SORBONNE UNIVERSITÉ

EURECOM
Sophia Antipolis

Learning from the sky: Design of autonomous radio-enabled unmanned aerial vehicles in mobile cellular networks

Dissertation

submitted to

Sorbonne Université

in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Author:

Omid ESRAFILIAN

Scheduled for defense on the 6th March of 2020 before a committee composed of:

Reviewers			
	Prof.	**Christian BETTSTETTER**	University of Klagenfurt , Austria
	Prof.	**Luiz DA SILVA**	Trinity College Dublin, Ireland
Examiners			
	Assistant Prof.	**Sofie POLLIN**	KU Leuven, Belgium
	Associate Prof.	**Mehdi BENNIS**	University of Oulu, Finland
	Associate Prof.	**Florian KALTENBERGER**	EURECOM, France
Thesis Advisor			
	Prof.	**David GESBERT**	EURECOM, France
Invited			
	Dr.	**David LOPEZ-PEREZ**	Nokia Bell Labs Distinguished Member of Technical Staff, Ireland

Apprentissage depuis le ciel: Design de drones autonomes equippés de lien radio pour les réseaux mobiles cellulaires

Thèse

soumise à

Sorbonne Université

pour l'obtention du Grade de Docteur

présentée par:

Omid ESRAFILIAN

Soutenance de thèse prévue le 06 Mars 2020 devant le jury composé de:

Rapporteur			
	Prof.	**Christian BETTSTETTER**	University of Klagenfurt , Austria
	Prof.	**Luiz DA SILVA**	Trinity College Dublin, Ireland
Examinateur			
	Assistant Prof.	**Sofie POLLIN**	KU Leuven, Belgium
	Associate Prof.	**Mehdi BENNIS**	University of Oulu, Finland
	Associate Prof.	**Florian KALTENBERGER**	EURECOM, France
Directeur de Thèse			
	Prof.	**David GESBERT**	EURECOM, France
Invité			
	Dr.	**David LOPEZ-PEREZ**	Nokia Bell Labs Distinguished Member of Technical Staff, Ireland

Abstract

The interactions between the areas of mobile networking and drone robotics are currently attracting significant attention from both the robotics and the telecommunications engineering communities. The key scenarios where such interactions manifest themselves include the so-called flying base station or flying radio access network (FRAN) on the one hand, and the drone as a flying terminal on the other hand.

The use of drones and unmanned aerial vehicles (UAV) as FRAN nodes is rapidly emerging as a powerful tool to complement traditional fixed terrestrial deployments. The advantage of using UAVs will be particularly felt in those use cases where being able to quickly deploy a network where and when it matters is critical. However, the success of so-called FRANs hinges on the ability of the UAVs to place themselves spatially in an efficient and autonomous manner.

Having this in mind, the first part of this thesis aims to investigate current works and technologies of UAV-assisted wireless communications and develops novel methods for both the placement and trajectory design of a UAV as a flying RAN in the wireless networks for both mobile broadband coverage scenarios and IoT data harvesting scenarios. We highlight how the exploitation of city 3D maps can bring about substantial benefits for the reliable self-placement of flying radios. A suite of methods are presented that lie at the cross-road between machine learning and traditional communication theoretic network design.

Regardless of the placement or trajectory design, all the algorithms operate on the basis of an array of side-information such as node GPS location, the 3D map of the city, and terrain-dependent propagation parameters allowing the prediction of radio signal strengths. While such data may be collected via the network beforehand allowing placement or the trajectory to be optimized before the actual UAV flight, part or all of the information may also have to be discovered or learned by the UAV. In this regard, a part of this thesis is devoted to discussing how to learn and estimate such information just from the UAV-borne measurements.

Assuming the availability of safe cellular connectivity beyond visual line of sight, UAVs are becoming appealing solutions for a wide range of applications in the areas of transportation, goods delivery, and system monitoring. All these use cases pertain to the UAV as an aerial terminal scenario, the availability of a reliable radio link is essential to make sure the drone can be guided effectively towards completion of its mission. The main challenge however in these domains is the design of trajectories which indeed can guarantee reliable and seamless cellular connectivity all along the path

while allowing the completion of the UAV mission which is today a lacking feature of existing technologies. Therefore, in the second part of this thesis, we propose a novel approach for optimal trajectory design between a pre-determined initial location and a given destination point by leveraging on a coverage map. The coverage map can be obtained with a combination of a 3D map of the environment and radio propagation models. We establish a graph theory-based framework to evaluate the feasibility of the problem and obtain a high-quality approximate solution to the optimal UAV trajectory design problem.

Lastly, we come back to the FRAN scenario and discuss the experimental verification of the placement algorithm of a UAV relay in LTE networks and we present the design of the Rebot (Relay Robot) prototype. The Rebot functions both as an outdoor LTE relay between ground users and a fixed base station, as well as an autonomous robot capable of positioning itself at a throughput maximizing location.

Abrégé

Les interactions entre les domaines des réseaux mobiles et de la robotique des drones attirent actuellement l'attention des communautés de la robotique et de l'ingénierie des télécommunications. Les principaux scénarios dans lesquels ces interactions se manifestent comprennent la station de base volante ou le réseau d'accès radio volant (FRAN), d'une part, et le drone en tant que terminal volant, d'autre part.

L'utilisation de drones et de véhicules aériens sans pilote (UAV) comme nœuds FRAN est en train de devenir rapidement un outil puissant pour compléter les déploiements terrestres fixes traditionnels. L'avantage de l'utilisation des drones se fera particulièrement sentir dans les cas où il est essentiel de pouvoir déployer rapidement un réseau là où et quand cela est nécessaire. Toutefois, le succès des FRAN dépend de la capacité des drones à se placer dans l'espace de manière efficace et autonome.

Dans cette optique, la première partie de cette thèse vise à étudier les travaux et les technologies actuelles des communications sans fil assistées par des drones et à développer de nouvelles méthodes pour le placement et la conception de la trajectoire d'un drone en tant que RAN volant dans les réseaux sans fil pour les scénarios de couverture mobile à large bande et de collecte de données de l'IdO. Nous soulignons comment l'exploitation des cartes 3D des villes peut apporter des avantages substantiels pour l'autopositionnement fiable des radios volantes. Une série de méthodes sont présentées, qui se situent à la croisée des chemins entre l'apprentissage machine et la conception théorique traditionnelle des réseaux de communication.

Quel que soit le placement ou la conception de la trajectoire, tous les algorithmes fonctionnent sur la base d'un ensemble d'informations complémentaires telles que la localisation GPS du nœud, la carte 3D de la ville et les paramètres de propagation dépendant du terrain, ce qui permet de prédire la puissance des signaux radio. Ces données peuvent être collectées au préalable via le réseau, ce qui permet d'optimiser le placement ou la trajectoire avant le vol du drone, Une partie ou la totalité des informations peut également devoir être découverte ou apprise par l'UAV. À cet égard, une partie de cette thèse est consacrée à la discussion sur la façon d'apprendre et d'estimer ces informations uniquement à partir des mesures effectuées par le drone.

En supposant la disponibilité d'une connectivité cellulaire sûre au-delà de la ligne de vue, les drones deviennent des solutions attrayantes pour un large éventail d'applications dans les domaines du transport, de la livraison de marchandises et de la surveillance des systèmes. Tous ces cas d'utilisation concernent le drone en tant que scénario de terminal aérien, la disponibilité d'une liaison radio fiable est essentielle pour s'assurer que le drone

peut être guidé efficacement vers l'accomplissement de sa mission. Toutefois, le principal défi dans ces domaines est la conception de trajectoires qui peuvent effectivement garantir une connectivité cellulaire fiable et sans faille tout au long du trajet tout en permettant l'achèvement de la mission du drone, qui est aujourd'hui une caractéristique manquante des technologies existantes. C'est pourquoi, dans la deuxième partie de cette thèse, nous proposons une nouvelle approche pour la conception de trajectoires optimales entre un emplacement initial prédéterminé et un point de destination donné en s'appuyant sur une carte de couverture. La carte de couverture peut être obtenue en combinant une carte 3D de l'environnement et des modèles de propagation radio. Nous établissons un cadre basé sur la théorie des graphes pour évaluer la faisabilité du problème et obtenir une solution approximative de haute qualité au problème de conception optimale de la trajectoire des drones.

Finalement, nous revenons sur le scénario FRAN et discutons de la vérification expérimentale de l'algorithme de placement d'un relais de drone dans les réseaux LTE et nous présentons la conception du prototype Rebot (Relay Robot). Le Rebot fonctionne à la fois comme un relais LTE extérieur entre les utilisateurs au sol et une station de base fixe, ainsi que comme un robot autonome capable de se positionner à un endroit maximisant le débit.

Acknowledgements

I would like to express my sincere appreciation to my supervisor, Professor David Gesbert, for the continues support of my PhD study and related research, for his patience, motivation, and immense knowledge. He convincingly guided and encouraged me to be professional and do the right thing even when the road got tough. Without his persistent help, the goal of this thesis would not have been realized.

My sincere thank also goes to my fellow lab-mate, Dr. Rajeev Gangula, for stimulating discussions, for all the moments we were working together before the deadlines, and for all the fun we have had in last years. During the years that I have spent at EURECOM, I had a chance to make amazing friends. I want to thank all of them for the fun moments that we had together.

Last but not the least, I wish to acknowledge the support and great love of my family: my parents and my brother and sister. They supported me spiritually throughout writing this thesis and this work would not have been possible without their input.

This work was supported by the ERC under the European Union Horizon 2020 research and innovation program (Agreement no. 670896).

Contents

Abstract i
Abrégé [Français] iii
Acknowledgements v
Contents vii
List of Figures x
List of Tables xv
Acronyms xvii
Notations xix

1 Introduction **1**
1.1 Background and Motivations 2
1.1.1 Role Played by Node Localization 4
1.2 Aims and Objectives 5
1.3 Research Methodology and Assumptions 7
1.4 Outline of the Thesis 8

2 System Model **13**
2.1 Introduction 13
2.2 Channel Models 13
2.2.1 Segmented Channel Model 14
2.2.2 Probabilistic Link Attenuation Models 15
2.3 Communication Performance Metric 16
2.3.1 SINR 16
2.3.2 Outage 17
2.3.3 UAV Energy Consumption 17
2.3.4 Communication Throughput 18
2.4 Trajectory Design in UAV-aided Wireless Networks 18
2.5 Cellular-connected UAV Trajectory Design 19

3 Map-based Placement and Trajectory Design in UAV-aided Wireless Networks **21**
3.1 Introduction 21

3.2 Optimal Trajectory Design for an Intelligent Data Harvesting 22
3.2.1 Communication System Model 23
3.2.2 Joint Scheduling and Trajectory Optimization 23
3.2.3 LoS Probability Model Using Map Compression 24
3.2.4 Proposed Solution for Communication Trajectory Optimization . . 25
3.2.5 Iterative Algorithm 28
3.2.6 Proof of Convergence 28
3.2.7 Trajectory Initializing 30
3.3 Optimal UAV Relay Placement in LTE Networks 31
3.3.1 Communication Model 31
3.3.2 UAV Placement Optimization 32
3.4 Numerical Results 33
3.5 Conclusion 37

4 Active Learning for Channel Estimation: Map-based approaches 39
4.1 Introduction 39
4.2 UAV Kinematic Model 39
4.3 Learning Trajectory Design 40
4.3.1 Measurement Collection and Channel Learning 40
4.3.2 Optimization Problem 42
4.3.3 Dynamic Programming 43
4.4 Numerical Results 44
4.5 Conclusion 46

5 UAV-aided Radio Node Localization 47
5.1 System Model 49
5.2 User Localization and Channel Model Learning 50
5.2.1 PSO Techniques 51
5.2.2 Single User Case 52
5.2.3 Multi User Case 54
5.3 Trajectory Design for Accelerated Learning 54
5.3.1 Fisher Information Matrix 55
5.3.2 Cramér-Rao Bound Analysis 55
5.3.3 Trajectory Optimization 56
5.3.4 Greedy Trajectory Design 57
5.4 Numerical Results 59
5.5 Conclusion 61

6 3D City Map Reconstruction from Radio Measurements 65
6.1 Introduction . . . 65
6.2 System Model . . . 67
6.3 LoS vs. NLoS Classification . . . 67
6.3.1 Target User Clustering . . . 67
6.3.2 Optimization of Target User Group . . . 68
6.3.3 Radio Propagation Parameter Learning . . . 68
6.3.4 User Classification . . . 69
6.4 3D City Map Reconstruction . . . 69
6.4.1 Optimum UAV Altitude . . . 71
6.5 Numerical Results . . . 73
6.6 Conclusion . . . 75

7 UAV Trajectory Design Under Cellular Connectivity Constraints 79
7.1 Introduction . . . 79
7.2 System Model . . . 80
7.2.1 Communication Model . . . 81
7.2.2 Problem Formulation . . . 81
7.3 Feasibility Check . . . 84
7.4 Trajectory Optimization . . . 86
7.5 Numerical Results . . . 87
7.6 Conclusion . . . 88

8 Experimental Studies 91
8.1 Introduction . . . 91
8.2 System Design . . . 92
8.2.1 UAV Design . . . 92
8.2.2 OAI eNBs . . . 94
8.2.3 Autonomous Placement . . . 94
8.3 UAV Placement . . . 95
8.3.1 Channel Parameter Estimation . . . 96
8.3.2 Placement Algorithm . . . 96
8.4 Experimental Results . . . 97
8.5 Conclusion and Discussion . . . 101
8.5.1 Design Improvement . . . 101
8.5.2 Channel Models . . . 101

9 Conclusion 103

Appendices 105

A Chapter 3 Appendices **107**
A.1 The derivation of the average channel gain 107
A.2 Proof of Lemma 3.2.1 107
A.3 Proof of Proposition 3.2.1 109

B Chapter 5 Appendices **111**
B.1 Proof of convergence for multi-user localization 111
B.2 Derivation of FIM 112

C The estimate of the map reconstruction error **115**

D Chapter 7 Appendices **117**
D.1 Proof of Proposition 7.2.1 117
D.2 Proof of Proposition 7.3.1 117
D.3 Proof of Lemma 7.4.1 118

Résumé [Français] **118**

List of Figures

1.1 Illustration of some promising FRANs use cases: Drone as a relay in LTE networks, and drone as a base station for IoT data harvesting. 3
1.2 Different self-placement problems in the context of UAV communications. 4
1.3 The overall taxonomy of tackled problems in this thesis. 9

2.1 Two-segment modeling for Air-to-ground channels in a given city. 15

3.1 Global LoS probability compared with the local LoS probability learned form the 3D map for three ground nodes. 34
3.2 Optimal drone trajectory and ground node scheduling for different flight times. As the flight time increases, the UAV gets closer to individual ground nodes. 34
3.3 (a) Throughput performance versus iteration and (b) drone altitude evolution versus iteration. 35
3.4 Performance of the map-based algorithm in comparison with the probabilistic approach and the deterministic Algorithm for 6 ground nodes versus increasing the flight time. 36
3.5 Performance comparison of the map-based Algorithm and the probabilistic approach versus the average building height for a fixed flight time. 37
3.6 Top view of the city and the optimal position of the UAV for 3 static ground nodes using the Map-based placement. The optimal altitude is 270 meters. 38

4.1 A fragment of the 3D path graph, arbitrary base position $\mathbf{v}_{\mathrm{I}}$, and terminal position $\mathbf{v}_{\mathrm{F}}$ 43
4.2 (a) Top view of the optimal learning trajectory using proposed algorithm. (b) The UAV elevation along the trajectory. 45
4.3 Comparison of the MSE for different learning trajectories. 46

5.1 An example of the possible actions in the greedy trajectory design at the n-th time step. To find the best position for the drone in the next time step all the eight drone's adjacent positions at time step n need to be evaluated. 59

5.2 (a) RMSE of user localization. (b) CDF of user localization for a fixed trajectory length. 62
(a) 62
(b) 62
5.3 User localization error by increasing the number of users for the map-based algorithm while the UAV follows arbitrary trajectories with a fixed length of 900 meters. 63
5.4 (a) Top view of the generated trajectory using the sub-optimal approach. (b) Drone altitude along the trajectory. 64
(a) 64
(b) 64

6.1 Top view of the city building map and outdoor ground user locations which are shown by circles. 66
6.2 Building map reconstruction: Example with 3 RSSI measurements. The first and third users were successfully labeled as LoS, while the second was correctly labeled NLoS. As a result any building between the drone and user 1 (resp. user 3) must lie below line 1 (resp. below line 3). 70
6.3 Determining the best radius of the ring-like area for K-means algorithm using proposed criterion. Solid line denotes the classification error for each radius and dashed line is the corresponding criterion, predicting the same minimum. 73
6.4 K-means result for measured RSSI for users in target area and estimated RSSI for them using MLE. 74
6.5 Reconstruction normalised mean square error (NMSE) for both smoothed and rough map versus different UAV elevations. 74
6.6 (a) Side view of the 3D city map. The city map is discretized over small grid squares of length of 5 meters. (b) Reconstructed map after refinement and smoothing with NMSE equals to 0.35. 76
6.7 Map reconstruction error versus increasing the number of users. 77

7.1 Coverage area of a given BS and the sectors. 82
7.2 Top view of the city, the BSs positions, coverage area of each BS, and the common area. The UAV flies at 50m and the BSs are on the ground level. 84
7.3 Top view of the city, BS locations, the generated trajectories and its lengths for different algorithms. The coverage area of each BS is highlighted with green color. 88
7.4 Outage versus the trajectory length for different algorithms. 89

8.1 UAV-based LTE relaying system. 93
8.2 Custom-built UAV. 94
8.3 UAV placement algorithm. 95
8.4 Outdoor experiment setup with only fixed eNB. 97
8.5 Outdoor experiment setup with UAV relay. 98
8.6 Channel measurements for LoS scenario. 99

8.7 Channel measurements for NLoS scenario. 100
8.8 Instance of throughput comparison at a given user location. 100

C.1 One dimensional simplified deterministic city model. 115

9.1 Illustration de quelques cas d'utilisation de RAN volants prometteurs: Drone-en-tant-que-relais pour la connectivité des utilisateurs au sol (DaaR), Drone-en-tant-que-station-de-base pour la collecte de données IoT (DaaB). 120
9.2 Exemple de conception de trajectoire pour la collecte de données IoT (3 capteurs Cas). 123
9.3 Performances de l'algorithme basé sur la carte pour les appris et les vrais paramètres de canal en comparaison avec d'autres algorithmes pour 6 nuds au sol par rapport à l'augmentation du temps de vol. 124
9.4 (a) Vue de dessus de la trajectoire d'apprentissage optimale en utilisant l'algorithme proposé. (b) L'élévation de l'UAV le long de la trajectoire. . 125
9.5 Comparaison de l'erreur quadratique moyenne (MSE) pour différentes trajectoires d'apprentissage. 126
9.6 (a) Vue de dessus de la trajectoire générée en utilisant l'approche sous-optimale. (b) Altitude du drone le long de la trajectoire. 128
(a) 128
(b) 128
9.7 Comparaison du MSE pour différentes trajectoires d'apprentissage pour le cas de l'utilisateur unique. 129
9.8 (a) Vue latérale du plan de ville 3D. (b) Carte reconstruite avec NMSE égale à 0,35. 131
9.9 Vue de dessus de la ville, des emplacements BS, des trajectoires générées et de ses longueurs pour différents algorithmes. La zone de couverture de chaque BS est surlignée en vert. 133
9.10 coupure en fonction de la longueur de la trajectoire pour différents algorithmes. 134
9.11 coupure en fonction de la longueur de la trajectoire pour différents algorithmes. 135

List of Tables

1.1 The left most columns indicates the optimization and learning problems pursued in this thesis. The required information and key assumptions for each algorithm is clarified in the other columns. 8

8.1 Pathloss parameters. 99

Acronyms and Abbreviations

The acronyms and abbreviations used throughout the manuscript are specified in the following. They are presented here in their singular form, and their plural forms are constructed by adding and s, e.g. BS (base stations) and BSc (Base stations). The meaning of an acronym is also indicated the first time that it is used.

UAV	Unmanned Aerial Vehicle
RAN	Radio Access Network
FRAN	Flying Radio Access Network
BS	Base Station.
GPS	Global Positioning System
LTE	Long-Term Evolution
BVLoS	Beyond Visual Line of Sight
FAA	Federal Aviation Administration
HAP	High Altitude Platforms
LAP	Low Altitude Platforms
DaaT	Drone-as-a Terminal
IoT	Internet-of-Things
QoS	Quality of Service
RSSI	Received Signal Strength Indicator
aka	also known as
MIMO	Multiple Input Multiple Output
ToA	Time of Arrival
TDoA	Time Difference of Arrival
AoA	Angle of Arrival
MANAL	Mobile Anchor Node Assisted Localization
LoS	Line of Sight
NLoS	Non Line of Sight
SINR	Signal-to-Interference-plus-Noise Ratio
SNR	Signal-to-Noise Ratio
DP	Dynamic Programming
MSE	Mean Squared Error
SSE	Sum Squared Error
RMSE	Root Mean Square Error
NMSE	Normalised Mean Square Error

EM	Expectation Maximization
PSO	Particle Swarm Optimization
FIM	Fisher Information Matrix
CRB	Cramer-Rao Bound
CDF	Cumulative Distribution Function
PDF	Probability Density Function
TDM	Time-Division Multiplexing
TDMA	Time-Division Multiple Access
RBF	Radial Basis Function
SVM	Support Vector Machine
OAI	OpenAirInterface
PSD	positive semi-definite
PD	positive definite
LS	Least Squares
i.i.d	independent and identically distributed
MLE	Maximum Likelihood Estimation
RSRP	Reference Signal Received Power

Notations

The next list describes an overview on the notation used throughout this manuscript. We use boldface uppercase letters ($\mathbf{A}$) for matrices, boldface lowercase letters for vectors ($\mathbf{a}$), and regular letters for scalars (a or A). Sets are represented by calligraphic uppercase letters ($\mathcal{A}$).

$\lvert a \rvert$	Absolute value of the variable a
$\lVert \mathbf{a} \rVert$	Euclidian norm of the vector $\mathbf{a}$
$\lvert \mathcal{A} \rvert$	Cardinality of the set $\mathcal{A}$
$(\mathbf{a}, \mathbf{b})$	The edge connecting two vectors $\mathbf{a}, \mathbf{b}$
$\mathbf{A}^{\mathrm{T}}$	Transpose of the matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	Inverse of the matrix $\mathbf{A}$
$\mathbf{A}^{\dagger}$	Adjugate of the matrix $\mathbf{A}$
$\mathrm{tr}(\mathbf{A})$	Trace of the matrix $\mathbf{A}$
$\det(\mathbf{A})$	Determinant of the matrix $\mathbf{A}$
$\mathrm{E}[.]$	Expectation
$\Pr(x)$	The probability of an event x
$[m, n]$	The set of integers from m to n, $m < n$
$\lfloor . \rfloor$	Floor function
$\triangleq$	Used for definition
$a := b$	Used for assigning b to a
$\nabla^2 h(.)$	Hessian matrix of the function $h(.)$
$\mathcal{N}(0, \sigma^2)$	Zero-mean Gaussian distribution with variance σ^2
$\exp(.)$	Exponential function
$\log_2(a)$	Logarithm base 2 of the positive number a
$\log_{10}(a)$	Logarithm base 10 of the positive number a
$Cov\{\mathbf{a}\}$	Covariance matrix of the random vector $\mathbf{a}$
$\min(a, b)$	Returns the minimum value between two scalars a and b

Chapter 1

Introduction

Unmanned aerial vehicles (UAVs), also known as drones, have gained significant attention over the past few years, owing to their autonomy, flexibility, and a broad range of application domains. Historically, UAVs were largely used in military applications for remote surveillance and armed attacks, to reduce the casualties and the pilot losses. The latest advances in UAVs manufacturing and their reducing cost have made possible the various use of UAVs in civilian and commercial applications. Indeed, UAVs have been considered as enablers of numerous applications that include military, surveillance and monitoring, telecommunications, traffic control, rescue operations, package delivery, etc. During the past years the U.S. federal aviation administration (FAA) released the operational rules for regular civilian use of micro aerial vehicles and they have employed the UAVs for different purpose incorporating the beyond visual line of sight (BVLoS) flights, patrolling, night time surveillance, package delivery, etc. [1; 2]. The UAV industry growth is potentially enormous and with increasing efforts from governments facilitating regulatory framework [3; 4], UAV market is projected to reach $63.6 billion by 2025 [5], and also the further growth of global UAV industry is encouraged in the coming new years. Overall, the UAVs have emerged as a promising technology that can provide new jobs and prolific business opportunities.

Depending on the applications, the UAVs can be put into different categories. There are many types of UAVs in accordance with the different criteria such as flying time, payload, endurance, flying altitude, etc. In general, UAVs can be categorized based on the flying altitude into high altitude platforms (HAPs) and low altitude platforms (LAPs). HAPs fly in an altitude range above 17 Km, while LAPs can fly at lower altitudes in a range of tens of meters up to a few kilometers. In general, LAPs are faster and more maneuverable than the HAPs [6; 7].

In terms of the wing configuration, fixed-wing and rotary-wing UAVs are two main types of UAVs which are largely used. Fixed-wings UAVs are capable of higher maximum flying speed and can carry heavier payloads for longer distances in comparison with the rotary-wing UAVs and they need to move forward in order to remain aloft. On the contrary, rotary-wing UAVs, such as multi-rotors, are able to take off and land vertically and can hover over a given area [8]. For this, selecting an appropriate type of the UAVs to efficiently accomplish their missions is crucial.

A critical aspect of the UAV technology is the reliability and performance of wireless communications to/from the UAV. The UAVs need to communicate and exchange the safety essential information with different parties such as the nearby aerial vehicles, remote pilot, etc. to ensure a safe and reliable flight operation. This type of communication is known as control and non-payload communication. Depending on the mission of the UAV, it may need to transmit and receive high-quality areal images, real-time videos, and relaying data packets between the ground nodes and BSs or the terrestrial infrastructures. This type of communication is considered as payload communications.

To satisfy both payload and non-payload types of communications for the UAVs viewed as *aerial users or terminals*, suitable communication equipment and technologies are required for a seamless and a high throughput connectivity which realizes a reliable wireless communications for both air-to-air (i.e. between the UAVs during their missions) and air-to-ground (i.e. between the UAVs and the ground BSs or the ground nodes) links. To this end, different communication technologies can be employed incorporating communication through direct links, satellites, cellular, and ad-hoc networks. To ensure a seamless and reliable connectivity a hybrid network can used in which each UAV is equipped with more than one of the aforementioned technologies.

On the other hand, latest advances in wireless communication and robotics in building the miniature communication equipment have made it possible to mount compact and lightweight BSs and relays on UAVs which can be used to assist the terrestrial wireless communications to improve the quality of the service provided to the ground users and increase the overall performance of the network. This sort of exploitation of the UAVs to help the network, as FRANs or flying BSs, is termed *UAV-aided wireless communications* in this thesis.

In this study, we discuss the different aspects of integrating UAVs into wireless communications. In general, we split our discussion into two categories, namely UAVs as an aerial user, and UAV-aided wireless communications. In the following section, we further elaborate on the different applications of the UAVs in wireless communications, motivations, and challenges. In particular challenges related to the autonomous placement/trajectory of such UAVs in mobile communication network dependent scenarios.

1.1 Background and Motivations

The exploitation of drones/UAVs within future wireless cellular communication networks has recently gained significant attention. Several scenarios have been articulated in the literature. In this study, we refer to these possible scenarios as flying base station or flying radio access network (FRAN), and the Drone-as-a Terminal (DaaT).

The DaaT scenario is concerned with the exploitation and adaptation of wireless networks to serve as a carrier mechanism for UAV control commands and possible video feeds. It also includes designing a trajectory for the drones such that it always stays connected to the terrestrial infrastructures while flying, which has use cases range from transportation, surveillance, to goods delivery where the seamless connectivity matters [9; 10]. Generally speaking, the exploitation of the UAVs as aerial terminals is termed as the *cellular-connected UAVs* [11; 12]. The problem of designing a mission trajectory

which saves flying energy while guaranteeing reliable connectivity to the local cellular network is addressed in part of this thesis (Chapter 7).

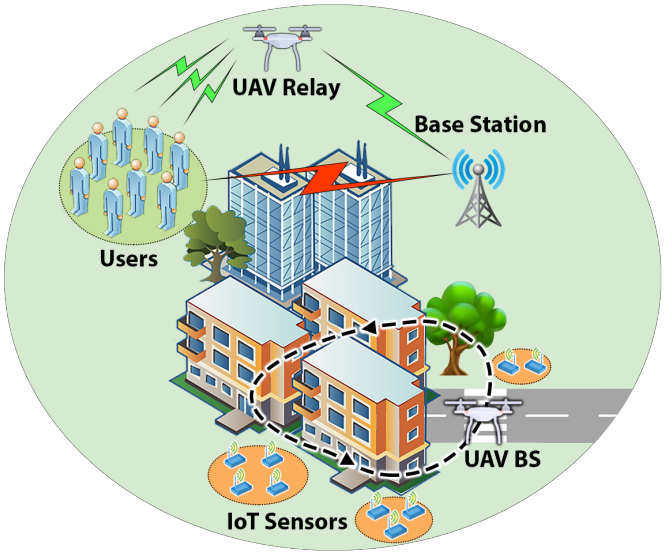


Figure 1.1 – Illustration of some promising FRANs use cases: Drone as a relay in LTE networks, and drone as a base station for IoT data harvesting.

In contrast, the FRAN or flying BS frameworks view the UAV as an integral piece of the radio access network (RAN) infrastructure, as shown in Fig. 1.1. The UAV acts as a flying base station (BS) which can, for example, harvest data sent from ground nodes or sensors, or as a flying relay towards a ground fixed BS in case the application's latency requirements necessitate the data to be transmitted in real-time to the core network (e.g. internet access by ground users). In general, this sorts of applications of the UAVs to aid the wireless networks is termed as the *UAV-assisted wireless communications*. The most promising feature of FRANs is to allow a flexible deployment of radio resources when and where they are most needed. Use cases range from disaster recovery scenarios, servicing of temporary cultural/sporting events, road traffic assistance, hot-spots coverage, and (Internet-of-Things) IoT data harvesting (smart city, agriculture, etc.) [6; 13]. While research challenges dealing with the practical evaluation and realization of UAV-aided system gains abound, the problem of determining how to best (self-)position the flying radios remains a critical and fascinating issue [14] which forms the core of this thesis.

Generally speaking, the placement problem and the path planning, respectively, refers to determining a location and a path for a UAV-mounted radio so as to best serve the data needs by one or more ground nodes. The placement and the trajectory design algorithms must be adaptive to context parameters, such as the user locations, the traffic distribution, the per-node quality of service (QoS) requirements, and the propagation conditions. Ideally, the algorithms operate in an autonomous fashion (i.e. replacing the human pilot) either onboard the drone or in a ground-based computing unit that is connected to the drone.

From an algorithmic perspective, it is important to distinguish between the (i) static placement problem from (ii) path planning. Static placement involves finding a single 3D location for the UAV, from where it will provide connectivity to not-too-distant ground nodes. While the solution may be updated when large-scale system parameters vary such as traffic or ground user location distributions, the UAV location is otherwise stable and can benefit from energy-saving mechanisms, such as onboard inflation-based devices [15] or the ability to exploit nearby resting spots [16]. In many scenarios, however, there is interest in flying along a path pattern cycle that brings the UAV closer in turn to each ground node, as this can be shown to significantly improve average throughput over a static deployment [17]. Note that apart from link throughput considerations, advanced path optimization methods may also take into account specific kinematics energy consumption models, obstacle avoidance mechanisms, as well as realistic robot dynamics (maximum speed and acceleration/deceleration) leading to a potentially complex mixed communication-robotics optimization framework. While this complex problem has been addressed in some of the literature [17; 18], we are not emphasizing such aspect in this thesis and focus on the sole interaction between the trajectory design and the communication models, while simplifying the mechanical robotics aspects. In Fig. 1.2, Different self-placement problems (including the static placement and the trajectory design) regarding the UAV communications, which are also addressed in this thesis, are listed.

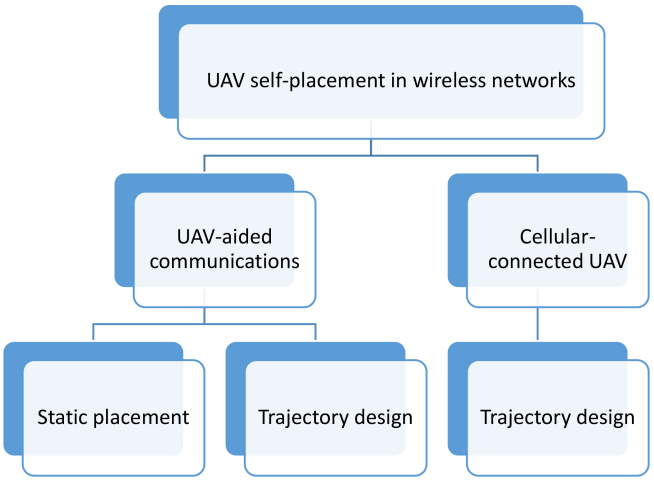


Figure 1.2 – Different self-placement problems in the context of UAV communications.

1.1.1 Role Played by Node Localization

In the context of the FRAN, where the UAV has a communication service mission to complete with respect to ground users/nodes, a fundamental piece of side-information lies in the knowledge (or not) of the ground node locations. The immediate solution to localize a node is to use the Global Positioning System (GPS), while it is readily available to localize the UAV itself, may not be viable or functioning to localize the ground nodes

at all times. This is because low-cost node may simply not be equipped with a GPS module or due to the fact that the functionality of the GPS is limited to environments where the nodes can maintain line of sight to satellites. For instance, the GPS sometimes fails in dense urban cities, in the indoor scenarios, and in the harsh environments due to the presence of the blocking objects (i.e. tall buildings) and the signal degradation. For this, network-based radio localization is a research area which has attracted increasing interest and different approaches have been developed during the past years to decrease the dependency of solutions on the sole GPS approach. In this thesis we are interested in investigating how UAV can play a new positive role into this matter.

In a typical network, there are two types of nodes, namely anchors and users. Anchors have perfectly known its positions, which can be stationary or mobile nodes like flying BSs or relays, whereas users often have unknown locations (and need to be estimated). Network's nodes localization can follow a two step approach (i) distance measurement between the nodes in the network and (ii) the nodes localization based on the inter-nodes distances. Based on the distance measurement techniques employed, localization algorithms can be categorized into range-based localization algorithms and range-free localization approaches. In Chapter 5, different aspects of node localization in wireless networks are discussed in greater detail.

1.2 Aims and Objectives

This study aims to investigate placement and trajectory design problems for both the UAV-aided communication network scenario (aka flying RAN, flying BS, flying relay) and the cellular-aided UAV scenario (aka flying terminal problem).

The premium offered by the FRAN over its conventional fixed counterpart essentially lies in the ability to bring the RAN closer to the user or generally to increase link quality. As a result, the role played by statistical channel models in the placement and trajectory design solutions is critical. The assumption of line-of-sight (LoS) channels or the use of simple statistical blocking models (i.e. modeling the LoS probability) has proved an excellent way to derive early insights into the problem and to allow for closed-form average performance analysis [7]. Unfortunately, the simplistic or probabilistic nature of such approaches limits our ability to guarantee performance in an actual on-field UAV deployment. For example, a statistically optimized placement algorithm might suggest a UAV location that one eventually discovers to be severely affected by a local blockage in practice (e.g. unforeseen presence of a tall building) forcing the robot to some sub-optimal path recomputing.

Hence the first chapters of this study are devoted to developing methods allowing to go around the simplistic LoS channel model issue for the sake of in-field UAV deployments. The thesis overall raises the reader's attention onto the role played by the exploitation of suitable maps. By maps, we here refer to a geographically indexed data set which can be used to better predict the actual channel conditions for any specific pair of UAV and ground node locations. Several types of inter-related maps can be considered, including throughput maps, radio (link strength) maps, and physical 3D maps. In this dissertation, we specifically consider the exploitation of the 3D map. In Chapter 3, we give an overview

of the UAV placement approaches for the drone relay and drone BS settings and then we introduce map-based UAV placement and trajectory design methods. The channel models, possibly derived from 3D maps, are first described. While 3D maps are rich in information, they are also difficult to exploit directly in a placement and trajectory optimization problem because of the highly irregular behavior of radio blockage, especially in urban environments. As a key contribution of this thesis, we derive placement methods (both static and path-based) which build on map data while going around the apparent quasi non-differentiable nature of the problem.

An essential component of the aforementioned scenarios and applications is the knowledge of the user locations and of the model characteristics of the radio channels, which usually are not available upon UAV take-off. For this, we also investigate the problem of exploiting the UAV flight so as to learn the radio channels and localizing the ground nodes from the radio measurements themselves. In general, the performance of any learning process highly depends on the training data set, which in this case is collected radio measurements from the ground nodes. To this end, we further devise an optimized trajectory for the UAV by capitalizing on the 3D map to intelligently collect measurements from the ground nodes in order to improve the performance of learning and the localization.

As another important side-information, several of the proposed algorithms operate on the basis of a city 3D map where the UAV is deployed. However, this information also may not be available and we highlight the potential of again exploiting the UAV mission flight so as to estimate the 3D buildings map from UAV-borne measurements. Hence, flying a drone merely for constructing a map would be costly, we propose an algorithm to construct the 3D map of the city (as a by-product) from only the radio measurements while the drone is accomplishing its communication service (or other) mission and flying over the network.

Turning to the context of the flying radio terminal, a tremendous increase in the use of the UAVs for the different applications such as transportation, goods delivery, system monitoring, etc. has been observed during the past few years. The enabling of such applications requires safe and seamless connectivity between the fixed infrastructure in the ground and the UAVs. As mentioned before, a key challenge in this area lies in the design of trajectories which, while allowing the completion of the UAV mission, can guarantee reliable cellular connectivity all along the path. Akin to the problem of trajectory design for the flying BS case, previous approaches in the flying terminal domain have also considered simplistic propagation model assumptions (e.g. LoS based) or more advanced models but with computationally demanding optimized solutions [9; 19; 10; 20]. In this study, we also propose a novel approach for trajectory design between a starting location and a given destination such that getting a reliable QoS from the cellular network all along the path.

Finally, we discuss practical prototype realizations carried as part of this PhD work in our drone lab. In particular, we designed a flying radio prototype, so-called Rebot (Relay Robot), which comprises UAVs equipped with long-term evolution (LTE) base station which can provide an end-to-end LTE connectivity between ground users and the core network. It also functions as an autonomous robot capable of positioning itself at a

throughput maximizing location. Up to our knowledge, this was among the very first such prototypes worldwide of this kind.

1.3 Research Methodology and Assumptions

In order to solve the UAV placement problems in UAV-aided wireless communication, various approaches are here studied. In this thesis, we specifically focus on how to efficiently exploit the knowledge of a 3D map of the environment to guarantee the performance predicted by the proposed algorithms.

First, we study the problem of UAV placement and the communication trajectory design to maximize the performance of the network in terms of the worst user throughout. The critical components for the UAV placement and the trajectory design are the radio channel parameters of the environment where the UAV is deployed and the location awareness of the nodes in the network. During the communication phase, we assume that these pieces of information are available, while they often have to be estimated from the UAV-borne measurements.

To deal with this problem, we also formulate the problem of learning the channel parameters from a set of measurements which has been collected by the UAV while following an arbitrary trajectory over the network. To minimize the error of the estimated channel parameters we then optimize the UAV learning trajectory for collecting measurements. Note that, we are treating the learning and the communication phases that are separated in time. This allows us to obtain optimal trajectories for both phases independently, i.e., if one is only interested in learning or communication scenario this solution serves the purpose.

To learn the channel parameters we assume that the locations of the ground nodes are known while may not be true in most of the cases. For this, in a separate chapter of this thesis, we study the problem of node localization in wireless networks by capitalizing on the city 3D map information. Similarly, a node in the network can be localized from UAV-borne radio measurement. Note that, to localize a node the radio channel parameters need to be known, hence the UAV requires not only to localize the users but also to learn the channel at the same time. Similar to channel learning, we first address the node localization from a given set of measurements when the UAV follows an arbitrary trajectory and then we optimize the UAV trajectory for the further improvement in the localization accuracy.

All the aforementioned algorithms build on the availability of the city 3D map while in some cases the map information may not available or costly to access. To cope with this problem, we propose a method to construct a 3D model of a city just by exploiting the radio measurements collected by the UAV from the outdoor ground nodes, whose locations are known, in the city.

We finally investigate the problem of cellular-enabled UAV communication. A novel approach will be proposed to design the shortest trajectory for a UAV to guarantee a seamless connection to the infrastructure on the grounds (i.e. ground BS) all along the UAV path while allowing the completion of the UAV mission. To this end, the trajectory design algorithm leveraging a coverage map that can be obtained with a combination of

3D map of the environment, radio propagation models, and the locations of the ground BSs.

In table 1.1, we summarized all the key assumptions and information which are required for each algorithm proposed in this thesis.

Information / Algorithm	3D map	Nodes locations	Channel parameters
Placement and trajectory design	✓	✓	✓
Channel learning	✓	✓	×
Node localization	✓	×	×
City map reconstruction	×	✓	×
Cellular-enabled UAV trajectory design	✓	✓	✓

Table 1.1 – The left most columns indicates the optimization and learning problems pursued in this thesis. The required information and key assumptions for each algorithm is clarified in the other columns.

1.4 Outline of the Thesis

We partition our discussion into two main applications of UAV-assisted wireless networks, and cellular-connected UAVs, where the UAV is viewed as an integral piece of the RAN infrastructure or as an aerial user terminal, respectively. Note that, the majority of the thesis contributions are obtained in the first scenario, i.e UAV assisted wireless networks. Moreover, we investigate the problem of network localization using radio measurements. Specifically, we highlight how the exploitation of the 3D map which can bring about substantial benefits for the reliable placement and trajectory design of flying radios, and a precise node localization.

An outline of the dissertation along with a brief summary of the contributions of each chapter is provided below. The overall taxonomy of tackled problems is highlighted in Fig. 1.3.

Chapter 2 - System Models

In this chapter, we aim to discuss the essential concepts pertained to UAV communication which are useful for research in different frameworks of UAV communications platforms. We specifically talk about the channel model, performance metrics, mathematical formulation for performance optimization via the placement and devising a trajectory for the UAV, and the different aspects of the UAV-aided node localization in wireless networks.

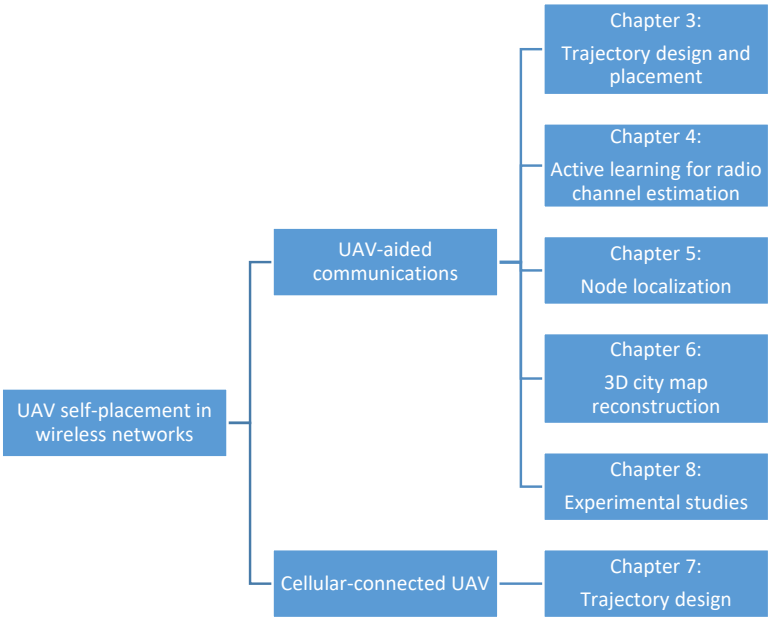


Figure 1.3 – The overall taxonomy of tackled problems in this thesis.

Chapter 3 - Map-based Placement and Trajectory Design in UAV-aided Wireless Networks

Several new and fascinating issues arise from the study of flying BSs in a wireless network. These can be broadly categorized into placement and trajectory design problems. While placement problem deals with finding flexible yet static locations of the UAV BSs, trajectory design involves finding UAV trajectories. When it comes to placement or trajectory design problems, most existing solutions rely on simplified channel attenuation models which are based on either (deterministically guaranteed) LoS links, or predictive models for the probability of occurrence of a LoS link. Most of the previous approaches lack actual performance guarantees for either placement or trajectory design algorithms when used in a real-life navigation scenario. In order to circumvent this problem, we propose a solution by embedding of actual 3D city map data in the UAV placement and the trajectory design via a *map compression* method which lets us resort the problem to convex optimization arguments.

Part of the work in this chapter has resulted in the following publications:

- *Esrafilian, Omid, Rajeev Gangula, and David Gesbert. "UAV-relay Placement with Unknown User Locations and Channel Parameters." In 2018 52nd Asilomar Conference on Signals, Systems, and Computers.*
- *Esrafilian, Omid, Rajeev Gangula, and David Gesbert. "Learning to communicate in UAV-aided wireless networks: Map-based approaches." IEEE Internet of Things Journal 6, no. 2 (2018): 1791-1802.*

Chapter 4 - Active Learning for Channel Estimation: Map-based approaches

One of the common assumptions in previous works regarding UAV communications is that the radio channel model parameters are assumed to be known for the optimal UAV placement or the UAV trajectory design. However, in reality, these parameters need to be learned based on the measurements collected from the ground users. In this chapter, we only focus on learning the radio channel parameters. We formulate and solve a learning trajectory optimization problem in order to minimize the estimation error of the channel model parameters. The devised trajectory allows the UAV to exploit the map and quickly learn the propagation parameters within a limited flying time.

The results presented in this chapter have been published in the following publication

- *Esrafilian, Omid, Rajeev Gangula, and David Gesbert. "Learning to communicate in UAV-aided wireless networks: Map-based approaches." IEEE Internet of Things Journal 6, no. 2 (2018): 1791-1802.*

Chapter 5 - UAV-aided Radio Node Localization

Regardless of the placement or trajectory design for the UAV, the algorithms usually operate on the basis of an array of information including the ground nodes location information. Using the GPS to obtain the location of the nodes is not a viable solution all the time due to the GPS signals degradation or blockage effect of the obstacles in the environment. To address this problem in this chapter, we formulate the problem of user localization by processing the received signal strength indicator (RSSI) measurements which are collected by a UAV while capitalizing on 3D map data of the city. Since the localization highly depends on the knowledge of the radio channels between the users and the UAV, thus the UAV needs not only to localize the users but also to learn the channels at the same time. Moreover, we formulate a novel resource-constrained UAV trajectory optimization problem for the further improvement of the localization accuracy.

