS-MAP are shown to depend on the localizer density. In Chapter 4, we introduced the OPDV algorithm that seeks to find optimal localizer placements in the given clutter topology, thus significantly reducing the number of required localizers when compared to random localizer placements. OPDV's multi-hop error represents a lower bound on the distance error between two nodes in the midst of clutter when multi-hop distances are used. In this chapter, we are interested in looking at the influence of clutter topology on the residual multi-hop error of these OPDV-mandated localizer chains between the ranging nodes (anchor and sensor robot, for instance). Again, since it is hard to obtain an analytical formulation tying the OPDV error with the clutter topology, we will explore other empirical methods to predict (aggregate) OPDV error for a given clutter topology.

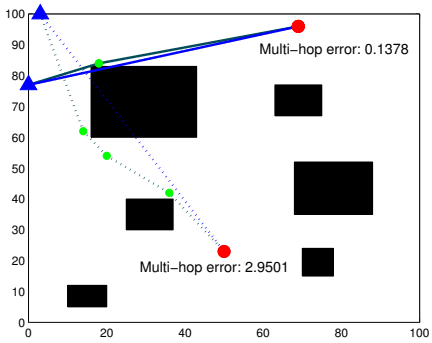
Fig. 7.3 shows OPDV multi-hop localizers for random positions of anchors and sensor robots, for the three clutter topologies C1, C2 and C3. Here, we fix



(a) Clutter Topology C1



(b) Clutter Topology C2



(c) Clutter Topology C3

Figure 7.3: Sample anchor-sensor-robot pairs (and corresponding OPDV residual errors) for clutter topologies C1, C2 and C3.

Clutter Topology	OPDV multi-hop error			Number of localizers
	mean	median	max	mean
C1	6.0073	4.9979	25.7636	1.5100
C2	4.1762	3.1223	18.9953	1.7660
C3	2.4250	1.6647	14.5509	1.9800

Table 7.1: OPDV distance error and average number of intermediate localizers required for clutter topologies C1, C2 and C3. 1000 samples of random anchor and sensor robot positions were used. The position of the anchor is fixed around the boundaries of the enclosure.

the anchor to random positions on the boundary of the enclosure. We can see that larger sizes of clutter objects lead to larger OPDV distance errors, as in the case of clutter topologies C1 and C2, when compared to topology C3. We can find aggregate values of the OPDV distance error and the number of localizers in Table 7.1. In the second half of this chapter, we look at the influence of various characteristic features of clutter topology in determining OPDV error

and the average number of localizers needed. This information can in turn be used to evaluate the feasibility of using localizer-based multi-hop localization in a given clutter topology.

7.2 Characteristic Features of Clutter Topology

In this section we will enumerate various characteristic features related to a clutter topology. These features are various forms of representing the spacing between the clutter objects. Some of the features require the complete knowledge of the clutter topology, such as the occupancy grid representations, while others can be obtain through practical means, such as the clutter spacing distributions.

7.2.1 Clutter Area Fraction

The ratio of the clutter area to the total area, ca, is an important indicator of the level of clutter in the environment:

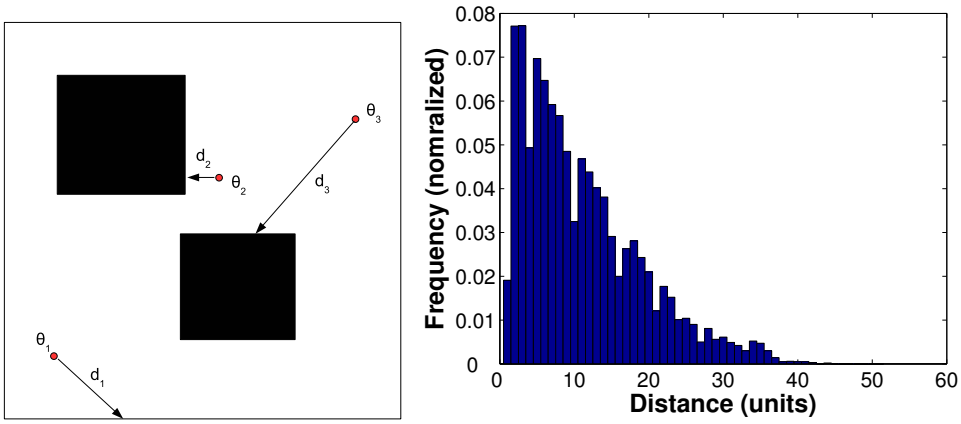
$$ca = \frac{\text{Total Area of Clutter}}{\text{Total Area of Enclosed Environment}}$$

In the case of semi-open areas, a convex hull can be constructed across the openings in the environments in order to calculate the total area of the environment. However, it cannot independently help estimate NLOS incidence or bias error as NLOS distance measurements are dependent both on the spacing in the environment between the clutter and the enclosure boundaries, and the total amount of clutter in the environment.

7.2.2 Clutter Spacing Distribution

The space in midst of the obstacles and bounding enclosure plays a vital role in determining the NLOS distance biases. We define two types of clutter spacing distributions:

- *Linear Clutter Spacing Distribution* $\mathcal{CD}_l$. This is the distribution obtained by measuring the space from a random position in the clutter topology (outside any obstacle) in a random direction until it strikes an obstacle



(a) Samples for $\mathcal{CD}_l$ (b) $\mathcal{CD}_l$ obtained with 1000 samples

Figure 7.4: Derivation of linear clutter spacing distribution, $\mathcal{CD}_l$. Here, three random points are chosen along with three random direction θ_1, θ_2 and θ_3, that give the corresponding spacing distances d_1, d_2 and d_3. In practice, one can build these distributions with 500 or 1000 samples points.

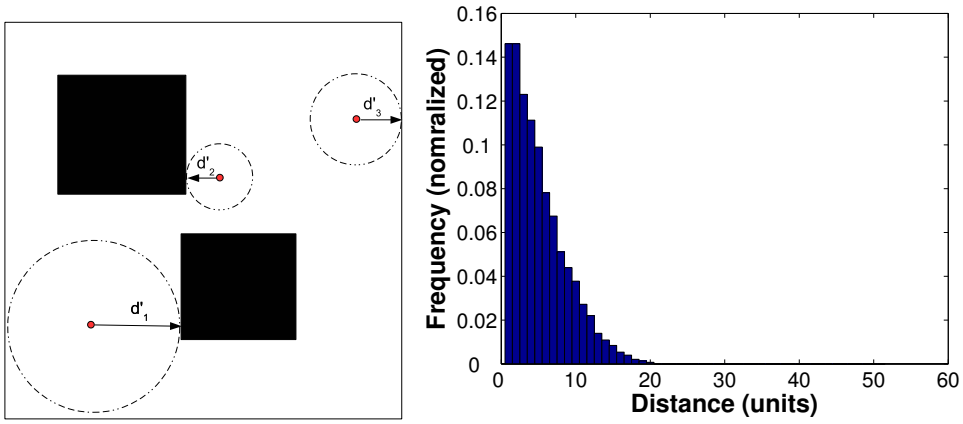
or the enclosure walls in its path, as seen in Fig. 7.4(a). We obtained the $\mathcal{CD}_l$ for clutter topology C2 as shown in Fig. 7.4(b).

- *Radial Clutter Spacing Distribution* $\mathcal{CD}_r$. This is the distribution obtained by measuring the maximum circle that can be drawn centered at a (random) point such that it does not intersect an obstacle or the enclosure walls, as shown in Fig. 7.5(a). We obtained the $\mathcal{CD}_r$ for clutter topology C2 as shown in Fig. 7.5(b).

A desirable feature of $\mathcal{CD}_l$ and $\mathcal{CD}_r$ is that they can be obtained through practical means without the need for explicit knowledge of the clutter topology. For example, a laser rangefinder can be used to obtain $\mathcal{CD}_l$ while a laser range scanner can be used to obtain $\mathcal{CD}_r$.

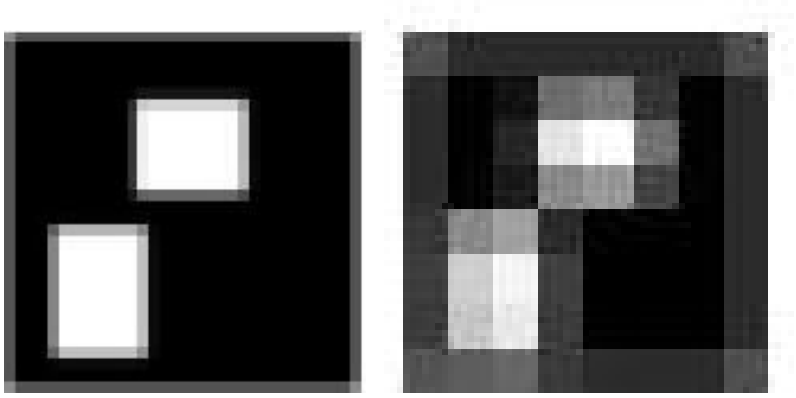
7.2.3 Occupancy Grid

The occupancy grid of the clutter topology is a literal representation of the actual map of the clutter topology scaled by a factor s. In other words, if occ_1 represents the matrix representation of the clutter topology, with 1s denoting the enclosure boundaries and clutter and 0s denoting free space, occ_S is the corresponding matrix with dimensions scaled by a factor of S, where $S = \frac{norm([X_L^O Y_L^O])}{norm([X_L Y_L])}$ and $(X_L^O,\ Y_L^O)$ are the original dimensions of the clutter map image and $(X_L,\ Y_L)$ are the dimensions of the scaled image. For example,



(a) Samples for $\mathcal{CD}_r$ (b) $\mathcal{CD}_r$ obtained with 1000 samples

Figure 7.5: Derivation of radial clutter spacing distribution, $\mathcal{CD}_r$. Here, three random points are chosen along with three random direction θ_1, θ_2 and θ_3, that give the corresponding spacing distances d'_1, d'_2 and d'_3 that represent the radii of the smallest circles that can be fitted before touching, in the *first* instance, a clutter piece or the enclosure wall. In practice, one can build these distributions with 500 or 1000 samples points.



(a) $occ_{(3.125)}$ (b) $occ_{(12.5)}$

Figure 7.6: Occupancy grid representation of clutter topology C2

if occ_1 is a 100x100 matrix, then $occ_{12.5}$ will be represented by a 8x8 matrix and $occ_{3.125}$ by a 32x32 matrix. We select a bi-linear interpolation scheme for matrix compression. Fig. 7.6 shows the graphical representations of $occ_{3.125}$ and $occ_{12.5}$ for the clutter topology C2. The idea behind this representation is to capture the low-level structure of the clutter topologies while abstracting the higher level representation.

(a) $ft_{3.125}$ of C1 (b) $ft_{3.125}$ of C2 (c) $ft_{3.125}$ of C3

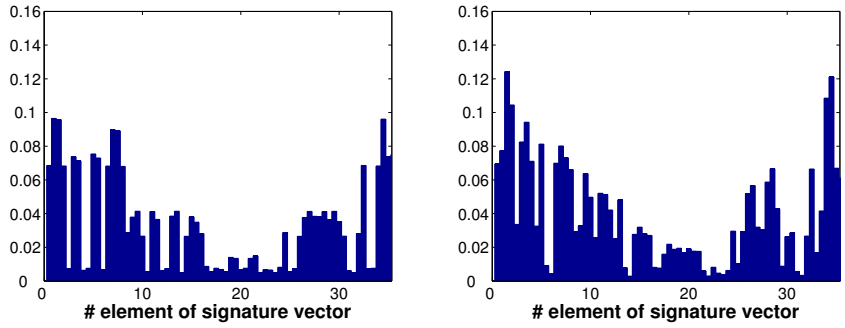
Figure 7.7: Fourier transform characterization of clutter topologies C1, C2, C3.

7.2.4 Fourier Transformation

The Fourier transform is a popular representation in the image processing research community [32, 126, 67]. It highlights the dominant spatial frequencies as well as the dominant orientations of the structures contained in the image. The Fourier transformation of an image provides its representation in the frequency domain. A two-dimensional Fourier transformation (ft_S) of the occupancy grid occ_S, where S is the scale factor of the occupancy grid, is used to characterize the spacing and structure of the clutter. We use the magnitude of the 2D Fourier transform as the feature in our analysis. Fig. 7.7 shows the pictorial representation of $ft_{3.125}$ for clutter topologies C1, C2 and C3. Here the centres of each image represent the intensity of lower frequency sinusoids while the higher frequency sinusoids are represented towards the boundaries. Fig. 7.7(c) shows the increased intensity of high frequency sinusoidal waves towards the boundaries representing the numerous, dispersed clutter in the clutter topology C3, when compared to topologies C1 and C2.

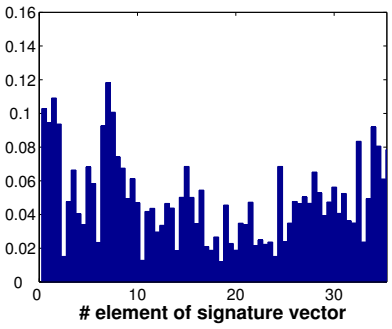
7.2.5 GIST Characterization

The GIST [100] ($\mathcal{G}$) characterization of an image, widely used in the area of image classification, defines a set of 'perceptual' dimensions, such as naturalness, openness, roughness, expansion, ruggedness, which represent the dominant spatial structure of the image. A set of 384 Gabor filters, each with a distinct set of parameters like frequency, orientation, etc., are individually convolved with the image to give a 384 length signature vector. The technique can be used to capture low level details of the image, for example, in our case, the spacing and shapes of the clutter topology, while abstracting away the high-dimensional detail. Fig. 7.8 shows the variation in the GIST characterization for the clutter



(a) $\mathcal{G}_{C1}$

(b) $\mathcal{G}_{C2}$



(c) $\mathcal{G}_{C3}$

Figure 7.8: GIST characterization of clutter topologies C1, C2 and C3.

topologies C1, C2 and C3.

