Establishing and Optimising Unmanned Airborne Relay Networks in Urban Environments

by
Pawel Ladosz

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of
Doctor of Philosophy of Loughborough University

February 2019

Abstract

This thesis assesses the use of a group of small, low-altitude, low-power (in terms of communication equipment), fixed-wing unmanned aerial vehicles (UAVs) as a mobile communication relay nodes to facilitate reliable communication between ground nodes in urban environments. This work focuses on enhancing existing models for optimal trajectory planning and enabling UAV relay implementation in realistic urban scenarios. The performance of the proposed UAV relay algorithms was demonstrated and proved through an indoor simulated urban environment, the first experiment of its kind.

The objective of enabling UAV relay deployment in realistic urban environments is addressed through relaxing the constraints on the assumptions of communication prediction models assumptions, reducing knowledge requirements and improving prediction efficiency. This thesis explores assumptions for urban environment knowledge at three different levels: i) full knowledge about the urban environment, ii) partially known urban environments, and iii) no knowledge about the urban environment.

The work starts with exploring models that assume the city size, layout and its effects on wireless communication strength are known, representing full knowledge about the urban environment. It combines a mathematical model with a non-linear model predictive (NMPC) based trajectory planner for path planning and particle swarm optimisation (PSO) for positioning. This approach shows a stable performance improvement, and it has been used for the rest of the work with minor changes. A new communication metric, modified global message connectivity, is also proposed to enable simultaneous performance improvement for connections under certain threshold.

Fully predictable communication strength shows very good performance, but it requires a good knowledge about the urban environment, which is usually not the case. A new method of reducing the level of knowledge requirements is introduced by using a probabilistic based communication model with a measurement-based approach. Parameters of the probabilistic communication model are updated depending on one of the four pre-defined urban environment types. Neural network is used to determine which of the four pre-defined urban environment types to use based on signal strength reading from the ground. This approach shows a similar performance to the existing probabilistic model with correct urban environment type estimation, and an improved performance to an incorrectly estimated urban environment type.

In the final approach, the study of measurement-based communication models is continued, where UAVs fully learn how to predict wireless signal strength between arbitrary points in the air and on the ground. This is done to fully eliminate a need for knowledge about urban environment. Only fixed ground nodes are considered in this case due to high computational demand. With the compromise, the prediction and path planning can be done almost in real time. Through careful design, the proposed UAV deployment techniques are tested and verified in simulated urban environment.

Acknowledgements

This PhD has been an amazing journey and I would like to express my gratitude for all the support, guidance and encouragement I received from many significant people.

First and foremost, I would like to express my deepest gratitude to my supervisors, Professor Wen-Hua Chen, Doctor Hyondong Oh, and Doctor Cunjia Liu. Professor Wen-Hua Chen provided great guidance on the research direction, while gave me the freedom of exploration. Doctor Hyondong Oh guided me through the whole PhD with his supervision on ideas, future directions, as well as detailed results analysis. Doctor Cunjia Liu gave me critical feedback for the controllers and practical aspect of implementing this work.

I also wish to thank Doctor Gan Zheng for the fruitful discussions regarding the feasibility of the proposed approach from the communication community perspective. This help allowed me to gain a fresh insight into modelling and quantifying wireless communication in a better and more realistic way.

Special thanks go to Matthew Coombes, William Eaton and Michael Hutchinson for the countless hours we spent together to develop the experimental testbed in the lab. Without those inputs, none of those experiments presented in the thesis would be possible.

Additionally, I would like to thank CDT-EI for sponsorship and help and support throughout my PhD. Their help made my PhD a better, smoother experience. Further, the extensive training offered by the CDT gave me skills which not many other PhD students have.

In the end, I would like to thank my dear Lisha Tan and my family for their never-ending support, encouragement and inspiration throughout my PhD.

Contents

List of Figures

List of Tables

List of Abbreviations

BATMAN	Better Approach to Mobile Ad-hoc Network
CEO	Cross Entropy Optimiser
GA	Genetic Algorithm
GMC	Global Message Connectivity
GP	Gaussian Process
KF	Kalman Filter
LAP	Low Altitude Platform
LoS	Line of Sight
LTE	Long-Term Evolution
mGMC	Modified Global Message Connectivity
MILP	Mixed Integer Linear Programming
MST	Minimum Spanning Tree
NLoS	Non-Line of Sight
NMPC	Non-Linear Model Predictive Control
NN	Neural Network
PSO	Particle Swarm Optimisation
RH	Receeding Horizon
ROS	Robot Operating System
ROTH	Run on Target Hardware
RSSI	Recived Signal Strength Indicator
SNR	Signal-to-Noise Ratio

UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
WCC	Worst Case Connectivity
WSN	Wireless Sensor Network

List of Symbols

a	random factor, input variable
$\overline{a}$, $\overline{b}$	factors depending on α_0, β_0 and γ_0
$A^{'}$	Adjacency matrix
b	factor to prevent too fast convergence
B	bandwidth
c	Speed of light in a vacuum
C	Population ofr roulette based selection, cost for a given control sequence,Antenna gains and shadowing factor
C_v	Cost of violating conststraints
d	Distance
D	Number of inputs to neural network, Distance
$\mathcal{D}$	Set of measured communication channels strengths
$E()$	error function
f_c	Central frequency
f_{GP}	Gaussian Process Function
f	solution fitness
F	Set of solution fitnesses
Fc	Set oif cumulative fitness function
F_k	State transition matrix
g_0	Population
g_c	Crossover point
G	Position of ρ-quartile in a number of samples and M, Channel gain
Gm	Chromosome from a breeding population
h	multipath fading, height of UAV, height of buildings
$h()$	actiation function
$\mathbb{I}$	indicator function
J_{GMC}	Global Message Connectivity performance index
J_{WCC}	Worst Case Connectivty performance index
J_{mGMC}	Modified Global Message Connectivity performance index
J_d	Overall performance index
J_e	Cost of broken constraints

$k()$	Covariance function
k_1, k_2, g_1, g_2	Communication Properties for given urban environment type
K	Boltzman constant
L Free space path loss	
l_b	length of LOS block inside the building
m	Number of UAVs
$m()$	Mean function
M	Population size, number of neurons in hiddne layer, sample size
$\mathbb{M}$	Number of nodes in the network
N	horizon step number
N_t	number of observations
n	Number of ground nodes
p_c, q_c and r_ω	constant weighting factors for NMPC based trajectory planner
P	Transmission Power
P_d	Minimum communication probability
$P(LOS)$	Probability of being in LOS
P_r	Probability of succesful communication, recived singal strength
P_s	Probability of succesful communication
P_t	Probability treshold
$\mathbf{P}$	Probability in Bernoulli random number
$Q()$	Complimentary error function
r	Random number, Ground distance
T	ambient temperature
T_s	time constant
u_ω	a command input in form of turning rate
U	Control input sequence
U_p	Chromosome of control sequence
$\mathbf{U}$	set of possible changes in control sequence input in a sample M
v	Velocity, Velocity of particles
v_{max}	Maximum UAV velocity
v_{min}	Minimum UAV velocity
$\mathbf{v}_k$	Measrument noise
W	Weighted edge cost
W^b	Weighted edge cost for building obstruction
W^d	Weighted edge cost for minimum communication probability
W^h	Weighted edge cost due to UAV height
W^{pb}	Penalty cost due to LOS block
W^{ph}	Penalty cost due to UAV height
w	Weight
x	UAV x-position

$\bar{\mathbf{x}}^{g,pos}$	3D Position of UGV
$\bar{\mathbf{x}}^{pos}$	3D position of the UAV
$\mathbf{x}$	State of the UAV
$\mathbf{x}^g$	State of the ground vehicle
X	Position of particles
y	UAV y-position, outputs from neural network
$\mathbf{z}_k$	GPS state
α	Path loss factor
α_0	Ratio of the built-up land area to the total land area
β	used for computing mGMC perfomance index
β_0	Mean number of buildings per unit area
γ	Minimal acceptable link quality
γ_t	Threshold value for CEO algorithm
γ_0	Scale factor that describes the building heights in the Rayleigh probability density function
Γ	signal-to-noise ratio
Δu_ω	the maximum allowable control input difference for heading rate
Δu_v	the maximum allowable control input difference for velocity
Δ_b	Control input change in binary form
ε	correction factor for PSO
η_k	a process noise
θ	Elevation angle
$\boldsymbol{\theta}$	Set of hyperparameters
μ	Noise power
μ_p	Gaussian Process predicted mean
μ_g	Gaussian Process predicted mean approximated as a surface
ρ	Parameter to be tuned for the best performance in CEO algorithm
σ	observed noise, standard deviation
$\sigma(a)$	sigmoid function
σ_p	Gaussian Process predicted variance
σ_x and σ_y	Standard deviations of positions of x and y, respectively
τ_ω	Constant to account for actuator response delay in yaw
ψ	heading
Ψ	Shadow fading
ω	inertia of particle, yaw rate of an UAV
ω_{min}	min yaw rate of an UAV
ω_{max}	max yaw rate of an UAV
$\{\}_{ij}$	Between nodes i and j, between neuron i and j
$\{\}_j$	Node j
$\{\}_{j0}$	Activation parameter

$\{\}_{LOS}$	in LOS
$\{\}_{NLOS}$	in NLOS

Chapter 1

Introduction

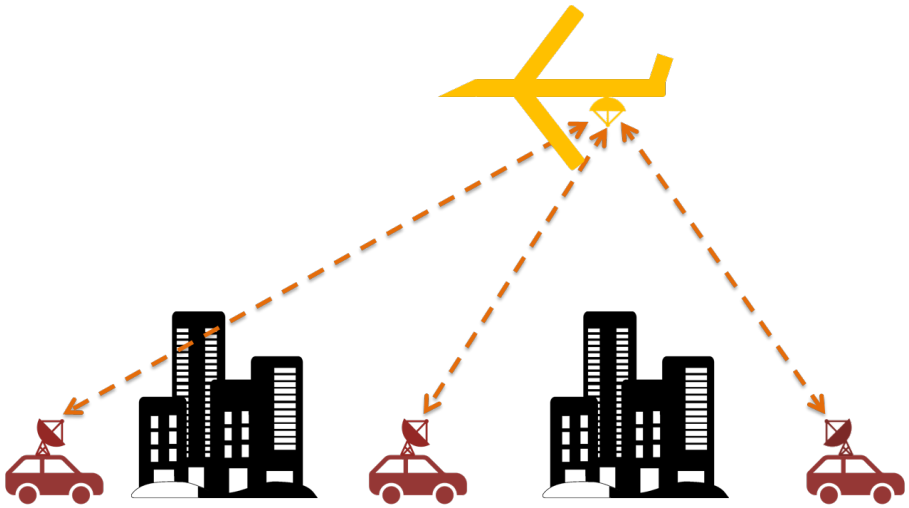


Figure 1.1: Illustration of the communication relay scenario in urban environment.

1.1 Background and Motivation

Unmanned aerial vehicles (UAVs) applications have been progressing at a significant pace In recent years. Thanks to by the significant reduction in costs and operating complexity, UAVs became accessible for the mass market with the low training requirement.

With the reduced entry barrier, UAVs have been applied in fields like film making, remote sensing, security, search and rescue; future applications like autonomous delivery have gained considerable attention from both academy and industry.

There exists limited research on the application of UAVs as wireless communication relays due to prohibitively high costs and training requirements in the past. With the significant cost reduction in UAVs, relays become a feasible temporary communication infrastructure when ground-based stations or satellites are unavailable or severely limited in bandwidth. Such scenarios could occur in:

- Infrastructure damage caused by natural disasters or military deployments;
- Insufficient bandwidth within the existing infrastructure due to extraordinary demand during public events.

In general, temporary relays missions can be split into missions in open fields and urban environments. Benefiting from the open environment and lack of Line of Sight (LOS) obstructions, open field environments can be relatively easily served by a ground-based stationary or mobile radio platform [1]. Provision of wireless communication in urban environments is much more challenging due to the physical obstruction and occlusion by buildings, resulting in significant transmission delays, limited coverage and high transmit power demand [2].

To mitigate issues in urban environments, two technologies have been used traditionally: ground-based radios and satellite communications. Both are liable to be obstructed by buildings, thus reducing signal strength significantly. Moreover, satellites are also limited by the nature of pre-planned orbits. Compared with the traditional methods, using UAVs as wireless communication relays is a promising alternative as they can be deployed in the mission area rapidly without reliance on existing infrastructure and relocated easily to adapt to dynamic communication demands and environments.

This thesis proposes the use of a group of small, low-altitude, low-power (in terms of communication equipment) fixed-wing UAVs as a mobile communication relays nodes to facilitate reliable communication between ground nodes in an urban environment. In this work, the focus is optimisation of the deployment of the UAVs, including its positioning and trajectory planning, by improving the communication prediction models of communication strength on the communication strength between arbitrary points in the air and on the ground. This improvement is achieved by introducing a new way of quantifying communication improvement in the group of ground nodes and employing measurement based methodologies built on online learning approaches.

1.2 Research Methodology

For trajectory planning and positioning for the UAVs relays, information or prediction of the communication strength between arbitrary points in the air and on the ground becomes quite important. Predicting the communication strength between those arbitrary points is very challenging in urban environments, as a wireless signal is likely to be absorbed, reflected and refracted by buildings. Modelling those effects precisely is currently impossible since a wide range of factors like signal frequency, materials and obstacle shapes may affect the communication strength in various ways in different environments [3].

Previous research explored several methods to obtain the prediction on communication strength between two nodes by modelling the building absorption, reflection and refraction effects in various manners. Those approaches can generally be divided into two categories:

model-based approach and measurement-based approach [4]. Existing methods in either approach are not ready for implementation in a realistic scenario due to the prerequisite or ongoing data requirements as discussed below.

1.2.1 Model-based Approaches for Known Urban Environment

The main limitation of model-based approaches is a large number of parameters required for computing communication quality between two nodes, which are challenging to estimate before the mission or online. Mainly, when model-based approaches are used in urban environments, usually some knowledge about positions, shape, materials and their effects on wireless communication are necessary. The knowledge available is generally not sufficient for good positioning and trajectory planning.

Model-based approaches can be subdivided into the range-only and channel propagation [4].

The range-only approach such as [5,6] defines a radius within which communication is assumed to be possible. In the channel propagation approaches [7–10], the communication strength usually depends on the distance and the existence of direct LOS between nodes. Another channel model approach is the probabilistic-based method also termed the low-altitude platform model proposed by [11], which approximates the probability of LOS occurrence based on the types of urban environments.

In this work, when knowledge about the urban environment is used, it is referring to the prediction of air to ground communication links in an urban environment. When known urban environment term is used, it means that the signal strength between air and ground can be fully predicted using a model-based approach. This implies that all the knowledge needed for a good air to ground communication strength prediction is available such as positions and sizes of buildings and their effect on wireless communication strength. Partially known urban environment means that some parameters for the model for air to ground communication strength prediction are known, while others are unknown and need to be either supplied or predicted during relay mission. Finally, an unknown urban environment means that either no model or very naive model is available for air to ground prediction and needs to be fully learned during relay mission time. Also at this stage, it is worth clarifying the difference between trajectory planning and positioning. Positioning is defined as finding a single point UAVs should stay for a prolonged period of time. Trajectory planning is defined as finding a trajectory (a group of waypoints) which the UAV should follow for a given period of time, with possible updates throughout the mission.

1.2.2 Measurement-based Approaches for Unknown and Partially Known Urban Environment

Measurement-based approaches are utilised to cope with the difficulties of estimating communication parameters involved in implementing model-based approaches. This is achieved by using UAVs to collect the signal strength data from the ground nodes during the mission. The data collected is used to model air-to-ground signal strength online, and trajectory planning is performed on the created model.

Measurement-based approaches can be subdivided into gradient following approaches and learning approaches.

In the gradient following methods such as in [1], the UAV collects the signal strength from ground nodes to calculate the signal strength gradient and move towards the optimal position gradually. Meanwhile, learning- based approaches such as [12,13] rely on collecting the signal strength data to create or update a communication model using machine learning techniques.

While measurement based approaches offer to solve a number of parameters issues of model-based approaches, they have two significant limitations. First, measurement-based approaches are mainly used for stationary environments because of the high data volume required for learning and prediction for mobile ground nodes. Second, gradient following methods are unable to cope with non-linearities and discontinuities in the wireless signal introduced by urban environments.

1.3 Research Contribution

This thesis assessed the UAVs relays performance in complex urban environments and enhanced the existing communication prediction models for optimal trajectory planning. Those enhancements make it possible to implement UAVs relays in realistic scenarios. The first experiment of this type was developed and conducted to show the performance of UAVs relays in a simulated urban environment. This thesis makes the following main contributions in this area.

- A new way of measuring communication performance for the group of ground nodes, modified global message connectivity (mGMC) is devised. In previous work, connection performance improvement was constrained to improving either the weakest connection or the sum of all connections. With the proposed metric, all connections with sub-par performance can be dealt with simultaneously, thus offering benefits of both improving weakest connections and the sum of all connections.

- Neural network for updating the probabilistic model to predict wireless signal strength in the urban environment is proposed. This approach is a combination of a learning-based measurement technique with a probabilistic low altitude platform (LAP)

communication model [11]. By the combination of model-based approaches and measurement-based approaches, this hybrid approach overcomes the limitation of heavily relying on communication parameters known a priori, in the model-based approaches, and that of high data volume requirements, in the measurement-based approaches. This hybrid approach makes it possible to apply the measurement-based technique to dynamic scenarios where ground nodes are moving.

- Predicting a communication map in a complex urban environment using the Gaussian Process technique is proposed. Two different schemes of using the Gaussian Process (GP) framework to assist UAV trajectory planning are compared: i) pre-scanning followed by the Non-linear model predictive control (NMPC) trajectory planner with GP, and ii) the NMPC trajectory planner with GP. The first method initially performs pre-scanning of the region of interest to build an accurate communication channel map, then plans the optimal relays trajectory with this map. The second method updates the communication channel map and trajectory planning continuously using online measurements and GP during the flight. As shown in chapter 4, the first shows similar communication improvement amongst ground nodes as model-based approaches while second from fast start to relaying mission. For the first time, the performance, benefits and shortcoming of such an approach in an environment with complex wireless signal strength are compared with other approaches, including the state-of-art technique proposed in [11].

1.4 Outline

The chapter outline is presented in Fig. 1.2. The diagram presents an overview of changes of key ideas in three concept streams (indicated in green columns on the right) across chapters of this thesis (rectangles of different colours stretching from left to right). The trajectory planning and positioning stream show how optimisation of trajectory and positioning evolves across the chapters. Similarly, ground nodes mobility show whether ground nodes can move or are stationary in a given chapter. Finally, the communication modelling concept stream reflects three different communication models used throughout the chapters in this thesis. The remainder of the thesis is organised as follows

- Chapter 2. In this chapter, relevant literature is reviewed, and the problem of optimising trajectory of UAVs for communication relays in urban environments to be addressed in this thesis is defined.

  The literature review is split into three sections: wireless communication in an urban environment, trajectory planning and positioning and communication modelling for UAVs. The first section reviews the complexity of behaviour of a wireless network in urban environments. The complexity arises as wireless communication is affected by multiple phenomena such as diffraction, reflections and absorptions. In the second

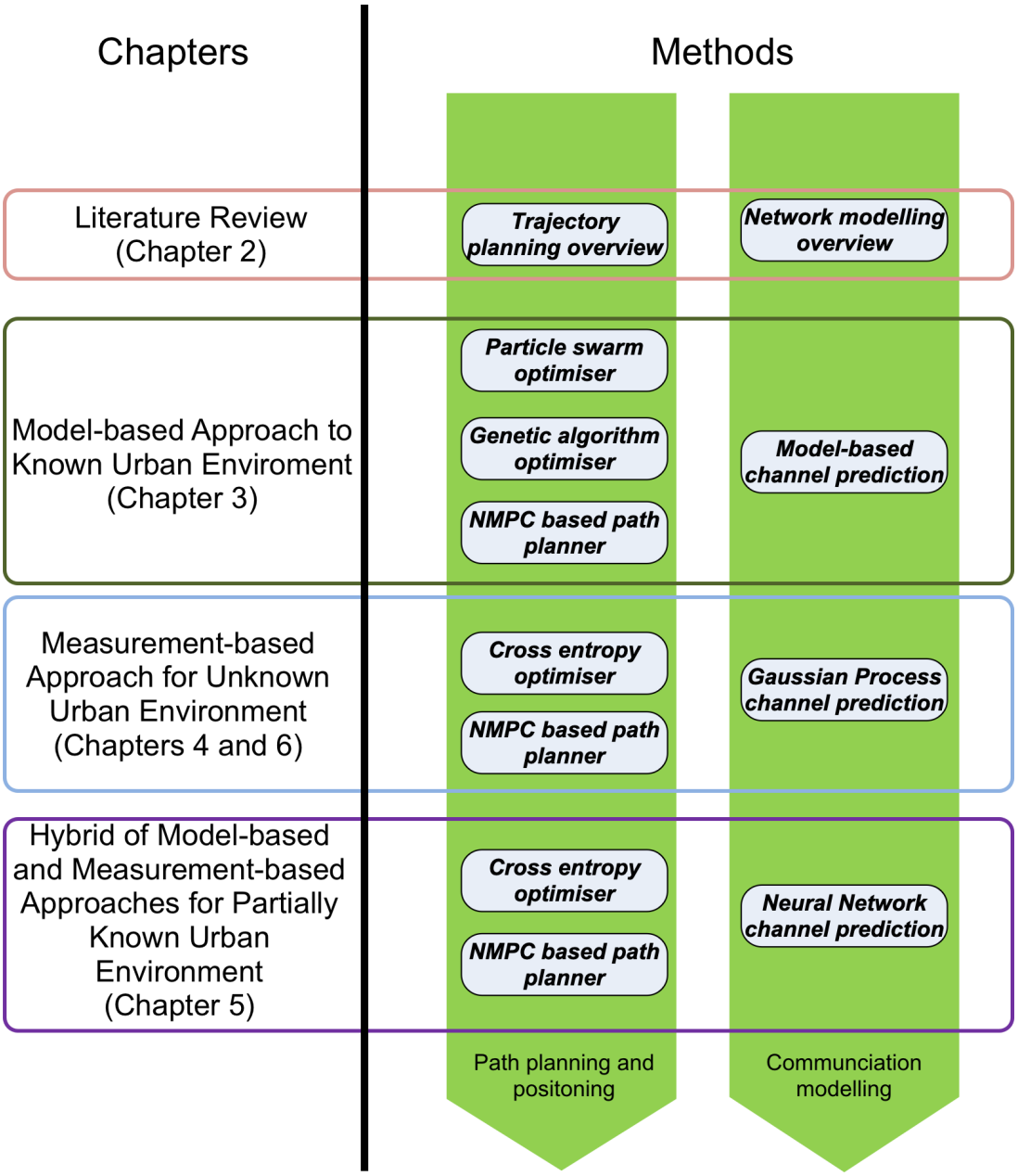


Figure 1.2: Overview of chapters in this thesis

section, issues with determining trajectory and position planning are considered. The complexity of the problem is mainly caused by the vast dimensions of the problems, limited available knowledge regarding ground nodes positions and trajectories, and computational time. The third section introduces the existing attempts to solve the problem of predicting wireless signal strength between arbitrary points in various types of environment.

The literature review reveals that the current communication prediction models could not cope with urban environments adequately. To deploy UAVs for communication relays in an urban environment successfully, it is essential to address the challenges of predicting communication strength between any pair of points on the

ground and in the air.

- Chapter 3. In this chapter, the research focuses on optimally deploying a UAV using particle swarm optimisation (PSO) for stationary ground nodes and NMPC for mobile ground nodes. PSO, genetic algorithm (GA), NMPC based trajectory planner and model-based wireless communication strength prediction are introduced and assessed. For stationary ground nodes, a PSO based positioning algorithm is developed to deploy UAVs as a optimal communication relays. This approach is extended to the NMPC based trajectory planner with GA being used to find a trajectory of UAVs with mobile ground nodes. Compared with stationary ground nodes, this is a more realistic and common situation encountered by UAVs. In this chapter, the feasibility of the proposed PSO positional planner and the proposed NMPC based trajectory planner is investigated in terms of computational time and performance. Additionally, the performance of the proposed mGMC is assessed in this chapter. The experiment is carefully designed and presented at the end of this chapter, where a quadrotor UAV is used in an indoor area to relay communication between three ground vehicles.

- Chapter 4. This chapter is a further exploration of removing constraints of UAV communication strength prediction in urban environments. Further to previous efforts of relaxing the assumption of sufficiently good knowledge about environments, a situation where UAVs has no knowledge about communication loss in the operational environment is considered. In this case, a communication model has to be fully learned online during the mission. Due to the computational complexity involved in this situation, only stationary ground nodes are considered in this study.

- Chapter 5. This chapter improves the NMPC based trajectory planner with a new optimiser called cross entropy optimiser. It also introduces a new approach to compute wireless communication signal strength using a Neural Network (NN). The NN is combined with a probabilistic communication performance metric to predict communication model parameters during a mission. Such an approach allows us to relax the assumption that sufficient knowledge about the operational environment is available, while still coping with mobile ground nodes.

- Chapter 6. This chapter is devoted to the experimental setup and tests of the proposed GP channel prediction approach. In an indoor laboratory, a small scale urban environment including building blocks is set up and simulated. The experiment involves a quadrotor UAV and two ground nodes in a simulated city. To mimic communication loss due to buildings and other obstacles in an indoor environment realistically, multiple water containers were used. Water has good absorption properties in typical bands used for wireless communication. In the experiment, the GP approach was compared to the model-based approach (the model was similar

to that from chapter 3). It was shown that despite assuming no apriori knowledge about the propagation of a wireless signal in this environment GP channel prediction performance is close to the model-based approach.

- Chapter 7. The final chapter summarises the contributions of the thesis and suggests the directions for future work.

1.5 Publications

1.5.1 Journals

1. P. Ladosz, H. Oh and W. Chen, "Trajectory Planning for Communication Relay Unmanned Aerial Vehicles in Urban Dynamic Environments", 2018, " *Journal of Intelligent & Robotic Systems*, pp. 7-25

2. P. Ladosz, H. Oh, G. Zheng and W. Chen, " A Hybrid Approach of Learning and Model-Based Channel Prediction for Communication Relay UAVs in Dynamic Urban Environments", 2019 *IEEE Robotics and Automation Letters*, *Accepted for publication*

3. P. Ladosz, H. Oh, G. Zheng and W. Chen, "Gaussian Process-Based Channel Prediction for Communication Relay UAV in Urban Environments", 2019, *IEEE Transactions on Aerospace and Electronic Systems*, *Accepted*

1.5.2 Conferences

1. P. Ladosz, H. Oh and W. Chen, "Optimal positioning of communication relay unmanned aerial vehicles in urban environments," *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, Arlington, VA, 2016, pp. 1140-1147. doi: 10.1109/ICUAS.2016.7502562

2. P. Ladosz, H. Oh and W. Chen, "Prediction of air-to-ground communication strength for relay UAV trajectory planner in urban environments," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, 2017, pp. 6831-6836. doi: 10.1109/IROS.2017.8206603

1.5.3 Book Chapter

1. P. Ladosz, M. Coombes, J. Smith, M. Hutchinson, "A Generic ROS Based System for Rapid Development and Testing of Algorithms for Autonomous Ground and Aerial Vehicles", 2018 *Robot Operating System (ROS). Studies in Computational Intelligence, vol 778*

Chapter 2

Literature Review

For efficent utilisation of Unmanned Aerial Vehicles (UAVs) as communication relays, multiple problems need to be addressed, such as trajectory planning and positoning, modelling of wireless communication, network topologies, routing protocols and UAV battery management. Among those problems, path planning and modelling of wireless communication are reviewed in detail. However, to fully understand the magnitude of the problem, it is important to first understand the behaviour of wireless signal in an urban environment.

2.1 Wireless Communication in Urban Environment

Wireless communication in an urban environment is characterised by a stochastic difficult to predict behaviour caused by reflections, absorptions and refractions. Experiments have been performed to characterise wireless communication behaviour in urban environments.

First, the effect of different materials used in buildings is discussed. An experiment performed by [14] measured the effect of reflection and absorption brick, tinted glass, clear glass and wall (made of a combination of a variety of materials) for 28 GHz network. It was shown that tinted glass had the highest reflection coefficient of 89%, meaning most of the wireless signal was reflected. Also, concrete drywall and clear glass materials have shown very similar reflection coefficients of about 70%. Finally, it was shown that common office objects walls, doors, cubicles and others significantly affected the behaviour of wireless communication.

Effect of low-built-up homes is shown in [15]. In their experiment, they measured the performance of 5.85 GHz network around three different houses each constructed at different times and made of different materials. Each of the homes has shown different attenuation of the wireless signal. Even theoretically same materials have shown different losses. Sometimes losses difference were twice as high between homes built in two different architectural styles. Another exploration of the wireless signal behaviour around low-built-up areas was performed by [16]. Unlike [15], this work checked frequencies between 2000 and 16000 MHz. It has shown that depending on frequency penetration across buildings can be very different. In general the lower the frequency, the better the penetration. Those

experiments show how different materials are interacting with different electromagnetic wave frequencies can have a significantly different effect on the wireless signal. Thus to be able to predict signal strength, the composition of each building would need to be known.

Wireless communication performance around taller buildings was explored by [17], [18] and [19]. Davidson et al. [18] measured the wireless signal penetration loss for various tall buildings. The transmitter was located between 1 or 2 km away from buildings, while receivers were located inside. It was shown that receivers located at different floors received signal of different strength. Tannis Et Al. [17] explored how the wireless communication signal strength varied for two bands 880 MHz and 1922 MHz in different buildings such as office, mall and hotel. In this work, it is shown that on average signal difference between a line of sight (LOS) and non-line of sight (NLOS) was 3 dB, which means that 3 dB is lost due to buildings.

Series of experiments to test how the wireless propagation is affected before, during and after building collapse, across series of bands was performed in [20–23]. Three different buildings were tested: highrise ex-hotel, stadium and convention centre. The experiment is a simulation of how the wireless signal changes during a natural disaster scenario, which is one of the key usages of UAV relays explored in this thesis. Destruction of buildings resulted in a significant change in communication performance at each receiver site. Most of the time the receivers saw a degradation of signal quality due to material built-up around the receivers. Such a signal strength change could make even perfect model obsolete very quickly.

Overall, it was shown that communication strength performance varies highly depending on the building's shape, size and construction materials. Because of this, the wireless signal has an extremely complex behaviour and is difficult to predict.

Previously effects of simple single buildings cases were presented. Here this discussion is extended to account for multiple buildings or cities. Multiple buildings cause more complex behaviour of wireless communication, as multiple effects are combined at the same time. With multiple buildings, two types of tests were performed: with stationary antennas and with mobile antennas (i.e. mounted on a car or other vehicles).

Experiments were performed where two antennas are placed a certain distance from each other in an urban environment. One such a work in [24] explored how NLOS affects performance at high frequencies such as 23 or 28 GHz. It has been shown that signal in an urban environment has considerable variance through extensive testing in New York and Austin, Texas. Murdock, et al. [25] performed a test at the University of Texas campus which attempted to measure the effects of diffraction. It was shown that with diffraction the likelihood of successful link establishment was 10% to 20% higher than without diffraction. Important remark for the wireless signal propagation was made in [26]. This work showed that, the ratio of windows to building area could have a significant impact on the wireless signal propagation. Moreover, the presence of tinted metal in the windows also affected the wireless signal propagation profoundly. Rapport, et al. [27]

focused on observing the variance of the signal. In this paper, it was shown that, in NLOS, the variance of the received signal strength increases significantly. The difference in signal arrival time was explored in [28]. An experiment performed in New York showed that, due to multipath fading, data arrives at receiver postion much later for NLOS case than LOS case. Wireless communication strength experiments in 700 MHz and 4.9 GHz bands were performed in [19] to asses multipath losses performance for LOS and NLOS around single buildings. Multipath losses are caused by signal taking different routes to arrive at the receiver. It was found that multipath losses in urban environments were independent of whether transmitter-receiver pairs were in LOS and NLOS. With this, it can be seen that buildings not only affect signal strength directly but also contribute to creating multiple paths signal can travel, increasing stochastic behaviour of the wireless signal. Laurila et al. [29] performed an experiment in Helsinki which classified propagation into four different urban environment classes to account for the obstructions by buildings. In this paper, it was shown that the received signal strength indicator (RSSI) is affected by objects next to the base station the most. They have also shown that signal components received over the rooftop of a building are often a result of reflections from other highrise buildings in the area. Groups of buildings cause signal strength to behave in a highly stochastic way; thus the wireless signal strength in an urban environment is difficult to predict accurately. Moreover, it is shown that due to buildings, the wireless signal often arrives with significant delays.

Work in [30, 31] considers moving antennas mounted on a car. In Otto et al. [30] experiment was devised in three different environment types: open fields, suburban and urban. It was shown that distance based loss is similar across all three environments but signal variance the highest in urban and lowest in open fields environment. Similarly, variance of signal strength increases between NLOS and LOS. Two tests were performed in [31] with vehicles driving in the same direction in one test and the opposite direction in the other test, to see how RSSI changes with distance in different environments. It is noted that path loss was the highest in highway environment, followed by the urban and rural environment.

With such a stochastic and complicated behaviour caused by buildings, using a mathematical model for communication prediction is not trivial. Two reviews comparing the accuracy of wireless communication models in an urban environment was performed in [32] and [33]. Wu et al. [32] assessed communication models available for predicting communication strength in an urban environment for Jinan City in China. In this work, several models were compared with the measurement data they collected, with two models showing good fit. However, it was remarked that the fact those two models are such a good fit is down to the geography of the place, i.e. those two models were created in a very similar urban environment. On the other hand, other models developed for substantially different urban environments did not work well. Abhayawardhan et al. [33] presented another review of the suitability of empirical models in an urban environment. It can be

seen again that models do not fit with measurements well. One of the models (ECC-33) performed well in a single urban environment but didn't perform so well in other urban environments. Also, it is worth noting even for these with good fitness, it is only achieved on average but not at individual point.

2.2 Communication Modelling for UAVs

To effectively plan the UAV relay trajectory, it is essential to predict the air-to-ground communication channel quality for arbitrary UAV positions. In the literature, several approaches were used for wireless communication channel prediction. They can be primarily divided into the model-based and measurement-based approaches [4].

2.2.1 Model Based Approaches

In the model-based approaches, communication channel strength between nodes is calculated by using a pre-defined model with known communication parameters which are dependent on the mission environment. Model-based approaches can be further subdivided into range-only and channel propagation methods.

Range-only

The range-only approach is one of the simplest methods of defining communication model. This type of methodology is also referred to as a disc-based model [4]. It relies on defining a radius within which communication is possible. There were several approaches which used range only models to the great effect.

Simple range-only models were explored in [5, 6, 34]. In those approaches, communication is assumed to be possible only within a certain radius away from the node. It was shown that the range only method has high computational efficiency while providing reasonably good performance in terms of improving communication performance.

Burdakov et al. [35] extended the range only method with added LOS obstruction element. They assumed that communication is possible within a given radius only if LOS is present. With this simple extension, they were able to use the range-only approach in an environment with obstacles, with a minimal computational cost increase.

Another approach of using range-only models to cope with the urban environment was presented in [36]. In their approach, the radius of possible communication changes based on the type of urban environment given node is in. In general, the more complex the environment is, the shorter the possible communication radius will be.

Despite the fast computational speed, range-only approaches fail to account for many essential elements of communication prediction. For example, it is challenging to account for directional effects such as antennas directionality. Moreover, range-only approaches, despite some attempts, struggle in urban environments. While simple non-line of sight

(NLOS)/LOS can be considered, more complex effects such as diffractions cannot be dealt with.

Channel Propagation

Channel propagation methods are a popular way of modelling communication performance. Those methods were created to account for more factors affecting communication compared with range-only methods. In this method, a mathematical model is used to model the channel propagation between different nodes. The models usually account for various things such as distance, propagation loss and shadow fading.

A simple channel propagation method was presented in [37]. In their model, data rate between various nodes is computed based on the distance between nodes, as well as maximum bandwidth a node of concern can support.

Wu et al. [38] proposed a very comprehensive channel model for general usage. Their model accounted for factors such as azimuth and elevation between two nodes (to account for different antenna propagation properties), the distance between two nodes, general obstructions by buildings as well as Doppler frequencies to account for refractions caused by buildings.

Kim et al. [7] and Han Ee al. [8] use a model which is based on the distance between two nodes, transmitted power, background noise and channel gain. Model is then used to give a probability of achieving desired link strength. Such an approach allows accounting for desired communication strength from the user in a probabilistic manner.

Multiple antennas systems were considered in [39]. Their models calculated predicted SNR using a channel matrix (number of communication channels available based on numbers of antennas in the system). The model was initially developed by [40] and [41].

SPLAT! [42] is a simulation software used in [43,44] to predict communication strength between air and ground. SPLAT! is a popular software used to predict communication strength between two points. It can account for NLOS obstructions as well as simple distance-based models. NLOS obstructions are dealt with in a simple manner where if two nodes are in NLOS, their communication is not possible.

Mozaffari et al. [45] described the positioning of an UAV in an urban environment based on a probabilistic communication model from [46]. The model is based on distance-based loss, antenna gain, transmitted power and finally random Gaussian variable representing the noise. The noise component is different depending on whether two nodes are in LOS or not. Additionally, in this work, it is shown that the probability of being in LOS is dependant on the angle between air and ground vehicles. The similar probabilistic approach is presented in [47].

Channel based methods can be used to account for the majority of effects of an urban environment on the communication. However, to achieve that, they usually require i) a good knowledge about the urban environment and ii) a good estimation of many paramet-

ers. As described in Sec. 2.1, communication strength between two arbitrary points in a city is affected by many factors such as number, positions and makeup of obstacles (e.g. buildings) and even the presence of trees and bushes. It is just impossible to account for all of them in any model. Additionally, it is not easy to correctly estimate model parameters suitable for a mission in a given environments a priori. With the wrong parameters, the performance of the model is significantly depreciated. For those reasons, it is difficult to use channel-based models to full capacity.

2.2.2 Measurement Based Approaches

To address the issues with the model-based approaches, measurement-based approaches were introduced. They rely on collecting communication data to create and update the communication prediction models in real time. They can be divided into the gradient following and the learning methods.

Gradient Based

In the gradient following methods such as [1, 48, 49], measurements (e.g. SNR or RSSI) are used to calculate the gradient, and the UAV follows the gradient to a better channel position gradually. Eventually, optimal position is reached when no increase of RSSI is possible by any movement of the UAV.

A gradient following method based on potential fields was presented in [50]. This work is about communication relay between two ground nodes using a chain of multiple ground vehicles. To generate UAV guiding potential fields, a communication measurements strength from other ground vehicles are used.

Gil et. al. [51] showed a method of determining communication quality using both signal strength and channel phase. That allows determination of change signal quality much quicker as channel phase changes can be observed faster than signal strength changes. Additionally, it uses synthetic aperture radar to determine a signal direction for further improvement of trajectory planning.

Gradient following communication models are characterised by low computation effort and not relying on fitting appropriate models to appropriate urban environments. However, gradient following methods is unable to cope with non-linearities and discontinuities in the signal strength which are present in urban environments.

Learning Based

To cope with the limitation of gradient-based methods, learning based methods were devised to allow measurement based methods to cope with more complex scenarios such as urban environments.

Hsieh proposed one of the first learning methods in [52]. This method relied on splitting an urban environment into grid points. The grid would generally follow the outline of the

roads in a city to guarantee all points are reachable. Robots would then survey the grid to find connection strength between grid points, to be later used for trajectory planning.

Carfang et al. [53] used scaled unscented Kalman filter to predict parameters in the communication model. It used Rappaport's long-distance model for signal estimation and then used a Kalman filter to estimate parameters in the communication model equation. The approach was shown to work during the experimental flight where it was shown parameters converge relatively quickly.