Parts of results presented in this chapter have been published in the following publications

- *Esrafilian, Omid, Rajeev Gangula, and David Gesbert. "UAV-relay Placement with Unknown User Locations and Channel Parameters." In 2018 52nd Asilomar Conference on Signals, Systems, and Computers.*

And the following publication will be submitted soon:

- *Esrafilian, Omid, Rajeev Gangula, and David Gesbert. "Simultaneous Learning and Path Planning (SLAP) in UAV-Aided Wireless Communications", To be submitted.*

Chapter 6 - 3D City Map Reconstruction from UAV-based Radio Measurements

The essential assumption in the previous chapters is the awareness of the 3D map of the areas where the UAV is deployed. Traditionally, 3D city map reconstruction uses photogrammetry techniques which need high-resolution stereo images and extensive processing capability. Since the UAV can fly over the city and make measurements from

the ground users, in this chapter we aim to propose a novel algorithm to reconstruct 3D city maps by hinging on the radio measurements. The proposed approach relies on the unique ability for a UAV-to-ground communication system to detect and classify LoS vs. non-line-of-sight (NLoS) channels towards ground users using machine learning tools.

The work presented in this chapter was resulted in the following publication:

- *Esrafilian, Omid, and David Gesbert. "3D city map reconstruction from UAV-based radio measurements." In 2017 IEEE Global Communications Conference (GLOBECOM).*

Chapter 7 - UAV Trajectory Design Under Cellular Connectivity Constraints

In this chapter, we investigated the problem of UAV trajectory design under cellular connectivity constraints. A key challenge in this problem lies in the design of trajectories between pre-determined starting and destination locations which, while allowing the completion of the UAV mission, can guarantee reliable cellular connectivity all along the path. We proposed a novel approach that strikes a trade-off between performance (i.e. path length reduction) and complexity by exploiting the 3D map of the environment and employing the graph theory. We established a graph theory-based framework to first evaluated the feasibility of the problem and then to obtain a high-quality approximate solution to the UAV trajectory design problem.

The results presented in this chapter was submitted for publication:

- *Esrafilian, Omid, Rajeev Gangula, and David Gesbert. "3D-Map Assisted UAV Trajectory Design Under Cellular Connectivity Constraints." 2020 IEEE international conference on communications (ICC).*

Chapter 8 - Experimental Studies

In this chapter we evaluate the performance of the proposed placement algorithm for a UAV relay in LTE networks through the experimental verification, and we will also discuss practical prototype realizations.

The results presented in this chapter was submitted for publication:

- *Gangula, Rajeev, Esrafilian, Omid, et al. "Flying rebots: First results on an autonomous UAV-based LTE relay using open airinterface." 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC).*

Chapter 9- Conclusion

This chapter concludes the dissertation.

Other research contributions

Some of the works performed during this Ph.D. thesis which has not been included in this dissertation due to space limitation have been published in the following:

- *Esrafilian, Omid, and David Gesbert. "Simultaneous user association and placement in multi-uav enabled wireless networks." In WSA 2018; 22nd International ITG Workshop on Smart Antennas.*
- *Chen, Junting, Omid Esrafilian, David Gesbert, and Urbashi Mitra. "Efficient algorithms for air-to-ground channel reconstruction in UAV-aided communications." In 2017 IEEE Globecom Workshops (GC Wkshps).*
- *Gangula, Rajeev, Paul de Kerret, Omid Esrafilian, and David Gesbert. "Trajectory optimization for mobile access point." In 2017 51st Asilomar Conference on Signals, Systems, and Computers.*

Chapter 2

System Model

2.1 Introduction

In this chapter, we aim to present the essential concepts pertaining to UAV communication which are useful for research in different frameworks of UAVs serving as aerial terminals or communication platforms. We specifically talk about the channel model, performance metrics, and mathematical formulation for performance optimization via devising a trajectory for the UAV.

Note that in this thesis, we consider a low altitude UAV (when compared with e.g. balloons) which is flying over an urban area consisting of a number of city buildings. There are K static ground nodes which are randomly scattered over the city, and $\mathbf{u}_k = [x_k, y_k, z_k]^{\mathrm{T}} \in \mathbb{R}^3, k \in [1, K]$ represents the k-th user's location. The location of the UAV at time t is denoted by $\mathbf{v}(t) = [x(t), y(t), z(t)]^{\mathrm{T}} \in \mathbb{R}^3$. We assume that the drone is equipped with a GPS receiver, hence $\mathbf{v}(t)$ is known at any time.

2.2 Channel Models

UAV communications mainly involve three types of links, namely the ground BS-UAV link, the UAV-ground terminal link, and the UAV-UAV link. As the communication between UAVs which are flying high enough with moderate inter-UAVs distance typically occurs in clear airspace, the UAV-UAV channel is usually characterized by the simple free-space path loss model [21; 22]. Therefore, we rather focus on the Air-to-ground terminal (and vice-versa) channel model. In general, the existing channel models for the extensively studied terrestrial communication systems can be applied to UAV communications. However, as UAV systems involve transmitters and/or receivers with altitude much higher than those in conventional terrestrial systems, customized mathematical models have been developed to more accurately characterize the unique propagation environment for UAV communications at a different altitude. Significant efforts have been devoted to the channel measurements and modeling for UAV communications, where some recent surveys on them can be found in e.g., [23; 24; 25]. In this section, we provide an overview of the UAV channel model to facilitate the performance analysis of UAV communication

systems.

2.2.1 Segmented Channel Model

The quality of channel models lies in their ability to correctly predict power attenuation as a function of distance, frequency, as well as to characterize seemingly random blockage. While some models encompass multipath fading, the placement optimization time scale is usually much longer than the fast fading coherence time, hence fast fading can be averaged out in the first approximation. A simple received signal strength indicator (RSSI) model then ensues [26; 27]:

$$\gamma_s = \frac{\beta_s}{d^{\alpha_s}} \xi_s \, , \; s \in \{1, \cdots, S\}, \tag{2.1}$$

where s is a class index, α_s is a path loss exponent, β_s is a channel gain offset, and ξ_s is a random variable that captures additional non-predictable behavior (log-normal shadowing, additional noise sources, etc.), and d is the distance between the receiver and the transmitter. The model in (2.1) is often referred to as segmented, with s called the segment value, reflecting the degree of link's obstruction and its strong dependence on local terrain scenario. A simplified analysis may consider a single segment ($S = 1$), e.g free space propagation or LoS everywhere. The next most popular scenario just distinguishes between $S = 2$ segments, with $s \in \{\text{LoS}, \text{NLoS}\}$ links. In a deterministic segmented channel model, the segment value s is directly predicted from a 3D terrain map or possibly from UAV radio measurements. For example regarding the LoS/NLoS classification of a link, we can leverage the knowledge of a 3D city map [28]. Based on such map, we can predict LoS (un)availability on any given links from a trivial geometry argument: For a given UAV position, the BS is considered in LoS to the UAV if the straight line passing through the UAV's and the ground node's position lies higher than any buildings in between. The channel gain in dB can be written as

$$g_s = ß_s - \alpha_s \varphi(d) + \eta_s, \tag{2.2}$$

where $g_s = 10 \log_{10} \gamma_s$, $ß_s = 10 \log_{10} \beta_s, \varphi(d) = 10 \log_{10} (d)$, $\eta_s = 10 \log_{10} \xi_s$, and η_s is modeled as a Gaussian random variable with $\mathcal{N}(0, \sigma_s^2)$.

For a greater reality match, one may increase the number S of segments to account for meaningful intermediate degrees of obstruction such as concrete building vs. wood-walled structure vs. foliage, etc. [29]. Of course, expanding the segment values means added complexity as well as a greater noise sensitivity when doing model classification on the basis of real link strength measurements. Note that in reality, the segment value can be considered stable while the UAV is flying over some limited neighborhood, while sharply transitioning to another value as the UAV goes over a street corner or flies behind a large building. A piece-wise quasi-static behavior of s is typically assumed for segmented channel models. In fig. 2.1, an illustration of the segmented channel model by considering two ground users in a given city is shown. The status of each ground node-UAV link is determined by leveraging the 3D map of the city.

It is noteworthy that channel measurements and modeling for UAV communications are still ongoing research. Incorporating various issues into the channel modeling would

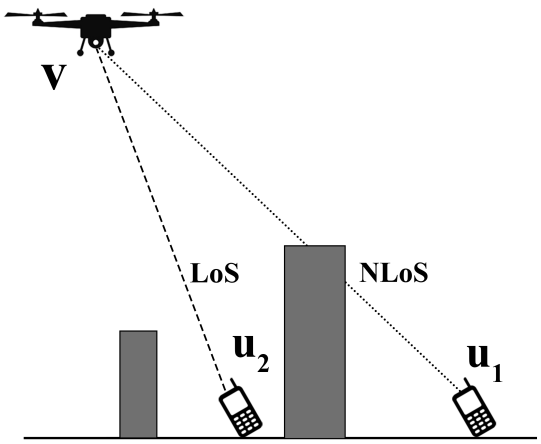


Figure 2.1 – Two-segment modeling for Air-to-ground channels in a given city.

be beneficial for the precise performance analysis and the practical design of the UAV communication systems. For example, analyzing the effects of UAV mobility patterns and the blades rotation on the channel, the multiple input multiple output (MIMO) and massive MIMO channel modeling [30; 31], the millimeter wave UAV channel modeling [32], etc. Such aspects remain however beyond the scope of this thesis.

2.2.2 Probabilistic Link Attenuation Models

As mentioned in the previous section, in a deterministic segmented channel model, the segment value s is directly predicted from a 3D terrain map or possibly from UAV radio measurements. However, in the absence of maps, probabilistic segmented models can be employed, whereby the segment value is simply governed by a likelihood parameter [15; 33]. The two-segment case ($S = 2$), has been particularly used in the literature to date as it naturally lends itself to closed-form tractability. In this case, a given link l between a ground node and the UAV is simply classified as LoS or NLoS where the LoS probability $p_{\text{LoS}}(l)$ follows a parametric model. A popular model is as follows [15]:

$$p_{\text{LoS}}(l) = \frac{1}{1 + \exp\left(-a\theta_l + b\right)}, \tag{2.3}$$

where θ_l denotes the elevation angle between the horizontal plane and the axis between the drone and the ground node. Finally, $\{a, b\}$ are the model coefficients which are specific for the city that the UAV is deployed and can be learned via a fitting method using labeled training data points, for instance using prior radio measurements or a terrain map. In order to infer the average link gain, a weighted average of γ_s over random value s, accounting for the segmented probability, can be carried out as follows:

$$\Lambda(l) = \sum_{s \in \{\text{LoS,NLoS}\}} \gamma_s p_s(l), \tag{2.4}$$

where $p_{\mathrm{NLoS}}(l) = 1 - p_{\mathrm{LoS}}(l)$. While they lend themselves to closed form mean gain analysis and give useful insights in the role played by certain system design parameters, global probabilistic models such as (2.4) are difficult to rely on by themselves in real-life robotic placement problems. The main issue is that such approaches can't guarantee an actual communication link's quality at a probabilistically-optimized UAV location. Instead, it is essential to exploit local information relevant to the actual terrain surrounding the UAV. There are several ways to infer such information in practice, be it from 3D terrain data or radio measurements from the scene of interest as we detail in Section 3.2.3.

2.3 Communication Performance Metric

Regarding the UAV communications, similar performance metrics can be used as the conventional terrestrial communications such as link signal-to-interference-plus-noise ratio (SINR), outage (or coverage) probability, communication throughput, etc. Moreover, in certain scenarios, more practical metrics such as mission completion time [34; 35] or the energy consumption [36] may be of interest. In the following, we elaborate on some of the performance metrics in the context of UAV communications.

2.3.1 SINR

Let's assume a UAV carrying radio equipment and a set of K ground nodes. For the communication link between the UAV at time t and the k-th ground node, when the UAV is transmitting, the SINR at the receiver can be expressed as

$$\rho_k(\mathbf{v}(t)) = \frac{P\,\gamma(\mathbf{u}_k)}{\sum_{i\in[1,K]\setminus k} I_i + \sigma^2}, \tag{2.5}$$

where P is the UAV transmission power, $\gamma(\mathbf{u}_k)$ is the desired received signal power at k-th ground node, I_i denotes the interference from the ground node i, and σ^2 is the noise power at the receiver.

When the UAV is receiving information from the k-th ground user, its SINR is given by

$$\rho_k(\mathbf{v}(t)) = \frac{P_k\,\gamma(\mathbf{u}_k)}{\sum_{i\in[1,K]\setminus k} \tilde{I}_i + \sigma^2}, \tag{2.6}$$

where P_k is the k-th ground node transmission power, and $\tilde{I}_i$ is the interference from the ground node i received at the UAV.

It is worth mentioning that, for the air-ground link when the UAV is transmitting, by changing the UAV location its own link SINR is just affected only through the desired signal power; while in the case with a UAV receiver, it has an effects on the link SINR in a more complicated manner, through both the desired signal and undesired interference powers from other ground nodes.

In a special case, when there is no interference, the SINR is equivalent to signal-to-noise ratio (SNR). the SNR at the receiver when the UAV is transmitting is given

by

$$\rho_k(\mathbf{v}(t)) = \frac{P\,\gamma(\mathbf{u}_k)}{\sigma^2}, \tag{2.7}$$

and when the UAV is receiving information from the k-th ground user, its SNR equals to

$$\rho_k(\mathbf{v}(t)) = \frac{P_k\,\gamma(\mathbf{u}_k)}{\sigma^2}. \tag{2.8}$$

Note that, in this thesis we assume that there is no interference between the nodes in the network by virtue of the orthogonal channel access. For this, hereafter we just use the term SNR.

2.3.2 Outage

Assuming that the UAV flies along a certain trajectory where its location at any time t is given by $\mathbf{v}(t), 0 \leq t \leq T$, where T is the mission completion time. The outage is defined as the amount of time where the SNR for the link between the UAV and the k-th ground node falls below a minimum SNR threshold $\rho_{\min}$, mathematically speaking:

$$T_{\text{out}} = \int_{t=0}^{T} \mathbb{I}\{\rho_k(\mathbf{v}(t)) < \rho_{\min}\}\,dt, \tag{2.9}$$

where $\rho_k(\mathbf{v}(t))$ denotes the SNR of the link between the UAV at time t and the k-th ground node, and $\mathbb{I}\{A\}$ is an indicator function returning value 1 if A is true. In Chapter 7, a trajectory is designed in order to satisfy the outage constraint $\rho_k(\mathbf{v}(t)) \geq \rho_{\min}$ all along the path while minimizing the flying time.

The outage can also be written in terms of probability. Since the SNR generally varies in both space and time and thus can be modeled as a random variable. For a minimum SNR threshold $\rho_{\min}$, the outage for the link between the UAV and the k-th ground node is given by

$$\mathrm{P}_{\text{out}}(\mathbf{u}_k) = \min_{0 \leq t \leq T} \Pr(\rho_k(\mathbf{v}(t)) < \rho_{\min}). \tag{2.10}$$

Note that for a given UAV locations, the outage probability needs to take into consideration the randomness in both time (i.e. the shadowing effect) as well as space (the link LoS/NLoS probabilities).

2.3.3 UAV Energy Consumption

One of the critical issues of UAV communications is the limited onboard UAVs energy. For this, the energy efficiency of UAV communications is of the essence [36; 17; 37]. If we assume that the UAV flies with a constant speed, then the overall mission time (or the trajectory length) can be considered using a simplified mechanical robotic model to be equivalent (proportional) to energy consumption. For instance in an intelligent data harvesting scenario by the UAV from ground sensors, we will seek to collect data from the ground level sensors within the minimum flying time.

2.3.4 Communication Throughput

Within the UAV mission time T, the instantaneous achievable rate (in a single user case) for the link between the k-th ground node and the UAV at time t is as follows

$$R_k(t) = \log_2(1 + \rho_k(\mathbf{v}(t))). \tag{2.11}$$

The average achievable communication throughput over the random channel shadowing effect is thus given by

$$\hat{R}_k(t) = \mathrm{E}[\log_2(1 + \rho_k(\mathbf{v}(t)))]. \tag{2.12}$$

Then the average communication throughput of the link between the k-th node and the UAV can be written as

$$R_k = \mathrm{E}\left[\int_0^T R_k(t)\, dt\right] = \int_0^T \hat{R}_k(t)\, dt. \tag{2.13}$$

The communication throughput in a multi-user case and assuming a time-division multiple access (TDMA) scheme is discussed in Chapter 3.

2.4 Trajectory Design in UAV-aided Wireless Networks

The key advantage of using UAVs in wireless communication lies in the high mobility and fast deployment of the UAVs which can bring significant gains to the performance of the network besides the traditional communication designs such as scheduling and resource allocation. For the performance optimization of the UAV communication systems, the generic mathematical problem can be considered as follows [13]

$$\begin{aligned}
\max_{\mathbf{v}(t),\mathcal{A}(t)} \quad & U(\mathbf{v}(t), \mathcal{A}(t)) && \text{(2.14a)}\\
\text{s.t.} \quad & f_i(\mathbf{v}(t)), i \in [1, I_1], && \text{(2.14b)}\\
& g_i(\mathcal{A}(t)), i \in [1, I_2], && \text{(2.14c)}\\
& h_i(\mathbf{v}(t), \mathcal{A}(t)), i \in [1, I_3], && \text{(2.14d)}
\end{aligned}$$

where $\mathcal{A}(t)$ is a set of variables pertaining to the all communication relevant design over the time, such as scheduling, transmit power control, etc., and $U(.,.)$ denotes the desired utility function to be maximized and depending on the application can be any of the metrics introduced in the previous section. $f_i(.)$ represents the constraints related to the UAV trajectory at any time t, $g_i(.)$ captures all the communication related constraints, and $h_i(.,.)$ involves both UAV trajectory and communication variables such as the SNR constrained which is a function of the UAV location and the communication parameters. Note that problem (2.14) is recast as a static placement problem if the UAV location is independent of the time t which is a special case of the trajectory design problem, in other words $\mathbf{v}(t) = \mathbf{v}, \forall t$.

The UAV trajectory constraints can depend on the type of the UAV, the application, the mission requirement, etc. In the following we refer to some of the common UAV trajectory constraints:

- **Altitude range:** In practice, a maximum and a minimum flying altitude need to be considered for the UAV. Typically, the maximum flying altitude is limited by the regulation depending on the flying area, and the minimum altitude is chosen to avoid the collision with the obstacles (i.e. in a city the minimum altitude can be set as the height of the tallest building in the city).
- **Initial/destination locations:** In most of the cases, it's required that the UAV starts from an initial point and terminates in a pre-determined destination location. Thus, the initial and destination location constraints can be expressed as

$$\mathbf{v}(0) = \mathbf{v}_{\mathrm{I}},\ \mathbf{v}(T) = \mathbf{v}_{\mathrm{F}}, \tag{2.15}$$

  where $\mathbf{v}_{\mathrm{I}}, \mathbf{v}_{\mathrm{F}}$ are the initial and the destination locations, respectively.
- **Maximum velocity:** Depending on the type of the UAV, the maximum velocity can be different:

$$\dot{\mathbf{v}}(t) \leq v_{\max}. \tag{2.16}$$

  Note that, for the fixed-wing UAVs a minimum velocity needs to be considered as well.

More constraints, depending on the application, can be listed down such as maximum acceleration, obstacle and collision avoidance, safe flying area, etc. which is out of the scope of this dissertation. The problem of trajectory design and the UAV placement with specific constraints will be discussed in details in Chapter 3.

2.5 Cellular-connected UAV Trajectory Design

The enabling of safe cellular controlled UAVs beyond visual line of sight is expected to open important future opportunities in different domains. A key challenge in this area lies in the design of trajectories which, while allowing the completion of the UAV mission, can guarantee reliable cellular connectivity all along the path. In general, the problem of designing a trajectory for the UAV under cellular connectivity constraints can be formulated as follows:

$$\max_{\mathbf{v}(t),\mathcal{A}(t)} \quad U(\mathbf{v}(t)) \tag{2.17a}$$

$$\text{s.t.} \quad f_i(\mathbf{v}(t)), i \in [1, I_1], \tag{2.17b}$$

$$g_i(\mathcal{A}(t)), i \in [1, I_2], \tag{2.17c}$$

$$h_i(\mathbf{v}(t), \mathcal{A}(t)), i \in [1, I_3], \tag{2.17d}$$

where $\mathcal{A}(t)$ is a set of variables pertaining to the all communication relevant designs over the time, and $U(.)$ is the desired utility function to be maximized. In the context of cellular-connected UAV trajectory design, the utility function is mainly the UAV energy consumption (or equivalently the flying time if the UAV movies with a constant velocity over the time). $f_i(.)$ represents the constraints related to the UAV trajectory at any

time t and can be any of the constraints listed in the previous section. $g_i(.)$ contains all the communication related constraints. One of the common communication related constraints is the minimum SNR constraint which is as follows:

$$\min_{0 \leq t \leq T} \max_{k \in [1,K]} \rho_k(\mathbf{v}(t)) \geq \rho_{\min}, \tag{2.18}$$

where $\rho_k(\mathbf{v}(t))$ denotes the SNR of the link between the UAV at time t and the k-th ground BS. Constraint (2.18) implies that the minimum SNR obtained by the UAV during the mission needs to be greater than or equal to the threshold $\rho_{\min}$. Constraints $h_i(.,.)$ involves both UAV trajectory and communication variables. In Chapter 7, the problem of trajectory design for a UAV under cellular connectivity constraint is discussed in details.

Chapter 3

Map-based Placement and Trajectory Design in UAV-aided Wireless Networks

3.1 Introduction

The use of UAVs as FRANs in future wireless communication networks is currently gaining significant attention for its ability to yield ultra-flexible deployments, in use cases ranging from disaster recovery scenarios, coverage of flash-crowd events, and data harvesting in IoT applications [6; 38; 39].

Several new and fascinating issues arise from the study of flying RANs in a wireless networks. These can be broadly categorized into placement and path planning problems. While placement problem deals with finding flexible yet static locations of the UAV BSs, path planning involves finding UAV trajectories. In both cases the aim is to optimize metrics like throughput, network coverage, energy efficiency, etc., [15; 40; 41; 42; 43; 44; 45; 46; 47; 48; 17].

When it comes to placement or trajectory design problems, most existing solutions rely on simplified channel attenuation models which are based on either (deterministically guaranteed) LoS links [43; 44; 45; 46; 17; 48; 47], or predictive models for the probability of occurrence of a LoS link [41; 42; 15; 40; 49]. In the latter approach, a *global* statistical model predicts the LoS availability as a function of, e.g., UAV altitude and distance to the user. The advantage of the global statistical LoS model lies in its simplicity for system analysis. However, it lacks actual performance guarantees for either placement or trajectory design algorithms when used in a real-life navigation scenario. The key reason for this is that, the *local* terrain topology may sharply differ from the predictions drawn from statistical features. In order to circumvent this problem, the embedding of actual 3D city map data in the UAV placement algorithms has been recently proposed [43; 42]. Map-based approaches help providing a reliable prediction of LoS availability for any pair of UAV and ground node locations, hence lead to improved performance guarantees. However, the gain comes at the expense of computational and memory costs related

to processing of the map data. So far, map-based approaches have been investigated mainly for static UAV placement [43; 44; 42]. In many scenarios, including IoT data harvesting, there is an interest in flying along a path that brings the UAV-mounted BS closer to each and every ground node. However, to the best of our knowledge, none of the previous works have considered the crucial advantage of exploiting 3D map data in communication-oriented UAV trajectory design.

This chapter investigates the problem of optimal placement and trajectory design of a UAV in wireless networks. In the first part, we focus on the problem of the UAV trajectory design in an intelligent data harvesting scenario in which a UAV-mounted flying BS is providing data communication services to a number of radio nodes spread over the ground. The trajectory optimization is then combined with a node scheduling algorithm to maximize the worst average throughput among the ground nodes.

The second part of this chapter is devoted to studying the problem of optimal placement of a UAV which provides communication services by acting as a flying wireless relay between a fixed BS and ground users. The proposed approach builds on the knowledge of the terrain topology where the network is deployed and aims at finding the optimal position of the UAV that maximizes the throughput in the max-min sense.

3.2 Optimal Trajectory Design for an Intelligent Data Harvesting

A wireless communication system where a UAV-mounted flying BS serving K static ground level nodes (IoT sensors, radio terminals, etc.) in an urban area is considered. The k-th ground node, $k \in [1, K]$, is located at $\mathbf{u}_k = [x_k, y_k]^{\mathrm{T}} \in \mathbb{R}^2$. By no means the ground level node assumption is restrictive, the proposed algorithms in this work can in principle be applied to a scenario where the nodes are located in 3D. The UAV's mission consists of a communication phase of duration T. We assume that the propagation parameters of the environment have been learned beforehand by collecting the radio measurements from the ground users (for more details see Chapter 4). These parameters are then used to optimally serve the ground nodes. The time-varying coordinate of the UAV/drone is denoted by $\mathbf{v}(t) = [x(t), y(t), z(t)]^{\mathrm{T}} \in \mathbb{R}^3, 0 \leq t \leq T$, where $z(t)$ represents the altitude of the drone.

For the ease of exposition, we assume that the time period T is discretized into N equal-time slots. The time slots are chosen sufficiently small such that the UAV's location, velocity, and channel gains can be considered to remain constant in one slot. Hence, the UAV's position $\mathbf{v}(t)$ is approximated by a sequence

$$\mathbf{v}[n] = [x[n], y[n], z[n]]^{\mathrm{T}}, \ n \in [1, N]. \tag{3.1}$$

We assume that the ground nodes and the drone are equipped with GPS receivers, hence the coordinates $\mathbf{u}_k, \forall k$ and $\mathbf{v}[n], \ n \in [1, N]$ are known.

3.2.1 Communication System Model

We assume that the ground nodes are served by the drone in a TDMA manner. Let $q_k[n]$ denote the scheduling variable, then the TDMA constraints can be written as

$$\sum_{k=1}^{K} q_k[n] \leq 1\,,\ n \in [1, N], \tag{3.2}$$

$$q_k[n] \in \{0, 1\}\,,\ n \in [1, N],\ k \in [1, K], \tag{3.3}$$

where $q_k[n] = 1$ indicates that the node k is scheduled in time slot n. For the scheduled node, the average throughput is given by

$$C_k[n] = \mathrm{E}\,\log_2\left(1 + \frac{P\gamma_k[n]}{\sigma^2}\right), \tag{3.4}$$

where $\gamma_k[n]$, according to (2.1), is the channel gain between the k-th node and the UAV at time step n, P denotes the up-link transmission power of the ground node, and the additive white Gaussian noise power at the receiver is denoted by σ^2. Hence, the average achievable throughput of the k-th ground node over the course of the communication trajectory is given by

$$C_k = \frac{1}{N}\sum_{n=1}^{N} q_k[n]C_k[n]. \tag{3.5}$$

3.2.2 Joint Scheduling and Trajectory Optimization

We consider the problem of efficient data harvesting where data originates from ground nodes and efficiency is meant in a max-min sense across the nodes. The problem of maximizing the minimum average throughput among all ground nodes by jointly optimizing node scheduling and UAV's trajectory can be formulated as

$$\max_{\mathcal{X},z,\mathcal{Q}} \min_{k\in[1,K]} C_k \tag{3.6a}$$

$$\text{s.t. } \|\mathbf{v}[n] - \mathbf{v}[n-1]\| \leq \rho_{\max},\ n \in [2, N], \tag{3.6b}$$

$$\mathbf{v}[1] = \mathbf{v}[N], \tag{3.6c}$$

$$h_{\min} \leq z[n] \leq h_{\max},\ n \in [2, N] \tag{3.6d}$$

$$(3.2),\ (3.3), \tag{3.6e}$$

where $\mathcal{Q} = \{q_k[n],\ \forall n,\ \forall k\}$ is the set of scheduling variables, and $\mathcal{X} = \{(x[n], y[n]),\ \forall n\}$ denotes the discretized trajectory set of length N in 2D. We assume that the drone flies at a fixed altitude $z[n] = z,\ \forall n$. The maximum speed constraint of the UAV is reflected in (3.6b), where $\rho_{\max} = v_{\max}T/N$, and $v_{\max}$ is the maximum speed of the UAV. (3.6c) implies a possible loop trajectory constraint[1]. (3.6d) implies that the drone always flies above all the city's buildings, where $h_{\min}$ is the height of the tallest building in the city, and below altitude $h_{\max}$.

Problem (3.6) is challenging to solve due to the following issues:

[1]This is by no means a restriction, the starting and the terminal points of the trajectory can be any arbitrary locations.

- The scheduling variables $q_k[n]$ are binary and include integer constraints.
- The objective function (3.6a) is a non-convex with respect to the drone trajectory variables.
- Since the 3D city map, node locations, and UAV location at time n are known, then in theory the LoS or NLoS status of the link can be finely predicted and, hence, the link gain $\gamma_k[n]$ can be computed from (2.1) up to the random shadowing. Unfortunately, such a direct exploitation of the rich *raw* map data leads to a highly non-differentiable problem in (3.6).

We overcome these difficulties by approximation using the same framework as [46] by employing the block-coordinate descent [50] and sequential convex programming [51] techniques. However, the key difference is that we optimize the drone's altitude, and also exploit the 3D city map by introducing a *statistical map compression* approach that enables us to take into account the LoS and NLoS predictions.

3.2.3 LoS Probability Model Using Map Compression

Statistical map compression approach relies on converting 3D map data to build a reliable node location dependent LoS probability model. The LoS probability for the link between the drone located at altitude z and the k-th ground node in the n-th time slot is given by

$$p_k[n] = \frac{1}{1+\exp(-a_k\theta_k[n]+b_k)}, \tag{3.7}$$

where $\theta_k[n] = \arctan(z/r_k[n])$ denotes the elevation angle and $r_k[n]$ is the ground projected distance between the drone and the k-th node located at $\mathbf{u}_k$ in the time slot n, and $\{a_k, b_k\}$ are the model coefficients.

The LoS probability model coefficients $\{a_k, b_k\}$ are learned (i.e. by utilizing logistic regression method[52]) by using a training data set formed by a set of tentative UAV locations around the k-th ground node along with the true LoS/NLoS label obtained from the 3D map. Interestingly, the model in (3.7) can be seen as a localized extension of the classical (global) LoS probability model used in [15; 40]. The key difference lies in the fact that, a *local* LoS probability model will give performance guarantees which a global model cannot.

Using (3.7), the average channel gain of the link between the drone and the k-th ground node in the n-th time slot is

$$\mathrm{E}[\gamma_k[n]] = \left(\frac{(d_k[n])^{(A-1)\alpha_{\mathrm{LoS}}} - B}{1+\exp(-a_k\theta_k+b_k)} + B\right)\frac{\beta_{\mathrm{LoS}}}{(d_k[n])^{\alpha_{\mathrm{NLoS}}}}, \tag{3.8}$$

where $B = \frac{\beta_{\mathrm{NLoS}}}{\beta_{\mathrm{LoS}}}$, $A = \frac{\alpha_{\mathrm{NLoS}}}{\alpha_{\mathrm{LoS}}} \geq 1$, and $d_k[n] = \sqrt{z^2 + (r_k[n])^2}$ is the distance between the k-th ground node and the drone. The details of the proof are given in Appendix A.1.

3.2.4 Proposed Solution for Communication Trajectory Optimization

In this section, we first approximate the original optimization problem in (3.6) to a *map-compressed* problem and then present an iterative algorithm using block coordinate descent for solving it. Using (3.8) and Jensen's inequality [53], the average throughput upper-bound then can be written as

$$C_k^{\text{up}} = \frac{1}{N}\sum_{n=1}^{N} q_k[n] C_k^{\text{up}}[n] \ , \ k \in [1, K], \tag{3.9}$$

where

$$C_k^{\text{up}}[n] = \log_2\left(1 + \frac{P\,\text{E}[\gamma_k[n]]}{\sigma^2}\right). \tag{3.10}$$

We then approximate the original problem in (3.6) into the following *map-compressed* problem:

$$\max_{\mathcal{X},z,\mathcal{Q},\mu} \ \mu \tag{3.11a}$$
$$\text{s.t. } C_k^{\text{up}} \geq \mu \ , \ \forall k, \tag{3.11b}$$
$$0 \leq q_k[n] \leq 1 \ , \ \forall k, \ \forall n, \tag{3.11c}$$
$$\text{(3.6b), (3.6c), (3.6d), (3.2),} \tag{3.11d}$$

where the constraints in (3.11c) represent the relaxation of the binary scheduling variable into continuous variables. Moreover, map compression allows us to circumvent the non-differentiability aspect of the original problem (3.6) by compressing the 3D map information into a probabilistic LoS model. However, (3.11) is still difficult to solve since it is a joint scheduling and path planning problem and is not convex. To make this problem more tractable, we split it up into three optimization sub-problems and then classically iterate between them to converge to a final solution. Note that, the iteration index of the proposed algorithm is denoted by "j".

Scheduling

For a given UAV planar trajectory $\mathcal{X}$ and altitude z, the ground node scheduling can be optimized as

$$\max_{\mathcal{Q},\mu} \ \mu \tag{3.12a}$$
$$\text{s.t. } C_k^{\text{up}} \geq \mu \ , \ \forall k, \tag{3.12b}$$
$$\text{(3.2), (3.11c).}$$

This problem is a standard Linear Program (LP) and can be solved by using any optimization tools such as CVX [54].

Optimal Horizontal UAV Trajectory

For a given scheduling decision $\mathcal{Q}$, and drone's altitude z, we now aim to find the optimal planar trajectory by solving

$$\max_{\mathcal{X},\,\mu} \ \mu \tag{3.13a}$$

$$\text{s.t. } C_k^{\text{up}} \geq \mu \ , \forall k, \tag{3.13b}$$

$$\text{(3.6b), (3.6c).} \tag{3.13c}$$

The optimization problem (3.13) is not convex, since the constraint (3.13b) is neither convex nor concave. In general, there is no efficient method to obtain the optimal solution. Therefore, we adopt sequential convex programming technique for solving (3.13). To this end, the following results are helpful.

Lemma 3.2.1. *The function* $h(x,y) \triangleq \log\left(1 + f\left(x\right) g\left(y\right)\right)$ *is convex if* $\hat{h}(x,y) \triangleq \log\left(f\left(x\right) g\left(y\right)\right)$ *is convex and* $f(x) > 0$, *and* $g(y) > 0$.

Proof. See Appendix A.2. □

Proposition 3.2.1. *For any constant* $\tau, \lambda > 0$, *the function*

$$c(x,y,d) \triangleq \log\left(1 + \left[(\frac{1}{1+x})(\frac{1}{y}) + \tau\right] \frac{1}{d^{\lambda}}\right)$$

is convex.

Proof. See Appendix A.3. □

By defining the auxiliary variables $f_k[n]$, $w_k[n]$, $l_k[n]$, and $\theta_k[n]$, we can rewrite (3.13) as follows

$$\max_{\mathcal{V},\mathcal{X},\mu} \ \mu \tag{3.14a}$$

$$\text{s.t. } \frac{1}{N}\sum_{n=1}^{N} c_k\left(f_k[n], w_k[n], l_k[n]\right) \geq \mu \ , \forall k, \tag{3.14b}$$

$$w_k[n] = \left(\left(z^2 + l_k[n]\right)^{(A-1)\alpha_{\text{LoS}}/2} - B\right)^{-1} \ , \forall k, \forall n, \tag{3.14c}$$

$$f_k[n] = \exp\left(-a_k\theta_k[n] + b_k\right) \ , \forall k, \forall n, \tag{3.14d}$$

$$l_k[n] = \left(r_k[n]\right)^2 \ , \forall k, \forall n, \tag{3.14e}$$

$$\theta_k[n] = \arctan\left(z/\sqrt{l_k[n]}\right) \ , \forall k, \forall n, \tag{3.14f}$$

$$f_k[n], w_k[n], l_k[n], \theta_k[n] \geq 0 \ , \forall k, \forall n, \tag{3.14g}$$

$$\text{(3.6b), (3.6c),} \tag{3.14h}$$

where $\mathcal{V} = \{f_k[n], w_k[n], l_k[n], \theta_k[n] \,|\, \forall k, \forall n\}$ consists of all the auxiliary variables and

$$c_k\left(f_k[n], w_k[n], l_k[n]\right) \triangleq$$

$$\log_2\left(1+\left(\frac{1}{w_k[n](1+f_k[n])}+B\right)\frac{P\,\beta_{\mathrm{LoS}}}{\sigma^2\left(z^2+l_k[n]\right)^{\alpha_{\mathrm{NLoS}}/2}}\right). \tag{3.15}$$

Using Proposition 3.2.1, it can be easily seen that (3.15) is a convex function of variables $f_k[n], w_k[n],$ and $l_k[n]$. In constraint (3.14c), $w_k[n]$ can be convex or concave function depending on the value of B. However, in our case it is always convex since $z \geq h_{min}$, in a realistic scenario $\left(z^2+l_k[n]\right)^{(A-1)\alpha_{\mathrm{LoS}}/2} \gg B$. Moreover, all constraints (3.14d) to (3.14f) comprise convex functions. In order to solve problem (3.14), we utilize the sequential convex programming technique which solves instead the local linear approximation of the original problem. To form the local linear approximation, we use the given variables $\mathcal{X}^j, z^j$ in the j-th iteration of the algorithm to convert the above problem to a standard convex form. For the ease of exposition, we use $c_k[n]$ instead of $c_k\left(f_k[n], w_k[n], l_k[n]\right)$. First, let's start with constraint (3.14b), since any convex functions can be lower-bounded by its first order Taylor expansion, then we can write

$$\frac{1}{N}\sum_{n=1}^{N} q_k[n]c_k[n] \geq \frac{1}{N}\sum_{n=1}^{N} q_k[n]\tilde{c}_k[n] \geq \mu_{hp},$$

where $\tilde{c}_k[n]$ is an affine function and equals to the local first order Taylor expansion of $c_k[n]$ and μ_{hp} is a lower bound of μ. Similarly, we can convert (3.14c) to (3.14f) into the standard convex form by replacing them with their first order Taylor expansion. We can approximate problem (3.14) as follows

$$\max_{\mathcal{V},\mathcal{X},\mu_{hp}} \ \mu_{hp} \tag{3.16a}$$

$$\text{s.t. } \frac{1}{N}\sum_{n=1}^{N} q_k[n]\tilde{c}_k[n] \geq \mu_{hp}\,,\ \forall k, \tag{3.16b}$$

$$f_k[n] \geq \widetilde{f}_k[n]\ ,\ \forall k,\ \forall n, \tag{3.16c}$$

$$w_k[n] \geq \widetilde{w}_k[n]\ ,\ \forall k,\ \forall n, \tag{3.16d}$$

$$l_k[n] \geq \widetilde{l}_k[n]\ ,\ \forall k,\ \forall n, \tag{3.16e}$$

$$\theta_k[n] \geq \widetilde{\theta}_k[n]\ ,\ \forall k,\ \forall n, \tag{3.16f}$$

$$\text{(3.14g), (3.6b), (3.6c),} \tag{3.16g}$$

where the superscript " ˜ " denotes the local first order Taylor expansion. Now, we have a standard convex problem which can be solved by any convex optimization tools like CVX[2]. We denote the generated trajectory by solving (3.16) as $\mathcal{X}^{j+1}$.

Optimal UAV Altitude

Now we proceed to optimize the UAV altitude for a given horizontal UAV trajectory $\mathcal{X}$ and scheduling decision $\mathcal{Q}$. Similar to the preceding section, first we introduce auxiliary variables h, $m_k[n],$ and $o_k[n]$ consisting of convex functions as follows

$$m_k[n] = \exp\left(-a_k \arctan\left(z/r_k[n]\right)+b_k\right)\ ,\ \forall k,\ \forall n,$$

[2]Note that to minimize the approximation error, a tight local Taylor approximation is needed.

$$o_k[n] = \Big(\big(h + (r_k[n])^2\big)^{(A-1)\alpha_{\text{LoS}}/2} - B \Big)^{-1} \ , \forall k, \forall n,$$
$$h = z^2.$$

Akin to the last section, we find the UAV altitude by using the sequential convex programming with given local point z^j in the j-th iteration and the generated horizontal trajectory $\mathcal{X}^{j+1}$ in the last section. Finally, the UAV altitude is optimized as follows

$$\max_{\mathcal{W}, z, \mu_{alt}} \ \mu_{alt} \tag{3.17a}$$
$$\text{s.t.} \ \frac{1}{N} \sum_{n=1}^{N} q_k[n] \, \tilde{c}_k[n] \geq \mu_{alt} \ , \forall k, \tag{3.17b}$$
$$m_k[n] \geq \widetilde{m}_k[n] \ , \forall k, \forall n, \tag{3.17c}$$
$$o_k[n] \geq \widetilde{o}_k[n] \ , \forall k, \forall n, \tag{3.17d}$$
$$h \geq \widetilde{h}, \tag{3.17e}$$
$$m_k[n], o_k[n], \, h > 0, \, \forall k, \, \forall n, \tag{3.17f}$$
$$(3.6\text{d}), \tag{3.17g}$$

where $\tilde{c}_k[n]$ is the first order Taylor expansion of $c_k\left(m_k[n], o_k[n], h\right)$ which is a convex function and is defined similar to (3.15), and $\mathcal{W} = \{m_k[n], o_k[n], h \,|\, \forall k, \, \forall n\}$ comprises all the auxiliary variables. The superscript " ˜ " denotes the local first order Taylor expansion, and μ_{alt} is a lower bound of μ. We denote the drone altitude which is obtained by solving (3.17) as z^{j+1} to be used in the next iteration.

3.2.5 Iterative Algorithm

According to the preceding analysis, now we propose an iterative algorithm to solve the original optimization problem (3.6) by applying the block-coordinate descent method [50]. As mentioned earlier, we split up our problem into three phases (or blocks) of ground node scheduling, drone horizontal trajectory design, and flying altitude optimization over variables $\{\mathcal{Q}, \mathcal{X}, z\}$. In each iteration, we update just one set of variables at a time, rather than updating all the variables together, by fixing the other two sets of variables. Then, the output of each phase is used as an input for the next step. The rigorous description of this algorithm is summarized in Algorithm 3.2.1.

3.2.6 Proof of Convergence

In this section we prove the convergence of Algorithm 3.2.1 in a similar manner of [46]. To this end, in the j-th iteration we denote the $\mu(\mathcal{Q}^j, \mathcal{X}^j, z^j)$, $\mu_{hp}(\mathcal{Q}^j, \mathcal{X}^j, z^j)$, $\mu_{alt}(\mathcal{Q}^j, \mathcal{X}^j, z^j)$ as the optimal objective values of problems (3.12), (3.16), and (3.17) , respectively. From step (2) of Algorithm 3.2.1 for the given solution $\mathcal{Q}^{j+1}$, we have

$$\mu(\mathcal{Q}^j, \mathcal{X}^j, z^j) \leq \mu(\mathcal{Q}^{j+1}, \mathcal{X}^j, z^j),$$

Algorithm 3.2.1 Iterative algorithm for solving optimization problem (3.6).

1. Initialize all variables $\left\{\mathcal{Q}^j, \mathcal{X}^j, z^j\right\}$, $j = 1$.
2. Find the optimal solution of the scheduling problem (3.12) for given $\left\{\mathcal{X}^j, z^j\right\}$. Denote the optimal solution as $\mathcal{Q}^{j+1}$.
3. Generate the optimal communication trajectory in horizontal plane ($\mathcal{X}^{j+1}$) by solving (3.16) with given variables $\left\{\mathcal{Q}^{j+1}, \mathcal{X}^j, z^j\right\}$.
4. Solving optimization problem (3.17) given variables $\left\{\mathcal{Q}^{j+1}, \mathcal{X}^{j+1}, z^j\right\}$ and denote the solution as z^{j+1}.
5. Update $j := j + 1$.
6. Go to step 2 and repeat until the convergence (i.e. until observing a small increase in objective value).

since the optimal solution of problem (3.12) is obtained. Moreover, we can write

$$\begin{aligned}\mu(\mathcal{Q}^{j+1}, \mathcal{X}^j, z^j) &\overset{(a)}{=} \mu_{hp}(\mathcal{Q}^{j+1}, \mathcal{X}^j, z^j) \\ &\overset{(b)}{\leq} \mu_{hp}(\mathcal{Q}^{j+1}, \mathcal{X}^{j+1}, z^j) \\ &\overset{(c)}{\leq} \mu(\mathcal{Q}^{j+1}, \mathcal{X}^{j+1}, z^j).\end{aligned}$$

Step (a) holds due to $\mu_{hp}(\mathcal{Q}^{j+1}, \mathcal{X}^j, z^j)$ being a tight local first order Taylor approximation of problem (3.14) at the local points. Step (b) is true, since we can find the optimal solution of problem (3.16) with the given variables $\{\mathcal{Q}^{j+1}, \mathcal{X}^j, z^j\}$, and (c) holds because $\mu_{hp}(\mathcal{Q}^{j+1}, \mathcal{X}^{j+1}, z^j)$ is the lower bound of the objective value $\mu(\mathcal{Q}^{j+1}, \mathcal{X}^{j+1}, z^j)$. Then, by proceeding to step (4) of Algorithm 3.2.1 and given variables $\{\mathcal{Q}^{j+1}, \mathcal{X}^{j+1}, z^j\}$, we obtain

$$\begin{aligned}\mu(\mathcal{Q}^{j+1}, \mathcal{X}^{j+1}, z^j) &\overset{(d)}{=} \mu_{alt}(\mathcal{Q}^{j+1}, \mathcal{X}^{j+1}, z^j) \\ &\overset{(e)}{\leq} \mu_{alt}(\mathcal{Q}^{j+1}, \mathcal{X}^{j+1}, z^{j+1}) \\ &\overset{(f)}{\leq} \mu(\mathcal{Q}^{j+1}, \mathcal{X}^{j+1}, z^{j+1}).\end{aligned}$$

Step (d) is true since the local first order Taylor approximation in (3.17) is tight for the given local variables $\{\mathcal{Q}^{j+1}, \mathcal{X}^{j+1}, z^j\}$. (e) holds since, the optimization problem (3.17) can be optimally solved, and (f) is true due to $\mu_{alt}(\mathcal{Q}^{j+1}, \mathcal{X}^{j+1}, z^{j+1})$ is a lower bound of the objective value $\mu(\mathcal{Q}^{j+1}, \mathcal{X}^{j+1}, z^{j+1})$. Finally, we have

$$\mu(\mathcal{Q}^j, \mathcal{X}^j, z^j) \leq \mu(\mathcal{Q}^{j+1}, \mathcal{X}^{j+1}, z^{j+1}).$$

Which indicates that the objective value of Algorithm 3.2.1 after each iteration is non-decreasing and since it is upper bounded by a finite value, so the convergence of Algorithm 3.2.1 is guaranteed.