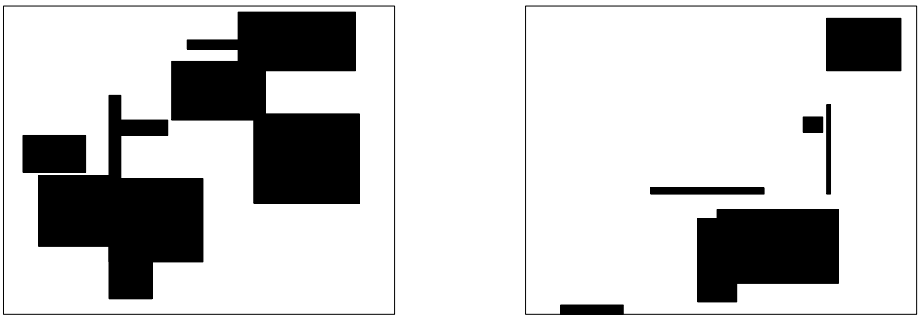
7.3 Effect of Clutter Topology on NLOS Bias

NLOS bias is hard to model accurately as it is shown to depend on the (arbitrary) underlying topology/arrangement of the obstacles in the environment [41, 52, 133, 5, 6, 58]. For this reason, in current literature, it is typically assumed to be uniform [57, 128, 91, 23], Gaussian [13] or exponential [23, 42, 46] for the sake of convenience. Jourdan et al. [58] mention the use of empirically derived NLOS bias distributions in calculating the Position Error Bound (PEB) for localization in a cluttered environment. We look at the estimation of two facets of NLOS error when the clutter topology is given: NLOS incidence probability and NLOS bias distributions. What we really want to investigate is how the various features of clutter topology, outlined in the previous section, play a role in estimating NLOS error. It is desired to be able to predict NLOS error with those characteristic features that can be practically ascertained, such as the clutter spacing distributions. For estimating NLOS error and probability, we also use the communication range (C) of the ranging nodes as an additional input. This is vital as NLOS distances are possible only when the ranging signals have enough energy to reflect off a number of surfaces and still reach the other node.

7.3.1 Experimental Setup

In this section we will discuss the setup and fidelity of the experimental data. We use a ray-tracer tool, first described in Section 3.2.1, to generate NLOS distances between two ranging nodes in midst of clutter. The ray tracer assumes that the ranging signal follows the Fresnel laws of reflection when it strikes obstacles or enclosure boundaries and do not pass through clutter itself.

Clutter topology generation. We build complex clutter topologies by sequentially overlaying simple boxes over each other. We use a 2D rectangular enclosure area with dimensions L_x and L_y. We then generate ncl boxes with random start points (x_i, y_i) and dimensions lx and ly, where $lx = \mathcal{U}(x_i, x_i + \frac{L_x}{dimfac})$ and $ly = \mathcal{U}(y_i, y_i + \frac{L_y}{dimfac})$. ncl is a randomly chosen natural number between [1,10] and $dimfac$ is set to 3 in order to allow boxes to have dimensions at most a third of the enclosure dimensions. Each box is validated to fit inside the rectangular enclosure. Fig. 7.9 shows a couple of random clutter samples used in our analysis.



(a) Clutter sample 1 (b) Clutter sample 2

Figure 7.9: Arbitrary clutter samples generated by overlaying simple rectangular boxes in the enclosure area.

NLOS characterization data. Given a clutter topology, we generate the NLOS incidence probability f and the NLOS bias distribution $\mathcal{P}$ by taking $\mathcal{N}$ samples of uniform random generated points in the clutter topology. f is calculated as the ratio of the NLOS distances observed in the $\mathcal{N}$ samples taken. The histogram $\mathcal{P}$ is built using the bias of the NLOS distances in the $\mathcal{N}$ samples. One can use techniques like LOESS [25, 60] in order to obtain a smoothed probability distribution from $\mathcal{P}$.

In order to establish the consistency of the NLOS characterization data, we vary the number of samples $\mathcal{N}$ and measure the consistency of the f and $\mathcal{P}$ values. In other words, we want to determine, if we take $\mathcal{N}$ random samples, multiple times, how close will the values of f (or of $\mathcal{P}$) be to each other. We use the three clutter topologies shown in Fig. 7.1 for our analysis. For each clutter topology, we repeatedly (10 times) take $\mathcal{N}$ samples and calculate f and $\mathcal{P}$. For each clutter topology, we compute the dissimilarity between the 10 instances of f (and similarly between the 10 instances of $\mathcal{P}$), as $dist_f$.

For f, we use average pair-wise distances between the 10 instances of f, for 3 clutter topologies each, to measure consistency. $dist_f$ represents the average of the pair-wise distances over three clutter topologies. In the case of the bias distribution $\mathcal{P}$, we use the Kullback Leibler Divergence (KLD) distance [65] to measure similarities (or dissimilarities) between instances of $\mathcal{P}$. The KLD distance is an inherently non-symmetric metric and for two distributions P and

Sample Size ($\mathcal{N}$)	$dist_f$	$\sigma(dist_f$
100	0.0565	0.0536
500	0.0244	0.0171
1000	0.0199	0.0141
5000	0.0083	0.0055
10000	0.0070	0.0056
50000	0.0023	0.0017

Table 7.2: Measure of consistency level of NLOS incidence probability (f) in generated data.

Q, is given by

$$D_{KL}(P\|Q) = \sum_i P(i) ln \frac{P(i)}{Q(i)} \tag{7.1}$$

$$\tag{7.2}$$

where both P and Q sum up to 1, $Q(i) > 0$ every time $P(i) > 0$ and ln represents the natural logarithm. We use the *symmetric* version of the KLD distance given by Eqn. 7.3 to evaluate the dissimilarity between two histograms/distributions.

$$D_{KS}(P,Q) = D_{KL}(P\|Q) + D_{KL}(Q\|P) \tag{7.3}$$

The consistency of $\mathcal{P}$ is measured in terms of the dissimilarity between between the 10 instances of $\mathcal{P}$ for a given value of $\mathcal{N}$. We measure this dissimilarity as the average of the symmetric Kullback Leibler Divergence D_{KS} between the first instance and each of the remaining nine instances of $\mathcal{P}$. $dist_{\mathcal{P}}$ represents the average over three clutter topologies.

Tables 7.2 and 7.3 represent the consistency of the NLOS characterization data we generate for our analysis. We choose $\mathcal{N}$ as 50000 for our data generation, which yields a consistency error of approximately 0.002 for both f and $\mathcal{P}$.

7.3.2 Estimation of NLOS Incidence Probability

In this section, we will look at the estimation of the NLOS probability of an arbitrary clutter topology when we are given only the features that characterize the clutter topology. We use a Support Vector Regressor (SVR) which uses a given training set and is able to predict the NLOS incidence probability f.

Sample Size ($\mathcal{N}$)	$dist_{\mathcal{P}}$	$\sigma(dist_{\mathcal{P}}$
100	1.7861	0.6555
500	0.3253	0.1061
1000	0.1364	0.0517
5000	0.0287	0.0165
10000	0.0072	0.0024
50000	0.0016	0.0004

Table 7.3: Measure of consistency level of NLOS bias distribution ($\mathcal{P}$) in generated data.

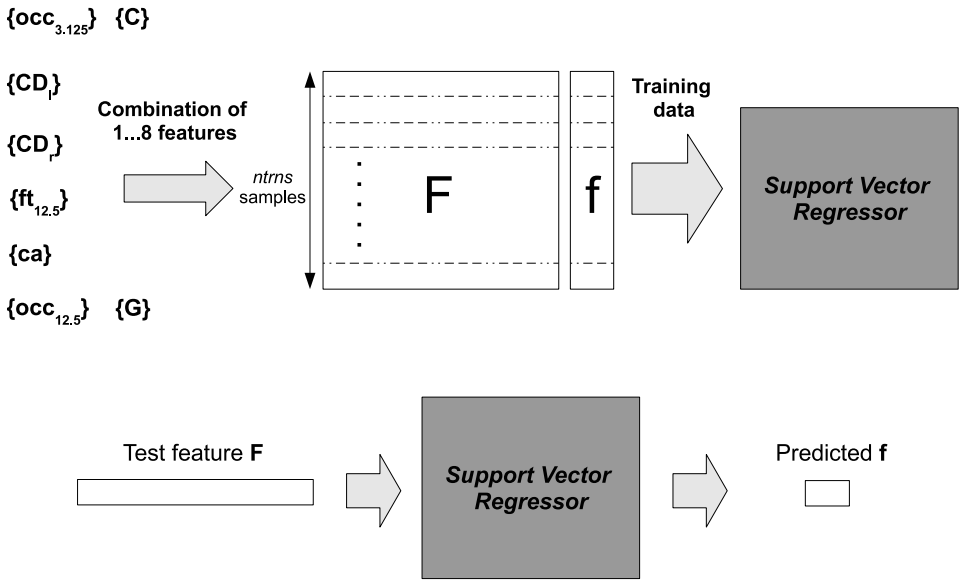


Figure 7.10: Estimation of NLOS incidence probability f using Support Vector Regressor (SVR)

7.3.2.1 Support Vector Regressor

An ϵ-SVR [27, 122] solves the following optimization problem

$$\text{minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\zeta_i + \zeta_i^* \tag{7.4}$$

$$\text{subject to } \begin{cases} y_i - < w, x_i > -b & \leq \epsilon + \zeta_i \\ < w, x_i > +b - y_i & \leq \epsilon + \zeta_i^* \\ \zeta_i, \zeta_i^* & \geq 0 \end{cases}$$

where ζ_i and ζ_i^* are slack variables and ϵ is the precision. The unknown parameters, w and b, are determined based on the training set $\{x_k, y_k\}_{k=1}^{N}$, where

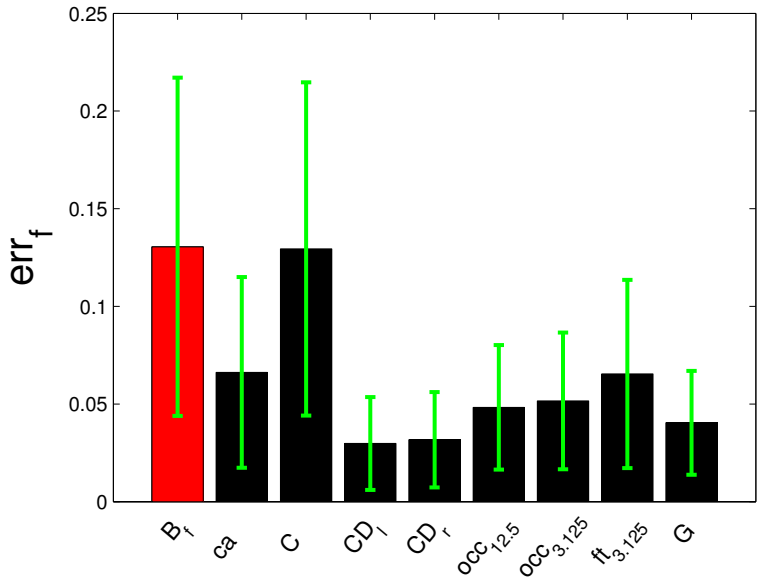


Figure 7.11: Estimation error of f, err_f, for individual features. Here $\mathcal{B}_f$ represents the baseline solution, *ca* represent clutter area, $\mathcal{C}$ represents communication range, CD_l represents the linear clutter spacing distribution, CD_l represents the linear clutter spacing distribution, $occ_{3.125}$ and $occ_{12.5}$ represent occupancy grids with granularity 3.125 and 12.5 respectively, $fft_{3.125}$ represents the Fourier transform with granularity 3.125, and finally, $\mathcal{G}$ represents the GIST representation.

$x_k \in \mathbb{R}^n$ is the input and $y_k \in \mathbb{R}$ the respective output. In our case, the input x_i represents the feature set $\mathcal{F}_f^n$ of the i^{th} clutter topology sample, where n denotes the number of features we are considering at a time. The output is the NLOS incidence probability f_i. We found that non-linear ϵ-SVR together with the Radial Basis (RBF) kernel gives the best results during our analysis, when compared to linear, polynomial and sigmoid kernels. Fig. 7.10 gives a graphical overview of the process of estimation of f using the SVR.

7.3.2.2 Results

We use a training set of $ntrn$ (=10000) and test set of n_{ts} (=5000) distinct randomly generated clutter topologies. Each sample is split into features (input) and labels (output). Three values have been used for communication range $\mathcal{C}$: 0.7D, 1.4D and 3.5D. While the labels are set to f, the features, drawn from those defined in Section 7.2, are varied in number and composition. The SVR is first trained with the training set and we then supply it with a feature set

Estimation Technique using SVR	err_f (mean)	err_f (std)
$\mathcal{B}_f$	0.1305	0.0866
$\mathcal{F}_f^1 = \{CD_l\}$	0.0298	0.0238
$\mathcal{F}_f^2 = \{CD_l, CD_r\}$	0.0272	0.0230
$\mathcal{F}_f^3 = \{ca, CD_l, CD_r\}$	0.0294	0.0236
$\mathcal{F}_f^4 = \{ca, \mathcal{C}, CD_l, CD_r\}$	0.0272	0.0230

Table 7.4: Mean and standard deviation of the absolute error in estimating NLOS incidence probability (f) using SVR. Here, the set $\mathcal{F}_f^n$ denotes the set of n features which achieve the minimum mean err_f.

$\mathcal{F}_f$ from the test set and ask it to predict the value of f.

First, we analyze the estimation accuracy of f when the SVR is used with only one feature at a time. Fig. 7.11 shows the performances of the features are taken individually to predict f. The clutter spacing distributions, $\mathcal{CD}_l$ and $\mathcal{CD}_r$, perform the best, followed by the GIST representation of the clutter topology, $\mathcal{G}$. The communication range, $\mathcal{C}$, performs the worst since it alone plays a minor role in determining whether a range measurement will be NLOS or not.