Another approach proposed by Carfang et al. in [12, 54] used the Gaussian process (GP) to iteratively learn radio frequency environment to improve the efficiency of the data ferrying UAV. Network strength is said to consist of two components RF propagation and white noise. White noise is dependant on location and thus can be estimated using a Gaussian process. In a simulation, it was shown that the theoretical maximum bandwidth provided by the ferry could be 21 Mbps, while in reality at the beginning of the experiment it was 6 Mbps. After a few iterations of algorithm 21 Mbps was eventually reached.

Fink et al. [55–57] proposed several other approaches using GP Their method was using a group of mobile robots estimate signal strength for a stationary ground node. GP was used in the estimation which was proven to work well for this prediction.

GP was also used in [58–60] for predicting communication strength to support the operation of ground robots. Both approaches used robots to perform a certain mission, while mapping communication strength to ensure communication to the base can be achieved.

Mostofi and her group have developed two ways of predicting spatial channel capacity. The first methodology discussed in [61–63] was using compressive signal theory and the other one [64, 65] was a probabilistic method. The compressive signal theory relied on the transfer of spatial signal data to a space where most of the signal is contained within much fewer terms only. Those terms can be estimated from measured data easier than in spatial domain and transferred back to the spatial domain, where it can be used for signal prediction. A probabilistic method was developed that relies on estimating the shadowing and path loss components for a given area while ignoring multipath fading.

Another approach to learning the RF signal was proposed by [66], which used a methodology called segmented regression. It clustered points based on RSSI values. The advantage of this approach was that instead of trying to use a continuous function to approximate discontinuous communication strength function, it could approximate discontinuous function directly. Such an approximation was a more accurate representation of wireless signal behaviour, resulting in better planner performance.

Learning methods can cope well with urban environments as it learns communication performance during the mission and it can cope with the discontinuous signal from ground nodes. The disadvantage of learning based methods is the time needed by the UAV to learn communication performance.

2.2.3 UAV Network Test

To complete a discussion on wireless communication, experiments involving UAVs are discussed hereafter. It is worth noting that using UAVs to measure signal strength variation in an urban environment is challenging at the time of writing as UAV flight is restricted by law limitations heavily. Nevertheless, for the sake of completeness, it is important to consider the performance of UAV even if it is in open environments such as in Yanmaz et Al. [67]. There also exists one work which considers an urban environment in a limited capacity in VanDerBergh et Al. [68].

Yanmaz et all. [67] performed an experiment involving UAV and a ground station. They measured how radio communication changes as a function of distance and height of the UAV. The signal was linear with distance, but it still has a significant variation despite there being no buildings or obstacles in the line of sight.

Van Der Bergh et Al. [68] is a paper about using Long Term term evolution (LTE) networks for UAV relay between many nodes. Their work focused on measuring the performance of such a relay without any optimal placement algorithms. They flew UAV up to a 120 m and detected communication strength in what seems to be an urban environment (or environment with obstacles), to the nearest LTE station.

2.2.4 Discussion

Multiple methods to predict wireless signal strength were presented. Range-only approaches work very well for a simple open-field environment, where the signal can be simply predicted. Channel propagation methods which add additional elements of prediction to a simple range, can be used in a slightly more complex scenario. However, model-based methods require the estimation of several parameters for them to work well. Measurement-based techniques solve this problem by relying on collecting signal strength data and reaching appropriate to the data which is observed. Out of gradient-based and learning based, only learning based methodologies are suitable for urban environments. Gradient-based approaches would struggle with discontinuities and non-linearities introduced by urban environments.

With those limitations in mind, this work uses channel propagation methods and learning based method. Channel propagation methods work well when there is proper knowledge about an urban environment. Learning based methods are used to supplement the channel model, where knowledge about the urban environment is incomplete or non-existent.

2.3 Trajectory Planning for fixed wing aircraft and Positioning for rotary wing aircraft

This section explores trajectory planning and positioning algorithms for an unmanned aerial vehicle (UAV) relay. The nature of both vehicles dictates this usage. Rotary wing UAVs can hover so it can stay at a predefined the position, while fixed-wing UAV needs to move at a certain speed, so trajectory planning is necessary. Trajectory planning and positioning are necessary to utilise limited UAVs available in a mission effectively.

2.3.1 Positioning for Optimal Configuration of UAV RelayNetwork

Relay positioning algorithms can be divided into three categories: i) convex optimisation [69], ii) bio-inspired heuristics optimisation and iii) other heuristic optimisation. Convex optimisation relies on careful formulation and simplification of the problem so that it can be solved by gradient ascending/descending algorithms. Bio-inspired heuristics optimisation uses bio-inspired algorithms to solve the positioning problems, while other heuristic optimisations rely on either geometry or special assumptions and constraints of the problem to provide a fast suboptimal solution.

Convex Optimisation

One of the key techniques used in convex optimisation is Linear programming used for example in [36, 70, 71]. Flushing et al. presented a mixed integer linear programming (MILP) approach for placing relays in Wireless Sensors Network system [70], where the mission region is split into sub-regions of square shapes. Each square defines the density of nodes within it, and then UAVs are positioned so that they all serve a similar number of ground stations. The algorithm described in [71] used UAVs to increase the second smallest eigenvalue of the Laplacian matrix of the network graph. This lead to improved connectivity in a wireless sensor network. This number also called Fiedler value of the network, measures how well the group of nodes is connected. They used semi-definite formulation with SDPA-M software package to solve the problem.

MILP based approaches are guaranteed to find the optimal solution given that problem can satisfy framework constraints. Moreover, there exist multiple tried and tested MILP solvers which can arrive at the optimal solution quickly. However using MILP requires special formulating a problem in a specific way, which is not always possible.

Another set of approaches can be classified as gradient descent. In those approaches, UAVs follows a trajectory of the decreasing RSSI to reach an optimal position as described in [1, 50, 72]. One of those approaches was shown in [72] where a gradient following controller for a rotary wing UAV was developed. The algorithm was used to find the UAV position which improves all connections among a group of ground nodes. They have performed an extensive indoor experiment using a group of quadrotors and ground sensors.

Dixon et al. [1] used the measured signal to noise ratio (SNR) and a gradient following technique to guide multiple UAVs to optimal positions between two ground nodes. They have performed an outdoor proof-of-concept experiment with fixed-wing UAV. Zavlanos et al. [50] used the potential vector field approach to solve Unmanned ground vehicles (UGVs) positioning problem similarly to gradient following algorithms. They have used a group of UGVs to help communication between two base stations. The key advantage of the proposed approach was that it could run in a decentralised manner.

Gradient-based approaches are very fast to compute as they do not rely on a complex model of UAV kinematics or wireless communication strength and can be run in a decentralised manner with minimum efforts.

The critical advantage of convex optimisation techniques is high computational efficiency and the guarantee of reaching optimal answer given fulfilment of specific requirements. However, fulfilling those requirements is not always possible, resulting in poor performance. This is particularly true in an urban environment where complex communication models introduce non-linearities and discontinuities in cost functions, causing convex optimisation to be stuck in local minima.

Bio-inspired Heuristics Optimisation

For coping with more complex communication models, bio-inspired heuristics optimisation techniques could be used. Bio-inspired heuristics optimisation techniques mostly use optimisation algorithms such as particle swarm optimisation (PSO) and genetic algorithm (GA). Bio-inspired heuristics optimisation techniques trades-in guarantees of optimality for improved performances in complex urban environment scenarios.

Magan-Carrion [34] used PSO to find optimal positions of relays. This approach splits problem into two parts: i) initial guess of where the UAV relay placement might be based on heuristics; ii) The placement is then updated using PSO for further refinements. Such an approach was introduced to allow placement of multiple communication relays (10 or 20) quickly.

Performance of GA and PSO in a small urban environment scenario was compared in [73]. It was shown that these two have a very similar performance in terms of determining optimal positioning of UAVs relays. They have also shown the performance of the proposed methodology in an outdoor experiment with a group of UGVs.

In [74] usage of a multi-subpopulation GA to position UAVs relays in a scenario with multiple ground nodes was explored. In this approach objectives such as fault tolerance, redundancy and network coverage are considered. The approach consists of is a set of smaller GAs, each evolving with different initial algorithm parameters, with the best solution chosen among the set of smaller GAs. It has been shown that their approach is better for this problem than standard optimisers such as PSO.

Bio-inspired heuristics optimisation techniques are ideally suited to cope with more

complex scenarios such as urban environments. In an urban environment, signal strength is not continuous as a function of the position of air and ground vehicles, resulting in a non-convex cost function. Convex based algorithms have limited performance with non-convex cost functions as they tend to get stuck in local minima. Bio-Inspired heuristics optimisation techniques have mechanisms to prevent this issue; however, they sacrifice a guarantee of optimality and computational speed.

Other Heuristic Optimisation

Other Heuristic Optimisation approaches are used to either arrive at the close to optimal answer quickly or to try to cope with non-convex cost function issue. Other heuristic optimisation approaches use either geometric features or a significant simplification of the problem to be able to apply brute force search methods efficiently.

Chandrashekar's [75] approach is based on grouping ground nodes to a set of clusters based on communication strength. UAVs are then used to connect separate groups. With the grouping, the problem was greatly simplified, so brute force approach was applied to find the optimal positioning of relays between clusters.

Another approach which relied on great simplification and brute force usage was shown in [76]. In their scenarios, the coverage provided by a set of base stations was reduced due to a natural disaster occurring. UAVs are used to cover the area that was no longer covered by damaged or destroyed base stations. With such a setup, a brute force solution can be easily obtained by placing UAVs in any non-covered area.

The approach presented in [36] has shown a 3D placement problem, rather than 2D placement in other works, that can be solved successfully using MILP. To mitigate the computational time issue, their algorithm consisted of two phases i) using one-dimensional bisection search to fix UAV altitude, and ii) solving the problem on a given altitude as a mixed- integer non-linear problem (MINLP) problem.

Geometry based heuristics were presented in [77–79] to solve relay placement in an urban environment. In [77] approach relies on first choosing line and angle UAV will be at from ground nodes. Then UAV positions are adjusted on that line until a maximum number of ground vehicles are in LOS. This approach guarantees very fast convergence and low computational cost. Lyu et al. [78] presented a heuristic approach where UAVs are placed between the edges of the polygon which joins all unconnected ground nodes. At the beginning the shortest edges are filled, to be followed with longest edges until all relays are placed. Essentially the idea is very similar to clustering. Nikolov presented approach where UAVs are placed sequentially on the edges of overlapping communication regions between ground nodes [79]. Relay position is then optimised within the overlapping region using a gradient-based approach. Once the optimal position has been found, new UAVs are added until all relays are positioned within the network.

Xu et al. [80] proposed another approach using the heuristic method for relay place-

ment. Their algorithm performs both UAV placement and node scheduling (i.e. at which time which node should transmit data). Their algorithm first fixes a position of the relay, then uses that position to compute optimal scheduling. Based on that they can solve both problems quickly.

The computational speed is a crucial advantage of other heuristic optimisation approaches where the features or characters of a specific problem is fully explored. It is particularly useful if the algorithm is to be deployed on a vehicle where often computational power and communication bandwidth to an external computational station are limited. However, due to a limited scope of formulation, usage of heuristics optimisation is limited to the specific scenarios they were designed for.

2.3.2 Trajectory Planning to search for Optimal Solutions

Path planning is predominantly used for fixed-wing aircraft due to its suitability to fixed-wing kinematics constraints, i.e. fixed-wing UAV needs to keep moving to maintain a suitable lift required for not falling out of the sky. Path planning algorithms can be categorised into convex optimisation, bio-inspired heurisitic optimisation, game theory and other heuristic optimisation.

Convex Optimisation

In convex optimisation for trajectory planning, two techniques are predominant MILP and gradient-based optimisers.

Examples of work using MILP has been shown in [43] and [81]. Gortli et al. [43] investigated trajectory planning using MILP. In their approach trajectory is pre-generated before the mission and is based on communication strength simulation done in communication strength simulation software called SPLAT!. Additionally, they considered factors such as refuelling and collision avoidance. Flushing et al. developed another MILP approach in [81]. In this work, the problem of balancing communication needs and performing other tasks at the same time was considered.

Gradient-based approaches were considered in [82], [7] and [83]. In [82] consideration of energy efficiency for the UAV relay trajectory planning was introduced. They used gradient based convex optimisation and showed that communication energy is negligible compared to the energy used by the UAV during the flight. In their approach, the trajectory planning is done off-line before the mission and then UAV follows it. Kim et al. [7] proposed an approach based on a non-linear model predictive control (NMPC) and gradient-based optimisation. The NMPC based trajectory planner is used to plan the path with mobile ground nodes. Additionally, the Kalman filter is used to predict future trajectories of ground vehicles. The approach proposed in this work can update the trajectory periodically during a mission to account for new positions of ground nodes. NMPC based trajectory planner is one of the most important methodologies used for

UAV trajectory planning due to its low computational load, transferability across many different problem domains and ease of implementation. NMPC trajectory planners have been explored in many applications. Singh et al. [84] used this methodology to navigate between waypoints in a dynamic urban environment. Kange et al. used NMPC methodology for trajectory following in [85]. NMPC was also used for cooperative target tracking by [86]. Autonomous landing and mapping were proposed in [87]. Another popular usage of NMPC methodology is formation flying explored for example in [88] or [89]. Choi et al. [83] used UAVs to minimise the link capacity (i.e. how much data given link can transfer) amongst a group of moving ground nodes. In their approach, a UAV follows the gradient of signal strength to position itself amongst a mobile group of ground nodes. However, ground nodes have to be moving in generally the same direction; otherwise, the optimal position would not be reached.

Trajectory planning is a more complex problem than positioning, and convex trajectory planning is limited to solving simple problems only. Trajectory planning can be thought of as multiple positioning problems, solved together at the same time to generate a group of connected points for UAV to fly to, thus by nature it is a more complex problem. Moreover, some additional constraints need to be considered such as kinematics constraints of the UAV to make sure a designed trajectory can be traversed. All these issues result in a significant increase in computational time to find the optimal trajectory, compared to finding the optimal positioning.

Bio-inspired Heuristic Optimisation

Bio-inspired heuristic optimisation for trajectory planning is used to cope with more complex communication models present in the urban environment.

Carfang et al. [12] used GA to iteratively learn radio frequency environment to improve the efficiency of the data ferrying UAV. This work uses a receding time horizon and GA to predict future UAV trajectory. In a simulation, it was shown that the theoretical maximum bandwidth provided by the ferry could be 21 Mbps, while in reality at the beginning of the experiment it was 6 Mbps. After a few iterations of algorithm 21 Mbps was eventually reached.

Ho et al. [90] used the PSO method to guide a UAV on minimum energy trajectory while maximising the connection time between the UAV and ground nodes. The ground nodes, in this case, were elements of a wireless sensor network and the objective was to minimise transmission power required to transfer data back to the UAV. It was necessary to optimise transmission power as sensors usually have minimal energy storage capacity; thus it is desirable to keep their energy usage to the minimum.

Bio-inspired heuristic optimisation trajectory planning is ideal to cope with complex cost functions in an urban environment while sacrificing the guarantee of optimality of convex optimisation. However, planning a full path requires a significant computational

effort. To cope with this, most bio-inspired heuristic optimisation approaches uses a technique called receding horizon (also called model predictive control (MPC)). Rather than optimising the full path, it defines a time or distance limit within which path planning problem is solved. This significantly reduces the time needed for optimisation as it only solves a part of the problem at a time.

Game Theory

Approaches presented in [91, 92] are a series of work which focused on trajectory planning to improve coverage of mobile ground nodes in open fields environments. This work focuses on non-cooperative games with the objective of adding more ground nodes connections to each UAV while keeping old ones connected (adding new ground nodes has a higher payoff, than keeping old ones).

Roh et al. [93] presented approach about controlling ground vehicles trajectories while improving wireless communication amongst them. In this setup, some of the ground vehicles are said to be performing their mission, while others are controlled to fulfil communications needs. There are two objectives of the game i) improve a number of links between ground vehicles and ii) preserve average data transmision rate of already existing links.

Game theory based approach was used to collect data from a wireless sensor network (WSN) in [94]. In their approach, the objective of the game was to increase mutual information gain by efficiently planning a trajectory for the UAV. Mutual information gain is defined as the amount of new information UAV gains during its flight from WSN.

Choi et al. [95] used differential games to solve UAV relay problem between a moving ground node and the base station. The UAV aimed to connect a ground rover to a base station by adjusting its heading and angle.

Game theory is another technique which can be used to find a close to an optimal solution with low computational load requirements. The key advantage of the game theory is the fact that it can be run in a decentralised fashion. One of the critical problems is the limited ability to cope with uncertainties and inability to cope well with non-linear cost functions.

Other Heuristic Optimisation

Due to the complexity of solving a trajectory planning problem, several heuristic algorithms have been developed to allow quick computation of suboptimal trajectory taking advantages of specific features of a problem under study.

De Freitas et al. [5] and Basu et al. [96] showed an approach based on flocking methodology. UAV is attracted to the last known neighbour if the number of connections to the UAV drops to below a pre-specified threshold. This approach has shown good improvement in communication in large groups of ground nodes with little computational

time.

Optimal transport theory has been implemented by [97] to plan a trajectory for communication relay amongst a group of static nodes in an urban environment. In short, their algorithm can be summarised as i) cluster nodes to UAVs using k-means clustering ii) UAV moves to reduce the distance between it and members of its cluster. Continue i) and ii) repeatedly until UAV doesn't improve its position any more. Finally, optimal transport theory is used to find the most energy efficient trajectories for UAV to follow.

Dubins paths were used by [98] to find trajectories for a group of relay UAVs to facilitate communication between a group of mobile ground nodes and ground station. Using Dubins paths make computation of trajectory very fast.

Gil et al. proposed an approach to place relays in the middle of groups of WSNs [99]. The main contribution of this paper was the usage of reachability analysis to guarantee a certain level of performance for a certain amount of time. Their algorithm is based on reducing the longest communication link (longest in terms of distance). In a open-field environment that is generally the weakest link.

The heuristic algorithm to solve the UAV trajectory planning problem while improving the security of the network was proposed in [100]. Their algorithm relied on splitting the problem into two subproblems: 1) optimising transmission power for a given trajectory and 2) optimising trajectory for given transmission power. The advantage is that the optimal solution can be analytically found for subproblem 1, while optimisation is only needed for subproblem 2.

An auction-based algorithm called consensus based bundle algorithm (CBBA) was proposed in [101]. In auction-based algorithms, task allocation problem (i.e. relay, tracking) is formulated as an auction between agents. Each agent runs independently, and they negotiate assignment of all tasks. As an auction-based algorithm, CBBA has several advantages such as scalability, fast computational time and ability to fix its mistakes. This work is more focused on balancing communication tasks with other mission components rather than trajectory planning.

Heuristic approaches for trajectory planning problems are particularly valuable as an attempt to solve the computational time issue. However, due to simplifications, most of them are unable to solve communication relay in urban environment problem in realistic environments.

2.3.3 Discussion

Several approaches were presented for both trajectory planning and positioning. Out of all approaches, convex optimisation and game theory are not suitable for urban environments as their performance is suboptimal within urban environments. Such a weak performance is caused by the complexity of predicting signal strength between two arbitrary points in urban environments. Due to the presence of buildings and effects they have on wire-

less signal strength, such a prediction is non-linear with numerous discontinuities. It is very likely that those two approaches would struggle with this complex signal strength prediction, and would get stuck in local minima.

Heuristics optimisations are the methodologies which can potentially cope with complex communication models in an urban environment. However, general heuristic-based (here also named as other heuristic-based) approaches usually work well only in the specified, narrowly-defined scenario since they explore the structure or features of a specific scenario. On the other hand, bio-inspired heuristic optimisation can work with multiple scenario types. Based on the assessment, bio-inspired heuristic optimisation optimisation is explored for both trajectory planning and positioning in this work.

2.4 Background Knowledge

2.4.1 Introduction to NMPC-Based Trajectory Planning

NMPC based trajectory planning algorithm is used in this work to determine path for the UAV. To compute path NMPC combines finite time horizon with a non-linear model of the controlled system. Time horizon determines how far prediction of vehicle behaviour is made into the future. In order to account for behaviour throughout the prediction and not only at the beginning and end of prediction, the prediction time is discretised into N horizon step, each of them of pre-specified length. At each horizon step change of control input to UAV is possible. Best possible combination of control inputs is then determined by optimisation procedure. At the end of optimisation, only control input from first horizon step is provided to actual controller, and procedure repeats for next time step.In this approach, the concept of a receding time horizon is utilized, which determines how far prediction of the UAV behaviour is made into the future as illustrated in Fig. 2.1. Optimization is performed to compute the best set of control inputs over the receding horizon is discretized into N time steps. This discretization is done because predicting all possible trajectories of the UAV using a continuous input might be computationally intensive even with fast optimization tools. Thus, it is more appropriate that a control input is approximated to a set of discrete values. Only a control input from the first horizon step is used and the optimization process repeats in the next time step to refine the trajectory planning.

2.4.2 Discrete Genetic Algorithm

Genetic algorithm is used extensively throughout this work, so a detailed description is provided. A genetic algorithm has been shown to efficiently plan a path for a UAV target tracking problem in [86] with discrete control; thus this work utilises a similar GA algorithm, version of which is more broadly described in [102]. Finding the optimal solution sequence U^i (regardless of the cost function) is done in the following way. At the

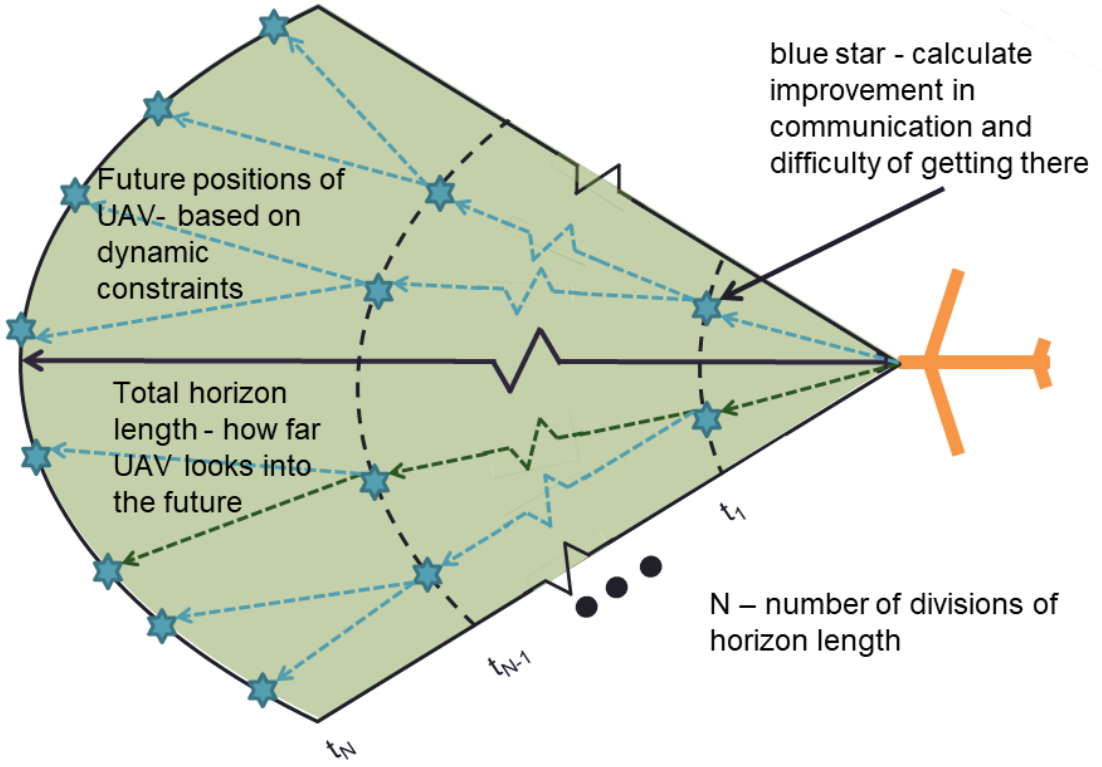


Figure 2.1: Illustration of the NMPC trajectory planner. At each star point, the UAV makes the decision on where to go next which creates a set of possible trajectories. Even thought the plan is made over the multi-step horizon length, only the first step input is taken and a new plan is made once the UAV arrives at the next point.

initialisation stage of GA, a first generation is created. It is a very important part of every GA as it largely determines the quality of problem space exploration and thus the quality of an available solution. Each chromosome consists of potential control sequence U_p^i and can be denoted as:

$$U_{p,j}^i = \begin{pmatrix} u_{\omega,0}^i & u_{\omega,1}^i & \dots & u_{\omega,N-1}^i \end{pmatrix} \quad \forall j \in 1, 2..., M \tag{2.1}$$

subject to

$$u_{\omega,k+1}^i = u_{\omega,k}^i + a\Delta u_\omega \tag{2.2}$$

where M is a population size, N is a horizon step number and a is chosen randomly from the set $\{-1, 0, 1\}$. Initial member of chromosome group g_0 is based on the control command executed in the previous step. Each subsequent command in a given sequence is determined by Eq. (2.2).

Each chromosome needs to be evaluated to determine its quality. The solution fitness f of each chromosome is based on the value of J_d defined in Eq. (5.15). A solution fitness can be found as:

$$f = \frac{1}{(J_d + J_e)^b} \tag{2.3}$$

where b is a factor determined experimentally to prevent one solution from overwhelming the optimisation process too early, and J_e is an additional cost which can be defined as:

$$J_e = \begin{cases} A, & \text{if any constraints are not satisfied is not satisfied.} \\ 0, & \text{otherwise} \end{cases} \tag{2.4}$$

where $A > 0$ is a penalty value due to constraint break.

At the reproduction stage, a new population is created while taking into account for the fitness evaluation. The process has three steps: selection, crossover and mutation.

Selection In selection, a set of breeding population is chosen from the already existing population. Selection is based on a widely used roulette wheel method. Let us define a set of chromosomes as:

$$C = \{U^i_{p,1}, U^i_{p,2}, ..., U^i_{p,M}\} \tag{2.5}$$

and a set of fitness function corresponding to each chromosome as:

$$F = \{f_1, f_2, ..., f_M\} \tag{2.6}$$

The sum of a fitness functions is then calculated as:

$$F_s = \sum_{i=1}^{M} f_i \tag{2.7}$$

Also, a cumulative fitness function as a percentage of a total cost function is defined as:

$$Fc = \{fc_1, fc_2, \cdots, fc_M\} \tag{2.8}$$

where

$$fc_j = \sum_{i=1}^{j} \frac{f_i}{F_s} \times 100; \quad \forall j \in \{1, 2, \cdots, M\} \tag{2.9}$$

Then uniformly distributed random number $\mu \sim \mathcal{U}\{0, 100\}$ is generated and compared against cumulative fitness function. Last parent for who cumulative percentage fitness is smaller than the random number is used in a breeding population. This can be formally expressed as:

$$Gm_k = \underset{U^i_{p,j} \in C}{\arg\min}(fc_j - \mu \geq 0); \quad \forall k \in \{1, 2, \cdots, M\} \tag{2.10}$$

where G_m is a chromosome from a breeding population. The process continuous until there are M parents in a breeding population. Such a selection leaves the majority of the breeding population as good parents while allowing for space exploration by keeping a few bad ones.

Crossover Crossover is used to combine two parents chromosomes together which follows the procedure outlined below.

1. Two parents chromosomes Gm_x and Gm_y are randomly selected from a breading population.

2. Crossover point g_c is randomly selected within two parents strings where $c \in \{1, 2, \cdots, N-1\}$ and crossover point is between c and $c+1$

3. Parts from parent from start to crossover point and from crossover to to the end of other parent are added added.

Mutation To help with problem space exploration, mutation is used as a final step of reproduction. Mutation is calculated separately for each member of each chromosome. As this is very short optimisation and problem space needs to be explored quickly, for the majority of optimisation, mutation is set as a chance of 50% i.e. each member of each chromosome has a 50% chance of mutation. For the final few iterations, mutation is set for a significantly lower 17% chance to allow for convergence. If a given member mutates, then its new value can be denoted as follows:

$$u^i_{\omega,k} = u^i_{\omega,k-1} + a\Delta u_\omega \tag{2.11}$$

where k is a mutating chromosome index and a is a number randomly selected from set $\{-1, 0, 1\}$; although this might lead to a situation where constraints Eqs. (5.17) and (5.18) are violated, such solutions are quickly removed due to a high cost correlated with constrain violation.

Lastly, a convergence criterion for the GA algorithm is the number of acceptable generations N_g.

2.4.3 Cross-Entropy Optimisation

Cross-entropy (CE) method was originally proposed as an adaptive variance algorithm for estimating probabilities of rare events. When applying to optimization, the basic principle of the cross-entropy optimization algorithm is to first associate it with each optimization problem a rare event estimation problem, and then to solve this estimation problem efficiently by an adaptive algorithm [103]. This algorithm constructs a random sequence of solutions which converges probabilistically to the optimal or near-optimal solution. The generic cross-entropy optimization (CEO) involves several iterative steps and can be summarized as follows:

1. Define the initial set of probability density function (pdf) following a certain distribution with parameter v_0.

2. Generate samples using the previously-defined pdf and parameters.

3. Calculate the value of the cost function associated with each sample and update the distribution parameter using selected samples.

4. Stop if the convergence criterion is reached, otherwise return to step 2.

To apply CEO to solve our problem (5.15), we first develop a new representation of it, and then describe how to achieve Step 2) and Step 3) in the above generic algorithm.

Define $\mathbf{U} \in \mathbb{R}^{N\times M}$ as a matrix of possible control sequences where each element is defined as $u_{\omega,ij}$ with $i \in \{0, 1...N-1\}$ and $j \in \{0, 1...M-1\}$, where M is the number of samples per iteration in the CEO algorithm and N is the number of control inputs. Then, let $\mathbf{\Delta}_m \in \mathbb{R}^{N\times M}$ be a matrix of possible changes of the control input from the previous step where each element is taken one of three values, i.e., $\Delta_{m,ij} \in [-\Delta u_\omega, 0, \Delta u_\omega]$. The elements in $\mathbf{U}$ should satisfy $u_{\omega,ij} = u_{\omega,i-1j} + \Delta_{ij}$ with i, j retaining previous meaning and $u_{\omega,i-1j}$ being control input executed by the UAV at the previous time step. To apply the CE algorithm to $\mathbf{\Delta}_m$, it needs to be represented in the binary form. As the matrix has only three possible options for each value it can be conveniently represented as a combination of two binary variables, so $\mathbf{\Delta}_b \in \mathbb{R}^{2N\times M}$. The transformation between the $\mathbf{\Delta}_b$ and $\mathbf{\Delta}_m$ is summarized in Table 2.1.

Table 2.1: transformation between $\mathbf{\Delta}_b$ and $\mathbf{\Delta}_m$

$\Delta_{m,ij}$	$\Delta_{b,ij}$
$-\Delta u_\omega$	00
0	01
Δu_ω	11

Each element in $\mathbf{\Delta}_b$ can be defined as a binary random variable

$$\Delta_{b,ij} = \begin{cases} 1, & \text{if } r > P_j \\ 0, & \text{otherwise} \end{cases} \tag{2.12}$$

where r is the uniform random variable between 0 and 1 and $\mathbf{P} \in \mathbb{R}^{2N\times 1}$ is a matrix of $P_1, P_2, ..., P_j$ with values between 0 and 1, and represents the parameters of the Bernoulli probability density function for generating samples. It is values of $\mathbf{P}$ which will be updated in the optimisation procedure to produce better solutions to (5.15). With this the cost function can be evaluated:

$$C_j = J(U_j) \tag{2.13}$$

where each C_j is a cost of a given control sequence U_j and is the part of the matrix $\mathbf{C} \in \mathbb{R}^{M\times 1}$ and $U_j = (u_{\omega,0j}, u_{\omega,1j}, \ldots, u_{\omega,N-1j})$ is the part of the $\mathbf{U}$ matrix. In order to prevent solutions which violate constraints of (5.17) and (5.18), a large penalty cost of C_v is added to the objective function if any of the constraints is violated.

To update values of $\mathbf{P}$, first, the objective function values of samples will be sorted in the ascending order as

$$\mathbf{C}_{sort} = \text{sort}(\mathbf{C}). \tag{2.14}$$

And position of ρ-quartile (ρ is a parameter to be tuned for the best performance) is found in

$$G = M(1-\rho) \tag{2.15}$$

where G is the position of ρ-quartile in a number of samples and M is the number of samples per iteration in the CEO algorithm. As G is not guaranteed to be an integer, interpolation needs to be used to find value of cost at position G denoted as γ_t (also referred to as threshold value), starting with finding positions of two cost values closest to G, i.e.

$$x_0 = \text{floor}(G) \tag{2.16}$$

$$x_1 = \text{ceiling}(G) \tag{2.17}$$

Then defining the actual value of the cost at positions (x_0, x_1) as:

$$y_0 = \mathbf{C}_{sort,x_0} \tag{2.18}$$

$$y_1 = \mathbf{C}_{sort,x_1} \tag{2.19}$$

where $\mathbf{C}_{sort,x_0}$ and $\mathbf{C}_{sort,x_1}$ indicates value of cost function at positions x_0, x_1 respectively. Finally interpolation is used as:

$$\gamma_t = \begin{cases} \frac{y_0(x_1-G)+y_1(G-x_0)}{x_1-x_0}, & \text{if } |x_1-x_0|>0 \\ y_0, & \text{otherwise} \end{cases} \tag{2.20}$$

With this threshold γ_t, $\mathbf{P}$ can be updated using two simple steps. First, find the set of new probabilities:

$$\mathbf{P}_{new} = \frac{\sum_{j=0}^{M} \mathbb{I}_{\mathbf{C}>=\gamma_t} \Delta_{b,ij}}{\sum_{j=0}^{M} \mathbb{I}_{\mathbf{C}>=\gamma_t}}, \tag{2.21}$$

where $\mathbb{I}$ is defined as the indicator function and $j = \{0, 1...M-1\}$. Secondly, generate new Bernoulli samples using the following parameter update:

$$P_j = \alpha P_{new,j} + (\alpha - 1)P_{j-1}, \tag{2.22}$$

where we have introduced the smoothing factor α to prevent convergence to local optimum. With the above steps, the generic cross-entropy optimization is readily applied to solve our problem (5.15).

2.4.4 Introduction to Neural Network

$P(LOS, \theta_{ij})$ in Eq. (5.8) and consequently, the signal strength of ground nodes obtained by the UAV depends on the urban environment type. When the urban type is unknown, the learning-based approach could be utilised with collected signal strength data to predict the environment type.

In this section neural networks are described. Neural network is a common technique used for classification. The origin of Neural Network can be traced as far as 1943 in [104],

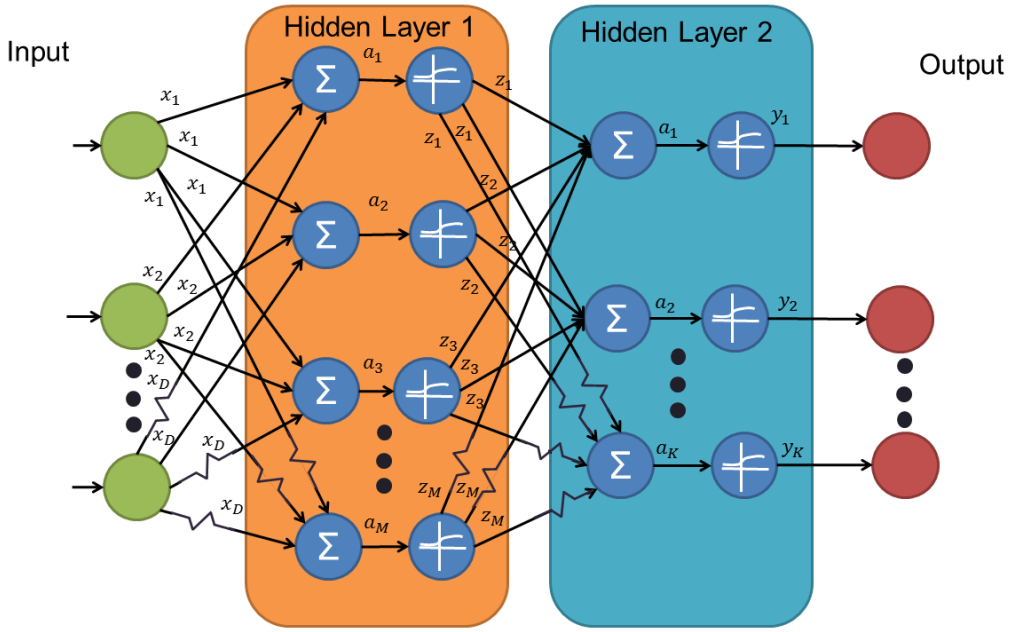


Figure 2.2: Neural Network schematics showing parts of the network

where attempt was made to model information processing in biological system. Since then it has evolved multiple times to become a useful tool for classifcation problems. Let's explore how to formulate and train such a sample neural network. Please note that formulation in this section is adapted from [105]. Simple neural network are split in three groups: input, output and two hidden units as indicated in Fig. 2.2. It is worth noting that there are several way to define layers in the neural network. Here we adopt terminology defined by [105] which is that number of hidden layers is number of adaptive layers. First from figure 2.2 the input to neural network is $(x_1, x_2..., x_D)$ and D is a number of inputs to the neural network designed by user. Ouptus from neural network are denoted as $(y_1, y_2, ..., y_K)$ and K is number of predicted classes as dictated by problem formulation.With the diagram in mind let's consider formulation of this simple neural network. We start by defning an initial combination of of input variables as:

$$a_j = \sum_{i=1}^{D} w_{ij}^{(1)} x_i + w_{j0}^{(1)} \tag{2.23}$$

With $j = 1, 2, ..., M$, M is a design parameter which defines number of neurons in a hidden layer and suberscript (1) refers to the fact that those are parameters on first layer. w_{ij} is the wieght paramters and w_{j0} are bias parameters. Finally to obtain values $(z_1, z_2, ..., z_M)$ nonlinear activation function is applied $h(\cdot)$

$$z_j = h(a_j) \tag{2.24}$$

In this work activation function used is a logistic sigmoid activation function described in more details later on in this section. The values from formula 2.26 are once again combined

with weights by:

$$a_k = \sum_{j=1}^{M} w_{ik}^{(1)} z_j + w_{k0}^{(1)} \tag{2.25}$$

Finally for normalisation we apply logistic sigmoid function to this layer as:

$$y_k = \sigma(a_k) \tag{2.26}$$

Note that for the clarity distinction and generality between two layers $\sigma(\cdot)$ and $h(\cdot)$ are defined separately. Generally those two functions could be different functions, only in this work they are defined as the same. With this definition full formulation can be represented as:

$$y_k(\mathbf{x}, \mathbf{w}) = \sigma(\sum_{j=1}^{M} w_{ik}^{(1)} (h(\sum_{i=1}^{D} w_{ij}^{(1)} x_i + w_{j0}^{(}1))) + w_{k0}^{(}1)) \tag{2.27}$$

Where $\mathbf{x}$ are grouped inputs and $\mathbf{w}$ are grouped weights and bias paramters This equation is the most general form of Neural Network which can be further modified and simplified, but that is beyond scope of this overview. To train the neural network $\mathbf{w}$ needs to be defined such that it fits training data the best. Let's explore how learning in neural network is performed. In this work we employ scaled conjugate gradient backpropagation with cross entropy error function. According to [105] cross entropy error function is a standard choice of error function for classifcation problem and is defined as:

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_k(\mathbf{x}_n, \mathbf{w}) \tag{2.28}$$

With error function defined any optimiser could be used, here scaled conjugate gradient backpropagation method is used as defined by [106]. This is one the most sucesful and widely used optimisation method for usage in Neural Network.