3.2.7 Trajectory Initializing

In this section, we propose a simple strategy to initialize the drone trajectory to be optimized later on by the introduced Algorithm 3.2.1. The initial trajectory is in form of a circle which is centered at $\mathbf{c}_{\text{trj}} = (x_{\text{trj}}, y_{\text{trj}})$ and the radius r_{trj} which is given by

$$r_{\text{trj}} = \frac{L_{\max}}{2\pi},$$

where $L_{\max} = T \cdot v_{\max}$. To determine the $\mathbf{c}_{\text{trj}}$, we use the notion of the (weighted) center of gravity of the ground nodes [42]. Moreover, the flying altitude is initialized at $h_{\max}$.

3.3 Optimal UAV Relay Placement in LTE Networks

In this section, we consider a scenario where a UAV is acting as a relay between a fixed BS and ground users in an urban area consists of a number of city buildings. There are K static ground level users randomly scattered over the city, and $\mathbf{u}_k = [x_k, y_k]^{\mathrm{T}} \in \mathbb{R}^2$. denotes the k-th user location. We assume that the 3D map of the city and the location of the users are known. The location of the UAV is denoted by $\mathbf{v} = [x, y, z]^{\mathrm{T}} \in \mathbb{R}^3$. Note that the placement problem can be considered as a special case of the trajectory design problem when $\mathbf{v}(t) = \mathbf{v}, \forall\, t$. We denote the location of the BS as $\mathbf{x}_b = [x_b, y_b, z_b]^{\mathrm{T}} \in \mathbb{R}^3$, which is also known.

Similar to section 3.2.3, for a link between a drone located at altitude z and the k-th user, the LoS probability is modeled by:

$$p_k = \frac{1}{1 + \exp\left(-a_k\, \theta_k + b_k\right)}, \tag{3.18}$$

where $\theta_k = \arctan(z/r_k)$ denotes the elevation angle and r_k is the ground projected distance between the drone and the k-th user located at $\mathbf{u}_k$, and $\{a_k, b_k\}$ are the model coefficients.

Using (3.18), the average channel gain between the k-th user and the drone can be written as follows:

$$\begin{aligned} \mathrm{E}\left[\gamma_k\right] &= p_k \gamma_{\mathrm{LoS},k} + (1 - p_k)\gamma_{\mathrm{NLoS},k} \\ &= \left(\frac{d_k^{(A-1)\alpha_{\mathrm{LoS}}} - B}{1 + \exp(-a_k \theta_k + b_k)} + B \right) \frac{\beta_{\mathrm{LoS}}}{d_k^{\alpha_{\mathrm{NLoS}}}}, \end{aligned} \tag{3.19}$$

where γ_k, according to (2.1), is the channel gain between the k-th user and the drone, $B = \frac{\beta_{\mathrm{NLoS}}}{\beta_{\mathrm{LoS}}}$, $A = \frac{\alpha_{\mathrm{NLoS}}}{\alpha_{\mathrm{LoS}}} \geq 1$, and $d_k = \sqrt{z^2 + r_k^2}$ is the distance between the k-th user and the drone.

3.3.1 Communication Model

The UAV serves only one user among the K users at a time by acting as a relay. We assume that a decode-and-forward type of relay protocol is used. If the UAV is serving k-th user, the achievable throughput on the UAV-user link can be upper-bounded by

$$C_k^{\mathrm{up}} = \log_2 \left(1 + \frac{P_d\, \mathrm{E}\left[\gamma_k\right]}{\sigma^2} \right). \tag{3.20}$$

It's also assumed that the UAV is connected to the BS at all times and the throughput of the BS-UAV link is upper-bounded by

$$C_b^{\mathrm{up}} = \log_2 \left(1 + \frac{P_b\, \mathrm{E}\left[\gamma_b\right]}{\sigma^2} \right), \tag{3.21}$$

where P_d, P_b are the downlink transmit powers of the UAV and BS, respectively. The additive white Gaussian noise at the receivers is denoted by σ^2. γ_b is the BS-UAV channel

gain. $\mathrm{E}[\gamma_b]$ is computed by averaging over the LoS component as the UAV is always connected to the BS with a LoS link at all times.

Since we assume decode-and-forward type of relaying, using the above bounds, the achievable throughput for the k-th user can be approximated as

$$C_k = \min(C_k^{\mathrm{up}}, C_b^{\mathrm{up}}). \tag{3.22}$$

3.3.2 UAV Placement Optimization

Since only one user can be served by the UAV at any time, in order to provide a performance guarantee to the connected user, we aim to find the UAV position which maximizes the minimum achievable rate of all the users. Using the upper-bound in (3.22), the placement problem can be formulated as

$$\max_{\mathbf{v}} \min_{k\in[1,K]} C_k \tag{3.23a}$$

$$\text{s.t. } h_{\min} \leq z \leq h_{\max}, \tag{3.23b}$$

The problem shown in (3.23) is non-convex, hence hard to solve. To solve (3.23), similar to Section 3.2, we propose an iterative algorithm by employing the block-coordinate descent to split it up into two sub-problems and then we utilize the sequential convex programming technique to solve each sub-problem. The algorithm then iterates between two phases to converge to a final solution. Note that, we use the notion of the center of gravity of ground users to initialize the drone position. Moreover, the flying altitude is initialized at $h_{\max}$. The convergence of the algorithm can be established along similar lines to the one provided in Section 3.2.6.

Planar Optimal UAV Placement

For a given drone's altitude z, the UAV position in the horizontal plane can be optimized by solving

$$\max_{x,y} \min_{k\in[1,K]} C_k. \tag{3.24}$$

This problem is not convex and to solve this problem first we introduce auxiliary variables f_k, w_k, l_k, l_b, and θ_k. We then rewrite (3.24) as follows

$$\max_{\mathcal{V},x,y,\mu} \mu \tag{3.25a}$$

$$\text{s.t. } c_k(f_k, w_k, l_k) \geq \mu \ , \forall k, \tag{3.25b}$$

$$c_{\mathrm{b}}(l_{\mathrm{b}}) \geq \mu \tag{3.25c}$$

$$w_k = \left(\left(z^2 + l_k\right)^{(A-1)\alpha_{\mathrm{LoS}}/2} - B\right)^{-1} \ , \forall k, \tag{3.25d}$$

$$f_k = \exp\left(-a_k\theta_k + b_k\right) \ , \forall k, \tag{3.25e}$$

$$l_k = r_k^2 \ , \forall k, \tag{3.25f}$$

$$l_{\mathrm{b}} = x^2 + y^2 \tag{3.25g}$$

$$\theta_k = \arctan\left(z/\sqrt{l_k}\right) \ , \forall k \tag{3.25h}$$

$$f_k, w_k, l_k, \theta_k, l_{\mathrm{b}} \geq 0 \ , \forall k, \tag{3.25i}$$

where

$$c_k\left(f_k, w_k, l_k\right) \triangleq$$

$$\log_2\left(1 + \left(\frac{1}{w_k(1+f_k)} + B\right) \frac{P\,\beta_{\mathrm{LoS}}}{\sigma^2\left(z^2 + l_k\right)^{\alpha_{\mathrm{NLoS}}/2}}\right), \tag{3.26}$$

and

$$c_{\mathrm{b}}\ \left(l_{\mathrm{b}}\right) = \log_2\left(1 + \frac{P_{\mathrm{b}}\,\beta_{\mathrm{LoS}}}{\sigma^2\left(\left(z - z_{\mathrm{b}}\right)^2 + l_{\mathrm{b}}\right)^{\alpha_{\mathrm{LoS}}/2}}\right).$$

It can be shown that $c_k\left(f_k, w_k, l_k\right), c_{\mathrm{b}}\ \left(l_{\mathrm{b}}\right)$ are convex and also all the constraints (3.25d) to (3.25h) comprise convex functions, however (3.25) in general is not convex optimization problem and it can be solved similar to the approach introduced in Section 3.2.4 by applying the sequential convex programming.

UAV Altitude Optimization

Now we proceed to optimize the UAV altitude for a given horizontal UAV position (x, y). The UAV altitude is optimized as follow

$$\max_{z} \ \min_{k\in[1,K]} C_k. \tag{3.27}$$

This problem is not convex and is solved in a similar manner as the last section by introducing same auxiliary variables and then applying the sequential convex programming.

3.4 Numerical Results

We consider a dense urban Manhattan-like area of size 600×600 square meters, consisting of a regular street and buildings. The height of the building is Rayleigh distributed within the range of 5 to 40 meters [15]. The average building height is 14 m. Propagation parameters are chosen as $\alpha_{\mathrm{LoS}} = 2.27$, $\alpha_{\mathrm{NLoS}} = 3.64$, $ß_{\mathrm{LoS}} = -30\,\mathrm{dB}$, $ß_{\mathrm{NLoS}} = -40\,\mathrm{dB}$ according to an urban micro scenario in [55]. The variances of the shadowing component in LoS and NLoS scenarios are $\sigma^2_{\mathrm{LoS}} = 2$ and $\sigma^2_{\mathrm{NLoS}} = 5$, respectively. The transmission power for ground nodes is chosen as $P = P_d = 30$ dBm, and for the BS's transmission power is $P_b = 36$ dBm, and the noise power at the receiver is -80 dBm. The UAV has a maximum speed of $v_{\max} = 10\,\mathrm{m/s}$.

We first start by evaluating the performance of the communication trajectory design algorithm. Since the communication trajectory design depends on the local LoS probability model, we first need to learn the probability model coefficients in (3.7). For

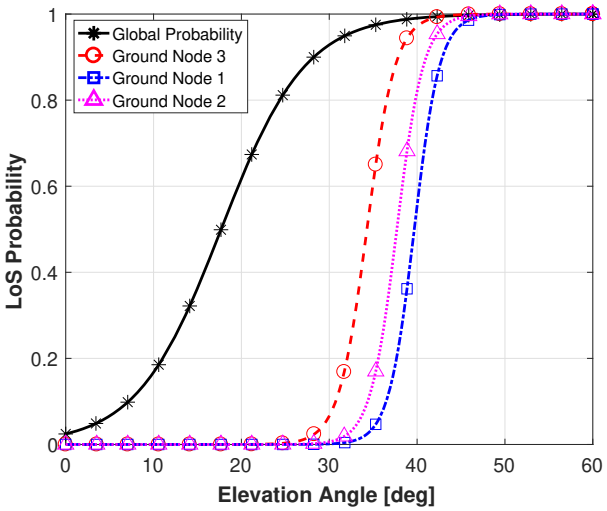


Figure 3.1 – Global LoS probability compared with the local LoS probability learned form the 3D map for three ground nodes.

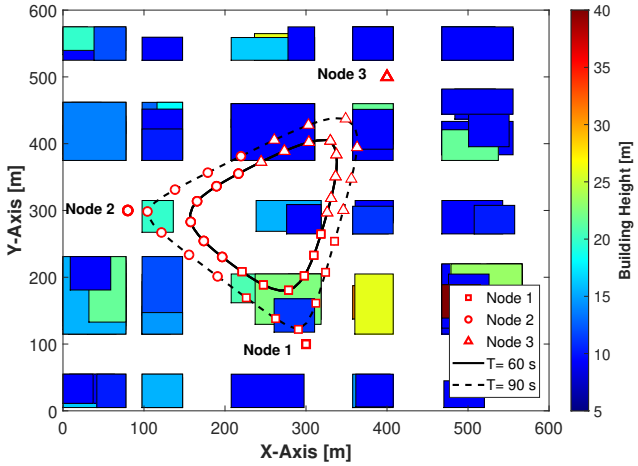
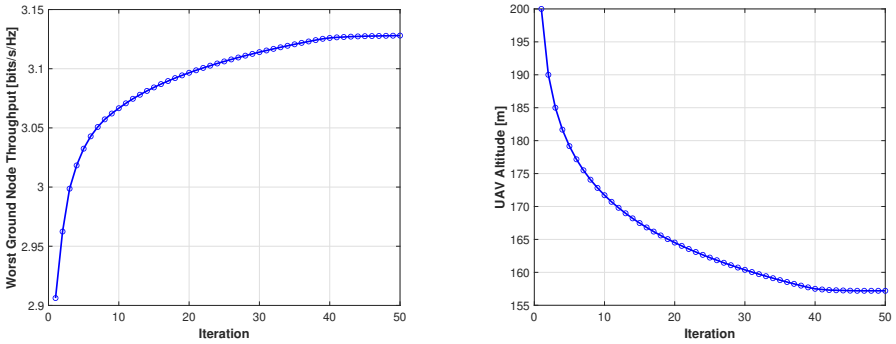


Figure 3.2 – Optimal drone trajectory and ground node scheduling for different flight times. As the flight time increases, the UAV gets closer to individual ground nodes.



(a) (b)

Figure 3.3 – (a) Throughput performance versus iteration and (b) drone altitude evolution versus iteration.

this, we employed the logistic regression method on the training data set obtained by randomly sampling around each ground node. The labeling is done with the true LoS status obtained from the 3D city map. Fig. 3.1 shows the obtained LoS probabilities for $K = 3$ ground nodes whose locations are shown in Fig. 3.2. We also plot the global LoS probability which is computed from the characteristics of the 3D map according to [15]. It is clear that the local probabilities have the sharper transitions and thus provide more information per ground node (i.e. if the node is surrounded by the tall buildings or is in a large open area), while the global probability can be considered as the average of the local LoS probability of the nodes in different locations.

In Fig. 3.2, we show the generated trajectory over the city buildings for different flight times (T). It is clear that by increasing T, the UAV exploits the flight time to improve the ground node link quality by enlarging the trajectory and moving towards the ground nodes. It is crucial to note that the generated trajectory is closer to the ground nodes which has the lower LoS probability (i.e. the ground nodes who are close to buildings or surrounded by tall skyscrapers). In Fig. 3.2, the drone tries to get closer to the ground nodes 1 and 2 since they are close to the buildings which mostly block the LoS link to the drone. Moreover, we illustrated the result of the ground node scheduling during the trajectory with different markers which are assigned to each node. Namely triangles, squares, and circles pertain, respectively, to ground nodes 1 to 3. For example, square markers shown on the trajectory indicate that the drone is serving the ground node 1 at that time.

We then outline the convergence behavior of Algorithm 3.2.1 by assuming $K = 3$ and $T = 90$ s. The drone altitude and worst ground node throughput versus iteration are shown in Fig.3.3. As we expected, the worst ground node throughput in each iteration improves and finally converges to a finite value.

In Fig. 3.4, the performance of the proposed map-based algorithm in comparison with two other approaches, which are briefly explained below, versus the flight time by considering $K = 6$ ground nodes is shown.

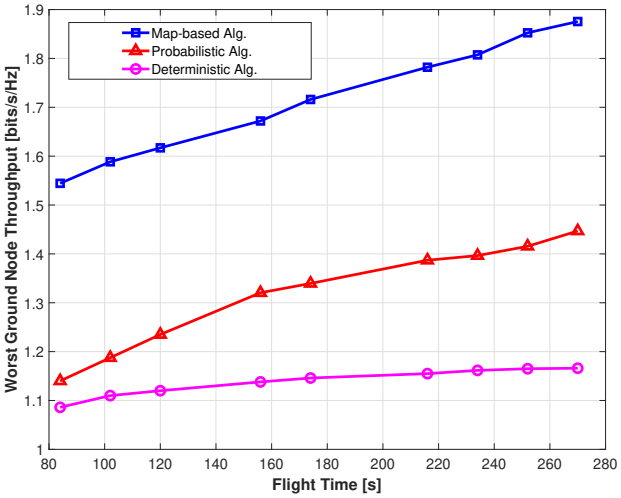


Figure 3.4 – Performance of the map-based algorithm in comparison with the probabilistic approach and the deterministic Algorithm for 6 ground nodes versus increasing the flight time.

- **Probabilistic algorithm**

  In the probabilistic approach, we consider the same trajectory design algorithm as proposed in this chapter with the difference that for a link between the drone located at altitude z and the k-th ground node, the LoS probability at the time step n is given by

  $$p_k[n] = \frac{1}{1 + \exp\left(-a\,\theta_k[n] + b\right)},$$

  where parameters $\{a, b\}$ are computed according to [15] and based on the characteristics of the 3D map. In other words, we use a global LoS probability model.

- **Deterministic algorithm**

  In the deterministic algorithm, an optimal trajectory is generated based on the method introduced in [46] which considers a single deterministic LoS channel model for the link between the drone and the ground nodes. In order to have a fair comparison, we modified this method by using an average path-loss instead of the pure LoS channel model. the channel parameters pertaining to the average path-loss model are learned by fitting one channel model for the whole measurements gathered from both LoS and NLoS ground nodes.

In general, the map-based algorithm outperforms the other approaches which is expected since in the proposed algorithm we utilize more information through the 3D map.

In Fig. 3.5 a performance comparison of the proposed map-based algorithm and the probabilistic approach under a fixed flight time by increasing the average buildings height

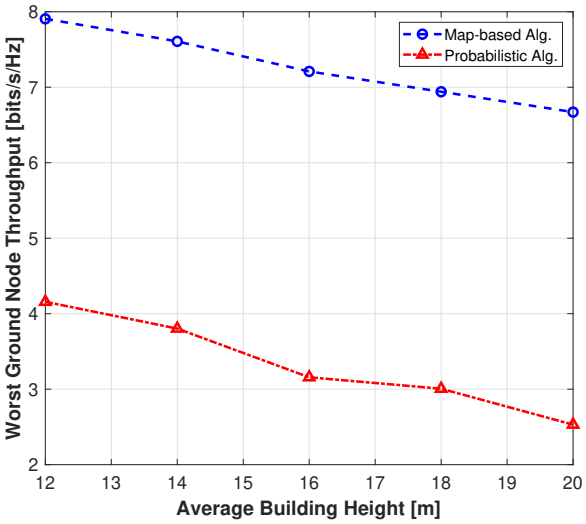


Figure 3.5 – Performance comparison of the map-based Algorithm and the probabilistic approach versus the average building height for a fixed flight time.

is illustrated. The map-based algorithm provides a better services to the ground nodes. For both algorithms by increasing the buildings height, the performance degraded since it is more likely that the link between the ground nodes and the drone over the course of the trajectory being NLoS.

In Fig. 3.6, the top view of the city and the optimal position of the UAV calculated by using the map-based placement algorithm for an example of 3 static ground nodes is shown. The BS is located at $\mathbf{x}_b = [0, 0, 50]^{\mathrm{T}}$ meters. The optimal altitude for the UAV is 270 meters. it is worth mentioning that, the optimal position of the UAV is closer to the ground node which is experiencing the worse channel condition (which in this case is user 2). In other words, the UAV tries to favor the worst node in the network.

3.5 Conclusion

This chapter considered the problem of trajectory design for a UAV BS that is providing communication services for a number of ground nodes in the context of an IoT data harvesting scenario. We have developed throughput-optimized trajectories such that the amount of data collected from each of the ground nodes is maximized. We have proposed an iterative algorithm that leverages the knowledge of the 3D city map via a novel map-compression method and uses the block coordinate descent and sequential convex programming techniques. It is also shown that the proposed algorithm is guaranteed to converge to at least a locally optimal solution. Moreover, we discussed the problem of optimal placement for a UAV relay in a LTE network which is a special case of the trajectory design problem.

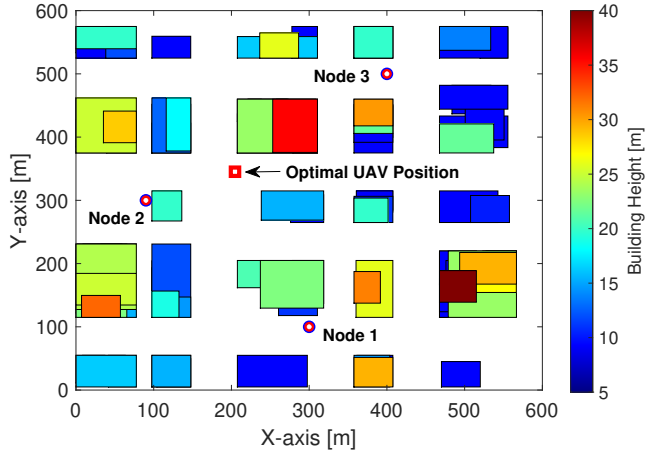


Figure 3.6 – Top view of the city and the optimal position of the UAV for 3 static ground nodes using the Map-based placement. The optimal altitude is 270 meters.

Chapter 4

Active Learning for Channel Estimation: Map-based approaches

4.1 Introduction

A common assumption in the study of UAV communications is that the channel model parameters are assumed to be known when designing the UAV trajectory or finding the UAV optimal location. However, in reality, these parameters need to be learned based on the measurements collected from the ground users. As a result, an important question arises: What is an efficient way of collecting radio measurements by the UAV from the ground users in order to estimate the channel model parameters?

In this chapter, we formulate and solve *a learning* trajectory optimization problem in order to minimize the estimation error of the channel model parameters while capitalizing on 3D map data. The devised trajectory allows the UAV to exploit the map and quickly learn the propagation parameters.

We consider a wireless communication system where a UAV-mounted flying BS serving K static ground level nodes (IoT sensors, radio terminals, etc.) in an urban area. The location of the k-th ground node is assumed to be known and is denoted by $\mathbf{u}_k = [x_k, y_k, 0]^{\mathrm{T}} \in \mathbb{R}^3, k \in [1, K]$. The UAV's mission during learning the channel parameters is of duration T. For the ease of exposition, we assume that the time period T is discretized into N equal-time slots. Hence, the UAV's position $\mathbf{v}(t)$ is approximated by a sequence

$$\mathbf{v}[n] = [x[n], y[n], z[n]]^{\mathrm{T}}, \ n \in [1, N]. \tag{4.1}$$

The UAV locations is also assumed to be known at any time slots.

4.2 UAV Kinematic Model

During the mission, drone's position evolves according to

$$\mathbf{v}[n+1] = \mathbf{v}[n] + \begin{bmatrix} \cos(\phi[n])\cos(\psi[n]) \\ \sin(\phi[n])\cos(\psi[n]) \\ \sin(\psi[n]) \end{bmatrix} \rho[n]\,, \tag{4.2a}$$

$$h_{\min} \leq z[n] \leq h_{\max},\ \forall n \in [1, N-1], \tag{4.2b}$$

where in the n-th time slot, $0 \leq \rho[n] \leq \rho_{\max}$ represents the distance traveled by the drone, $0 \leq \phi[n] \leq 2\pi$ and $-\frac{\pi}{2} \leq \psi[n] \leq \frac{\pi}{2}$ represent the heading and elevation angles, respectively. The maximum distance traveled in a time slot is denoted by $\rho_{\max}$ and it depends on the UAV maximum velocity. The constraint (4.2b) reflects the fact that the drone always flies at an altitude higher than h_{min} and lower than $h_{\max}$, with $h_{\min}$ being the height of the tallest building in the city.

4.3 Learning Trajectory Design

In this section, our goal is to find a UAV trajectory, over which the channel measurements are collected from the ground nodes, that results in the minimum estimation error of the channel model parameters. While the problem of learning the channel parameters from a pre-determined measurement data set has been addressed in the prior literature [27; 56], the novelty of our work lies in the concept of optimizing the flight trajectory *itself* so as to accelerate the learning process. The channel measurement collection and learning process are described next.

4.3.1 Measurement Collection and Channel Learning

The measurement harvesting is performed over a UAV trajectory that starts at a base position $\mathbf{v}_{\mathrm{I}} \in \mathbb{R}^3$ and ends at a terminal position $\mathbf{v}_{\mathrm{F}} \in \mathbb{R}^3$. Mathematically,

$$\mathbf{v}[1] = \mathbf{v}_{\mathrm{I}},\ \mathbf{v}[N] = \mathbf{v}_{\mathrm{F}}. \tag{4.3}$$

The base position is typically the take-off base for the UAV while $\mathbf{v}_{\mathrm{F}}$ can be selected in different ways, including $\mathbf{v}_{\mathrm{F}} = \mathbf{v}_{\mathrm{I}}$ (loop). In the n-th time interval, $n \in [1, N]$, the measurements collected from the ground nodes can be written as

$$\mathbf{g}_{s,n} = \left[g_{s,1}, g_{s,2}, \cdots, g_{s,\delta_{s,n}}\right]^{\mathrm{T}},$$

where $g_{s,i}$, as defined in (2.2), is the channel gain (in dB scale) of the i-th measurement, $i \in [1, \delta_{s,n}]$, and $\delta_{s,n}$ is the number of measurements obtained for the propagation segment group $s \in \{\mathrm{LoS}, \mathrm{NLoS}\}$. For the LoS/NLoS classification of the measurements, we leverage the knowledge of a 3D city map [28]. Based on such map, we can predict LoS (un)availability on any given UAV-ground nodes link from a trivial geometry argument: For a given UAV position, the ground node is considered in LoS to the UAV if the straight line passing through the UAV's and the ground node's position lies higher than any buildings in between.

Using (2.2), the i-th measurement can be modeled as

$$g_{s,i} = \mathbf{a}_{s,i}^{\mathrm{T}}\, \boldsymbol{\omega}_s + \eta_{s,i}, \tag{4.4}$$

where $\mathbf{a}_{s,i} = [-\varphi(d_i),\, 1]^{\mathrm{T}}$ with d_i being the distance between the drone and ground node in the i-th measurement, $\boldsymbol{\omega}_s = [\alpha_s,\, ß_s]^{\mathrm{T}}$ is the vector of channel parameters, and $\eta_{s,i}$ denotes the shadowing component in the i-th measurement. The measurements collected in the n-th interval can now be written as

$$\mathbf{g}_{s,n} = \mathbf{A}_{s,n}\, \boldsymbol{\omega}_s + \boldsymbol{\eta}_{s,n}, \tag{4.5}$$

where $\mathbf{A}_{s,n} = \left[\mathbf{a}_{s,1}, \cdots, \mathbf{a}_{s,\delta_{s,n}}\right]^{\mathrm{T}}$, $\boldsymbol{\eta}_{s,n} = [\eta_{s,1}, \cdots, \eta_{s,\delta_{s,n}}]^{\mathrm{T}}$. Finally, we stack up the measurements gathered by the drone up to time step n as

$$\bar{\mathbf{g}}_{s,n} = \bar{\mathbf{A}}_{s,n}\, \boldsymbol{\omega}_s + \bar{\boldsymbol{\eta}}_{s,n}, \tag{4.6}$$

where

$$\begin{aligned}
\bar{\mathbf{g}}_{s,n} &= \left[\mathbf{g}_{s,1}^{\mathrm{T}}, \cdots, \mathbf{g}_{s,n}^{\mathrm{T}}\right]^{\mathrm{T}}, \\
\bar{\mathbf{A}}_{s,n} &= \left[\mathbf{A}_{s,1}^{\mathrm{T}}, \cdots, \mathbf{A}_{s,n}^{\mathrm{T}}\right]^{\mathrm{T}}, \\
\bar{\boldsymbol{\eta}}_{s,n} &= \left[\boldsymbol{\eta}_{s,1}^{\mathrm{T}}, \cdots, \boldsymbol{\eta}_{s,n}^{\mathrm{T}}\right]^{\mathrm{T}}.
\end{aligned}$$

Assuming that the measurements collected over a trajectory are independent, the maximum likelihood estimation (MLE) of $\boldsymbol{\omega}_s$, $s \in \{\mathrm{LoS}, \mathrm{NLoS}\}$ based on the measurements collected up to time step n is given by [27; 28]

$$\hat{\boldsymbol{\omega}}_{s,n} = \left(\bar{\mathbf{A}}_{s,n}^{\mathrm{T}}\, \bar{\mathbf{A}}_{s,n}\right)^{-1} \bar{\mathbf{A}}_{s,n}^{\mathrm{T}}\, \bar{\mathbf{g}}_{s,n}. \tag{4.7}$$

By substituting (4.6) in (4.7), we obtain

$$\hat{\boldsymbol{\omega}}_{s,n} - \boldsymbol{\omega}_s = \left(\bar{\mathbf{A}}_{s,n}^{\mathrm{T}}\, \bar{\mathbf{A}}_{s,n}\right)^{-1} \bar{\mathbf{A}}_{s,n}^{\mathrm{T}} \bar{\boldsymbol{\eta}}_{s,n}. \tag{4.8}$$

Since $\hat{\boldsymbol{\omega}}_{s,n}$ is unbiased, the mean squared error (MSE) of the estimated parameters can be obtained as [52]

$$\begin{aligned}
\mathrm{E}\left\|\hat{\boldsymbol{\omega}}_{s,n} - \boldsymbol{\omega}_s\right\|^2 &= \mathrm{tr}\left[Cov\left\{\hat{\boldsymbol{\omega}}_{s,n}\right\}\right] \\
&= \sigma_s^2\, \mathrm{tr}\left[\left(\bar{\mathbf{A}}_{s,n}^{\mathrm{T}}\, \bar{\mathbf{A}}_{s,n}\right)^{-1}\right].
\end{aligned} \tag{4.9}$$

Let

$$e_s[n] \triangleq \mathrm{tr}\left[\left(\bar{\mathbf{A}}_{s,n}^{\mathrm{T}}\, \bar{\mathbf{A}}_{s,n}\right)^{-1}\right],$$

and assuming that $\sigma_{\mathrm{NLoS}}^2 = \kappa \cdot \sigma_{\mathrm{LoS}}^2$, $\kappa \geq 1$ [55], the total estimation error in both propagation segments is given by

$$\sum_s \mathrm{E}\left\|\hat{\boldsymbol{\omega}}_{s,n} - \boldsymbol{\omega}_s\right\|^2 = \sigma_{\mathrm{LoS}}^2 \left(e_{\mathrm{LoS}}[N] + \kappa\, e_{\mathrm{NLoS}}[N]\right). \tag{4.10}$$

Note that a full rank $\bar{\mathbf{A}}_{s,n}$ is assumed in calculating the error for both LoS and NLoS categories over the course of the trajectory. If there are not enough measurements in a particular segment by the end of the trajectory, the estimation error is assigned as infinity in that segment.

4.3.2 Optimization Problem

The optimal learning trajectory that minimizes the estimation error can be formulated as

$$\min_{\Phi,\Psi,\mathcal{R}} \; e_{\mathrm{LoS}}[N] + \kappa\, e_{\mathrm{NLoS}}[N] \tag{4.11a}$$

$$\text{s.t. (4.2), (4.3)} \tag{4.11b}$$

where Φ, Ψ, and $\mathcal{R}$ are defined as

$$\begin{aligned}
\Phi &= \{0 \le \phi[n] < 2\pi,\, n \in [1, N-1]\}\,, \\
\Psi &= \left\{-\frac{\pi}{2} \le \psi[n] \le \frac{\pi}{2},\, n \in [1, N-1]\right\}, \\
\mathcal{R} &= \{0 \le \rho[n] \le \rho_{\max},\, n \in [1, N-1]\}\,.
\end{aligned}$$

As the estimation error depends on the matrix $\bar{\mathbf{A}}_{s,N}$ which has a very complicated expression in terms of $\phi[n], \psi[n]$, and $\rho[n]$, it is hard to obtain an analytical solution for problem (4.11) in general. Therefor, we tackle (4.11) by discretizing the optimization variables and then employing dynamic programming (DP) [57] to find the solution. To apply DP, the estimation error $e_s[N]$ needs to be rewritten as follows

$$\begin{aligned}
e_s[N] &= \mathrm{tr}\left[\left(\begin{bmatrix} \bar{\mathbf{A}}_{s,N-1} \\ \mathbf{A}_{s,N} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \bar{\mathbf{A}}_{s,N-1} \\ \mathbf{A}_{s,N} \end{bmatrix}\right)^{-1}\right] \\
&\overset{(a)}{=} e_s[N-1] - r_s[N] \\
&\overset{(b)}{=} e_s[1] - \sum_{n=2}^{N} r_s[n],
\end{aligned} \tag{4.12}$$

where we denote $r_s[n]$ as the amount of improvement in the estimate within time slot n, and it is given by

$$r_s[n] = \mathrm{tr}\left[\mathbf{H}_{s,n}\, \mathbf{A}_{s,n}^{\mathrm{T}} \left(\mathbf{I} + \mathbf{A}_{s,n}\, \mathbf{H}_{s,n}\, \mathbf{A}_{s,n}^{\mathrm{T}}\right)^{-1} \mathbf{A}_{s,n}\, \mathbf{H}_{s,n}\right], \tag{4.13}$$

$\mathbf{H}_{s,n} \triangleq \left(\bar{\mathbf{A}}_{s,n-1}^{\mathrm{T}}\, \bar{\mathbf{A}}_{s,n-1}\right)^{-1}$, $\mathbf{I}$ is the identity matrix, (a) follows from the matrix inversion lemma, and (b) follows from the recursive relation. Now (4.11) can be reformulated as

$$\min_{\Phi,\Psi,\mathcal{R}} \; \sum_{n=1}^{N} \tilde{e}_{\mathrm{LoS}}[n] + \kappa\, \tilde{e}_{\mathrm{NLoS}}[n] \tag{4.14a}$$

$$\text{s.t. (4.2), (4.3)} \tag{4.14b}$$

where

$$\tilde{e}_s[n] = \begin{cases} e_s[1] & n = 1 \\ -r_s[n] & n \in [2, N] \end{cases}.$$

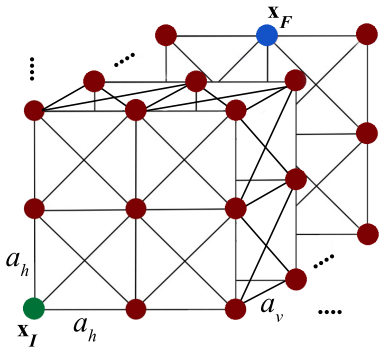


Figure 4.1 – A fragment of the 3D path graph, arbitrary base position $\mathbf{v}_{\mathrm{I}}$, and terminal position $\mathbf{v}_{\mathrm{F}}$.

4.3.3 Dynamic Programming

To solve (4.14) by DP, we constraint (and thus approximate) the possible drone locations and the optimization variables to a limited alphabet and then use Bellman's recursion to compute the optimal discrete trajectory. We start by introducing some notations.
Let $\mathbf{v}[n]$, $n \in [1, N]$ denotes the states and $\boldsymbol{\pi}[n] = [\phi[n], \psi[n], \rho[n]]^{\mathrm{T}}$ represents the input action at time $n \in [1, N-1]$ such that

$$\begin{aligned}
\phi[n] &\in \left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}\right\}, \\
\psi[n] &\in \left\{-\frac{\pi}{2}, -\frac{\pi}{4}, 0, \frac{\pi}{4}, \frac{\pi}{2}\right\}, \\
\rho[n] &\in \left\{0, a_h, a_v, a_h\sqrt{2}, \sqrt{a_h^2 + a_v^2}, \sqrt{2a_h^2 + a_v^2}\right\},
\end{aligned} \tag{4.15}$$

where a_h and a_v denote the discretization unit used in discretizing the city map into a 3D grid (hereafter termed as path graph) of admissible drone locations. Depending on the action $\boldsymbol{\pi}[n]$ in $\mathbf{v}[n]$, the state $\mathbf{v}[n+1]$ can be computed by using (4.2) and (4.15). In Fig. 4.1, a part of the path graph, arbitrary base position $\mathbf{v}_{\mathrm{I}}$, and terminal position $\mathbf{v}_{\mathrm{F}}$ are illustrated. The vertices and the edges of the path graph can, respectively, be interpreted as the admissible states and input actions in each time slot.

In order not to exceed the flight time constraint T, N can be selected as [1]

$$N = \left\lfloor \frac{T}{T_e} \right\rfloor,$$

where $\lfloor . \rfloor$ denotes the floor function and $T_e = \frac{\sqrt{2a_h^2 + a_v^2}}{v_{\max}}$ is the minimum required time for taking the longest edge between two adjacent vertices in the path graph while the drone moves with maximum speed $v_{\max}$.

DP in a forward manner is now used to solve for (4.14) by taking into account the finite alphabet constraint (4.15). Thus, by reformulating (4.14a) we can associate with

[1] Note that this is a conservative choice. In practice, N could be slightly higher given the UAV may use some of the short edges.

our problem the performance index

$$J_i(\mathbf{v}[i]) = \Omega(\mathbf{v}[1]) + \sum_{n=2}^{i} L[n], \tag{4.16}$$

where $[2, i]$ is the time interval of interest and $L[n] = \tilde{e}_{\mathrm{LoS}}[n] + \kappa\, \tilde{e}_{\mathrm{NLoS}}[n]$. $\Omega(\mathbf{v}[1])$ stands for the initial cost and given by

$$\Omega(\mathbf{v}[1]) = \begin{cases} L[1] & \mathbf{v}[1] = \mathbf{v}_{\mathrm{I}} \\ \infty & \text{otherwise} \end{cases}.$$

According to Bellman's equation, the optimal cost up to time $n+1$ is equal to

$$J^*_{n+1}(\mathbf{v}[n+1]) = \min_{\boldsymbol{\pi}[n]} \left\{ L[n+1] + J^*_n(\mathbf{v}[n]) \right\}, \; n \in [1, N-1], \tag{4.17a}$$

$$J^*_1(\mathbf{v}[1]) = \Omega(\mathbf{v}[1]), \tag{4.17b}$$

where $\boldsymbol{\pi}[n]$ is the input action vector. Thus, the optimal input action $\boldsymbol{\pi}^*[n]$ at time n is the one that achieves the minimum in (4.17a). Finally, the optimal policy (trajectory) can be found by solving (4.17) for all $n \in [1, N-1]$ and by choosing $\mathbf{v}[N] = \mathbf{v}_{\mathrm{F}}$. Note that the number of computations required to find the optimal trajectory is given by [57]

$$\mathcal{V} \cdot \Pi \cdot N,$$

where $\mathcal{V}$ is the number of admissible states (i.e. the number of vertices in the path graph), and Π is the number of quantized admissible input actions.

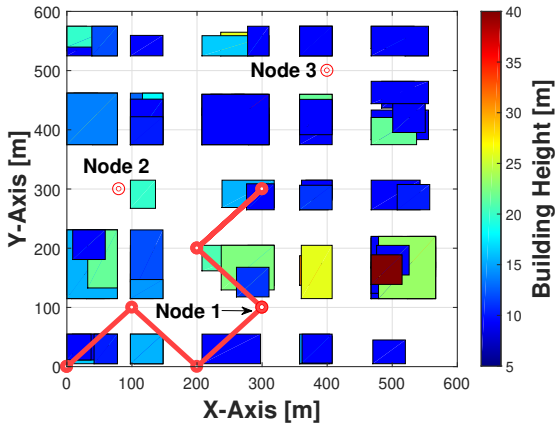
Note that the error $L[n]$ only depends on the UAV location through its distance from the ground users and the LoS/NLoS status. Since we have the knowledge of the 3D map and the ground nodes' locations, (4.17) can be solved offline without collecting any measurements. Once the optimal trajectory is calculated, UAV follows this trajectory to collect the measurements and then estimates the channel parameters.

4.4 Numerical Results

We consider a dense urban Manhattan-like area of size 600×600 square meters, consisting of a regular street and buildings. The height of the building is Rayleigh distributed within the range of 5 to 40 meters [15]. The average building height is 14 m. True propagation parameters are chosen as $\alpha_{\mathrm{LoS}} = 2.27$, $\alpha_{\mathrm{NLoS}} = 3.64$, ß$_{\mathrm{LoS}} = -30\,\mathrm{dB}$, ß$_{\mathrm{NLoS}} = -40\,\mathrm{dB}$ according to an urban micro scenario in [55]. The variances of the shadowing component in LoS and NLoS scenarios are $\sigma^2_{\mathrm{LoS}} = 2$ and $\sigma^2_{\mathrm{NLoS}} = 5$, respectively. The UAV has a maximum speed of $v_{\max} = 10\,\mathrm{m/s}$.

An illustration of the optimal learning trajectory is presented in Fig. 4.2 for $K = 3$ ground nodes. In this scenario, the UAV flies from the base position $\mathbf{v}_{\mathrm{I}} = [0, 0, 50]^{\mathrm{T}}$ meters towards the terminal location $\mathbf{v}_{\mathrm{F}} = [300, 300, 50]^{\mathrm{T}}$ meters under the flight time constraint $T = 100\,\mathrm{s}$. To discretize the search space over the city for creating the 3D path

(a)



(b)

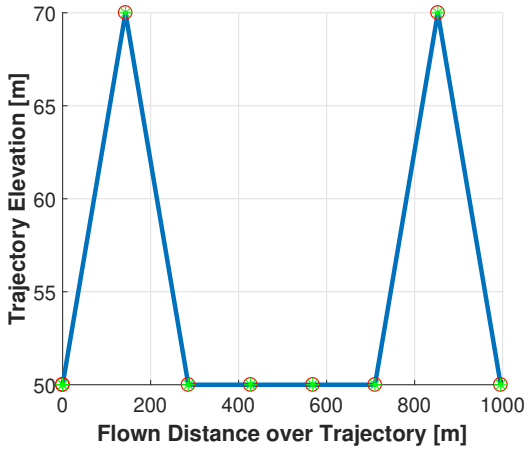


Figure 4.2 – (a) Top view of the optimal learning trajectory using proposed algorithm. (b) The UAV elevation along the trajectory.

graph, we chose $a_h = 100$ m and $a_v = 20$ m as defined in Section 4.3.3. It is interesting to note that, the trajectory experiences a wide array of altitudes there by improving the learning performance of both LoS and NLoS segments. For the ease of exhibition, we plotted the generated trajectory in two different figures. Fig. 4.2-a shows the top view of the generated trajectory while the elevation of the trajectory as a function of the flown distance is depicted in Fig. 4.2-b.

In Fig. 4.3 the performance of the optimal trajectory in terms of the MSE of the learned channel parameters $(\alpha_s, ß_s; s = \{\text{LoS}, \text{NLoS}\})$ is shown as a function of the number of ground nodes. The duration of the learning phase $T = 100$ s. We perform Monte-Carlo simulations over random user locations. We also compare the performance of the optimal trajectory with that of randomly generated arbitrary trajectories. For a given realization, arbitrary trajectory of duration T is generated. It is clear that, the

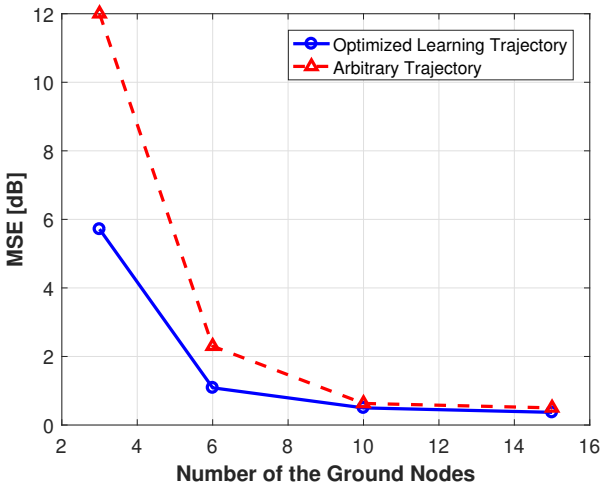


Figure 4.3 – Comparison of the MSE for different learning trajectories.

channel can be learned more precisely by taking the optimized learning trajectory. Also, the learning error is reduced when the number of ground nodes increases, since the chance of obtaining measurement from both LoS and NLoS segments increases.

4.5 Conclusion

In this chapter, we investigated the problem of learning trajectory optimization for a UAV BS which is serving a set of ground nodes in order to minimize the estimation error of the channel model parameters. To solve this problem we proposed an algorithm to design the learning trajectory using the active learning and by leveraging the 3D map of the environment. The devised trajectory allows the UAV to exploit the map and quickly learn the propagation parameters.

Chapter 5

UAV-aided Radio Node Localization

One of the critical components in UAV communication systems is the location awareness of the nodes in wireless networks. When there is no provision for nodes communicating their GPS coordinates or when the GPS accuracy is limited in challenging environments such as indoors or dense urban cities, network localization becomes a challenging problem which has attracted significant interest during the past few years [58; 59].

In a typical network, there are two types of nodes, namely anchors and users. As elements of the network infrastructure, anchors have perfectly known positions, which can be stationary or time-varying like in the case of flying BSs or relays, whereas users often have unknown locations (and need to be estimated). Network node localization can follow a two step approach (i) distance estimation between the nodes in the network and (ii) node localization based on inter-node distances. Based on the distance estimation techniques employed, localization algorithms can be categorized into range-based localization algorithms and range-free localization approaches. In the range-based localization, the distances between user nodes are estimated by exploiting some physical properties of communication signals (e.g. the relation between the distance and received signal strength), while the range-free localization algorithms estimate a user coordinate using connectivity information between users without ranging information (i.e. no time, angle or power information is used). As an example for the range-free localization, as a rough estimation, the location of a node can be determined as the center of gravity of all the anchor nodes which it can connect to (e.g. if the user is within the coverage area of the anchor node) [60].

When it comes to range-based localization algorithms, different methods can be utilized to measure the distance between the nodes such as RSSI, time of arrival (ToA), time difference of arrival (TDoA), and angle of arrival (AoA). The RSSI information infers the distance between the agent nodes and the anchor nodes from the fact that the attenuation of the radio signal increases as the distance between the receiver and transmitter increases. In ToA, the timing information or the propagation time of the communication signal is used to measure the distance between user nodes and the anchor node. The distance between two nodes is calculated using the TDoA, and the AoA

methods use the angle at which signals are received at the anchor nodes in some reference frame.

Among all the aforementioned methods, localization using RSSI measurements offers an attractive solution for practical purposes as the RSSI measurements are easy to obtain in many wireless networks and are inexpensive unlike the timing-based methods which require tight synchronization and calibration of the transceivers [61; 62; 63].

The problem of network localization based on RSSI measurements obtained by static anchor nodes and unknown propagation pathloss model parameters has been studied in many papers, an early contribution to the problem was [64]. However, the authors in [64] use a pathloss model that does not differentiate between LoS and NLoS conditions. Since we are interested in using a flying UAV as a mobile anchor to localize the ground users in an urban environment, the air-to-ground propagation models between the UAV and ground users will much benefit from a segmented pathloss model where the propagation condition are allowed to switch between LoS and NLoS depending on the UAV location [7; 27]. The authors in [65] extended the work of [64] to a segmented pathloss model which is suited to urban environments having LoS and NLoS conditions. They model the collected RSSI measurements as a mixture of two Gaussian distributions with weights representing the probability of the measurement obtained from the LoS or NLoS segment. The problem of joint user localization and pathloss parameter estimation is then solved by an iterative algorithm using the expectation maximization (EM) criterion. Note that [64] and [65] do not use any side information in the form of 3D building maps. Hence a statistical path loss model that exploits probabilities of being in LoS or NLoS modes has to be used, as in [65]. If a 3D city map containing the building structure information is given as extra information to the localization problem, then the LoS/NLoS classification task of a measurement can be improved as we observe in this thesis. Indeed, it has been observed in [66] that building map-aware statistical models for RSSI measurements in an indoor environment for localization leads to a better performance than the map-unaware methods.