Next, we use combinations of various features for the estimation of f. The results for feature sets that give the best results (lowest err_f) are shown in Table 7.4. Here $\mathcal{F}_f^n$ represents the feature set that produces the least error among any other combination of n features. The estimation accuracy is denoted by the mean (absolute) estimation error, denoted by err_f and its standard deviation. We see that the clutter spacing distributions play a significant role in determining the value of f and are aided by other features like communication range ($\mathcal{C}$) and ca. We note that combinations of features can give non-intuitive results as well. For example, in Fig. 7.11, the GIST representation $\mathcal{G}$ performs well on its own, better than ca and $\mathcal{C}$. However when taken in combination with other features, we found that the feature set $\mathcal{F}_f$=$\{CD_l, CD_r, \mathcal{G}\}$ performs worse than $\mathcal{F}_f$=$\{ca, CD_l, CD_r\}$ and $\mathcal{F}_f$=$\{ca, \mathcal{C}, CD_l, CD_r\}$. For comparison purposes we use a baseline $\mathcal{B}_f$, which is simply the average of $\{f_i\}_{i=1}^{ntrn}$, where f_i is the NLOS probability of the i^{th} training sample and $ntrn$ is the total number of training samples. We see that there is a substantial improvement of the prediction accuracy with the use of the SVR over the (naïve) average of all the training data results. We can get an average estimation error of only 0.0272 using only two features, $\mathcal{F}_f = \{CD_l, CD_r\}$, as shown in Table 7.4.

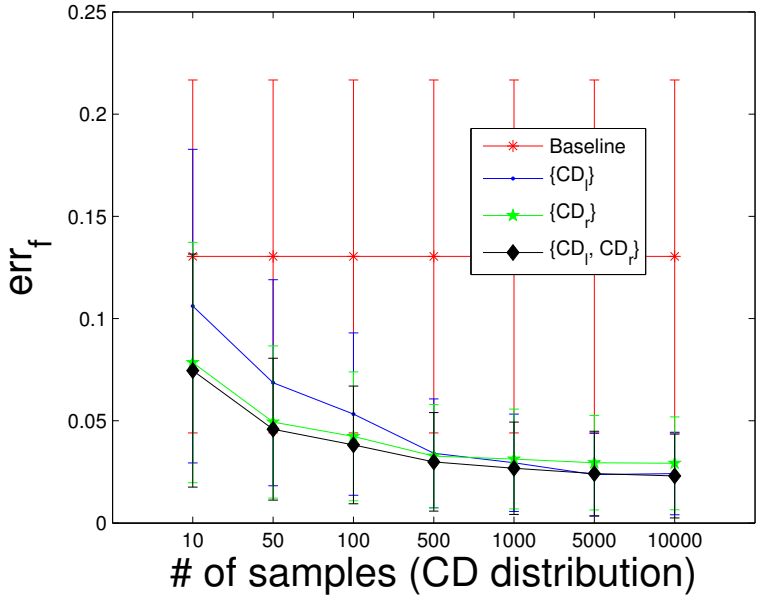


Figure 7.12: Effect of the sample size for the clutter spacing distributions, CD_l and CD_r, on the accuracy of the NLOS probability (f) estimation using SVM regression.

Finally, we investigate the effect of consistency of the spacing distributions on err_f, since we find that $\mathcal{CD}_l$ and $\mathcal{CD}_r$ are the most effective features in the accurate prediction of f. Fig. 7.12 shows the effect on the number of samples used to build the CD_l and CD_r distributions on the actual estimation error of the NLOS probability (f). Having a sample size as low as 100 can yield an estimation error of 0.05. This shows the viability of these clutter features in a practical scenario where f has to be estimated in arbitrary cluttered environments.

7.3.3 Estimation of NLOS Bias Distribution

We look at the estimation of the NLOS bias distribution from clutter topology features. We, first, briefly discuss the basics of the k-nearest neighbours (k-NN) approach we use for estimating the bias distribution. We then discuss the results and the dependence of the accuracy of the bias distribution estimation on various parameters of the k-NN estimation technique.

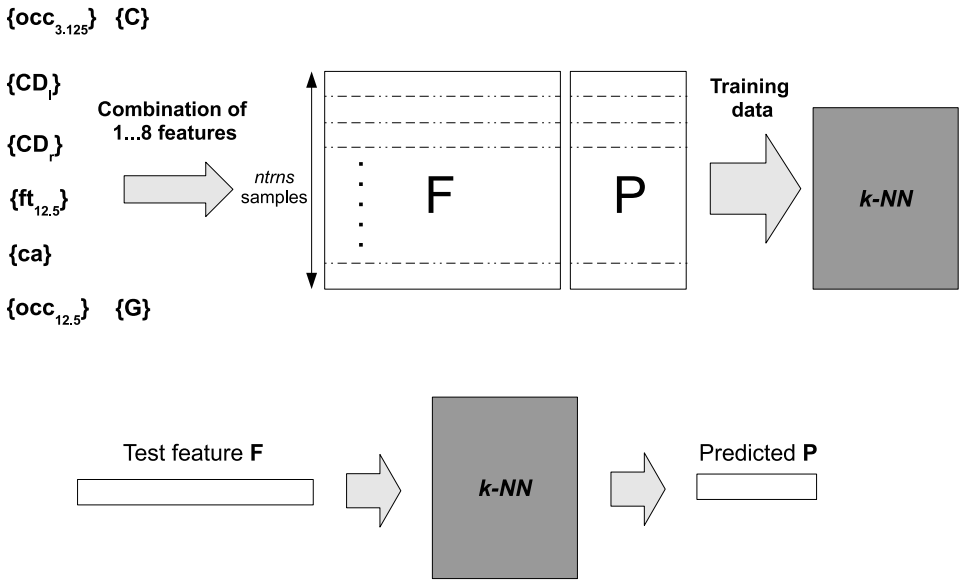


Figure 7.13: Estimation of NLOS bias distribution/histogram $\mathcal{P}$ using k-NN algorithm.

7.3.3.1 k-Nearest-Neighbour Estimation

The k-Nearest-Neighbour estimation algorithm is a method for classifying objects based on the closest *training* examples in the feature space. It is an instance-based learning technique where an object is classified by the majority vote of its neighbours, with the assumption that the object is probably the class most common amongst its k neighbours. In our case, instead of classification, we estimate the bias distribution as a function of the individual bias distributions of the k-nearest neighbours.

Suppose our feature set, $\mathcal{F}_{\mathcal{P}}$, is an arbitrary set of the features discussed in Section 7.2. Treating the feature set as input, we have $\{x_i\}_{i=1}^{ntrn}$ where $ntrn$ is the number of training samples and $x_i \in \mathcal{R}^n$. The $ntrn$ samples of the training set can be viewed to represent a set of points in n-dimensions. Given a new feature set input $\mathbf{x} \in \mathcal{R}^n$, we try to find the k-nearest neighbours of the point $\mathbf{x}$ in the n-dimensional space. The bias distribution $\mathcal{P}(\mathbf{x})$ is obtained as a linear interpolation of the bias distributions [**?**] of its k-nearest neighbours using the Euclidean distances to neighbours as interpolation weights. Fig. 7.13 gives a graphical overview of the process of estimation of $\mathcal{P}$ using the k-NN algorithm.

Let $\mathcal{NN}_{\mathbf{x}}^k$ ($\exists p, 1 \leq p \leq ntrn$ and $\mathcal{NN}_{\mathbf{x}}^k = x_p$) denote the k^{th} nearest neighbours of $\mathbf{x}$, separated by the Euclidean distance $d_{\mathbf{x}}^k$. If $\mathcal{P}(\mathcal{NN}_{\mathbf{x}}^k)$ is the NLOS bias distribution of the k^{th} nearest neighbour of $\mathbf{x}$, the NLOS bias distribution

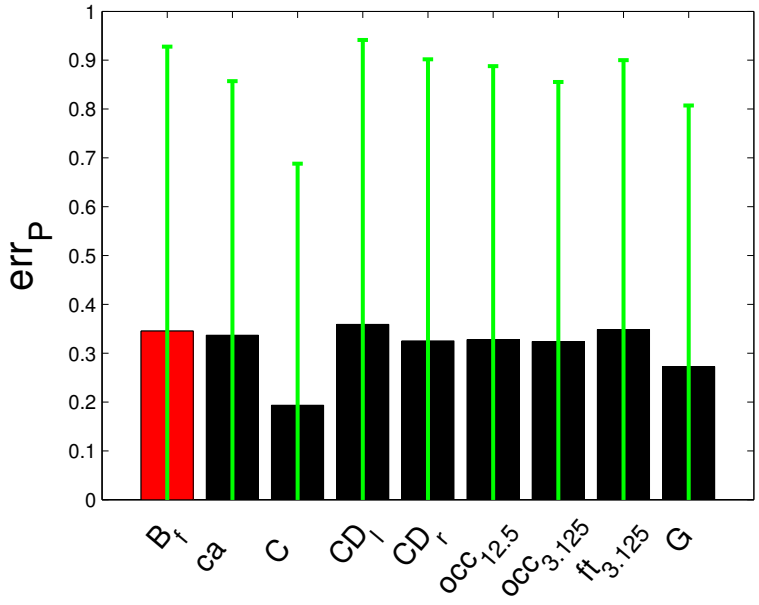


Figure 7.14: Estimation error of $\mathcal{P}$, $err_{\mathcal{P}}$, for individual features.

$\mathcal{P}(\mathbf{x}$ is estimated as

$$\mathcal{P}(\mathbf{x}) = \sum_{j=1}^{k} \gamma_j \mathcal{P}(\mathcal{NN}_{\mathbf{x}}^{j}) \tag{7.5}$$

$$\gamma_j = \frac{e^{-(d_{\mathbf{x}}^{i})^2}}{\sum_{i=1}^{k} e^{-(d_{\mathbf{x}}^{i})^2}} \tag{7.6}$$

where γ_j denotes the j^{th} weight in the weighted sum calculation. Since the interpolation weights are normalised, the resulting $\mathcal{P}(\mathbf{x}$ is guaranteed to be a probability distribution as well.

7.3.3.2 Results

We use a training set of $ntrn$ (= 10000) and test set of n_{ts} (= 5000) distinct clutter topologies. Here, the search space is built with features which can vary in composition. $\mathcal{F}_{\mathcal{P}}^{i}$ denotes the feature set with i features mentioned in Section 7.2. We exhaustively try the various features, individually as well as in combination with other features. The k-NN algorithm is initially trained with the training set. We set k=10 for our analysis.

We begin by analyzing the impact of individual features in estimating $\mathcal{P}$ using the k-NN algorithm. $err_{\mathcal{P}}$ is calculated as the symmetric KLD metric (D_{KS}) between the estimated bias distribution and the actual bias distribution

Estimation Technique using k-NN	$err_{\mathcal{P}}$ (mean)	$err_{\mathcal{P}}$ (std)
$\mathcal{B}_{\mathcal{P}}$	0.3452	0.5827
$\mathcal{F}_{\mathcal{P}}^{1} = \{\mathcal{C}\}$	0.1934	0.4947
$\mathcal{F}_{\mathcal{P}}^{2} = \{\mathcal{C}, CD_r\}$	0.1620	0.3812
$\mathcal{F}_{\mathcal{P}}^{3} = \{ca, \mathcal{C}, CD_r\}$	0.1551	0.3783
$\mathcal{F}_{\mathcal{P}}^{4} = \{ca, \mathcal{C}, CD_l, CD_r\}$	0.2426	0.4272

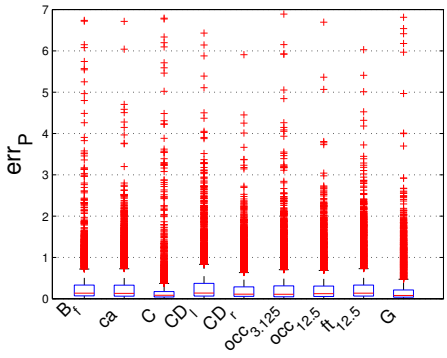
Table 7.5: Estimation of NLOS bias distribution ($\mathcal{P}$) using k-NN. Here, the set $\mathcal{F}_{\mathcal{P}}^{i}$ denotes the set of i features which achieve the minimum $err_{\mathcal{P}}$

of the test data, which measures the dissimilarity between the two distributions. We use the mean and standard deviation of $err_{\mathcal{P}}$ for the entire test set to measure the accuracy of NLOS bias distribution estimation. Fig. 7.14 shows the performance of individual features in predicting the $\mathcal{P}$ for the test set. We find that none of the individual features perform well on average in estimating $\mathcal{P}$. The reason for this is that the NLOS bias distribution is too complex for a single feature to be able to predict it accurately.

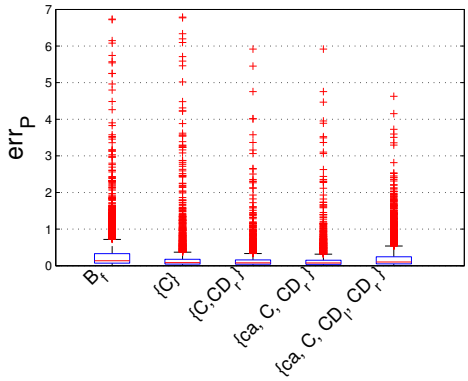
Next, we analyze the efficacy of various combinations of features in predicting $\mathcal{P}$ for the test set. We use exhaustive combinations of features, using two, three and four features together. We list the results for feature sets that give the best results (lowest $err_{\mathcal{P}}$) in Table 7.5. Here, $\mathcal{F}_{\mathcal{P}}^{n}$ represents the feature set that produces the least error when any combination of n features is used. $\mathcal{B}_{\mathcal{P}}$ represents a baseline solution with the given training data, where $\mathcal{P}$ is estimated by taking the average of all NLOS bias distributions in the training data.

Table 7.5 shows that communication range $\mathcal{C}$ plays a significant role in improving the estimation error $err_{\mathcal{P}}$. However we do note the unusually large variances in $err_{\mathcal{P}}$ when compared to the mean $err_{\mathcal{P}}$ values, which suggests that the k-NN algorithm is not able to find accurate estimates for a small number of test samples. We also note that increasing the number of features beyond three does not necessarily improve the estimation accuracy of $\mathcal{P}$. For instance, $err_{\mathcal{P}}$ is lower for $\mathcal{F}_{\mathcal{P}}^{3}$ than $\mathcal{F}_{\mathcal{P}}^{4}$ as shown in Table 7.5.