To complete this overview of NN it is necessary to described logistic sigmoid function. Logistic sigmoid function is defined as:

$$\sigma(a) = \frac{1}{1 + \exp -a} \tag{2.29}$$

Sigmoid functions are used to normalise the output from given neuron between zero and one and is depicted in figure 2.3. In other words regardless of the value of input value, the output of this function is always between 0 and 1. In neural network sigmoid function is necessary to ensure each neuron output is similar. Without that some neurons would overhelm the prediction very quickly, resulting in false predictions.

2.5 Summary

Establishing wireless communication networks are vital in various circumstances such as rescue missions in a natural disaster. The literature review has shown that utilising wireless

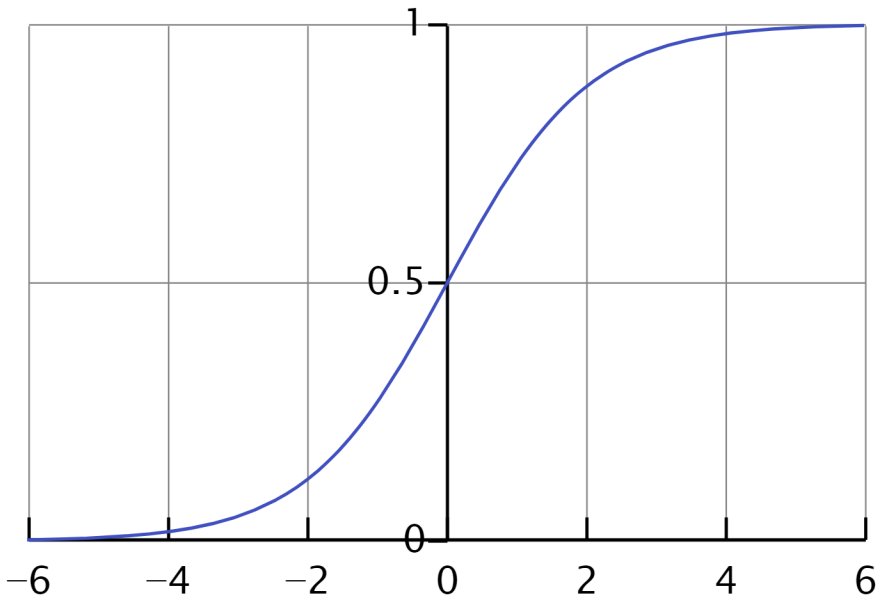


Figure 2.3: Sigmoid function

communication in an urban environment is challenging due to obstructions and occlusions by buildings, resulting in significant delays and limited range of bandwidth. The following technologies have been used as an attempt to mitigate wireless communication issues in urban environments: infrastructures (e.g. cell tower), ground-based radios and satellite communications. The first is often unavailable due to the damage caused by the natural disaster. Moreover, in an emergency, there is little time to put a temporary infrastructure in place. The other two do not rely on existing infrastructure, but they cannot solve the issue of obstruction by buildings entirely. Additionally, satellites are limited by the nature of their pre-planned orbits.

Compared with the traditional method, utilising UAVs to construct communication relay can be a promising alternative solution. The UAVs relays should be a group of low flying (below 500 meters), and small UAVs and they have advantages in rapid deployment to the mission area and swift relocation to encounter the effect of obstruction by buildings.

To allow usage of limited UAVs resources, it is necessary to consider how to optimise the UAV assets utilisation through choosing the optimal number of UAVs to cover a given area and planning optimal trajectories. In this context, trajectory planning and positioning is an important aspect to consider. Trajectory planning and positioning are used to determine which position UAVs should be at a given time to provide the best improvement to communication to a group of ground nodes.

Trajectory planning and positioning has challenges from the available computational power and complex wireless signal prediction. Computational power becomes a significant issue when pre-planning cannot be performed, or plan changes during a mission. In those situations, trajectory and position need to be computed relatively quickly to guarantee

minimum disruption to the wireless network. Facing constraint from the computational power, it is better to reach a sub-optional answer timely, rather than an optimal answer too late. Bio-inspired heuristic based optimisers are the only approach which can consistently cope with complex wireless signal prediction in an urban environment. Also, it should be adopted for both trajectory planning and positioning problem. As outlined in the literature review, bio-inspired heuristic based optimisers struggle with computational time. To mitigate it, careful formulation of the cost function and a limited number of optimisation steps are introduced. For trajectory planning, the receding horizon approach is also utilised to mitigate the computational time problem.

A critical part of trajectory planning and positioning is to determine the positions of best communication improvement. To achieve that, the capability of predicting communication strength between arbitrary points on the ground and in the air is essential.

Such a prediction in an urban environment is challenging because of the absorption, reflection and refraction effects that buildings have on the wireless signal. Researchers have been trying to cope with the challenge through a model-based approach and measurement-based approach. Model-based methods rely on the mathematical model to predict wireless signal, while measurement-based methods rely on collecting signals from the ground nodes to update or develop a communication model during the mission time.

The main limitation of model-based approaches is that they need to know a large number of parameters required to compute the communication quality between two nodes, which can be challenging to estimate before the mission or online. Measurement-based approaches were devised to cope with the challenge of estimating communication parameters accurately, but they were used mainly for stationary environments because the amount of data required for learning and prediction for mobile ground nodes was too high. To enable the deployment of UAV relay in an urban environment, it is essential to relax the prediction constraints, reducing information required and improving predicting efficiency.

Chapter 3

Model-based Approach to Known Urban Environment

In this chapter, it is assumed that communication can be predicted between arbitrary air and ground points using model only. In this model-based approach, signal strength can be predicted using a distance between two nodes and detailed knowledge about the urban environment. The model-based approach offers several benefits. First of all, provided that the detailed map of an urban environment is known, it guarantees close to optimal relay performance. Moreover, it can start relay mission immediately upon arrival to the mission area. Finally, model-based approaches have certain guaranteed connection strength predictions in a specific environment.

Using this model provides excellent flexibility in terms of using either trajectory planning or positioning as well as the mobility of ground vehicles. With this in mind, two options were explored: positioning of rotary wing UAV with a group of stationary ground nodes and trajectory planning of fixed-wing unmanned aerial vehicles (UAVs) with a group of mobile ground nodes. Those two approaches allow exploiting unique characteristics of each vehicle type. Namely, with stationary ground nodes, it is relatively easy to define optimal point for a UAV to hover, making rotary wing aircraft a natural fit to this kind of mission. However, fixed-wing aircraft is much more suited to mobile nodes scenarios where it needs to move most of the time.

The algorithms proposed in this chapter are designed to try to satisfy three conditions i) maximise communication performance with consideration of signal quality reduction due to obstacles (i.e. LOS obstruction), ii) obey dynamic constraints of UAVs (applicable to fixed-wing only) and iii) run as fast as possible for the online implementation. To fulfil the outlined requirements, a particle swarm optimisation (PSO) technique for positioning and non-linear model predictive control (NMPC) based trajectory planner [7] are used. PSO is particularly suitable to solve the problem in an urban environment as it can address with discontinuities and non-linearities in the communication strength model introduced by buildings. Near-real-time implementation is achieved by carefully defining

the cost function to be as efficient as possible and sacrificing optimality when necessary by restricting a number of iterations of PSO. The NMPC framework was chosen since the kinematic constraints of the UAV and communication can be easily combined in a single formulation, while being adjustable to allow the real-time implementation. Besides, as the NMPC updates the trajectory periodically, the trajectory can be adjusted quickly in case of unpredicted events such as path change by ground nodes. Finally, to speed up computation, the NMPC algorithm is simplified by assuming a constant speed and discrete heading change of the UAV.

To model the communication performance, this chapter follows a model-based approach from [8]. The model is combined with three different communication performance metrics: the global message connectivity (GMC), the worst case connectivity (WCC) and the modified global message connectivity (mGMC). The three performance metrics are used during optimisation to quantify the performance of the solution and to focus on specific connections within the group. The GMC is used to improve the sum of all connections amongst ground nodes the WCC focuses the poorest/weakest link only. Meanwhile, the proposed mGMC uses a certain number of weak connections based on the desired communication quality requirement to combine the benefit of the GMC and the WCC.

3.1 Scenarios and Algorithm Overview

Figure 3.1 shows a sample scenario considered for trajectory planning in this study. The yellow circles represent ground nodes/vehicles which are assumed to be performing their own individual missions, black dotted lines represent the trace of ground vehicles, aircraft models represent UAVs and blue cuboids represent buildings. The path and positions of ground vehicles are not controlled by the communication optimisation algorithm. They are in the postions and trajectories necessitated by their own mission requriements. This assumption is justified by considering mission types performed by rescue worker. Rescue worker maybe transporting a victim to the nearest hospital, where every minute counts. Moreover he could be unable to move in an emergency situations. Those two example shows that it is desirable to allow ground nodes to perform their mission to their needs. For postionining of a rotary wing aircraft with stationary ground nodes, the scenario is the same other than, positions of ground nodes are fixed.

Assumptions made in this work are listed as:

(i) The urban environment is known prior to the mission;

(ii) The UAVs can obtain the current position of the ground vehicles via communication; and

(iii) If a line-of-sight (LOS) is obstructed by buildings between two nodes, a communication quality is reduced proportionally to the length of the obstruction.

(iv) UAVs can communicate amongst themselves

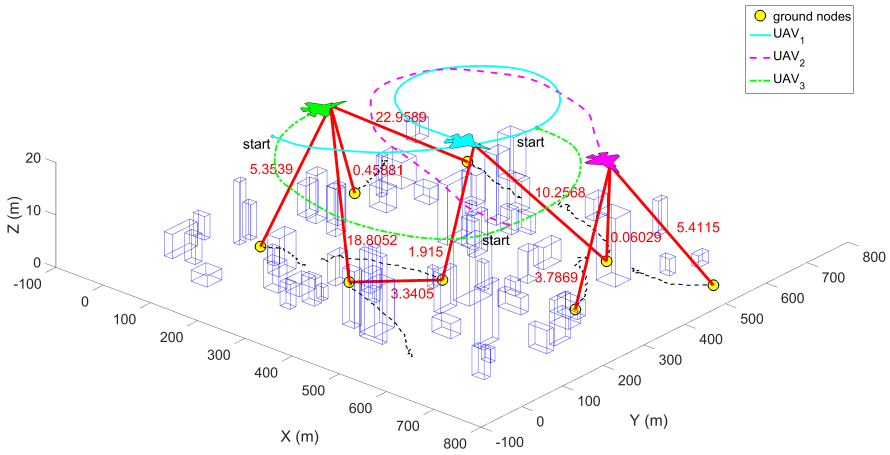


Figure 3.1: A sample scenario of the proposed approach.

The flow chart of the proposed trajectory planning algorithm for communication relay UAVs in an urban environment is shown in Fig. 3.2. Firstly, each UAV estimates states and also predicts the future position of ground vehicles for a certain time steps ahead with the Kalman filter and the GPS position of ground vehicles as sensor measurements. Based on position estimates, UAVs then calculate the communication cost in consideration of the communication range, signal-to-noise ratio and line-of-sight obstruction. The trajectory and the minimum spanning tree (MST) for a given UAV is found using a discrete genetic algorithm (GA)-based optimisation with the desired communication performance metric. Algorithm for positoning differes in three places. First Ground vehicles do not move. Second there is no need for ground vehicle estimation as their current and future postions are perfectly known (i.e. they do not change with scenario time). Finally instead of Discrete GA optimisation, particle swarm optimiser was used.

3.2 Communication Model

Communication model formulation consist of three parts. First part is accounting for reduction of communication due to distance. Secondly we convert it to cost sutiable for optimisation and we add extra reduction in communication performance based on how much obstruction by building occurs and extra reduction in communication performance if UAV gets too low. Thirdly Formulation is completed by combining the costs of communication between single pair of air and ground nodes into a group suitable for optimisation.

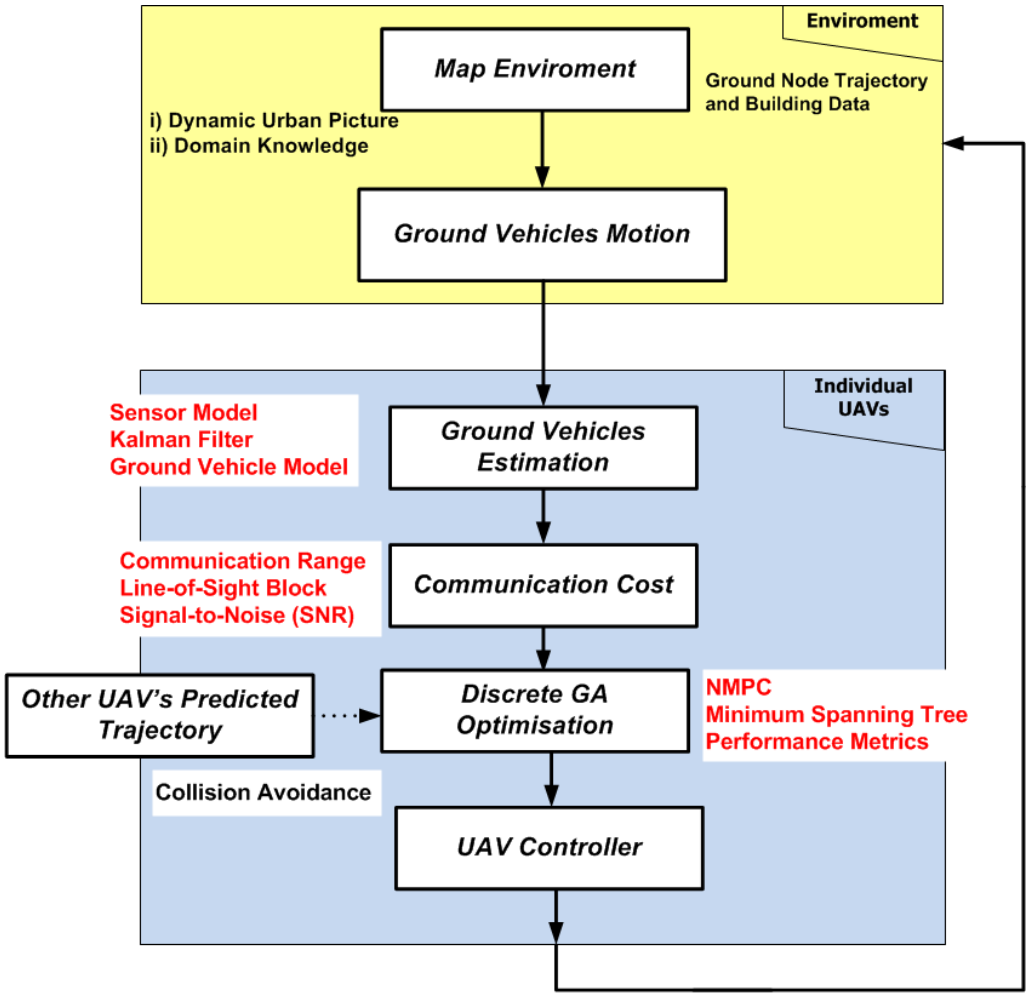


Figure 3.2: Flow chart of the NMPC-based trajectory planning algorithm for communication relay UAV.

3.2.1 Line-of-sight Communication Probability

Finding an optimal position requires establishing a node-to-node communication quality. In this work, a communication quality is represented as a probability of successful communication using an open space communication model [7, 8]. In this model, node i has the transmission power of P_i and observes a noise σ_i^2 while receiving a signal from other members of the group. Then, the received signal-to-noise ratio (SNR) Γ_{ij} for a signal transmitted from the i-th node and received by j-th node is given as:

$$\Gamma_{ij} = \frac{P_i G_{ij}}{\sigma_i^2} \tag{3.1}$$

where G_{ij} is a channel gain and can be expressed as:

$$G_{ij} = \frac{C_{ij}|h_{ij}|^2}{D_{ij}^{\alpha}} \tag{3.2}$$

where C_{ij} is a constant accounting for antenna gains and shadowing, h_{ij} is responsible for multipath fading, α is a path loss factor and D_{ij} is the distance between two nodes.

For the computation of a probability of successful communication, a minimal acceptable link quality γ needs to be determined. If Rayleigh fading is assumed i.e. h_{ij} is the complex Gaussian with zero mean and unit variance, then the probability of successful transmission from i-th node to j-th node is equal to:

$$P_r{}^{ij}\left(\Gamma_{ij} \geq \gamma\right) = \exp\left(-\frac{\sigma_j^2 \gamma D_{ij}^{\alpha}}{C_{ij} P_i}\right) \tag{3.3}$$

It is worthwhile noting that, at this time, the model does not consider effects like diffractions and reflections of wireless signal, as this would significantly increase the computational load.

3.2.2 Communication Cost

For optimisation purposes, a probability calculated in Eq. (3.3) is transformed into a cost. Then, effects of LOS obstructions are added. The weighted edge cost between two nodes (i.e. i and j) is defined by using the above probability of successful transmission as:

$$W_{ij}^{o} = -\ln\left(P_r^{ij}\right) \tag{3.4}$$

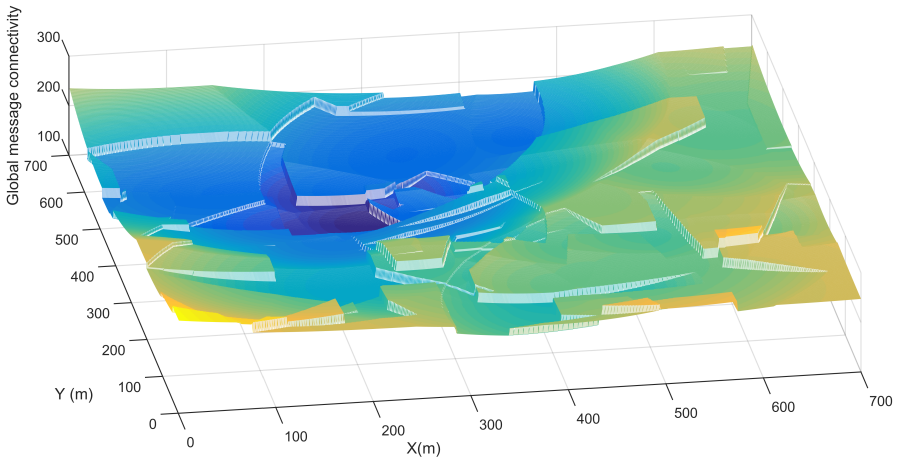
Above equation implies that the higher the probability of successful communication is, the lower the cost is, and this weight will be used as a cost to be minimised. In order to consider buildings, an additional weight is added whenever a connection passes through the building. This weight has a relatively high value to strongly discourage connections through buildings and is represented as:

$$W_{ij}^{b} = \begin{cases} W_{ij}^{pb} & \text{if the LOS is blocked} \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

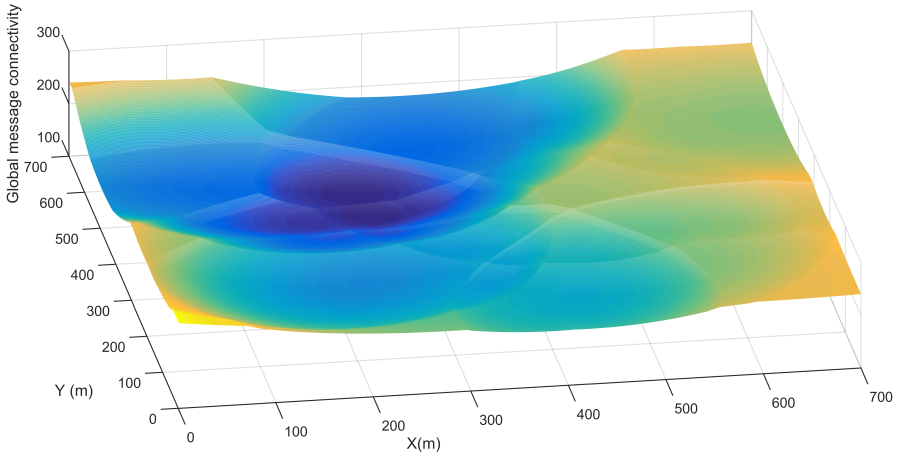
where W_{ij}^{pb} is a penalty cost due to LOS block. The computation of LOS block is explained in the next section. Note that, non-smooth characteristic of W_{ij}^{pb} can make the solution space highly non-convex with many local minima, as illustrated in Fig. 3.3a. This figure shows a certain communication performance metric when using a single relay UAV at different grid locations at a constant altitude (different communication performance metrics will be explained in Section 3.2.3). To mitigate the issue, a new weight is introduced by using the length of LOS block l_b inside the building (i.e. the length of the intersection/overlap between the LOS and the obstructing building as illustrated in Fig. 3.4) as:

$$W_{ij}^{b} = \begin{cases} W_{ij}^{pb} l_b & \text{if the LOS is blocked} \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

where l_b is the distance of LOS intersection with an obstructing building and W_{ij}^{pb} is a base cost. Figure 3.3b clearly shows a much smoother cost function, which facilitates an



(a) Non-smooth weight due to discrete LOS obstruction.



(b) New weight considering the length of overlap

Figure 3.3: Surface plot for the communication performance function at different relay UAV locations.

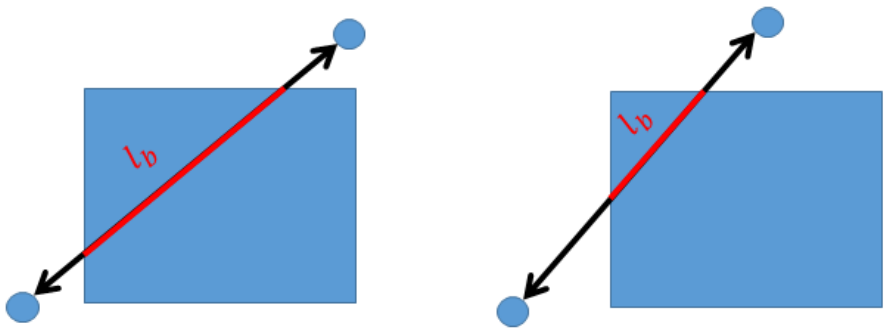


Figure 3.4: Illustration of new weight computation. The case on the left would result in a higher cost than that on the right since the longer portion of the LOS line overlaps with the building.

optimisation process. Matlab library written in IRNA [107] is used to find the intersection and the distance l_b between the LOS line and the building and is ilustrated in Fig. 3.4.

For static scenarios, the weight for the UAV height restriction is also added. As in Eq. (3.3), a communication quality is better if nodes are close to each other, implying it is likely that UAVs' optimal position would be dangerously close to the ground and buildings in order to maximise the communication quality. To prevent this from happening, a minimum allowable UAV height is imposed as:

$$W_{ij}^{h} = \begin{cases} W_{ij}^{ph}, & \text{if below minimum height} \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

where W_{ij}^{ph} is a penalty cost due to the height restriction. It is worthwhile noting that this weight is not related to the communication in a strict sense; it only serves as a barrier to prevent UAVs from flying too low.

Finally, the total communication weight, W_{ij}^{t}, can then be expressed as:

$$W_{ij}^{t} = W_{ij}^{o} + W_{ij}^{b} + W_{ij}^{h}. \tag{3.8}$$

As the number of nodes involved in a scenario increases, the number of connections in the network increases significantly. As a result, efficient sharing of information becomes a problem. To deal with this issue, this work uses a minimum spanning tree (MST) concept. The MST is defined as a subset of graph where all nodes are connected to each other but there are no loops, having a minimum (or at least the same as minimum, as there can be several minimum spanning trees in a single scenario) sum of edge weights [108]. In simple terms, the MST finds the least costly and least number of connections to connect all the members in the group.

3.2.3 Communication Performance Metrics

In the team of networked nodes, optimal UAV position or trajectory will be different depending on performance metric (index) used in optimisation. This work considers three communication metrics: the global message connectivity (GMC), the worst case connectivity (WCC), and the modified global message connectivity (mGMC).

Global message connectivity (GMC) The GMC is defined as a probability of message being successfully transmitted to all nodes within the minimum spanning tree (MST) [8]. As all positions of ground nodes, UAVs and buildings are known, Eq. (3.8) can be used to find node to node communication cost. With those individual link weight, the MST can be constructed. Let the Adjacency matrix of the MST be represented as $A' \in \mathbb{R}^{(n+m)\times(n+m)}$, where n is a UAV number and m ground vehicle number, then $A'_{ij} = 1$ if the link from node i to node j is the part of the MST, and $A'_{ij} = 0$, otherwise. In the MST, a probability of message being successfully transmitted to all nodes is a sum of all connections within

the MST, thus the performance index can be denoted as:

$$J_{GMC}(\bar{\mathbf{x}}^{pos}, \bar{\mathbf{x}}^{g,pos}) = \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} A'_{ij} W^t_{ij}. \tag{3.9}$$

Note that W^t and consequently A' depend on 3-D location of UAVs (represented as $\bar{\mathbf{x}}^{pos} \in \mathbb{R}^{3\times n}$) and ground nodes (represented as $\bar{\mathbf{x}}^{g,pos} \in \mathbb{R}^{3\times m}$). thus by minimising the GMC performance index ($J_{GMC}(\bar{\mathbf{x}}^{pos}, \bar{\mathbf{x}}^{g,pos})$) with respect to UAV position, the communication relay optimal position can be found as:

$$\min_{\bar{\mathbf{x}}^{pos}} J_{GMC}(\bar{\mathbf{x}}^{pos}, \bar{\mathbf{x}}^{g,pos}) = \min_{\bar{\mathbf{x}}^{pos}} \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} A'_{ij} W^t_{ij}. \tag{3.10}$$

A simple illustration of using the GMC as a performance index is shown in Fig. 3.5. This metric improves global communication quality rather than focusing only on a particular link(s).

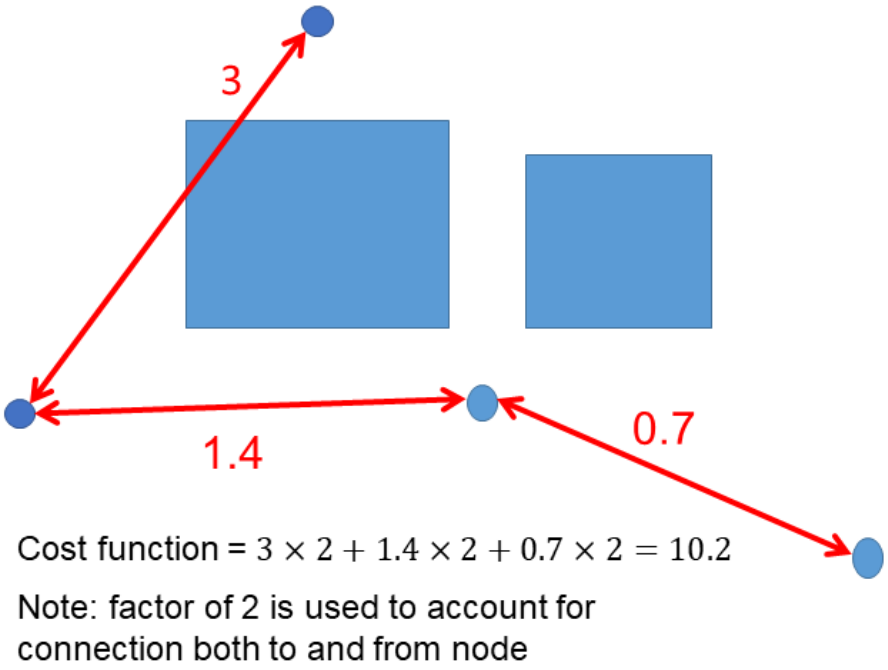


Figure 3.5: Cost function with the GMC explained on a simple example. The blue square is a building, blue circles are nodes and red lines is the MST.

Worst case connectivity (WCC) The WCC is defined as the link with the lowest probability of successful communication of all the links within the MST [8]. Using the MST definition from previous paragraph, UAV relay position can be found by minimising the weight of worst link within the MST with respect to UAV locations $\bar{\mathbf{x}}^{pos}$ as:

$$\min_{\bar{\mathbf{x}}^{pos}} J_{WCC}(\bar{\mathbf{x}}^{pos}, \bar{\mathbf{x}}^{g,pos}) = \min_{\bar{\mathbf{x}}^{pos}} \left(\max_{\forall i,j \in \mathbb{M}} (W^t_{ij} A'_{ij}) \right) \tag{3.11}$$

where $\mathbb{M} = \{1, \cdots, n+m\}$ is a set of all nodes in the network (i.e. m UAVs and n ground nodes). This performance metric is a worst connection within the MST as shown in fig. 3.6, and UAV is trying to improve that connection. This index performs exceptionally in case of numerous disconnected group of ground nodes.

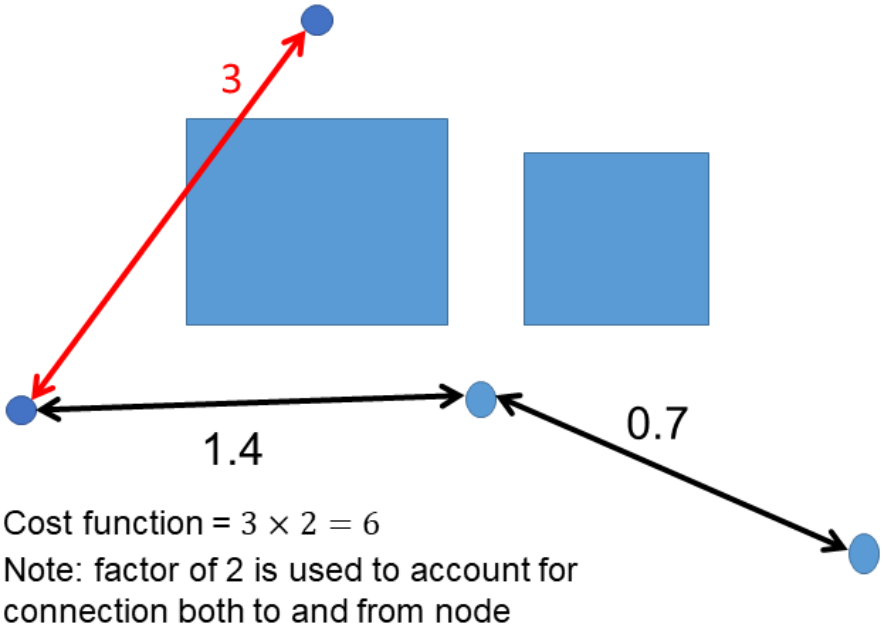


Figure 3.6: Cost function with the WCC explained on a simple example.

Modified global message connectivity (mGMC) The aforementioned WCC and GMC focus on improving one worst connection and overall communication performance respectively. If a need arises to improve both overall and node to node communication quality, neither of the metric is suitable. To this end, the mGMC metric is proposed as a compromise between the GMC and the WCC. For the mGMC performance index, constant β is defined as the number of weak connections in the initial MST to be improved. The constant can be obtained by considering user defined minimum communication probability P_d of the network.

Firstly, desired probability needs to be converted into weight as:

$$W^d = -\ln(P_d) \tag{3.12}$$

then W^t_{ij} for all $i, j \in \mathbb{M}$ is compared against W^d to find connections weaker than the desired probability in the following manner:

$$W^{tc}_{ij} = \begin{cases} 1, & \text{if } W^t_{ij} A'_{ij} > W^d \\ 0, & \text{otherwise} \end{cases} \tag{3.13}$$

where $W^{tc} \in \mathbb{R}^{(n+m)\times(n+m)}$ is a temporary matrix. Lastly, constant β is calculated as:

$$\beta = \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} W^{tc}_{ij} \tag{3.14}$$

β is then used to compute the mGMC performance index, starting with rearranging the connectivity matrix into a vector:

$$W^{ts} = vec(W^t A^{'}) \tag{3.15}$$

where $W^{ts} \in \mathbb{R}^{(n+m)^2 \times 1}$. This vector is sorted in a descending order and stored as W^{ts}. The mGMC cost can then be found as:

$$\min_{\bar{\mathbf{x}}^{pos}} J_{mGMC}(\bar{\mathbf{x}}^{pos}, \bar{\mathbf{x}}^{g,pos}) = \min_{\bar{\mathbf{x}}^{pos}} \sum_{k=1}^{\beta} W_k^{ts} \tag{3.16}$$

The mGMC focuses neither on all connection within group nor the worst connections, but rather on improving number of connection decided by user requirements. Figure. 3.7 shows the computation using the mGMC metric on a simple example. For this case, as β is 4 (considering bi-directional links) and $W^d = 1$, the total mGMC metric cost is equal to $(3.0 + 1.4) \times 2 = 8.8$.

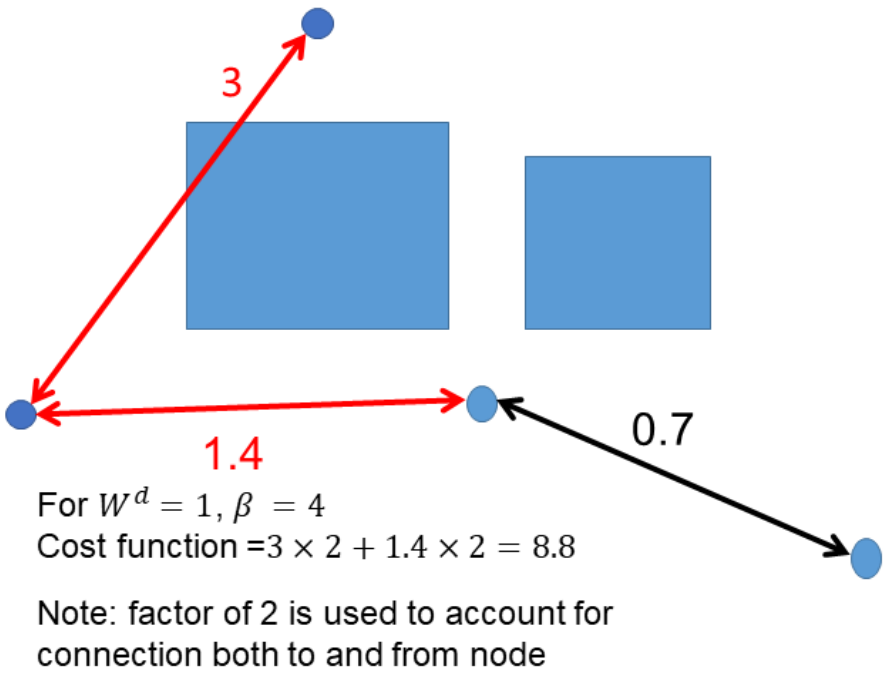


Figure 3.7: Cost function with the mGMC explained on a simple example.

3.3 Relay UAV Optimal Positioning and Trajectory Planning

Distinguishing and investigating the advantages of each performance metrics introduced from the previous Section can be performed easily on a static scenario as removing time dependence reduces the number of variables affecting the performance of each metric. Thus, static scenarios are considered as well. As the nature of the dynamic and static scenarios

is different (i.e. the one requires trajectory planning with consideration of UAV dynamics while the other is simple positioning), each algorithm follows a different procedure to find an optimal solution.

3.3.1 Relay UAV Positioning for Optimal Configuration

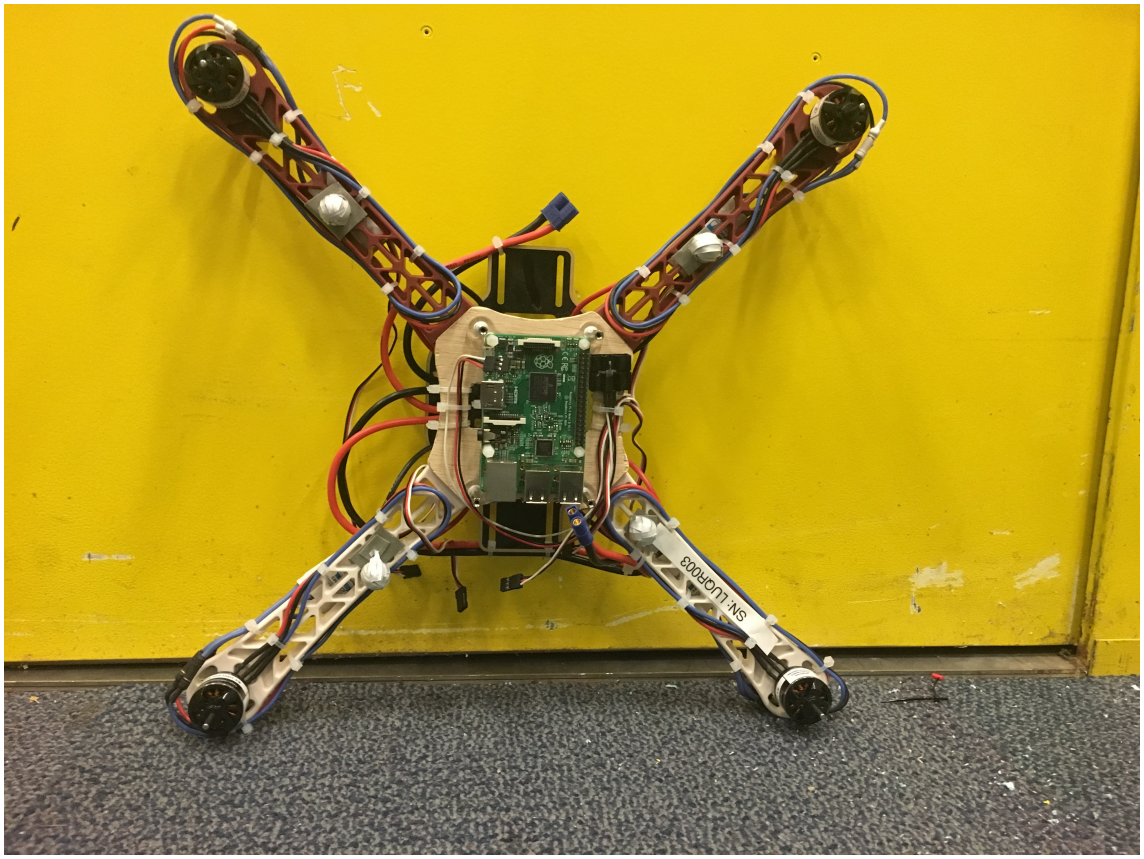


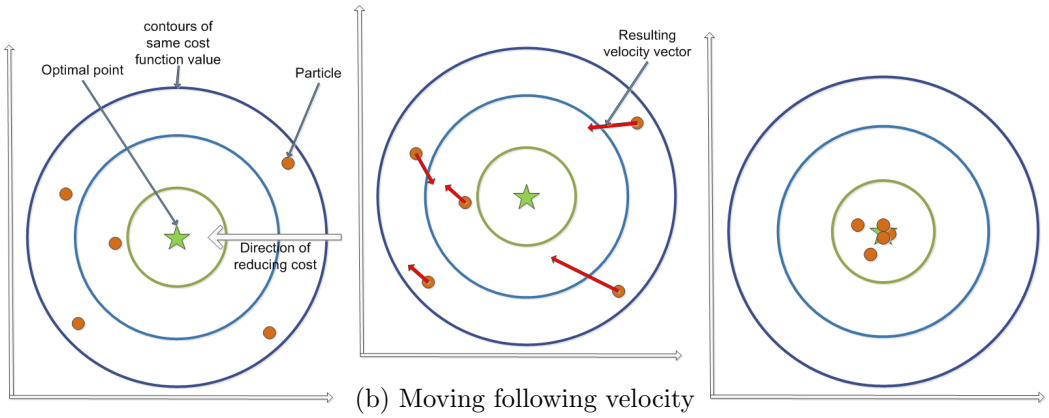
Figure 3.8: Sample quadcopter UAV which could be used with this approach.

For positioning we propose usage of multirotor UAV such as one in Fig. 3.8. Such a vehicle has a few advantages over fixed wing aircraft. The main one being low skills ceilling requred. Wtih this type of vehicle minimal skill and supervision is required from the operator. While the proposed algorithm can work with any size of the vehicles, in here we focus on small class of UAVs (below 7 kg).