Instead of using different static anchor nodes, a mobile anchor equipped with a GPS device in the form of a UAV can be utilized to collect the radio measurements from nodes at different geographical locations. While some amount of research has been done on the problem of ground-based mobile anchor node assisted localization techniques [61], using UAVs as aerial anchors to localize the ground users has also gained interest. The advantage of UAV anchors is the flexibility in moving in 3D, hence gaining the ability to favorably impact fading statistics by increasing the probability of LoS conditions which are known to be beneficial for achieving better localization performance [67; 68].

Using UAV anchors for network localization offers another degree of freedom in terms of their trajectory, which can be optimized for accelerated learning of the node locations. In [69; 70] a simple trajectory is generated by taking the effect of the NLoS links between the anchor nodes and the users into consideration. The authors in [71] focused on the experimental exploration of static trajectories applied to the localization of wireless nodes using UAVs. They investigated two types of trajectories called Triangle and Circle, and showed that the Triangle is better suited for their unique indoor-outdoor setting. In [72; 73; 74] the problem of trajectory design of multiple flying anchors (UAVs) for

RSSI-based node localization is investigated. Authors in [72; 73; 74] assumed that the UAVs fly high enough so that they can establish LoS links to any nodes in the network at all times, and the effect of NLoS measurements is ignored in the localization process which is not very realistic for mid-altitude UAVs. In [75], the trajectories for multiple UAVs are designed using the ToA estimate.

In this chapter, we focus on the range-based localization by using the RSSI measurements collected by a flying anchor. This is up to our knowledge the first work dealing with UAV-based user localization which also optimizes the flight trajectory under a fixed mission time while exploiting a realistic segmented channel model fed from a 3D map. Specifically, our contributions are as follows:

- We formulate and solve the problem of joint map-based channel learning and user localization from the RSSI measurements collected from the users in both LoS and NLoS conditions by the UAV. The LoS and NLoS classification information is derived by confronting location estimates with the 3D map information.

- We propose an on-line algorithm to design an optimized trajectory for the UAV to improve the performance of the localization and channel learning under a given mission duration. The intuition behind it is that not all measurement points in the network are equally worthy so as to bring information about node location. An optimized trajectory will minimize flying time while making sure the collected data is as informative as possible.

5.1 System Model

We consider a scenario where a UAV-mounted access point is connected to K ground level users in an urban area consisting of a number of city buildings. The users are scattered all over the city and $\mathbf{u}_k = [x_k, y_k]^{\mathrm{T}} \in \mathbb{R}^2$, $k \in [1, K]$ denotes the k-th user's location which is not known. The user locations are considered static for the sake of derivation of the algorithm, yet in practice, adaptive/iterative versions of the proposed methods are possible to obtain that will track moving users. The UAV's mission is of the duration T, during which, the aim of the UAV is to estimate the unknown user locations by collecting radio measurements from them. A two-segment channel model (LoS/NLoS) is used as introduced in Section 2.2.1. Note that, in this chapter, we assume the channel model parameters are unknown at the initial stage of the algorithm. Importantly, a 3D map of the city is assumed to be available.

For ease of exposition, we assume that the time period $[0, T]$ is divided into N equal length intervals of duration $\delta = T/N$, indexed by $n = 1, \ldots, N$. The value of δ is chosen to be sufficiently small such that UAV's location, velocity, and heading angles can be considered constant in one interval. In the n-th interval the UAV/drone position is denoted by $\mathbf{v}[n] = [x[n], y[n], z[n]]^{\mathrm{T}} \in \mathbb{R}^3$. We assume that the drone is equipped with a GPS receiver, hence $\mathbf{v}[n]$ is known. Moreover during the mission, the UAV's position evolves according to the UAV model mentioned in Section 4.2.

5.2 User Localization and Channel Model Learning

In this section, we propose a map based algorithm to simultaneously localize users and learn the channel model's pathloss parameters by employing the particle swarm optimization (PSO) technique. The motivations to use a PSO-based approach are exposed in more details in the next section. The measurement collection process and then the estimation problem are described next. Note that in the initial description stage, we assume the UAV describes an arbitrary trajectory of given length. The trajectory is optimized in the later sections of this chapter. The knowledge of the 3D map of the environment plays a crucial role in the estimation process. Note that in this chapter, we assume the values of the shadowing effect covariance pertaining to both LoS and NLoS segments ($\sigma_s^2, s \in \{\text{LoS}, \text{NLoS}\}$) are known.

Let an arbitrary trajectory be taken by the UAV during the mission such that it flies over N different locations denoted by $\{\mathbf{v}[1], \ldots, \mathbf{v}[n]\}$. In each of these locations, the UAV collects the radio measurements form all K users (in total $K \cdot N$ measurements). Let $g_{n,k}$ represent the RSSI measurement (in dB scale) obtained from the k-th user by the UAV in the n-th interval. Using (2.2), $g_{n,k}$ is modeled as

$$g_{n,k} = w_{n,k}\, h_{n,k,\text{LoS}} + (1 - w_{n,k})\, h_{n,k,\text{NLoS}} \tag{5.1}$$

where $\omega_{n,k} \in \{0, 1\}$ is the classifier binary variable (yet unknown) which indicates if a measurement falls into the LoS or NLoS category, and $h_{n,k,s}$ is a model for a RSSI measurement collected form user k at the n-th time step which belongs to the segment $s \in \{\text{LoS}, \text{NLoS}\}$ and is defined as

$$h_{n,k,s} = \lambda_n(\boldsymbol{\theta}_s, \mathbf{u_k}) + \eta_{n,k,s}. \tag{5.2}$$

where $\boldsymbol{\theta}_s = [\alpha_s, ß_s]^{\text{T}}, s \in \{\text{LoS}, \text{NLoS}\}$, and

$$\lambda_n(\boldsymbol{\theta}_s, \mathbf{u_k}) = ß_s - \alpha_s \varphi_n(\mathbf{u_k}), \tag{5.3}$$

$\varphi_n(\mathbf{u_k}) = 10 \log_{10}(d_n(\mathbf{u_k}))$, and $d_n(\mathbf{u_k}) = \|\mathbf{u_k} - \mathbf{v}[\mathbf{n}]\|$ being the distance between the k-th user and the UAV in its n-th location. $\eta_{n,k,s}$ represents a random shadowing component with zero-mean Gaussian distribution with variance σ_s^2. We assume that NLoS measurements have a higher shadowing affect ($\sigma_{\text{NLoS}} \geq \sigma_{\text{LoS}}$) [55].

Assuming that collected measurements conditioned on channel parameters [1] and user positions are i.i.d, then the MLE of $\boldsymbol{\theta}_s, \mathbf{u_k}$ leads to minimizing the sum squared error (SSE),

$$\text{SSE} = \sum_{k=1}^{K} \sum_{n=1}^{N} \omega_{n,k} \left| g_{n,k} - \lambda_n(\boldsymbol{\theta}_{\text{LoS}}, \mathbf{u}_k) \right|^2 + (1 - \omega_{n,k}) \left| g_{n,k} - \lambda_n(\boldsymbol{\theta}_{\text{NLoS}}, \mathbf{u}_k) \right|^2. \tag{5.4}$$

The optimal channel parameters and user location estimates that minimize the SSE in (5.4) can be formulated as

$$\min_{\omega_{n,k}, \boldsymbol{\theta}_s, \mathbf{u}_k} \text{SSE}. \tag{5.5}$$

[1] This amounts to assuming the shadowing coefficients are independent over successive UAV locations, which is a classical simplifying assumption, see for e.g. [27]

As the estimation error in (5.4) comprises the classifier binary variables $\omega_{n,k}$ and $\lambda_n(\boldsymbol{\theta}_s, \mathbf{u}_k)$ is a non-linear function of the parameter $\mathbf{u}_k$, in general it is challenging to solve the simultaneous classification, localization and channel learning problem (5.5). To tackle this problem, the PSO technique is used, aided with the exploitation of the 3D map of the environment. PSO is a computational technique which iteratively explores the search space of a given problem to find the settings or parameters required to maximize a particular objective. PSO works by simultaneously maintaining several candidate solutions in the search space, hence helping to prevent from getting stuck in the local minima which often arises in non-convex optimization. No matter the complexity of the objective function, PSO is found from experience to often find a good solution in such problems [76; 77]. In the following sections, we first briefly explain the PSO algorithm and then we propose a solution to efficiently solve problem (5.5).

5.2.1 PSO Techniques

PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality so called fitness value. PSO solves a problem by first generating a population of random candidate solutions, called particles, and moving these particles around in the search-space based on a simple mathematical formulae capturing each particle's position and velocity, as described below. During each iteration, each candidate solution is evaluated by the objective function, determining the fitness of that solution.

We consider a set of particles $\mathcal{C} = \{\mathbf{c}_j \in \mathbb{R}^D, j = 1, \cdots, C\}$, where D is the dimension of each particle. Each particle has the same dimension as unknown parameters to be estimated. The fitness and the velocity of each particle need to be stored. Additionally, each particle remembers the best fitness value it has achieved so far during the operation of the algorithm, referred to as the *individual best fitness*, as well as the candidate solution that achieved this fitness, referred to as the *individual best candidate solution*. Finally, the PSO algorithm maintains the best fitness value achieved among all particles in the population, called the *global best fitness*, as well as the candidate solution that achieved this fitness, called the *global best candidate solution*.

The velocity of each particle in the swarm is updated using the following equation:

$$\dot{\mathbf{c}}_j^{(i+1)} = \nu\, \dot{\mathbf{c}}_j^{(i)} + \varepsilon_1 r_1 (\mathbf{c}_j^{\mathrm{b}} - \mathbf{c}_j^{(i)}) + \varepsilon_2 r_2 (\mathbf{c}^* - \mathbf{c}_j^{(i)}), \tag{5.6}$$

where $\mathbf{c}_j^{(i)}, \dot{\mathbf{c}}_j^{(i)}$ are the j-th particle's position and its velocity at the i-th iteration of PSO, respectively. $\mathbf{c}_j^{\mathrm{b}}$ is the individual best candidate solution for particle j at iteration i, and $\mathbf{c}^*$ is the the swarm's global best candidate solution. The parameters $\nu, \varepsilon_1, \varepsilon_2$ are user-supplied coefficients. ν is an inertial coefficient which can either dampen the particle's inertia or accelerate the particle in its original direction. ε_1 is called the cognitive coefficient which affects the size of the step the particle takes toward its individual best candidate solution, and ε_2 is the social coefficient which modulates the step the particle takes toward the global best candidate solution so far. The values $r_1, r_2\, (0 \leq r_1 \leq 1,\, 0 \leq r_2 \leq 1)$ are random values regenerated from a uniform distribution

for each velocity update. Once the velocity for each particle is calculated, each particle's position is updated by applying the new velocity to the particle's previous position:

$$\mathbf{c}_j^{(i+1)} = \mathbf{c}_j^{(i)} + \dot{\mathbf{c}}_j^{(i+1)}. \tag{5.7}$$

This process is repeated until some stopping condition is met (e.g. a preset number of iterations of the PSO algorithm, a predefined target fitness value). Since, PSO algorithm works by jointly maintaining several candidate solutions in the search area, hence it is less likely to be trapped into a local minimum compared to the gradient-based optimizer. Due to the fact that problem (5.5) is highly non-linear, PSO is highly suitable in our case.

As mentioned earlier, each particle has the same dimension as the unknown vectors, which in our case, are the channel parameters for both LoS and NLoS segments, the classification variables, and the user locations. Then, each particle is consisting of the following elements

$$\mathbf{c}_j = [w_{1,1}, \cdots, w_{N,K}, \mathbf{u}_1^{\mathrm{T}}, \cdots, \mathbf{u}_K^{\mathrm{T}}, \boldsymbol{\theta}_{\mathrm{LoS}}^{\mathrm{T}}, \boldsymbol{\theta}_{\mathrm{NLoS}}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{(KN+2K+4)}, \forall j \tag{5.8}$$

where KN stands for the classification variables, $2K$ indicates the users' locations in two dimensions, to this is added four unknown variables which are $\alpha_s, ß_s, s \in \{\mathrm{LoS}, \mathrm{NLoS}\}$. Hence in general, solving (5.5) even with a PSO approach will be challenging due to the high dimensionality of the problem.

To deal with this problem, we propose to use the following trick. Assume a 3D map of the environment is available as side information. This map helps to directly relate the user location to the propagation classification variables, hence the latter variables can be removed from the particle. This dramatically reduces the complexity of the problem by compacting the dimension of particles to $\mathbf{c}_j \in \mathbb{R}^{2K}$ as follows

$$\mathbf{c}_j = [\mathbf{u}_1^{\mathrm{T}}, \cdots, \mathbf{u}_K^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{2K}, \forall j \tag{5.9}$$

More specifically, propagation segment classification variables can be directly inferred from the user and UAV locations from a trivial geometry argument: For a given UAV position, the user is considered in LoS to the UAV if the straight line passing through the UAV's and the user position lies higher than any buildings in between. In turn, having classified each measurement into LoS or NLoS, then the channel parameters can be learned easily as well from the measurements by resorting to a standard least square (LS) technique (see Section 4.3.1). As a result, the channel parameters estimates also need not be included into the particles. In the following, we first consider the single user case and then we proceed to the multi-user scenario.

We denote the individual best fitness values by $F_j, j \in [1, C]$, and the global best fitness value by F^*. both individual best fitness and global best fitness values are initialized by a positive large number (e.g. $+\infty$).

5.2.2 Single User Case

In a single user case ($K = 1$), each particle has a dimension of $\mathbf{c}_j := \mathbf{u}_1 \in \mathbb{R}^2$ (i.e. each particle is a potential candidate for the user location). Therefore, the SSE for a given

particle, which is an estimate of the user location, is given by:

$$\mathrm{SSE}(\mathbf{c}_j^{(i)}) = \sum_{s\in\{\mathrm{LoS},\mathrm{NLoS}\}} \sum_{n\in\mathcal{M}_{s,1,j}} \left| g_{n,1} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{c}_j^{(i)}) \right|^2, \tag{5.10}$$

where $\mathbf{c}_j^{(i)}$ is the j-th particle at the i-th iteration of the PSO algorithm, and $\mathcal{M}_{s,1,j}$ is a set of time indices of measurements collected from user 1 which are at segment s by assuming that the user 1 location is same as particle j. To form $\mathcal{M}_{s,1,j}$, a 3D map of the city is utilized. For example, measurement $g_{n,1}$ is considered LoS, if the straight line passing through $\mathbf{c}_j^{(i)}$ and the drone location $\mathbf{v}[n]$ lies higher than any buildings in between.

Having formed $\mathcal{M}_{s,1,j}$, for the given particle $\mathbf{c}_j^{(i)}$, (5.10) can be minimized just by optimizing over the channel parameters $\boldsymbol{\theta}_s$ as follows

$$\mathrm{SSE}^*(\mathbf{c}_j^{(i)}) := \min_{\boldsymbol{\theta}_{\mathrm{LoS}},\boldsymbol{\theta}_{\mathrm{NLoS}}} \mathrm{SSE}(\mathbf{c}_j^{(i)}). \tag{5.11}$$

(5.11) is an LS problem and can be easily solved (see Section 4.3.1). We indicate the channel parameters found by solving (5.11) for the given particle $\mathbf{c}_j^{(i)}$ as $\hat{\boldsymbol{\theta}}_{s,j}$. We denote the index of the best particle minimizing (5.10) as

$$j^* := \arg\min_{j\in[1,C]} \mathrm{SSE}^*(\mathbf{c}_j^{(i)}). \tag{5.12}$$

Consequently, the individual best fitness and global best fitness values are updated as follows

$$\begin{aligned} F_j &:= \min\left(\mathrm{SSE}^*(\mathbf{c}_j^{(i)}),\, F_j\right),\ j\in[1,C], \\ F^* &:= \min\left(\mathrm{SSE}^*(\mathbf{c}_{j^*}^{i}),\, F^*\right). \end{aligned} \tag{5.13}$$

$\mathbf{c}_j^{\mathrm{b}}, \mathbf{c}^*$ are also updated accordingly. In addition, We denote the learned channel parameters corresponding to $\mathbf{c}^*$ by $\boldsymbol{\theta}^*_{\mathrm{LoS}}, \boldsymbol{\theta}^*_{\mathrm{NLoS}}$. The PSO algorithm then proceeds to the next iteration by updating the particles. We assume that the PSO terminates after I iterations.

In accordance with (5.13), F^* is non-increasing at each iteration, therefore the original problem (5.5) can be upper-bounded as:

$$\mathrm{SSE}^* \le F^*, \tag{5.14}$$

where SSE^* is the global minimum value of (5.5). By assuming a large number of particles, SSE^* can be approximated by the upper-bound as follows

$$\mathrm{SSE}^* \approx F^*. \tag{5.15}$$

Finally, $\mathbf{c}^*$, $\boldsymbol{\theta}^*_{\mathrm{LoS}}, \boldsymbol{\theta}^*_{\mathrm{NLoS}}$ are considered as the user location and channel parameters estimates, respectively, for problem (5.5).

5.2.3 Multi User Case

Now we proceed to the multi user case. Solving problem (5.5) with PSO algorithm for more than 1 user, even by exploiting the 3D map information, is challenging, since for any possible combination of particles, an LS problem need to be solved for finding channel parameters estimates, which is computationally complex (i.e. the complexity of the problem exponentially increases with the number of users). To tackle the complexity, we employ a block coordinate descent technique which tries to solve the original problem iteratively and at each iteration, only one set of variables is updated (while fixing all the other variables), rather than updating all the variables together. More precisely, at each iteration we fix all users location estimates except one, and we solve problem (5.5) for that particular user. At the same time, at each iteration, the channel parameters are also estimated. Note that, at each iteration of the algorithm, the problem becomes a single user case which has been addressed earlier. In this manner, the complexity of the algorithm linearly increases with the number of users. The proof of convergence and more details of the iterative algorithm is provided in Appendix B.1.

5.3 Trajectory Design for Accelerated Learning

When it comes to the user localization using a mobile anchor, one of the fundamental issues is how to optimally gather measurement from the users, i.e. measurements that are maximally informative about the parameters that we seek to estimate over the UAV mission time. In control and learning, such optimization framework is often termed as optimal design of experiments [78; 79]. The relevance of this problem to our localization scenario can be understood as follows: The measurements collected from NLoS links usually lead to a degradation of the localization accuracy due to the higher shadowing effect of the NLoS channel. However, designing a trajectory for the drone to establish LoS links to all users at all times is not a viable solution because there may not exist a continuous trajectory which fulfills this constraints and because of the limited mission time. Therefore, having an autonomous trajectory design algorithm that strikes a balance between collecting LoS measurements from the users and the mission time is of the essence here. As an approach to this problem, we investigate the problem of trajectory optimization of a UAV that somehow minimizes the user localization error. Our approach relies on the notion of Fisher information matrix (FIM) [80; 81]. This matrix helps measure the degree of dependence of observed random measurements on the user locations, hence it can provide information on the best accuracy with which such location can be estimated, for any given trajectory. Our goal is to exploit structural properties of the FIM so as to design an optimal policy for the drone to collect the best possible measurements from users. In the following, we first elaborate on the FIM and its properties and then optimize the UAV's trajectory.

5.3.1 Fisher Information Matrix

For a set of RSSI measurements obtained from segment $s \in \{\text{LoS}, \text{NLoS}\}$, the FIM of the measurements with respect to the variables we seek to estimate is given by

$$\mathbf{F}_s = \mathrm{E}\left[\frac{\partial \mathcal{L}_s}{\partial \phi_s}\frac{\partial \mathcal{L}_s}{\partial \phi_s}^{\mathrm{T}}\right], \tag{5.16}$$

where $\phi_s = [\alpha_s, ß_s, x_1, y_1, \cdots, x_K, y_K]^{\mathrm{T}}$. $\mathcal{L}_s$ is the likelihood of the measurements collected from segment s and is defined as

$$\mathcal{L}_s = \sum_{n=1}^{N} \sum_{k \in \mathcal{K}_{n,s}} \log\, f_{n,k,s}, \tag{5.17}$$

where $f_{n,k,s}$ is the probability density function (PDF) of the n-th measurement from user k in segment s, and $\mathcal{K}_{n,s}$ is a set of user indices that are in segment s at time step n. Then the FIM for all measurements in segment $s \in \{\text{LoS}, \text{NLoS}\}$ up to time step N is given by

$$\begin{aligned} \mathbf{F}_{N,s} &= \sum_{n=1}^{N} \sum_{k \in \mathcal{K}_{n,s}} \mathbf{H}_{n,k,s} \\ &= \mathbf{F}_{N-1,s} + \sum_{k \in \mathcal{K}_{n,s}} \mathbf{H}_{N,k,s}, \end{aligned} \tag{5.18}$$

where $\mathbf{H}_{n,k,s}$ is derived in Appendix B.2. Note that, (5.18) implies that the FIM is cumulative over the time.

5.3.2 Cramér-Rao Bound Analysis

According to the Cramér-Rao bound (CRB) [82], the MSE of the estimated parameters $\hat{\phi}_s$ is lower bounded by

$$\mathrm{MSE}(\hat{\phi}_s) \geq \mathrm{tr}(\mathbf{F}_{N,s}^{-1}). \tag{5.19}$$

Considering the cumulative property of $\mathbf{F}_{N,s}$, we can write

$$\begin{aligned} \mathbf{F}_{N,s}^{-1} &= \left[\mathbf{F}_{N-1,s} + \sum_{k \in \mathcal{K}_{n,s}} \mathbf{H}_{N,k,s}\right]^{-1} \\ &\overset{(a)}{=} \mathbf{F}_{N-1,s}^{-1} - \mathbf{R}_{N,s} \\ &\overset{(b)}{=} \mathbf{F}_{1,s}^{-1} - \sum_{n=2}^{N} \mathbf{R}_{n,s}, \end{aligned} \tag{5.20}$$

where (a) follows from the matrix inversion lemma, and (b) follows from the recursive relation. We denote $\mathbf{R}_{n,s}$ as the improvement in the estimate within time slot n and is given by

$$\mathbf{R}_{n,s} = \mathbf{F}_{n-1,s}^{-1}\left(\sum_{k \in \mathcal{K}_{n,s}} \mathbf{H}_{n,k,s}^{-1} + \mathbf{F}_{n-1,s}^{-1}\right)^{-1} \mathbf{F}_{n-1,s}^{-1}. \tag{5.21}$$

5.3.3 Trajectory Optimization

We are interested to find a trajectory for the drone under a total flight time constraint which starts from the base point $\mathbf{v}_{\mathrm{I}}$ and ends up at the terminal point $\mathbf{v}_{\mathrm{F}}$ and which minimizes the estimation error of parameters ϕ_s as follows

$$\min_{\chi} \quad \mathrm{MSE}(\hat{\phi}_{\mathrm{LoS}}) + \mathrm{MSE}(\hat{\phi}_{\mathrm{NLoS}}) \tag{5.22a}$$

$$\text{s.t.} \quad T_{\mathrm{F}} \leq T, \tag{5.22b}$$

$$\mathbf{v}[1] = \mathbf{v}_{\mathrm{I}},\, \mathbf{v}[N] = \mathbf{v}_{\mathrm{F}}, \tag{5.22c}$$

where $\chi = \{\mathbf{v}[n],\, n \in [1, N]\}$ denotes the discretized trajectory set in 3D, and $\mathrm{MSE}(\hat{\phi}_s)$ is the MSE of the estimated parameters $\hat{\phi}_s, s \in \{\mathrm{LoS}, \mathrm{NLoS}\}$. The drone's flying time is denoted by T_{F} and starts when the drone leaves the base position $\mathbf{v}_{\mathrm{I}}$. By assuming a constant velocity for the drone as $v_{\max}$, then T_{F} is given by

$$T_{\mathrm{F}} = \frac{1}{v_{\max}} \sum_{n \in [2,N]} \|\mathbf{v}[n] - \mathbf{v}[n-1]\|. \tag{5.23}$$

Problem (5.22) is challenging to solve since obtaining a closed form expression for $\mathrm{MSE}(\hat{\phi}_s), s \in \{\mathrm{LoS}, \mathrm{NLoS}\}$ is not easy. Therefore, instead of solving (5.22), we find a trajectory to minimize the CRB. We can then write

$$\min_{\chi} \quad \mathrm{tr}(\mathbf{F}^{-1}_{N,\mathrm{LoS}} + \mathbf{F}^{-1}_{N,\mathrm{NLoS}}) \tag{5.24a}$$

$$\text{s.t.} \quad (5.22\mathrm{b}), (5.22\mathrm{c}). \tag{5.24b}$$

By substituting (5.20) in (5.24a) we obtain

$$\min_{\chi} \quad \sum_{n=1}^{N} e_{\mathrm{LoS}}[n] + e_{\mathrm{NLoS}}[n] \tag{5.25a}$$

$$\text{s.t.} \quad \frac{1}{v_{\max}} \sum_{n \in [2,N]} \|\mathbf{v}[n] - \mathbf{v}[n-1]\| \leq T, \tag{5.25b}$$

$$(5.22\mathrm{c}), \tag{5.25c}$$

where

$$e_s[n] = \begin{cases} \mathrm{tr}(\mathbf{F}^{-1}_{1,s}), & n = 1 \\ -\mathrm{tr}(\mathbf{R}_{n,s}), & n \in [2, N]. \end{cases} \tag{5.26}$$

As $e_s[n]$ has a complicated expression in terms of the drone location, it is hard to obtain an analytical solution for problem (5.25). To tackle this problem, dynamic programming [57] can be applied by discretizing the search space comprising discrete states (i.e. each state is a feasible discrete location in the 3D space where the drone can travel to).

Note that $\mathbf{F}_{N,s}$ highly depends on the channel parameters and user locations. Since the true values of these parameters are not available, then the estimate of channel parameters and user locations can be used to compute $\mathbf{F}_{N,s}$. In this case, (5.25) becomes as an online

learning and trajectory design problem in which the UAV seeks to intelligently collect measurements from users in order to improve the performance of the estimation. Since, the performance of the learning highly depends on the collected measurements, therefore, the UAV needs to actively optimize its trajectory over the time to maximize the learning performance. In this manner, after obtaining new measurements, unknown parameter estimates are updated, and accordingly, a new trajectory is generated. For example, in the first time step ($n = 1$), for the lack of sufficient collected measurements from users, channel parameters and user locations estimates are initialized randomly. Then, $\mathbf{F}_{N,s}$ is computed based on the initial values and a trajectory is designed by solving (5.25). After that, the UAV collects new measurements by following the devised trajectory in the next time step ($n = 2$). Having collected new measurements, user location estimates, channel parameters, and the matrix $\mathbf{F}_{N,s}$ are updated. Consequently, a new trajectory has to be generated by taking into account the UAV flying time constraint. This procedure is repeated until the UAV reaches the terminal point. Mathematically speaking, the trajectory design problem in the n-th time step can be reformulated as follows

$$\min_{\chi_{n:N}} \quad \mathrm{tr}(\hat{\mathbf{F}}^{-1}_{N,n,\mathrm{LoS}} + \hat{\mathbf{F}}^{-1}_{N,n,\mathrm{NLoS}}) \tag{5.27a}$$

$$\text{s.t.} \quad \mathbf{v}[n] = \mathbf{x}_n,\, \mathbf{v}[N] = \mathbf{v}_{\mathrm{F}}, \tag{5.27b}$$

$$(5.25\mathrm{b}), \tag{5.27c}$$

where $\chi_{n:N} = \{\mathbf{v}[i], i \in [n, N]\}$, and $\hat{\mathbf{F}}_{N,n,s}$ is the estimation of $\mathbf{F}_{N,s}$ given the learned channel parameters and user location estimates up to time step n, and

$$\mathbf{x}_n = \begin{cases} \mathbf{v}_{\mathrm{I}}, & n = 1 \\ \mathbf{v}[n], & n > 1. \end{cases} \tag{5.28}$$

Then similar to (5.24), a dynamic program can be applied to generate the trajectory at each time step. Note that the number of computations needed to find the trajectory equals to

$$\frac{\mathcal{V}^2 N(N-1)}{2}, \tag{5.29}$$

where $\mathcal{V}$ is the number of discrete states in the search space. Since we are interested to find a 3D trajectory, $\mathcal{V}$ is not small. Consequently, finding the optimal trajectory is not directly feasible from problem (5.27). To deal with this problem, in the following we elaborate on a low-complexity greedy algorithm to find a sub-optimal trajectory.

5.3.4 Greedy Trajectory Design

As mentioned in the last section, at each time step the trajectory needs to be updated after collecting new measurements. In other words, it is sufficient to just determine where the drone needs to go in the next time step, instead of designing the entire trajectory at each step. To this end, we propose a greedy trajectory optimization. In this approach, the trajectory is designed locally and piece by piece. At each time step, the UAV seeks to find the best location for the next step to go and collect measurements that can

potentially offer the maximum improvement in channel parameters and user location estimates. The greedy trajectory optimization can be formulated as follows

$$\min_{\mathbf{v}[n+1]} \quad \mathrm{tr}(\hat{\mathbf{F}}^{-1}_{n+1,n,\mathrm{LoS}} + \hat{\mathbf{F}}^{-1}_{n+1,n,\mathrm{NLoS}}) \tag{5.30a}$$

$$\text{s.t.} \quad \frac{1}{v_{\max}} \sum_{i\in[2,n+1]} \|\mathbf{v}[i] - \mathbf{v}[i-1]\| \leq T, \tag{5.30b}$$

$$(5.22\text{c}), \tag{5.30c}$$

where n is the current time step. By using (5.20), problem (5.30) is equivalent to the following problem

$$\max_{\mathbf{v}[n+1]} \quad \mathrm{tr}(\hat{\mathbf{R}}_{n+1,n,\mathrm{LoS}} + \hat{\mathbf{R}}_{n+1,n,\mathrm{NLoS}}) \tag{5.31a}$$

$$\text{s.t.} \quad (5.30\text{b}).(5.22\text{c}), \tag{5.31b}$$

where $\hat{\mathbf{R}}_{n+1,n,s}$ is the estimation of $\mathbf{R}_{n+1,s}$ given learned channel parameters and user location estimates up to time step n. Problem (5.31) implies that the best drone position in the next time step is the one that maximizes the improvement in the estimation (equivalently minimizing the estimation error) while guaranteeing that the drone will end up in the terminal point at the end of the mission (time step N). Thus, to solve (5.31) we can associate with our problem the following performance index

$$L(\mathbf{v}[n]) = \begin{cases} \mathrm{tr}(\hat{\mathbf{R}}_{n+1,n,\mathrm{LoS}} + \hat{\mathbf{R}}_{n+1,n,\mathrm{NLoS}}), & \frac{\|\mathbf{v}[n]-\mathbf{v}_{\mathrm{F}}\|}{v_{\max}} \leq T - \sum_{i=2}^{n} T_i \\ -\infty, & \text{otherwise} \end{cases}, \; n \geq 2, \tag{5.32}$$

where

$$T_i = \frac{1}{v_{\max}} \|\mathbf{v}[i] - \mathbf{v}[i-1]\|. \tag{5.33}$$

The definition in (5.32) imposes the drone to reach the terminal point $\mathbf{v}_{\mathrm{F}}$ within the total flying time constraint T. Finally, we can reformulate (5.31) as follows

$$\max_{\mathbf{v}[n+1]} \quad L(\mathbf{v}[n+1]) \tag{5.34a}$$

$$\text{s.t.} \quad \mathbf{v}[1] = \mathbf{v}_{\mathrm{I}}. \tag{5.34b}$$

Assuming that the drone's position evolves at each time step according to:

$$\mathbf{v}[n+1] = \mathbf{v}[n] + \begin{bmatrix} \cos(\phi[n])\cos(\psi[n]) \\ \sin(\phi[n])\cos(\psi[n]) \\ \sin(\psi[n]) \end{bmatrix} \rho[n], \tag{5.35a}$$

$$h_{\min} \leq z[n] \leq h_{\max}, \; \forall n \in [1, N-1], \tag{5.35b}$$

where $\phi[n], \psi[n], \rho[n]$ are defined in Section 4.2, the next optimal drone position is given by

$$\begin{aligned} \mathbf{v}^*[n+1] := \arg \max_{\phi[n],\psi[n],\rho[n]} \quad & L(\mathbf{v}[n]), \\ \text{s.t.} \quad & (5.35), (5.34\text{b}). \end{aligned} \tag{5.36}$$

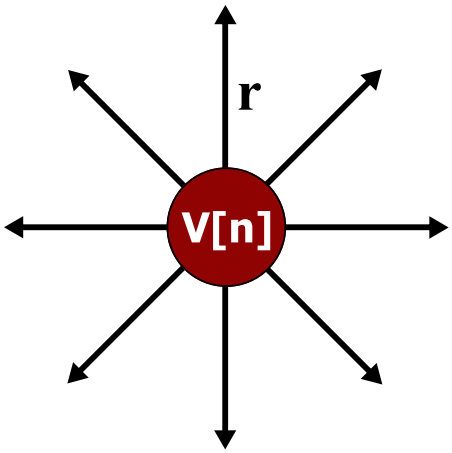


Figure 5.1 – An example of the possible actions in the greedy trajectory design at the n-th time step. To find the best position for the drone in the next time step all the eight drone's adjacent positions at time step n need to be evaluated.

To solve problem (5.36), we initialize $n = 1$ and the drone starts flying from base point $\mathbf{v}[1] = \mathbf{v}_{\mathrm{I}}$. Then, to find the best drone position in the next time step ($\mathbf{v}[n+1], n \in [1, N-1]$), we discretize the search space around the current drone location and we calculate (5.32) for all adjacent points. Then the neighbor point with the maximum value is chosen as the drone location in the next step. If the value of all the adjacent point are calculated as infinity, then the drone moves towards the terminal location $\mathbf{v}_{\mathrm{F}}$ by $\frac{\|\mathbf{v}[n]-\mathbf{v}_{\mathrm{F}}\|}{N-n}$ meters, where n is the current time step.

In Fig. 5.1, an example of the greedy trajectory design at the n-th time step is shown. In this example, the set of feasible drone positions comprises eight adjacent points of current drone location by selecting the input actions as follows

$$\begin{aligned} \phi[n] &\in \left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}\right\}, \\ \psi[n] &= 0, \\ \rho[n] &\in \{r\}, \end{aligned} \tag{5.37}$$

where r denotes the discretization unit. To find the best position for the drone, in the next time step, all eight drone's adjacent positions at time step n need to be evaluated.

5.4 Numerical Results

In this section, we provide numerical results to show the performance of the proposed algorithms. We consider a dense urban city neighborhood of size $600 \times 600\, m^2$, comprising buildings and regular streets. The height of the buildings is Rayleigh distributed in the range of 5 to $40\, m$ [7]. Different propagation parameters . During the simulations, in both the cases of random and optimized trajectories, the drone starts flying from the starting

point $\mathbf{v}_{\mathrm{I}} = [0, 0, 50]^{\mathrm{T}}$ meters and ends up at the terminal points $\mathbf{v}_{\mathrm{F}} = [300, 300, 50]^{\mathrm{T}}$ meters.

In Fig.5.2-a, we compare the performance for the localization algorithm proposed in this chapter with [83] and [65] over random Monte-Carlo iterations. The difference between existing methods [83] and [65] compared to our method is as follows: In both studies, a set of arbitrary RSSI measurements is used for the purpose of user localization and channel estimation (no trajectory optimization is considered) by assuming a two-segment radio channel model without exploiting the 3D map information. In [65], an EM algorithm is utilized for RSSI measurements classification task and estimating the radio channel parameters and user locations, while in [83], an unsupervised learning method is introduced to jointly classify the RSSI measurements and learn the channel model parameters and localize users. The root mean square error (RMSE) of the user localization for the proposed map-based algorithm, for a random trajectory, is shown by the blue solid-line marked with squares, and the map-based localization error when the drone follows the optimal trajectory is shown by the red dashed line marked with triangles. The result of the algorithm introduced in [65] for UAV random trajectories is depicted by the red dashed-dotted-line marked with squares. The blue dashed line marked with circles represents the localization accuracy for [83] when the UAV takes a random trajectory. To have a fair comparison, a set of random trajectories was generated and then used for all algorithms. For further comparison, we also proposed an algorithm to generate an optimized trajectory for the localization method in [83] by using the global LoS probability [15]. In this approach, we consider the same trajectory design algorithm as proposed in Section 5.3.4 with the difference that instead of using the 3D map information, we assign a LoS probability for each user. Then the LoS probability of a link between k-th user and the drone at time step n is given by

$$p_k[n] = \frac{1}{1 + \exp\left(-a\,\theta_k[n] + b\right)}, \tag{5.38}$$

where $\theta_k[n] = \arctan(z[n]/r_k[n])$ denotes the elevation angle and $r_k[n]$ is the ground projected distance between the drone and the k-th node located at $\mathbf{u}_k$ in the time slot n. Parameters $\{a, b\}$ are the model coefficients which are computed according to [15] and based on the characteristics of the city. We then assume that the k-th user is in LoS condition to the drone at time step n if its LoS probability is greater than 0.5. The localization RMSE pertaining to this method is shown by the red solid-line marked with circles. It is clear that the map-based methods (both for the random and the optimized trajectories) outperform the other approaches.

In Fig. 5.2-b, the cumulative distribution function (CDF) vs. localization RMSE for different approaches in a single user scenario is shown. For all approaches, the drone takes the optimal trajectory with a length of 1000 meters. It shows that using the 3D map can bring substantial gains to the localization accuracy.

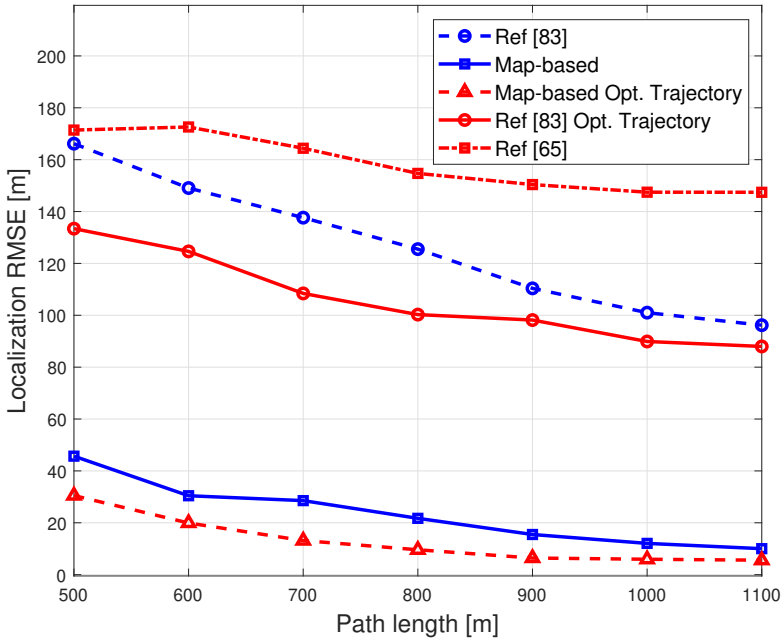
In Fig. 5.3, we investigate the effect of increasing the number of users on the performance of the map-based localization algorithm while the drone takes a random trajectory with a fixed length of 900 meters for different Monte-Carlo iterations. It is observed that the estimation error improves by increasing the number of users despite

the fact that by increasing the number of users the number of unknown parameters also increases. This is because by increasing the number of users the number of gathered measurements linearly increases while the unknown parameters regarding learning the channel are fixed. Consequently, the algorithm can learn the channel faster and more accurate which improves the localization performance as well.

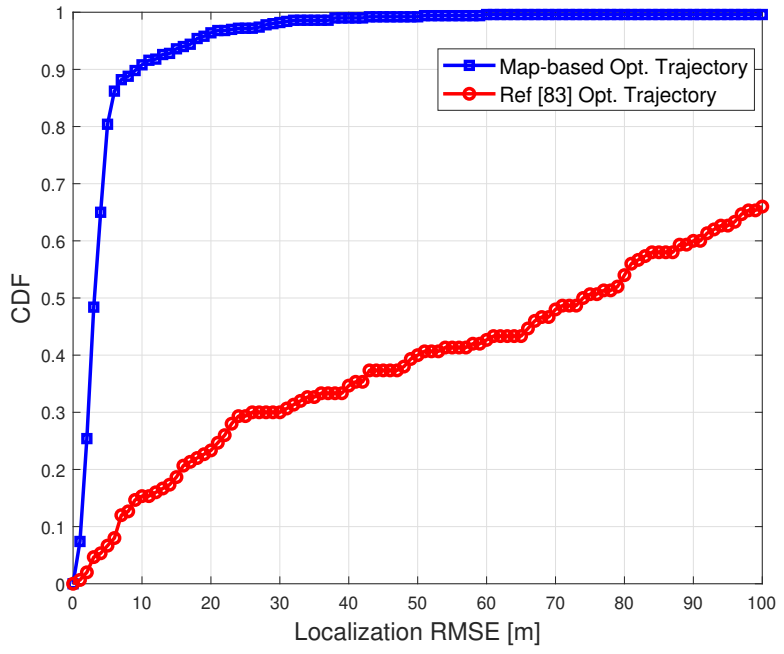
An example of the optimized trajectory generated according to Section 5.3.4 is shown, from a top view, in Fig. 5.4-a. The altitude of the drone along the trajectory is illustrated in Fig. 5.4-b.

5.5 Conclusion

In this chapter, we considered an instance of the UAV-aided wireless network consisting of ground users which are randomly scattered in a city. The goal of the UAV is to estimate the user locations from the collected RSSI measurements by capitalizing on the 3D map information of the city. Moreover, we proposed an online algorithm to design an optimized trajectory for the UAV to improve the performance of the localization and channel learning under a given mission duration. The simulations show a considerable gain brought by exploiting the 3D map in the performance of the node localization.



(a)



(b)

Figure 5.2 – (a) RMSE of user localization. (b) CDF of user localization for a fixed trajectory length.

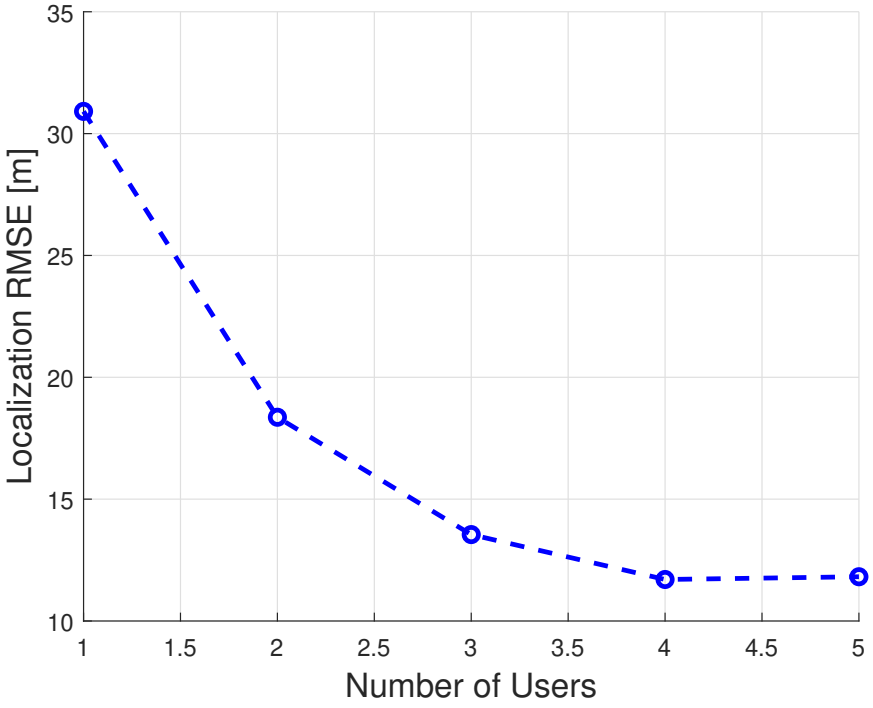
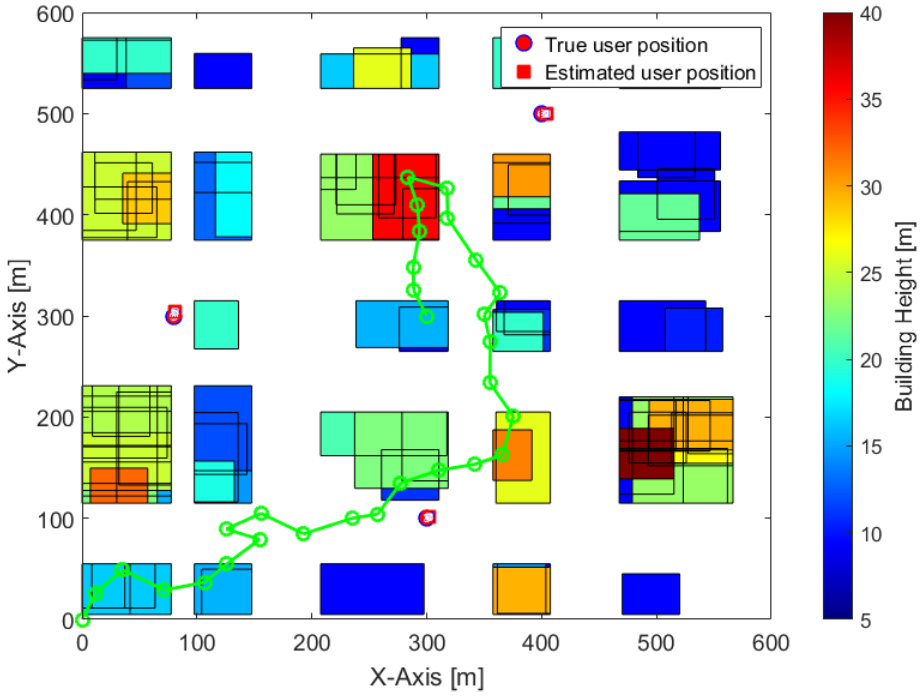
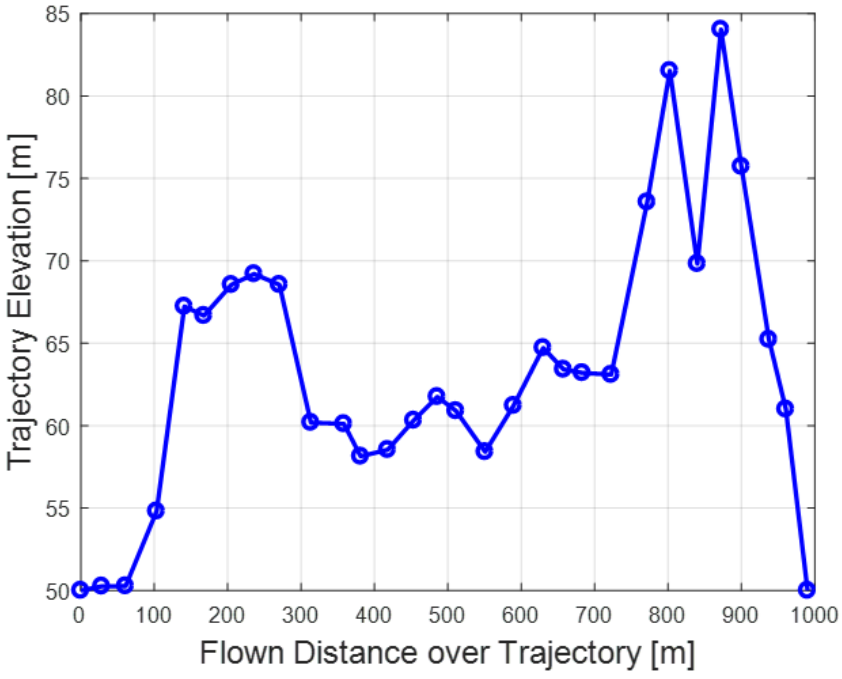


Figure 5.3 – User localization error by increasing the number of users for the map-based algorithm while the UAV follows arbitrary trajectories with a fixed length of 900 meters.