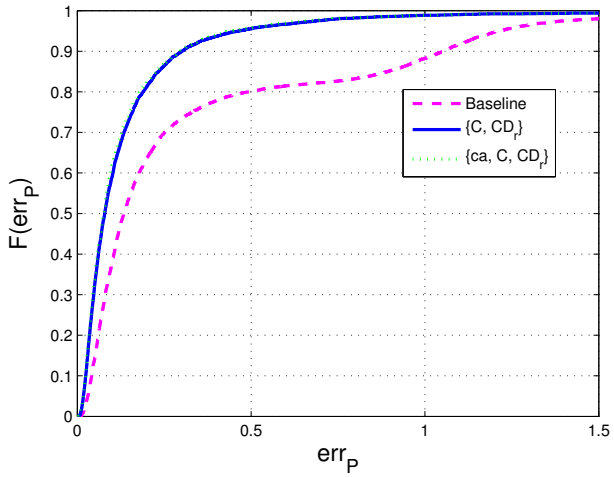
Though the k-NN algorithm is unable to predict $\mathcal{P}$ with good accuracy for every element of the test set, we nevertheless find that in a large number of cases the $err_{\mathcal{P}}$ is low. We can deduce that large outliers seen in the box-plots Fig. 7.15(a) and Fig. 7.15(b) are the reason for the high standard deviation seen in Fig. 7.14 and Table 7.5 respectively. Fig. 7.15(c) shows the empirical



(a) Distribution of $err_{\mathcal{P}}$ for test data when individual clutter features are used



(b) Distribution of $err_{\mathcal{P}}$ for test data when combination of clutter features are used

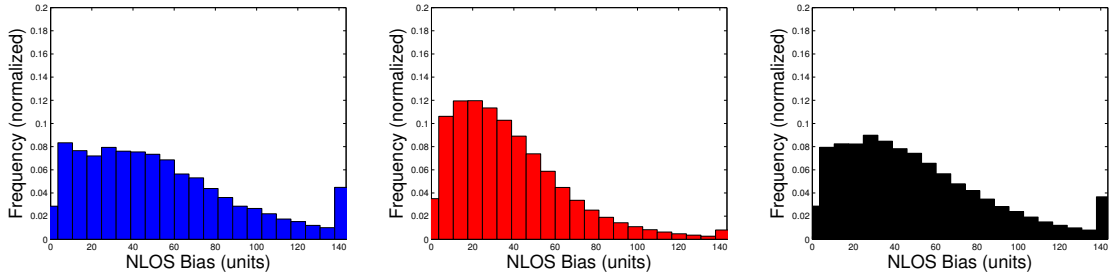


(c) Empirical distribution of $err_{\mathcal{P}}$

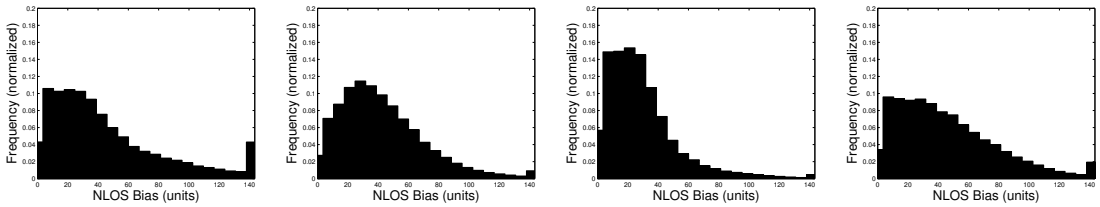
Figure 7.15: Distribution of estimation error $err_{\mathcal{P}}$

cumulative distribution of $err_{\mathcal{P}}$, for feature sets $\mathcal{F}^2_{\mathcal{P}}$ and $\mathcal{F}^3_{\mathcal{P}}$ where we see that in case of $\mathcal{F}^3_{\mathcal{P}}$, more than 80% of the test set yields an $err_{\mathcal{P}} < 0.2$. Furthermore, we see that there is a substantial improvement in the accuracy of predicting $\mathcal{P}$ when compared to using the naive average of all training set $\mathcal{P}$ ($\mathcal{B}_{\mathcal{P}}$).

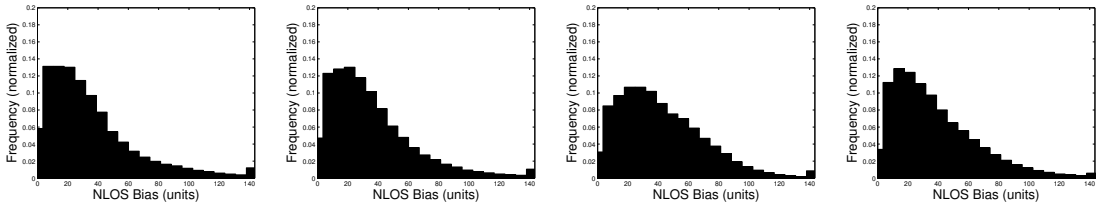
Next, we provide a visual representation of the estimated $\mathcal{P}$ for various features. Fig. 7.16 shows a comparison of the estimation of $\mathcal{P}$ by various features sets for a single sample clutter topology. Fig. 7.16(d) – 7.16(k) show the results for individual features, while Fig. 7.16(a) and 7.16(b) show actual bias distribution and the *baseline* bias distribution (obtained by averaging all $\mathcal{P}$ in the training set) respectively. Fig. 7.16(c) shows the estimation of $\mathcal{P}$ using the feature set $\mathcal{F}_{\mathcal{P}} = \{ca,\ \mathcal{C},\ CD_r\}$, which gives a low dissimilarity measure D_{KS} of 0.01.



(a) Actual $\mathcal{P}$ (b) Baseline, $D_{KS} = 0.2510$ (c) $\mathcal{F}_{\mathcal{P}}$=$\{ca, \mathcal{C}, CD_r\}$, D_{KS}=0.01



(d) $\mathcal{F}_{\mathcal{P}}$=$\{ca\}$ (e) $\mathcal{F}_{\mathcal{P}}$=$\{\mathcal{C}\}$ (f) $\mathcal{F}_{\mathcal{P}}$=$\{CD_l\}$ (g) $\mathcal{F}_{\mathcal{P}}$=$\{CD_r\}$



(h) $\mathcal{F}_{\mathcal{P}}$=$\{occ_{12.5}\}$ (i) $\mathcal{F}_{\mathcal{P}}$=$\{occ_{3.125}\}$ (j) $\mathcal{F}_{\mathcal{P}}$=$\{ft_{3.125}\}$ (k) $\mathcal{F}_{\mathcal{P}}$=$\{\mathcal{G}\}$

Figure 7.16: Estimation of NLOS Bias distribution with k-NN.

Finally, in order to analyze the effect of neighbour size k, we perform the k-NN estimation for varying values of k. Fig. 7.17 demonstrates the effect of varying the maximum nearest neighbour set size on $err_{\mathcal{P}}$. We conclude that a neighbour set of size k=10 is the optimal size for $\mathcal{F}_f^2$ and $\mathcal{F}_f^3$ which yields the lowest $err_{\mathcal{P}}$.

7.3.3.3 Lognormal Fitting for NLOS Bias Distribution

We observed that the vast majority of NLOS bias distributions generated for our clutter topology data-set had the appearance of a lognormal distribution. In order to validate this assumption we performed the Kolmogorov-Smirnov (KS) test on the data after taking its logarithm. The KS test is a nonparametric method for determining the equality of continuous, one-dimensional probability distributions. It can be used either as a one sample test, to check whether a given sample belongs to a reference probability distribution, or as a two-sample

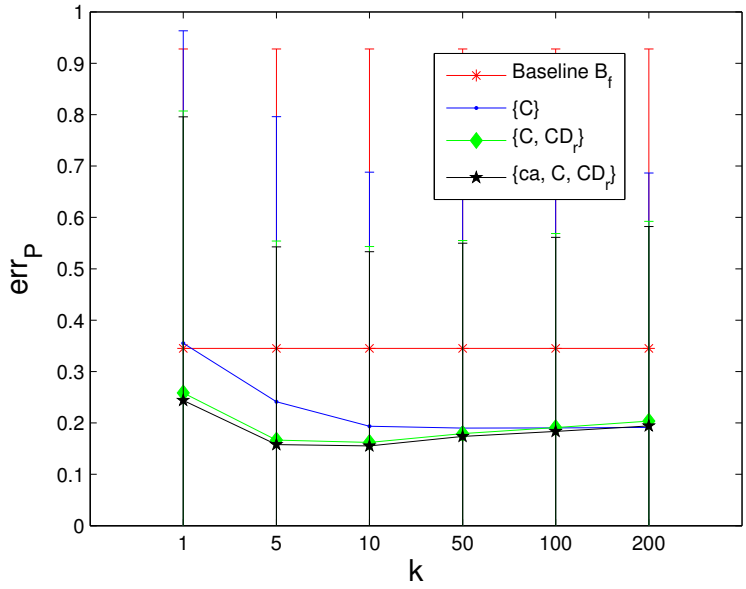


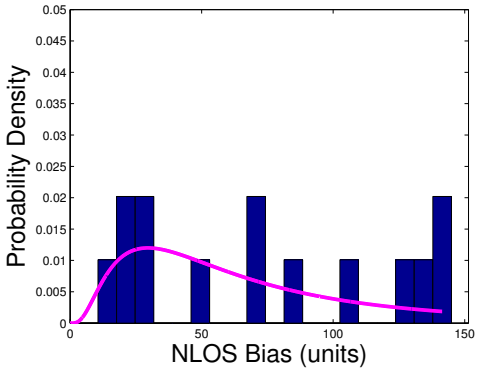
Figure 7.17: Effect of the maximum number of nearest neighbour k considered, on the accuracy of $\mathcal{P}$.

test where two samples are compared directly to check if they belong to the same distribution. The Lilliefors goodness-of-fit test is a modified form of the KS test for the normality of the given sample. The Lilliefors test is suitable for situations where a fully-specified null distribution is not given, while the one-sample KS test requires that the null distribution be completely specified.

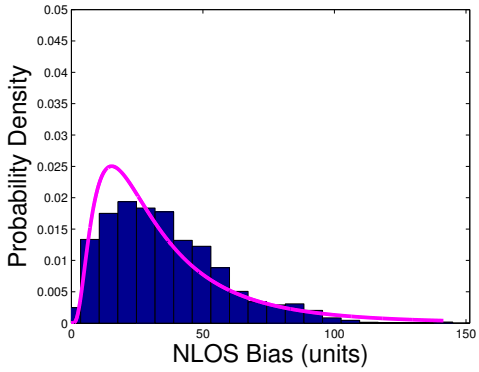
Fig. 7.18 shows the results of lognormal fitting for four distinct samples. The values of the NLOS incidence probability f are provided for reference. Of the four samples, only the first sample fails the Lilliefors test. One can deduce that since sample 1 had very few number of NLOS biases, with f equal to 0.0003, it could not represent a lognormal distribution. An intuitive explanation for the lognormal nature of NLOS distributions is that the generation of NLOS bias can be considered as a product of various independent random variables given the arbitrary shapes and spacing of clutter objects. This fact can be taken as a result of the central limit theorem in the log-domain.

Next, we use the k-NN technique to estimate the (μ, σ) pair of a lognormal-fitted NLOS bias distribution $\ln\mathcal{N}(\mu, \sigma)$. In this estimation process, the error is calculated as:

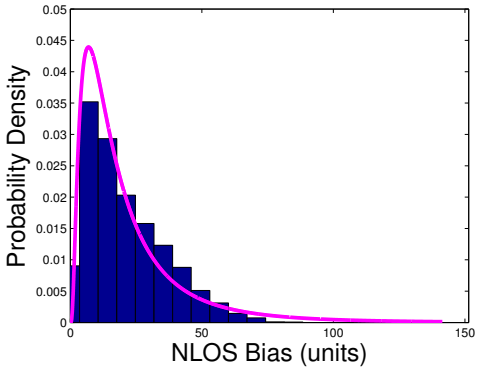
$$err_{\mathcal{P}l} = \sqrt{(\mu_{est} - \mu^2_{actual}) + (\sigma_{est} - \sigma^2_{actual})} \tag{7.7}$$



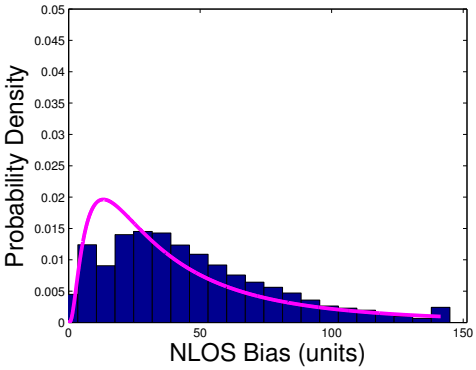
(a) Sample 1: $f = 0.0003$, ($\mu_{\mathcal{P}l} = 2.7195$, $\sigma_{\mathcal{P}l} = 0.8849$)



(b) Sample 2: $f = 0.2544$, ($\mu_{\mathcal{P}l} = 4.0433$, $\sigma_{\mathcal{P}l} = 0.8104$)



(c) Sample 3: $f = 0.5007$, ($\mu_{\mathcal{P}l} = 3.3256$, $\sigma_{\mathcal{P}l} = 0.7721$)



(d) Sample 4: $f = 0.7533$, ($\mu_{\mathcal{P}l} = 3.5132$, $\sigma_{\mathcal{P}l} = 0.9651$)

Figure 7.18: Lognormal distribution fitting ($\mathcal{P}l$) of the original histogram-based NLOS bias distributions ($\mathcal{P}$).

where $\ln\mathcal{N}(\mu_{est}, \sigma_{est})$ is the estimated NLOS bias distribution and the actual lognormal-fitted NLOS bias distribution is represented as $\ln\mathcal{N}(\mu_{actual}, \sigma_{actual})$. We perform the same experiments as in Section 7.3.3.2 using a training set of $ntrn$ (= 10000) and test set of n_{ts} (= 5000) distinct clutter topologies. The k-NN algorithm is initially trained with the training data. As in the previous case, we set k=10 for our analysis.