Firstly, in order to generate relay UAV optimal position for static scenarios using aforementioned performance metrics, the particle swarm optimisation (PSO) algorithm [109] is utilised. The PSO was used due to its ability to solve complex, non-smooth and non-convex problems. The PSO algorithm start by randomly spreading particles over the problem space and evaluating their cost function. The consecutive positions is found by computing velocity vector in each dimension of the problem as:

$$v_{i,d} = r_1 \omega v_{i-1,d} + \varepsilon r_2 \left(X_{ib,i-1,d} - X_{c,i-1,d}\right) + \varepsilon r_3 \left(X_{gb,i-1,d} - X_{c,i-1,d}\right) \tag{3.17}$$

where d denotes a problem space dimension, i represents the iteration step, ω represents



(a) Initial random positions (b) Moving following velocity vector (c) Final particles positions

Figure 3.9: Illustration of the particle swarm optimisation process.

the inertia of the particle, and ε is a correction factor. X_{ib}, X_{gb} and X_c represent the best position of a given particle, the global best position (considering all particles) and the current position of the particle, respectively. $v_{i,d}$ is the particle velocity, and r_1, r_2, r_3 are random numbers from zero to one. The calculation finish once convergence criterion (e.g. the maximum number of iterations) is satisfied. The PSO algorithm is illustrated in Fig. 3.9

3.3.2 UAV Trajectory Planning to Search for Optimal Solutions

In this section, trajectory planning for the fixed-wing communication relay UAV is discussed. The UAV trajectory planning problem in an urban environment presents several challenges. First of all, the fixed-wing type UAV (Fig. 3.10) considered in this paper is underactuated where the number of available control inputs is smaller than the number of controlled outputs. Besides, its control input (e.g. turning rates or speed command) is constrained and there is a nonholonomic (differential) state-vector constraint. These make it difficult for the fixed UAV to follow given arbitrary trajectories in 3-D space precisely. Moreover, due to the presence of buildings, the communication quality between nodes can change rapidly. For the implementation, the planning algorithm should be run in real time. Nonlinear model predictive control (NMPC)-based online trajectory planner has all of these features. UAV dynamic constraints can be added as optimisation constraints to be considered in the NMPC problem formulation. The algorithm can tackle rapid change of the communication quality. As the NMPC framework considers the future path of ground vehicles, the trajectory can also prevent the LOS obstruction from occurring in advance.

UAV Kinematic Model

This work utilises a simple two-dimensional UAV kinematic model [110] by assuming the UAV speed is constant. This simplification was made to: i) reduce optimisation time and

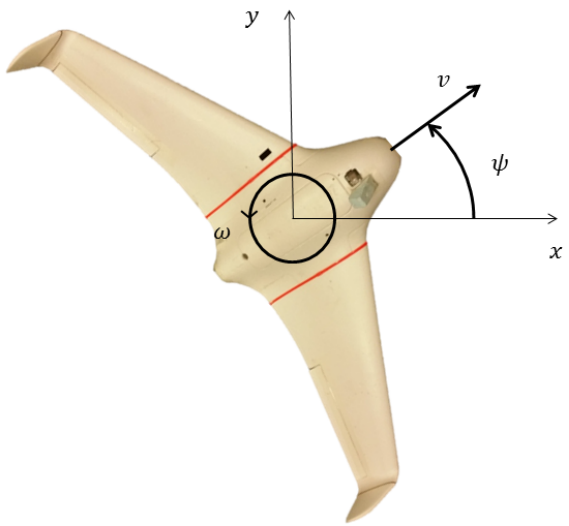


Figure 3.10: Example UAV with marked reference frame, speed, yaw rate and heading

ii) save the energy used by the UAV during its flight, where the UAV model is given as:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{\omega} \end{pmatrix} = f(\mathbf{x}, u_\omega) = \begin{pmatrix} v\cos\psi \\ v\sin\psi \\ \omega \\ -\frac{1}{\tau_\omega}\omega + \frac{1}{\tau_\omega}u_\omega \end{pmatrix} \tag{3.18}$$

where $\mathbf{x} = \begin{pmatrix} x & y & \psi & \omega \end{pmatrix}^T$ are the inertial position, heading and yaw rate of the UAV, respectively. τ_ω is time constant accounting for actuator response delay, which can be determined experimentally for given UAV model. u_ω is a command input in form of turning rate. In this work, the command is constrained by:

$$|u_\omega| \leq \omega_{max} \tag{3.19}$$

$$|u_{\omega,k} - u_{\omega,k-1}| = 0.1 \tag{3.20}$$

where k is a current time step. The first constraint limits the maximum heading rate of the vehicle. The second constraint limits the rate at which heading changes and allows for discretisation of the controller. Both constraints are the result of dynamic limits of a fixed-wing UAV. The UAV model from Eq. (5.13) is discretised using Euler integration as:

$$\mathbf{x}_{k+1} = f_d(\mathbf{x}_{\omega,k}, u_k) = \mathbf{x}_k + T_s f(\mathbf{x}_k, u_{\omega k}) \tag{3.21}$$

where $\mathbf{x}_k = \begin{pmatrix} x_k & y_k & \psi_k & \omega_k \end{pmatrix}^T$ and T_s is a sampling time.

Tracking Filter with Ground Vehicle and Sensor Models

While it was assumed that UAV knows current position of ground vehicle, it does not know the trajectory given ground node will take in the future. As a matter of fact ground node

may not know the trajectory it will take in the future. Besides communicating the planned trajectory would result in additional work overhead required by the ground node. From earlier use cases example, such a solution is unacceptable as it would result in increased load for ground nodes.

For the UAV to predict future states of ground vehicles (position, velocity and acceleration) which is required for NMPC-based trajectory planning, the Kalman filter (KF) is used. Generally, for a discrete KF, two steps are required using vehicle and sensor models: i) prediction step where vehicle states and error covariance are extrapolated and ii) update step where a correction is made.

The ground vehicle model is based on the work from [111] where a discrete state of the ground vehicle $\mathbf{x}_k^g = \begin{pmatrix} x_k^g & \dot{x}_k^g & \ddot{x}_k^g & y_k^g & \dot{y}_k^g & \ddot{y}_k^g \end{pmatrix}^T$ can be found using:

$$x_{k+1}^g = F_k x_k^g + \eta_k \tag{3.22}$$

where η_k is a process noise which represents the acceleration characteristics of the target, and F_k is a state transition matrix. The details of this model can be found in [111].

Provided that the ground vehicles are using GPS, their GPS positions are used as measurements for the UAV (via communication) for the position estimation as:

$$\mathbf{z}_k = H_k \mathbf{x}_k^g + \mathbf{v}_k \tag{3.23}$$

where the measurement matrix is:

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.24}$$

The measurement matrix implies that x and y positions of the ground nodes are the only measured variables by the GPS. The measurement noise is $\mathbf{v}_k \sim N(0, R_k)$ and the covariance matrix is given as:

$$R_k = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{3.25}$$

where σ_x and σ_y are the standard deviations of positions of x and y, respectively.

Performance Index

In a dynamic scenario, using performance metrics from Section 3.2.3 on its own is not sufficient as they do not account for dynamic properties of UAVs. Thus, the NMPC framework is combined with the aforementioned performance metric to create a trajectory plan for the UAV in the from of control input sequence $U^i = \left(u_{\omega,0}^i, u_{\omega,1}^i, \ldots, u_{\omega,N-1}^i \right)$ where i is i-th UAV and N is a horizon step. To find this control sequence, a new performance index is defined in Eq. (5.15) and minimised using a genetic algorithm (GA):

$$J_d = \Phi(\bar{x}_N, \bar{x}_N^g) + \sum_{k=0}^{N-1} L(\bar{x}_k, \bar{x}_k^g, u_{\omega,k}^i) \tag{3.26}$$

s.t.

$$\mathbf{x}^i_{k+1} = f_d(\mathbf{x}^i_k, u^i_{\omega,k}) \tag{3.27}$$

$$\omega_{min} \leq u^i_{\omega,k} \leq \omega_{max} \tag{3.28}$$

$$|u^i_{\omega,k} - u^i_{\omega,k-1}| = \Delta u_\omega \tag{3.29}$$

$$|C(x^i_k - (x^{j\neq i}_k)| > r_c \tag{3.30}$$

where:

$$\Phi(\bar{\mathbf{x}}_N, \bar{\mathbf{x}}^g_N) = p_c J_z(\bar{\mathbf{x}}^{pos}_N, \bar{\mathbf{x}}^{g,pos}_N) \tag{3.31}$$

$$L(\bar{\mathbf{x}}_N, \bar{\mathbf{x}}^g_N, u^i_{\omega,k}) = \frac{1}{2}\left[q_c J_{mGMC}(\bar{\mathbf{x}}^{pos}_k, \bar{\mathbf{x}}^{g,pos}_k) + r_v\left(\frac{u^i_{\omega,k}}{\omega_{max}}\right)\right]^2 \tag{3.32}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.33}$$

where J_{mGMC} represents the mGMC performance index and $\bar{\mathbf{x}}^{pos}_k$ and $\bar{\mathbf{x}}^{g,pos}_k$ are x and y position of UAVs and ground vehicles, respectively, from state vectors $\mathbf{x}_k = \begin{pmatrix} x & y & \psi & \omega \end{pmatrix}^T$ and $\mathbf{x}^v_k = \begin{pmatrix} x^v_k & \dot{x}^v_k & \ddot{x}^v_k & x^v_k & \dot{y}^v_k & \ddot{y}^v_k \end{pmatrix}^T$. In the objective function of $J(U_\omega)$ of (5.15), the first term $\Phi(\bar{x}_N, \bar{x}^g_N)$ is to consider the communication performance at the final receding horizon step. We separate the communication performance at the final step from the rest, as it is more important than intermediate steps and we assign a relatively higher weight for it. J_{comm} represents the average communication data rate $\sum_{i=1}^{n} R_i/n$ between the UAV and n ground nodes where R_i represents the link quality defined (4.8). The second term $\sum_{k=0}^{N-1} L(\bar{x}_k, \bar{x}^g_k, u_{\omega,k})$ consists of two parts as in (3.32). The first part is the sum of communication costs for all receding horizon steps except for the final step. The second part is a penalty for the UAV to make a turn; this is included to minimize the energy usage whenever possible under the assumption that turning consumes significantly higher energy than the straight flight. It is worthwhile mentioning that there exist other metrics such as the worst link between the UAV and ground nodes [8]. This would mean that the UAV focuses on improving the weakest connection rather than trying to improve the connection quality on average.

$u_{\omega,k}$ is the control input at the time instance k. $\bar{\mathbf{x}}^{pos} \in \mathbb{R}^{3\times 1}$ and $\bar{\mathbf{x}}^{g,pos} \in \mathbb{R}^{3\times n}$ represent the position of UAVs and ground nodes, respectively. p_c, q_c and r_ω are constant weighting factors, ω_{max} and ω_{min} are the maximum and minimum turning rates limited by the UAV kinematics, respectively, Δu_ω is the maximum allowable control input difference, and r_c is a safe distance between UAVs to prevent collision. The constraints in (5.17) and (5.18) are to limit the maximum heading rate of the UAV, and the rate of change of

heading, respectively. Note that, at any given time step, the UAV is assumed to be able to take one of three actions: keep current heading rate and change by Δu_ω either to left or right. While constraint from (5.19) is used to prevent collisions between UAVs.

Due to the large computation time to obtain desired commands, a decentralised approach is used in this work. The decentralised approach implies that each UAV needs to determine the MST independently and calculate its optimal control based on future predictions of other UAVs and ground vehicles, meaning that communication between UAVs is required within one sampling time. The sampling time can be adjusted to match the capability of communication equipment on-board UAVs. If all UAVs have the same positional information about each other, then each MST would be the same. However, it is possible that due to disruption and delays, each UAV would have different MSTs. One possible way of mitigating the issue is for UAVs to share their MST using their wireless link, and choose one with the lowest cost. The decentralised approach is also more robust since if one UAV fails, others can continue a communication relay mission.

3.4 Numerical Simulations

Static scenarios are first considered to perform initial comparison of three communication performance metrics: the global message connectivity (GMC), the worst case connectivity (WCC) and the modified global message connectivity (mGMC). After validating basic properties, dynamic scenarios are then considered to investigate the effect of moving ground vehicles on the proposed mGMC performance. Simulation parameters for communication equipment are based on a low-cost and low-power commercial off-the-shelf Ubiquity Pico station M2, as shown in Table 5.4.

Table 3.1: Radio communication parameter

Parameter	Value	Unit
Transmission power (P_i)	0.01	W
Noise power (σ^2)	1×10^{-9}	W
Max communication range	300	m
Attenuation factor (α)	3	n/a
Antenna gain C for UGV and UAV	1	n/a

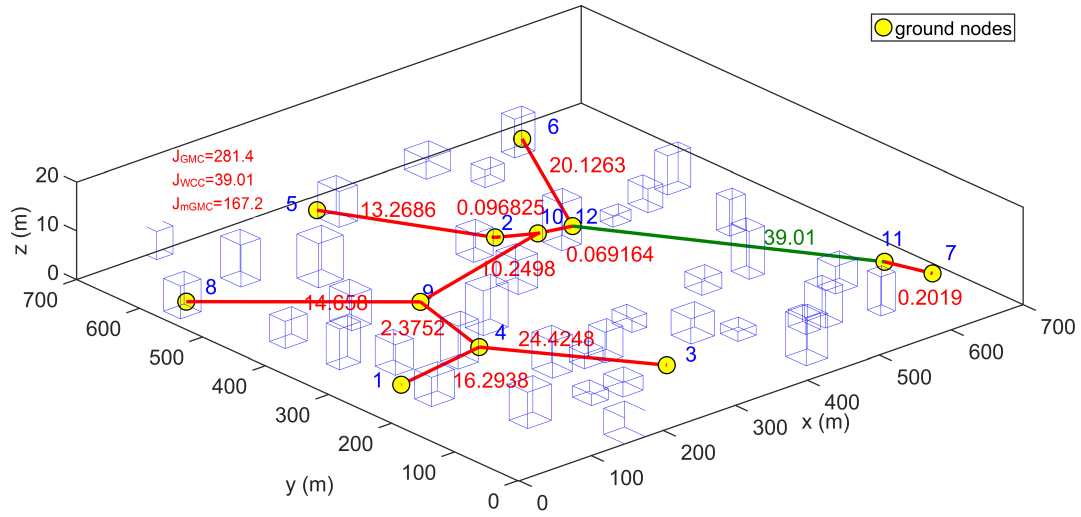
3.4.1 Static Scenarios

Sample Scenario

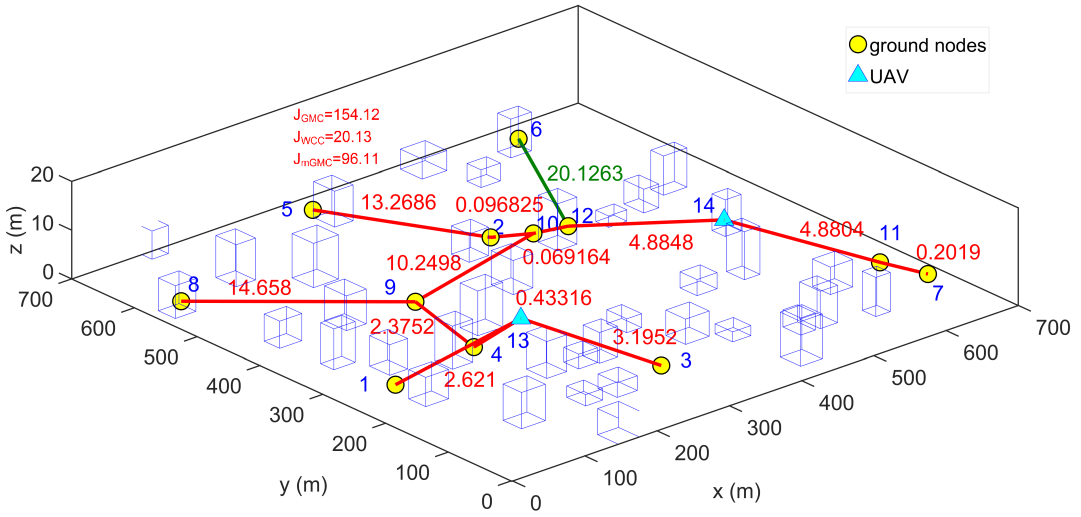
To establish the basic behaviour of communication performance metrics, a sample scenario with two relay UAVs, 12 ground nodes and 30 buildings is discussed. Figure 3.11 shows the

result of using different communication performance metrics in the optimisation process. For J_{mGMC} cost computation, $W^d = 20$ is used as a desired communication requirement, resulting in the number of weak connections $\beta = 6$ in the for the static scenario unless otherwise stated.

Firstly, Fig. 3.11a presents MST between ground nodes (shown as red lines) without relay UAVs. Numerous connections are weak due to: i) buildings obstructing line-of-sight (LOS) and ii) relatively large distances between nodes. The weakest connection within the MST is between nodes 11 and 12 (shown as green line) with the weighted edge cost of $J_{WCC} = 39.01$. The sum of all connection (termed as the global cost, hereafter) in the MST for this sample scenario is $J_{GMC} = 281.4$, and the sum of β worst connections is $J_{mGMC} = 167.2$.

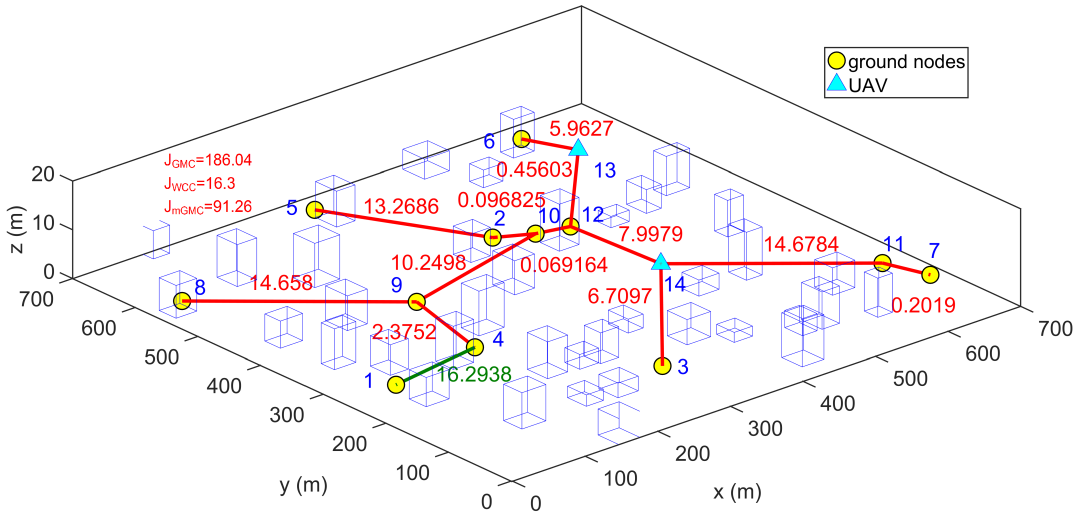


(a) MST without UAVs

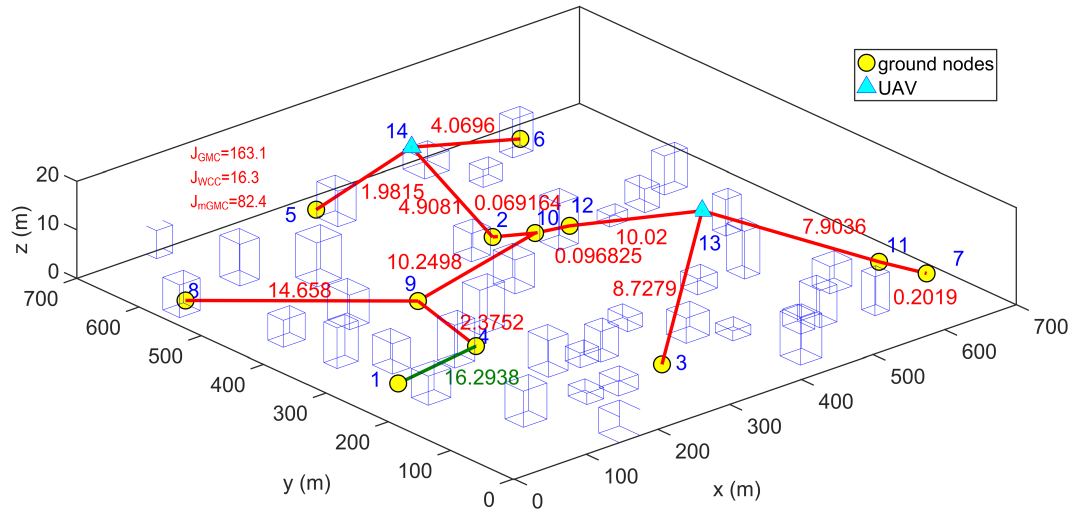


(b) Using the GMC with two UAVs

Figure 3.11: Optimal deployment results for relay UAVs using different communication performance metrics.



(c) Using the WCC with two UAVs



(d) Using the mGMC with two UAVs

Figure 3.11: Optimal deployment results for relay UAVs using different communication performance metrics (cont.).

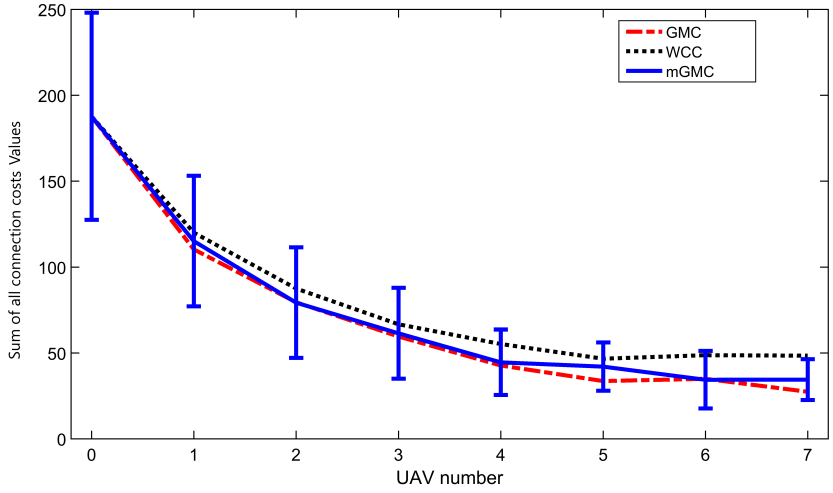
Utilising the GMC as the performance metric in the optimisation process is shown in Fig. 3.11b. As expected, using this metric reduces the global cost significantly to $J_{GMC} = 154.12$. However, the connection between nodes 6 and 12 becomes very weak ($W^t_{612} = 20.13$) since the weak connection is not particularly considered; the purpose of the GMC metric is to improve all connection overall. Figure. 3.11c shows the result of using WCC as the performance metric in optimisation. This metric improves the value of worst connection to 16.3 between nodes 6 and 12. However, the global cost $J_{GMC} = 186.04$ is the worst amongst three metrics. This comes from the nature of this metric which focuses on improving the weakest connection only while ignoring the global communication quality (i.e. all other connections).

Lastly, Fig. 3.11d shows the result of the proposed metric, mGMC. The resulting global cost for this metric is $J_{GMC} = 163.1$ which is in the middle between the GMC and the WCC at 154.12 and 186.04, respectively. Besides, the worst connection cost $J_{WCC} = 16.3$ is the same as using WCC as a metric. For the sum of three bi-directional worst connections (i.e. $\beta = 6$), this metric provides the best result ($J_{mGMC} = 82.4$) among three metrics. As the worst connection value is below 20, using the proposed mGMC metric satisfies the desired communication requirement, while providing the reasonable global communication performance as well.

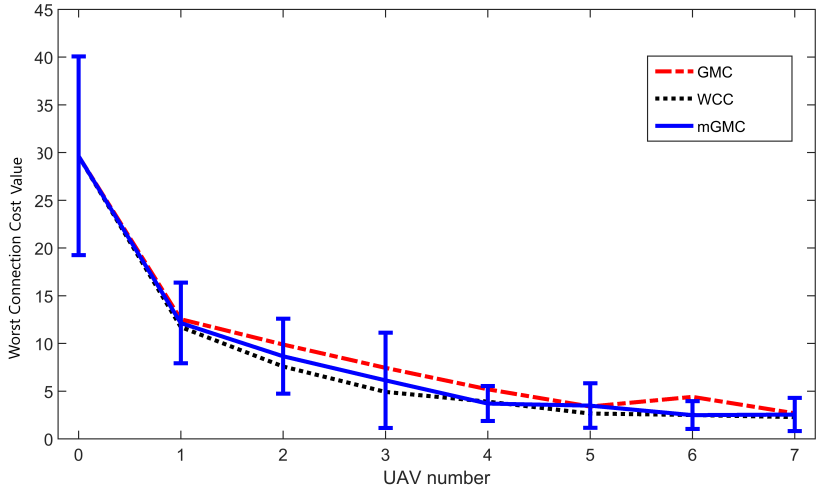
Monte Carlo Simulations

Monte Carlo simulations are used to confirm the aforementioned trends observed in the sample scenario. Figure 3.12 shows the results averaged over 30 independent Monte Carlo simulation runs. The global cost (J_{GMC}) change with the increasing number of UAVs is shown in Fig. 3.12a using three different performance metrics. It is worthwhile noting that in this simulation, W_d is adjusted so that $\beta = 6$ at all times. As expected, the GMC metric (shown as a blue solid line) produces the best result, however the result from the mGMC is not much worse, which is particularly the case for the small number of UAVs. With the small number of UAVs, there are a large number of bad connections and thus improving them led to a significant improvement on the global cost using either of the GMC or the mGMC. Similar trend is shown in the WCC, however, the metric gets worse much faster than the mGMC.

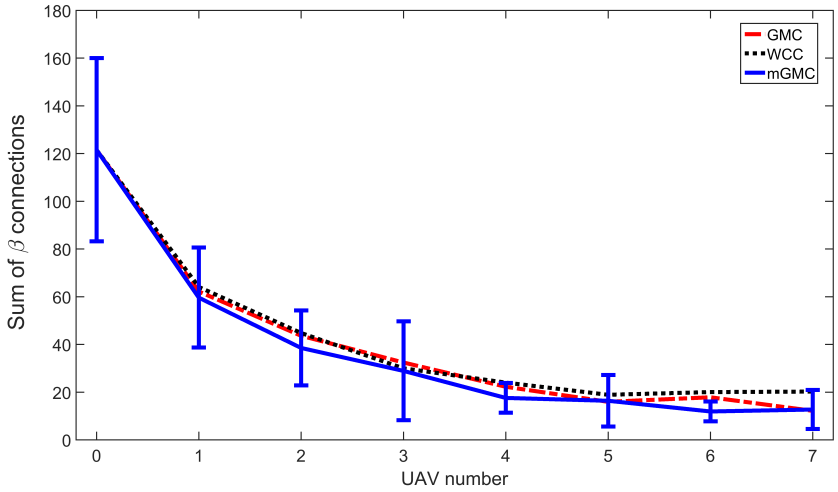
Figure. 3.12b with the WCC cost shows the reversed situation as in Fig. 3.12a. This means that using WCC as the performance index provides the best result in terms of improving the worst connection, while the result from the GMC metric is the worst. Comparing three three metrics with the sum of β=6 worst connection cost (J_{mGMC}) as shown in Fig. 3.12c, it can be seen that the mGMC metric case shows the best performance. It can also be seen that adding more UAVs gets less improvements. This is because as UAVs are added into the scenario, the number of bad connections is reduced until there are only good ones to improve.



(a) J_{GMC} cost



(b) J_{WCC} cost



(c) [J_{mGMC} cost

Figure 3.12: Monte Carlo simulation results using three different communication performance metrics with different number of UAVs.

3.4.2 Dynamic Scenarios

Having shown comparison of the proposed mGMC metric with other metrics on static scenarios, investigation of this metric on dynamic scenarios is performed. It is worthwhile noting that due to movement of ground vehicles and corresponding communication quality change, β also changes; thus, it is calculated at each simulation step to take into account all connections above W^d. The dynamic scenario with three UAVs, 12 ground nodes and 30 buildings uses the same communication parameters from Table 5.4, and other simulation parameters are shown in Table 5.5. In this section, the effect of following changes are mainly considered: i) change of horizon length in the NMPC problem formulation, ii) different movement pattern including random, loitering and the proposed method, iii) the number of UAV used, and iv) kinematic constraints of the UAV including the speed and turning rate.

Table 3.2: Mobile simulation parameter

Parameter	Value	Unit
Actuator delay (τ_ω)	1/3	s
UAV speed (v)	20	m/s
Heading rate constraint ($\omega_{min}, \omega_{max}$)	$(-0.2, 0.2)$	rad/s
Receding horizon step (N)	5	N/A
Maximum heading rate change (ΔU_ω)	0.1	rad/s
Weighting factors (p_c, q_c, r_ω)	$(1000, p_c/N, 1)$	N/A
Population size (M)	15	N/A
Acceptable number of generations (N_g)	15	N/A

Firstly, Fig. 3.13 shows that the horizon length has a critical impact on the performance when using the mGMC. If the horizon length is too small, UAVs will not see any possible trajectory which can improve the mGMC cost as the mGMC is only focused on a few worst connection, which might not be in that UAVs range. This situation is illustrated in Fig. 3.14. On the other hand, if the horizon is too long, ground vehicles' plan needs to be known well in advance and UAVs may overlook imminent weak connections for the sake of improving connections far away. To trade off between those two problems, the horizon length needs to be carefully determined. There might be two ways to change the horizon length: i) increasing the horizon steps with a fixed time interval or ii) increasing time interval between horizon steps with a fixed number of horizon steps. First method would result in significantly increased computational efforts but an accurate solution while the second does not increase computational efforts, but it reduces the accuracy of a solution. Since the computational speed is one of the requirements for the proposed algorithm as stated in the introduction, the second method is employed in this work. Table 3.3 shows

the different time intervals between horizon steps with five horizon steps (which results in different total horizon length from 5 to 25) used in the numerical simulation shown in Fig. 3.13. It can be seen that there is a significant performance improvement up to the horizon length of 15 and smaller improvement between 15 and 20. After the horizon length of 20, the performance of the mGMC cost starts to decrease (so value increases) again and can be associated to UAVs trying to improve connections too far in the future.Thus, the best horizon length for the mGMC in this size of scenario is 20 and this horizon length will be used in other simulations hereafter.

Table 3.3: Different time intervals between horizon steps

Total horizon length [sec]	time interval between horizon steps [sec]
5	$(0.5, 0.5, 1, 1, 2)$
10	$(0.5, 0.5, 2, 3, 3)$
15	$(0.5, 2.5, 3, 4, 5)$
20	$(0.5, 4.5, 5, 5)$
25	$(0.5, 4.5, 6, 6, 8)$

Figure 3.15 shows comparison between random movement, loitering (circling) around a starting point and using the proposed GA optimisation. At the beginning, all three methods show similar performance and the loiter shows even better performance during around 30 and 40 seconds of simulation. This might be because fixed-wing UAVs cannot change its heading or position instantaneously towards the optimal heading/position due to their dynamic constraints. However, the proposed method outperforms the other two for most of the time as it optimised its movement considering the movement of all ground nodes and the corresponding mGMC cost.

The effect of changing the number of UAVs is shown in Fig. 3.16. As expected, the more UAVs are used, the better the sum of β connections is obtained. Note that the difference between two and three UAVs is not that significant. This implies that there is a saturation point where adding more UAVs does not result in a significant increase of communication performance. As more UAVs are added, there are fewer weak connections to improve than if a small number of UAVs is present (e.g. it is possible a single connection is improved by two UAVs in a chain rather than a single UAV).

Different heading rates and speeds are used to determine how UAV dynamics affects the communication performance. Figures 3.17 and 3.18 show that higher turning rates and slower speeds results in better communication performance in terms of lowering the communication cost. In these conditions, UAVs can make tighter turns, thus they can better react to the local changes of ground node positions and corresponding communication structure (i.e. MST). Also, fig. 3.18 Allows for assessment of constant speed assumption.

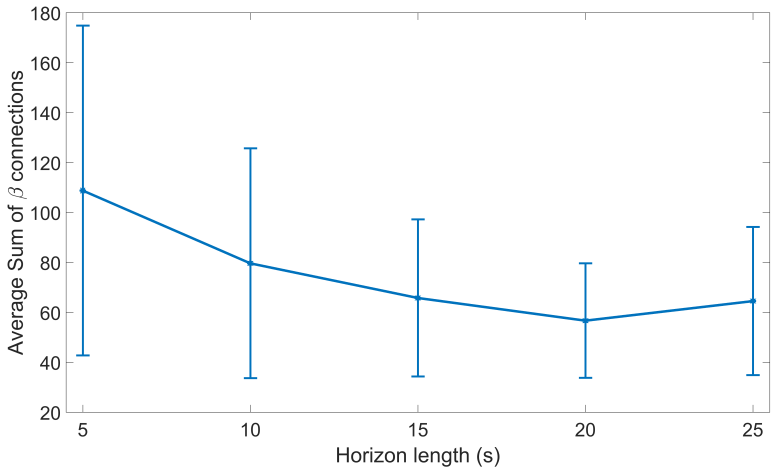


Figure 3.13: Averaged sum of β connections over ten scenarios.

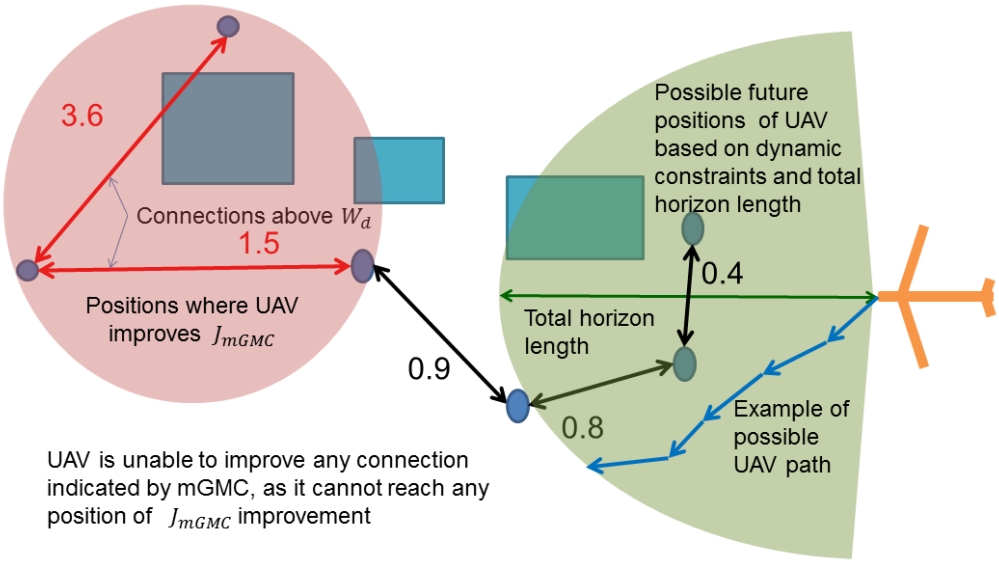


Figure 3.14: Illustration of a problem of too short horizon length where the UAV cannot see a move which would reduce the mGMC cost.

It can be seen that too low speed can result in a significant performance drop (25 to 40 second region on the graph). However, for most of the time speed of the UAVs do not seem to be affecting the performance of the relay. It would be worth optimising for speed if possible just to gain a little boost in performance, but it is not deemed vital.

To illustrate a significant reduction of the number of weak connections using the proposed algorithm, the mGMC cost and the number of the weak connection (communication quality above W^d) of a sample scenario with and without using UAVs is shown in Fig.3.19. In this scenario, UAVs cause a significant reduction of number of weak connections where the average number of weak connections with UAVs is about 1.6, while without the UAV is 4.4, proving the benefit of the proposed approach.

Currently, control commands need to be sent to UAVs every half a second, so each NMPC iteration needs to be finished within that time. It is also worth noting that al-

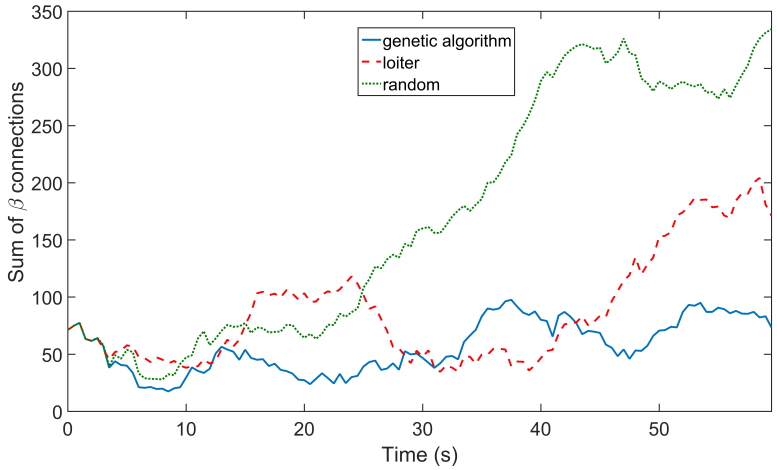


Figure 3.15: Comparison of NMPC-based trajectory planner using the genetic algorithm implementation with random motion and loitering.

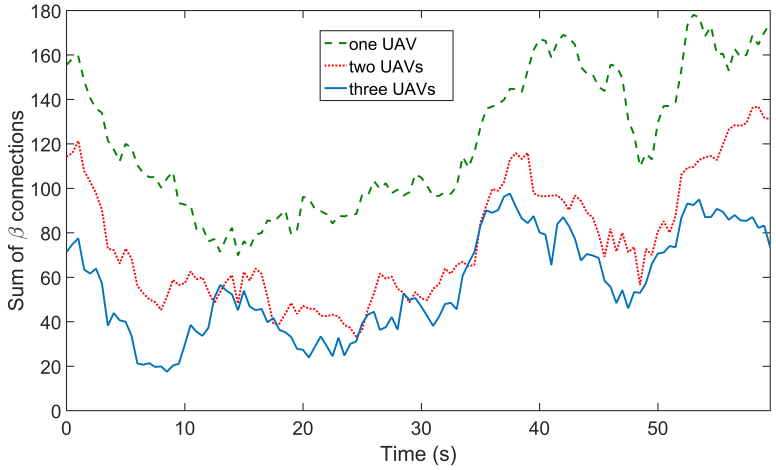


Figure 3.16: Effect of changing number of UAVs on the performance of the mGMC metric.

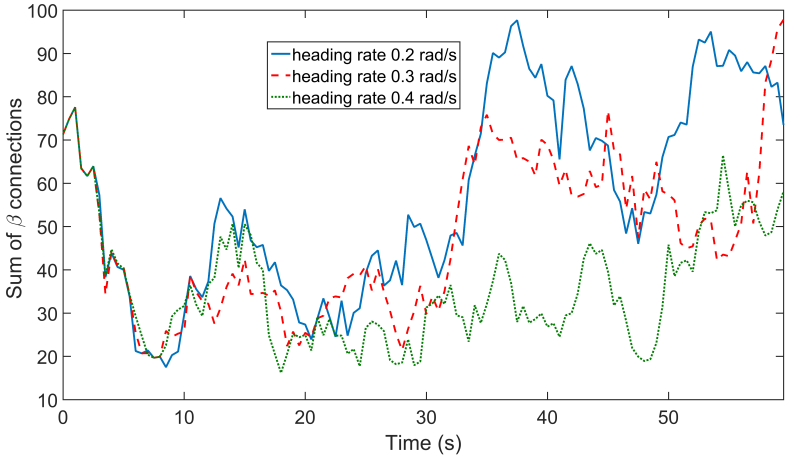


Figure 3.17: Effect of changing heading rate on the performance of the mGMC.