(a)



(b)

Figure 5.4 – (a) Top view of the generated trajectory using the sub-optimal approach. (b) Drone altitude along the trajectory.

Chapter 6

3D City Map Reconstruction from Radio Measurements

6.1 Introduction

While existing work on the use of UAV in radio access networks has focused on the problem of optimal UAV positioning [44; 84], and radio capacity maximization [85] or radio map predictions [27; 86], its potential interest as a tool to reconstruct 3D city maps, from the radio measurements it can make, has so far been completely overlooked. Hence this chapter constitutes a first attempt at it. Note that another motivation for reconstructing 3D maps using a drone lies in the fact that the 3D map itself can be exploited as a tool to enhance the solution to optimally positioning the very same drone in networking applications [56], as discussed in previous chapters.

The proposed principle of reconstructing building shapes and locations from radio measurements bears some intuitive analogy with the problem of reconstructing the shape of an object that is exposed under a series of light sources from the shadow images created from the object. Some major differences are to be noted however. In our context, the light/shade classification is replaced with a notion of LoS/NLoS classification for the radio channel. However, a major challenge lies in the fact that the LoS/NLoS data is not readily available from the measurement data itself. Or rather this information must somehow be inferred from the measurements via a learning algorithm which is proposed in the chapter. Secondly, the reconstruction of the very rich topology information of a city with hundreds of building is much more difficult than that of most common objects. Luckily, this problem is compensated by (i) the fact that the UAV can be tuned to record channel measurement from a very large number of ground user locations within a short time span, and (ii) the UAV positions can be modified and cumulated to offer a rich data set for learning, and ultimately giving an accurate description of the ground terrain. It should be noted that the notion of using signal strength measurements for characterizing the shape of a building is not new in itself. In [87] the authors propose the use of satellite-based GPS measurement to enhance 3D maps. However, the use of satellite is not as flexible as that of drones as their path cannot be optimized. Also in

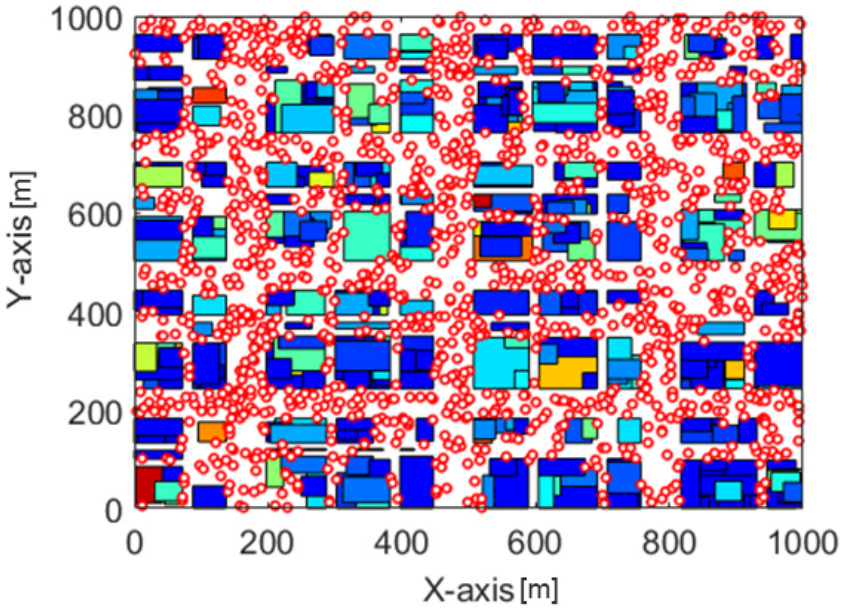


Figure 6.1 – Top view of the city building map and outdoor ground user locations which are shown by circles.

[88], authors propose the use of two drones to infer the inner structure of a building.

In summary, our contributions in this chapter are as follows:

- We propose a first attempt for reconstructing a whole 3D city map based on radio measurements made from a low altitude UAV with RSSI sensing capability. The RSSI levels are gathered at each selected UAV location from a large set of static outdoor ground users in a time-division multiplexing mode (TDM).

- The approach consists of two phases. In phase 1, machine learning tools are applied to learn the propagation parameters from the RSSI measurements from a target group of ground users. The learned parameters are in turn exploited to classify outdoor users in either LoS or NLoS categories. A technique is proposed to optimize the selection of the target user group. In phase 2, the LoS/NLoS classified data is exploited to yield a reconstructed city map via the low-complexity resolution of a large set of inequality equations. Note that, this chapter tackles the case of hard LoS/NLoS classification. The scenario of soft (probabilistic) LoS classification is studied in [27].

- Finally, the reconstruction quality reveals the notion of an optimal UAV altitude which is predicted analytically using an abstracted city model.

6.2 System Model

We consider an urban area randomly uniformly scattered with a (possibly large) set of K ground level users carrying radio transmitting equipment. All the users of interest are assumed to be outdoor users and surrounded by a number of city buildings as illustrated in Fig. 6.1. Note that, indoor users could potentially be exploited to improve reconstruction although this aspect is left for follow-up work. The city takes the shape of a square with length L. The true city map is drawn according to a random model where streets follow a Manhattan-like regular grid and building heights follow a bounded random distribution (see details in Section 6.5). The width of each building is random uniform with average W. The main street width is set to V (some lanes can be smaller). An UAV flies over the city, following a prescribed arbitrary trajectory, in order to measure the signal strength from the K ground level users (while possibly providing connectivity services) which have GPS-tracked positions denoted by $\mathbf{u}_k = [x_k, y_k]^{\mathrm{T}} \in \mathbb{R}^2, k \in [1, K]$. Over the course of its trajectory, the UAV collects $K \cdot N$ scalar RSSI measurements where N is the number of UAV locations at which multi-user measurements are made. The 3D n-th UAV location is denoted by $\mathbf{v}[n] = [x[n], y[n], h_d]^{\mathrm{T}} \in \mathbb{R}^3, n \in [1, N]$ where h_d is a fixed selected flying altitude (see Section 6.5).

6.3 LoS vs. NLoS Classification

Similar to detecting "shade" vs. "light" in an image, we propose to determine and exploit the LoS vs. NLoS status of any drone-user link. To this end, we adapt classical Support Vector Machine (SVM)–based machine learning methods to classify the users in LoS and NLoS categories from each UAV position $\mathbf{v}[n], n \in [1, N]$. In turn, maximum likelihood estimator will be employed to find the parameters of each LoS and NLoS channel model. Note that classification requires a labeled training data set which is not available in our setting, hence some unsupervised learning is required. However, in order to reduce complexity, we propose to apply unsupervised learning (in the form of K-means clustering) over a selected target group of users rather than on the entire network. Our simulations suggest that a satisfactory performance for channel parameter estimation can be reached this way. An approach for optimizing the selection of the target user group is later proposed.

6.3.1 Target User Clustering

We consider a fixed UAV location[1] $\mathbf{v}$ and a target sub-group of users, defined by the area located in a ring centered at the UAV with radius R and a width ΔR. The value of ΔR can be adjusted to control the target population size. The target user group is selected as a (narrow) ring in order to maximize distinguishability between LoS and NLoS effects on RSSI, while abstracting out the distance effect on path loss. In turn, the target users are clustered into LoS and NLoS categories based on the measured RSSI using a standard K-means approach [89].

[1] $\mathbf{v}$ can coincide with $\mathbf{v}[1]$ or be a default location such as the center of the network.

6.3.2 Optimization of Target User Group

Since the clustered target users will serve as labeled training data to classify the rest of the network users, it is essential that maximum LoS/NLoS distinguishability is achieved on the target group. The following method is proposed for selecting the ring radius R.

Let $\delta_{\mathrm{LoS}}(R)$ and $\delta_{\mathrm{NLoS}}(R)$ be the number of users in LoS and NLoS categories after clustering. Also let $\bar{g}_{\mathrm{LoS}}(R)$, $\bar{g}_{\mathrm{NLoS}}(R)$ be the average RSSI measured (in dB scale) over the users in the LoS and NLoS clusters respectively. The following criterion $J(R)$ attempts to strike a trade-off between the distinguishability between the RSSI of LoS and NLoS channels which is typically seen at larger distances, and the balance in LoS vs. NLoS population sizes which is observed from the UAV at closer distances. Indeed for a fixed UAV altitude, the users located at a greater distance from the UAV will all tend into the NLoS category. The criterion is selected as:

$$J(R) = \left(\frac{\min\left(\delta_{\mathrm{LoS}}(R), \delta_{\mathrm{NLoS}}(R)\right)}{\delta_{\mathrm{LoS}}(R) + \delta_{\mathrm{NLoS}}(R)} \left| \bar{g}_{\mathrm{LoS}}(R) - \bar{g}_{\mathrm{NLoS}}(R) \right| \right)^{-1}. \tag{6.1}$$

Inverse of $J(R)$ can be interpreted as an approximation of the average RSSI difference between LoS and NLoS users in the target group. The appropriate R can be found by optimizing the following expression using a discretized line search:

$$R^* = \arg\min_R J(R). \tag{6.2}$$

6.3.3 Radio Propagation Parameter Learning

Considering the clustered RSSI measurements, which are generated from section 6.3.1, we now want to estimate the propagation parameters for each of the LoS and NLoS scenarios. Considering d_k the distance of the k-th user in the target group from the UAV, then according to (2.2) the RSSI measurement for target user k with LoS/NLoS status s is given by

$$g_{k,s} = ß_s - \alpha_s \varphi(d_k) + \eta_s. \tag{6.3}$$

The distribution of RSSI conditioned on status s can be written as

$$p(g_{k,s}|d_k; \boldsymbol{\theta}_s) = \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\left\{ -\frac{\left(g_{k,s} - ß_s + \alpha_s \varphi(d_k)\right)^2}{2\sigma_s^2} \right\}, \tag{6.4}$$

where $\boldsymbol{\theta}_s = [\alpha_s, ß_s, \sigma_s]^{\mathrm{T}}$. Therefore, assuming the estimated LoS/NLoS labels are perfect, the likelihood function of parameters $\boldsymbol{\theta}_s$ for each cluster is denoted by

$$L(\boldsymbol{\theta}_s) = p(\mathbf{g_s}|\mathbf{d}; \boldsymbol{\theta}_s) = \prod_{k \in \mathcal{K}_s} p(g_{k,s}|d_k; \boldsymbol{\theta}_s), \tag{6.5}$$

where $\mathcal{K}_s$ is a set of indices of users that falls into segment s. Then the MLE [90] can be obtained from solving the following optimization problem

$$\max_{\boldsymbol{\theta}_s} \prod_{k \in \mathcal{K}_s} p(g_{k,s}|d_k; \boldsymbol{\theta}_s), \tag{6.6}$$

or equivalently instead maximizing the log likelihood $\ell(\boldsymbol{\theta}_s)$

$$\ell(\boldsymbol{\theta}_s) = \sum_{k \in \mathcal{K}_s} \log p(g_{k,s}|d_k; \boldsymbol{\theta}_s). \tag{6.7}$$

This function can be maximized by differentiating with respect to $\boldsymbol{\theta}_s$ and setting to zero (more details can be found in [27]).

6.3.4 User Classification

Equipped with estimates of the propagation parameters, we proceed with classifying the remaining network users into LoS or NLoS categories. Several methods can be resorted to. In this chapter, we use the SVM classifier for the sake of good performance, robustness and ease of implementation [91].

Considering the clustered RSSI observations obtained from Section 6.3.1 as training data set $\mathcal{T}\left(f_j, t_j \,|\, j \in [1, \delta_{\text{LoS}}(R) + \delta_{\text{NLoS}}(R)]\right)$ where f_j is the input feature and t_j corresponds to a LoS/NLoS label for the j-th user in the target group. We define f_j as

$$f_j = \|g_j - \hat{g}_{j,\text{LoS}}\|_2,\ t_j = \begin{cases} 1 & \text{LoS} \\ -1 & \text{NLoS} \end{cases}, \tag{6.8}$$

where g_j denotes measured RSSI and $\hat{g}_{j,\text{LoS}}$ is the predicted RSSI under a LoS channel hypothesis. Now armed with defined training data set $\mathcal{T}$, a SVM machine learning algorithm is designed to classify the users. We define

$$t(f) = \text{sign}(\boldsymbol{\omega}^{\text{T}} \psi(f) + \omega_0), \tag{6.9}$$

where $\psi(f)$ is the predetermined input feature mapping function and $(\boldsymbol{\omega}, \omega_0)$ are parameters learned from the training data. As feature mapping function, the radial basis function (RBF) is classically selected [92]. We apply (6.9) for every network user to find its LoS/NLoS label.

Note that the above algorithm treats each user independently. As a way to save on complexity or improve robustness it is possible to exploit the natural correlation in the LoS/NLoS status for users that are closely located. This can be easily implemented by discretizing the map into small grid squares (few meters) and assigning all users in the same grid squares to a unique status (e.g. determined from majority rule).

6.4 3D City Map Reconstruction

In order to carry out the map reconstruction, we discretize the city and represent it in the form of a grid map where each grid element consists of 2D coordinates and a height indicator which will serve as an estimate of the height for any building located as these coordinates. The height indicator of the all grids are initialized equal to the maximum possible building height $h_{\max}$, and then iteratively refined using LoS status information. The i-th element of the grid for the reconstructed map is denoted by $\hat{\mathbf{a}}_i = [x_{\mathbf{a},i}, y_{\mathbf{a},i}, \hat{z}_{\mathbf{a},i}]^{\text{T}} \in \mathbb{R}^3$ where $\hat{z}_{\mathbf{a},i}$ denotes the building height estimate at ground coordinates

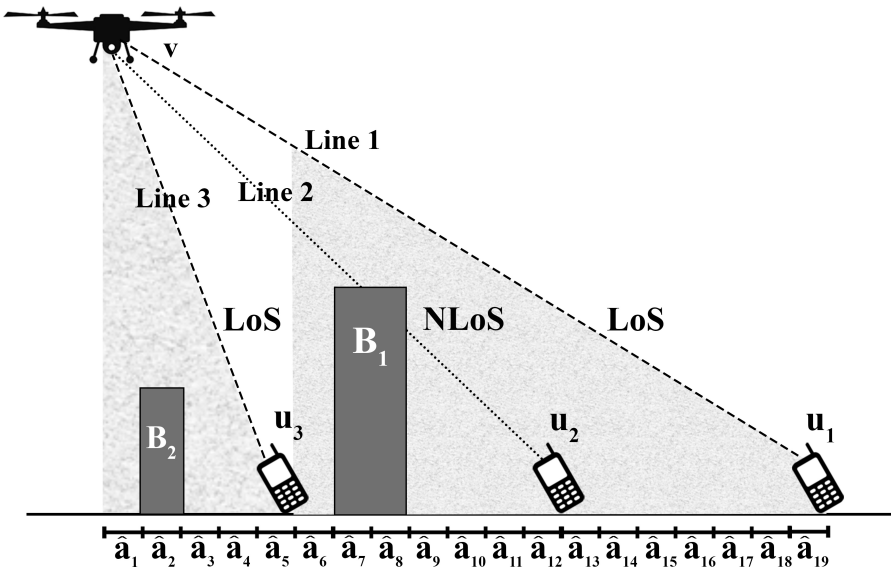


Figure 6.2 – Building map reconstruction: Example with 3 RSSI measurements. The first and third users were successfully labeled as LoS, while the second was correctly labeled NLoS. As a result any building between the drone and user 1 (resp. user 3) must lie below line 1 (resp. below line 3).

$(x_{\mathbf{a},i}, y_{\mathbf{a},i})$. In contrast, the true map's grid is defined as $\mathbf{a}_i = [x_{\mathbf{a},i}, y_{\mathbf{a},i}, z_{\mathbf{a},i}]^{\mathrm{T}} \in \mathbb{R}^3$. Note that by construction $z_{\mathbf{a},i} \geq 0$, $\hat{z}_{\mathbf{a},i} \geq 0$ and $z_{\mathbf{a},i} = 0$ points to a street location.

In our approach we *iteratively* solve for $\hat{z}_{\mathbf{a},i}$ for all i, by working out the sequence of inequalities implied by the existence of labels set to LoS pertaining to users that are informative of the i-th grid. As an illustration, let us take the example of k-th user at location $\mathbf{u}_k$ and the n-th UAV location $\mathbf{v}[n] = [x[n], y[n], h_d]^{\mathrm{T}}$. The equation of the 3D line which passes through these two points is given by

$$\mathcal{L} : \frac{x - x[n]}{x_k - x[n]} = \frac{y - y[n]}{y_k - y[n]} = \frac{z - h_d}{-h_d}. \tag{6.10}$$

Let $\hat{\mathcal{A}} = \{\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \cdots, \hat{\mathbf{a}}_n\}$ be defined as the set of grids which lie on projection of the line $\mathcal{L}$ onto X-Y plane. Assuming this user-drone link has a LoS label, we can write a set of inequalities for all $\hat{\mathbf{a}}_i$ as follows:

$$\hat{z}_{\mathbf{a},i} \leq \frac{(x_{\mathbf{a},i} - x[n])\,(-h_d)}{x_k - x[n]} + h_d\,;\; i \in [1, |\hat{\mathcal{A}}|], \tag{6.11}$$

where $|\hat{\mathcal{A}}|$ denotes the cardinality of $\hat{\mathcal{A}}$. The value in the right side of (6.11) corresponds to the z value of line $\mathcal{L}$ in the center of i-th grid $(x_{\mathbf{a},i}, y_{\mathbf{a},i})$. To put it differently, all the buildings along the line $\mathcal{L}$ must lie below it as illustrated in Fig. 6.2. After accounting for inequality (6.11), $\hat{z}_{\mathbf{a},i}$ is updated with $\min\left(\hat{z}_{\mathbf{a},i}, \frac{(x_{\mathbf{a},i} - x[n])(-h_d)}{x_k - x[n]} + h_d\right)$. Moreover, we

assume that all the buildings in the city are taller than a min value $h_{\min}$, therefore the grids with final height estimates lower than $h_{\min}$ are set to zero and considered as street locations. Inequalities (6.11) are solved for all given pairs of users and UAV locations where measurements are gathered. The rigorous description of this algorithm is summarized in Algorithm 6.4.1.

Clearly the map reconstruction performance highly depends on the trajectory $\mathcal{X} = \{\mathbf{v}[1], \cdots, \mathbf{v}[N]\}$ followed by the UAV to produce a rich enough measurement data set. The problem of optimizing for the UAV path is omitted and studied in future works. In this chapter a simple yet reasonable path planning is considered (see Section 6.5).

6.4.1 Optimum UAV Altitude

Although it is clear from intuition that the selection of a particular UAV altitude h_d impacts the map reconstruction performance, the derivation of an exact optimum is bound to be difficult, depending on many system parameters. Nevertheless our studies reveal two major factors in driving reconstruction performance. First an overly low flight height yields an excess in the number of NLoS reported measurements. Such measurements are less informative about the city topology than LoS reportings because they characterize the existence of some building somewhere crossing the line of sight but without a mention of where. In contrast, flying higher will cause more LoS reportings. Flying too high on the other hand will cause a gross overestimation of building heights ($\hat{z}_{\mathbf{a},i} = h_{\max}, \forall i$). In this section we build a small analytical deterministic model which captures these key effects so as to provide a guideline to design h_d. We consider a simplified one-dimensional city model with a continuum of outdoor users and where all the buildings are of equal length W, the inter building space of V and the city map side of L. All buildings are set to a height equals to that of the average in the true city H_{av}, yet assumed unknown here. We consider a single UAV location with height h_d. When modeling the reconstruction error it is assumed that the building height is unknown but only the upper bound $h_{\max}$ is known. After processing the LoS-based inequalities, the reconstruction error can be written as shown below.

Proposition 6.4.1. *The reconstruction error after accounting for LoS-based inequalities is as follows*

$$Er_T(h_d) = \left(\left\lfloor \frac{L}{W+V} \right\rfloor - M \right) (h_{\max} - H_{av}) + \sum_{i=1}^{M} \left| \frac{W(h_d - H_{av})}{2i(W+V)} \right|, \tag{6.12}$$

where $M = \left\lfloor \frac{V(h_d - H_{av})}{H_{av}(W+V)} \right\rfloor$.

Proof. See Appendix C. □

A suitable UAV altitude h_d is found via the following line search

$$\begin{array}{ll} \min\limits_{h_d} & Er_T(h_d) \\ \text{s.t} & Q \geq 0 \end{array}, \tag{6.13}$$

Algorithm 6.4.1 Pseudocode for map reconstruction

1. Initialize the height of all grids to $h_{\max}$
2. **For** $n \in [1, N]$ **then**
3. **For** $k \in [1, K]$ **then**
4. **If** $\mathbf{u}_k$ is LoS **then**
5. $\mathcal{L} \leftarrow$ 3D line passes through points $(\mathbf{v}[n], \mathbf{u}_k)$
6. $\mathcal{L}_{\text{Prj}} \leftarrow$ Projection of $\mathcal{L}$ on the ground
7. $\hat{\mathcal{A}}_n \leftarrow$ The grids which lie on the line $\mathcal{L}_{\text{Prj}}$
8. **For** $i \in [1, |\hat{\mathcal{A}}_n|]$ **then**
9. $h_i = \frac{(x_{\mathbf{a},i} - x[n])(-h_d)}{x_k - x[n]} + h_d$
10. $\hat{z}_{\mathbf{a},i} := \min\left(\hat{z}_{\mathbf{a},i}, h_i\right)$
11. **If** $\hat{z}_{\mathbf{a},i} < h_{\min}$ **then**
12. $\hat{z}_{\mathbf{a},i} = 0$
13. **End if**
14. **End**
15. **End if**
16. **End**
17. **End**

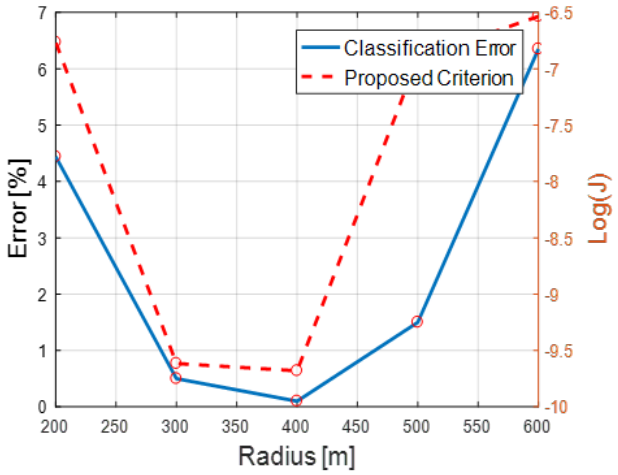


Figure 6.3 – Determining the best radius of the ring-like area for K-means algorithm using proposed criterion. Solid line denotes the classification error for each radius and dashed line is the corresponding criterion, predicting the same minimum.

where $Q = \left\lfloor \frac{L}{W+V} \right\rfloor - M$.

6.5 Numerical Results

We consider a dense urban Manhattan-like 1 km by 1 km area consisting of a regular street grid and buildings with uniform random height in the range of $h_{\min} = 5$ to $h_{\max} = 40$ meters. The width of each building is random uniform with average $W = 80$ meters and the main street width is set to $V = 60$ meters. The outdoor users are spread uniformly randomly as in Fig. 6.1. For reconstruction purposes, the map is discretized over a grid with 5 meters granularity. The UAV test trajectory is a square with length 800 meters, centered at the center of the map. $N = 32$ set of measurements are reported (every 100 meters). Propagation parameters are chosen as $\alpha_{\text{LoS}} = 2.27$, $\alpha_{\text{NLoS}} = 3.64$, $ß_{\text{LoS}} = -30\,\text{dB}$, $ß_{\text{NLoS}} = -40\,\text{dB}$ generalized from typical fixed BS WINNER II [93]. Finally we select $\sigma^2_{\text{LoS}} = 2$ and $\sigma^2_{\text{NLoS}} = 5$.

To further improve reconstruction performance, a map smoothing procedure is used, whose goal is to filter out unrealistic height variations among closely located points on the grid.

We first test the target group selection optimization method of Section 6.3. Results are shown in Fig. 6.3 which confirms the classification error to be minimized at the ring radius predicted by our algorithm.

In Fig. 6.4 the classification result using K-means for users in the target area is depicted. The users are almost fully distinguishable, note that the signals with higher RSSI are LoS users. Then using SVM classifier all users ($K = 1584$) are classified with 99.9% precision which indicates a good performance of the classifier for our purpose.

Next, we are interested in optimal flying altitude and run the optimization in (6.13)

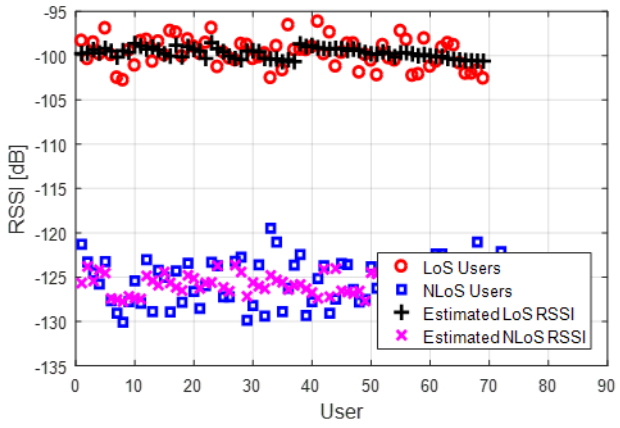


Figure 6.4 – K-means result for measured RSSI for users in target area and estimated RSSI for them using MLE.

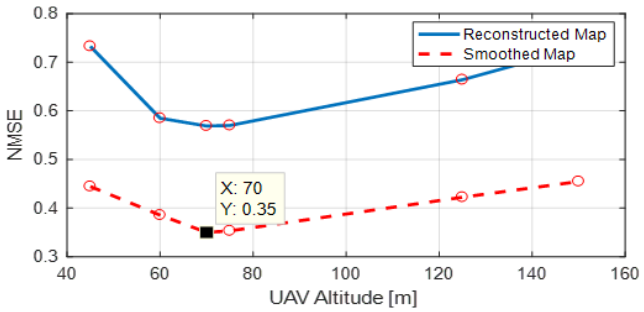


Figure 6.5 – Reconstruction normalised mean square error (NMSE) for both smoothed and rough map versus different UAV elevations.

to find a best altitude at 70 meters. The prediction is tested in Fig. 6.5 which shows the reconstruction error for a range of UAV altitudes, also predicting an optimum near 70 meters.

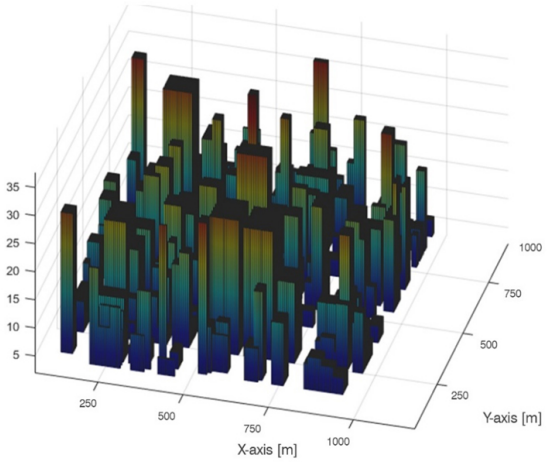
Fig. 6.6 shows an instance of reconstructed map after smoothing for the considered flight path with fixed altitude $h_d = 70$ meters. Note that, for the regions where there is no user reportings, the building estimated heights are set to the average building height by default.

Finally, the impact of user density is tested in Fig. 6.7 showing the improvement trend as the number of users (hence RSSI reportings) increase.

6.6 Conclusion

This chapter considered the problem of 3D city map reconstruction by exploitation of UAV-bound radio measurements. The proposed approach relies on the unique ability for a UAV-to-ground communication systems to detect and classify LoS vs. NLoS channels towards ground users using machine learning tools. The optimal height for the UAV is found using an analytical model which shows excellent predictive behavior. The present algorithm assumes perfect LoS/NLoS classification. In the case classification error statistics are available, robust approaches based on Bayesian estimation can be developed which are reported in companion work.

(a)



(b)

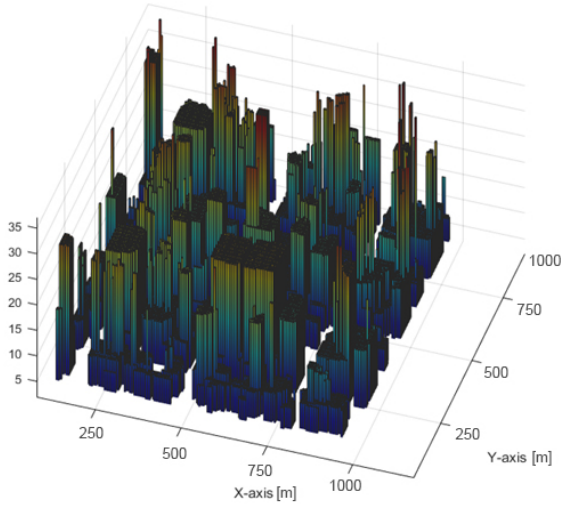


Figure 6.6 – (a) Side view of the 3D city map. The city map is discretized over small grid squares of length of 5 meters. (b) Reconstructed map after refinement and smoothing with NMSE equals to 0.35.

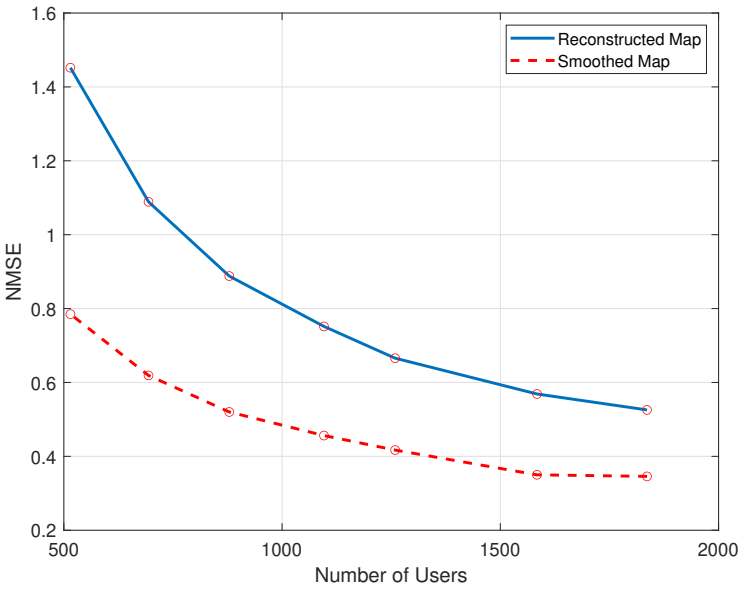


Figure 6.7 – Map reconstruction error versus increasing the number of users.

Chapter 7

UAV Trajectory Design Under Cellular Connectivity Constraints

7.1 Introduction

Cellular connected UAVs that can operate safely in beyond visual line of sight conditions are expected to open important future opportunities in the areas of transportation, goods delivery, and system monitoring. Ensuring ultra-reliable and low latency links between UAVs and their ground control stations plays a pivotal role in making these businesses a reality as many of the above mentioned application scenarios require UAVs to be autonomous or semi autonomous. Integrating UAVs into ubiquitous existing or future cellular networks as user terminals and connecting them with base stations offers simple and cost-effective solution to the UAV connectivity problem [11].

In spite of the promising results demonstrating the feasibility of supporting UAVs in current cellular networks, several new challenges have been highlighted in supporting aerial users in current cellular networks, which are otherwise developed for terrestrial users [94; 95; 96]. In particular, interference and abrupt changes in signal strength (compared to terrestrial users) have been observed in aerial users as the BS antennas are typically tilted a little downwards (intended for terrestrial users), thus making the aerial users experience side lobes.

However, the inherent advantage offered by UAVs in terms of 3D mobility can be exploited to efficiently design UAV paths to avoid the outage areas and exploit good channel conditions while not deviating too much away from the trajectories planned for original tasks. Motivated by this, several recent works have considered the problem of communication-aware trajectory design for cellular connected UAVs [9; 19; 10; 97; 98; 20]. Specifically, the problem of finding an optimal path in the sense of a shortest path between a departing point and a given destination such that the UAV consistently gets a reliable connection from the cellular network has been considered in [9; 19; 10; 97; 98]. The works in [9; 97] have considered the problem of finding the shortest path under cellular coverage constraints assuming that the UAV terminal experience LoS channels from the BSs at all times independent of UAV and BS locations. Convex optimization and graph based

approaches are used to optimize the trajectory. However, the chosen radio propagation model is not applicable in urban environments, where it is shown that air-to-ground channels exhibit switching from LoS and NLoS conditions depending on the UAV and BS locations, where NLoS conditions are caused by signal blockage, reflection and diffraction caused by city buildings [99; 7].

To overcome the drawback arising from using simple LoS channel models in urban environments, the works in [19; 10] have utilized a radio map of the environment that carries very fine grain information about the channel gains from all BSs in the trajectory optimization. While [19] considers only the altitude optimization of UAV, [10] optimizes trajectory in 2D while considering a fixed altitude. Both these works depend on discretizing the radio map of the overall flight region into finer grids and then use graph based algorithms to find the shortest path from the initial location to the destination. The complexity and performance trade-off of the shortest path algorithm depend on the number of nodes in the constructed graph, which in turns depend on the grid resolution used in discretizing the radio map. Note that the radio maps are not available on fly but need to be estimated offline by collecting lots of radio measurements from users in that environment [27].

Another approach to obtain realistic trajectories in complex urban environments is to use learning approaches which are model free [20; 98]. However, the drawback of such techniques is that they require a relatively high number of learning episodes to obtain the desired results.

In this chapter, we consider the problem of finding the shortest path between a starting location and a given destination such that a constant altitude flying UAV consistently gets a reliable QoS from the cellular network. Some of the key contributions are as follows:

- Instead of considering radio map which contains rich information channel gains but not easy to model analytically, and generally is not available for any arbitrary areas, we use the 3D map of the city along with a segmented pathloss model to construct coverage maps which serve as a high-quality approximation to the radio maps while having an analytical structure.
- Making use of the convexity of sub-regions within the coverage map, we prove that the optimal trajectory has a piecewise linear structure.
- By leveraging this optimal structure, we propose a low-complexity graph based shortest path algorithm that doesn't require discretizing the entire coverage map.

7.2 System Model

We consider a cellular connected UAV that flies over an urban area consisting of a number of city buildings for a duration of time T. The position of UAV at time $t \in [0, T]$ is denoted by $\mathbf{v}(t) = [x(t), y(t), h_d]^{\mathrm{T}} \in \mathbb{R}^3$, where h denotes the altitude of the UAV. For simplicity, the altitude of the UAV is set to a fixed value which is determined by the tallest building in the city to avoid the collision. We assume that the UAV is equipped with a

GPS receiver, hence $\mathbf{v}(t)$ is known. The UAV is presumed to fly from a pre-determined initial position $\mathbf{v}_{\mathrm{I}}$ at time $t = 0$ and has to reach to a terminal location $\mathbf{v}_{\mathrm{F}}$ by the end of the mission duration. The UAV flies at a constant speed of, hence the UAV's trajectory $\mathbf{v}(t), t \in [0, T]$ can solely be determined by the path it takes. During the mission, the UAV needs to be remained connected to one of the K outdoor static BSs which are randomly scattered with uniform distribution over the city. The k-th BS, $k \in [1, K]$, is located at $\mathbf{u}_k = [x_k, y_k, h_g]^{\mathrm{T}} \in \mathbb{R}^3$, where h_g stands for the height of the BS and is assumed to be the same for all BSs [1]. Moreover, we denote $\hat{\mathbf{u}}_k = [x_k, y_k, h_d]^{\mathrm{T}}$, $k \in [1, K]$ as the projections of the k-th BS locations on the 2D plane with the same altitude as the UAV.

7.2.1 Communication Model

We consider a cellular down-link scenario where the time varying SNR at the UAV from the k-th BS is given by

$$\rho_k(\mathbf{v}(t)) = \frac{P_b \gamma_{k,s}(t)}{\sigma^2}, \, 0 \leq t \leq T, \tag{7.1}$$

where P_b is the transmission power of the BS, $\gamma_{k,s}(t)$ is the channel gain between the k-th BS and the UAV flying at location $\mathbf{v}(t)$, σ^2 represents the noise power, and finally $s \in \{\mathrm{LoS}, \mathrm{NLoS}\}$ emphasizes the strong dependence of the propagation conditions in LoS or NLoS scenarios[27]. The channel gain between the UAV and the k-th BS is modeled as [27; 44]

$$\gamma_{k,s}(t) = \frac{\beta_s}{d_k(t)^{\alpha_s}}, \tag{7.2}$$

where α_s is a path loss exponent, β_s is a channel gain offset, and

$$d_k(t) = \|\mathbf{v}(t) - \mathbf{u}_k\|$$

represents the distance between the k-th BS and the UAV. Regarding the LoS/NLoS classification of the UAV-BS links, we leverage the knowledge of a 3D city map.

7.2.2 Problem Formulation

We are interested to find the shortest trajectory for the UAV between a predefined starting point $\mathbf{v}_{\mathrm{I}}$ and a terminal point $\mathbf{v}_{\mathrm{F}}$, while satisfying the minimum SNR $\bar{\rho}$ during the mission

$$\min_{0 \leq t \leq T} \max_{k \in [1,K]} \rho_k(\mathbf{v}(t)) \geq \bar{\rho}. \tag{7.3}$$

Since the UAV moves with a constant velocity, the trajectory optimization can be formulated as follows

$$\min_{T, \{\mathbf{v}(t), 0 \leq t \leq T\}} \quad T \tag{7.4a}$$

[1] By no means this is a restriction and the results presented in this chapter can be easily extended to the case with different BS heights.

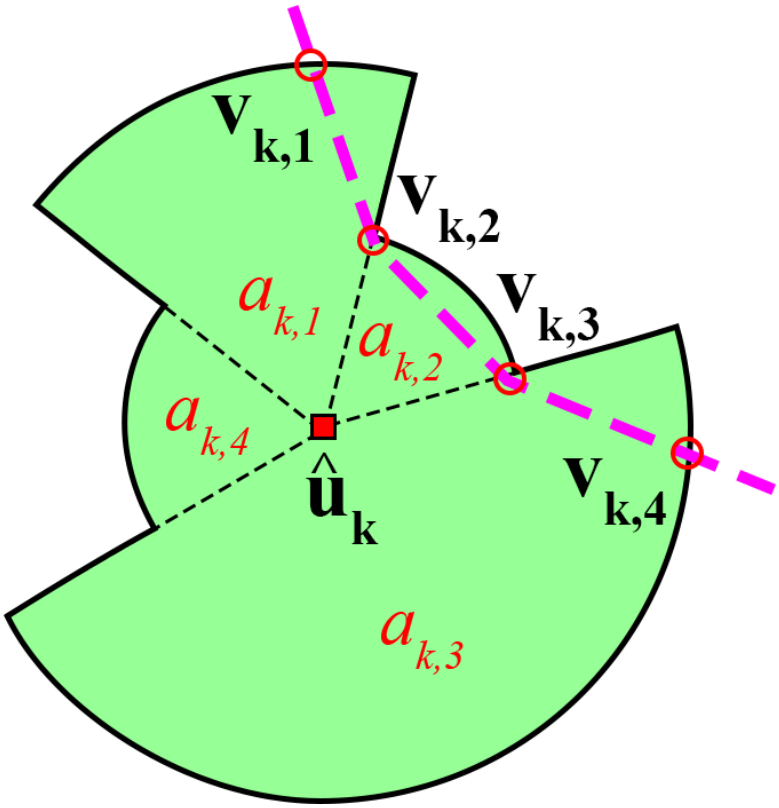


Figure 7.1 – Coverage area of a given BS and the sectors.

$$\text{s.t.} \quad (7.3), \tag{7.4b}$$

$$\mathbf{v}(0) = \mathbf{v}_{\mathrm{I}},\ \mathbf{v}(T) = \mathbf{v}_{\mathrm{F}}. \tag{7.4c}$$

This problem is not convex since the SNR in the constraint (7.3) is a non-differentiable and non-smooth function with respect to the UAV position due to the binary classification variable $s \in \{\text{LoS, NLoS}\}$, therefore this function is neither convex nor concave. Moreover, it is a functional optimization, hence, it is challenging to solve (7.4) optimally in general.

In the following, with some analysis we show that the optimal trajectory has some structures which can be exploited to make problem (7.4) more tractable. To this end, following results and definitions are helpful.

Definition 7.2.1. Coverage area: The coverage area of the BS is defined as a set of points with the same altitude as the UAV in which the SNR of the UAV-BS link will remain greater than or equal to $\bar{\rho}$. The coverage area of the k-th BS, $k \in [1, K]$ is defined as

$$\mathcal{A}_k = \{\mathbf{v} = [x, y, h_d]^{\mathrm{T}} \in \mathbb{R}^3 \,|\, \rho_k(\mathbf{v}) \geq \bar{\rho}\}. \tag{7.5}$$

Using the SNR expression in (7.1), the set of points $[x, y]$ that belong to set $\mathcal{A}_k$ can be written as

$$(x - x_k)^2 + (y - x_k)^2 \leq d_s, \tag{7.6}$$

where $d_s \triangleq \left(\frac{P_b \beta_s}{\sigma^2 \bar{\rho}}\right)^{\frac{2}{\alpha_s}} - (h_g - h_d)^2$. The radius d_s therefore depends on whether the point $\mathbf{v}$ is in LoS or NLoS with respect to the BS, which in turn depends upon the building distribution around that BS. Based on (7.6) and the 3D map, without loss of generality, the coverage areas $\mathcal{A}_k$ can be divided into M_k sectors

$$\mathcal{A}_k = \{a_{k,1} \cup \cdots \cup a_{k,M_k}\}, \tag{7.7}$$

where each $a_{k,i}$ is a convex shape which is a segment of a circle between two angles $\theta_{k,i}$ and $\theta_{k,i+1}$ with a radius of $r_{k,i}$. The radius $r_{k,i}$ depends on the building distribution and (7.6). For better understanding, an illustration of such coverage area of a BS is given in Fig.7.1 and in Fig. 7.2. For instance, regarding the coverage area depicted in Fig.7.1 for a given BS, we can write $\mathcal{A}_k = \{a_{k,1} \cup a_{k,2} \cup a_{k,3} \cup a_{k,4}\}$.

Definition 7.2.2. Coverage border: The coverage border is the perimeter of a coverage area of a given BS. The coverage border of the k-th BS, $k \in [1, K]$ is denoted $\mathcal{B}_k$.

Definition 7.2.3. Common areas and common borders: The common area between k-th and j-th BSs, $k, j \in [1, K], k \neq j$ represents the overlap regions of their coverage areas, i.e.,

$$\mathcal{C}_{j,k} = \mathcal{C}_{k,j} = \{\mathcal{A}_k \cap \mathcal{A}_j\}. \tag{7.8}$$

The borders of the common areas $\mathcal{C}_{j,k}$ is defined as the common borders which we denote by $\mathcal{D}_{j,k}$.

In Fig. 7.2, an example of the coverage areas, coverage borders, common areas, and common borders of two BSs is illustrated. The coverage area of each BS is depicted with a highlighted surfaces and the coverage borders are shown with solid black lines.

Proposition 7.2.1. *Problem* (7.4) *is equivalent to the following problem:*

$$\min_{N,\mathcal{V}} \quad \sum_{n \in [1,N-1]} \|\mathbf{v}_n - \mathbf{v}_{n+1}\|^2 \tag{7.9a}$$

$$s.t. \quad \rho(\mathbf{v}_n, \mathbf{v}_{n+1}) \geq \bar{\rho}\,, n \in [1, N-1], \tag{7.9b}$$

$$\mathbf{v}_1 = \mathbf{v}_I,\ \mathbf{v}_N = \mathbf{v}_F, \tag{7.9c}$$

where

$$\rho(\mathbf{x}, \mathbf{y}) = \min_{0 \leq \lambda \leq 1} \max_{k \in [1,K]} \rho_k \left(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}\right), \tag{7.10}$$

and $\mathcal{V} = (\mathbf{v}_n)_{n=1}^N$ *is the sequence of UAV trajectory points in* $\mathbb{R}^3$ *such that any two consecutive points are connected with a straight line.*

Proof. See Appendix D.1. □

Then to solve (7.9), we just need to optimize over a limited number of optimization variables, however this problem is still difficult to solve since constraint (7.9b) is neither convex nor concave. In what comes next, we develop a graph theory-based solution to this problem. First, we check the feasibility of problem (7.9) by proposing a graph theory based approach in a similar manner to the one proposed in [9]. We then derive a method to find a sub-optimal and efficient solution to problem (7.9).