We analyze the efficacy of various combinations of features in predicting $\mathcal{P}l$ for the test set. We use exhaustive combinations of features, with two, three and four features together. We list the results for feature sets that give the best results (lowest $err_{\mathcal{P}l}$) in Table 7.6. We use the mean and standard deviation of $err_{\mathcal{P}l}$ for the entire test set to measure the accuracy of NLOS bias distribution estimation. $\mathcal{F}^n_{\mathcal{P}l}$ represents the feature set that produces the least $err_{\mathcal{P}l}$ when any combination of n features is used. $\mathcal{B}_{\mathcal{P}l}$ represents a baseline solution with

Estimation Technique using k-NN	$err_{\mathcal{P}l}$ (mean)	$err_{\mathcal{P}l}$ (std)
$\mathcal{B}_{\mathcal{P}l}$	0.2721	0.2206
$\mathcal{F}^1_{\mathcal{P}l} = \{\mathcal{C}\}$	0.2080	0.1988
$\mathcal{F}^2_{\mathcal{P}l} = \{\mathcal{C}, CD_r\}$	0.1774	0.1611
$\mathcal{F}^3_{\mathcal{P}l} = \{ca, \mathcal{C}, CD_r\}$	0.1727	0.1608
$\mathcal{F}^4_{\mathcal{P}l} = \{ca, \mathcal{C}, CD_l, CD_r\}$	0.2136	0.1788

Table 7.6: Estimation of NLOS lognormal-fitted bias distributions $\mathcal{P}l$ using k-NN. Here, the set $\mathcal{F}^i_{\mathcal{P}l}$ denotes the set of i features which achieve the minimum $err_{\mathcal{P}l}$. $\mathcal{P}l$ represents a log-normal fitted version of the original NLOS bias distribution $\mathcal{P}$.

the given training data by taking the average of all $\mathcal{P}l$ in the training data. In other words, $\mathcal{B}_{\mathcal{P}l} = \ln\mathcal{N}(\mu_B, \sigma_B,$ where $\mu_B = \frac{1}{ntrn}\sum_{i=1}^{ntrn}\mu_i$, $\sigma_B = \frac{1}{ntrn}\sum_{i=1}^{ntrn}\sigma_i$ and $\ln\mathcal{N}(\mu_i, \sigma_i)$ represents the $\mathcal{P}l$ of the i^{th} training sample.

When compared to the results in Table 7.5 for histogram-based NLOS bias distributions, the results for lognormal fitted NLOS bias distributions are better. The standard deviation of $err_{\mathcal{P}l}$ is almost halved in Table 7.6 when compared to Table 7.5. Again, we find that the communication range $\mathcal{C}$ has a strong clustering effect on the results. In other words, NLOS bias distributions obtained with a particular value of $\mathcal{C}$ tend to be significantly different from those obtained with other values of $\mathcal{C}$. CD_r and ca contribute positively towards reducing estimation error. CD_l, however, has a detrimental effect on the estimation error by actually increasing its magnitude, similar to the results seen in Table 7.5.

Finally, in Fig. 7.19, we look at the lognormal-fitted NLOS bias estimation for the sample clutter topology, for which we demonstrated the performance of various features in the k-NN estimation (shown in Fig. 7.16). We find that the feature set $\{ca, \mathcal{C}, CD_r\}$ performs the best and gives an estimated $\ln\mathcal{N}(\mu, \sigma)$ that closely matches the actual NLOS bias distribution.

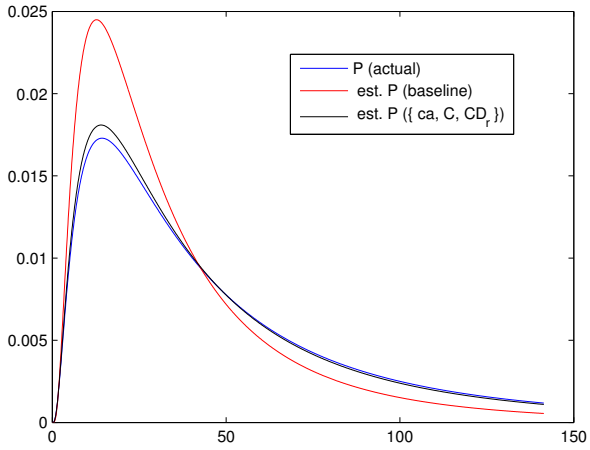


Figure 7.19: Estimation of lognormal fitted $\mathcal{P}$ using k-NN

7.4 Effect of Clutter Topology on Multi-Hop Error

Previously, we studied the impact of clutter topology on NLOS bias. We will now look at the effect of clutter topology on multi-hop distance error, particularly the overestimate distance error of OPDV-mandated localizer chains. The OPDV algorithm, introduced in Chapter 4, outputs positions for intermediate localizers to be placed between two nodes (for example, an anchor and a sensor robot) in the midst of clutter such that the multi-hop overestimate error is minimized. This error depends on the positions of the two ranging nodes and the clutter in between. As the case with NLOS bias, it is difficult to arrive at an analytical formulation that encapsulates the multi-hop error with respect to the above mentioned factors. Instead, we use characteristic features of the clutter topology to estimate multi-hop error via machine learning techniques.

7.4.1 Experimental Setup

In this section, we will describe the setup for generating the experimental data. We aim to evaluate the multi-hop distance error when one or more localizers are used between two ranging nodes, for our case here — an anchor and a sensor robot. We use the basic version of the OPDV algorithm, described in Chapter 4, for obtaining the localizer positions in the multi-hop chain between an anchor and the sensor robot. The reason for this is that here we want to understand the multi-hop distance error with respect to only the clutter

topology and by introducing a localizer limit (N_{loc}) factor, we will introduce another external factor for the multi-hop distance error. For the same reason, we set the communication range of both anchors and localizers to the maximal value of D, the diameter of the cluttered environment. The position of the anchor is randomly chosen along the boundary of the enclosure, while the sensor robot position is chosen as a random position in the enclosure not occupied by clutter.

Clutter Topology Generation. We build complex clutter topologies, using the technique outlined in Section 7.3.1, by sequentially overlaying simple rectilinear structures over each other. As was the case with the NLOS bias analysis, we use a 2D rectangular enclosure area with dimensions L_x and L_y. We then generate ncl boxes with random start points (x_i, y_i) and dimensions lx and ly, where $lx = \mathcal{U}(x_i, x_i + \frac{L_x}{dimfac})$ and $ly = \mathcal{U}(y_i, y_i + \frac{L_y}{dimfac})$. The values of ncl and $dimfac$ are determined by the clutter class. For each sample, we first randomly choose a clutter topology class from the three clutter classes FL (few large-size clutter), MS (many small-size clutter) and FS (few small-size clutter). In the case of clutter class FS, we set ncl as a randomly chosen natural number between [1, 8] and $dimfac = 3$. In the case of clutter class MS, ncl is randomly chosen between [1, 20] and $dimfac = 8$. Finally, in the case of clutter class FS, ncl is chosen between [1, 8] and $dimfac$ is set to 8.

For each clutter topology sample, we calculated the OPDV multi-hop paths for $N_o(= 1000)$ pairs of anchor and sensor robot positions. The multi-hop error is calculated as the overestimate error between the cumulative multi-hop distance and the true distance. We choose three types of aggregates for representing the multi-hop error: mean multi-hop error, median multi-hop error and maximum multi-hop error. The median is more resilient to outliers than the mean. The purpose of these aggregate forms of the multi-hop error is to help evaluate the efficacy of using multi-hop localization for a given clutter topology.

7.4.2 Estimation of OPDV Multi-Hop Error

In this section, we will look at the estimation of the OPDV multi-hop error for an arbitrary clutter topology when we are only given features characterizing the clutter topology. We use a Support Vector Regressor (SVR) that learns a given training set and is able to predict the aggregate value of the multi-hop error. The details of the SVR are mentioned previously in Section 7.3.2.1.

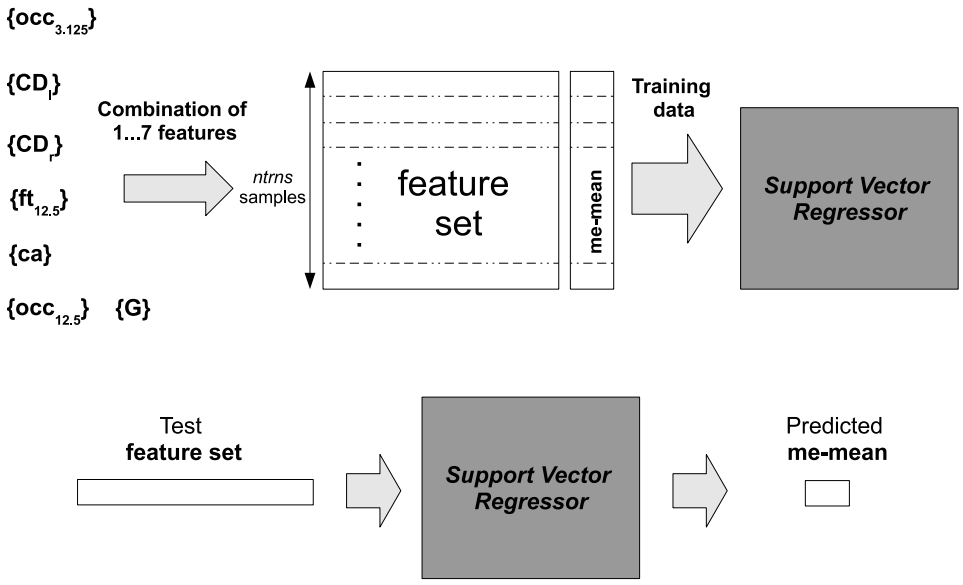


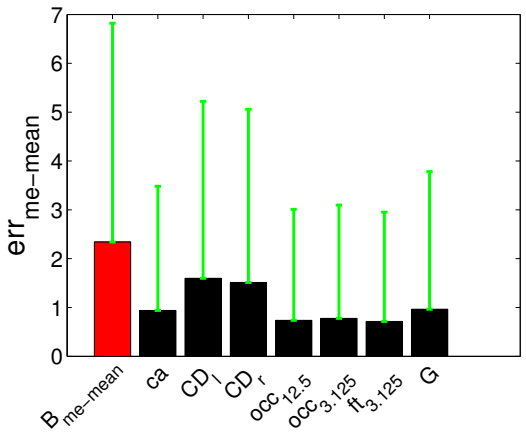
Figure 7.20: Estimation of mean OPDV multi-hop error (*me-mean*) using Support Vector Regressor (SVR). A similar procedure is used for estimating median multi-hop error (*me-median*) and maximum multi-hop error (*me-max*).

Estimation Technique using SVR	$err_{me-mean}$ (mean)	$err_{me-mean}$ (std)
$\mathcal{B}_{me-mean}$	2.3425	2.1367
$\mathcal{F}^1_{me-mean} = \{ft_{3.125}\}$	0.7139	1.5251
$\mathcal{F}^2_{me-mean} = \{occ_{3.125},\ ft_{3.125}\}$	0.7166	1.5243
$\mathcal{F}^3_{me-mean} = \{occ_{12.5},\ occ_{3.125},\ ft_{3.125}\}$	0.7167	1.5230
$\mathcal{F}^4_{me-mean} = \{ca,\ occ_{12.5},\ occ_{3.125},\ ft_{3.125}\}$	0.7164	1.5227

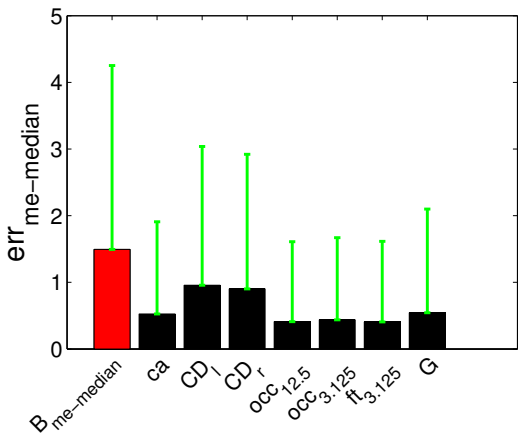
Table 7.7: Estimation of *me-mean* with combination of clutter topology features. Here, the set $\mathcal{F}^i_{me-mean}$ denotes the set of i features which achieve the minimum $err_{me-mean}$.

Here too we use the non-linear ϵ-SVR with the Radial Basis (RBF) kernel. We use a combination of various features of the clutter topology to predict *mean multi-hop error* (*me-mean*), *median multi-hop error* (*me-median*) and *maximum multi-hop error* (*me-max*). Fig. 7.20 shows the schematic diagram for the entire process of training and predicting aggregate multi-hop error values with the SVR.

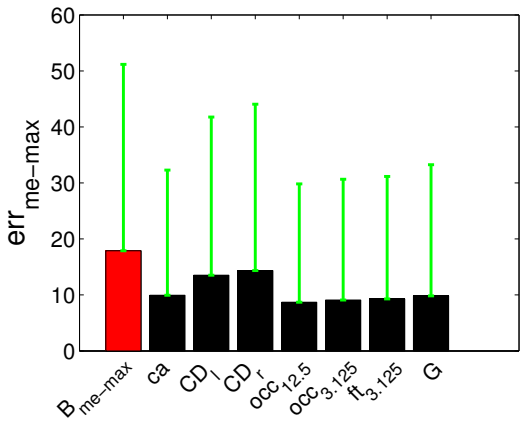
For a given test clutter topology, we represent the estimation error for multi-hop error mean as $err_{me-mean}$, the estimation error for multi-hop error median as $err_{me-median}$, and the estimation error for the multi-hop error maximum as err_{me-max}. The performance of the individual features for each of the aggregate



(a) Estimation accuracy of *me-mean*.



(b) Estimation accuracy of *me-median*.



(c) Estimation accuracy of *me-max*.