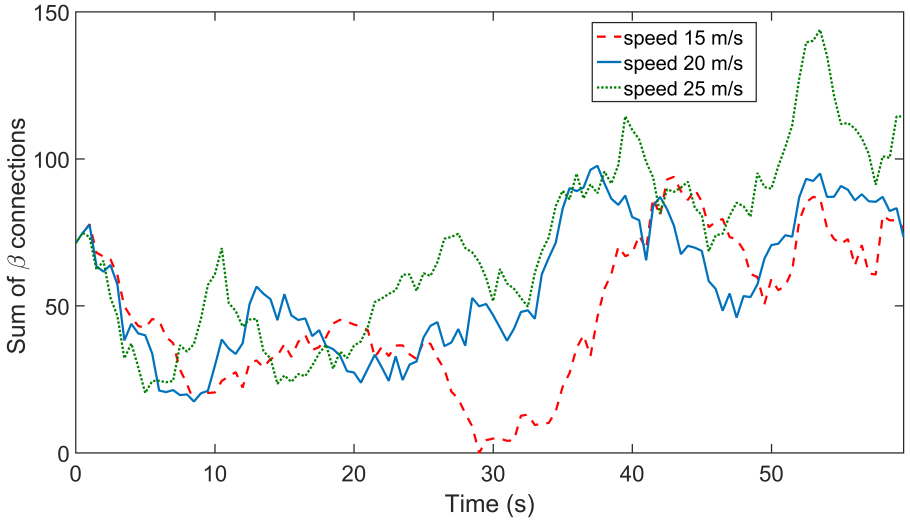


Figure 3.18: Effect of changing speed on the performance of the mGMC.

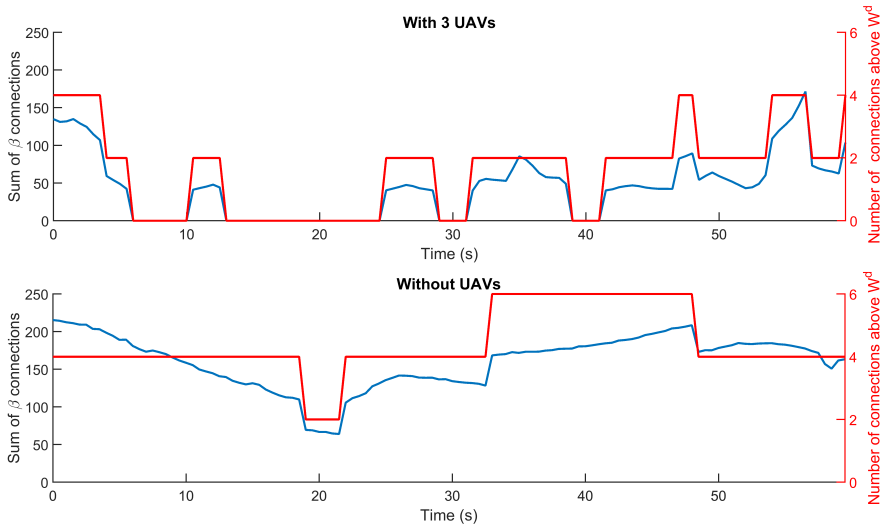


Figure 3.19: Comparison of performance (J_{mGMC} cost) between cases with and without UAVs.

gorithm can be run in a decentralized manner (i.e. each UAV performs its own NMPC computation based on the prediction on what the rest of the group will do). Thus, computation time per UAV is representative of how fast the group can finish one iteration. Table 3 shows the computation cost of one iteration, implemented in Matlab, using the same scenario described before with and without buildings in the environment. This implies the significant impact of LOS obstruction detection on the computation time. Figure 11 represents the computation time with respect to the number of ground vehicle. Possible approaches to address the issue on the computation time include: i) implement the algorithm in C/C++. Using a lower level language will reduce computation time significantly; ii) commands can be sent less often to the UAV. This will increase the allowed time

for each iteration to finish at the expense of communication performance of the algorithm; and iii) the number of iterations or population size of the genetic algorithm can be reduced to achieve significant reduction of the computation time. However, this is likely to reduce the quality of solution.

Table 3.4: Mean and standard deviation for a runtime per iteration averaged over ten sample scenarios

Computation time	Without buildings	With buildings
Mean time ± Std (sec)	0.29 ±0.0088	8.9±0.55

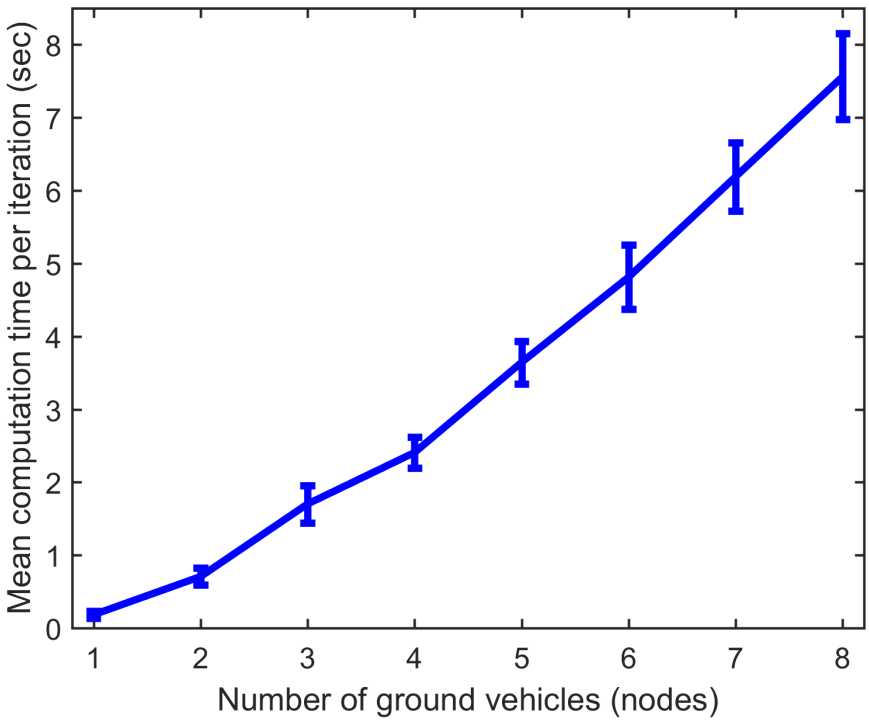


Figure 3.20: Computation time for a single UAV and 30 buildings with respect to the number of ground vehicles.

3.5 Indoor Flight Experiments

To validate and verify the feasibility and benefits of the proposed algorithms, flight tests are performed in an indoor environment. This section starts with describing the indoor experiment testbed, followed by experimental procedure and setup. Finally, sample captures from the experiments are presented. Note that a movie clip of indoor experiments and simulations presented in this report can be downloaded at

`https://dl.dropboxusercontent.com/u/17047357/Comm_Relay.zip`.

3.5.1 Indoor Experiment Testbed

The overview of an indoor experiment testbed system is shown in Fig. 3.21. The system can be largely divided into three sections: positioning system, air and ground vehicles and ground control station. From the following, each section is explained in detail.

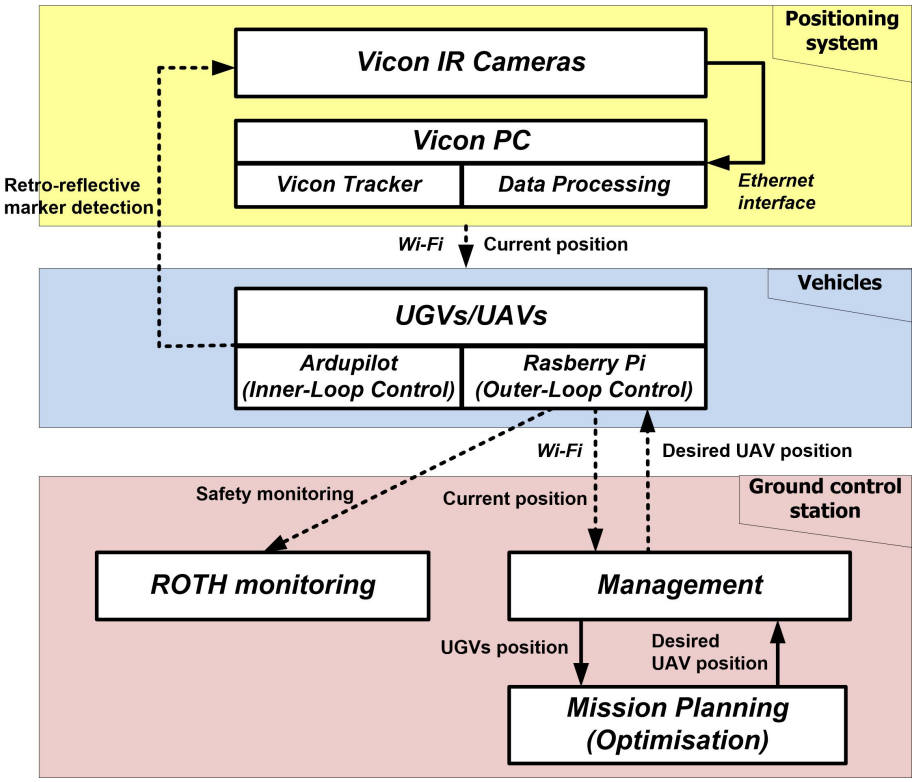


Figure 3.21: Indoor experiment testbed overview

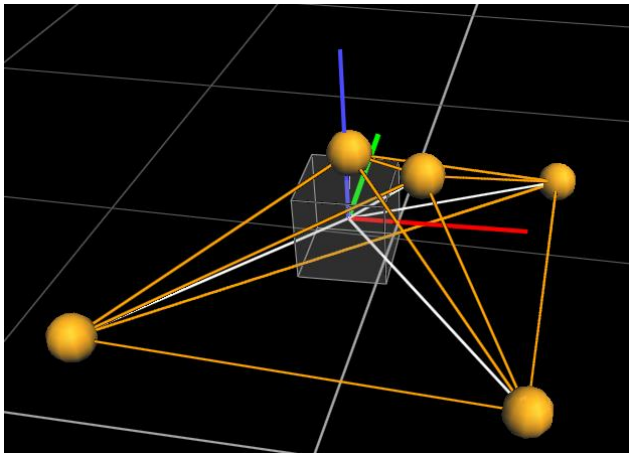
Vicon motion tracking-based positioning system

In order to perform experiments in an indoor environment, the position of each vehicle in three dimensional (3D) space is needed. A GPS signal indoor is too weak to be of any use. The Vicon motion tracking system is used as indoor GPS replacement shown in Fig. 3.22. The system uses a series of accurately located infrared (IR) cameras, which are equipped with infrared light emitting diodes (IRLEDs), to track the relative position of retro-reflective markers that have been attached to the moving vehicle. To obtain accurate data, markers need to be adequately separated and visible by a minimum of two cameras. This allows position and attitude data to be obtained to within 0.1mm and 1 deg, respectively. The position data for each vehicle is then read by Simulink using Vicon software development kit and forwarded to corresponding vehicles.

Vehicles: UAV and UGV

Figure 3.23 shows a quadrotor UAV used in this experiment. It is equipped with Ardupilot and Raspberry Pi B+: Ardupilot (commercially-available autopilot product) is used for the

(a) Vicon infrared camera



(b) Image of tracked reflective balls attached to the vehicle

Figure 3.22: Vicon motion tracking system

roll and pitch angle stabilisation of the platform during the flight (i.e. inner-loop control); and Raspberry Pi is for the outer-loop position and yaw control as well as receiving and sending information related to desired and current position of the vehicle. Along with the UAV, Lynxmotion Aluminium A4WD1 ground vehicles are used in the experiment as shown in Fig. 3.24. For the sake of keeping the system consistent across all the vehicles, the UGV uses the same hardware combination of ardupilot and Rasberry Pi as used for the UAV, with firmware adjusted for a ground vehicle.

Each vehicle runs a Simulink model, through run on target hardware (ROTH) capacity. ROTH works by firstly generating C++ code from the Simulink model. It is then uploaded to the Raspberry Pi and complied, thus allowing for direct execution on Raspberry Pi. In this experiment, each vehicle runs a PID controller for position and yaw control as shown in Fig. 3.25. It is worth noting that a UAV's control structure is slightly more complicated than that of the UGV as the UAV operates in 3D space. Besides, Figure 3.26 illustrates data flow and processes of each vehicle (represented as Embedded model) in relation to

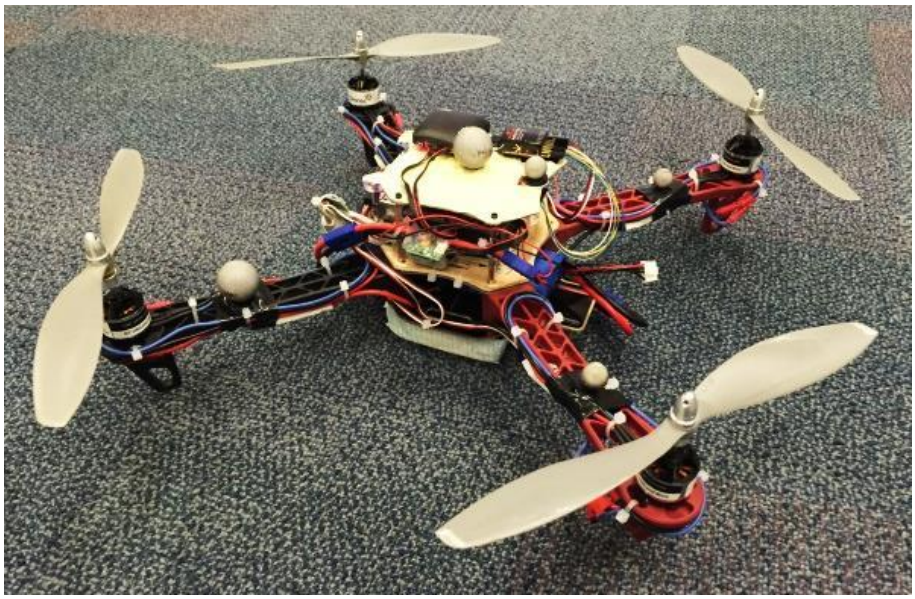


Figure 3.23: Custom-built quadrotor UAV.

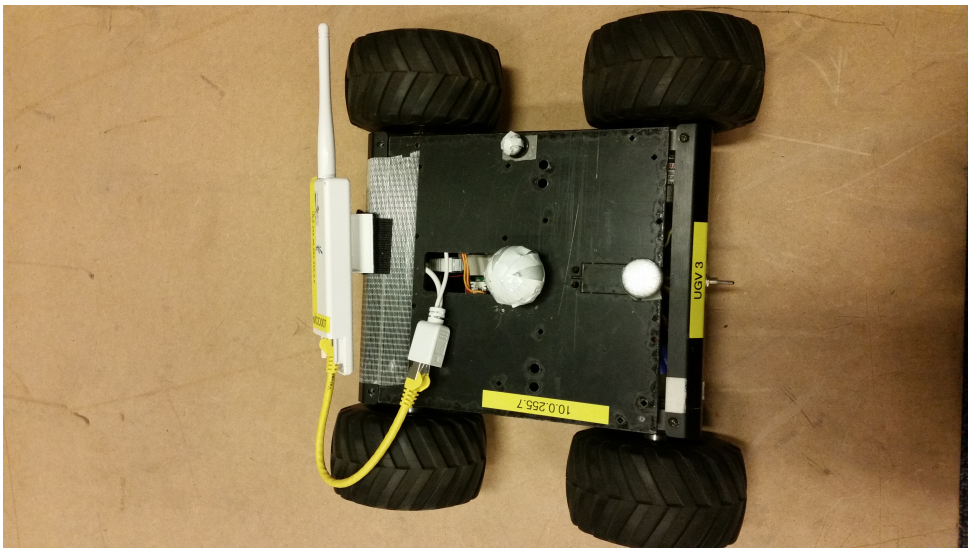


Figure 3.24: Lynxmotion Aluminium A4WD1 UGV.

the Vicon positioning system and a ground control station.

Ground control station

Ground control station consists of three separate models: management station, mission planning (optimisation) station and Run on target hardware (ROTH) monitoring station as shown in Fig. 3.21. First, the management station is responsible for controlling time synchronisation of the entire system and sending time-stamped desired waypoints to vehicles. It also has direct control over the quadrotor UAV in case that there are problems in autonomous flight. The mission planning station uses known desired position of UGVs to calculate next desired position of the relay UAV. It then transmits it to the UAV via the

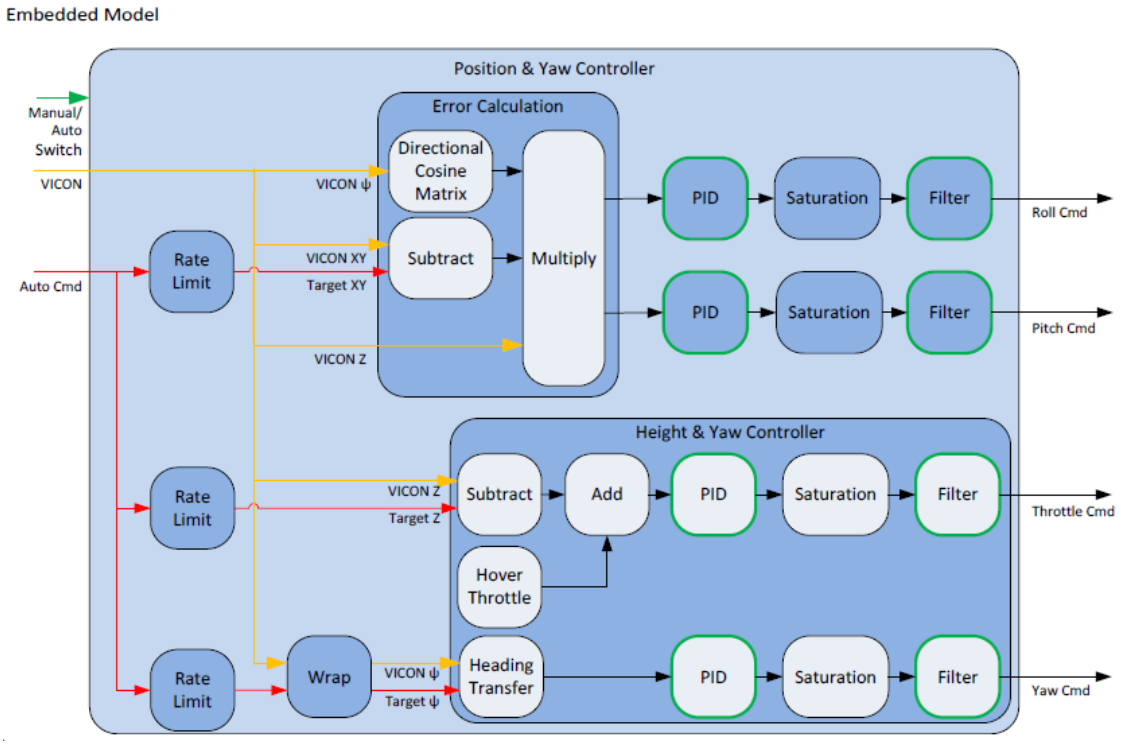


Figure 3.25: Schematic diagram of position and yaw controller with data flow and processes.

management station. The ROTH monitoring station monitors the status (e.g. position, attitude, operation mode, etc.) of each vehicle.

Although all models can be run on a single computer, each model in the ground control station runs on a separate machine for safety and computation efficiency. To be more specific, first of all, the management station is a safety critical model, so it is necessary for it to be separated from others. Moreover, since the mission planning station might require significant computing power for optimisation, it is decided to dedicate one computer to it as well. Lastly, since separate ROTH monitoring stations are required for each vehicle, it is worthwhile running on a single machine.

3.5.2 Experimental Procedure and Setup

Both optimal UAV deployment planning and convoy following trajectory planning scenarios are slightly modified for real implementation. The mission of the experiment is defined by a series of discrete time-stamped waypoints for UGVs. For this dynamic scenario, the relay UAV is supposed to be positioned accordingly depending on the position of UGVs. Two scenarios run in a similar manner, however time between each step is different for each scenario. The experiment procedure is shown in Fig. 3.27, and steps are outlined as below:

1) The UAV takes off and is piloted manually close to its starting point.

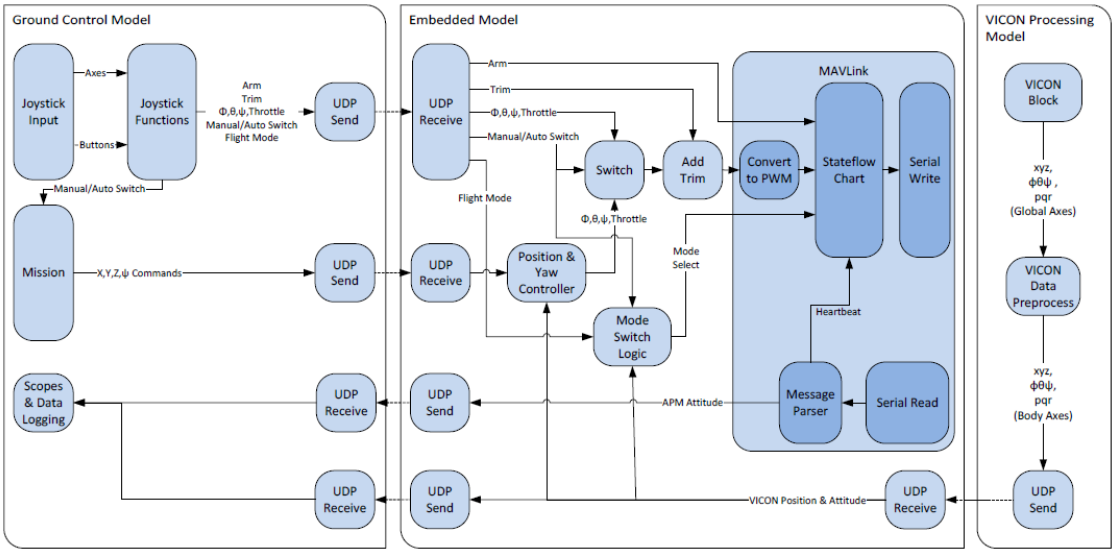


Figure 3.26: ROTH model structure with data flow and the processes to support manual and autonomous operation and the transmission of data. Note that the mission in the ground control model comes from either manual control or mission planning (optimisation) station.

2) The autonomous mission is triggered with ground vehicles at inital positions.

3) The mission planning station computes the desired position of the UAV using PSO.

4) The UAV reaches the desired position.

5) After the specified time, UGVs starts to move to the next position and calculation for the next UAV position starts.

6) UGVs reach their next position.

7) As soon as calculation is done, the UAV moves to the next desired position.

8) Steps 6∼8 are repeated for pre-specified number of steps depending on the mission.

The experiment was performed in an area with 5 by 5 meters size which can be covered by Vicon cameras. The city layout is shown in Fig. 3.28. It consists of 15 buildings arranged in a T-shaped junction. The building positions are accurately measured using Vicon system. Note that the T-junction is used to allow for flexibility in possible scenarios.

3.5.3 Experimental Results

In this experiment, three UGVs and one UAV are used. The time gap between UGV waypoints are set to 40 seconds. Figure 3.29(a) shows the first optimal position of the UAV for the initial position of UGVs with red lines representing the MST. In this case, the UAV is required to function as a relay between all three UGVs as the LOS between

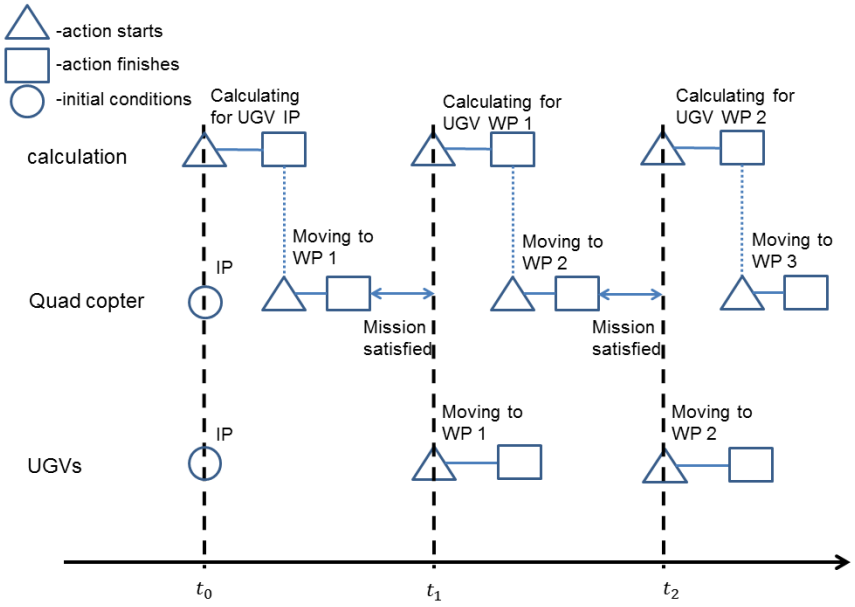


Figure 3.27: Experiment timeline.

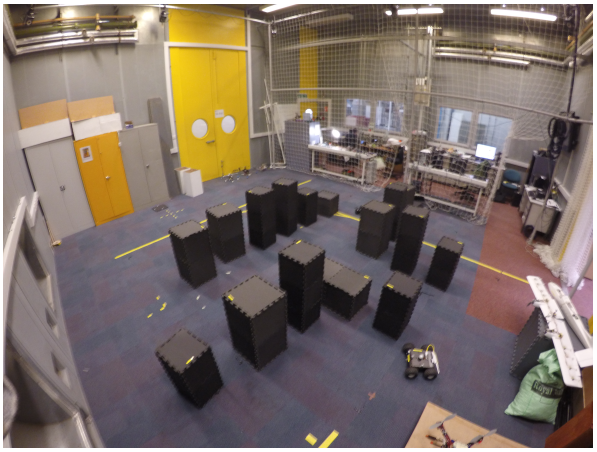
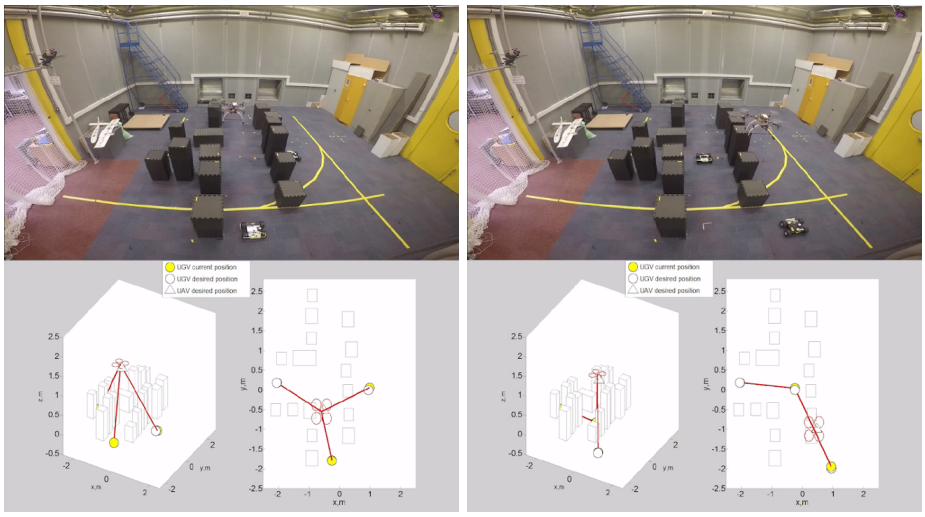


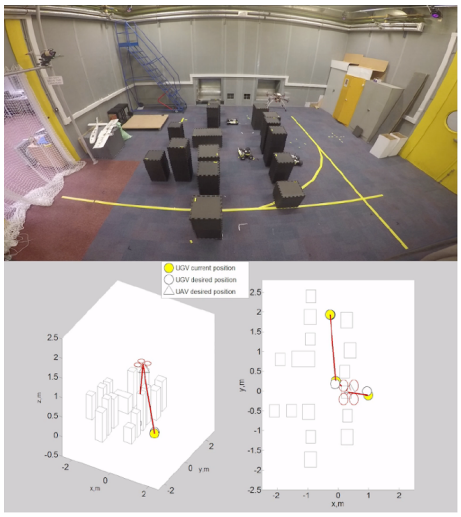
Figure 3.28: Urban environment used in experiment. Black boxes represent buildings.

all of them is obstructed by buildings. This frame shows the great benefit of using the relay UAV as otherwise communication would not be possible. Figure 3.29(b) shows the status about 60 seconds later. The UGV in the middle lane and the UGV in the right lane changed their position, which makes the LOS to each other blocked. The UAV ensures communication between those two vehicles. As the other UGV (in the left lane) has a direct LOS to the middle lane UGV, those two are directly connected, not through the UAV. For the final UGV waypoints as shown in Fig. 3.29(c), the UAV moves to the position which improves communication between UGVs in the middle lane and the right lane as the LOS is blocked.



(a) First UGV waypoints

(b) Second UGV waypoints



(c) Third UGV waypoints

Figure 3.29: Snapshots from the experiment which finds the optimal UAV positions and the MST (red lines) for UGV waypoints.

3.6 Summary

This chapter has presented a trajectory planning and positoning algorithms for fixed-wing communication relay UAVs to enhance communication quality of the ground mobile network in an urban environment. To plan the optimal trajectory, discretised NMPC trajectory planning was proposed using a genetic algorithm-based optimisation method. While to position UAV, approach based on PSO was proposed. Both algorithms were paired with the GMC, WCC and mGMC communication performance metric targeted to improve only the necessary weak connections in the network. For positoning performance of GMC, WCC and mGMC was validated, both on single sample scenario and Monte Carlo simulations. For trajectory planner, it was shown how different horizon length, the

number of the UAV used, and kinematic constraints of the UAV such as the heading rate and the speed can affect the communication performance. Benefits of proposed approaches were presented by comparing performance with and without UAVs. An indoor flight test showed potential for the near real-time algorithm execution.

Chapter 4

Measurement-based Approach for Unknown Urban Environment

In this chapter, a new communication model is introduced to address two of the challenges of the communication prediction models described in chapters 3 and 5. First, it was assumed that all obstacles present in the urban environment reduce communication performance to a similar degree. However, buildings are made from different compositions of materials which as introduced in Chap. 2. Each composition affects communication performance in different ways. Even the presence of small objects such as trees and lamp posts can affect wireless communication performance. Moreover, previous approaches required either i) knowledge of the 3D map of the environment in Chap. 3 or ii) that it fits one of the predefined urban environment types from Chap. 5. Such knowledge is sometimes unavailable. With those issues, an UAV would have to use an incomplete understanding resulting in a mediocre performance (an example of this is shown in Fig 5.7). To cope with situations of lack or incomplete knowledge we propose an approach which learns the majority of communication model during the mission. This is achieved by using machine learning to predict the effects of urban environments on the wireless signal without any prior knowledge.

For this prediction GP machine learning is employed. This technique has three main advantages, which makes it suitable to solve the aforementioned limitations. First of all, it is a non-parametric machine learning technique. Non-parametric property implies that with careful formulation and appropriate choice of mean and covariance functions, it can be used without any pretraining. By omitting pretraining, we can reduce the risk of not preparing for a given scenario type from the training dataset. Instead, GP relies on collecting communication strength data from ground nodes during a UAV flight and then fitting the GP communication strength model to collected data. The second benefit is that GP can fix its mistakes in prediction. Due to the stochastic nature of wireless communication, it is possible that false prediction is made. However, The data obtained in later phases of flight can fix this mistake and adjust the forecast accordingly. Finally,

with GP, it is possible to account for any wireless signal interference, not only ones created by buildings. Other devices could cause such an interference with a similar frequency or active jamming.

GP is paired with non-linear model predictive control (NMPC) based trajectory planner. This trajectory planner is ideally suited as it can adapt trajectory to improving communication strength prediction. With the progression of the relay mission, a UAV collects more communication strength data between air and ground to improve its prediction. NMPC is ideally position to exploit the updates as the planned trajectory is updated periodically as part of the formulation.

With its ability to solve some of the issues of previous two models, GP has two significant limitations i) it is unable to deal with mobile ground nodes and ii) it is challenging to include the height in path prediction. Both those problems are a consequence of a way GP makes its predictions. In GP, communication strength prediction depends on a number and a location of collected strength data. This is not an issue for stationary ground nodes, as UAV can quickly fly to scan the area. However, for mobile ground nodes, it would be necessary for both ground nodes and UAV to travel to learn communication strength between an arbitrary point in the air and on the ground. This would result in the unacceptably high time needed for collecting the data. Moreover, In GP the more the data points, the slower the prediction time. With communication strength collected on the ground and in the air, the prediction time would be considerably large even for modern computers. Similar reasoning can be used to understand the problem with adding height to path planning. For GP to be able to make predictions at different heights, a UAV would need to perform the scan in a 3D space, rather than on a 2D plane, taking a significant amount of time. This could be somewhat mitigated by having multiple UAVs each covering different heights. However, This does not solve the issue of having too many data points needed for reasonably fast prediction. To conclude, rather than replacing of the other two models, GP is another option which offer certain advantages in suitable circumstances.

4.1 Problem Overview

4.1.1 Scenario and Assumptions

We consider an urban environment where a fixed-wing UAV serves as a flying communication relay station for ground nodes, as illustrated in Fig. 5.5. There are multiple ground nodes where the line-of-sight (LOS) between them is likely to be obstructed by buildings in a complex urban environment. The purpose of the relay UAV trajectory planning is to fly in a way to improve the average communication performance between the UAV and all ground nodes from its initial position. We consider randomly generated cities with different building heights and different ground node locations.

The assumptions used in this study include: i) ground nodes are stationary and able to transmit their accurate position to the relay UAV prior to a mission; ii) the positions and shapes of buildings are unknown; iii) path loss and transmitted power of communication strength between any two nodes are empirically known, but shadow fading and needs to be predicted.

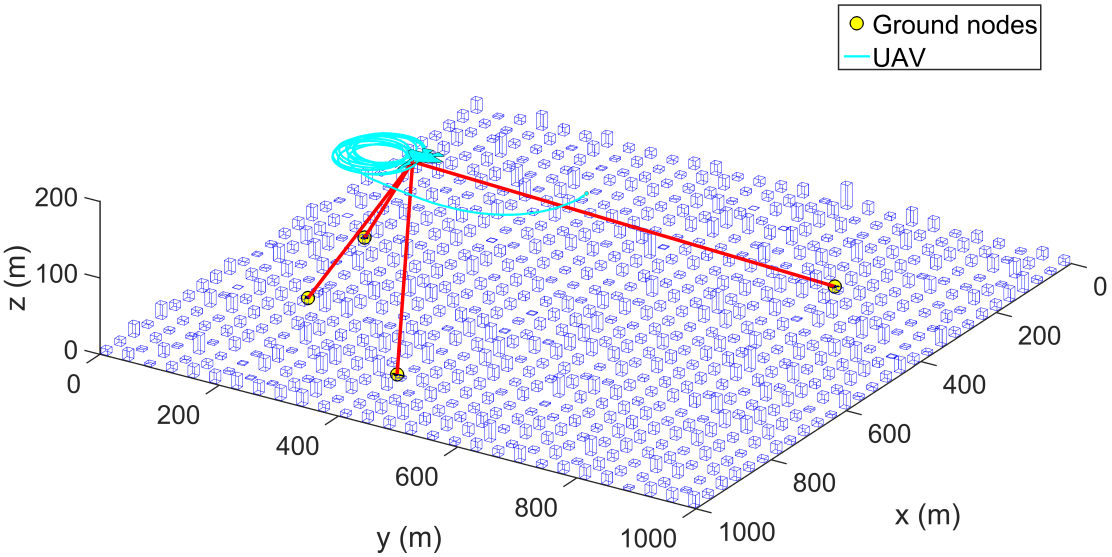


Figure 4.1: A sample urban scenario with four ground nodes, a relay UAV and buildings with different heights. The proposed trajectory planning guides the UAV to fly from the initial position to the optimal position that can provide the best communication quality to ground nodes.

4.1.2 Algorithm Overview

In order to plan the online trajectory of the relay UAV to improve the quality of air-to-ground communication, accurate channel prediction between the UAV and ground nodes is of primary importance. To this end, this study proposes two learning schemes using the Gaussian Process (GP) combined with the nonlinear model predictive control (NMPC)-based trajectory planning as shown in Fig. 4.2 and summarized as follows:

- **Scan+NMPC with GP** is an approach where the relay UAV performs a pre-specified scanning pattern flight in the area of interest (see Fig. 4.3) to collect communication strength measurement from ground nodes. The collected data can then be used by GP to build the map of communication channel strength between an arbitrary UAV position and ground nodes. With this communication map, the NMPC-based trajectory planner can be used to guide the UAV to the optimal position in terms of maximizing the air-to-ground communication performance.

- **NMPC with GP** is different from the above in the sense that the scanning is not performed. Instead, the UAV plans its trajectory with its current best knowledge

on air-to-ground communication. For the initial plan, it uses the knowledge it has a priori (e.g. the communication map built from the empirical communication model only without any data). Then, the UAV periodically collects communication channel strength data to update the communication map using GP, while executing its optimal control command obtained from the trajectory planner.

The NMPC with GP method could save the time spent on the scanning flight, but the accuracy of the communication quality prediction might be worse than that of scan +NMPC with GP. Besides NMPC with GP will be useful in a situation where there is a change in the communication environment during the mission.

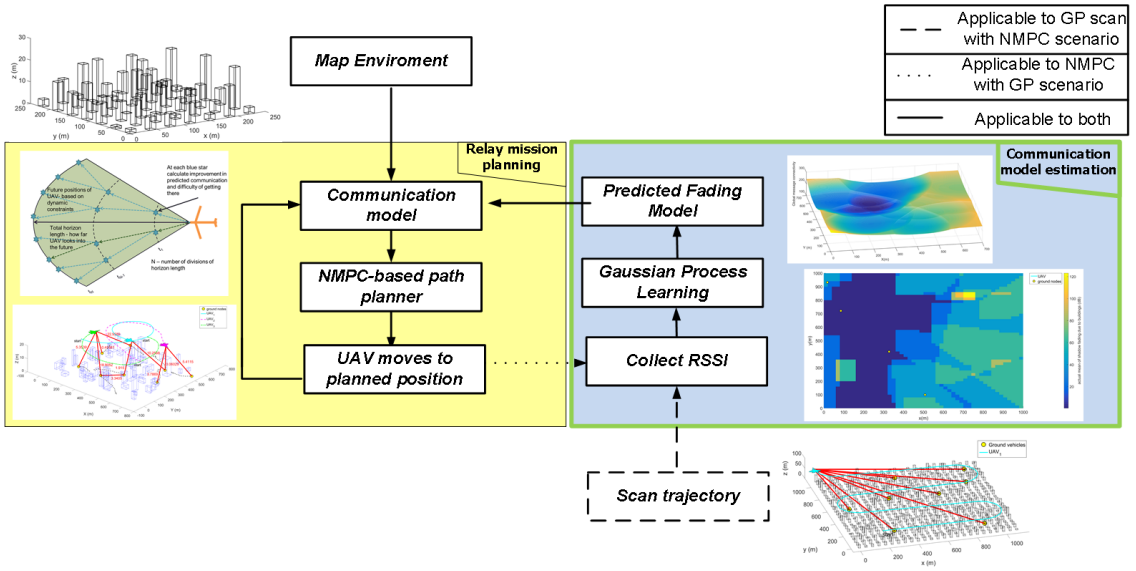


Figure 4.2: Overview of the two proposed algorithms: i) scan+NMPC with GP and ii) NMPC with GP for the communication relay mission.