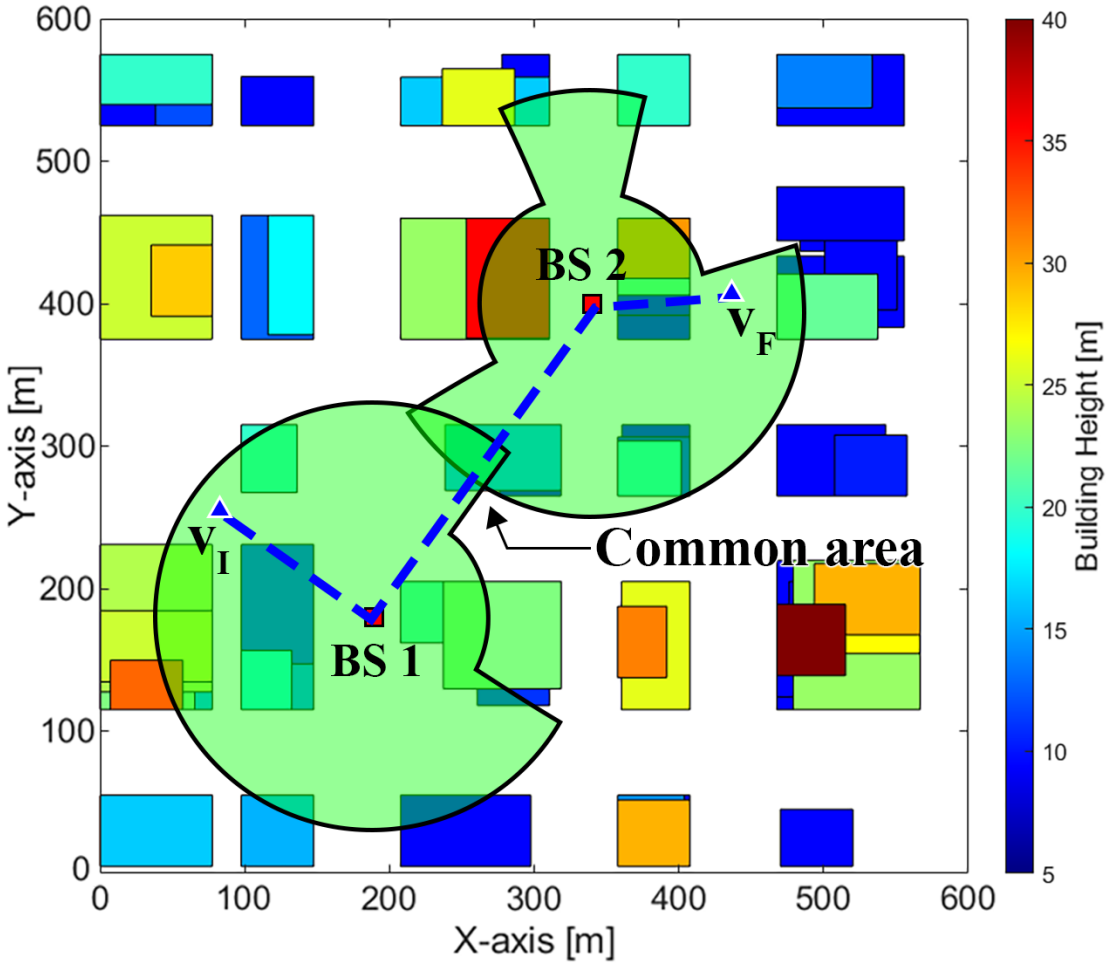


Figure 7.2 – Top view of the city, the BSs positions, coverage area of each BS, and the common area. The UAV flies at 50m and the BSs are on the ground level.

7.3 Feasibility Check

In this section, we investigate the feasibility of problem (7.9) by leveraging the graph theory approach. A trajectory sequence $\mathcal{V} = (\mathbf{v}_n)_{n=1}^{N}$ is a feasible solution to problem (7.9) if constraints (7.9b) is satisfied. In general, obtaining a feasible solution to problem (7.9) is not trivial, since the coverage area of BSs have non-convex shapes and the exhaustive search inherently cannot be avoided. For further simplification, we uniformly discretize the coverage border of each BS, which was defined in Definition 7.2.2, into Q samples. The discretized coverage border of the k-th BS, $k \in [1, K]$ is denoted by $\hat{\mathcal{B}}_k$, $|\hat{\mathcal{B}}_k| = Q$. We then define $\hat{\mathcal{D}}_{k,j}$ as a set of discrete points on the common boarders between k-th and j-th BSs, $k, j \in [1, K], k \neq j$ which is given by

$$\hat{\mathcal{D}}_{k,j} = \left\{ \mathcal{D}_{k,j} \cap \hat{\mathcal{B}}_k \right\} \cup \left\{ \mathcal{D}_{k,j} \cap \hat{\mathcal{B}}_j \right\}, \tag{7.11}$$

where $\mathcal{D}_{k,j}$ was defined in Definition 7.2.3. We now propose a method to check the feasibility of the original problem by leveraging graph theory approaches. Let's denote an undirected graph by $G = (\mathcal{N}, \mathcal{E})$. We define $\mathcal{N}$ as a set of graph's nodes which is given by $\mathcal{N} = \{\mathbf{v}_\mathrm{I} \cup \mathcal{U} \cup \mathcal{D} \cup \mathbf{v}_\mathrm{F}\}$, where $\mathcal{U} = \{\hat{\mathbf{u}}_k,\, k \in [1, K]\}$ is a set comprising the projections of the BSs locations, and $\mathcal{D}$ is defined as

$$\mathcal{D} = \bigcup_{k,j \in [1,K], k \neq j} \hat{\mathcal{D}}_{k,j}. \tag{7.12}$$

The set of graph's edges is denoted by $\mathcal{E}$ which is given by

$$\begin{aligned} \mathcal{E} = &\{(\hat{\mathbf{u}}_k, \mathbf{v}_\mathrm{I}) |\, \mathbf{v}_\mathrm{I} \in \mathcal{A}_k,\, k \in [1, K]\} \\ &\cup \{(\hat{\mathbf{u}}_k, \mathbf{x}_{k,j}) |\, \forall \mathbf{x}_{k,j} \in \hat{\mathcal{D}}_{k,j},\, k, j \in [1, K],\, k \neq j\} \\ &\cup \{(\hat{\mathbf{u}}_k, \mathbf{v}_\mathrm{F}) |\, \mathbf{v}_\mathrm{F} \in \mathcal{A}_k,\, k \in [1, K]\}. \end{aligned} \tag{7.13}$$

We also assign a weight value to each edge of the graph corresponding to its length. Note that, the edge $(\mathbf{v}_\mathrm{I}, \hat{\mathbf{u}}_k)$ exists if the starting point $\mathbf{v}_\mathrm{I}$ lies in the coverage area of the k-th BS. Moreover, $(\hat{\mathbf{u}}_k, \mathbf{x}_{k,j})$ represents an edge between the k-th BS and all the points $(\mathbf{x}_{k,j})$ in the discretized coverage borders with its neighbour BS j.

Proposition 7.3.1. *All the edges defined in* (7.13) *satisfy the constraint* (7.9b).

Proof. See Appendix D.2. □

Since all edges of graph G satisfy SNR feasibility constraint, the trajectory optimization problem (7.9) is feasible if we can find a path from starting node $\mathbf{v}_\mathrm{I}$ to the terminal node $\mathbf{v}_\mathrm{F}$ in graph G. To this end, we employ the Dijkstra [100] algorithm with the worst-case complexity of $\mathcal{O}(|\mathcal{E}| + |\mathcal{N}| \log |\mathcal{N}|)$ which obtains a shortest path between $\mathbf{v}_\mathrm{I}$ and $\mathbf{v}_\mathrm{F}$. We denote such a solution as the base trajectory $\mathcal{V}_b = (\mathbf{v}_n^b)_{n=1}^N$. Note that, if the algorithm cannot find a path between $\mathbf{v}_\mathrm{I}$ and $\mathbf{v}_\mathrm{F}$, problem (7.9) is infeasible.

The base trajectory starts from the initial point $\mathbf{v}_\mathrm{I}$ and it goes on top of the closest BS to the $\mathbf{v}_\mathrm{I}$. The UAV then tries to reach to the terminal point by visiting the minimum number of the BSs. From one BS to another one the UAV crosses over a point inside the discretized common border of the two BSs.

An illustration of the base trajectory between the starting point and the terminal point is shown in Fig. 7.2. For ease of exhibition we consider merely two BSs. It can be seen that, the base trajectory starts from $\mathbf{v}_\mathrm{I}$ and heads towards the closest BS, which is BS1 here, and then it goes to the neighbour BS by passing over common borders between BSs. Finally, the trajectory terminates by going from BS2 in a straight line towards $\mathbf{v}_\mathrm{F}$.

We denote the BSs which are sequentially visited by the base trajectory as:

$$\mathcal{U}^b = (\hat{\mathbf{u}}_k) \,|\, \hat{\mathbf{u}}_k \in \mathcal{V}_b. \tag{7.14}$$

We also define an index set $\mathcal{I}^b = (I_{b,1}, \cdots, I_{b,K'})$, where $I_{b,j}$ is the BS's index of the j-th element in $\mathcal{U}^b$, and $K' = |\mathcal{U}^b|$. As an example, let's assume that the base trajectory visits the sequence of the BSs $\mathcal{U}^b = (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_3, \hat{\mathbf{u}}_4, \hat{\mathbf{u}}_7)$, then the index set $\mathcal{I}^b$ is given by

$$\mathcal{I}^b = (1, 3, 4, 7). \tag{7.15}$$

As it is shown in Fig. 7.2, the base trajectory is not an efficient solution since the trajectory needs to fly over the BSs to reach to the terminal point. In the next section, we propose a method to improve the base trajectory.

7.4 Trajectory Optimization

In this section we aim to find a sub-optimal and high-quality approximate solution to (7.9) by improving the base trajectory. As mentioned earlier, the base trajectory is not an efficient solution since it requires to visit the BSs to get to the terminal location. For example in Fig. 7.2, the optimal trajectory is a straight line from $\mathbf{v}_{\mathrm{I}}$ to $\mathbf{v}_{\mathrm{F}}$. To tackle this problem, in this section we aim to improve the base trajectory obtained in Section 7.3 by employing the graph theory methods.

We then construct an undirected graph $G = (\mathcal{N}, \mathcal{E})$. For ease of exposition we use the same notations as Section 7.3. The nodes of the graph is defined as follows

$$\mathcal{N} = \{\mathbf{v}_{\mathrm{I}} \cup \mathcal{U}^b \cup \mathcal{D}^b \cup \mathbf{v}_{\mathrm{F}}\}, \tag{7.16}$$

where $\mathcal{D}^b \subset \mathcal{D}$ which is defined as

$$\mathcal{D}^b = \left\{ \bigcup_{j \in [1, K'-1]} \hat{\mathcal{B}}_{I_{b,j}, I_{b,j+1}} \right\}. \tag{7.17}$$

The edges of the graph are given by

$$\begin{aligned}
\mathcal{E} = &\{(\mathbf{v}_{\mathrm{I}}, \hat{\mathbf{u}}_{I_{b,1}})\} \\
&\cup \{(\mathbf{v}_{\mathrm{I}}, \mathbf{x}_{1,2}) | L(\mathbf{v}_{\mathrm{I}}, \mathbf{x}_{1,2}) \in \mathcal{A}_{I_{b,1}}, \forall \mathbf{x}_{1,2} \in \hat{\mathcal{B}}_{I_{b,1}, I_{b,2}}\} \\
&\cup \{(\mathbf{x}_{k-1,k}, \mathbf{x}_{k,k+1}) | L(\mathbf{x}_{k-1,k}, \mathbf{x}_{k,k+1}) \in \mathcal{A}_{I_{b,k}}, \\
&\forall \mathbf{x}_{k-1,k} \in \hat{\mathcal{B}}_{I_{b,k-1}, I_{b,k}}, \forall \mathbf{x}_{k,k+1} \in \hat{\mathcal{B}}_{I_{b,k}, I_{b,k+1}}, k \in [2, K^{'} - 1]\} \\
&\cup \{(\hat{\mathbf{u}}_k, \mathbf{x}_{k,j}) | \forall \mathbf{x}_{k,j} \in \hat{\mathcal{B}}_{I_{b,k}, I_{b,j}},\ k, j \in [1, K^{'}], k \neq j\} \\
&\cup \{(\mathbf{v}_{\mathrm{F}}, \mathbf{x}_{K'-1,K'}) | L(\mathbf{v}_{\mathrm{F}}, \mathbf{x}_{K'-1,K'}) \in \mathcal{A}_{I_{b,K'}}, \\
&\quad \forall \mathbf{x}_{K'-1,K'} \in \hat{\mathcal{B}}_{I_{b,K'-1}, I_{b,K'}}\} \\
&\cup \{(\mathbf{v}_{\mathrm{F}}, \hat{\mathbf{u}}_{I_{b,K'}})\},
\end{aligned} \tag{7.18}$$

where $L(\mathbf{x}, \mathbf{y})$ is a line segment between two points $\mathbf{x}, \mathbf{y}$ which is defined as follows:

$$L(\mathbf{x}, \mathbf{y}) = \{\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}, \forall \lambda, 0 \leq \lambda \leq 1\}. \tag{7.19}$$

We also assign a weight value to each edge of the graph corresponding to its length. All the edges $(\mathbf{v}_{\mathrm{I}}, \hat{\mathbf{u}}_{I_{b,1}})$, $(\hat{\mathbf{u}}_k, \mathbf{x}_{k,j})$, $(\mathbf{v}_{\mathrm{F}}, \hat{\mathbf{u}}_{I_{b,K'}})$ are defined in a similar manner to (7.13), and similar to Proposition 7.3.1, it can be shown that constraint (7.9b) is always satisfied for any of these edges. $(\mathbf{v}_{\mathrm{I}}, \mathbf{x}_{1,2})$ is the edge between initial location $\mathbf{v}_{\mathrm{I}}$ and any points inside the discretized common borders of $I_{b,1}$-th and the $I_{b,2}$-th BS, and it exists if this edge lies inside $\mathcal{A}_{I_{b,1}}$. The edge $(\mathbf{v}_{\mathrm{F}}, \mathbf{x}_{K'-1,K'})$ is also defined similarly. The edge $(\mathbf{x}_{k-1,k}, \mathbf{x}_{k,k+1})$

represents an edge between all the points in the discretized common borders of the $I_{b,k}$-th BS and it's neighbor BSs $I_{b,k-1}, I_{b,k+1}$. Edge $(\mathbf{x}_{k-1,k}, \mathbf{x}_{k,k+1}) \in \mathcal{E}$, if the line $L(\mathbf{x}_{k-1,k}, \mathbf{x}_{k,k+1})$ lies inside $\mathcal{A}_{I_{b,k}}$, which can be efficiently checked by the following result.

Lemma 7.4.1. *Let $\mathbf{x}, \mathbf{y} \in \mathcal{A}_k$, to determine if the line $L(\mathbf{x}, \mathbf{y})$ is inside coverage area $\mathcal{A}_k$, only a limited number of points along $L(\mathbf{x}, \mathbf{y})$ need to be evaluated.*

Proof. See Appendix D.3. □

Having constructed graph G using Lemma 7.4.1, since any edges of the graph is covered by at least one BS, then constraint (7.9b) will always be satisfied if the UAV moves along any edges of the graph. So, problem (7.9) is cast as finding a shortest path between $\mathbf{v}_\mathrm{I}, \mathbf{v}_\mathrm{F}$ in graph G. Similar to Section 7.3, we use the Dijkstra algorithm to find the shortest trajectory.

7.5 Numerical Results

We consider a dense urban Manhattan-like area of size $2 \times 2\,\mathrm{km}^2$, consisting of a regular street grid and buildings. The building heights are Rayleigh distributed within the range of 5 to 70 (m) [7]. Propagation parameters for the UAV-BS links are selected as $\alpha_\mathrm{LoS} = 2.2$, $\alpha_\mathrm{NLoS} = 2.8$, $\beta_\mathrm{LoS} = 10^{-4}$, and $\beta_\mathrm{NLoS} = 10^{-4}$ according to an urban micro scenario in [101]. The UAV's path originates at $\mathbf{v}_\mathrm{I} = (300, 300, 80)$ m and terminates at $\mathbf{v}_\mathrm{F} = (1500, 1500, 80)$ m.

The cellular network consists of $K = 25$ BSs which are randomly scattered over the city. All the BSs have the same height $h_g = 20$ m and we assume that the UAV flies with the fixed altitude $h_d = 80$ m. Fig. 7.3 illustrates BSs and the coverage map where the highlighted regions represent areas where the minimum SNR constraint (7.3) is satisfied.

The base trajectory and the optimized trajectory described in Sections 7.3 and 7.4 are shown in Fig. 7.3. We have compared our method to the other graph based approaches proposed in [10] where the whole map within the flying area needs to be quantified into grids. We consider the quantization unit to be $10 \times 10\,\mathrm{m}^2$ which results in total $\Delta^2 = 4 \times 10^4$ number of nodes in the graph. It can be seen from Fig. 7.3 that our method provides the best solution in terms of the path length. The base trajectory has the maximum length among all the solutions as it is forced to visit BSs along its way to the destination.

In Fig. 7.4, we evaluate the performance of the different approaches in terms of the outage over 1000 Monte-Carlo simulations with different BS locations. The outage is defined as the amount of time the SNR constraint in (7.3) is not satisfied while following the devised trajectory. The outage of the straight trajectory between the starting and the terminal points is illustrated as well. It can be seen that constraint (7.3) is always guaranteed when the UAV moves along our proposed trajectories while there is no hard guarantee for the other approaches. In general, our graph-based trajectory performs better than the other methods.

Finally, we compare the complexity of our proposed algorithms. Our approach which requires only discretizing the coverage border of each BS into Q samples (ref Sec. 7.3)

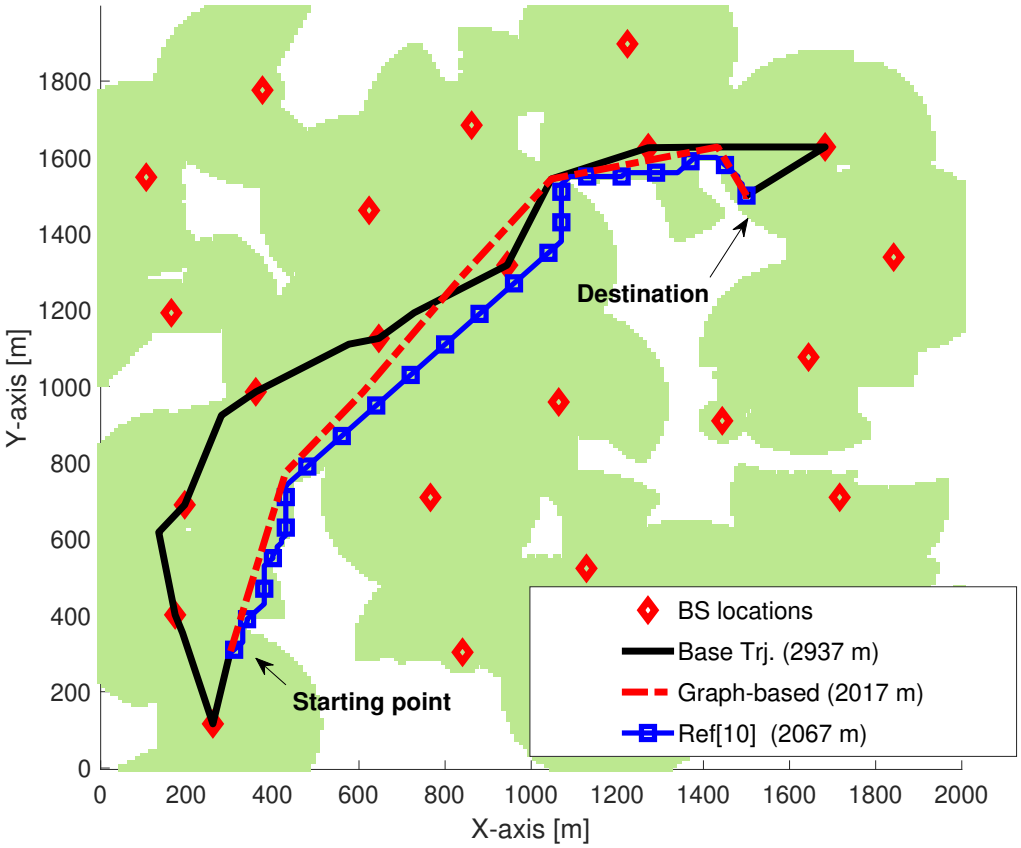


Figure 7.3 – Top view of the city, BS locations, the generated trajectories and its lengths for different algorithms. The coverage area of each BS is highlighted with green color.

which are later used as nodes in the graph. An upper bound on the complexity of our graph-based algorithm is given by $\mathcal{O}(|\mathcal{U}^b|^2Q^2 + KQ\log KQ)$. It is shown that the complexity of the optimal algorithm introduced in [10] is given by $\mathcal{O}\left(K\Delta^2 + \Delta^2\log\Delta\right)$, where Δ relates to the quantization of the map. In this simulation we assumed grid size to be $10 \times 10\,\mathrm{m}^2$ which resulted in total $\Delta^2 = 4 \times 10^4$ number of nodes. It is clear that the complexity of our proposed algorithms are considerably less than the method in [10], since $Q \ll \Delta$. Moreover, the complexity of our algorithm just increases with the number of BSs rather then the size of the flying area, since Q does not change by increasing the size of the flying area.

7.6 Conclusion

This chapter investigated the problem of UAV trajectory design under cellular connectivity constraint to minimize its trajectory length between a pre-determined initial location and

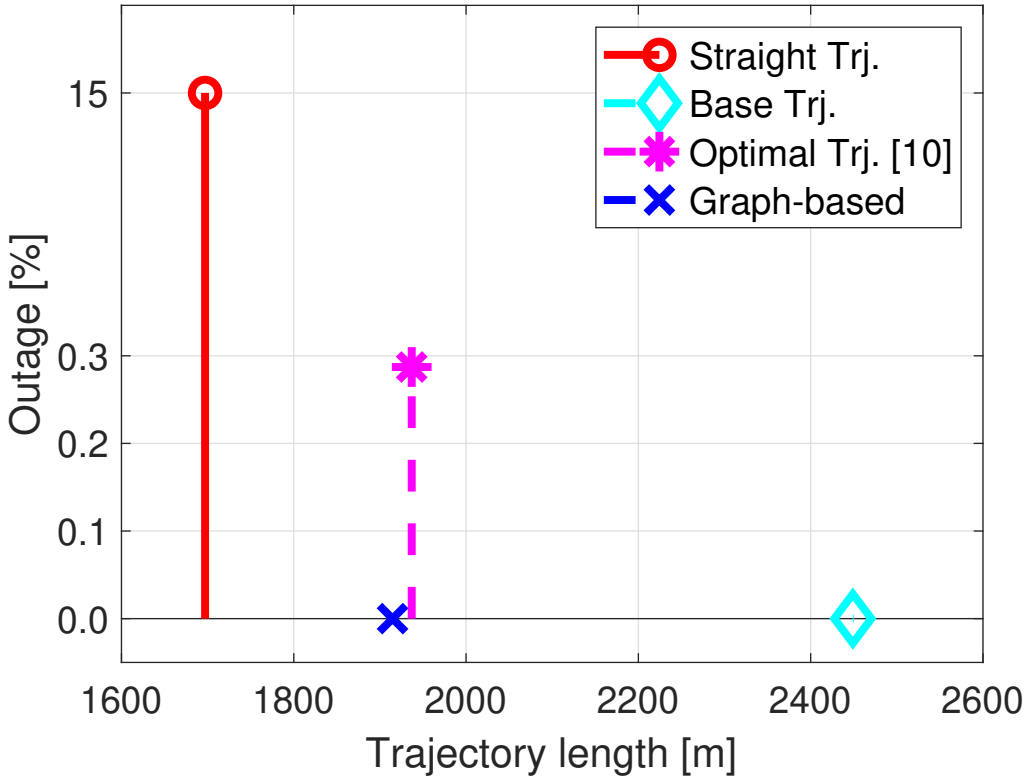


Figure 7.4 – Outage versus the trajectory length for different algorithms.

a given destination point in an urban environment. We proposed a novel approach to trajectory design that strikes a trade-off between performance (i.e. path length reduction) and complexity by exploiting the 3D map of the environment and employing the graph theory. We established a graph theory based framework to first evaluate the feasibility of the problem and then to obtain a high-quality approximate solution to the UAV trajectory design problem. The performance of the proposed solutions was validated with a set of Monte-Carlo simulations.

Chapter 8

Experimental Studies

8.1 Introduction

In this chapter, we introduce the design of the "Rebot" (Relaying Robot) as a FRAN for future wireless networks. Depending on the application, FRANs may use HAPs or LAPs to deploy the aerial BSs. Several substantive projects have looked into the aspects of providing broadband wireless access using HAPs [102],[103]. Well known examples from industry are, project loon from Google where balloons are used as an HAP [104], and Facebook's project [105] where long-endurance solar plane is used as an HAP. Typical altitude of these platforms are between 18 and 25 kilometers. Moreover, project loon claims to provide LTE connectivity to the users on the ground with connection speeds of up to 10 Mbps using their balloon relay network [104]. The advantages of using HAPs include wider coverage area, longer endurance and hence long time connectivity, which makes them suitable for applications such as providing connectivity in rural and remote areas where network infrastructure is not available.

On the other hand, FRANs based on LAPs, such as small commercial UAVs are in general faster to deploy and configure, and have lower implementation cost than the HAPs. This makes them suitable for applications like on-demand wireless services, providing temporary connectivity in unpredicted events, etc. Moreover, since UAVs fly at low-altitude they can contribute in maintaining short range LoS links to ground users which can lead to significant increase in the throughputs.

Few works have considered using micro UAVs to deploy or carry aerial BSs to provide LTE connectivity. In [106], Nokia Bell Labs demonstrated a UAV based delivery of a Nokia's small cell to a desired location [106]. The carried small cell is self powered and has a wireless backhaul. However, in this scenario UAV is only used as a means to carry the small cell to a stationary landing spot, akin to UAV based goods delivery systems. In a blog post by Nokia [107], telecommunications operator EE and Nokia have used a UAV-mounted tiny BS to provide LTE services in rural areas of Scotland. They used a satellite based backhaul link which connects the UAV-mounted BS to the core network of EE. Operator AT&T used a UAV that carries a small BS which also provides LTE services [108]. However, the UAV-mounted BS is tethered to the ground by a fiber optic and power cable. Again, a satellite based backhaul solution was used. Finally, the

ABSOLUTE project consortium has designed and analyzed a hybrid system architecture based on LTE and satellite based connectivity using Helikite platforms [109].

To the best of our knowledge, most prior work designing the UAV based FRANs hinges on the simplifying assumption that the UAV serves as a carrier and the BS as a communication payload, and their functionalities are mostly kept decoupled. However, a joint design of robotic and communication capabilities would substantially enhance the overall performance of FRANs, affecting the communication throughput by the optimal placement or trajectory design [44; 15; 110; 40; 41; 111; 48; 45; 47]. Towards this end, in this chapter we introduce the concept of **Rebot**: It not only functions as a LTE relay between the ground user and a fixed BS, but also acts as an autonomous robot by positioning itself based on suitable radio measurements, so as to maximize the throughput offered to the ground user. Note that the first results revealed in this chapter consider the case of a single ground user, while an extension to many users is currently ongoing. Key ingredients of this chapter are:

- The design of a UAV mounted LTE relay which provides end-to-end LTE connectivity between a ground user and the core network.
- The relay solution that is embedded on the UAV is based on OpenAirInterface (OAI) BS or eNB[1] [112], which is an open-source software.
- The interaction between the UAV's flight controller and a placement algorithm which exploits the radio channel measurements (provided by OAI eNB's) to autonomously place the UAV-relay so as to maximize the throughput of the user.

8.2 System Design

We consider the design of a UAV that acts as a relay between the user and a fixed eNB as shown in Figure 8.1. The UAV is used to boost the LTE connectivity to the user. The equipment, tools and the software used for designing this system are described next.

8.2.1 UAV Design

Since the experiment requires the interaction between the UAV or drone and the embedded OAI eNB, we needed a fully customized drone to enable us sending control commands to the drone and reading drone information like instantaneous drone location. For this, we have designed a customized drone by considering the required flight time and maximum payload. To build the drone we used an off-the-shelf Quad-Rotor carbon body frame with diameter of 60 cm, DJI propulsion system and PIXHAWK 2 flight controller which is an open-source flight controller and allows us to manipulate the drone by the output of autonomous placement algorithm based on the radio measurements obtained from OAI eNBs. Note that the overall weight of the drone without considering the communication parts is about 2 kg. To control and fly the drone manually (in emergency cases) we use a

[1] In this chapter we use several acronyms from 3GPP-LTE terminology without explicitly stating them.

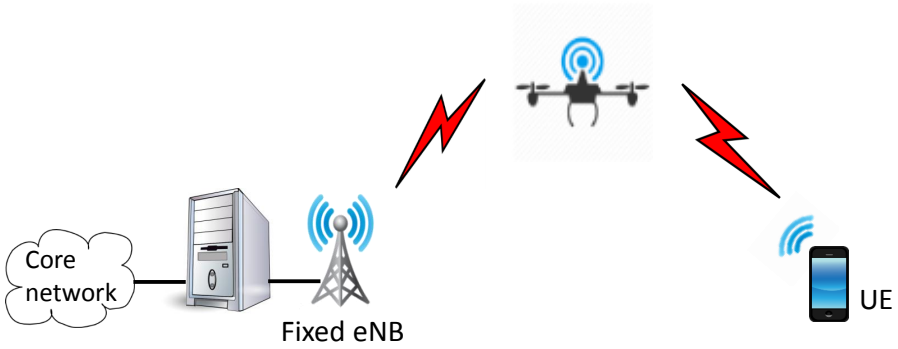


Figure 8.1 – UAV-based LTE relaying system.

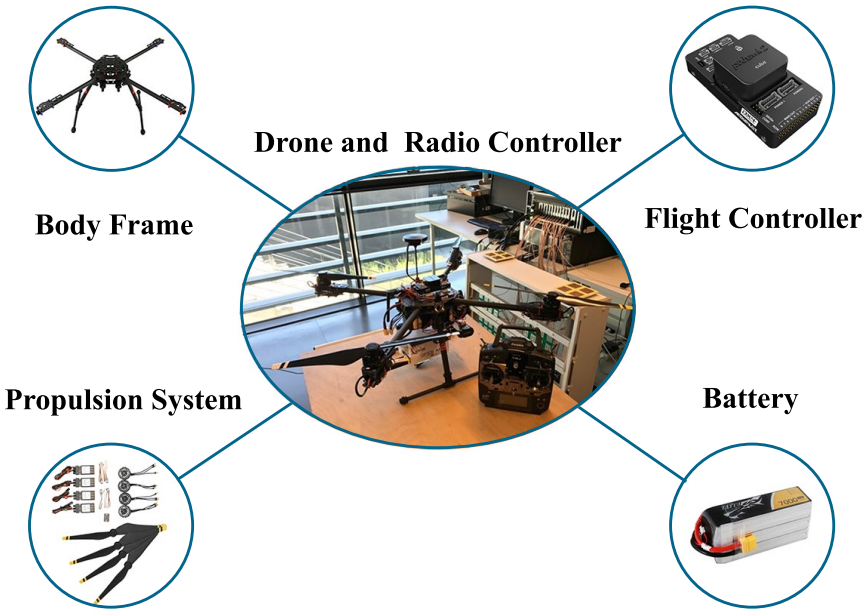


Figure 8.2 – Custom-built UAV.

Futaba T8J radio controller (RC) which is an 8 channel radio controller and works in the 2.4 GHz frequency ISM band. Different parts of the drone are shown in Figure 8.2.

8.2.2 OAI eNBs

There are in total two OAI eNBs used in this setup, one used as a fixed eNB on the ground and another is mounted on the UAV which is used as a relay. The OAI's eNB software is compliant with 3GPP LTE standards, and runs on a commodity x86 based Linux computing equipment. Both eNBs are configured to run in TDD mode in LTE frequency band 38 where EURECOM has the license to transmit. The UAV to ground user link and the backhaul link between UAV and the fixed eNB use orthogonal 5 MHz bandwidth channels within band 38. We use USRP platform [113] along with a custom designed power amplifier by EURECOM as the RF front end. The maximum transmission power of the eNB is 23 dBm.

The choice of UAV's eNB configuration has a direct impact on the design of the UAV. Higher bandwidth configurations requires generally higher computing power which will limit the flying time of the UAV. Hence, there is an interesting trade off between complexity, throughput, weight and the power consumption of the eNB solution that is mounted on the UAV.

8.2.3 Autonomous Placement

The autonomous placement software allows the UAV to position itself to maximize the throughput to the ground user. This requires communication between the placement

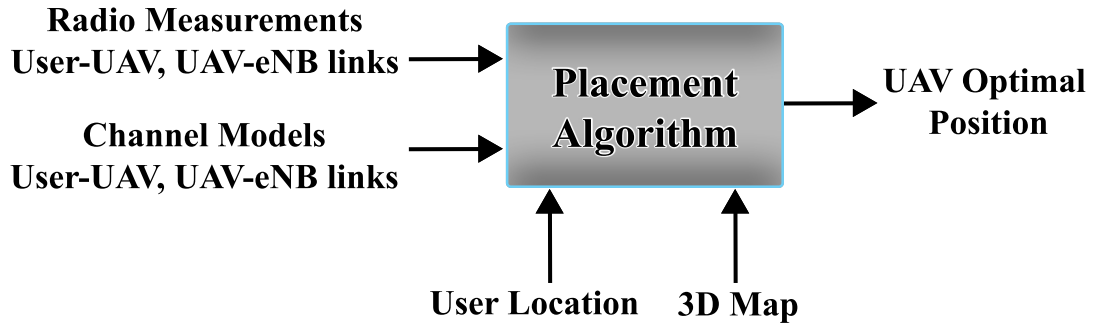


Figure 8.3 – UAV placement algorithm.

algorithm and the flight controller. The optimal UAV position is updated according to the instantaneous user location and then is sent to the UAV flight controller by the placement algorithm. The block-diagram of the autonomous placement algorithm is depicted in Figure 8.3.

Generally, these algorithms depend on wireless channel parameters that vary slowly with time such as pathloss or shadowing. In some scenarios, the wireless channel parameters are estimated beforehand in an offline fashion and then given to the algorithm, whereas in some others the UAV has to learn them on the fly by making radio measurements. Our system design allows us to implement both types of algorithms.

The placement algorithm can be implemented in an on-board computer along with OAI eNB/relay on the UAV or at a ground station. If the algorithm is computed at a ground station, the new coordinate is sent to the UAV by using the backhaul link between the fixed eNB and the UAV. Learning wireless channel parameters on the fly by making radio measurements is a computationally expensive task. In such scenarios, it is therefore favorable to implement the algorithm at a ground station where computing cost and power consumption is not an issue as opposed to on the UAV.

For the experiment presented in this work, we use a placement algorithm which has access to the channel parameters that are estimated beforehand in an offline fashion. The algorithm is described in the next section.

8.3 UAV Placement

The autonomous placement algorithm relies on the fact that information regarding the 3D map of the environment, and the wireless channel parameters is known in advance. The 3D map can be obtained from either photogrammetry or radio based reconstruction approaches (see Chapter 6), while the wireless channel parameters needs to be estimated. The channel model and the method for estimating the parameters involved are explained next.

8.3.1 Channel Parameter Estimation

We use the same channel model for both UAV-eNB and UAV-user links as described in Section 2.2.1. To learn the channel parameters, a set of measurements was collected by the UAV from the ground nodes. The measurements are labeled as LoS or NLoS by leveraging the 3D map information of the environment. having classified each measurement into LoS or NLoS, then the channel parameters can be learned easily from the measurements by resorting to a standard LS technique (see Section 4.3.1). The estimated parameters along with the 3D map are then used in the placement algorithm.

8.3.2 Placement Algorithm

The aim of the UAV placement algorithm is to find the optimal UAV position that maximizes the downlink throughput of the ground user. However, the throughput in a LTE system depends not only on channel gains but on many parameters such as scheduling, modulation and coding, etc., which makes the problem intractable. Therefore we resort to an approximation where we try to find a UAV position that maximizes the minimum of the average channel gains of the UAV-user and UAV-eNB links. This serves as a good approximation as we use a decode-and-forward type of relay protocol on the UAV, and the transmission powers of the UAV and the fixed eNB are kept same in our system. Note that the placement algorithm depends on the channel gains which are defined according to (2.1), hence, we consider channel parameters that vary slowly with time. This assumption is justifiable as the time scale of UAV mobility is much larger than the fast fading channel variations. Before presenting the details of the algorithm, we introduce some notations.

The downlink channel gains for the UAV-User and eNB-UAV links are denoted by γ_u and γ_b, respectively. The user's coordinate is denoted by $\mathbf{u}$ while that of the fixed eNB is denoted by $\mathbf{x}_b$. We assume that the UAV can fly over a selective search area in 3D which is denoted by $\mathcal{C}$. The altitude of this search area is restricted to be in between $h_{\min}$ and $h_{\max}$ with the value of $h_{\min}$ is greater than the heights of all the buildings where the experiment is conducted. The UAV placement algorithm then solves

$$\max_{\mathbf{v}\in\mathcal{C}} \min\left\{\mathrm{E}\left[\gamma_u\right], \mathrm{E}\left[\gamma_b\right]\right\}, \tag{8.1}$$

where $\mathbf{v}$ represents the coordinate of the UAV and the expectation is taken over the shadowing coefficient which has a zero-mean Gaussian distribution. From now we use the optimal UAV position in the sense of (8.1). While the ground eNB's coordinates are fixed, the coordinates of the user and the UAV are obtained using GPS receivers which are embedded in both devices. The placement problem (8.1) can be solved similar to the approach explained in Section 3.3 by exploiting the 3D map which contains the information regarding LoS/NLoS nature of the channels, and the coordinates $\mathbf{u}, \mathbf{v}$ and $\mathbf{x}_b$.

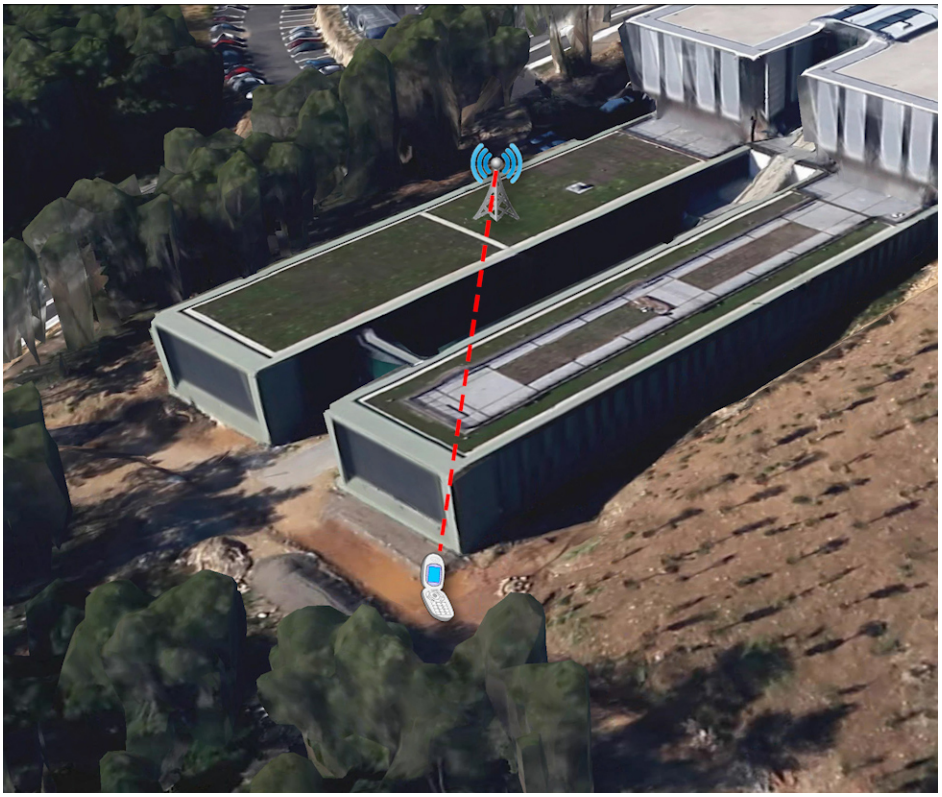


Figure 8.4 – Outdoor experiment setup with only fixed eNB.

8.4 Experimental Results

We conducted our experiments in EURECOM's premises. The ground user has a commercial Moto G(3rd gen) mobile handset. The fixed eNB's antenna is mounted on a mast situated on the top of a building block, while the user is located on the ground. Both the fixed eNB and the UAV are equipped with a single vertically polarized dipole antenna. The user is typically obstructed by the building, hence, always in NLoS with respect to the fixed eNB. The experimental setup is shown in Figures 8.4 and 8.5. When using UAV as a relay, its position is obtained using the UAV placement algorithm described in Section 8.3. For applying the algorithm, we first need to estimate the channel parameters based on the measurements that are collected in the environment where the experiment is conducted.

Since the pathloss parameters have strong dependence on the LoS or NLoS nature of the channel, we make measurements in both scenarios. Figures 8.6 and 8.7 show the channel gains as a function of distance between the transmitter and the receiver in LoS and NLoS scenarios, respectively. The channel gains are obtained from the measured reference signal received power (RSRP) values. The corresponding best-fit path loss parameters can be obtained as described in Section 8.3.1, and they are given in Table 8.1. Note that the channel model presented here does not correspond to a general wireless

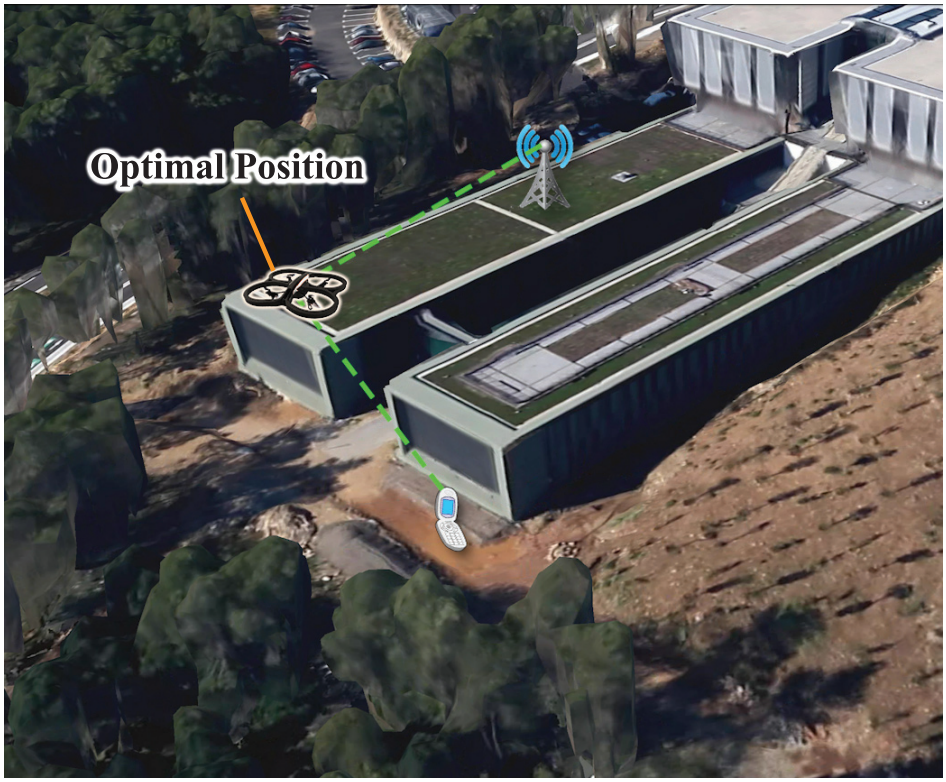


Figure 8.5 – Outdoor experiment setup with UAV relay.

Parameter	LoS	NLoS
α	2.34	3.75
ß	-58	-51.2

Table 8.1 – Pathloss parameters.

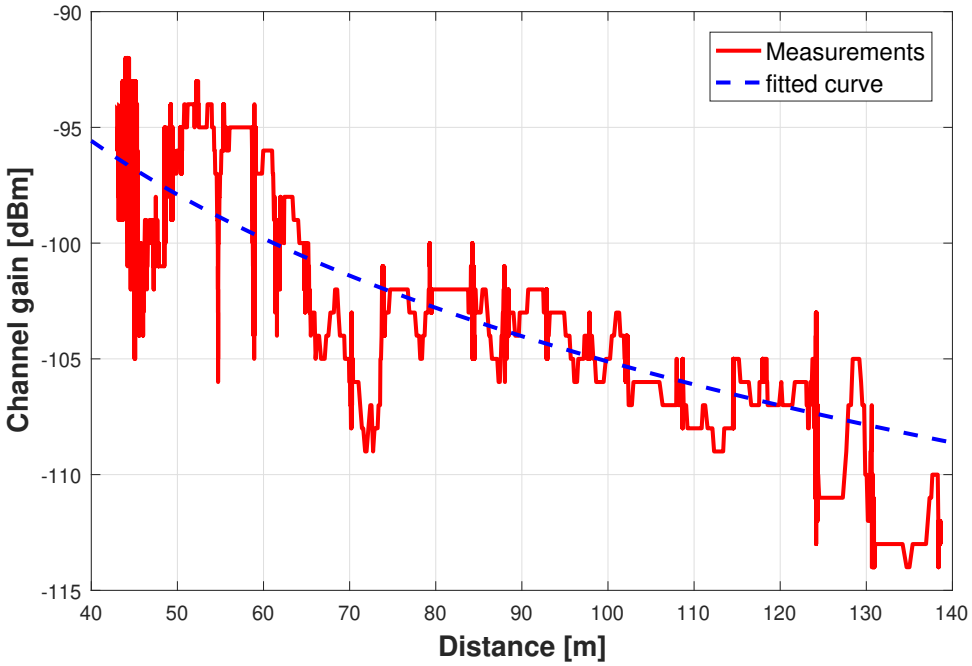


Figure 8.6 – Channel measurements for LoS scenario.

channel and it is highly dependent on our system setup. To get accurate wireless channel models, in addition to the distance one needs to take into account the height of the transmitter, its antenna orientation and gain, etc. The model presented here can only be used in this specific scenario. Study of general channel model for UAVs is itself an interesting problem [23], which is out the of scope of this thesis.

The estimated parameters are then fed to the algorithm which predicts the optimal location for the UAV. In Figure 8.8, we compare the downlink throughput of the user in scenarios shown in Figures 8.4 and 8.5, respectively. The downlink throughput is measured using the iperf application, which generates UDP traffic from the core network to the user. If the user moves to a new location, the position of the UAV is updated according to the placement algorithm. This is demonstrated in our recent demo [114].

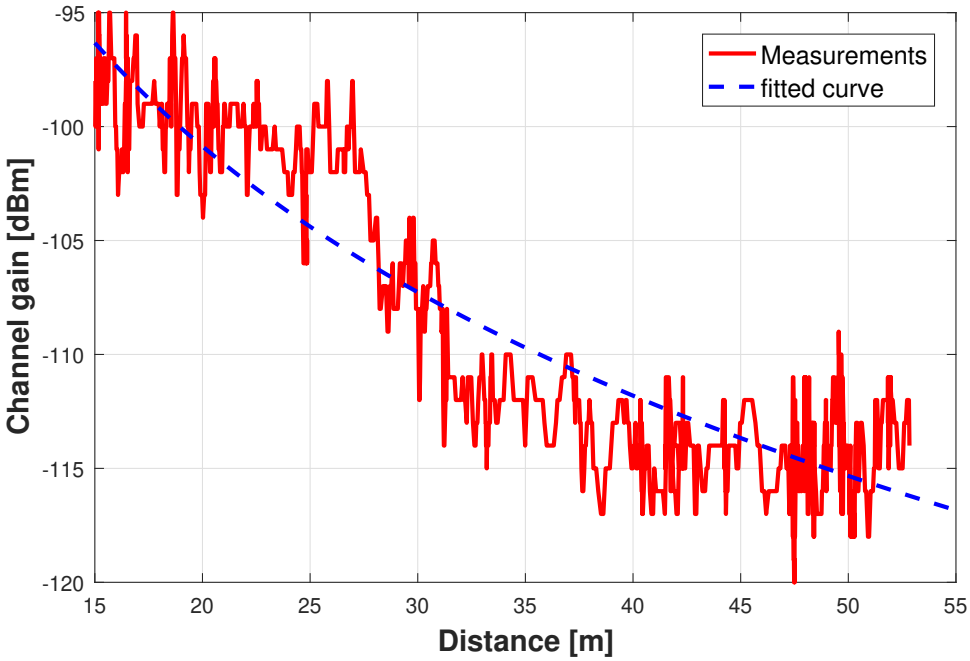


Figure 8.7 – Channel measurements for NLoS scenario.