Figure 7.21: SVR based estimation of *me-mean*, *me-median* and *me-max* with individual features of clutter topology.

values of multi-hop error is shown in Fig. 7.21. Similarly, the best performances of using 2, 3 and 4 feature combinations for *me-mean*, *me-median* and *me-max* are presented in Tables 7.7, 7.8 and 7.9 respectively.

We see that features $occ_{3.125}$, $occ_{12.5}$ and $ft_{3.125}$ perform the best among all other features. $ft_{3.125}$ performs the best in case of estimating *me-mean* and *me-median*, while $occ_{12.5}$ offers the highest estimation accuracy for *me-max*. We see that using combinations of two or more features does not lead to any significant improvement in estimation accuracy. We also see that the standard deviations for all the estimations are larger than the means, indicating that the SVR estimator is not able to perform well for a small minority of test

Estimation Technique using SVR	$err_{me-median}$ (mean)	$err_{me-median}$ (std)
$\mathcal{B}_{me-median}$	1.4946	1.2637
$\mathcal{F}^1_{me-median} = \{ft_{3.125}\}$	0.4062	0.8031
$\mathcal{F}^2_{me-median} = \{occ_{3.125}, ft_{3.125}\}$	0.4019	0.7718
$\mathcal{F}^3_{me-median} = \{occ_{12.5}, occ_{3.125}, ft_{3.125}\}$	0.4020	0.7711
$\mathcal{F}^4_{me-median} = \{ca, occ_{12.5}, occ_{3.125}, ft_{3.125}\}$	0.4018	0.7708

Table 7.8: Estimation of *me-median* with combination of clutter topology features. Here, the set $\mathcal{F}^i_{me-median}$ denotes the set of i features which achieve the minimum $err_{me-median}$.

Estimation Technique using SVR	err_{me-max} (mean)	err_{me-max} (std)
$\mathcal{B}_{me-max}$	17.8846	15.4092
$\mathcal{F}^1_{me-max} = \{occ_{12.5}\}$	8.7026	12.4167
$\mathcal{F}^2_{me-max} = \{occ_{12.5}, ca\}$	8.6956	12.4052
$\mathcal{F}^3_{me-max} = \{occ_{12.5}, occ_{3.125}, occ_{3.125}\}$	8.8393	12.4456
$\mathcal{F}^4_{me-max} = \{ca, occ_{12.5}, occ_{3.125}, ft_{3.125}\}$	8.8614	12.7020

Table 7.9: Estimation of *me-max* with combination of clutter topology features. Here, the set $\mathcal{F}^i_{me-max}$ denotes the set of i features which achieve the minimum err_{me-max}.

samples, which give large estimation errors. Nevertheless, these results show that employing clutter topology information delivers much better performance than the baseline, which simply represents the average of all training data, in the estimation of the multi-hop error.

On comparing these results with those of NLOS bias estimation in Section 7.3, we note that the spacing distributions, CD_l and CD_r, do not perform as well as the other features, unlike the case with NLOS bias estimation. The reason could be that the multi-hop paths between two ranging nodes in the clutter is hardly affected by the spacing between the clutter objects. Instead they depend on the positions of the clutter with respect to the positions of the two ranging nodes. In the case of NLOS incidence probability, the spacing between the clutter indirectly determines that probability of an NLOS distance between two random points in the given cluttered environment. Similarly, the spacing between the clutter objects has a strong influence on the length of the reflected signal that determines the NLOS bias. Thus, we see that multi-hop error is mainly determined by the relative positions of the clutter objects with respect to the enclosure, represented by the features $occ_{12.5}$, $occ_{3.125}$ and $ft_{3.125}$.

7.5 Discussion

In this chapter, we explored the influence of clutter topology on two types of distance errors: non-line-of-sight (NLOS) bias and overestimate error of the multi-hop localizer chains generated by OPDV. Since arriving at an analytical formulation to connect the clutter topology directly to these distance errors is hard, given the arbitrariness of the clutter topologies and the relative positions of the two ranging nodes within the cluttered environment, we turn to employing machine learning techniques on empirically derived data. First, we enumerate the characteristic features of a given clutter topology in Section 7.2. Thereafter, we look at the estimation of NLOS error in Section 7.3 and multi-hop error in Section 7.4. We are able to estimate NLOS incidence probability with an error of 0.03. In case the estimation of NLOS bias distribution, we show that, while the estimation does not work well for a small minority of test samples, we are able to achieve a high estimation accuracy for the remaining test samples. We saw that the clutter spacing distributions, CD_l and CD_r, perform exceptionally well in the case of estimating both NLOS probability and bias distributions. This is encouraging as these features can be extracted through practical means in an unknown clutter topology, while the other features, such as the occupancy grids, require complete knowledge of the clutter topology — something that is not always practically possible in realistic deployment scenarios. In the case of estimation of the aggregate values of the multi-hop error for a given clutter topology, we see that features such as the occupancy grid perform better than clutter spacing distributions, given the minor influence that the spacing between clutter objects has on the multi-hop paths outputted by OPDV between the ranging nodes. We saw in Section 7.4 that, while the SVR-based estimation of multi-hop error performs well for the majority of test cases, the poor performance of a minority of the test samples leads to a large variance in the estimation error. We are able to predict the mean, median and maximum OPDV error with an average error of only 0.72, 0.41 and 8.69 units respectively, for an enclosure area of 10,000 square units. Thus, we can use this estimation to evaluate if shortest path distance (SPD) based multi-hop localization algorithms, like DV-Distance and MDS-MAP, can be employed to achieve a given localization accuracy.

In the next chapter, we conclude this dissertation by highlighting its key contributions and discussing its limitations and directions for future work.

Chapter 8

Conclusion and Future Work

In this chapter, we will conclude the dissertation and discuss future directions of research. We begin by summarizing the main contributions of the dissertation in Section 8.1. We enumerate the limitations of this work in Section 8.2. Finally, in Section 8.3, we discuss ideas for future work.

8.1 Summary of Contributions

In this section we summarize the main contributions of this dissertation:

1. We introduced the concept of intermediate nodes termed *localizers* for improving localization accuracy in cluttered NLOS-prone environments. In Chapter 3, we evaluated a number of multi-hop localization algorithms that employ localizers in such environments and compared their performance with each other and with single hop localization under a variety of parameters such as communication range and localizer density. We showed that the multi-hop localization techniques (namely DV-Distance and MDS-MAP) that used shortest path multi-hop distances to the anchors, in lieu of direct distances, perform much better than single-hop localization and iterative localization. Such techniques benefit from using minimal multi-hop distances instead of the NLOS prone direct distances.

2. We proposed the Optimal Placement for DV-Distance (OPDV) localizer placement algorithm as a centralized oracle-type algorithm that takes into account the complete clutter topology to generate optimal positions for localizers that minimize the multi-hop distance errors between the anchors and a sensor robot. We proposed a basic version and constrained-localizer version for OPDV.

3. We proposed Adaptive Placement for DV-Distance (APDV) as a distributed localizer placement algorithm that can be implemented in practical settings. APDV does not require the localizers to know the clutter topology or even localize themselves in order to move to positions that improve the localization accuracy of a sensor robot. Furthermore the APDV algorithm does not require explicit identification of LOS and NLOS distances and it works in situations where NLOS distances constitute the vast majority of anchor-to-robot distances. Both OPDV and APDV algorithms are described in Chapter 4.

4. We evaluated the OPDV and APDV algorithms through extensive simulations in Chapter 5. We studied the impact of various OPDV parameters such as communication range, grid spacing and number of localizers on the performance of OPDV. We similarly looked at the impact of APDV parameters such as step-size threshold, number of probing directions and number of localizers, as well as external factors like robot odometry errors and line-of-sight (LOS) distance errors on the efficacy of APDV. We show that APDV approaches oracle-type OPDV in localization accuracy, despite being a distributed algorithm and not requiring explicit clutter topology information. We show that APDV outperforms competing techniques in scenarios where most distances are NLOS in nature.

5. We described, in Chapter 6, an implementation of APDV on real robots. We evaluated the working of APDV for a variety of clutter topologies and showed that we can achieve almost 95% reduction in the multi-hop distance error.

6. Finally in Chapter 7, we evaluated the relationship between clutter topology and distance errors, both NLOS bias as well multi-hop overestimate error. We defined characteristic features for clutter topologies and used machine learning techniques on empirical data to understand the effect of clutter topology features on NLOS bias and multi-hop distance error. We showed that, using clutter topology features, we can estimate NLOS bias and multi-hop error with good accuracies in a majority of test cases.

8.2 Limitations

The work done in this dissertation has the following limitations:

1. **Optimal localizer placement for a given area**. Currently the OPDV algorithm gives optimal localizer placements for a given position of a *single* sensor robot, which can be extended to discrete points on a trajectory through the cluttered environment. This entails new positions for the localizers for every new step of the sensor robot. While this yields the best improvement for the sensor robot in terms of localization accuracy, there might be scenarios where localizers cannot be moved around often. In this scenario, we need to look at the problem of optimal localizer placement for a *given deployment area* that minimise an aggregate function of distance error over the entire deployment area.

2. **Limited range of Cricket motes in NLOS settings**. Due to the severely reduced operating range of the Cricket motes in the presence of clutter (approximately 400-500 cm for consistent distance measures), the experiments described in Chapter 6 are mainly limited to showing that the APDV algorithm is successful in reducing multi-hop distance overestimates in small deployment areas.

3. **Effect of robot odometry errors on APDV performance**. As seen in Chapter 5, robot odometry errors, especially rotational odometry errors, can have a significant impact on the performance of APDV. This problem cannot even be remedied by repeating the alignment procedure multiple times, since APDV makes the basic assumption that *a robot is able to return to its original position* after the exploratory steps taken during the alignment procedure. Even small inconsistencies in returning to the original position can lead to NLOS distance measurements between the robot and its neighbours.

4. **APDV completion time/energy costs**. The APDV algorithm can take considerable time and energy to complete, particular if the EF variant is used, thus creating a tradeoff between localization accuracy and cost/delay. For example, localizers need to travel an average cumulative distance of 150 steps while assisting a sensor robot at a single static position.

5. **Initial localizer positions**. The dissertation does not investigate strategies for initial localizer deployments, particularly for situations where a

sensor robot has to follow a pre-determined trajectory through the cluttered environment. As we have shown in Chapter 5, the positions of the localizers at the start of APDV have a significant impact on the subsequent performance of the algorithm.

6. **Uncertain clutter topology information**. In Chapter 7, we see that it is possible to use certain features of the clutter topology to characterize the distance error obtained when two nodes are ranging in the cluttered environment. However, in realistic scenarios, it might not be possible to obtain an accurate and complete picture of the clutter topology, something that features like occupancy grids and Fourier characterization require. The dissertation does not look into the problem of distance error characterization in the presence of uncertain clutter topology information.

8.3 Directions for Future Work

Finally, we conclude this chapter by discussing directions for future work. Some of these directions would help overcome limitations outlined in the previous section. Others are natural extensions of the research done in this dissertation.

1. **Optimal localizer placement over entire deployment area**. It would be interesting to devise algorithms that produce optimal localizer placements that minimize aggregate error over an entire deployment area. For example, we can find the positions of N localizers in the cluttered environment such that the maximum error over the entire (discretized) area is minimized. With such a set of optimal localizer positions, one can envisage deploying a group of sensor robots with the assurance that their maximum localization error is *bounded for every position* in the cluttered environment. The challenge here is to develop an approximation algorithm that achieves the same result as that of the brute-force approach, while being practical in terms of computational cost.

2. **Alternative validations of the APDV algorithm**. We can further evaluate the performance of APDV in a variety of hardware and topologies. The evaluation of APDV discussed in Chapter 6, with the omni-directional dual-Cricket motes formation, suffers from large errors in the

presence of clusters of erroneous distances, even for line-of-sight (LOS) scenarios. Superior omni-directional ranging hardware can significantly improve the quality of the evaluation. For example, the SpiderBat ultrasound positioning system [99] uses 8 ultrasound transducers to emulate omni-directional time-difference-of-arrival ranging and gives ranging error with only 0.539 cm standard deviation for a distance of almost 1400 cm. We can obtain a better evaluation of APDV even with numerous actual NLOS distances, if we only replace the current dual-Cricket mote formation with the Spiderbat.

3. **Optimizing APDV operating costs**. We could investigate ways of optimizing the completion time of APDV, thus making it more applicable to scenarios that demand low latency for localization. One possible direction is to look at coordinated alignment procedures for multiple localizers in the multi-hop chain between the anchor and the sensor robot. Additionally, we can use other forms of sensor data such as inertial readings and obstacle detection sensor readings to improve the speed and reduce cost during the alignment procedure.

4. **Strategies for initial localizer deployment for APDV algorithm**. Strategies for distributing a group of localizers in the cluttered environment, before the sensor robot actually starts on its pre-determined trajectory, need to be developed. The key challenge here is to move the localizers to cover the entire cluttered environment without requiring them to localize using range-based localization. We can look at dispersal algorithms for mobile robots [81, 102] that use obstacle-sensor information as well as inter-robot communication to achieve this task.

5. **Distance error characterization with uncertain clutter topology information**. Efforts can be made to investigate the distance error characterization, discussed in Chapter 7, in situations where the information of the area's clutter topology is incomplete.

Finally, to conclude, in this dissertation, we looked at the novel application of multi-hop localization techniques in cluttered NLOS-prone environments. We demonstrated that localizer placement algorithms, which can carefully place intermediate nodes amidst clutter, significantly improve the resulting localization accuracy. We also showed that clutter topology does have a definite impact

on distance errors in cluttered environments and also quantified the effect of various features of a clutter topology through estimation of distance error itself. This dissertation differs from previous work because it considers the scenario where the majority of anchor distances are NLOS in nature, while at the same time, not requiring any explicit identification of NLOS distances. It shows that using multi-hop localization with careful localizer placements outperforms current state-of-art research. This work can be applied in a number of real-world applications such as industrial environmental monitoring and disaster recovery, where robots have to operate in cluttered environments. The concept of multi-hop localization can also be extended to improve accuracy of GPS positioning devices in the urban clutter of cities.