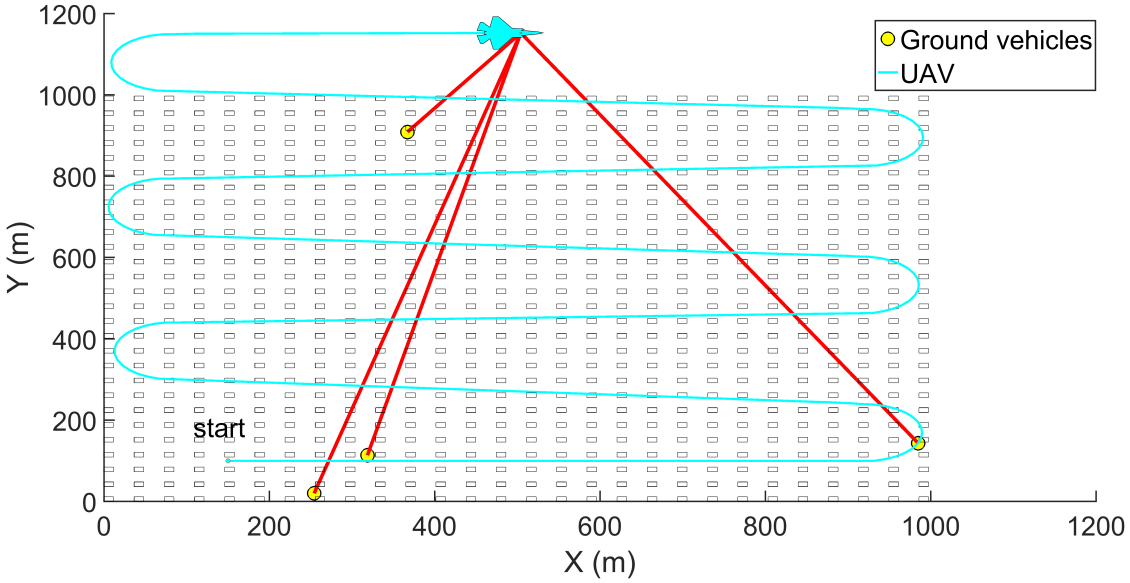


Figure 4.3: Example pattern for the UAV scan flight on a sample scenario.

4.2 Communication Channel Model and GP-Based Channel Prediction

4.2.1 Air-to-ground Channel Modelling

In this chapter communication model used is adapted from section 5.2.1. While they are very similar, they are used in a different context and difference are present so for conviniece of the reader full formulation is repeated here.

To realistically model communication quality between nodes, the channel model proposed by [46] is used in this work. This model is designed for aerial platforms based on multiple ray tracing simulations in different types of urban environments, proposed by the International Telecommunication Union (ITU-R) [112]. The urban type is characterized by three parameters α_0, β_0 and γ_0: α_0 is a ratio of the built-up land area to the total land area, β_0 is a mean number of buildings per unit area (buildings/km^2) and γ_0 is a scale factor that describes the building height in the Rayleigh probability density function as:

$$P(h) = \frac{h}{\gamma_0^2} \exp\left(\frac{-h^2}{2\gamma_0^2}\right), \tag{4.1}$$

where h is the building height in meters. The model considers four environments: i) Suburban, ii) Urban, iii) Dense urban, and iv) High-rise urban, where the corresponding parameters are summarized in Table 5.1. To fully define the urban type, a layout of a city is also needed along with parameters α_0, β_0 and γ_0. This study uses 'standard city' layout as defined by [46] as depicted in Fig. 5.3. As building heights are characterised by a probability density, it is convenient to use this model when performing Monte Carlo simulations with different heights of the buildings without the risk of buildings overlapping with each other.

Table 4.1: different city environment parameters

Environment	α_0	β_0	γ_0
Suburban	0.1	750	8
Urban	0.3	500	15
Dense Urban	1	300	20
High-rise Urban	1	300	50

The communication channel model used for the above urban environment can be represented as a general formulation [45]:

$$P_{r,i} = P_{t,i} - L_{dB,i} - \Psi \tag{4.2}$$

where $P_{r,i}$ (dBm) is the received power strength at the UAV from the node i, $P_{t,i}$ (dBm) is the transmit power of the node i, $L_{dB,i}$ represents the path loss between node the i and

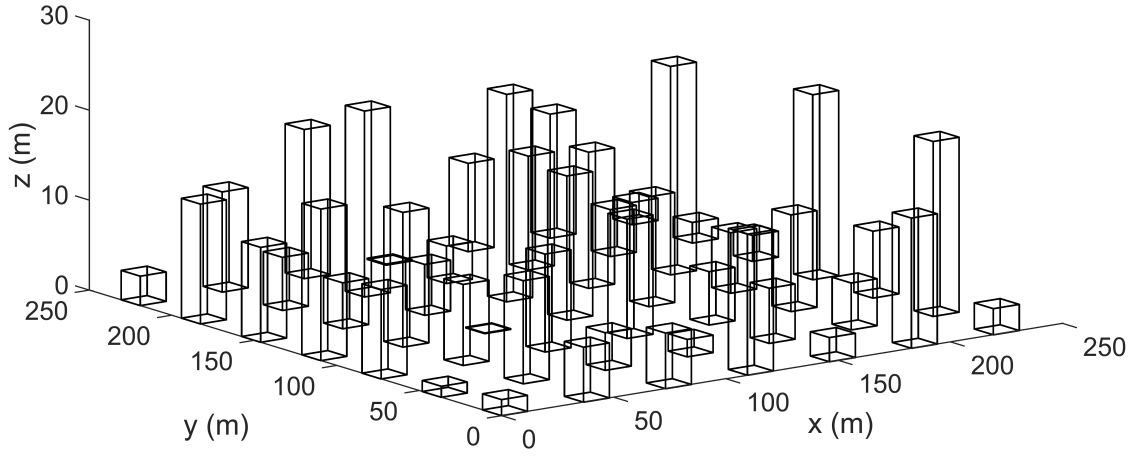


Figure 4.4: A sample city generated with parameters: $\alpha_0 = 0.1$, β_0 =750 and $\gamma_0 = 8$.

the UAV. Ψ is the unknown shadow fading assumed to be a Gaussian distributed random variable as $\Psi \sim \mathcal{N}(\mu, \sigma^2)$, where μ and σ^2 are fading mean and variance. Note that in this model, shadow fading accounts only for diffraction on multipath fading caused by buildings. $L_{dB,i}$ can be represented as:

$$L_{dB,i} = 10\alpha \log_{10}\left(\frac{4\pi f_c d_i}{c}\right), \tag{4.3}$$

where f_c is the central frequency, α is the path loss exponent, d_i is the distance between the ground node i and the UAV, and c is the speed of light. As $P_{t,i}$ and $L_{dB,i}$ are constants and known, $P_{r,i}$ also follows Gaussian distribution as:

$$P_{r,i} \sim \mathcal{N}(P_{t,i} - L_{dB,i} - \mu_i, \sigma_i^2). \tag{4.4}$$

To obtain the average SNR Γ_i, we subtract noise power $P_{n,i}$ from the mean received power strength as:

$$\Gamma_i = P_{t,i} - L_{dB,i} - \mu_i - P_{n,i}, \tag{4.5}$$

where $P_{n,i}$ is a noise power (dBm) calculated as:

$$P_{n,i} = 10\log_{10}(KTB_i) + 30, \tag{4.6}$$

where K is the Boltzmann constant, T is the ambient temperature and B_i is the bandwidth allocated to the ground node i. Γ_i is converted to the absolute value from decibel by:

$$\Gamma_i^{'} = 10^{\Gamma_i/10}. \tag{4.7}$$

Assuming channel reciprocity, the achievable data rate of the link between the UAV and the node i can be calculated as:

$$R_i = B_i \log_2(1 + \Gamma_i^{'}). \tag{4.8}$$

The shadow fading Ψ used in the (5.2) is represented as the Gaussian distribution where mean μ and variance σ^2 can take two values as $\Psi_{LOS} \sim \mathcal{N}(\mu_{LOS}, \sigma_{LOS}^2)$ and $\Psi_{NLOS} \sim$

$\mathcal{N}(\mu_{NLOS}, \sigma^2_{NLOS})$ in decibels for LOS and non-LOS (NLOS) cases, respectively. Here, μ_{LOS} and μ_{NLOS} are assumed to be constants for a given environment type (i.e., Suburban, Urban, Dense urban and High-rise urban), whereas σ^2_{LOS} and σ^2_{NLOS} can be modelled as:

$$\sigma^2_{LOS} = k_1 \exp(k_2 \theta_i), \quad \text{and} \tag{4.9}$$

$$\sigma^2_{NLOS} = g_1 \exp(g_2 \theta_i), \tag{4.10}$$

where θ_i is the elevation angle between the ground node i and the UAV, and k_1, k_2, g_1, and g_2 are constants dependent on the environment type.

Note that the GP approach (which will be explained in the next subsection) is used to predict Ψ_{LoS} and Ψ_{NLoS} while $P_{t,i}$ and $L_{dB,i}$ are assumed to be known a priori. While model used in this work assumes Gaussian distribution on the shadow fading, GP can be used to predict other distributions as well. The GP approach can predict errors and uncertainties in $P_{t,i}$ and $L_{dB,i}$ (such as wrongly guessed α parameter) as well, however, for the sake of simplicity, ease of comparison and prediction time those errors and uncertainties are not considered in this study. Besides, some other shadow fading components such as atmospheric effects could be easily incorporated into this model if necessary.

4.2.2 GP- based Channel Prediction

It is difficult to obtain the channel model accurately since it would require perfect knowledge of the dynamic communication environment of an urban area. In this work, instead of relying only on the model-based approach, we propose a GP-based learning approach with collected communication strength data to learn the effect of shadow fading to be combined with the known part of the model.

GP is one of powerful machine learning techniques, which can be described as:

$$f_{GP} \sim GP(m(\mathbf{x}), k(\mathbf{x}', \mathbf{x})) \tag{4.11}$$

where $m(\mathbf{x})$ is the mean function and $k(\mathbf{x}', \mathbf{x})$ is the covariance function between $\mathbf{x}'$ and $\mathbf{x}$. Here, $\mathbf{x}$ is an input vector from the training data consisting of the position of the UAV and ground nodes and $\mathbf{x}'$ is a matrix of an arbitrary UAV positions for which communication strength between air and ground is predicted. A training set with N_t observations is expressed as $\mathcal{D} = \{(\mathbf{x}_n, y_n) | n = 1, \cdots, N_t\} = \{\mathbf{X}, \mathbf{y}\}$ where $\mathbf{y}$ is a set of measured communication channel strength from all ground nodes.

Generally, covariance functions used for GP can be split into stationary and non-stationary. Stationary covariance functions are used when the original functions have certain patterns, while non-stationary covariance functions are used when the original function has no repeatable patterns. Fig. 4.5 shows the surface plot of the mean of shadow fading from (5.2) for four ground nodes with respect to changing positions of the UAV flying at 150 meters high. Since the spaces between areas are irregular with no repeatable patterns, a non-stationary covariance function is more appropriate for this problem. Here,

we use the squared exponential covariance function [113] with the spatially-varying length scale parameters as:

$$k(\mathbf{x}', \mathbf{x}) = \sigma_f^2 \left(\frac{a}{b}\right)^{\frac{n}{2}} \exp\left(\frac{||\mathbf{x} - \mathbf{x}'||^2}{b}\right), \tag{4.12}$$

where $a = 2l(\mathbf{x})l(\mathbf{x}')$, $b = 2l^2(\mathbf{x}) + l^2(\mathbf{x}')$ and n is a number of variables being correlated. $l(\cdot)$ represents the spatially varying length scale hyper parameters which are of the same form as the mean function. A constant mean function is defined as:

$$m(\mathbf{x}', \mathbf{x}) = c, \tag{4.13}$$

where c is a hyperparameter to be optimised.

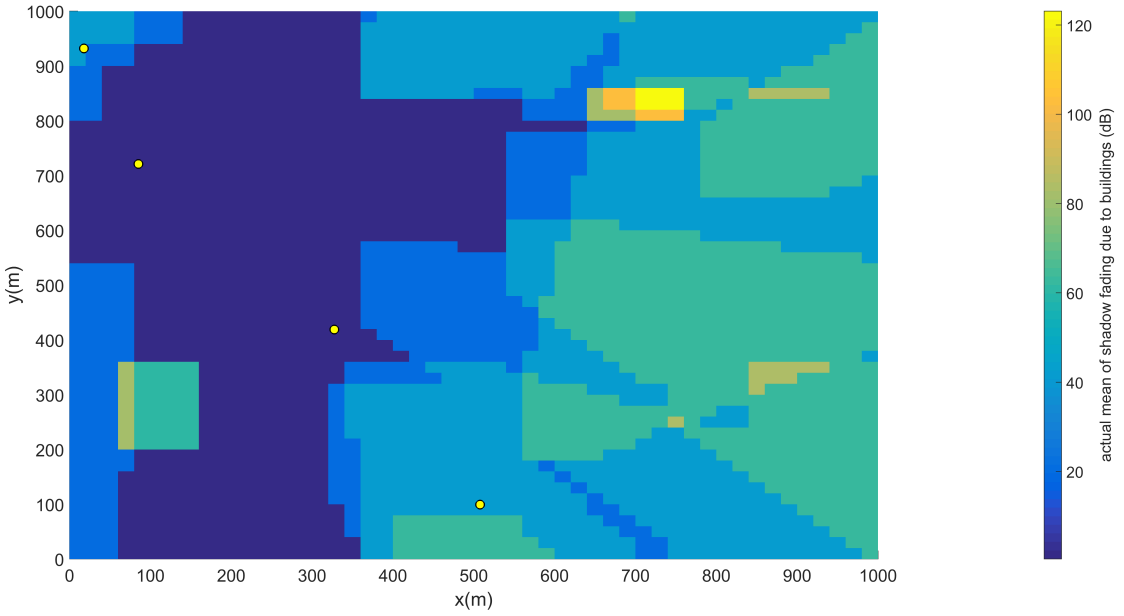


Figure 4.5: The sum of shadow fading for four ground nodes as seen by the UAV flying at 150 m height.

Given the GP model with the mean and covariance function, the fitness of this model to the training set $\mathcal{D}$ can be evaluated using the marginal likelihood conditioned on hyperparameters $\boldsymbol{\theta}$ (i.e. parameters to be trained in the mean and covariance function):

$$\begin{aligned} L(\boldsymbol{\theta}) = \log(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = -\frac{1}{2}\log|\boldsymbol{C_n}| \\ -\frac{1}{2}(\mathbf{y} - m(\mathbf{x}))^T(\boldsymbol{C_n})^{-1}(\mathbf{y} - m(\mathbf{x})) - \frac{N_t}{2}\log(2\pi) \end{aligned} \tag{4.14}$$

where $\boldsymbol{C_n} = \boldsymbol{\Sigma} + \overline{\sigma}_n^2 \mathbf{I}_{N_t}$, in which $\boldsymbol{\Sigma}$ denotes a set of covariance functions of $N_t \times N_t$ size with entries $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ for $i, j = 1, ..., N_t$, $\overline{\sigma}_n^2$ is the hyperparameter accounting for noisy data, and N_t is the number of observations.

Hyperparameters $\boldsymbol{\theta}$ can be tuned by maximizing the likelihood function (4.14) using the conjugate gradients method [12], as,

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}}(L(\boldsymbol{\theta})). \tag{4.15}$$

Given the training set $\mathcal{D}$ and the covariance function with the trained hyperparameters, the mean and variance at an arbitrary test position $\mathbf{x}'$ are computed as:

$$\mu_p(\mathbf{x}') = m(\mathbf{x}') + \boldsymbol{k}(\mathbf{x}, \mathbf{x}')^T (\boldsymbol{C_n}^{-1}(\boldsymbol{y} - \boldsymbol{m}(x')) \tag{4.16}$$

$$\sigma_p^2(\mathbf{x}') = k(\mathbf{x}', \mathbf{x}') - \boldsymbol{k}(\mathbf{x}, \mathbf{x}')^T (\boldsymbol{C_n})^{-1} \boldsymbol{k}(\mathbf{x}, \mathbf{x}'). \tag{4.17}$$

Therefore, the GP approach can learn and predict shadow fading parameters μ_i and σ_i^2 in (5.4) without a priori knowledge. Note that, although the shadow fading is normally modelled as the Gaussian distribution and GP as a regression tool assumes that the function output follows the Gaussian distribution, it does not require the actual fading to be Gaussian distributed.

Computation time for predicting μ_i in GP increases quadratically with number of data points collected, i.e. $O(N_t^2)$. This would be particular problematic for Scan+NMPC with GP approach, where number of datapoints for good prediction can be very high. To mitigate this approximation is used, where GP prediction is performed at pre-specified sparse grid points in xy-plane in the mission area after collecting the data from the scan, and during the relay mission, bilinear interpolation is used for prediction at arbitrary points. Such a grid is described by distance between each point d_g and limits $[x_{g,low}, x_{g,high}, y_{g,low}, y_{g,high}]$.

4.2.3 Prediction for Scan+NMPC With GP

Computation time for predicting μ_i in GP increases quadratically with number of data points collected, i.e. $O(N_t^2)$. To mitigate this, GP prediction is only performed at pre-specified grids in the mission area after collecting the data from the scan, and during the relay mission, bilinear interpolation is used for prediction at arbitrary points. Let $\mathbf{x_{gird}}$ be a matrix of (x, y) coordinates with length N_g so that points are spaced with the equal distance d_g and within $[x_{g,low}, x_{g,high}, y_{g,low}, y_{g,high}]$ limits, where N_g is the number of data points needed to fill the area within the limits. Then, the mean of the shadow fading at those points is predicted as:

$$\mu_g(\mathbf{x_{gird}}) = m(\mathbf{x_{gird}}) + \boldsymbol{k}(\mathbf{x}, \mathbf{x_{gird}})^T (\boldsymbol{C_n}^{-1}(\boldsymbol{y} - \boldsymbol{m}(\mathbf{x_{gird}})), \tag{4.18}$$

With (4.18), bilinear interpolation can be used to find the mean of the shadow fading at arbitrary locations as needed.

4.3 Numerical Simulation Results

In this section, the performance of the proposed GP based channel prediction and the UAV trajectory planning is evaluated via numerical simulation results. Four ground nodes are used and the flight height of the UAV is set to 150 m. Other parameters for communication and trajectory planning used for simulations are summarized in Table 5.4 and Table 5.5.

Table 4.2: Communication parameter

Parameter	Value	Unit
Transmission power (P_t)	40	dBm
Frequency f_c	2.0	GHz
Attenuation factor (α)	2.5	n/a
Communication Properties k_1, k_2, g_1, g_2	$11.25, 0.06, 32.17, 0.03$	n/a
mean LOS fading μ_{LOS}	0.1	n/a
mean NLOS fading μ_{NLOS}	21	n/a

Table 4.3: Simulation parameter

Parameter	Value	Unit
Actuator delay (τ_ω)	1/3	sec
UAV speed (v)	20	m/s
Heading rate constraint ($\omega_{min}, \omega_{max}$)	$(-0.4, 0.4)$	rad/s
Receding horizon step (N)	5	N/A
Horizon steps	$(0.5, 4.5, 5, 5, 5)$	sec
Maximum heading rate change (Δu_ω)	0.1	rad/s
Weighting factors (p_c, q_c, r_ω)	$(-1000, p_c/N, 1)$	N/A
Urban Enviroment Parameters (α_0, β_0, γ_0)	(0.1, 750, 8)	N/A
No. of data points for scan+NMPC with GP (per ground node)	320	n/a
Collection rate for NMPC with GP	2	sec
CEO sample size	8	n/a
CEO time step	6	n/a
CEO (α, ρ)	$(0.4, 0.2)$	n/a
Ambient temperature T	293	K
Bandwidth B_i	5	MHz
Grid parameters ($d_g, x_{g,low}, x_{g,high}, y_{g,low}, y_{g,high}$)	$(50, -300, 1300, -300, 1300)$	m

Seven methods are compared as described in Table 4.4 where two proposed methods using GP (scan+NMPC with GP and NMPC with GP) are already described in Section II.B; Three are model-based approaches and two are machine learning based approaches. The model-based approaches were devised to provide a baseline comparison and are defined as follows

- **NMPC with known map** assumes perfect knowledge of an urban map and the communication channel strength between the UAV and ground nodes. This method is to provide the baseline performance of the trajectory planner.
- **NMPC with LAP** is an approach which uses the LAP communication model de-

scribed in Section III.C with the NMPC-based trajectory planner. This is one of the recent techniques, which determines the LOS obstructions based on the probabilistic model.

- **Optimal point** provides the upper bound of the communication performance. The map is discretized into a grid of the predefined size and at each point of the grid, the communication performance is computed as if the UAV is there. Among those points, the optimal position which gives the best communication performance is chosen assuming that the UAV is able to hover in a single spot, even though this is not physically possible for the fixed-wing UAV considered in this paper.

Machine learning based approaches were used to verify performance of GP as a suitable technique to solve this problem. The machine learning based approaches are characterised below:

- **Scan+NMPC with SVM**. It is the same approach as Scan+NMPC with GP, however, it uses a support vector machine (SVM) for regression instead of GP to predict air to ground links. Here SVM with Gaussian kernel is used.
- **Scan+NMPC with NN**. It is the same approach as Scan+NMPC with GP, however, it uses Neural Network (NN) for regression instead of GP to predict air to ground links. Here Neural network with 10 hidden neurons is exploited with Bayesian Regularization learning method due to superior performance with noisy data.

Table 4.4: Air-to-ground link prediction schemes

	Path loss	Shadow fading
Scan+NMPC with GP		predicted by GP
NMPC with GP		predicted by GP
NMPC with known map	known	known
NMPC with LAP		predicted by (5.9)
Optimal point		known

4.3.1 The Comparison of CEO and GA

For trajectory planning, here, the same approach as described in chapter 3 is utilised. The main difference is that kalman filter is not used. With fixed ground node position, Kalman filter to predict trajectory is not necessary. Instead we assume UAV knows ground vehicle postion from start of mission. moreover Cross-Entropy Optimiser (CEO) was utilised instead of Genetic Algoritm. To evaluate the performance of the CEO algorithm, we compare it to our previous roulette-based GA as used in [9]. To conduct this comparison,

50 Monte Carlo simulations using NMPC with the known map and parameters from 5.5 are performed and the result is represented in Fig. 4.6. Each algorithm is tested with different iteration number and sample/population size which is reflected in number of calls to cost function on the y-axis. Inderectly this also allow comparison of the computational time as computing cost function is main contributor to computational time. CEO achieves a better solution using less cost function calls than GA does. For instance, CEO achieves the communication rate of 16.7 Mbps in just 40 calls and then increases to around 17 Mbps using 20 more calls. Meanwhile, GA reaches 16.2 Mbps in 40 calls and then steadily increases to about 16.7 Mbps, which takes about 200 calls to the cost function.

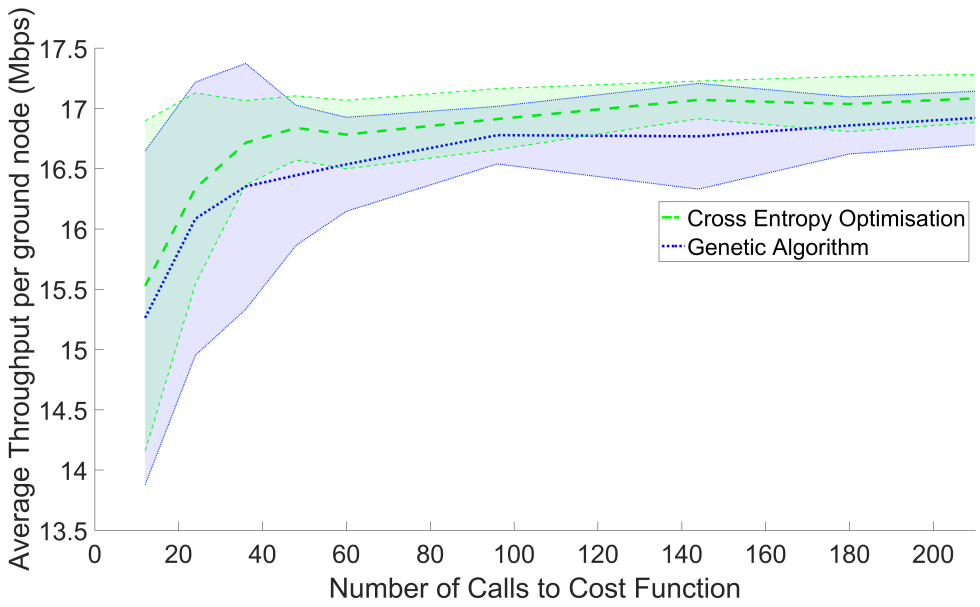


Figure 4.6: Comparison of the achievable communication rates of the proposed CEO algorithm with the GA algorithms. The shaded patch with boundaries represents the 1-σ standard deviation. The CEO algorithm shows much faster convergence and achieves higher data rate as compared to the genetic algorithm.

4.3.2 The Comparison of GP Performance for Different Scan Patters

To determine a good scan pattern comparison between three different scans were performed. Here, 3,5 and 7 UAV parallel flights (sweeps) are compared. Each sweep is a single line, which is parallel to the x-axis, in a path travelled by the UAV (see Fig. 4.3). At the end of each sweep, UAV turns to align for next sweep. The radius of the turn is smaller for a higher number of sweeps per scan flight. Performance of scan+NMPC with GP for a different number of sweeps is summarised in Fig. 4.7. It can be seen that performance with 7 sweeps is slightly better than 5 and 3 sweeps. This small difference is caused by a relatively small mission area used in this work. The number of sweeps is a parameter set by the user depending on the mission area, size and available time. In this work, 7 sweeps are used.

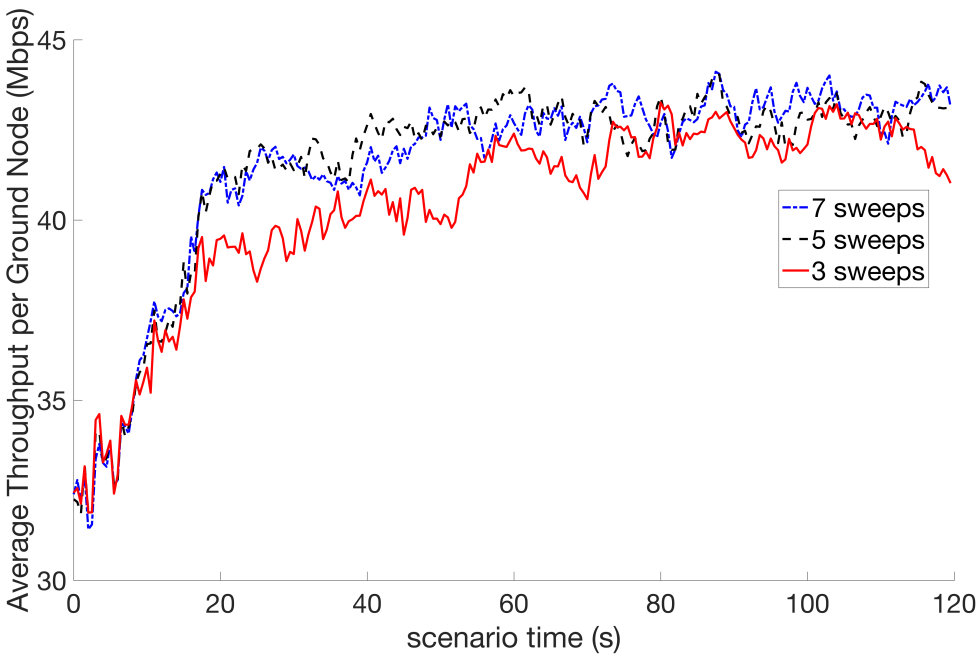


Figure 4.7: Comparison of performance of scan+NMPC with GP based on different number of sweeps. Each sweep is defined as a single parallel line flown by the UAV.

4.3.3 The quality of GP based Channel Prediction

To validate the performance of the GP-based channel prediction, Monte Carlo simulations over 30 scenarios are run on four different cases: i) 'no buildings', ii) 'around buildings' (four buildings around a ground node), iii) 'far buildings' (four buildings at least 200 m away from a ground node) and iv) 'all buildings' (using the suburban environment model). Each scenario consists of a single ground node for the reduced computation time.

Error for an Urban Map Using Scan+NMPC With GP

Fig. 4.8 shows how the root mean square RMS error in communication channel prediction changes with the number of data points collected from four scenarios. In this work, the RMS error is taken to be absolute difference between the mean of the shadow fading predicted by GP and the mean of the shadow fading defined in the (5.2). All four curves in Fig. 4.8 follow the exponential decrease in error until it reaches almost a constant error value. In general, results can be split into two groups (based on the magnitude of the error): i) 'no buildings' and 'far buildings' and ii) 'around buildings' and 'all buildings'.

The first group has much smaller overall error compared to the second group as scenarios for the first group are rather simple. Although the 'far buildings' case has the same amount of buildings as ' around buildings', buildings far away are not that important as they will rarely (if at all) cause LOS obstruction as illustrated in Fig. 4.9. The second group has a much higher overall error. This is because buildings are close to the ground node, affecting the LOS condition significantly and thus making the channel prediction

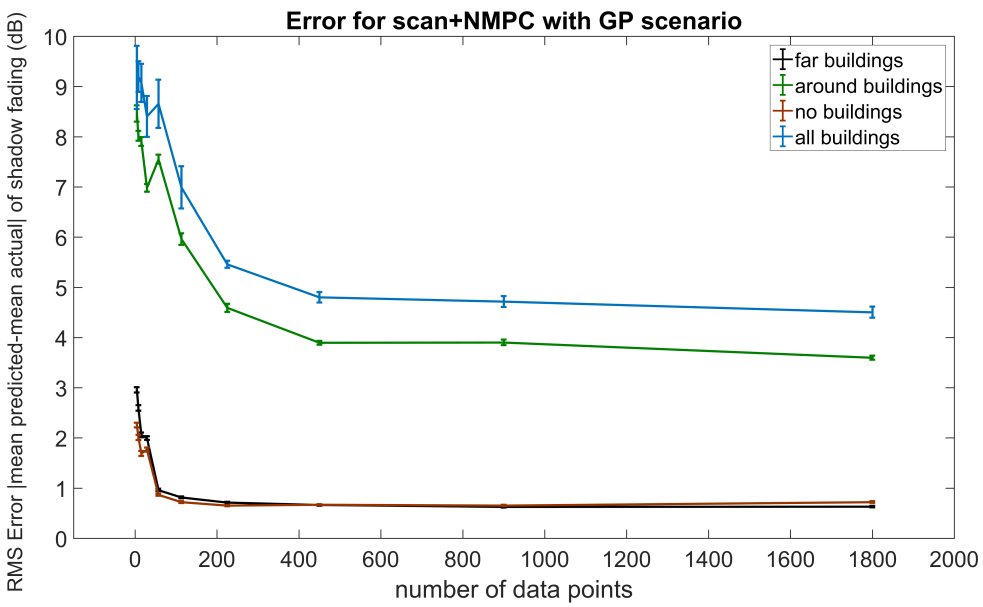


Figure 4.8: Comparison of errors in channel prediction using scan+NMPC with GP.

more complex. In a full city scenario (i.e. 'all buildings' case), there are more buildings close to the ground node, and thus prediction is even more difficult in this case. One sample error of predicted communication map from 'all buildings' case is shown in Fig. 4.10. With 200 data points used to create this figure, the prediction is relatively satisfactory, which means that GP is able to predict all major fading features affecting the ground node.

Error for UAV's Flight Trajectory Using NMPC With GP

Predicting the communication quality using NMPC with GP is more challenging as the UAV collects communication strength data over its flight trajectory only without pre-scanning. This means that there are many areas the UAV has no knowledge about the communication environment. Just increasing the number of data points collected over the travelled path does not provide more useful global information about the area which the UAV did not visit. Thus, the error over a full communication map does not yield a comparable situation to the scan+NMPC with GP case. For this reason, for NMPC with GP, the error over the travelled path only is computed, and the results are shown in Fig. 4.11. Note that the magnitude of the error is much lower than the previous scan case as this is computed only over the flown path, rather than full area. In this figure, the trends are similar to the scanning case, however, the variance is a lot higher. This high variance can be explained by the fact that the UAV can take a different path depending on earlier prediction. As the UAV collects communication data over its path in randomly generated scenarios, GP prediction will not be the same for different simulations runs.

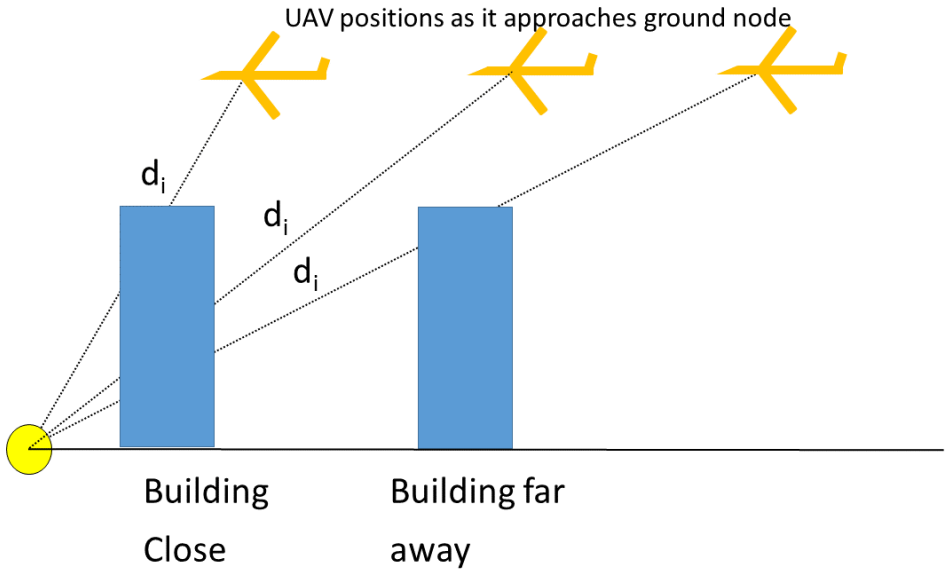


Figure 4.9: An illustration on the blockage effect of close and far buildings for air-to-ground communication.

Computational Time

To decide the optimal number of data points to be collected for a given scenario, not only the quality of prediction but also the computation time is of importance. It is worthwhile noting that the GP computational speed is mostly a function of i) the number of data points and ii) complexity of mean and covariance functions, however, it is not dependent on how complex the actual function is. Also for scan+NMPC with GP this represents only computational time to predict each point on the grid, rather than the time needed for each planning step. Computational time for each NMPC trajectory planner timestep for each of the approaches is discussed later on in table 4.7. Fig. 4.12 shows the computational time for the 'all buildings' scenario, generated using the desktop PC and repeating each scenario 50 times. It is worth noting that this figure is used to extract general trends since the actual performance will vary depending on the computer hardware used. The fitted curve is a quadratic function. As the computational time for data points rapidly gets unreasonably high and after 400 data points, improvement in prediction is negligible (as seen in Fig. 4.8), it is suggested that around 200 points per ground node would be an appropriate number.

4.3.4 Trajectory Planning Results

Next, we examine the average achievable communication rate as a result of different trajectory planning approaches. Each method runs over 50 Monte Carlo simulations in a suburban environment with the same starting point. For the Scan+NMPC with GP case, the UAV returns to its starting position after the scan flight.

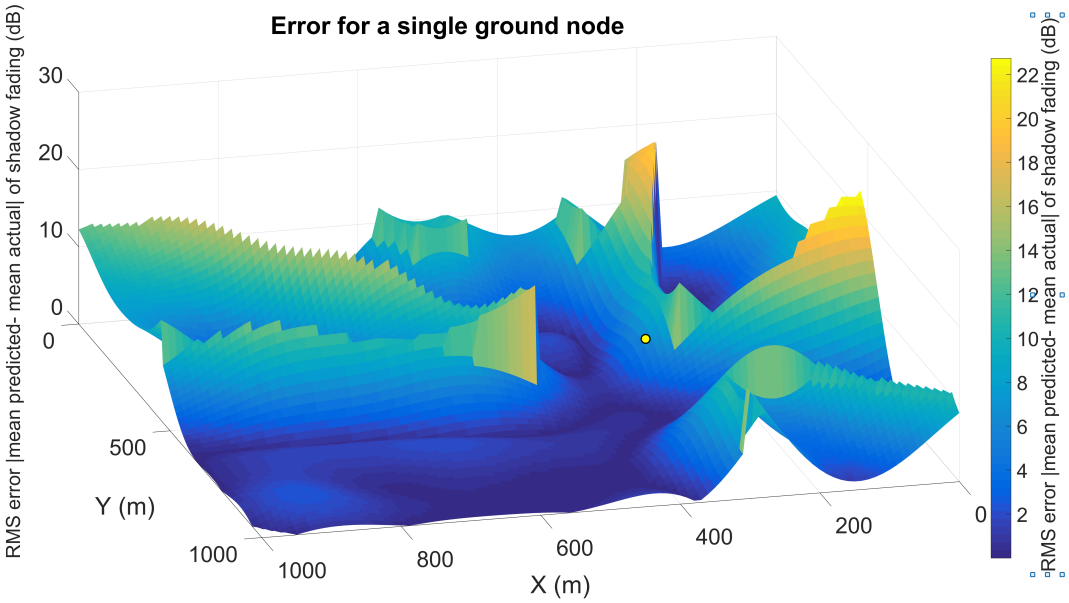


Figure 4.10: Error in channel prediction with the 'all buildings scenario' after scanning with 200 data points collected. The ground node position (yellow circle) is elevated to 20 meters to improve visibility.

Table 4.5: Average communication throughput over the final 20 seconds

Method	Average Throughput per Ground Node (Mbps)
Optimal Point	38.7
NMPC with known map (Perfect)	34.4
Scan+NMPC with GP	33.5
NMPC with GP	32.2
NMPC with LAP	31.8
LAP wrong assumptions	25.8

The averaged throughput across the whole trajectory and in the final 20 seconds are shown in Fig. 4.13 and Table 4.5, respectively for the suburban environment for comparison with model based approaches. Figure 4.15 and Table 4.6 shows simulation results for the dense urban environment for comparison with model based approaches. It can be seen that the optimal point approach sets the upper bound of the performance as expected. NMPC with known map has the best performance among NMPC-based approaches, closely followed by Scan+NMPC with GP and then NMPC with GP. Although NMPC with GP has worse performance than that of Scan+NMPC, it is beneficial to some extent as it can start the relay mission immediately without spending time on the scan flight. This means that, depending on user requirements (e.g. whether timely communication provision is important or not), different approaches could be preferred. It is worthwhile noting that

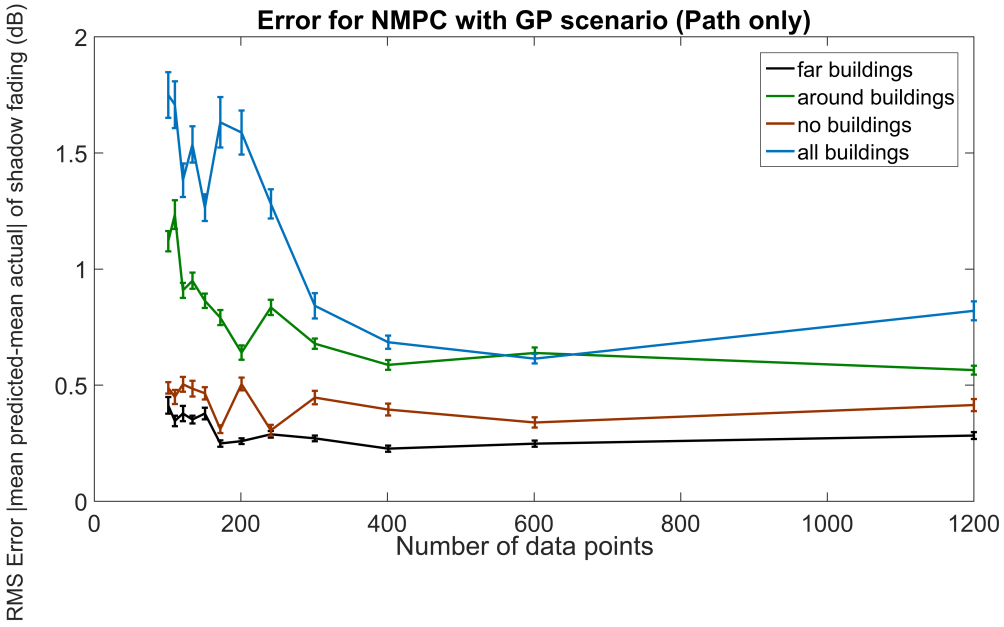


Figure 4.11: Comparison of errors in prediction from NMPC with GP. This error is calculated only over the path of UAV, rather than full area, as in NMPC with GP prediction outside path is very uncertain.