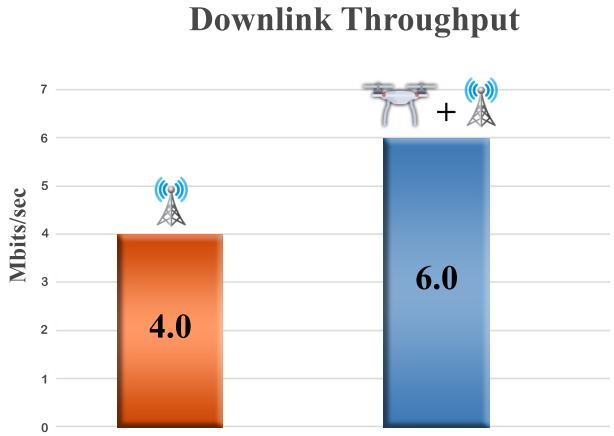


Figure 8.8 – Instance of throughput comparison at a given user location.

8.5 Conclusion and Discussion

In this chapter, we have illustrated the design of a custom-built UAV relay based on OAI, and then presented experimental results related to throughput improvement offered to a ground user by using this relay. Although, these are initial results based on a single user and single UAV scenario, we believe that this experiment is an initial step towards building more advanced UAV-based LTE relay networks. We have used an autonomous placement algorithm which updates the UAV position in real time based on the user location, 3D map of the environment and a wireless channel model. The output of the placement algorithm often results in a UAV position where it has LoS links to the user and the fixed eNB.

While experimenting with this UAV relay prototype, we have faced some interesting issues both in the system design and algorithm development, which we intend to address in our future works. They are presented below.

8.5.1 Design Improvement

In the current prototype we have used a vertically polarised dipole antenna for the UAV relay. However, the choice of the antenna and how to optimally mount it on the UAV is not considered in the design. It is well known that the radiation pattern and the polarization losses depends on the orientation of the antenna, and also a conducting surface near the antenna (carbon frame in the case of UAV) might change its radiation pattern. Knowing the antenna pattern and the possible polarization losses is essential in coming up with channel models based on the measurements done by this UAV.

8.5.2 Channel Models

Although the UAV placement algorithm used in this chapter can be adapted to any channel model, further work is needed to analyze the UAV-user and UAV-eNB links. The channel measurements and the model fitting should take into account the impact of UAV height, its antenna orientation with respect to the receiver or transmitter i.e., UAV yaw angle etc., for example as done in [115].

Chapter 9

Conclusion

The focus of this thesis was on the investigation of the current works and technologies of the UAV related communications systems and develops novel methods for both placement and trajectory design of a UAV as a flying radio by capitalizing on the city 3D map information.

In the first part of this dissertation, we discussed the different aspects of trajectory design and placement of UAV-assisted wireless networks by exploiting a 3D map of the city. While 3D maps are rich in information, hence difficult to be used directly in UAV placement and path planning because of the highly irregular behavior of radio blockage, especially in urban environments. To solve this issue, we introduced a map compression technique which relies on converting map data to build a reliable node location dependent LoS probability model which allows us to tackle the problem with standard optimization tools. The trajectory optimization was then combined with a node scheduling algorithm. The advantages of the map compression method are illustrated in the context of intelligent IoT data harvesting. Furthermore, we discussed the problem of optimal placement for a UAV relay in LTE networks which is a special case of the trajectory design problem.

We also investigated the problem of learning the channel parameters and localizing the ground nodes from the UAV-borne measurement by leveraging the 3D map of the city. For the further improvement of the channel parameters estimation and the node localization accuracy, we devised a trajectory for the UAV to intelligently collect radio measurements from the ground nodes. Moreover, an algorithm was proposed to construct the 3D map of the city (as a by-product) from only radio measurements collected by the UAV while the UAV is accomplishing its mission and flying over the network.

The second part of this thesis focuses on UAV trajectory design under cellular connectivity constraints. A key challenge of this problem lies in the design of trajectory which, while allowing the completion of the UAV mission, can guarantee reliable cellular connectivity all along the path. We proposed a novel approach for trajectory design using a coverage map that can be obtained with a combination of a 3D map of the environment and radio propagation models. Leveraging on the convexity of sub-regions within the coverage map, we proposed a low-complexity graph-based algorithm which has been shown to achieve quasi-optimal performance at a fraction of the computational cost of known optimal methods.

Finally, we discussed practical prototype realizations. We introduced the design of the "Rebot" (Relaying Robot) for future wireless networks, which comprises a customized integrated UAV relay and its communication layer based on OpenAirInterface to provide an end-to-end LTE connectivity between ground users and the core network. It also functions as an autonomous robot capable of positioning itself in real-time at a throughput maximizing location.

Despite extensive studies on UAV-based wireless communications, there exist many outstanding issues in this regard which have been overlooked. Key open problems in the context of UAV communication systems lie in different areas such as accurate radio channel modeling for UAV communications, active trajectory design, cooperative learning, interference management, etc. For instance, while a substantial number of studies have been conducted relating to the UAV trajectory design in wireless networks, there are still several open issues including obstacle aware and collision-free path planning, resource-constrained trajectory optimization by taking into account the UAV dynamics model, etc. Another important aspect of UAV communications is decentralized learning and control of a fleet of UAVs for cooperative task completion in scenarios in which using more than one UAV is required (e.g. accommodating huge flash crowds, search and rescue in a large scale, etc.).

Appendices

Appendix A

Chapter 3 Appendices

A.1 The derivation of the average channel gain

The average channel gain of the link between the drone and the k-th ground node in the n-th time slot is given by

$$\mathrm{E}[\gamma_k[n]] = p_k[n]\gamma_{k,\mathrm{LoS}}[n] + (1 - p_k[n])\gamma_{k,\mathrm{NLoS}}[n], \tag{A.1}$$

where $p_k[n]$ denotes the LoS probability, $\gamma_{k,\mathrm{LoS}}[n]$ and $\gamma_{k,\mathrm{NLoS}}[n]$, respectively, denote the channel gain in LoS and NLoS propagation segments. Expanding (A.1) we have

$$\begin{aligned}\mathrm{E}[\gamma_k[n]] &\overset{(a)}{=} p_k[n]\frac{\beta_{\mathrm{LoS}}}{(d_k[n])^{\alpha_{\mathrm{LoS}}}} + (1 - p_k[n])\frac{\beta_{\mathrm{NLoS}}}{(d_k[n])^{\alpha_{\mathrm{NLoS}}}} \\ &\overset{(b)}{=} \left(\frac{(d_k[n])^{(A-1)\alpha_{\mathrm{LoS}}} - B}{1 + \exp\left(-a_k\theta_k + b_k\right)} + B\right)\frac{\beta_{\mathrm{LoS}}}{(d_k[n])^{\alpha_{\mathrm{NLoS}}}},\end{aligned} \tag{A.2}$$

where step (a) holds by substituting the values of $\gamma_{k,\mathrm{LoS}}[n]$ and $\gamma_{k,\mathrm{NLoS}}[n]$ form (2.1) into (A.1), and (b) is obtained by substituing (3.7), where $B = \frac{\beta_{\mathrm{NLoS}}}{\beta_{\mathrm{LoS}}}$, $A = \frac{\alpha_{\mathrm{NLoS}}}{\alpha_{\mathrm{LoS}}} \geq 1$, and $d_k[n] = \sqrt{z^2 + r_k^2[n]}$ is the distance between the k-th ground node and the drone. Note that, in order to ease the notation, the average random shadowing is assumed absorbed into β_s in (A.2) i.e., $\beta_s \triangleq \beta_s exp(\sigma_s^2/2)$, $s \in \{\mathrm{LoS}, \mathrm{NLoS}\}$.

A.2 Proof of Lemma 3.2.1

By proving that the Hessian of the function $h \triangleq h(x, y)$, is a positive semi-definite (PSD) matrix, we prove the convexity of h. We start by considering the Hessian of function $\hat{h} \triangleq \hat{h}(x, y)$

$$\nabla^2\hat{h} = \begin{bmatrix} \frac{f_{xx}f - f_x^2}{f^2} & 0 \\ 0 & \frac{g_{yy}g - g_y^2}{g^2} \end{bmatrix} \geq 0, \tag{A.3}$$

where $f \triangleq f(x)$, $g \triangleq g(x)$, $f_x \triangleq \frac{\partial f}{\partial x}$, $g_y \triangleq \frac{\partial g}{\partial y}$, $f_{xx} \triangleq \frac{\partial^2 f}{\partial x^2}$, and $g_{yy} \triangleq \frac{\partial^2 g}{\partial y^2}$. Since $\hat{h}$ is convex, $\nabla^2 \hat{h}$ is PSD and has non-negative diagonal elements. Hence, for $f > 0$, $g > 0$,

$$f_{xx} \geq \frac{f_x^2}{f}, g_{yy} \geq \frac{g_y^2}{g}. \tag{A.4}$$

Also, it can easily be deduced that

$$f_{xx} g \geq 0,\ g_{yy} f \geq 0. \tag{A.5}$$

The hessian of $h \triangleq h(x, y)$ is given by

$$\nabla^2 h = \begin{bmatrix} \frac{g^2\left(f_{xx}f - f_x^2\right) + f_{xx}g}{(1+fg)^2} & \frac{f_x g_y}{(1+fg)^2} \\ \frac{f_x g_y}{(1+fg)^2} & \frac{f^2\left(g_{yy}g - g_y^2\right) + g_{yy}f}{(1+fg)^2} \end{bmatrix}.$$

If $\det(\nabla^2 h) \geq 0$ and $\mathrm{tr}(\nabla^2 h) \geq 0$, then $\nabla^2 h$ is PSD [116]. Let us rewrite $\nabla^2 h$ as a summation of two matrices $\nabla^2 h = \mathbf{M}_1 + \mathbf{M}_2$, where

$$\mathbf{M}_1 = \begin{bmatrix} \frac{g^2\left(f_{xx}f - f_x^2\right)}{(1+fg)^2} & 0 \\ 0 & \frac{f^2\left(g_{yy}g - g_y^2\right)}{(1+fg)^2} \end{bmatrix},$$

$$\mathbf{M}_2 = \begin{bmatrix} \frac{f_{xx}g}{(1+fg)^2} & \frac{f_x g_y}{(1+fg)^2} \\ \frac{f_x g_y}{(1+fg)^2} & \frac{g_{yy}f}{(1+fg)^2} \end{bmatrix}.$$

Since $\det(\nabla^2 h)$ is a 2×2 matrix, we can write it as [117],

$$\det(\nabla^2 h) = \det(\mathbf{M}_1) + \det(\mathbf{M}_2) + \mathrm{tr}(\mathbf{M}_1^\dagger \mathbf{M}_2),$$

where $\mathbf{M}_1^\dagger$ is the adjugate of $\mathbf{M}_1$. From (A.3), it can easily be shown that $\det(\mathbf{M}_1) \geq 0$. Also, using (A.4) we can see that

$$\begin{aligned} \det(\mathbf{M}_2) &= (1+fg)^{-2}\left[\left(f_{xx}f\right)\left(g_{yy}g\right) - f_x^2 g_y^2\right] \\ &\geq 0. \end{aligned}$$

Finally, from (A.3) and (A.5), we have

$$\begin{aligned} \mathrm{tr}(\mathbf{M}_1^\dagger \mathbf{M}_2) &= f^2\left(g_{yy}g - g_y^2\right) f_{xx} g + g^2\left(f_{xx}f - f_x^2\right) g_{yy} f \\ &\geq 0. \end{aligned}$$

Therefore, we can conclude that $\det(\nabla^2 h) \geq 0$. It remains to prove that $\mathrm{tr}\left(\nabla^2 h\right) \geq 0$. Using (A.3) and (A.5), we can see that the diagonal elements of $\nabla^2 h$ are positive and hence the $\mathrm{tr}\left(\nabla^2 h\right) \geq 0$. Consequently, we can see $\nabla^2 h$ is PSD, which concludes the proof.

A.3 Proof of Proposition 3.2.1

Let $f(x) = \frac{1}{1+x}$, $g(y) = \frac{1}{y}$, $h(d) = 1/d^{\lambda}$ and $q(x,y) = f(x)g(y) + \tau$, $\tau \geq 0$. For positive $f \triangleq f(x)$ and $g \triangleq g(y)$, since $\log(f\,g)$ is strictly convex, using Lemma 3.2.1, we can infer that $\log(q(x,y))$ is also strictly convex. Finally, from the above arguments we can easily see that the function

$$\hat{c}(x,y,d) = \log\left(q(x,y)\,h(d)\right),\ k \geq 0$$

is also strictly convex.

The Hessian of $\hat{c}(x,y,d)$ is given by

$$\nabla^2\hat{c} = \begin{bmatrix} \frac{\left(q_{xx}q - q_x^2\right)}{q^2} & \frac{(q_{xy}q - q_x q_y)}{q^2} & 0 \\ \frac{(q_{yx}q - q_x q_y)}{q^2} & \frac{\left(q_{yy}q - q_y^2\right)}{q^2} & 0 \\ 0 & 0 & \frac{\left(h_{dd}h - h_d^2\right)}{h^2} \end{bmatrix}, \tag{A.6}$$

where $q \triangleq q(x,y)$ and $h \triangleq h(d)$. q_x, q_{xy} stand for the partial derivative of q and are defined as $q_x = \frac{\partial q}{\partial x}$, $q_{yx} = q_{xy} = \frac{\partial^2 q}{\partial x \partial y}$. $q_{xx}, q_{yy}, h_d, h_{dd}$ also are defined similarly. Since $\nabla^2\hat{c}$ is a positive definite (PD) and symmetric matrix, it has positive diagonal elements. Hence,

$$q_{xx} > \frac{q_x^2}{q} > 0. \tag{A.7}$$

Since $h > 0$, from (A.7) we have

$$h\,q_{xx} > 0. \tag{A.8}$$

Moreover, since $\log(f\,g)$ is strictly convex, we can write

$$f_{xx}f > f_x^2,\ g_{yy}g > g_y^2. \tag{A.9}$$

Using the above results, we now prove that the function

$$c(x,y,d) = \log\left(1 + q(x,y)\,h(d)\right)$$

is convex. The Hessian of $c \triangleq c(x,y,d)$ is

$$\nabla^2 c = \frac{1}{(1+q\,h)^2}\left(\mathbf{P} + \mathbf{Q}\right),$$

where

$$\mathbf{P} = (q\,h)^2\,\nabla^2\hat{c},$$

$$\mathbf{Q} = \begin{bmatrix} q_{xx}h & q_{xy}h & q_x h_d \\ q_{yx}h & q_{yy}h & q_y h_d \\ q_x h_d & q_y h_d & q\,h_{dd} \end{bmatrix}.$$

Matrix $\mathbf{P}$ is PD since $\nabla^2\hat{c}$ is PD and $q, h > 0$. In order to show that the Hessian matrix $\nabla^2 c$ is PD, we need to prove that $\mathbf{Q}$ is PD as the sum of two PD matrices is PD. According to [118], if all upper left $n \times n$ determinants of a symmetric matrix are positive, the matrix is PD. Matrix $\mathbf{Q}$ is symmetric, since q_{xy} and q_{yx} are equal to $f_x\,g_y$.

We start from the upper left 1×1 determinants of $\mathbf{Q}$ which equals to $q_{xx}h$. It follows from (A.8), that $q_{xx}h > 0$. Now, we proceed to show that the determinant of upper left 2×2 matrix of $\mathbf{Q}$ is positive. So, we can write

$$\frac{\det(\mathbf{Q}_{2\times 2})}{h^2} = \left(q_{xx}\, q_{yy} - q_{xy}^2\right) \tag{A.10}$$

$$\overset{(a)}{=} (f_{xx}f)(g_{xx}g) - f_x^2 g_y^2 \tag{A.11}$$

$$\overset{(b)}{>} 0, \tag{A.12}$$

where $\mathbf{Q}_{2\times 2}$ denotes the upper left 2×2 matrix of $\mathbf{Q}$, (a) is obtained by substituting $q_{xx} = f_{xx}g$, $q_{yy} = g_{yy}f$, $q_{xy} = f_x\, g_y$ in (A.10) and step (b) follows from (A.9). Then, we compute

$$\det(\mathbf{Q}) = h_d^2\,(h\,m) + h_{dd}h\,(h\,q\,p)\,,$$

where $m = 2q_{xy}q_xq_y - q_{xx}q_y^2 - q_{yy}q_x^2$, $p = q_{xx}q_{yy} - q_{xy}^2$. From (A.9), it can be shown that $m < 0$. From the convexity of $\hat{c}(x, y, d)$, by computing the determinant of upper left 2×2 matrix of $\nabla^2\hat{c}$ and performing some algebraic reductions we obtain

$$\begin{aligned} &m + q\,p > 0 \\ &\Rightarrow h\,q\,p > -h\,m > 0. \end{aligned} \tag{A.13}$$

Also, since $\log(h)$ is strictly convex, we can write

$$h_{dd}h > h_d^2. \tag{A.14}$$

Therefore, according to (A.13), (A.14), it can be seen that $\det(\mathbf{Q}) > 0$. Since all upper left $n \times n$ determinants of $\mathbf{Q}$ are positive, we conclude that the matrix $\mathbf{Q}$ is PD. Hence, $\nabla^2 c$ is also PD.

Appendix B

Chapter 5 Appendices

B.1 Proof of convergence for multi-user localization

As mentioned earlier, for the multi-user case we use the block coordinate descent method which is an iterative algorithm. In each iteration we fix all the user location estimates except one. Therefore, in each iteration of this algorithm the problem is recast as a single user case. For ease of exposition we merely assume one iteration for the PSO algorithm $I = 1$. For this, to avoid notation overload, we drop the superscript indicating the PSO algorithm iteration for each particle.

Assuming we are at q-th iteration of the block coordinate descent, for the first user it can be written

$$\begin{aligned} \mathrm{SSE}_1^{*(q)} &:= \min_{j\in[1,C]} \mathrm{SSE}_1^*(\mathbf{c}_j), \\ \mathbf{u}_1^* &:= \arg\min_{\mathbf{c}_j\in C} \mathrm{SSE}_1^*(\mathbf{c}_j), \end{aligned} \tag{B.1}$$

where $\mathrm{SSE}_1^{*(q)}$ is the minimum cost at iteration q and is solved just for the first user, and $\mathbf{u}_1^*$ is the corresponding user location estimate. $\mathrm{SSE}_1^*(\mathbf{c}_j)$ is defined as follows

$$\mathrm{SSE}_1^*(\mathbf{c}_j) = \min_{\boldsymbol{\theta}_{\mathrm{LoS}},\boldsymbol{\theta}_{\mathrm{NLoS}}} \mathrm{SSE}_1(\mathbf{c}_j), \tag{B.2}$$

where $\mathrm{SSE}_1(\mathbf{c}_j)$ is the cost function for particle j by fixing all the users' location except the first user which equals to

$$\begin{aligned} \mathrm{SSE}_1(\mathbf{c}_j) = &\sum_{s\in\{\mathrm{LoS},\mathrm{NLoS}\}} \sum_{n\in\mathcal{M}_{s,1,j}} \left| g_{n,1} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{c}_j^{(i)}) \right|^2 + \\ &\sum_{k=2}^{K} \sum_{s\in\{\mathrm{LoS},\mathrm{NLoS}\}} \sum_{n\in\widehat{\mathcal{M}}_{s,k}} \left| g_{n,k} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{u}_k^*) \right|^2, \end{aligned} \tag{B.3}$$

where $\mathbf{u}_k^*$ is the k-th user location estimate available from the last iteration $(q-1)$. $\widehat{\mathcal{M}}_{s,k}$ is a set of time indices of measurements collected from user k which are at segment s. In general, $\mathrm{SSE}_k(\mathbf{c}_j)$ has the following form

$$\mathrm{SSE}_k(\mathbf{c}_j) = \sum_{s\in\{\mathrm{LoS,NLoS}\}} \sum_{n\in\mathcal{M}_{s,k,j}} \left| g_{n,k} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{c}_j^{(i)}) \right|^2 + \sum_{m=1,m\neq k}^{K} \sum_{s\in\{\mathrm{LoS,NLoS}\}} \sum_{n\in\widehat{\mathcal{M}}_{s,m}} \left| g_{n,m} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{u}_m^*) \right|^2, \tag{B.4}$$

Now by having estimated the first user location, we then proceed to find the second user position as follows

$$\mathrm{SSE}_2^{*(q)} := \min_{j\in[1,C]} \mathrm{SSE}_2^*(\mathbf{c}_j), \tag{B.5}$$

and $\mathbf{u}_2^*$ is computed similar to (B.1). It can be written now

$$\mathrm{SSE}_1^{*(q)} \overset{(a)}{\geq} \mathrm{SSE}_2^{*(q)}. \tag{B.6}$$

Inequality (B.6) holds since the PSO algorithm guarantees improvement in the cost function. In a similar manner, for all the users it can be shown that

$$\mathrm{SSE}_1^{*(q)} \geq \mathrm{SSE}_2^{*(q)} \geq \mathrm{SSE}_3^{*(q)} \geq \cdots \geq \mathrm{SSE}_K^{*(q)}. \tag{B.7}$$

Then by proceeding to the next iteration $(q+1)$ we have

$$\mathrm{SSE}_1^{*(q)} \geq \mathrm{SSE}_1^{*(q+1)}. \tag{B.8}$$

And due to the fact that the SSE is lower bounded by zero then the convergence is proved.

B.2 Derivation of FIM

Following from (5.2), each measurement $h_{n,k,s}$ is modeled as a Gaussian random variable with $\mathcal{N}(\mu_{n,k,s}, \sigma_s^2)$ where $\mu_{n,k,s} = ß_s - \alpha_s \log_{10} \|\mathbf{v}[n] - \mathbf{u}_k\|^2$. Then the PDF of each measurement equals to

$$f_{n,k,s} = \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\left(-\frac{(h_{n,k,s} - \mu_{n,k,s})^2}{2\sigma^2} \right) \tag{B.9}$$

Now we compute the derivative of likelihood as follows

$$\frac{\partial \mathcal{L}_s}{\partial \phi_s} = \sum_{n=1}^{N} \sum_{k\in\mathcal{K}_{n,s}} \frac{\partial \log\, f_{n,k,s}}{\partial \phi_s} \tag{B.10}$$

where

$$\begin{aligned}\frac{\partial \log f_{n,k,s}}{\partial \phi_s} =& \frac{1}{\sigma_s^2}\begin{bmatrix} -2\log_{10}\|\mathbf{v}[n]-\mathbf{u}_k\| \\ 1 \\ \frac{-\alpha_s(x_k-x[n])}{\|\mathbf{v}[n]-\mathbf{u}_k\|^2\log 10} \\ \frac{-\alpha_s(y_k-y[n])}{\|\mathbf{v}[n]-\mathbf{u}_k\|^2\log 10}\end{bmatrix}(h_{n,k,s}-\mu_{n,k,s}) \\ \triangleq& \begin{bmatrix} \ell^{\alpha}_{n,k,s} \\ 1 \\ \ell^{x}_{n,k,s} \\ \ell^{y}_{n,k,s}\end{bmatrix}.\end{aligned} \tag{B.11}$$

Then the FIM is given by

$$\mathbf{F}_{N,s} = \mathrm{E}\left[\frac{\partial \mathcal{L}_s}{\partial \phi_s}\frac{\partial \mathcal{L}_s}{\partial \phi_s}^{\mathrm{T}}\right] = \sum_{n=1}^{N}\sum_{k\in\mathcal{K}_{n,s}}\mathbf{H}_{n,k,s}, \tag{B.12}$$

where

$$\mathbf{H}_{n,k,s} = \begin{bmatrix} (\ell^{\alpha}_{n,k,s})^2 & \ell^{\alpha}_{n,k,s} & \ell^{\alpha}_{n,k,s}\ell^{x}_{n,k,s} & \ell^{\alpha}_{n,k,s}\ell^{y}_{n,k,s} \\ (\ell^{\alpha}_{n,k,s}) & 1 & \ell^{x}_{n,k,s} & \ell^{y}_{n,k,s} \\ \ell^{x}_{n,k,s}\ell^{\alpha}_{n,k,s} & \ell^{x}_{n,k,s} & (\ell^{x}_{n,k,s})^2 & \ell^{x}_{n,k,s}\ell^{y}_{n,k,s} \\ \ell^{y}_{n,k,s}\ell^{\alpha}_{n,k,s} & \ell^{y}_{n,k,s} & \ell^{y}_{n,k,s}\ell^{x}_{n,k,s} & (\ell^{y}_{n,k,s})^2 \end{bmatrix}. \tag{B.13}$$

Note that, to calculate (B.12) the following results are useful

$$\mathrm{E}[h_{n,k,s}] = \mu_{n,k,s},\ \mathrm{E}\left[(h_{n,k,s}-\mu_{n,k,s})^2\right] = \sigma_s^2. \tag{B.14}$$

Appendix C

The estimate of the map reconstruction error

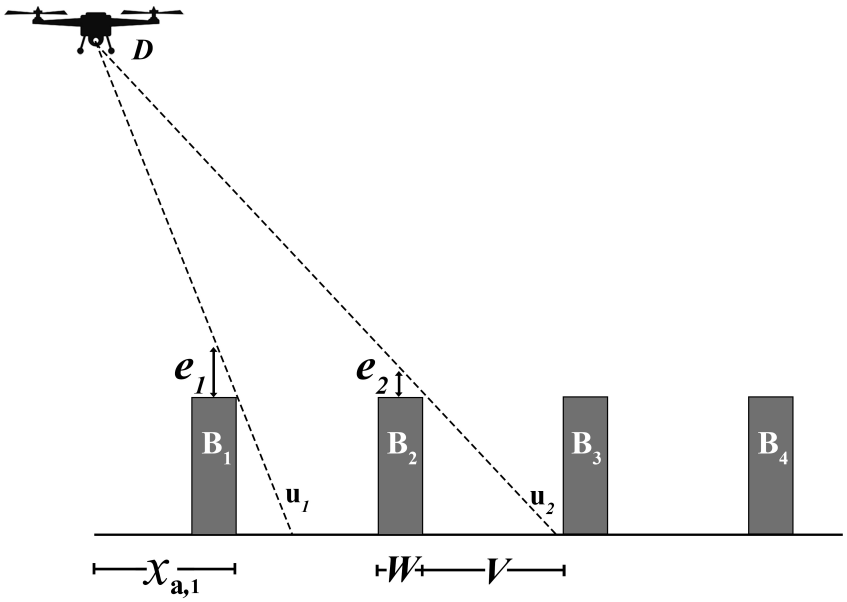


Figure C.1 – One dimensional simplified deterministic city model.

First note that for easy exposition we only exploit LoS users and we find the optimal height based on geometric approach. Referring to the illustration shown in Fig. C.1, we consider the subset of buildings for which there exists a LoS user whose line connecting to the UAV is tangent to the right top corner of the building. We denote by $M = \left\lfloor \frac{V(h_d - H_{\text{av}})}{H_{\text{av}}(W+V)} \right\rfloor$ the number of such buildings which is derived from the position of farthest LoS tangent user from the UAV ($\mathbf{u}_2$ in Fig. C.1). For the i-th such building, a height estimate can be obtained given by

$$\hat{z}_{\mathbf{a},i} = \frac{2\, x_{\mathbf{a},i}\, H_{\mathrm{av}} + W\,(h_d - H_{\mathrm{av}})}{2\, x_{\mathbf{a},i}}, \tag{C.1}$$

where $x_{\mathbf{a},i} = i\,(W+V)$ is the position of the i-th building. We denote the height estimation error of the i-th building by $e_i = |\hat{z}_{\mathbf{a},i} - H_{\mathrm{av}}|$. Then, The summed error over all estimated buildings is given by

$$Er\,(h_d) = \sum_{i=1}^{M} e_i = \sum_{i=1}^{M} |\hat{z}_{\mathbf{a},i} - H_{\mathrm{av}}|\,. \tag{C.2}$$

Note that buildings for which there is no tangent user cannot be directly estimated and are assigned the maximum height. Let us denote Q the number of such buildings, given by

$$Q = \left\lfloor \frac{L}{W+V} \right\rfloor - M. \tag{C.3}$$

The total error over all buildings is now given by

$$\begin{aligned} Er_T\,(h_d) &= Q\,(H_{\max} - H_{\mathrm{av}}) + \sum_{i=1}^{M} |\hat{z}_{\mathbf{a},i} - H_{\mathrm{av}}| \\ &= \left(\left\lfloor \frac{L}{W+V} \right\rfloor - M \right) (H_{\max} - H_{\mathrm{av}}) + \sum_{i=1}^{M} \left| \frac{W\,(h_d - H_{\mathrm{av}})}{2i\,(W+V)} \right|. \end{aligned} \tag{C.4}$$

Appendix D

Chapter 7 Appendices

D.1 Proof of Proposition 7.2.1

Let $\mathbf{v}^*(t), 0 \leq t \leq T$ be the optimal trajectory which traverses the k-th BS's coverage area $\mathcal{A}_k$. Without loss generality, let us assume that within coverage area $\mathcal{A}_k$ the trajectory traverses the n-th sector. We denote the intersections of $\mathbf{v}^*(t)$ with the boarders of sector $a_{k,n}$ as points $\mathbf{v}_{k,n}, \mathbf{v}_{k,n+1}$. For instance in Fig. 7.1, the optimal trajectory intersects the border of the sector $a_{k,1}$ in points $\mathbf{v}_{k,1}, \mathbf{v}_{k,2}$. Since $\mathbf{v}_{k,n}, \mathbf{v}_{k,n+1}$ both are inside $a_{k,n}$ and each sector has a convex shape, then the straight line connecting $\mathbf{v}_{k,n}, \mathbf{v}_{k,n+1}$ also lies inside $a_{k,n}$, mathematically we can write

$$\lambda \mathbf{v}_{k,n} + (1-\lambda)\mathbf{v}_{k,n+1} \in a_{k,n.}, \forall \lambda, 0 \leq \lambda \leq 1. \tag{D.1}$$

This implies that the constraint (7.3) is satisfied for any points on the straight line between $\mathbf{v}_{k,n}, \mathbf{v}_{k,n+1}$. Since, our objective is to minimize the travel time (or equivalently the length of the trajectory), then the optimal trajectory between $\mathbf{v}_{k,n}, \mathbf{v}_{k,n+1}$ is the straight line. Note that (D.1) can equivalently be written as

$$\rho(\mathbf{v}_{k,n}, \mathbf{v}_{k,n+1}) \geq \bar{\rho}. \tag{D.2}$$

Consequently without loss of optimality, the optimal trajectory can be represented as a sequence of the points such that any two consecutive points are connected with a straight line

$$\mathcal{V} = (\mathbf{v}_n)_{n=1}^{N} \,|\, \rho(\mathbf{v}_n, \mathbf{v}_{n+1}) \geq \bar{\rho}, n \in [1, N-1]. \tag{D.3}$$

Hence, problem (7.4) is equivalent to (7.9).

D.2 Proof of Proposition 7.3.1

Without loss of generality consider k-th BS having an coverage area $\mathcal{A}_k$. By definition, we can see that $\hat{\mathbf{u}}_k, \mathbf{x}_{k,j}, k \neq j$ lie inside $\mathcal{A}_k$. Since the coverage area $\mathcal{A}_k$ can be represented by a union convex non-overlapping sectors as defined in (7.7), by construction, there always exits a straight line path connecting $\hat{\mathbf{u}}_k$ and $\mathbf{x}_{k,j}$ which always lies inside the

coverage region $\mathcal{A}_k$. Therefore all edges $(\hat{\mathbf{u}}_k, \mathbf{x}_{k,j}), k \neq j$ satisfy the coverage constraint. Since, we assume that initial and terminal points of the UAV are always in the coverage area of at least one BS, it can be easily see that edges of the form $(\mathbf{v}_\mathrm{I}, \hat{\mathbf{u}}_k)$ and $(\hat{\mathbf{u}}_k, \mathbf{v}_\mathrm{F})$ also satisfy the constraint in (7.9b).

D.3 Proof of Lemma 7.4.1

Let's assume that the line $L(\mathbf{x}, \mathbf{y})$ sequentially traverses some sectors in $\mathcal{A}_k$, denoted by $(a_{k,1}, \ldots, a_{k,N'})$ with starting location $\mathbf{x} \in a_{k,1}$ and ending location $\mathbf{y} \in a_{k,N'}$. The set of intersections of the line with the boundaries of the sectors is denoted by a sequence of the points $(\mathbf{x}_j)_{j=1}^{J}$.

Since all the sectors are convex, it can be shown that if $\{\mathbf{x}_j, \mathbf{x}_{j+1}\}, j \in [1, J]$ belong to a same sector then the line $L(\mathbf{x}_j, \mathbf{x}_{j+1})$ lies inside $\mathcal{A}_k$. Therefore, to check if the line $L(\mathbf{x}, \mathbf{y})$ is inside the coverage area, it is enough to evaluate a limited number of points.

Résumé [Français]

L'utilisation de drones et de véhicules aériens sans pilote (UAV) en tant que nuds de réseau d'accès radio (RAN) volants devient rapidement un outil puissant pour compléter les déploiements terrestres fixes traditionnels. L'avantage d'utiliser des UAVs se fera particulièrement sentir dans les cas d'utilisation où il est essentiel de pouvoir déployer rapidement un réseau où et quand cela est important. Cependant, le succès des RAN dits volants dépend de la capacité des drones à se positionner spatialement de manière efficace et autonome. Dans cette optique, cette thèse vise à étudier les travaux et technologies actuels des communications sans fil assistées par UAVs et développe de nouvelles méthodes pour le placement et la conception de trajectoire d'un UAV en tant que RAN volant dans le réseau sans fil.

l'avantage du RAN volant par rapport à sa contrepartie fixe conventionnelle réside essentiellement dans la capacité à rapprocher le RAN de l'utilisateur ou généralement à augmenter la qualité de la liaison. Par conséquent, le rôle joué par les modèles de canaux statistiques dans les solutions de placement et de conception de trajectoire est essentiel.

L'hypothèse de canaux en visibilité directe (LoS) ou l'utilisation de modèles de blocage statistique simples (c'est-à-dire la modélisation de la probabilité LoS) s'est avérée un excellent moyen de tirer un aperçu précoce du problème et de permettre une analyse des performances moyennes sous forme fermée. Malheureusement, la nature simpliste ou probabiliste de ces approches limite notre capacité à garantir les performances dans un déploiement réel d'UAV sur le terrain. Par exemple, un algorithme de placement statistiquement optimisé pourrait suggérer un emplacement d'UAV que l'on découvre finalement être gravement affecté par le blocage local dans la pratique (par exemple, la présence imprévue d'un immeuble de grande hauteur) forçant le robot à un recalcul de chemin sous-optimal.

Cette étude est consacrée au développement de méthodes permettant de contourner ces inconvénients pour les déploiements d'UAV sur le terrain. Nous attirons l'attention du lecteur sur le rôle joué par l'exploitation de cartes adaptées. Par cartes, nous nous référons ici à un ensemble de données indexées géographiquement qui peuvent être utilisées pour mieux prédire les conditions réelles des canaux pour toute paire spécifique d'UAV et d'emplacements de nuds au sol. Plusieurs types de cartes interdépendantes peuvent être envisagées, y compris les cartes de débit, les cartes radio (force de liaison) et les cartes 3D physiques. Dans cette thèse, nous considérons l'exploitation de la carte 3D. Nous donnons un aperçu des approches de placement d'UAV pour les paramètres Drone-as-a-Relay (DaaR) et Drone-as-a-Base station (DaaB), puis nous introduisons des méthodes de

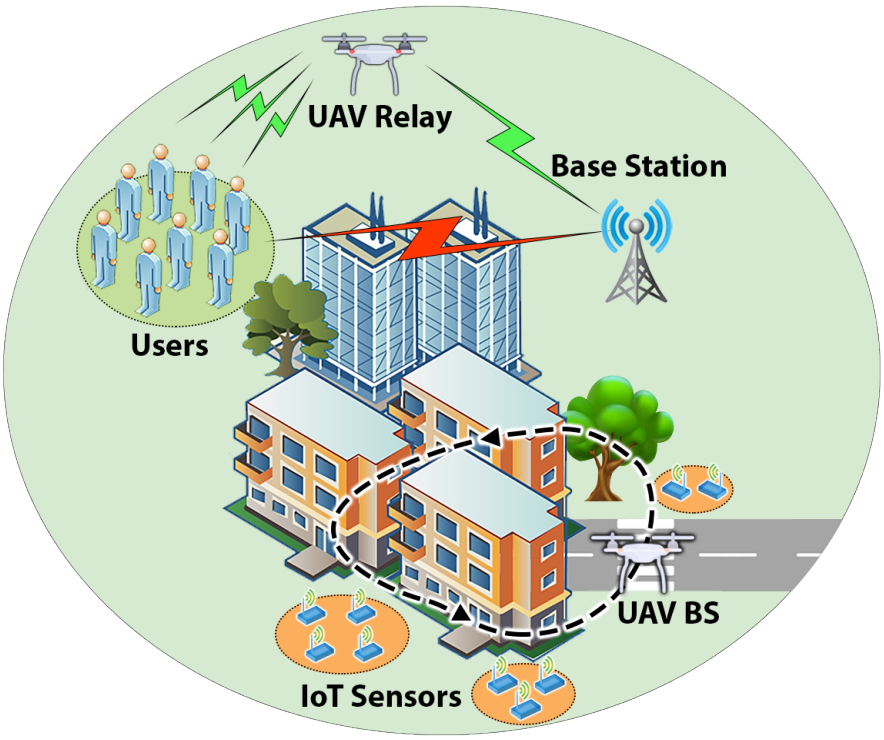


Figure 9.1 – Illustration de quelques cas d'utilisation de RAN volants prometteurs: Drone-en-tant-que-relais pour la connectivité des utilisateurs au sol (DaaR), Drone-en-tant-que-station-de-base pour la collecte de données IoT (DaaB).

placement et de conception de trajectoire d'UAV basées sur une carte. Les modèles de canaux, éventuellement dérivés de cartes 3D, sont d'abord décrits. Tandis que les cartes 3D sont riches en informations, elles sont également difficiles à exploiter directement dans un problème de placement et d'optimisation de trajectoire en raison du comportement très irrégulier du blocage radio, notamment en environnements urbain. Nous dérivons des méthodes de placement (à la fois statiques et basé sur la trajectoire) qui s'appuient sur des données cartographiques en faisant le tour de la nature apparemment quasi non différenciable du problème.

Les composants essentiels des scénarios et applications susmentionnés sont la connaissance des emplacements des utilisateurs et des caractéristiques des canaux radio, qui ne sont généralement pas disponibles. Pour cela, nous enquêtons également le problème de l'apprentissage des canaux radio et localisation des nuds au sol à partir des mesures radio. En général, les performances de tout processus d'apprentissage dépendent fortement de l'ensemble de données d'apprentissage, qui dans ce cas est des mesures radio collecté des nuds au sol. À cette fin, nous avons pour devise une trajectoire optimisée pour l'UAV en capitalisant sur la carte 3D pour collecter intelligemment les mesures depuis nuds au sol afin d'augmenter les performances d'apprentissage et de localisation.

Les algorithmes proposés fonctionnent sur la base de la ville 3D où l'UAV est déployé.

Cependant, ces informations peuvent ne pas être disponibles et doivent souvent être estimées à partir de mesures transmises par l'UAV drone. Étant donné que piloter un drone simplement pour construire une carte est souvent coûteux, nous proposons un algorithme pour construire la carte 3D de la ville comme sous-produit uniquement à partir de la mesure radio pendant que le drone accomplit sa mission et survole le réseau.

D'autre part, l'énorme augmentation de l'utilisation des UAVs pour les différentes applications telles que le transport, la livraison de marchandises, surveillance du système, etc. a été observée au cours des dernières années. L'habilitation de ces applications nécessite une connectivité sûre et transparente entre les infrastructures fixes au sol et UAVs. Un défi majeur dans ce domaine réside dans la conception de trajectoires qui, tout en permettant l'achèvement de la mission UAV, peut garantir une connectivité cellulaire fiable tout au long du chemin. Les approches précédentes dans ce domaine ont considéré la propagation simpliste des hypothèses de modèle (par exemple, basées sur la ligne de visée) ou des modèles plus avancés, mais avec des solutions optimisées exigeantes en calcul. Dans cette étude, nous proposons également une nouvelle approche pour la conception de trajectoires entre un emplacement de départ et une destination donnée de telle sorte que l'obtention d'une qualité fiable de service (QoS) du réseau cellulaire tout au long du trajet.

Enfin, nous discutons des réalisations de prototypes pratiques. Nous concevons un prototype radio volant, appelé Rebot (Relay Robot), qui comprend des UAVs équipé d'un relais LTE qui peut fournir une connectivité LTE de bout en bout entre les utilisateurs au sol et le réseau central. Il fonctionne également comme un robot autonome capable de se positionner à un emplacement maximisant le débit.

Les paragraphes suivants donnent un apercu des contributions principales des différents chapitre de la thèse.

Contribution 1 (Chapitre 3)- Emplacement basé sur la carte et conception de trajectoire dans les réseaux sans fil assistés par UAV

Plusieurs questions nouvelles et fascinantes découlent de l'étude des BS volants dans un réseau sans fil. Ceux-ci peuvent être largement classés en problèmes de placement et de conception de la trajectoire. Alors que le problème de placement concerne la recherche d'emplacements flexibles mais statiques des BS d'UAV, la conception de trajectoire implique la recherche de trajectoires d'UAV. En ce qui concerne les problèmes de placement ou de conception de trajectoire, la plupart des solutions existantes reposent sur des modèles d'atténuation de canal simplifiés qui sont basés sur des liaisons LoS (garanties de manière déterministe) ou sur des modèles prédictifs pour la probabilité d'occurrence d'une liaison LoS. La plupart des approches précédentes manquent de garanties de performances réelles pour les algorithmes de placement ou de conception de trajectoire lorsqu'ils sont utilisés dans un scénario de navigation réel. Afin de contourner ce problème, dans ce chapitre, nous proposons une solution en incorporant des données réelles de la carte de la ville 3D dans le placement de l'UAV et la conception de la trajectoire via une méthode de compression de carte qui nous permet de recourir au problème pour des arguments d'optimisation convexes. Un exemple de la trajectoire obtenue sous l'approche de compression de carte pour un paramètre IoT avec 3 nuds au sol est illustré sur la figure 9.2. Sur la figure 9.3, les performances de l'algorithme basé sur la carte proposé en comparaison avec deux autres approches par rapport au temps de vol en considérant 6 nuds au sol est montré.

Une partie des travaux de ce chapitre a donné lieu aux publications suivantes:

- *Esrafilian, Omid, Rajeev Gangula, and David Gesbert. "UAV-relay Placement with Unknown User Locations and Channel Parameters." In 2018 52nd Asilomar Conference on Signals, Systems, and Computers.*
- *Esrafilian, Omid, Rajeev Gangula, and David Gesbert. "Learning to communicate in UAV-aided wireless networks: Map-based approaches." IEEE Internet of Things Journal 6, no. 2 (2018): 1791-1802.*

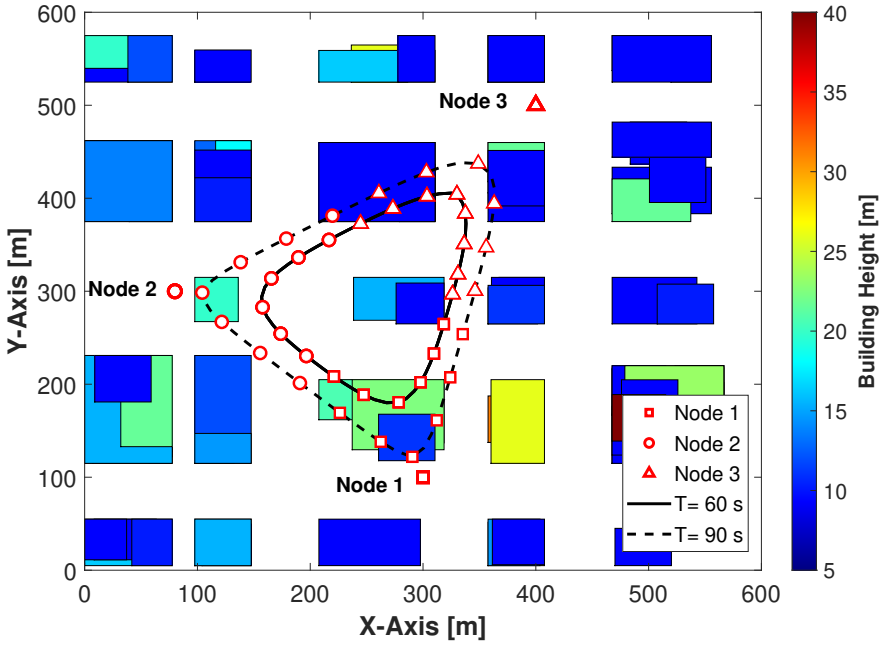


Figure 9.2 – Exemple de conception de trajectoire pour la collecte de données IoT (3 capteurs Cas).

Contribution 2 (Chapitre 4)- Apprentissage actif pour l'estimation des canaux : approche basée sur des cartes

L'une des hypothèses courantes dans les travaux précédents concernant les communications UAV, est que les paramètres du modèle de canal radio sont supposés être connus pour le placement optimal des UAVs ou la conception de la trajectoire des UAV. Cependant, en réalité, ces paramètres doivent être appris sur la base des mesures recueillies auprès des utilisateurs du sol. Dans ce chapitre, nous nous concentrons uniquement sur l'apprentissage des paramètres des canaux radio. Nous formulons et résolvons un problème d'optimisation de trajectoire d'apprentissage afin de minimiser l'erreur d'estimation des paramètres du modèle de canal. La trajectoire imaginée permet à l'UAV d'exploiter la carte et d'apprendre rapidement les paramètres de propagation dans un temps de vol limité. Sur la figure 9.4, un exemple de la trajectoire d'UAV conçue en utilisant notre algorithme proposé pour apprendre de manière optimale les canaux radio est montré. Sur la figure 9.5, les performances de la trajectoire optimale en termes de de l'erreur quadratique moyenne (MSE) des paramètres de canal appris sont représentées en fonction du nombre de nuds au sol.