Appendix A

OPDV Pseudocode

Procedure A.1: Input graph construction

Data: Enclosure Dimensions D_{encl}, Clutter configuration C_{cl}, Grid Granularity gs_{opdv}, Anchor positions A_1, A_2,....,A_l, Sensor robot position S, Anchor communication range $\mathcal{C}_a$, Localizer Communication range $\mathcal{C}_l$, NLOS edge function $\Phi(.)$

Result: OPDV Input Graph $\mathcal{G}$(V,E)

```
Add A_1, A_2,....,A_l to V ;
Get set of candidate localizer positions, S_lc, based on D_encl, C_cl and
gs_opdv ;
Add S_lc to V ;
Add P to V ;
forall the elements in V, v do
    if v is anchor vertex then
        communcation range C ← C_a
    else if v is localizer vertex then
        communcation range C ← C_l
    else v is sensor robot vertex
        communcation range C ← 0
    end
    Get neighbour set N of v using C ;
    forall the elements in N, n do
        d ← euclidean distance between n and v, ||n, v|| ;
        if clutter exist between n and v then
            Add edge (v,n) to E, with edge weight Φ(.) ;
        else
            Add edge (v,n) to E, with edge weight d ;
        end
    end
end
```

Procedure A.2: A* search

Data: $\mathcal{G}$(V,E), source node s, destination node d
Result: DV-path

Initialize priority queue pq ;
Closed set $\mathcal{CS} \leftarrow \emptyset$;
Cost set $\mathcal{CO} \leftarrow \emptyset$;
forall the *elements in V, v* **do**
  if *edge (s, v) exists* **then**
    $\mathcal{CO}[v] \leftarrow$ weight of (s, v) ;
  else
    $\mathcal{CO}[v] \leftarrow \infty$;
  end
end
pq.push(source node s) ;
while *pq is not empty* **do**
  $c \leftarrow$ pq.pop();
  if *c is destination node d* **then**
    return DV-path upto d ;
  else
    Add c to $\mathcal{CS}$;
    foreach *neighbour n of c* **do**
      if $n \notin \mathcal{CS}$ **then**
        gt $\leftarrow$ cost(s,c) + edge cost (c,n) ;
        if *($n \notin pq$)* **and** *($gt < \mathcal{CO}(n)$)* **then**
          g(n) $\leftarrow$ gt ;
          f(n) $\leftarrow$ gt + $\|n, d\|$;
          pq.push(n)
        else
          Ignore n ;
        end
      else
        Ignore n ;
      end
    end
  end
end

Algorithm A.3: Basic version of OPDV

Data: Enclosure Dimensions D_{encl}, Clutter configuration C_{cl}, Grid Granularity gs_{opdv}, Anchor positions A_1, A_2,....,A_l, Sensor robot position S, Anchor communication range $\mathcal{C}_a$, Localizer communication range $\mathcal{C}_l$, NLOS edge function Φ(.)
Result: DV paths

Use **input graph construction** procedure to obtain $\mathcal{G}$(V,E) ;
foreach *Anchor* A_i **do**
  Obtain DV-path between A_i and S using **A* search** procedure on $\mathcal{G}$;
end

Procedure A.4: Bellman-Ford search

Data: $\mathcal{G}$(V,E), source node s, destination node d, localizer limit H
Result: DV-path

```
Initilialize C[1..|V|][0..H] as C[s][h]=Inf, C[u][h]=0 for u!=s, for h=0..H;
Initilialize P[1..|V|][0..H] as P[g][h]={} for g=1..|V|, h=0..H;
for h ← 1 to H do
    for i ← 1 to |V| do
        C[i][h] ← C[i][h-1];
    end
    for j ← 1 to |E| do
        Let (u,v) represent j^th edge;
        Let w represent weight of j^th edge;
        if (C[u][h-1] + w) < C[v][h] then
            C[v][h] ← C[u][h-1] + w;
            P[v][h] ← {P[u][h-1]; v};
        end
    end
end
return P[d][H];
```

Algorithm A.5: Constrained-localizer version of OPDV

Data: Enclosure Dimensions D_{encl}, Clutter configuration C_{cl}, Grid Granularity gs_{opdv}, Anchor positions A_1, A_2,....,A_l, Sensor robot position S, Anchor communication range $\mathcal{C}_a$, Localizer communication range $\mathcal{C}_l$, NLOS edge function $\Phi(.)$, Number of available localizers N_{loc}
Result: DV paths

```
Use input graph construction procedure to obtain G(V,E);
Na ← l;
Nr ← N_loc;
foreach anchor A_i do
    N_la ← ceil(Na/Nr);
    Obtain DV-path between A_i and S using BF search procedure on G;
    Na ← Na− (number of localizers in DV-path for A_i);
    Nr ← Nr − 1;
end
```

Bibliography

[1] International Atomic Energy Agency. http://www.iaea.org/pris/.

[2] A. Agiwal, P. Khandpur, and H. Saran. LOCATOR: location estimation system for wireless LANs. In *Proceedings of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, pages 102–109. ACM, 2004.

[3] B. Alavi and K. Pahlavan. Modeling of the distance error for indoor geolocation. In *IEEE Wireless Communications and Networking Conference (WCNC)*, volume 1, pages 668–672. IEEE, 2003.

[4] J. Albowicz, A. Chen, and L. Zhang. Recursive position estimation in sensor networks. In *Proceedings of the 9th International Conference on Network Protocols*, pages 35–41. IEEE, 2001.

[5] N. Alsindi, B. Alavi, and K. Pahlavan. Spatial characteristics of UWB TOA-based ranging in indoor multipath environments. In *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–6. IEEE, 2007.

[6] N.A. Alsindi, B. Alavi, and K. Pahlavan. Measurement and modeling of UltraWideband TOA-based ranging in indoor multipath environments. *IEEE Transactions on Vehicular Technology*, 58(3):1046–1058, 2009.

[7] K.O. Arras, N. Tomatis, and R. Siegwart. Multisensor on-the-fly localization using laser and vision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 462–467. IEEE, 2000.

[8] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[9] P. Bahl and V. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Annual Conference of the IEEE Computer and Communications Societies (INFOCOMM)*, volume 2, pages 775–784, 2000.

[10] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

[11] R. Beckwith, D. Teibel, and P. Bowen. Unwired wine: Sensor networks in vineyards. In *IEEE Sensors*, pages 561–564. IEEE, 2004.

[12] M. Betke and L. Gurvits. Mobile robot localization using landmarks. *IEEE Transactions on Robotics and Automation*, 13(2):251–263, 1997.

[13] J. Borras, P. Hatrack, and N.B. Mandayam. Decision theoretic framework for NLOS identification. In *IEEE Vehicular Technology Conference*, volume 48, pages 1583–1587. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE), 1998.

[14] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework. *The International Journal of Robotics Research (IJRR)*, 23(12):1113–1139, 2004.

[15] M. Bryson and S. Sukkarieh. Bearing-only SLAM for an airborne vehicle. In *Australasian Conf. Robot. Autom.(ACRA 2005), Sydney, Australia*, 2005.

[16] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, 2000.

[17] W. Burgard, D. Fox, and S. Thrun. Probabilistic robotics, 2005.

[18] J.J. Caffery and G.L. Stuber. Overview of radiolocation in CDMA cellular systems. *IEEE Communications Magazine*, 36(4):38–45, 1998.

[19] S. Čapkun, M. Hamdi, and J.P. Hubaux. GPS-free positioning in mobile ad hoc networks. *Cluster Computing*, 5(2):157–167, 2002.

[20] R. Casas, A. Marco, JJ Guerrero, and J. Falco. Robust estimator for non-line-of-sight error mitigation in indoor localization. *EURASIP Journal on Applied Signal Processing*, 2006:156–156, 2006.

[21] Y.T. Chan, W.Y. Tsui, H.C. So, and P. Ching. Time-of-arrival based localization under NLOS conditions. *IEEE Transactions on Vehicular Technology*, 55(1):17–24, 2006.

[22] C.L. Chen and K.T. Feng. An efficient geometry-constrained location estimation algorithm for NLOS environments. In *IEEE Wireless Communications and Networking Conference (WCNC)*, volume 1, pages 244–249. IEEE, 2005.

[23] P.C. Chen. A non-line-of-sight error mitigation algorithm in locationestimation. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 316–320, 1999.

[24] K.S. Chong and L. Kleeman. Feature-based mapping in real, large scale environments using an ultrasonic array. *The International Journal of Robotics Research (IJRR)*, 18(1):3–19, 1999.

[25] W.S. Cleveland. Lowess: A program for smoothing scatterplots by robust locally weighted regression. *The American Statistician*, 35(1):54–54, 1981.

[26] T.H. Cormen. *Introduction to Algorithms*. The MIT press, 2001.

[27] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[28] A. J. Davison, Y. G. Cid, and N. Kita. Real-time 3D SLAM with wide-angle vision. In *Proc. IFAC/EURON Symp. Intelligent Autonomous Vehicles*, 2004.

[29] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[30] M. Deans and M. Hebert. Experimental comparison of techniques for localization and mapping using a bearing-only sensor. In *Experimental Robotics VII*, pages 395–404. Springer, 2001.

[31] R. Dechter and J. Pearl. Generalized best-first search strategies and the optimality of A*. *Journal of the ACM (JACM)*, 32(3):505–536, 1985.

[32] R.P. d'Entremont. Performance of the discrete fourier transform satellite imagery classification technique. Technical report, DTIC Document, 1980.

[33] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[34] J. Djugash, S. Singh, G. Kantor, and W. Zhang. Range-only SLAM for robots operating cooperatively with sensor networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2078–2084. IEEE, 2006.

[35] L. Doherty, K.S.J. Pister, and L. El Ghaoui. Convex position estimation in wireless sensor networks. In *Annual Conference of the IEEE Computer and Communications Societies (INFOCOMM)*, volume 3, pages 1655–1663, 2001.

[36] S.P. Drake and K. Doğançay. Geolocation by time difference of arrival using hyperbolic asymptotes. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 361–4, 2004.

[37] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006.

[38] J.S. Esteves, A. Carvalho, and C. Couto. Generalized geometric triangulation algorithm for mobile robot absolute self-localization. In *IEEE International Symposium on Industrial Electronics (ISIE)*, volume 1, pages 346–351. Ieee, 2003.

[39] Federal Communications Commission (FCC). 911-Wireless Services. http://www.fcc.gov/guides/wireless-911-services/.

[40] HT Friis. A note on a simple transmission formula. *Proceedings of the IRE*, 34(5):254–256, 1946.

[41] S. Gezici, H. Kobayashi, and H.V. Poor. Nonparametric nonline-of-sight identification. In *IEEE Vehicular Technology Conference*, volume 4, pages 2544–2548. IEEE, 2003.

[42] S. Gezici and Z. Sahinoglu. UWB geolocation techniques for IEEE 802.15.4a personal area networks. *Mitsubishi Electric Research Laboratory Technical Report TR-2004-110*, 2004.

[43] J. E. Guivant and E. M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 2001.

[44] F. Gustafsson and F. Gunnarsson. Positioning using time-difference of arrival measurements. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 6. IEEE, 2003.

[45] I. Guvenc and C.C. Chong. A survey on TOA based wireless localization and NLOS mitigation techniques. *IEEE Communications Surveys & Tutorials*, 11(3):107–124, 2009.

[46] I. Guvenc, C.C. Chong, and F. Watanabe. NLOS identification and mitigation for UWB localization systems. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1571–1576. IEEE, 2007.

[47] I. Guvenc, C.C. Chong, F. Watanabe, and H. Inamura. NLOS identification and weighted least-squares localization for UWB systems using multipath channel statistics. *EURASIP Journal on Advances in Signal Processing*, 2008:36, 2008.

[48] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[49] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *The 9th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, page 95. ACM, 2003.

[50] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global positioning System GPS: Theory and Practice*. Springer, Wien (Austria), 1993.

[51] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal. Efficient, generalized indoor wifi graphslam. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1038–1043. IEEE, 2011.

[52] JY Huang and Q. Wan. The CRLB for WSNs location estimation in NLOS environments. In *International Conference on Communications, Circuits and Systems (ICCCAS)*, pages 83–86. IEEE, 2010.

[53] P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[54] M. Hussain, Y. Aytar, A. Markham, and N. Trigoni. Characterization of Non-Line-of-Sight (NLOS) Bias via Analysis of Clutter Topology. In *IEEE/ION Position Location and Navigation Symposium (PLANS)*, 2012.

[55] G. İsmail, C. Chia-Chin, F. Watanabe, I. Hiroshi, et al. NLOS identification and weighted least-squares localization for UWB systems using multipath channel statistics. *EURASIP Journal on Advances in Signal Processing*, 2008, 2007.

[56] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal. ARIADNE: a dynamic indoor signal map construction and localization system. In *Proceedings of the 4th international conference on Mobile systems, applications and services*, page 164. ACM, 2006.

[57] T. Jia and R.M. Buehrer. Collaborative position location with NLOS mitigation. In *IEEE 21st International Symposium on Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops)*, pages 267–271. IEEE, 2010.

[58] D.B. Jourdan, D Dardari, and MZ Win. Position error bound for UWB localization in dense cluttered environments. *IEEE Transactions on Aerospace and Electronic*, 00:3705–3710, 2008.

[59] D.B. Jourdan, J.J. Deyst Jr, M.Z. Win, and N. Roy. Monte carlo localization in dense multipath environments using UWB ranging. In *IEEE International Conference on Ultra-Wideband*, pages 314–319. IEEE, 2005.

[60] D.B. Jourdan and N. Roy. Optimal sensor placement for agent localization. *ACM Transactions on Sensor Networks (TOSN)*, 4(3):13, 2008.

[61] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *ACM Sigplan Notices*, volume 37:10, pages 96–107. ACM, 2002.

[62] A.A. Kannan, G. Mao, and B. Vucetic. Simulated annealing based localization in wireless sensor network. In *IEEE Conference on Local Computer Networks (LCN)*. IEEE, 2005.