Table 4.6: Average communication throughput over the final 20 seconds

Method	Average Throughput per Ground Node (Mbps)
Optimal Point	35
NMPC with known map (Perfect)	23.4
Scan+NMPC with GP	22.3
NMPC with GP	20.7
NMPC with LAP	20.5
LAP wrong assumptions	19

Scan+NMPC with GP has the performance very similar to NMPC with known map (Perfect) in both suburban and dense urban environments, however, NMPC with GP is worse in suburban than dense urban environment.

Fig. 4.14 and Fig. 4.16 shows comparison between Scan+NMPC with GP, Scan+NMPC with SVM and Scan+NMPC with NN. First, it can be noted that Scan+NMPC with GP is the best for either of the tested urban environment. While not all machine learning techniques were tested, it suggests that GP is a good technique for this problem. The weak performance of the neural network suggests that applicability of NN regression is limited for this type of problems. Although the exact reason is uncertain at this stage, it is currently believed that NN struggles to learn good prediction with a limited number

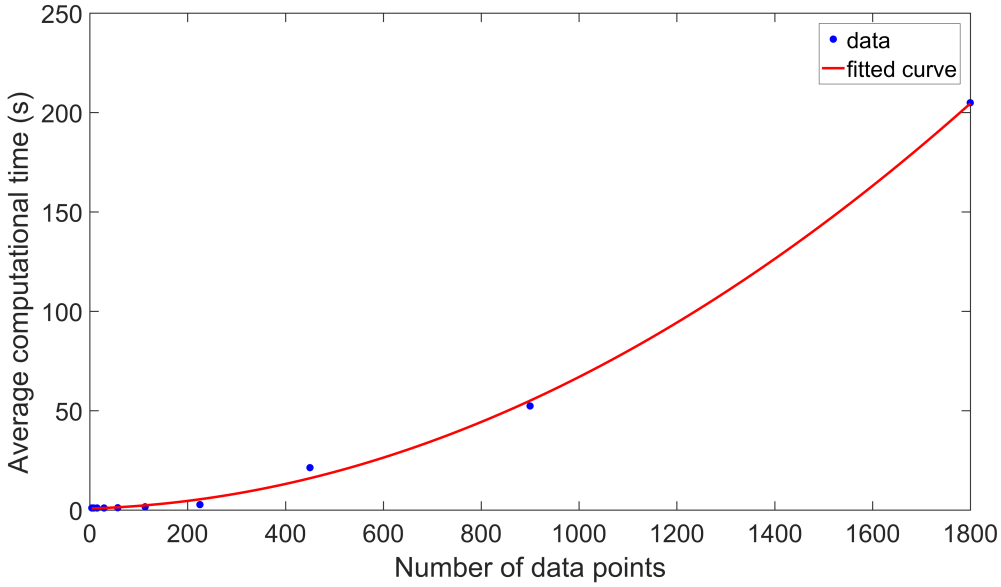


Figure 4.12: Computational time of GP-based channel prediction as a function of the number of data points collected for prediction.

and highly noisy data points.

Finally, comparison of the computational time per NMPC trajectory planner step is presented in the Table 4.7. For Scan+NMPC with GP time steps do not include initial time needed to create the grid from section III.C. This computation can be either offloaded to a very powerful computer or a cloud on the ground or it can be made when the UAV repositions itself to start the relay mission. Scan+NMPC with GP and NMPC with LAP have a similar computational load of about 0.7 seconds per time step. For NMPC with GP computational time is the highest, since the channel prediction using GP becomes slow as more data points are used.

Table 4.7: computational time per NMPC trajectory planner timestep (averaged over scenario time)

Method	Computational time per NMPC horizon step (s)
NMPC with known map (Perfect)	1.2
Scan+NMPC with GP	0.76
NMPC with GP	3.23
NMPC with LAP	0.77

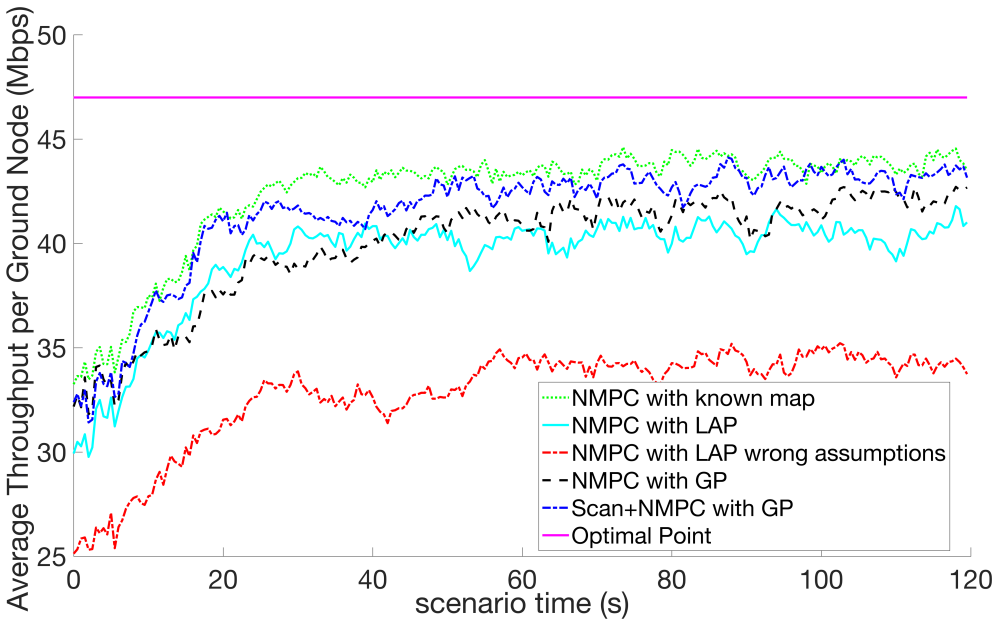


Figure 4.13: Communication performance comparison in a suburban environment between proposed approach and model based approaches.

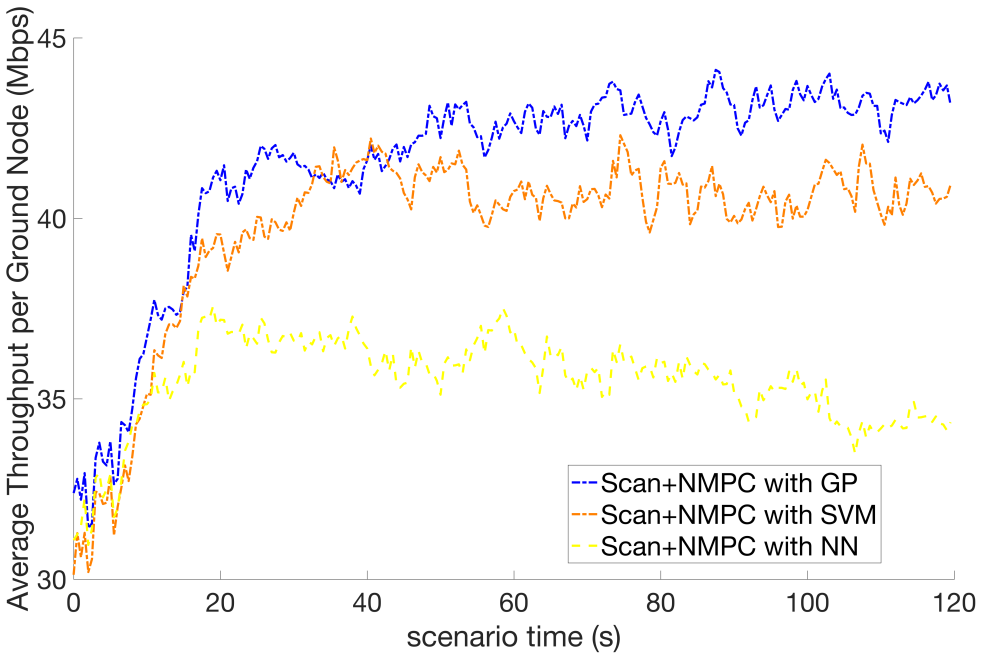


Figure 4.14: Communication performance comparison in a suburban environment between proposed approach three machine learning approaches.

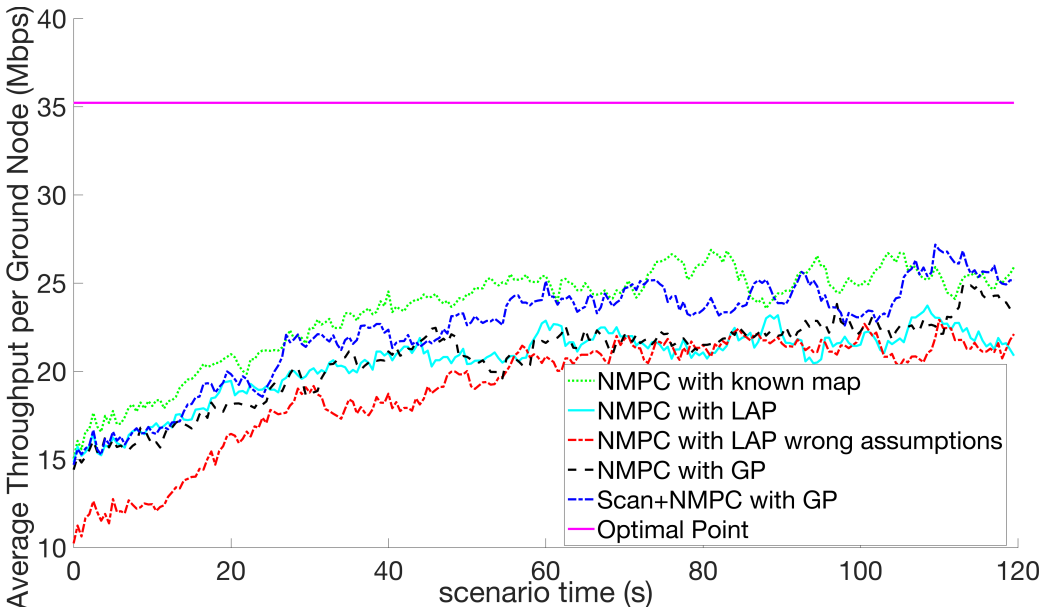


Figure 4.15: Communication performance comparison in a dense urban environment between proposed approach and model based approaches.

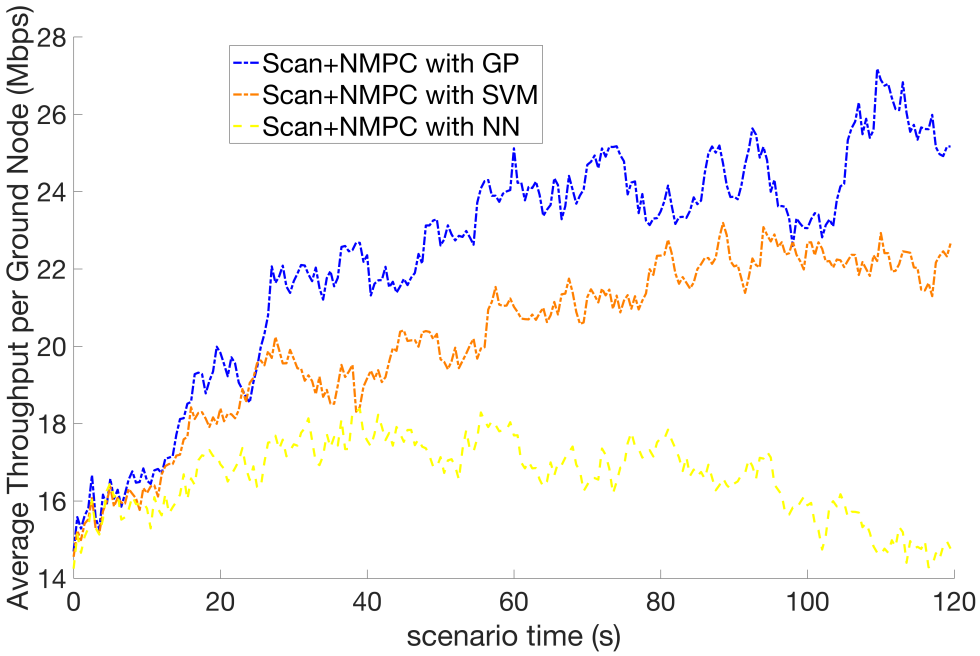


Figure 4.16: Communication performance comparison in a suburban environment between proposed approach three machine learning approaches.

4.4 Conclusions

The use of GP for channel prediction to support the UAV communication relay mission was presented in this paper considering complex urban environments. It was shown that creating a communication channel strength map using the GP approach before the mission with the scanning flight shows an advantage over creating the communication map while performing the mission (i.e. online) without scanning. However, collecting data during the mission in the online case allows the UAV to adapt to dynamic changes in the communication environment and adjust its position accordingly. Therefore, an appropriate method should be chosen depending on mission environments.

Chapter 5

Hybrid of Model-based and Measurement-based for Partially Known Urban Environments

In the previous chapter, we explored how UAVs communication relay performs with the usage of a model-based communication strength prediction. Here we propose a hybrid approach which combines learning based and model-based approaches to solve some of the limitations of the previous model. In particular for the model-based approach to work well, good knowledge of urban environment map (i.e. sizes and positions of buildings) was required. To solve this issue, a statistical model of the city with neural network machine learning is combined. To achieve that we use city model as defined by [46], which splits different urban environments into four different types: Suburban, Urban, Dense Urban and High-Rise Urban. In each of those environment types, communication strength between arbitrary points in the air and on the ground is predicted using a probability-based equation. This probabilist model is called low altitude platform (LAP). The equation is parameterised with different parameters, depending on the urban environment type. During a mission, unmanned aerial vehicle (UAV) collects signal strength readings and angle from ground nodes and uses a neural network to predict which one of the four environment UAV is in. Thus this approach does not need as much information about urban environment map as the previous one, just that it fits one of the pre-learned approaches. Similarly to the last chapter, mobile ground nodes could also be used as communication strength prediction is made with the probabilistic model once appropriate parameters have been predicted.

With the new approach to calculating communication performance, non-linear model predictive control (NMPC) trajectory planner from the previous chapter was extended to supplement heading optimisation with velocity optimisation. Heading limitation occurred due to highly computationally intensive communication performance prediction with the previous model-based approach. Particularly determining the building obstruction was

very computationally intensive. As a result, finding an optimal trajectory with both heading and velocity and the previous model-based approach was very slow. Probabilistic LAP model is significantly faster at predicting communication strength than previous model-based approach; thus more complex optimisation was introduced.

5.1 Problem Overview

5.1.1 Assumptions

A sample scenario considered in this work is illustrated in Fig. 5.5. In this scenario, there are a number of mobile ground nodes in an urban environment. UAVs fly in a way to assist mobile ground nodes with their wireless communication equipment for better communication performance.

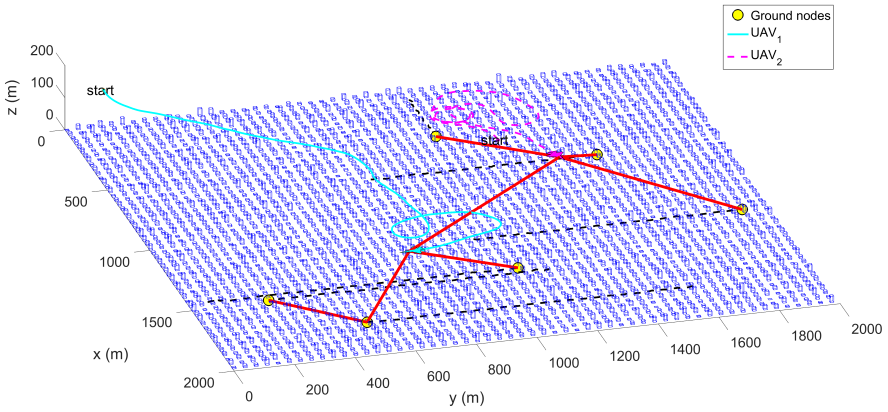


Figure 5.1: Illustration of the communication relay scenario.

The assumptions made in this work are stated as follows: i) an urban environment can be modelled as either of the four types: suburban, urban, dense urban and high-rise urban depending on the density and height of buildings; ii) the communication channel model consisting of path loss, transmitted power and shadow fading components is known empirically for each environment type, however the environment type for a given scenario is unknown; iii) ground nodes know their positions by using GPS and are able to share their current positions with UAVs using wireless network, but their future paths are unknown to the UAV; and iv) the position and the shape of buildings are unknown.

Assumption (iii) is vital for UAV to be able to predict communication strength correctly. If the ground node cannot communicate its position with the UAV, it would not be considered during optimisation. To mitigate this issue following approach could be used. First, UAV could predict where the ground node is based on apriori knowledge UAV has. Then UAV could search to regain communication with the lost ground node. However, this would result in temporarily worse communication improvement between other ground nodes. This is not covered in this work and remains an interesting future direction.

5.1.2 Overview of the Learning-Based Channel Prediction Algorithm

Figure 5.2 shows the overview of the optimal trajectory planning process with the hybrid channel prediction approach. To plan the trajectory of communication relay UAVs, first cross entropy optimiser (CEO) [103] randomly generates a set of possible trajectories. For each trajectory, the LAP model is used to compute the communication quality of the networked team. To use the correct LAP model, UAVs collect a pair of signal strength and elevation angle between the UAV and ground nodes (hereafter such a pair is called signal strength-angle pairs). With this data, the NN predictor is used to predict the current urban environment type. Until convergence is reached, the CEO algorithm changes the candidate trajectories and once the convergence criterion is satisfied, the best trajectory is sent to UAVs for execution. Note that this entire process is periodically performed to cope with the dynamic environment.

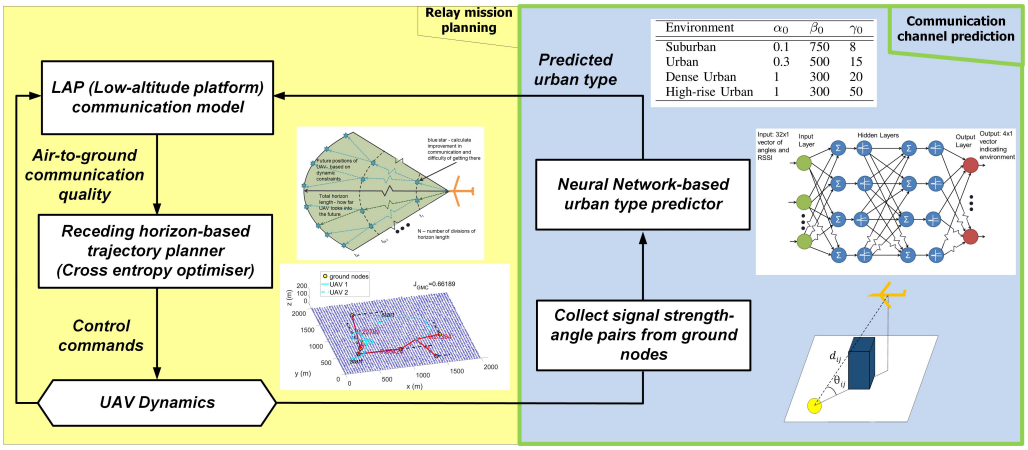


Figure 5.2: Overview of the optimal trajectory planning process for communication relay UAVs.

5.2 Learning-Based Communication Channel Prediction

5.2.1 Air-to-Ground Channel Modelling

The communication channel model used in this work is based on the multiple ray tracing simulation with four different types of urban environments: i) Suburban, ii) Urban, iii) Dense urban and iv) High-rise urban [46, 112]. Urban environment types are defined by three parameters: α_0, β_0 and γ_0, where α_0 is the ratio of the built-up land area to the total land area, β_0 is the mean number of buildings per unit area (buildings/km^2) and γ_0 is a scale factor that describes the building heights in the Rayleigh probability density function as:

$$P(h) = \frac{h}{\gamma_0^2} \exp\left(\frac{-h^2}{2\gamma_0^2}\right), \tag{5.1}$$

where h is the building height in meters. The values of the parameters for each urban environment are summarised in Table 5.1. To fully define a city, its layout is defined as a standard city layout [46] as shown in Fig. 5.3. The benefit of using this city model is easy generation of multiple cities with different heights for Monte Carlo simulations.

Table 5.1: different city environment parameters

Environment	α_0	β_0	γ_0
Suburban	0.1	750	8
Urban	0.3	500	15
Dense Urban	1	300	20
High-rise Urban	1	300	50

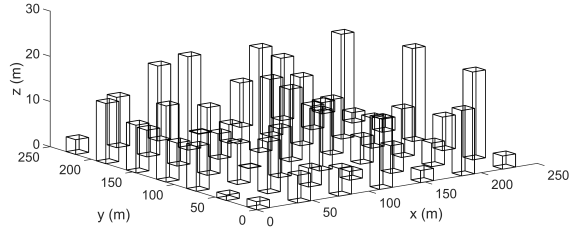


Figure 5.3: A sample city generated with parameters: $\alpha_0 = 0.1$, $\beta_0 = 750$ and $\gamma_0 = 8$.

With the above city model, the communication channel model can be defined as [45]:

$$P_{r,ij} = P_{t,j} - L_{dB,ij} - \Psi_{ij} \tag{5.2}$$

where $P_{r,ij}$ (dBm) is the received signal power strength of node i from node j. $P_{t,j}$ (dBm) is the transmitted power by node j, $L_{dB,ij}$ represents the free space path loss between nodes i and j, and Ψ_{ij} is the shadow fading component accounting for diffraction and multipath fading. Ψ is assumed to be a Gaussian random variable defined as $\Psi_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma_{ij}^2)$, where μ_{ij} and σ_{ij}^2 are the mean and variance parameters. $L_{dB,ij}$ can be represented as:

$$L_{dB,ij} = 10\alpha \log_{10}\left(\frac{4\pi f_c d_{ij}}{c}\right), \tag{5.3}$$

where f_c is the central frequency, α is the path loss exponent, d_{ij} is the distance between nodes i and j, and c is the speed of light. Note that $P_{r,ij}$ follows the Gaussian distribution and can be expressed as:

$$P_{r,ij} \sim \mathcal{N}(P_{t,ij} - L_{dB,ij} - \mu_{ij}, \sigma_{ij}^2). \tag{5.4}$$

To be more specific, the shadow fading Ψ_{ij} can take either of two distributions: $\Psi_{LOS,ij} \sim \mathcal{N}(\mu_{LOS,,ij}, \sigma_{LOS,ij}^2)$ and $\Psi_{NLOS,ij} \sim \mathcal{N}(\mu_{NLOS,ij}, \sigma_{NLOS,ij}^2)$ for cases of line-of-sight (LOS) and non-LOS (NLOS), respectively. In this work, $\mu_{LOS,ij}$ and $\mu_{NLOS,ij}$ are assumed to be

known and constant for a given urban environment type, whereas $\sigma^2_{LOS,ij}$ and $\sigma^2_{NLOS,ij}$ can be modelled as:

$$\sigma^2_{LOS} = k_1 \exp(k_2 \theta_{ij}), \quad \text{and} \tag{5.5}$$

$$\sigma^2_{NLOS} = g_1 \exp(g_2 \theta_{ij}), \tag{5.6}$$

where θ_{ij} is the elevation angle of the UAV with respect to the ground, and k_1, k_2, g_1 and g_2 are positive constants dependent on the urban environment type.

5.2.2 LAP Communication Model

To allow the UAV to predict the communication strength of ground nodes during trajectory planning, choice between $\Psi_{LOS,ij}$ and $\Psi_{NLOS,ij}$ needs to be made. For this, we use the method from [11], which predicts the probability of LOS (or NLOS) occurrence based on the elevation angle and urban environment type specified by α_0, β_0 and γ_0. In this approach, the LOS probability between two nodes can be determined as:

$$P(LOS) = \prod_{i=0}^{\overline{m}} [1 - \exp\left(-\frac{\left[h_{TX} - \frac{(i+1)(h_{TX}-h_{RX})}{\overline{m}+1}\right]^2}{2\gamma_0^2}\right)] \tag{5.7}$$

where $\overline{m} = \text{floor}(d_{ij}\sqrt{\alpha_0\beta_0} - 1)$, r is the ground distance between the UAV and the ground node, h_{TX} is the height of the UAV, and h_{RX} is the height of the ground node. Using Eq. (5.7) directly might incur a high computational time. This issue can be mitigated by using a sigmoid function approximation as:

$$P(LOS, \theta_{ij}) = \frac{1}{1 + \overline{a}\exp(-\overline{b}[\theta_{ij} - \overline{a}])}, \tag{5.8}$$

where θ_{ij} is the elevation angle between the air and ground nodes and $\overline{a}$ and $\overline{b}$ are parameters of the S-curve dependent on α_0, β_0 and γ_0 [11]. Then, the prediction of Ψ can be made by defining a LOS probability threshold P_t as:

$$\Psi_{ij} = \begin{cases} \Psi_{LOS,ij}, & \text{if } P(LOS, \theta_{ij}) > P_t, \\ \Psi_{NLOS,ij}, & \text{otherwise.} \end{cases} \tag{5.9}$$

5.2.3 Neural Network Channel Prediction

In this work, the input consists of 16 pairs of signal strength and elevation angles and the output is the urban type. The choice of that number is described later on in this section. Note that during the trajectory planning process, the urban type prediction by NN is periodically performed with most recently collected data. In case of multiple UAVs are involved, each UAV is assumed to be able to make its own prediction based on the reading it has obtained, and then share its prediction with the rest of the group for improving

the prediction accuracy. The output from NN is a vector $\mathbf{y} \in \mathbb{R}^{1\times 4}$ where each element is 0 or 1 and each element being responsible for one of the four environment types. For instance, $y_i = 1$ implies i-th urban environment type is predicted. NN was trained on 12 randomly generated scenarios (3 scenarios per each urban environment) with 12 stationary ground vehicles. The UAV performed the back and forth search pattern and collected 2172 signal strength and elevation angle pairs per scenario. In the case of this learning whether statrionary of mobile ground nodes were used didn't matter as the prediction only depends on angle between air and ground vehicles.

As introduced in 2.4.4 Neural network requires a fixed number of inputs. However number of obtained RSSI-angle pairs is unknown prior to the mission start as it depends on the data collection rate, mission length and a number of ground nodes. To mitigate this constraint we limit the number of RSSI-angle pairs NN is processing at any given timestep. The limit imposed needs to fulfil following requirements: i) be able to make prediction relatively quickly to allow prediction with minimal data and ii) make prediction accurate enough to allow completing of UAV mission. The choice of balance between those two metrics is rather arbitrary, in here we assume that to fulfil those requirements Neural Network should make a prediction with less than 32 of angle-RSSI pairs with 70% accuracy. To test how many RSSI-angle pairs is sufficient we tested 4, 8, 16, 32 and 64 pairs and compared the accuracy. Results of this test are summarised in table 5.2. It can be seen that from 4 to 16 angle-RSSI pairs accuracy is increasing steadily by about 10% for each increase in data points, while between 32 and 64 pairs increase is much smaller by 3% only. Thus 16 is chosen as a compromise between accuracy and number of data points. 4 and 8 do not offer desirable accuracy, while 32 and 64 do not offer a significant boost in accuracy to warrant an increase in datapoints requirements.

Table 5.2: prediction quality with different angle-RSSI pair

number of angle-RSSI pairs	Accuracy
4	50.1%
8	60.8%
16	70.9%
32	73%
64	76%

To investigate the performance of NN, a confusion matrix is used as shown in Fig. 5.4. The confusion matrix is a widely used technique for reporting quality of learning. It should be read as follows In the matrix, bold numbers are numbers of samples classified as given environment type, the percentage is the same number expressed as the percentage of the total sample. diagonal terms (in green) are the number of correctly classified environment

type along with the percentage of the total sample. The red off-diagonals elements are the number of target environments incorrectly classified. The edges of the matrix in grey are total percentages of correctly classified (green) and incorrectly classified (red) environment type. Each row is an output environment class as predicted by NN while columns are the target class i.e. what the prediction should be if NN made a correct prediction. Looking at single row will provide information on how many of given class was classified correctly (green box) and how many incorrectly and as what class they were classified (red box). This figure shows that the majority of errors occurs between classes which are next to each other. This is expected, environments which are close to each other are very similar in terms of probability of blocking signal and effect on RSSI. Thus Neural network struggles to make a correct prediction.

Figure 5.4: Confusion matrix for Neural Network

Finally, the proposed NN approach is compared against other machine learning techniques: k-nearest neighbour, support vector machine and decision trees. All of those approaches were trained on the same data set as NN. As it can be seen from Table 5.3,

NN is at least 9% more accurate than any other method.

Table 5.3: Accuracy of methods

Method	Accuracy
NN	70.9%
support vector machine	62.2 %
k-neareast neightbours	61.9 %
Decision tree	46 %

5.2.4 Communication Performance Metrics

To compute performance metrics which can be used to plan an optimal trajectory for the UAV, two steps are necessary: i) conversion of $P_{r,ij}$ to the probability of successful communication and ii) computation of the global message connectivity (GMC) [8] to define how much the UAV improves the communication performance of the networked group. To define probability, the signal to noise ratio (SNR) needs to be defined by subtracting the noise power $P_{n,ij}$ from the mean received power strength $P_{r,ij}$ in Eq. (5.4) as:

$$\Gamma_{ij} \sim \mathcal{N}\big(\mu_{\gamma,ij}, \sigma_{ij}^2\big). \tag{5.10}$$

where $\mu_{\gamma,ij} = P_{t,ij} - L_{dB,ij} - \mu_{ij} - P_{n,ij}$ and $P_{n,ij}$ is a noise power (dBm) calculated as:

$$P_{n,ij} = 10\log_{10}(KTB_{ij}) + 30, \tag{5.11}$$

where K is a Boltzman constant, T is ambient temperature, and B_{ij} is a bandwidth. Following integration steps from [114], the probability of sucesful communication is expressed as:

$$P_{s,ij}(\Gamma_{ij} \geq \gamma) = Q\left(\frac{\mu_{\gamma,ij} - \gamma}{\sigma_{ij}}\right), \tag{5.12}$$

where γ is the required minimum SNR defined by the user and Q is the complementary error function.

Thus, based on the predicted urban environment and its corresponding μ and σ^2, the UAV can make predictions of how it will improve communication in the group. It is worth noting that some other metrics could be used instead of GMC. One example could be the modified global message connectivity proposed in [9].

5.3 Receding Horizon-Based Online Trajectory Planning

Here we extend previous UAV model to allow change in both velocity and direction rather than direction only. While this should result in better answers, more optimiser steps

would be required to reach a good answer. However, Neural Network has a signficantly smaller computational time than any other presented approach so far (computational time needed) so it was deemed as a good compromise.

5.3.1 UAV Model

The model described in section 3.3.2 is adapted as:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{v} \\ \dot{\omega} \end{pmatrix} = f(\mathbf{x}, u) = \begin{pmatrix} vcos\psi \\ vsin\psi \\ \omega \\ -\frac{1}{\tau_v}v + \frac{1}{\tau_v}u_v \\ -\frac{1}{\tau_\omega}\omega + \frac{1}{\tau_\omega}u_\omega \end{pmatrix} \tag{5.13}$$

where $\mathbf{x} = \begin{pmatrix} x & y & \psi & v & \omega \end{pmatrix}^T$ are the inertial position, heading, speed and yaw rate of the UAV, respectively. τ_v and τ_ω are time constants account for actuator response delay, which can be determined experimentally for a given UAV model. u_ω is a command input in the form of the turning rate and u_v is a command input in the form of velocity. It is worth noting that v is constrained by maximum and minimum velocity v_{min}, v_{max} and ω is constrained by maximum and minimum heading rate $\omega_{min}, \omega_{max}$ due to physical limitations of the fixed-wing UAV. Similarly to previous chapter Eq. (5.13) in the receding horizon framework is discretised using Euler integration as:

$$\mathbf{x}_{k+1} = f_d(\mathbf{x}_k, u_k) = \mathbf{x}_k + T_s f(\mathbf{x}_k, u_k) \tag{5.14}$$

where $\mathbf{x}_k = \begin{pmatrix} x_k & y_k & \psi_k & v_k & \omega_k \end{pmatrix}^T$ and T_s is a sampling time.

To predict trajectory of ground vehicle we use same kalman filter as described in 3.3.2.

5.3.2 Receding Horizon Problem Formulation

Similary to section 5.3.1. the receding horizon formulation needed to be adapated to account for addtiional velocity term. The receding horizon-based trajectory planning is formulated to find the optimal set of command inputs $U^i = \begin{pmatrix} U^i_v, U^i_\omega \end{pmatrix}^T$ where $U^i_v = \begin{pmatrix} u^i_{v,0}, u^i_{v1}, \ldots, u^i_{v,N-1} \end{pmatrix}$, $U^i_\omega = \begin{pmatrix} u^i_{\omega,0}, u^i_{\omega,1}, \ldots, u^i_{\omega,N-1} \end{pmatrix}$ for i-th UAV, which minimises the following performance index J:

$$\min_U J(U) \triangleq \Phi(\bar{x}_N, \bar{x}^g_N) + \sum_{k=0}^{N-1} L(\bar{x}_k, \bar{x}^g_k, u_{\omega,k}, u_{v,k}) \tag{5.15}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = f_d(\mathbf{x}_k, u_{\omega,k}, u_{v,k}), \tag{5.16}$$

$$\omega_{min} \leq u_{\omega,k} \leq \omega_{max}, \tag{5.17}$$

$$|u_{\omega,k} - u_{\omega,k-1}| \in \{0, \Delta u_\omega\} \tag{5.18}$$

$$v_{min} \leq u_{v,k} \leq v_{max}, \tag{5.19}$$

$$|u_{v,k} - u_{v,k-1}| \in \{0, \Delta u_v\}, \tag{5.20}$$

where:

$$\Phi(\bar{\mathbf{x}}_N, \bar{\mathbf{x}}_N^g) \triangleq p_c \frac{1}{J_{GMC}(\bar{\mathbf{x}}_N^{pos}, \bar{\mathbf{x}}_N^{g,pos})}, \tag{5.21}$$

$$L(\bar{\mathbf{x}}_N, \bar{\mathbf{x}}_N^g, u_{\omega,k}, u_{v,k}) \triangleq \frac{1}{2}\left[q_c \frac{1}{J_{GMC}(\bar{\mathbf{x}}_k^{pos}, \bar{\mathbf{x}}_k^{g,pos})} + \right. \\ \left. + r_\omega \left(\frac{u_{\omega,k}}{\omega_{max}}\right) + r_v \left(\frac{u_{v,k}}{v_{max}}\right)\right]^2 \tag{5.22}$$

All terms retain their previous meaning unless stated otherwise. The major change come from addtional constraint in eqs. (5.17)∼(5.20). The third and fourth constraints are added to limit maximum nad minimum velocity of the UAV and discretises the problem to ensure velocity difference is not bigger than $\{0, \Delta u_v\}$. To compute optimal commands input based on performance index, Cross Entropy optimiser is employed.

5.4 Numerical Simulation Results

In this section, the performance of the proposed approach is investigated through a set of 48 Monte Carlo simulations. Parameters of simulations and wireless communication are given in Table 5.5 and 5.4, respectively. The proposed approach, receding horizon(RH)-based trajectory planning with the hybrid channel prediction (termed as RH+NN with the LAP model), is compared with three other approaches. First, the receding horizon-based trajectory planning with the known map is an approach where building sizes, positions and effects on the communication quality are fully known. It allows the UAV to make the perfect prediction of the communication performance for arbitrary points. RH with the known LAP model is similar to the proposed approach in that it uses Eq. (5.8) to predict the probability of being in LOS, but α_0, β_0 and γ_0 are assumed to be known a priori. RH with the incorrect LAP model is an approach where incorrect α_0, β_0 and γ_0 are used. For the MST construction during the prediction stage, connections between ground nodes are assumed to be in NLOS at all times.

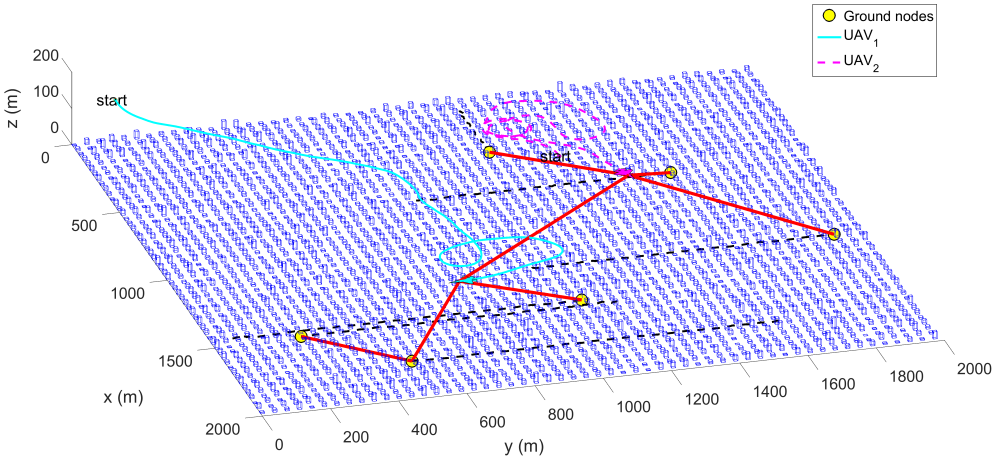


Figure 5.5: A sample urban scenario with six ground nodes, two relay UAVs and buildings with different heights.

Table 5.4: Communication parameter

Parameter	Value	Unit
Transmission power (P_t)	40	dBm
Frequency f_c	2.0	GHz
Attenuation factor (α)	2.5	n/a
Communication Properties (k_1, k_2, g_1, g_2)	(11.25, 0.06, 32.17, 0.03)	n/a
mean LOS fading μ_{LOS}	0.1	n/a
mean NLOS fading μ_{NLOS}	21	n/a
Bandwidth B_{ij}	5	MHz

Fig. 5.6 shows a sample scenario from Monte Carlo simulations with trajectories using aforementioned four methods. In this scenario, issues with RH with the incorrect LAP model (i.e. using incorrectly-guessed parameters) approach are apparent. Within the time frame of the scenario, other three approaches send one UAV to around $(x, y) = (1400, 500)$ and the other to $(x, y) = (400, 700)$ to serve as relay nodes there. However, RH with the incorrect LAP model makes both UAVs to stay around the $(x, y) = (400, 700)$ where they can only help three ground nodes. It is likely that RH with the incorrect LAP computed that there is no viable position where the relay UAV could help by flying to other position. Such an erroneous decision is made as RH with the incorrect LAP uses a city with buildings which are much higher and bigger than they actually are.

Figure 5.7 shows the comparison of the performance of four approaches. It can be seen that RH with the known map has the best performance. This is closely followed by RH with the known LAP model and RH+NN with the LAP model. RH+NN with the LAP model is slightly worse than RH with the known LAP model. The reason for this discrepancy is that there exists cases where NN makes the wrong prediction for the urban environment type, thus lowering the performance of the approach. Finally, RH with the incorrect LAP model shows the worst performance as the UAV would never reach the optimal position due to incorrect parameters.