Les résultats présentés dans cette section ont été publiés dans la publication suivante

- *Esrafilian, Omid, Rajeev Gangula, and David Gesbert. "Learning to communicate*

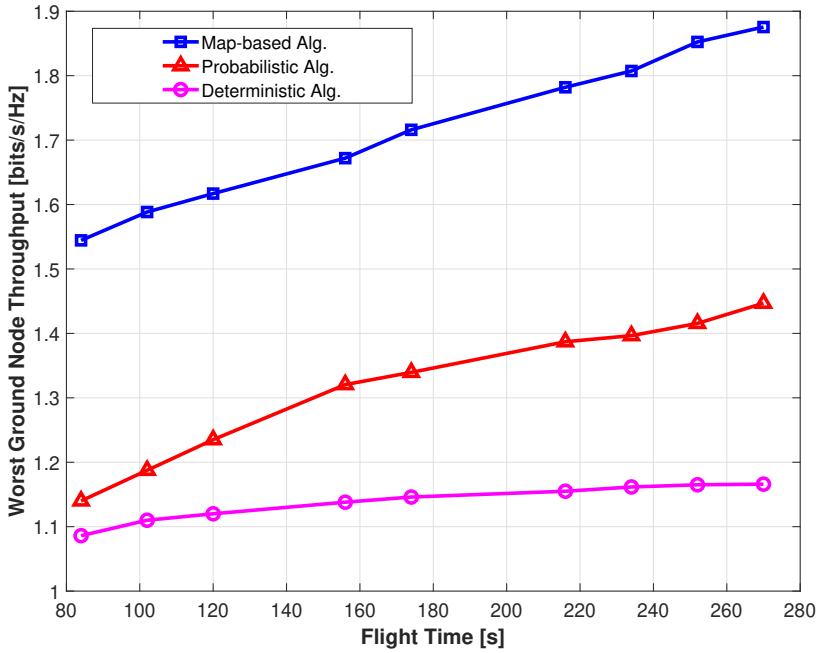
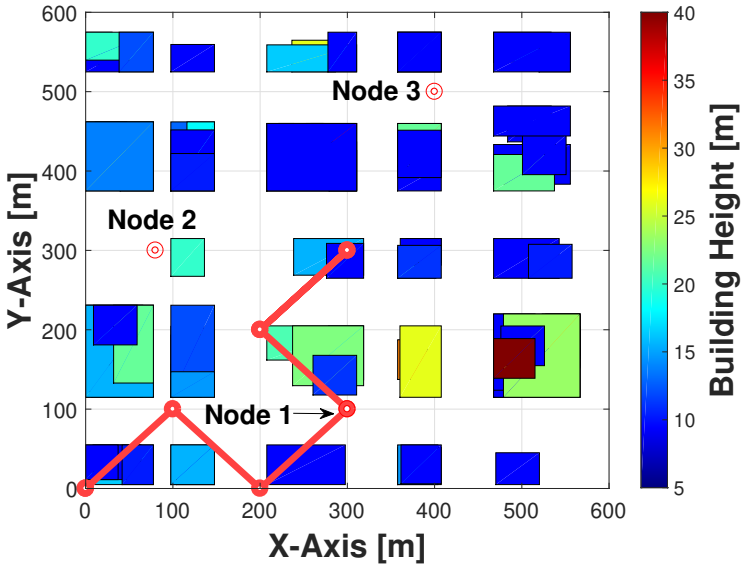


Figure 9.3 – Performances de l'algorithme basé sur la carte pour les appris et les vrais paramètres de canal en comparaison avec d'autres algorithmes pour 6 nuds au sol par rapport à l'augmentation du temps de vol.

(a)



(b)

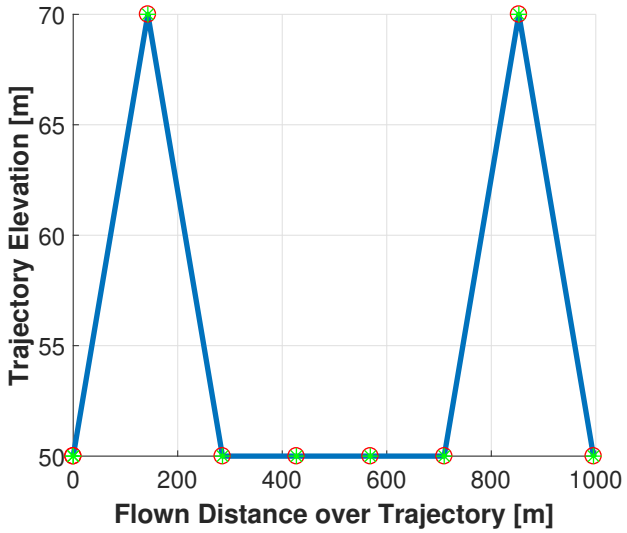


Figure 9.4 – (a) Vue de dessus de la trajectoire d'apprentissage optimale en utilisant l'algorithme proposé. (b) L'élévation de l'UAV le long de la trajectoire.

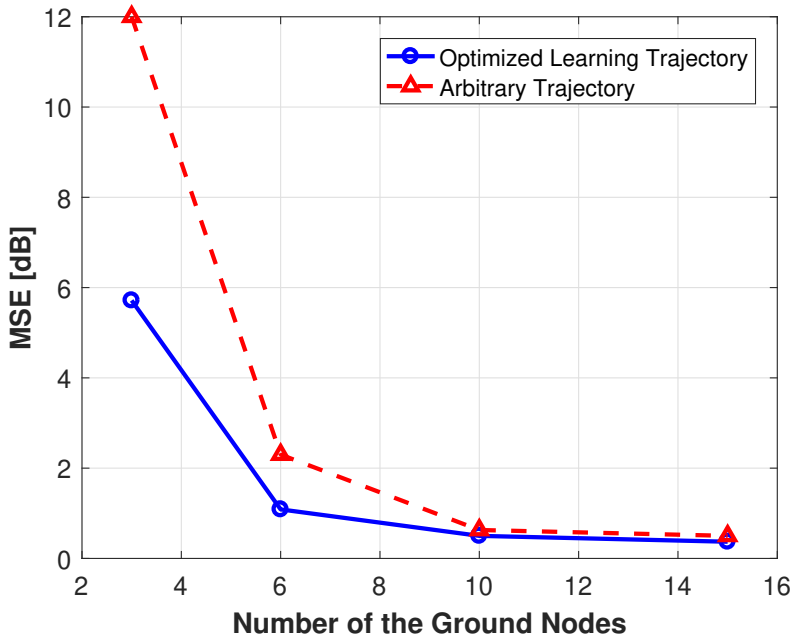


Figure 9.5 – Comparaison de l'erreur quadratique moyenne (MSE) pour différentes trajectoires d'apprentissage.

in UAV-aided wireless networks: Map-based approaches." IEEE Internet of Things Journal 6, no. 2 (2018): 1791-1802.

Contribution 3 (Chapitre 5) - Localisation des nuds dans les réseaux sans fil

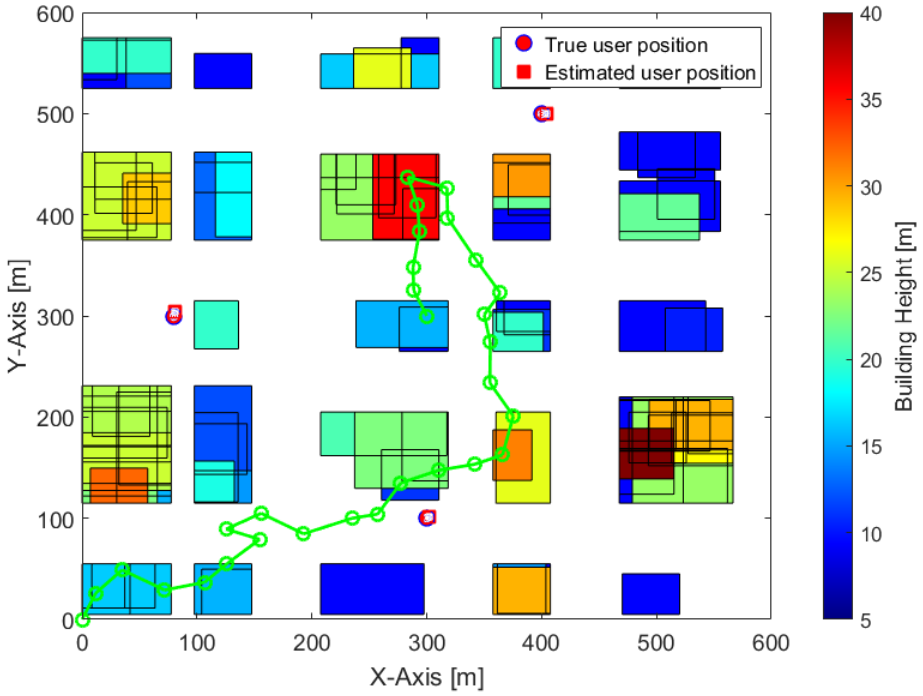
Indépendamment du placement ou de la conception de la trajectoire pour l'UAV, les algorithmes fonctionnent généralement sur la base d'un tableau d'informations comprenant les informations de localisation des nuds au sol. L'utilisation du GPS pour obtenir l'emplacement des nuds n'est pas une solution viable à tout moment en raison de la dégradation des signaux GPS ou de l'effet de blocage des obstacles dans l'environnement. Pour résoudre ce problème dans ce chapitre, nous formulons le problème de la localisation des utilisateurs en traitant les mesures de l'indicateur de force du signal reçu (RSSI) qui sont collectées par un UAV tout en capitalisant sur les données cartographiques 3D de la ville. Étant donné que la localisation dépend fortement de la connaissance des canaux radio entre les utilisateurs et le UAV, le UAV doit non seulement localiser les utilisateurs mais également apprendre les canaux en même temps. De plus, nous formulons un nouveau problème d'optimisation de trajectoire d'UAV contraint par les ressources pour améliorer encore la précision de localisation. Sur la figure 9.6, la sortie de l'algorithme de localisation de base de carte proposé pour un scénario comprenant 3 nuds au sol tandis que l'UAV suit la trajectoire optimale est représentée. Dans la figure 9.7, nous comparons les performances de l'algorithme de localisation pour le cas d'utilisateur unique proposé dans ce chapitre avec d'autres approches introduites dans [65; 83].

Certaines parties des résultats présentés dans cette section ont été publiées dans les publications suivantes

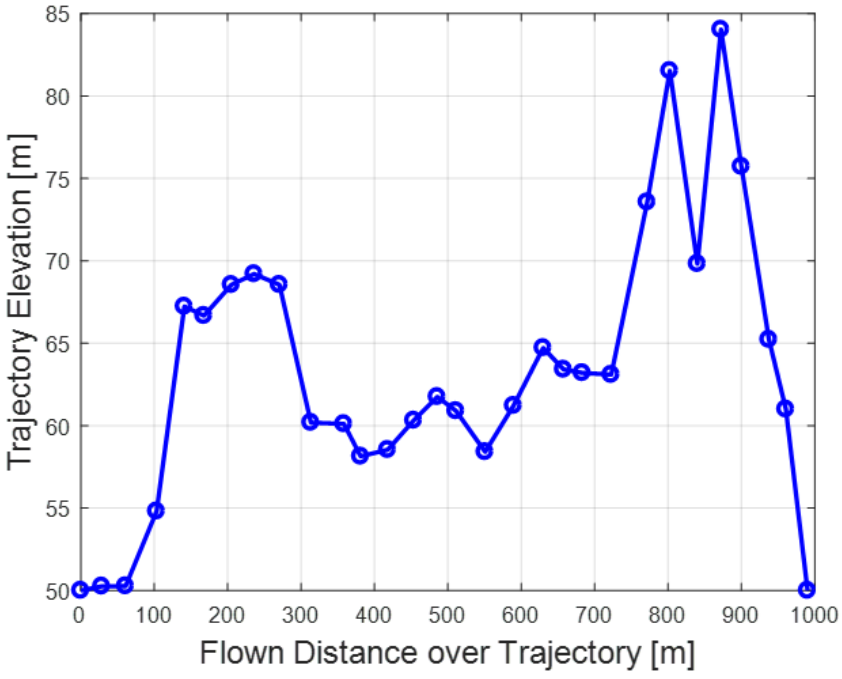
- *Esrafilian, Omid, Rajeev Gangula, and David Gesbert. "UAV-relay Placement with Unknown User Locations and Channel Parameters." In 2018 52nd Asilomar Conference on Signals, Systems, and Computers.*

Et la publication suivante sera soumise prochainement:

- *Esrafilian, Omid, Rajeev Gangula, and David Gesbert. "Simultaneous Learning and Path Planning (SLAP) in UAV-Aided Wireless Communications", To be submitted.*



(a)



(b)

Figure 9.6 – (a) Vue de dessus de la trajectoire générée en utilisant l'approche sous-optimale. (b) Altitude du drone le long de la trajectoire.

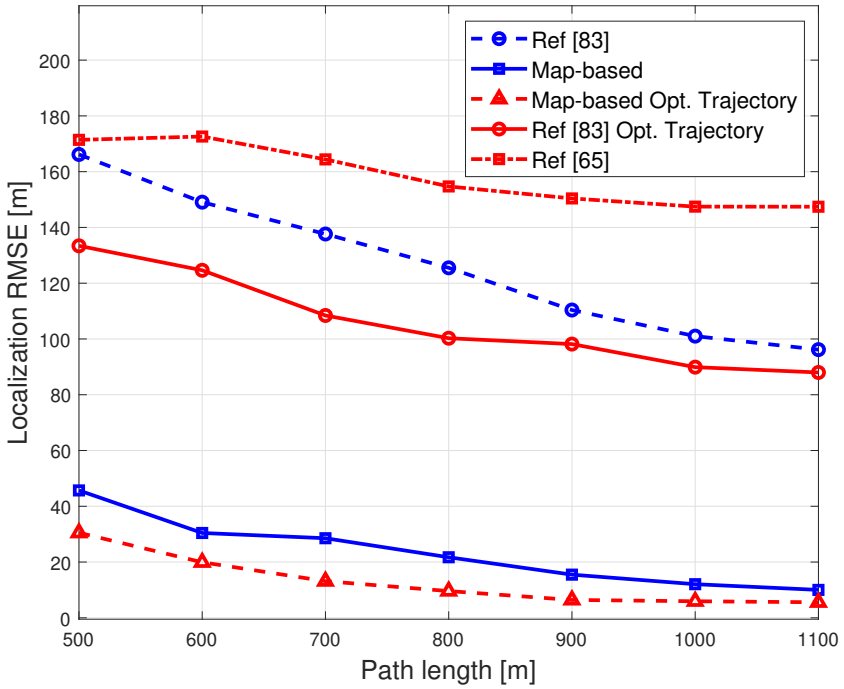


Figure 9.7 – Comparaison du MSE pour différentes trajectoires d'apprentissage pour le cas de l'utilisateur unique.

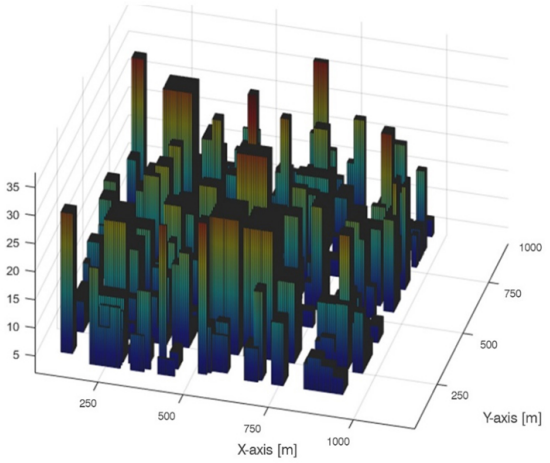
Contribution 4 (Chapter 6) - Reconstruction of the 3D city map from radio measurements based on UAVs

L'hypothèse essentielle dans les chapitres précédents est la connaissance de la carte 3D des zones où l'UAV est déployé. Traditionnellement, la reconstruction de plans de ville en 3D utilise des techniques de photogrammétrie qui nécessitent des images stéréo haute résolution et une capacité de traitement étendue. Étant donné que l'UAV peut survoler la ville et effectuer des mesures à partir des utilisateurs au sol, dans ce chapitre, nous visons à proposer un nouvel algorithme pour reconstruire des plans de ville 3D en articulant les mesures radio. L'approche proposée repose sur la capacité unique d'un système de communication UAV-sol à détecter et à classer les canaux LoS par rapport aux canaux sans visibilité directe (NLoS) vers les utilisateurs au sol à l'aide d'outils d'apprentissage de la machine. figure 9.8 montre un exemple de carte reconstruite tandis que l'UAV suit une trajectoire carrée centrée au centre de la ville avec une longueur de 800 mètres et une altitude fixe de 70 mètres. Pour reconstruire la carte 3D de la ville, l'UAV prend des mesures auprès de 1584 utilisateurs au sol extérieurs qui sont dispersés au hasard dans la ville.

Les travaux présentés dans ce chapitre ont donné lieu à la publication suivante :

- *Esrafilian, Omid, and David Gesbert. "3D city map reconstruction from UAV-based radio measurements." In 2017 IEEE Global Communications Conference (GLOBECOM).*

(a)



(b)

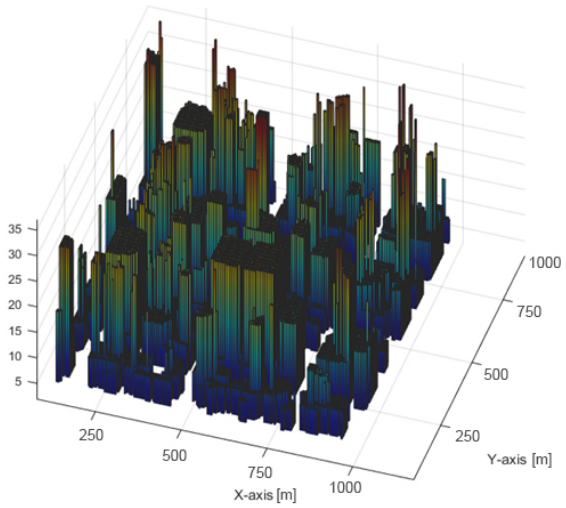


Figure 9.8 – (a) Vue latérale du plan de ville 3D. (b) Carte reconstruite avec NMSE égale à 0,35.

Contribution 5 (Chapitre 7) - Conception de trajectoire de drone assistée par carte 3D sous contraintes de connectivité cellulaire

Dans ce chapitre, nous avons étudié le problème de la conception de trajectoires d'UAV sous contrainte de connectivité cellulaire. Un défi clé de ce problème réside dans la conception de trajectoires entre un départ prédéterminé et une destination qui, tout en permettant l'achèvement de la mission d'UAV, peuvent garantir une connectivité cellulaire fiable tout au long du trajet. La mission UAV, peut garantir une connectivité cellulaire fiable tout au long du chemin. Nous avons proposé une nouvelle approche qui établit un compromis entre les performances (c'est-à-dire la réduction de la longueur du trajet) et la complexité en exploitant la carte 3D de l'environnement et en utilisant la théorie des graphes. Nous avons établi un cadre basé sur la théorie des graphes pour évaluer d'abord la faisabilité du problème, puis pour obtenir une solution approximative de haute qualité au problème de conception de trajectoire d'UAV. La figure 9.9 illustre 25 BS et la carte de couverture où les régions en surbrillance représentent les zones où la contrainte SNR minimale est satisfaite. Nous vérifions d'abord la faisabilité du problème en trouvant la trajectoire de base entre les emplacements de départ et de destination. Nous optimisons ensuite la trajectoire de base pour trouver une meilleure solution. Nous avons également comparé notre méthode aux autres approches basées sur des graphiques. Sur la figure 9.10, nous évaluons les performances des différentes approches en termes de panne sur 1000 simulations sur Monte-Carlo avec différents emplacements BS.

Les résultats présentés dans ce chapitre ont été soumis pour publication :

- *Esrafilian, Omid, Rajeev Gangula, and David Gesbert. "3D-Map Assisted UAV Trajectory Design Under Cellular Connectivity Constraints." Submitted In 2020 IEEE international conference on communications (ICC).*

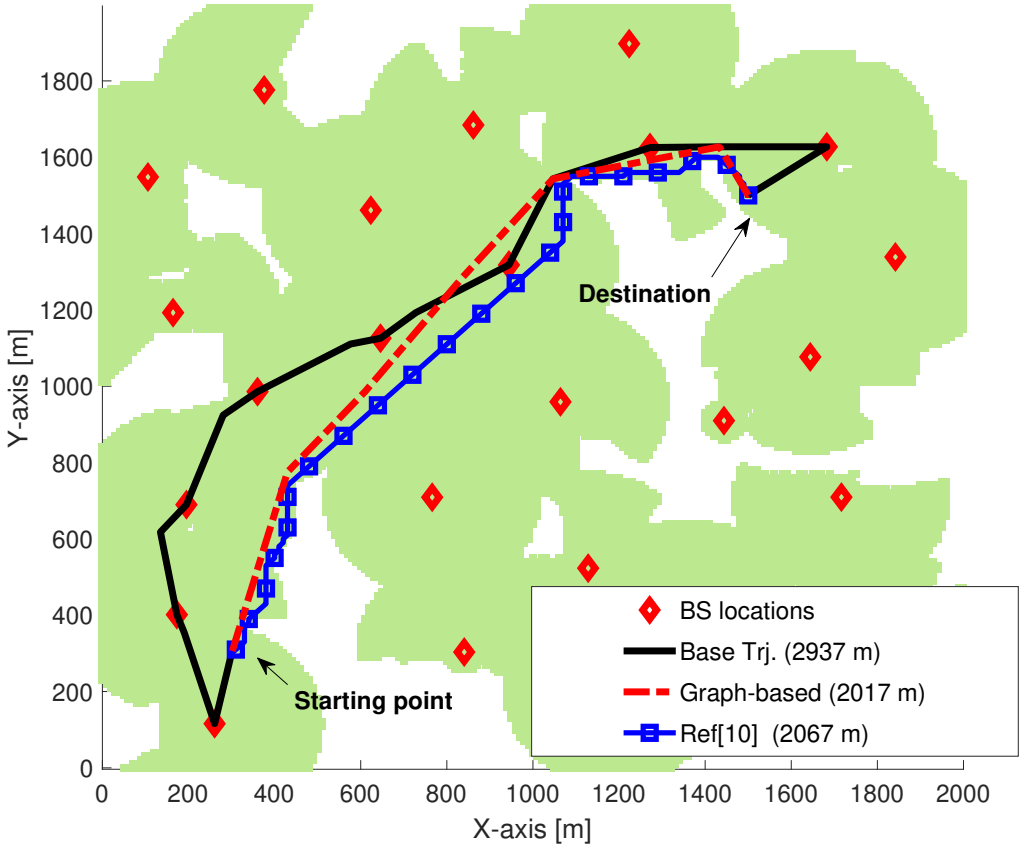


Figure 9.9 – Vue de dessus de la ville, des emplacements BS, des trajectoires générées et de ses longueurs pour différents algorithmes. La zone de couverture de chaque BS est surlignée en vert.

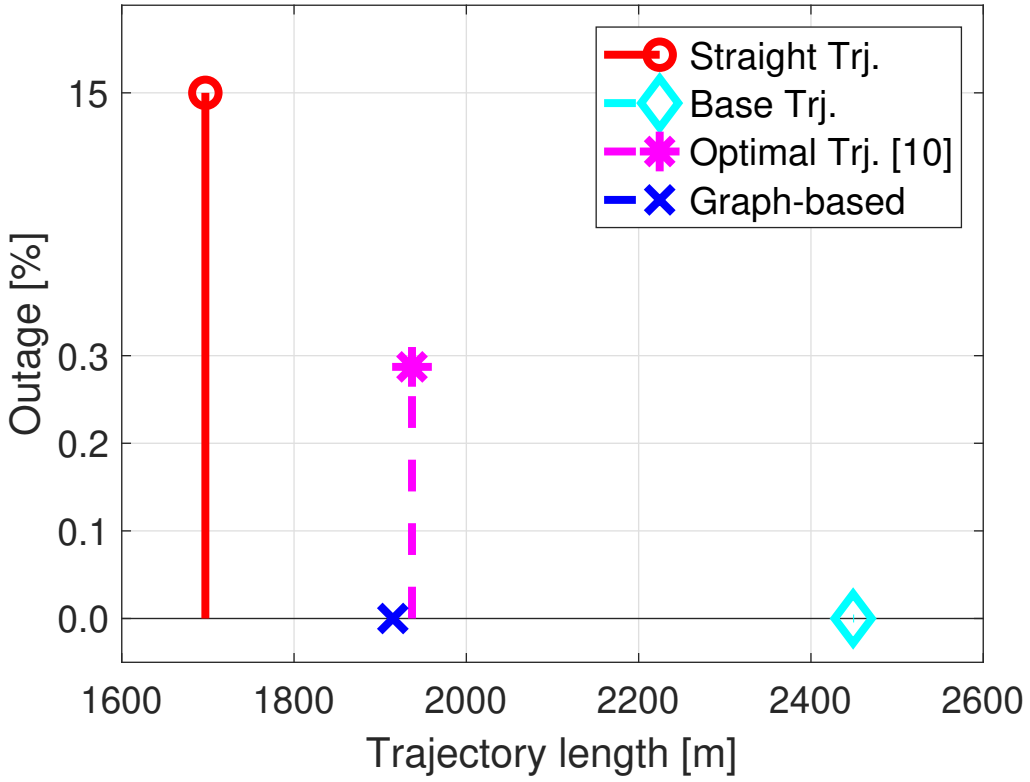


Figure 9.10 – coupure en fonction de la longueur de la trajectoire pour différents algorithmes.

Contribution 6 (Chapitre 8) - Vérification expérimentale

Dans ce chapitre, nous évaluons les performances des algorithmes proposés à travers la vérification expérimentale, et nous discuterons également des réalisations de prototypes pratiques. Ce chapitre présente le prototype du concept dit Rebot (Relay Robot). Le Rebot fonctionne à la fois comme un relais LTE extérieur entre les utilisateurs au sol et un BS fixe, ainsi que comme un robot autonome capable de se positionner à un emplacement maximisant le débit. Dans [4], un enregistrement vidéo de l'expérience sur le campus d'EURECOM est également capturé, illustrant à la fois l'avantage de débit, l'auto-placement axé sur l'apprentissage automatique et les capacités de suivi du Rebot. Différentes parties du prototype Rebot sont illustrées à la figure 9.11.

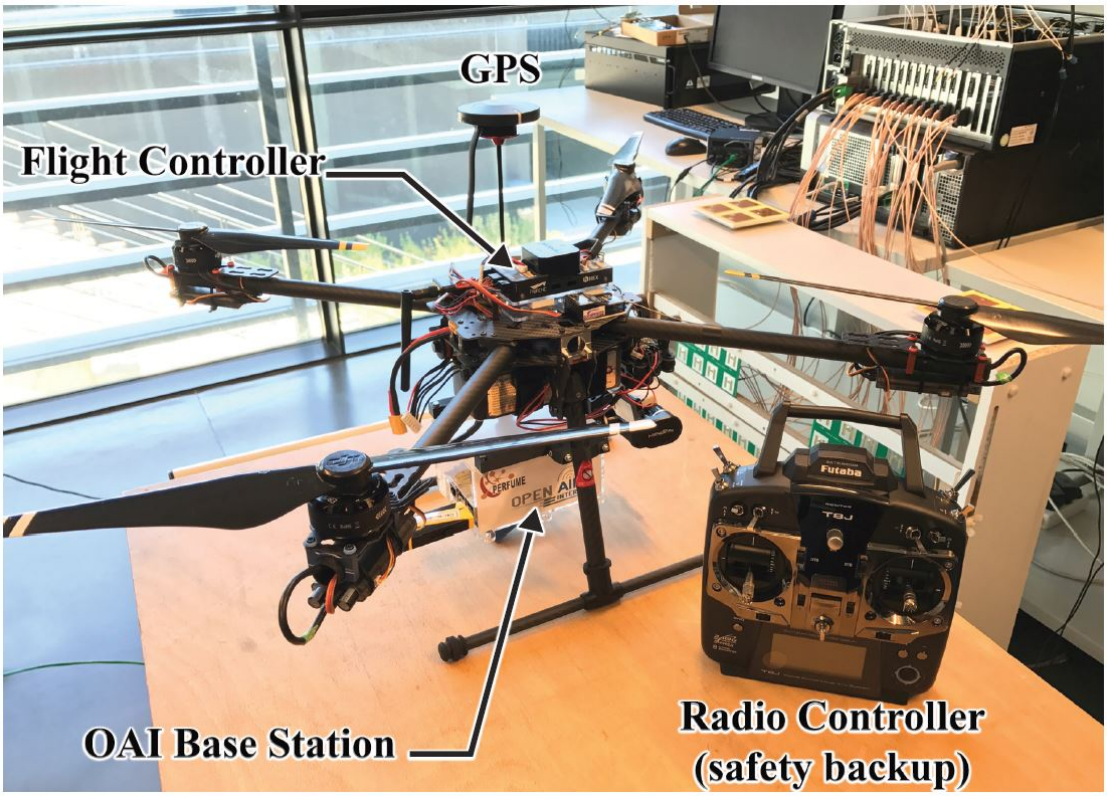


Figure 9.11 – coupure en fonction de la longueur de la trajectoire pour différents algorithmes.

Les résultats présentés dans ce chapitre ont été soumis pour publication:

- *Gangula, Rajeev, Esrafilian, Omid, et al. "Flying rebots: First results on an autonomous UAV-based LTE relay using open airinterface." 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC).*

Bibliography

[1] FAA, "UAS Integration Pilot Program," https://www.faa.gov/uas/programs_partnerships/integration_pilot_program/.

[2] F. News, "SUMMARY OF SMALL UNMANNED AIRCRAFT RULE," https://www.faa.gov/uas/media/part_107_summary.pdf.

[3] "U-space." [Online]. Available: https://www.sesarju.eu/U-space

[4] "Faa." [Online]. Available: https://www.faa.gov/uas/

[5] MarketsandMarkets, "Drone service market application forecast," 2019. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/drone-services-market-80726041.html

[6] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[7] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *IEEE Global Communications Conference (GLOBECOM)*, 2014.

[8] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of unmanned aerial vehicles*. Springer, 2015.

[9] S. Zhang, Y. Zeng, and R. Zhang, "Cellular-enabled UAV communication: A connectivity-constrained trajectory optimization perspective," *IEEE Trans. on Comm*, vol. 67, no. 3, pp. 2580–2604, 2018.

[10] S. Zhang and R. Zhang, "Radio map based path planning for cellular-connected UAV," *arXiv preprint arXiv:1905.05046*, 2019.

[11] Y. Zeng, J. Lyu, and R. Zhang, "Cellular-connected UAV: Potential, challenges, and promising technologies," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 120–127, Feb 2019.

[12] U. Challita, W. Saad, and C. Bettstetter, "Cellular-connected UAVs over 5G: Deep reinforcement learning for interference management," *arXiv preprint arXiv:1801.05500*, 2018.

[13] Y. Zeng, Q. Wu, and R. Zhang, "Accessing from the sky: A tutorial on UAV communications for 5G and beyond," *arXiv preprint arXiv:1903.05289*, 2019.

[14] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs." *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, 2016.

[15] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[16] R. Gangula, D. Gesbert, D. F. Külzer, and J. M. Franceschi, "A landing spot approach for enhancing the performance of UAV-aided wireless networks," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2018, pp. 1–6.

[17] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun*, vol. 16, no. 6, pp. 3747–3760, 2017.

[18] D. Yang, Q. Wu, Y. Zeng, and R. Zhang, "Energy tradeoff in ground-to-UAV communication via trajectory design," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6721–6726, 2018.

[19] S. De Bast, E. Vinogradov, and S. Pollin, "Cellular coverage-aware path planning for UAVs," in *Proc SPAWC*, July 2019.

[20] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2125–2140, April 2019.

[21] N. Goddemeier, K. Daniel, and C. Wietfeld, "Role-based connectivity management with realistic air-to-ground channels for cooperative uavs," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 5, pp. 951–963, 2012.

[22] N. Ahmed, S. S. Kanhere, and S. Jha, "On the importance of link characterization for aerial wireless sensor networks," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 52–57, 2016.

[23] W. Khawaja, I. Guvenc, D. W. Matolak, U.-C. Fiebig, and N. Schneckenberger, "A survey of air-to-ground propagation channel modeling for unmanned aerial vehicles," *IEEE Communications Surveys & Tutorials*, 2019.

[24] D. W. Matolak and R. Sun, "Unmanned aircraft systems: Air-ground channel characterization for future applications," *IEEE Vehicular Technology Magazine*, vol. 10, no. 2, pp. 79–85, 2015.

[25] A. A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, "A survey of channel modeling for UAV communications," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2804–2821, 2018.

[26] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[27] J. Chen, U. Yatnalli, and D. Gesbert, "Learning radio maps for UAV-aided wireless networks: A segmented regression approach," in *IEEE International Conference on Communications (ICC)*, 2017.

[28] O. Esrafilian and D. Gesbert, "3D city map reconstruction from UAV-based radio measurements," in *IEEE Global Communication Conference (GLOBECOM)*, 2017.

[29] K. T. Herring, J. W. Holloway, D. H. Staelin, and D. W. Bliss, "Path-loss characteristics of urban wireless channels," *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 1, pp. 171–177, 2009.

[30] H. Jiang, Z. Zhang, L. Wu, and J. Dang, "Three-dimensional geometry-based UAV-MIMO channel modeling for A2G communication environments," *IEEE Communications Letters*, vol. 22, no. 7, pp. 1438–1441, 2018.

[31] P. Chandhar and E. G. Larsson, "Massive MIMO for drone communications: Applications, case studies and future directions," *arXiv preprint arXiv:1711.07668*, 2017.

[32] W. Khawaja, O. Ozdemir, and I. Guvenc, "UAV air-to-ground channel characterization for mmWave systems," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2017, pp. 1–5.

[33] Z. Meng, Y. Chen, M. Ding, and D. López-Pérez, "A new look at UAV channel modeling: A long tail of LoS probability," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2019, pp. 1–6.

[34] Y. Zeng, X. Xu, and R. Zhang, "Trajectory design for completion time minimization in uav-enabled multicasting," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2233–2246, 2018.

[35] J. Gong, T.-H. Chang, C. Shen, and X. Chen, "Flight time minimization of uav for data collection over wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1942–1954, 2018.

[36] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.

[37] B. Galkin, J. Kibilda, and L. A. Da Silva, "UAVs as mobile infrastructure: addressing battery lifetime," *IEEE Communications Magazine*, 2019.

[38] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.

[39] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys & Tutorials*, 2019.

[40] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Optimal transport theory for power-efficient deployment of unmanned aerial vehicles," in *IEEE International Conference on Communications (ICC)*, 2016.

[41] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 434–437, 2017.

[42] O. Esrafilian and D. Gesbert, "Simultaneous User Association and Placement in Multi-UAV Enabled Wireless Networks," in *22nd International ITG Workshop on Smart Antennas (WSA)*. VDE, 2018.

[43] Ladosz Pawel and Hyondong Oh and Wen-Hua Chen, "Optimal positioning of communication relay unmanned aerial vehicles in urban environments." in *IEEE International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016.

[44] J. Chen and D. Gesbert, "Optimal positioning of flying relays for wireless networks: A LOS map approach," in *IEEE International Conference on Communications (ICC)*, 2017.

[45] R. Gangula, P. de Kerret, O. Esrafilian, and D. Gesbert, "Trajectory optimization for mobile access point," in *IEEE 51st Asilomar Conference on Signals, Systems, and Computers*, 2017.

[46] Wu, Qingqing, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks." *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, 2018.

[47] A. T. Klesh, P. T. Kabamba, and A. R. Girard, "Path planning for cooperative time-optimal information collection," in *IEEE American Control Conference*, 2008.

[48] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Transactions on Communications*, vol. 64, no. 12, pp. 4983–4996, 2016.

[49] B. Galkin, J. Kibilda, and L. Da Silva, "A stochastic model for UAV networks positioned above demand hotspots in urban environments," *IEEE Transactions on Vehicular Technology*, 2019.

[50] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion." *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[51] Q. T. Dinh and M. Diehl, "Local convergence of sequential convex programming for nonconvex optimization." *Recent Advances in Optimization and its Applications in Engineering*, pp. 93–102, 2010.

[52] S. Rogers and M. Girolami, *A first course in machine learning.* CRC Press, 2016.

[53] Cover, T. M., and J. A. Thomas, *Elements of information theory.* John Wiley & Sons, 2012.

[54] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming, version 2.0 beta." http://cvxr.com/cvx, 2013.

[55] 3GPP TR 38.901 version 14.0.0 Release 14, "Study on channel model for frequencies from 0.5 to 100 GHz," ETSI, Tech. Rep., 2017.

[56] J. Chen, O. Esrafilian, D. Gesbert, and U. Mitra, "Efficient algorithms for air-to-ground channel reconstruction in UAV-aided communications," in *IEEE Globecom Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles*, 2017.

[57] D. E. Kirk, *Optimal control theory: an introduction.* New York, USA: Dover, 2012.

[58] R. Zekavat and R. M. Buehrer, *Handbook of position location: Theory, practice and advances.* John Wiley & Sons, 2011, vol. 27.

[59] A. Boukerche, H. A. Oliveira, E. F. Nakamura, and A. A. Loureiro, "Localization systems for wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 6–12, 2007.

[60] H. Chen, P. Huang, M. Martins, H. C. So, and K. Sezaki, "Novel centroid localization algorithm for three-dimensional wireless sensor networks," in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing.* IEEE, 2008, pp. 1–4.

[61] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, and G. K. Karagiannidis, "A survey on mobile anchor node assisted localization in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2220–2243, 2016.

[62] R. C. Shit, S. Sharma, D. Puthal, and A. Y. Zomaya, "Location of things (LoT): A review and taxonomy of sensors localization in iot infrastructure," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2028–2061, 2018.

[63] F. Al-Turjman, "Positioning in the internet of things era: An overview," in *2017 International Conference on Engineering and Technology (ICET).* IEEE, 2017, pp. 1–5.

[64] X. Li, "RSS-based location estimation with unknown pathloss model," *IEEE Transactions on Wireless Communications*, vol. 5, no. 12, 2006.

[65] F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "Received signal strength-based joint parameter estimation algorithm for robust geolocation in LOS/NLOS environments," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013.

[66] F. Montorsi, F. Pancaldi, and G. M. Vitetta, "Map-aware models for indoor wireless localization systems: An experimental study," *IEEE Transactions on Wireless Communications*, vol. 13, no. 5, pp. 2850–2862, May 2014.

[67] S. Maranò, 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*, vol. 28, no. 7, pp. 1026–1035, Sep. 2010.

[68] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik, "Non-Line-of-Sight identification and mitigation using received signal strength," *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1689–1702, March 2015.

[69] H. Sallouha, M. M. Azari, A. Chiumento, and S. Pollin, "Aerial anchors positioning for reliable RSS-based outdoor localization in urban environments," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 376–379, 2017.

[70] R. Lima, K. Das, and D. Ghose, "Support vector regression based sensor localization using uav," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. ACM, 2019, pp. 938–945.

[71] O. Artemenko, A. Rubina, T. Simon, and A. Mitschele-Thiel, "Evaluation of different static trajectories for the localization of users in a mixed indoor-outdoor scenario using a real unmanned aerial vehicle," in *International Conference on Ad Hoc Networks*. Springer, 2015, pp. 123–133.

[72] Y. Ji, C. Dong, X. Zhu, and Q. Wu, "Fair-energy trajectory planning for cooperative uavs to locate multiple targets," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.

[73] F. Koohifar, I. Guvenc, and M. L. Sichitiu, "Autonomous tracking of intermittent RF source using a UAV swarm," *IEEE Access*, vol. 6, pp. 15 884–15 897, 2018.

[74] S. A. A. Shahidian and H. Soltanizadeh, "Autonomous trajectory control for limited number of aerial platforms in RF source localization," in *3rd International Conference on Robotics and Mechatronics (ICROM)*, 2015.

[75] ——, "Optimal path planning for two unmanned aerial vehicles in DRSS localization," *International Journal of Control, Automation and Systems*, vol. 16, no. 6, pp. 2906–2914, 2018.

[76] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *International Conference on Neural Networks (ICNN)*, 1995.

[77] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm intelligence*. Elsevier, 2001.

[78] P. Goos and B. Jones, *Optimal design of experiments: a case study approach*. John Wiley & Sons, 2011.

[79] F. Pukelsheim, *Optimal design of experiments*. SIAM, 2006.

[80] S.-i. Amari and H. Nagaoka, *Methods of information geometry*. American Mathematical Soc., 2007, vol. 191.

[81] A. Ly, M. Marsman, J. Verhagen, R. P. Grasman, and E.-J. Wagenmakers, "A tutorial on fisher information," *Journal of Mathematical Psychology*, vol. 80, pp. 40–55, 2017.

[82] A. Van den Bos, *Parameter estimation for scientists and engineers*. John Wiley & Sons, 2007.

[83] O. Esrafilian, R. Gangula, and D. Gesbert, "UAV-relay placement with unknown user locations and channel parameters," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 1075–1079.

[84] D. H. Choi, B. H. Jung, and D. K. Sung, "Low-complexity maneuvering control of a UAV-based relay without location information of mobile ground nodes," in *2014 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2014, pp. 1–6.

[85] Z. Becvar, M. Vondra, P. Mach, J. Plachy, and D. Gesbert, "Performance of mobile networks with UAVs: Can flying base stations substitute ultra-dense small cells?" in *European Wireless*. VDE, 2017, pp. 1–7.

[86] D. Romero, S.-J. Kim, G. B. Giannakis, and R. López-Valcarce, "Learning power spectrum maps from quantized power measurements," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2547–2560, 2017.

[87] A. T. Irish, J. T. Isaacs, F. Quitin, J. P. Hespanha, and U. Madhow, "Probabilistic 3D mapping based on GNSS SNR measurements," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 2390–2394.

[88] C. R. Karanam and Y. Mostofi, "3D through-wall imaging with unmanned aerial vehicles using wifi," in *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2017, pp. 131–142.

[89] MacQueen and James, "Some methods for classification and analysis of multivariate observations." vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[90] DeSarbo, W. S., and W. L. Cron, "A maximum likelihood methodology for clusterwise linear regression," *Journal of Classification*, vol. 5, no. 2, pp. 249–282, Sep 1988.

[91] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik, "Identification and mitigation of non-line-of-sight conditions using received signal strength," in *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2013, pp. 667–674.

[92] H. Schütze, C. D. Manning, and P. Raghavan, "Introduction to information retrieval," in *Proceedings of the international communication of association for computing machinery conference*, 2008, p. 260.

[93] Y. d. J. Bultitude and T. Rautiainen, "IST-4-027756 WINNER II D1. 1.2 V1. 2 WINNER II Channel Models," *EBITG, TUI, UOULU, CU/CRC, NOKIA, Tech. Rep., Tech. Rep*, 2007.

[94] "White paper: LTE Unmanned Aircraft Systems," Qualcomm, Tech. Rep., 2014.

[95] "White paper: Drones and networks: Ensuring safe and secure operations," Ericsson, Tech. Rep., 2018.

[96] S. Hayat, C. Bettstetter, A. Fakhreddine, R. Muzaffar, and D. Emini, "An experimental evaluation of LTE-A throughput for drones," in *Proc DroNet*, 2019.

[97] E. Bulut and I. Guevenc, "Trajectory optimization for cellular-connected UAVs with disconnectivity constraint," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2018.

[98] Y. Zeng and X. Xu, "Path design for cellular-connected UAV with reinforcement learning," *arXiv preprint arXiv:1905.03440*, 2019.

[99] Q. Feng, J. McGeehan, E. K. Tameh, and A. R. Nix, "Path loss models for air-to-ground radio channels in urban environments," in *IEEE 63rd Vehicular Technology Conference*, May 2006.

[100] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[101] K. Haneda *et al.*, "5G 3GPP-like channel models for outdoor urban microcellular and macrocellular environments," in *Proc VTC*, 2016.

[102] J. Thornton, D. Grace, T. K. C. Spillard, and T. Tozer, "Broadband communications from a high -altitude platform: the European HeliNet programme," *Electronics & Communication Engineering Journal*, vol. 13, June 2001.

[103] D. Grace, M. H. Capstick, M. Mohorcic, J. Horwath, M. B. Pallavicini, and M. Fitch, "Integrating users into the wider broadband network via high altitude platforms," *IEEE Wireless Communications*, vol. 12, no. 5, pp. 98–105, Oct 2005.

[104] Project Loon, https://x.company/loon/.

[105] Facebook Aquila, https://en.wikipedia.org/wiki/Facebook_Aquila.

[106] Nokia Bell Labs, "Drone Deployment of a Nokia Bell Labs F Cell," https://youtu.be/LVoYdwS4xok, 2016.

[107] Nokia and EE, "Pushing the limits of technology with drones," https://www.nokia.com/en_int/blog/nokia-ee-pushing-limits-technology-drones, 2016.

[108] AT&T, "AT&T's Cell on Wings (COW) in Action," https://youtu.be/WEIplszMzoI, 2017.

[109] S. Chandrasekharan, K. Gomez, A. Al-Hourani, S. Kandeepan, T. Rasheed, L. Goratti, L. Reynaud, D. Grace, I. Bucaille, T. Wirth, and S. Allsopp, "Designing and implementing future aerial communication networks," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 26–34, May 2016.

[110] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: Design, deployment and performance analysis," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[111] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for UAV-enabled multiple access," in *IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[112] OpenAirInterface, https://www.openairinterface.org.

[113] AT&T, "AT&T's Cell on Wings (COW) in Action," https://www.ettus.com, 2017.

[114] EURECOM, https://www.youtube.com/watch?v=GI_lOsg_qmQ.

[115] E. Yanmaz, R. Kuschnig, and C. Bettstetter, "Channel measurements over 802.11 a-based UAV-to-ground links," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*. IEEE, 2011, pp. 1280–1284.

[116] E. D. Klerk, *Aspects of semidefinite programming: interior point algorithms and selected applications.* New York, USA: Springer, 2006, vol. 65.

[117] U. Prells, M. I. Friswell, and S. D. Garvey, "Use of geometric algebra: compound matrices and the determinant of the sum of two matrices." in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 459, no. 2030. The Royal Society, 2003, pp. 273–285.

[118] G. Strang, *Introduction to linear algebra.* Wellesley, MA: Wellesley-Cambridge Press, 1993, vol. 3.