[63] J. Kim and S. Sukkarieh. Airborne simultaneous localisation and map building. In *International Conference on Robotics and Automation (ICRA)*, volume 1, pages 406–411. IEEE, 2003.

[64] W. Kim, J.G. Lee, and G.I. Jee. The interior-point method for an optimal treatment of bias in Trilateration location. *IEEE Transactions on Vehicular Technology*, 55(4):1291–1301, 2006.

[65] S. Kullback and R.A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[66] HT Kung, C.K. Lin, T.H. Lin, and D. Vlah. Localization with snap-inducing shaped residuals (SISR): Coping with errors in measurement. In *The 15th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 333–344. ACM, 2009.

[67] I. Kunttu, L. Lepisto, J. Rauhamaa, and A. Visa. Multiscale fourier descriptor for shape classification. In *Proceedings. 12th International Conference on Image Analysis and Processing (ICIAP)*, pages 536–541. IEEE, 2003.

[68] T. E. Kurt, P. Bouchier, and J. Pitts. Roombacomm java library. www.dprg.org/projects/2009-07a/, 2009.

[69] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks*, 43(4):499–518, 2003.

[70] S. Lanzisera, D. Zats, and K.S.J. Pister. Radio frequency time-of-flight distance measurement for low-cost wireless sensor localization. *IEEE Sensors Journal*, 11(3):837–845, 2011.

[71] J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, 1991.

[72] P. Levis, S. Madden, S. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, and E. Brewer. Tinyos: An operating system for sensor networks. In *Ambient intelligence*, pages 115–148. Springer, 2005.

[73] X. Li. An iterative NLOS mitigation algorithm for location estimation in sensor networks. *RN*, 4:1, 2006.

[74] Z. Li, W. Trappe, Y. Zhang, and B. Nath. Robust statistical methods for securing wireless localization in sensor networks. In *Proceedings of the 4th International Conference on Information Processing in Sensor Networks (IPSN)*, pages 91–98. IEEE, 2005.

[75] K. Lingemann, A. Nüchter, J. Hertzberg, and H. Surmann. High-speed laser localization for mobile robots. *Robotics and Autonomous Systems*, 51(4):275–296, 2005.

[76] J. Liu, Y. Zhang, and F. Zhao. Robust distributed node localization with error management. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 250–261. ACM New York, NY, USA, 2006.

[77] G. Mao, B. Anderson, and B. Fidan. Path loss exponent estimation for wireless sensor network localization. *Computer Networks*, 51(10):2467–2483, 2007.

[78] G. Mao, B. Fidan, and B.D.O. Anderson. Wireless sensor network localization techniques. *Computer Networks*, 51(10):2529–2553, 2007.

[79] S. Marano, W.M. Gifford, H. Wymeersch, and M.Z. Win. NLOS identification and mitigation for localization based on UWB experimental data. *IEEE Journal on Selected Areas in Communications (JSAC)*, 28(7):1026–1035, 2010.

[80] M. Maróti, P. Völgyesi, S. Dóra, B. Kusỳ, A. Nádas, Á. Lédeczi, G. Balogh, and K. Molnár. Radio interferometric geolocation. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (Sensys)*, pages 1–12. ACM, 2005.

[81] J. McLurkin and J. Smith. Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots. In *Distributed Autonomous Robotic Systems 6*, pages 399–408. Springer, 2007.

[82] E. Menegatti, A. Zanella, S. Zilli, F. Zorzi, and E. Pagello. Range-only SLAM with a mobile robot and a wireless sensor networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 8–14. IEEE, 2009.

[83] M. J. Milford and G. F. Wyeth. Single camera vision-only SLAM on a suburban road network. In *International Conference on Robotics and Automation (ICRA)*, pages 3684–3689. IEEE, 2008.

[84] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the National Conference on Artificial Intelligence*, pages 593–598. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002.

[85] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 18, pages 1151–1156, 2003.

[86] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (Sensys)*, pages 50–61. ACM New York, NY, USA, 2004.

[87] R.H. Myers. *Classical and modern regression with applications*. Duxbury Press Boston, 1986.

[88] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Proceedings of the 1st International Conference on Information Processing in Sensor Networks (IPSN)*, pages 553–553. Springer, 2003.

[89] M. Nakagami. The m-distribution-a general formula of intensity distribution of rapid fading. *Statistical Method of Radio Propagation*, 1960.

[90] S. Nawaz, M. Hussain, S. Watson, N. Trigoni, and P. N. Green. An underwater robotic network for monitoring nuclear waste storage pools. *Sensor Systems and Software (SCUBE)*, pages 236–255, 2010.

[91] S. Nawaz and N. Trigoni. Robust localization in cluttered environments with NLOS propagation. In *IEEE 7th International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 166–175. IEEE, 2010.

[92] S. Nawaz and N. Trigoni. Convex programming based robust localization in NLOS prone cluttered environments. In *10th International Conference on Information Processing in Sensor Networks (IPSN)*, pages 318–329. IEEE, 2011.

[93] P. Newman, D. Cole, and K. Ho. Outdoor SLAM using visual appearance and laser ranging. In *International Conference on Robotics and Automation (ICRA)*, pages 1180–1187. IEEE, 2006.

[94] P. Newman and K. Ho. SLAM-loop closing with visually salient features. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 635–642. IEEE, 2005.

[95] P. Newman and J. Leonard. Pure range-only sub-sea SLAM. In *International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1921–1926. IEEE, 2003.

[96] P Newman, J Leonard, Juan D. Tardos, and Jose Neira. Explore and return: Experimental validation of real-time concurrent mapping and localization. In *International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1802–1809. IEEE, 2002.

[97] D. Niculescu and B. Nath. Ad hoc positioning system (aps). In *IEEE Global Telecommunications Conference (GLOBECOM)*, pages 2926–2931, 2001.

[98] D. Niculescu and B. Nath. DV based positioning in ad hoc networks. *Telecommunication Systems*, 22(1):267–280, 2003.

[99] D. Niculescu and B. Nath. Error characteristics of ad hoc positioning systems (APS). In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2004.

[100] G. Oberholzer, P. Sommer, and R. Wattenhofer. The spiderbat ultrasound positioning system. In *Proceedings of the 8th International Conference on Embedded Networked Sensor Systems (Sensys)*, pages 403–404. ACM, 2010.

[101] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision (IJCV)*, 42(3):145–175, 2001.

[102] N. Patwari and A.O. Hero III. Using proximity and quantized RSS for sensor localization in wireless networks. In *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA)*, pages 20–29. ACM, 2003.

[103] D. W. Payton, M. J. Daily, B. Hoff, M. D. Howard, and C. L. Lee. Pheromone robotics. In *Intelligent Systems and Smart Manufacturing*, pages 67–75. International Society for Optics and Photonics, 2001.

[104] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pages 90–100. IEEE, 1999.

[105] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson. Analysis of wireless sensor networks for habitat monitoring. *Wireless sensor networks*, pages 399–423, 2004.

[106] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *The 6th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 32–43. ACM, 2000.

[107] Y. Qi and H. Kobayashi. Cramer-Rao lower bound for geolocation in non-line-of-sight environment. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3. IEEE; 1999, 2002.

[108] T.S. Rappaport and Safari Books Online (Firme). *Wireless communications: principles and practice*, volume 2. Prentice Hall PTR Upper Saddle River (New Jersey), 1996.

[109] A.L. Read. Linear interpolation of histograms. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 425(1):357–360, 1999.

[110] C. Rohrig and M. Muller. Indoor location tracking in non-line-of-sight environments using a ieee 802.15.4a wireless network. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 552–557. IEEE, 2009.

[111] S.J. Russell, P. Norvig, J.F. Canny, J.M. Malik, and D.D. Edwards. *Artificial Intelligence: A Modern Approach*, volume 2. Prentice Hall, Englewood Cliffs, NJ, 1995.

[112] Z. Sahinoglu and S. Gezici. Ranging in the IEEE 802.15.4a standard. In *IEEE Annual Wireless and Microwave Technology Conference (WAMICON)*, pages 1–5. IEEE, 2006.

[113] Z. Sahinoglu, S. Gezici, and I. Guvenc. *Ultra-wideband positioning systems*. Cambridge University Press Cambridge, UK:, 2008.

[114] C. Savarese, J. Rabaey, and K. Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *USENIX technical annual conference*, volume 2. Monterey, CA, 2002.

[115] A. Savvides, W.L. Garber, R.L. Moses, and M.B. Srivastava. An analysis of error inducing parameters in multihop sensor node localization. *IEEE Transactions on Mobile Computing (TMC)*, 4(6):567–577, 2005.

[116] A. Savvides, C.C. Han, and M.B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *The 7th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 166–179. ACM New York, NY, USA, 2001.

[117] A. Savvides, H. Park, and M.B. Srivastava. The n-hop multilateration primitive for node localization problems. *Mobile Networks and Applications*, 8(4):443–451, 2003.

[118] A. Savvides, M. Srivastava, L. Girod, and D. Estrin. Localization in sensor networks. *Wireless sensor networks*, pages 327–349, 2004.

[119] Y. Shang and W. Ruml. Improved MDS-based localization. In *Annual Conference of the IEEE Computer and Communications Societies (INFOCOMM)*, volume 4, pages 2640–2651. IEEE, 2004.

[120] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz. Localization from mere connectivity. In *Proceedings of the 4th ACM International Symposium on Mobile ad hoc networking & computing (MobiHoc)*, pages 201–212. ACM, 2003.

[121] Y. Shang, H. Shi, and A.A. Ahmed. Performance study of localization methods for ad-hoc sensor networks. In *IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, pages 184–193. IEEE, 2004.

[122] W. Shi and V.W.S. Wong. A two-phase algorithm for locating sensors in irregular areas. In *IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–6. IEEE, 2010.

[123] M.I. Silventoinen and T. Rantalainen. Mobile station emergency locating in GSM. In *IEEE International Conference on Personal Wireless Communications*, pages 232–238, 1996.

[124] G. Simon, P. Volgyesi, M. Maróti, and A. Ledeczi. Simulation-based optimization of communication protocols for large-scale wireless sensor networks. In *IEEE Aerospace Conference*, volume 3, 2003.

[125] A.J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[126] S. Spieker and C. Rohrig. Localization of pallets in warehouses using wireless sensor networks. In *16th Mediterranean Conference on Control and Automation*, pages 1833–1838. IEEE, 2008.

[127] M.A. Spirito and A.G. Mattioli. On the hyperbolic positioning of GSM mobile stations. In *International Symposium on Signals, Systems, and Electronics (ISSSE)*, pages 173–177. IEEE, 1998.

[128] G.L. Sun and W. Guo. Bootstrapping m-estimators for reducing errors due to non-line-of-sight (NLOS) propagation. *IEEE Communications Letters*, 8(8):509–510, 2004.

[129] X. Tang and W.K. Stewart. Optical and sonar image classification: wavelet packet transform vs fourier transform. *Computer Vision and Image Understanding*, 79(1):25–46, 2000.

[130] B. Tribelhorn and Z. Dodds. Evaluating the roomba: A low-cost, ubiquitous platform for robotics research and education. In *International Conference on Robotics and Automation (ICRA)*, pages 1393–1399. IEEE, 2007.

[131] S. Venkatesh and R.M. Buehrer. A linear programming approach to NLOS error mitigation in sensor networks. In *Proceedings of the 5th International Conference on Information processing in Sensor Networks (IPSN)*, pages 301–308. ACM, 2006.

[132] S. Venkatesh and R.M. Buehrer. NLOS mitigation using linear programming in ultrawideband location-aware networks. *IEEE Transactions on Vehicular Technology*, 56(5):3182–3198, 2007.

[133] S. Venkatraman, J. Caffery Jr, and H.R. You. Location using LOS range estimation in NLOS environments. In *IEEE Vehicular Technology Conference*, volume 2, 2002.

[134] C. Wang and L. Xiao. Sensor localization in concave environments. *Transactions on Sensor Networks (TOSN)*, 4(1):1–31, 2008.

[135] C. Wang, L. Xiao, and J. Rong. Sensor localization in an obstructed environment. In *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2005.

[136] W. Wang, T. Jost, and U.C. Fiebeg. Characteristics of the NLOS Bias for an Outdoor-to-Indoor Scenario at 2.45 Ghz and 52 GHz. In *IEEE Antannae and Wireless Propagation Letters*, volume 10. IEEE, 2011.

[137] X. Wang, Z. Wang, and B. O'Dea. A TOA-based location algorithm reducing the errors due to non-line-of-sight (NLOS) propagation. *IEEE Transactions on Vehicular Technology*, 52(1):112–116, 2003.

[138] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, 1997.

[139] K. Whitehouse and D. Culler. A robustness analysis of multi-hop ranging-based localization approximations. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN)*. ACM, 2006.

[140] K. Whitehouse, C. Karlof, and D. Culler. A practical evaluation of radio signal strength for ranging-based localization. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(1):41–52, 2007.

[141] K. Whitehouse, C. Karlof, A. Woo, F. Jiang, and D. Culler. The effects of ranging noise on multihop localization: an empirical study. In *Proceedings of the 4th International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE Press, 2005.

[142] S. B. Williams, P. Newman, G. Dissanayake, and H. Durrant-Whyte. Autonomous underwater simultaneous localisation and map building. In *International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1793–1798. IEEE, 2000.

[143] M.P. Wylie and J. Holtzman. The non-line of sight problem in mobile location estimation. In *5th IEEE International Conference on Universal Personal Communications*, volume 2, pages 827–831, 1996.

[144] B. Xiao, L. Chen, Q. Xiao, and M. Li. Reliable anchor-based sensor localization in irregular areas. *IEEE Transactions on Mobile Computing (TMC)*, pages 60–72, 2010.

[145] L. Xiong. A selective model to suppress NLOS signals in angle-of-arrival (AOA) location estimation. In *IEEE 9th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, volume 1, pages 461–465, 1998.

[146] A.M. Zoubir and B. Boashash. The bootstrap and its application in signal processing. *IEEE Signal Processing Magazine*, 15(1):56–76, 1998.