The quality of urban type prediction using NN is visualised on 3 randomly selected scenarios from Monte Carlo simulations in Fig. 5.8. With each time step, the UAV collects more communication data and makes a new prediction about its current environment. For

Table 5.5: Simulation parameter

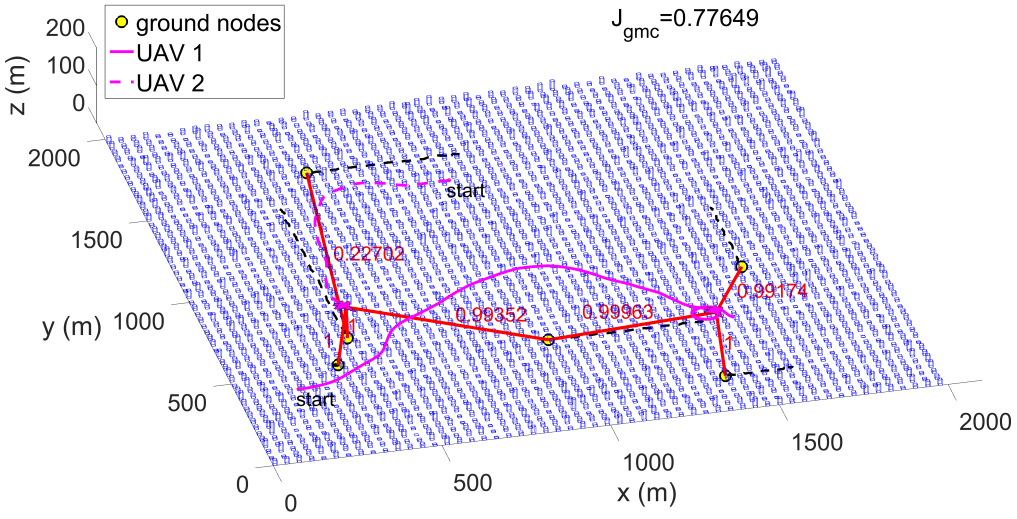
Parameter	Value	Unit
Actuator delay (τ_ω, τ_v)	(1/3,1/3)	sec
speed constraints (v_{min}, v_{max})	$(10, 30)$	m/s
Heading rate constraint $(\omega_{min}, \omega_{max})$	$(-0.4, 0.4)$	rad/s
Receding horizon step (N)	5	N/A
Horizon steps	$(0.5, 4.5, 5, 5, 5)$	sec
Maximum heading rate change (Δu_ω)	0.1	rad/s
Maximum velocity change (Δu_v)	5	m/s
Weighting factors (p_c, q_c, r_ω)	$(-1000, p_c/N, 1)$	N/A
Urban Enviroment Parameters (α_0, β_0, γ_0)	(0.1, 750, 8)	N/A
Urban Enviroment Parameters (α_0, β_0, γ_0) for NMPC with incorrect LAP model	(1 , 300, 20)	N/A
Ambient temperature T	293	K
Number of UAVs n	2	N/A
number of ground nodes m	6	N/A

scenario id 6, predictions are good and consistent throughout the simulation. Predictions for scenario id 22 and id 13 start off as incorrect, but about half way through the prediction converges to the correct urban environment type.

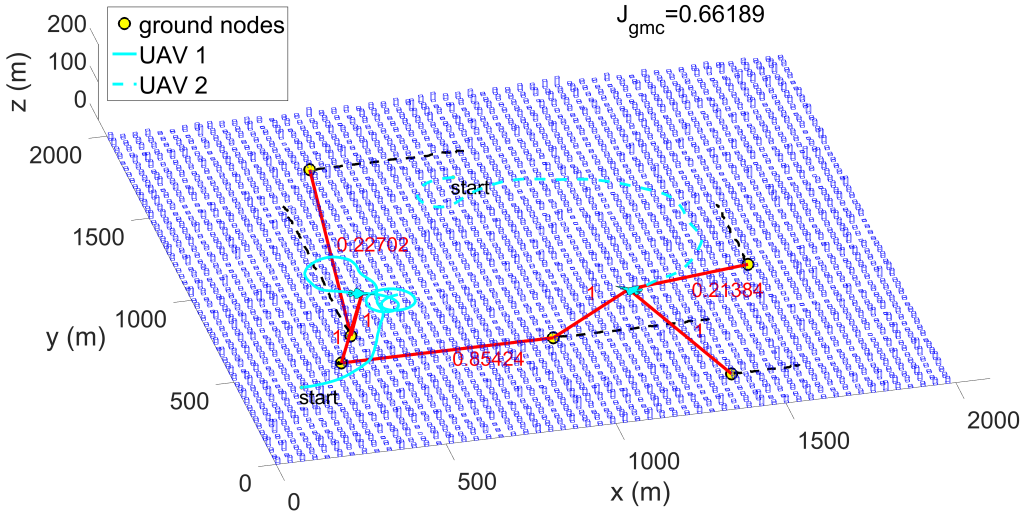
The performance with the increasing number of UAVs is investigated in Fig. 5.9 where the resuls are obtained by averaging from the final 20 seconds of 48 Monte Carlo simulations. With no communication relay UAV in the area, there is a very low probability of establishing a successful communication for the networked nodes of only 30%. With one UAV, there is a significant increase in the probability of successful communication while more UAVs provides higher probability but with lower increasing rate.

Finally, Fig. 5.5 shows a sample run from one of the 48 Monte carlo simulations. In this scenario UAV 2 starts very close to ground nodes while UAV 1 needs to travel across before it can fulfil its role as communication relay. It is also worth noting that UAV's my temporarily circle in place, but most of the time they try to move to improve wireless communication the best. The proposed trajectory planning guides the UAV to improve communciation strenght between the ground nodes as much as possible.

To complete the discussion we make a comparison of predicted trajectories by each



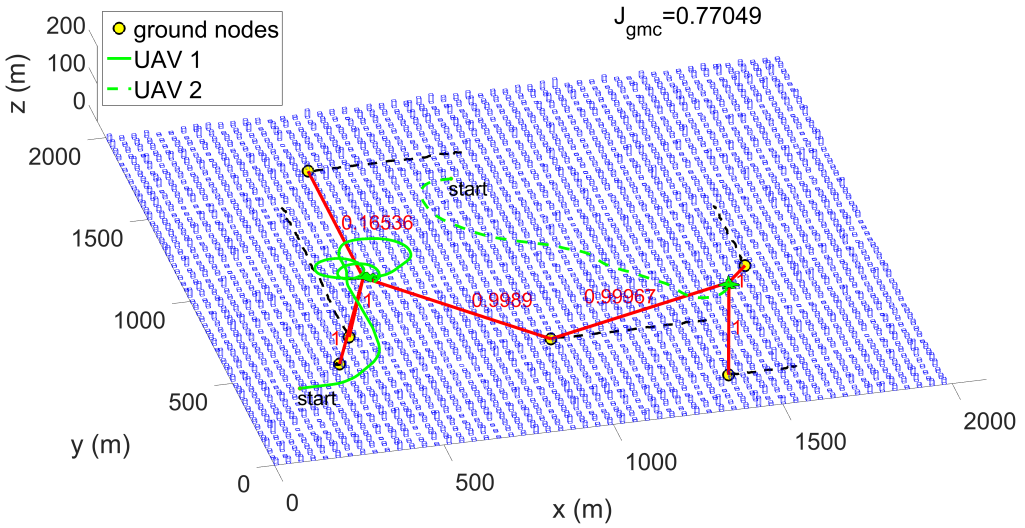
(a) RH with the known map



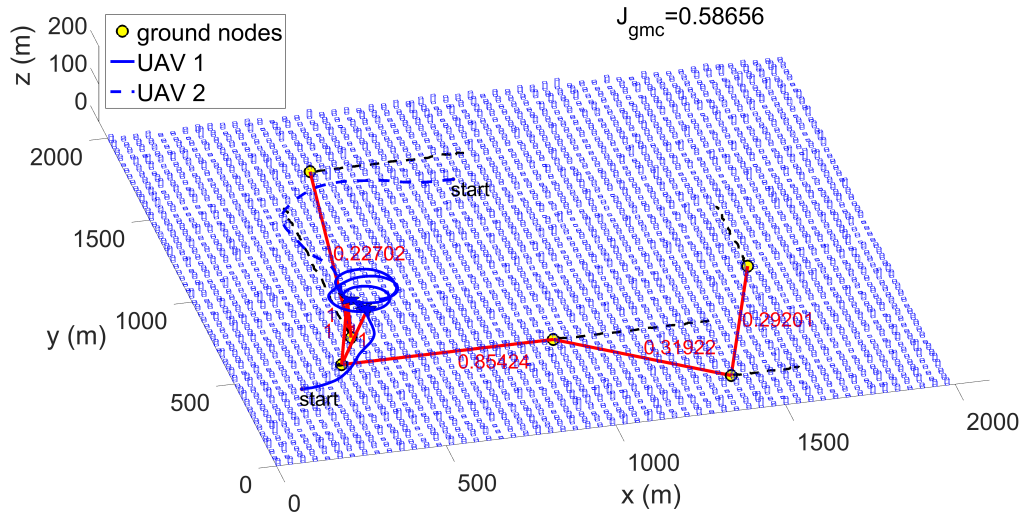
(b) RH+NN with LAP mode

Figure 5.6: Simulation results using different approaches. The red lines represent the MST with the corresponding probability of successful communication.

method. Fig. 5.6 shows a sample scenario out of 48 with trajectories outlined for each of the methodologies. In this scenario, some problems with assuming the wrong model in the LAP approach are clearly visible. Each of the other three approaches sends one UAV to stay around x:1400 and y:500 to serve as a relay to the group of three nodes there. However, NMPC with incorrect LAP model makes both UAVs to stay in the one area where they can only help three nodes. In this work, incorrect LAP uses a city with buildings which are much higher and bigger than they actually are. It is likely that NMPC with incorrect LAP computed that there is no position where it could help those three nodes thus it focuses both UAVs on helping the other three ground nodes.



(c) RH with the known LAP model



(d) RH with the incorrect LAP model

Figure 5.6: Simulation results using different approaches. The red lines represent the MST with the corresponding probability of successful communication (cont.).

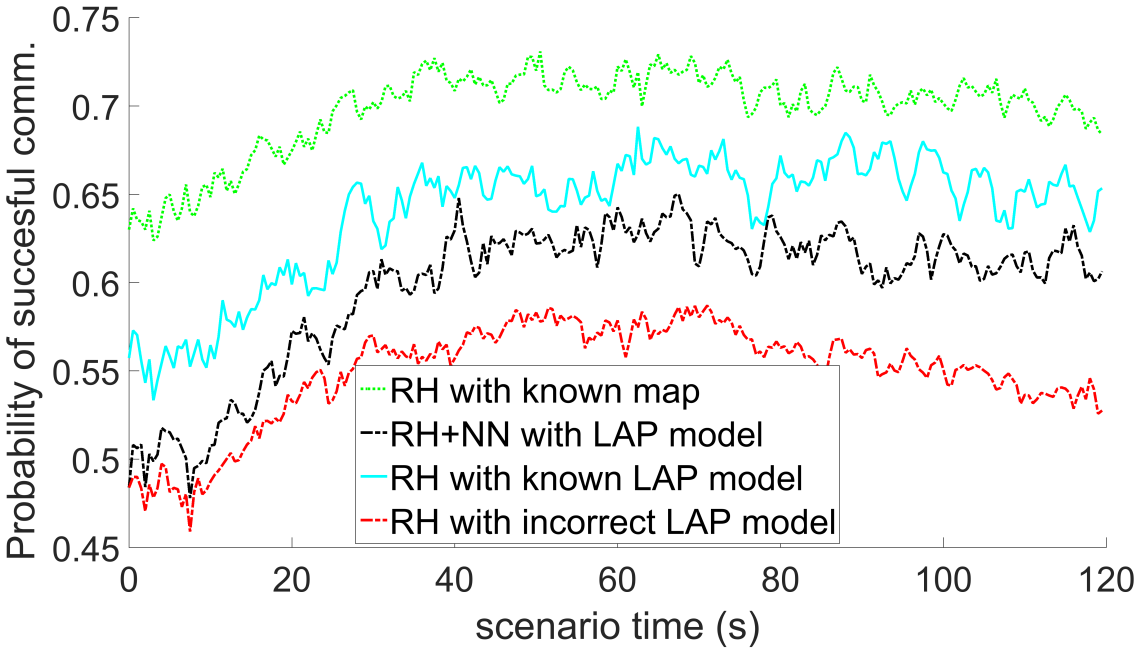


Figure 5.7: Performance comparison of the proposed method with others, averaged over 48 Monte Carlo simulations.

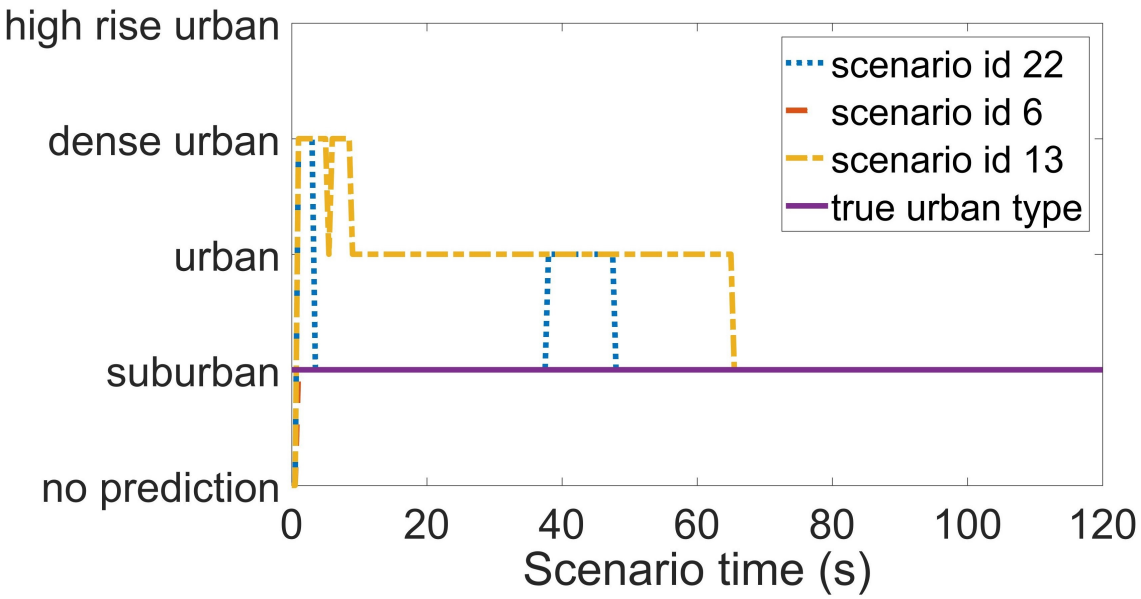


Figure 5.8: Prediction of the urban environment type with time for 3 randomly chosen scenarios. Each scenario id corresponds to the random scenario from Monte Carlo simulations.

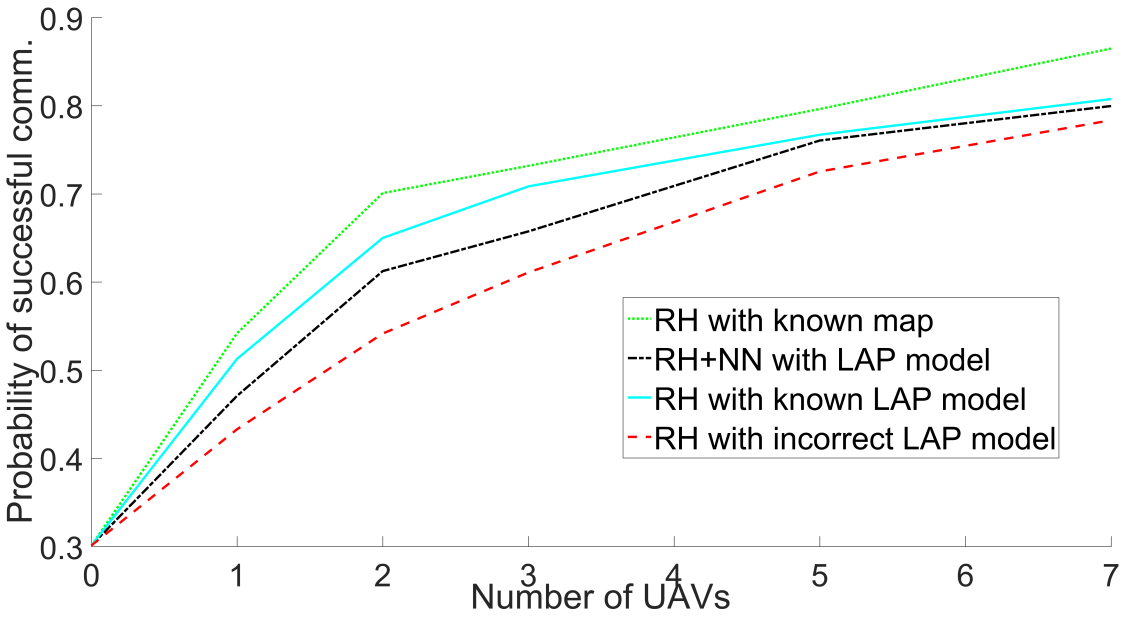


Figure 5.9: Averaged performance over 48 scenarios for the final 20 seconds of the flight with the different number of UAVs. No UAV case indicates the scenario with just ground nodes for the comparison purpose. As the number of UAVs increasess the probability of successful communication increases.

5.5 Summary

In this work, the use of Neural Network in support of UAV communication relays was explored. UAV relay was used to help communication amongst a group of mobile ground nodes. It was shown that the proposed approach can quickly predict the right urban environment type. It can be concluded that this approach is useful as it requires very little knowledge about the mission area, yet it can perform quickly with improved performance. Genetic Algorithm was replaced with cross entropy optimiser (CEO) as on average it reaches the same answer in a fewer computational steps. The velocity was added to trajectory planner to improve communication performance benefit offered by the UAV.

Chapter 6

Experiment in Unknown Simulated Urban Environment

The purpose of the experiment is to validate the GP channel prediction approach proposed in Chap. 4 in an experimental setting. This experiment includes three objectives: i)testing the sufficiency of computation speed, ii) testing the accuracy and consistency of GP made predictions, and iii) proposing a way to simulate an urban environment in an indoor area. Computational speed is a concern as UAV's flight time is limited. If most of this time is spend computing GP map, relay mission time would be heavily restricted. Computation needs to be performed quickly enough to guarantee time for relay mission. The second aim is to test GP performance in more realistic settings. In the simulation chapter, several assumptions were made regarding signal strength in an urban environment to simplify the model. Such assumptions do not necessarily translate very well to real-world experiments. Finally, real-world testing in an urban environment is challenging to perform due to flight restrictions in a real urban environment. Here an attempt is made to simulate an urban environment at a small scale while keeping intrinsic properties of the concerned environment. Since, water is good at absorbing wireless signal, even with smaller dimensions, water containers are used to emulate the role of buildings in real-world cities.

6.1 Overview

The snapshot of the experiment is shown in Fig. 6.1, while more detailed overview can be seen in 6.2. The test consists of two ground nodes, and a single UAV in a 5m by 5m by 6m (WxDxH) indoor area. The buildings are made of water containers. Water has excellent absorption properties at 2.4 GHz band commonly used by commercial wireless networks. UAV performs back and forth scan pattern on a fixed height to collect signal strength data from ground nodes. Based on collected data, a GP grid map is created. The experiment tests four different combinations between ground nodes and buildings. Those four tests

are broadly split as i) one ground node in the middle of the room, ii) two ground nodes in opposite sides of the room, iii) two ground nodes with one building blocking access to one of the ground nodes and iv) two grounds at random positions in a randomly generated city consisting of 15 buildings. The video of summary of the experiment can be found at `https://youtu.be/rFXolMM6CNA`.

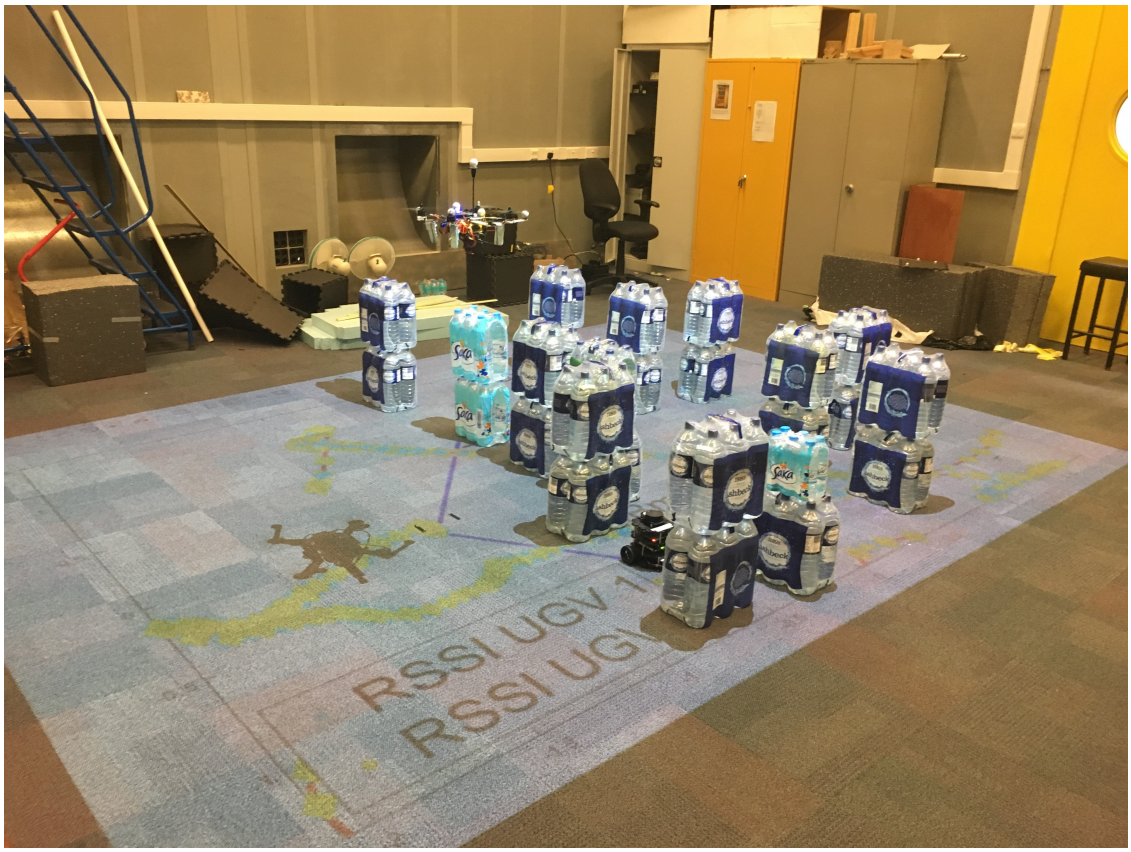


Figure 6.1: Snapshot of experiment

6.2 Experiment Prelimnaries

To perform the experiment, several aspects had to be considered and decided. Firstly, the network topology had to be determined to be determined. Network topology defines what is the structure of connections between vehicles. Secondly, the networking protocol had to be chosen. Networking protocol is responsible for routing packets around the network efficiently and organising and managing connections between different vehicles. Finally, the experimental setup both in terms of vehicles and software is described.

6.2.1 Overview of Network Topologies

Network topology determines the structure of connections amongst nodes (vehicles). [115] has presented a good overview of network topologies. In general, there are four different topologies: ring, star, tree and mesh, represented in Figure 6.3

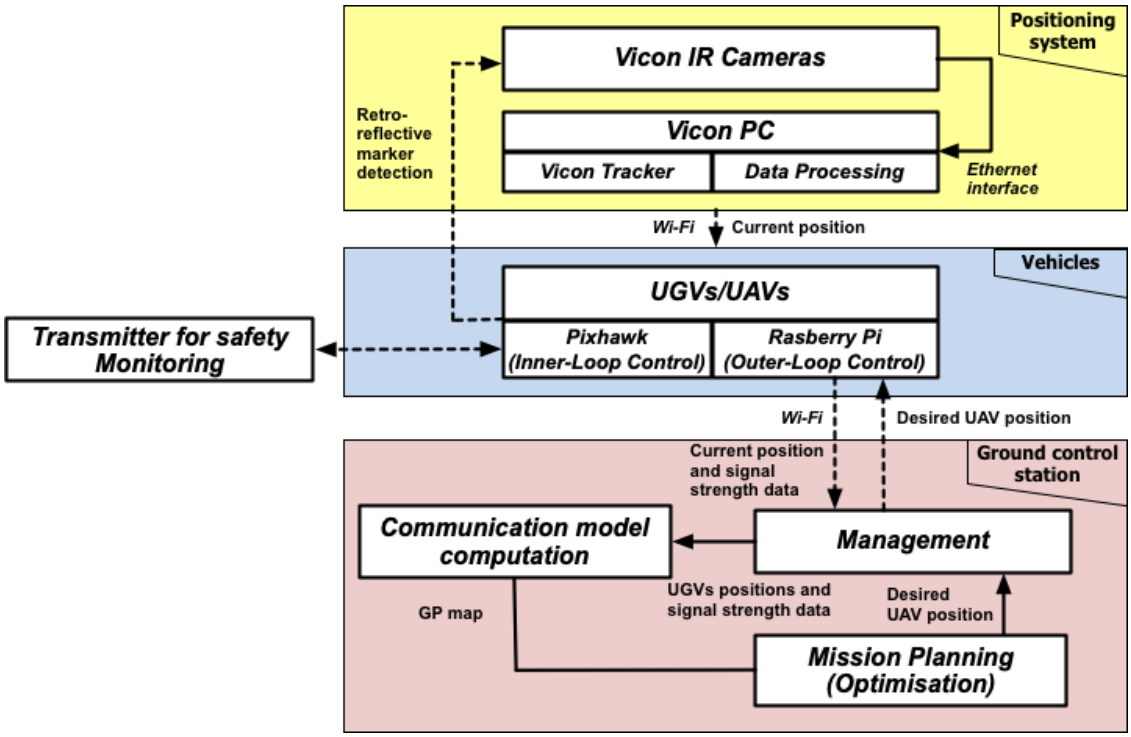


Figure 6.2: Overview of experimental setup

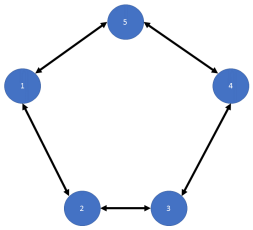
Ring - In a ring network, each node is connected to two other neighbouring nodes in a ring structure. To send data between nodes in the network, the message travels in a circle until it reaches the destination node. One of the advantages of the ring network is built-in redundancy for one connection failure, i.e. with one failed connection, data can still reach all nodes albeit at a reduced throughput. In a ring network, the speed of data transfer is slow due to the long distance between non-neighbouring nodes.

Star - In a star network, there is a single centre node connecting all others. Compared to ring networks, the throughput is significantly higher as the path taken by data is shorter. However, the redundancy is sacrificed by introducing a single point of failure in the form of the central node. With central node failure, no other node can exchange information.

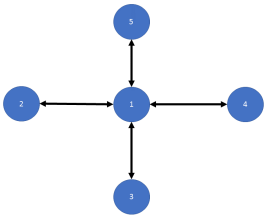
Tree - Tree topology groups nodes in a hierarchical structure. Tree topology is essentially multiple star topologies staggered on top of one another. The critical advantage of tree topology is the ease of adding extra nodes to the network. Tree topology has a high degree of redundancy, but it is relatively easy for a group of nodes to become detached from the rest of the nodes in case of single node failure.

Mesh - Mesh topology is a structure where all nodes are connected. For practical reasons, it is much more common to encounter partial mesh, where only specific connections between nodes are allowed. The key advantage is built-in redundancy, where the failure of a single link is unlikely to cause problems in the network. Another ad-

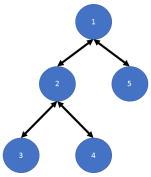
vantage is the ease of adding new nodes to the system. However, redundancy is not guaranteed. For example, it is possible for a mesh to have a single point of failure when the only one connection between two groups of connected nodes is present.



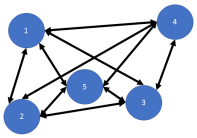
(a) ring network topology



(b) star network topology



(c) tree network topology



(d) mesh network topology

Figure 6.3: Overview of network topologies which could be used in the experiment. The blue filled circles represent a component of the network while black arrows are wireless links. Each component could be either a ground or an air vehicle.

From all aforementioned topologies, mesh network appears to be most suitable for UAV communication relay mission and this experiment. During relay mission, topology is likely to change for two reasons, and mesh networks are more adaptable in terms of topology changes than other types. First of all, the signal strength between nodes will

change during mission time, causing connections to break or appear. Secondly, it might be necessary to add or remove relays during mission time, for example for charging.

6.2.2 Overview of Mesh Networks Protocols

To allow data routing within mesh topology, the appropriate protocol had to be chosen. Several criterions dictated the choice of a routing protocol. The most important requirement is compatibility with the Robot Operating System (ROS). ROS is supported by all vehicles available for use in for this experiment. The second requirement is the ability to send data quickly and reliably to facilitate the positional data transfer between Vicon motion system and the UAV. This delay cannot be longer than 0.2 seconds; otherwise, autopilot assumes data is too old and starts an emergency descent procedure. To fulfil those requirements, three protocols BATMAN Adv. [116], IEEE 802.11s [117] and IEEE 802.15.4 ZigBee [118] were considered.

BATMAN Adv. - A better approach to mobile ad-hoc network (BATMAN) is designed as a low computational complexity, distributed networking protocol. In this approach, each node on the network only holds information about neighbours and general direction to the destination node, rather than full routing information. With limited knowledge, each node can determine the sub-optimal route fast. BATMAN is well documented and is capable of working with ROS.

802.11s - 802.11s is a standard of mesh networking developed by IEEE. It has several distinct features compared to other approaches. First of all, each node on the network can act as a mesh station, mesh access point or mesh portal. Mesh station is used to connect 802.11s to other 802.11 based networks. A mesh access point can forward and receive packets within the 802.11s network. Mesh portal has a very similar function to access point but provides services to other non-802.11 networks such as 802.3. For data transfer, the following procedure is obeyed. Initially, path request from the origin node to the destination node is sent out. Each node adds either its own ID in sequence and forwards it to its neighbours, or if it knows the route to the target node, it simply fills the rest of the table. Once the full destination node is reached, the optimal route is determined, and a route table with confirmation of destination is sent back to the origin. The route table is cached for some pre-specified amount of time for future usage.

802.15.4 - 802.15.4 is a mesh implementation relying on ZigBee infrastructure. Zigbee is a small low powered radio commonly found in UAVs' application due to its weight and size. In 802.15.4, one of the nodes is called the coordinator. The coordinator is responsible for holding information about routes and make them available on request from any of the ground nodes. With this single node holding all routing information, routing can be performed almost optimally.

6.2.3 Performance Comparison of Three Mesh Networks

A comparison between the performance of protocols was performed, To choose an appropriate networking protocol. There does not exist a direct comparison of the three protocols in literature to the best knowledge of the author. Thus the comparison is performed between BATMAN against 802.11s based on work in [119] first. In this work, it was shown that 802.11s had much lower throughput than BATMAN. The 802.11s standard is using the 802.11g standard as underlying architecture, as opposed to the 802.11n standard used in BATMAN. 802.11g has a maximum throughput of 54 Mbps while 802.11n has a throughput of 300 Mbps. On the other hand, 802.11s showed an advantage in two areas: reduced latency and increased data delivery reliability. Suboptimal routing methodology in BATMAN meant that many packets were simply lost or took a very long route. It is worth reiterating that one of the essential requirements was the reliability of data transfer in the network, to facilitate fast transmission of positional data to the UAV. Moreover, the high bandwidth is not that important as data size used in this experiment is small. Thus 802.11s with its increased data delivery reliability, and reduced latency is a better solution for this problem than BATMAN.

With 820.11s standard being better than BATMAN, it only remains to compare 802.11s vs 802.15.4. The choice between those two standards can be made using the second criterion, compatibility with ROS. 802.11s is compatible with ROS 'out of the box' while using 802.15.4 requires an external package such as [120]. Encoding and decoding data to and from ROS would likely introduce delays in data transfer. This could result in slow transmission of positional data, which can trigger an emergency landing procedure. With concerns that 802.15.4 has for ROS implementation, it was decided to follow the 802.11s standard in this experiment.

6.2.4 ROS Overview

This section will provide a brief overview of the robot operating system (ROS). ROS is a series of open source software packages which are designed to simplify control and communication between robots. In ROS, software is gathered into packages. A package is a standalone entity responsible for a broader function. For example, MAVROS is a package responsible for communication with mavlink enabled autopilots. Upon startup, each package launches one or more nodes. One of the nodes in the network is assigned with a master role, while others are slave nodes. The master is tasked with keeping track of a number of nodes connected and routing data between them. Slave nodes can disconnect and reconnect as necessary. However, once the master is initialised, it needs to stay connected; otherwise, the connections break. Nodes are responsible for communicating with each other to transfer various data types, such as positional data or motor commands. To achieve that, nodes use the concept of topics and message types. Topics are responsible to carry single message for a specific purpose, for example, topic \mavros \mocap \pose

carries x, y, z position and x, y, z, w rotation from motion capture system. Message types can be thought of as classes which define data structure within a given topic. For example, in topic \mavros \mocap \pose x, y, z position and x, y, z, w rotation would be all defined as float64 numbers. For a node to declare readiness to send data across the network, it is said to publish to a given topic. It is often said that a given node is a publisher to a topic. For a node to retrieve data from a given topic, it is said to subscribe to a given topic or node is called a subscriber to a topic. Until there is a subscriber to a topic, the publishing node will not send out any data to reduce the burden on networking. Each topic can have multiple subscribers and publishers. Publishers and subscribers are connected directly, with master used only initially to establish appropriate connections.

6.3 Experimental Hardware

6.3.1 Ground Vehicle

In this experiment turtlebot 3 burger UGV is utilised. The robot overview is shown in Fig. 6.4. Turtlebot is equipped with Raspberry Pi, open CR board and LIDAR. LIDAR is not utilised in this experiment so its description is omitted.

Raspberry Pi 3- Main computer onboard of turtlebot. It is running Ubuntu 16.04 with ROS kinetic. For wifi connectivity, the external dongle is attached. Although raspberry pi has a built-in wi-fi module, its drivers are incompatible with 802.11s, so external dongle had to be used. In turtlebot, the onboard computer is responsible for running the turtlebot package called: turtlebot3_bringup to enable control over the robot. Raspberry Pi is connecting the ROS node with an onboard OpenCR controller.

OpenCR- is a controller board for turtlebot. OpenCR board is responsible for translating commands from onboard computer to motors and power distribution. The board is based on Arduino microcontroller and is using the serial port to communicate with the raspberry pi computer. It is worth noting that LIDAR is not connected to OpenCR board, instead, it is connected to raspberry pi directly via USB port.

6.3.2 Air Vehicle

In this experiment custom built quadcopter UAV shown in Figure 6.5 was used. It is a standard F300 size frame with large 3 cell 5000 mAh battery for extended endurance. A more specific component breakdown in both hardware and software is shown in Fig. 6.6.

What follows is a detailed breakdown of system components for a quadrotor UAV used in this experiment.

Raspberry Pi - A popular microcomputer running Ubuntu 16.04 and ROS Kinetic. The Pi on board the aircraft serves two main purposes. Firstly, it is used to connect quadrotor with rest of the 802.11s network. Secondly, it acts as the interpreter and relay for messages between various sources which would not be able to communicate directly.

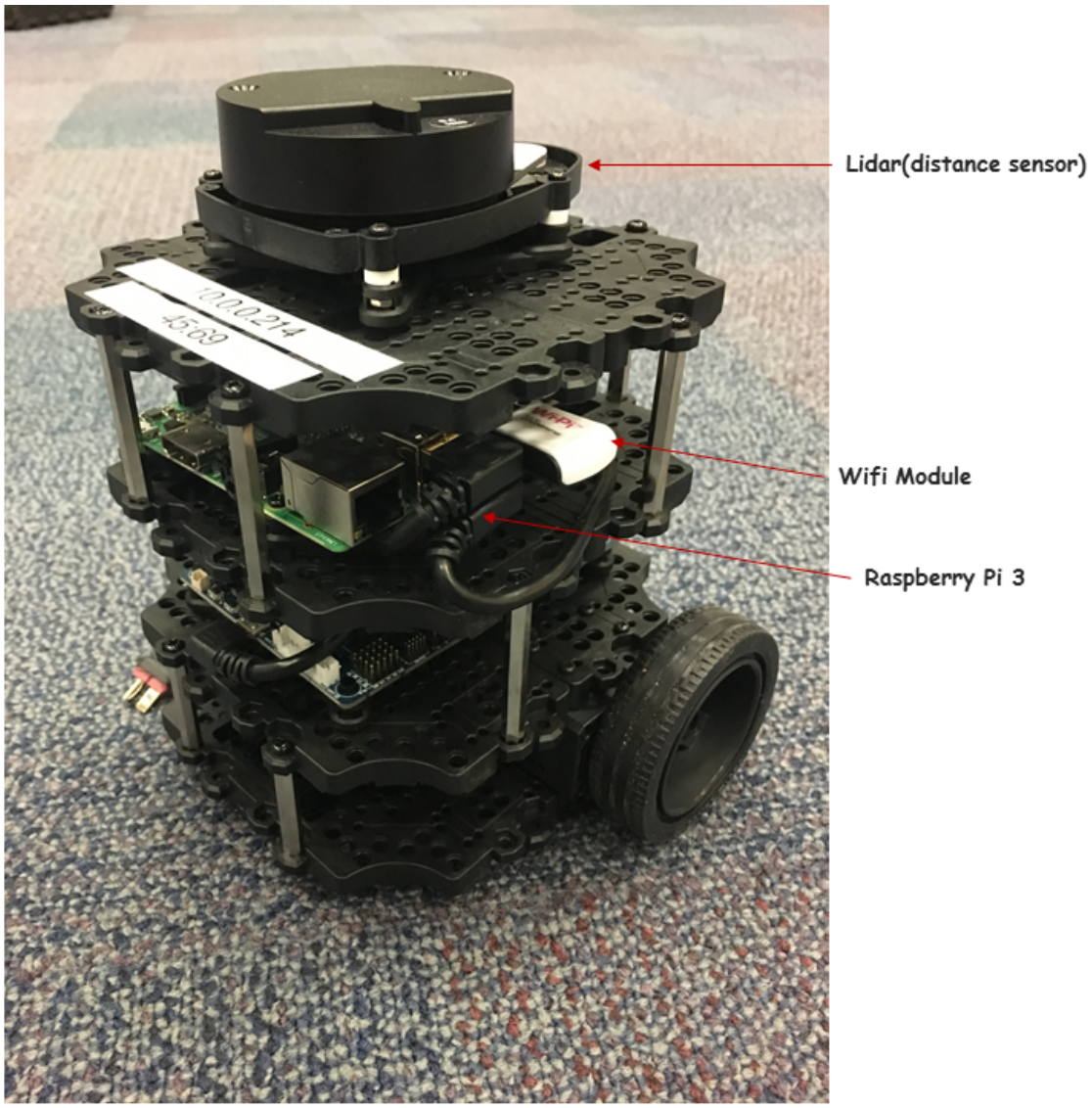


Figure 6.4: Overview of turtlebot 3 UGV used in this experiment with important components highlighted

For example, MAVlink messages sent out by the Pixhawk on the serial port have to be interpreted and made available to the ROS network so that it can be used by the controller and read by the PC. Similarly, messages from the controller or PC need to be converted to a suitable format to be sent to and understood by the Pixhawk. The Pi is powered via its' Micro USB port and communicates with the Pixhawk via a serial port, as shown in Fig. 6.7.

MAVROS **-** [121] This is a ROS package which acts as a bridge between ROS and the autopilot. It interprets messages emitted by the autopilot in the form of MAVLink messages to ROS topics. MAVROS topics include, but are not limited to transmitter output, position in a global and local frame, aircraft attitude and speed. One of the critical features of MAVROS is the ability to send commands to a vehicle through a facility called "offboard mode". There are specific topics which allow control of attitude,

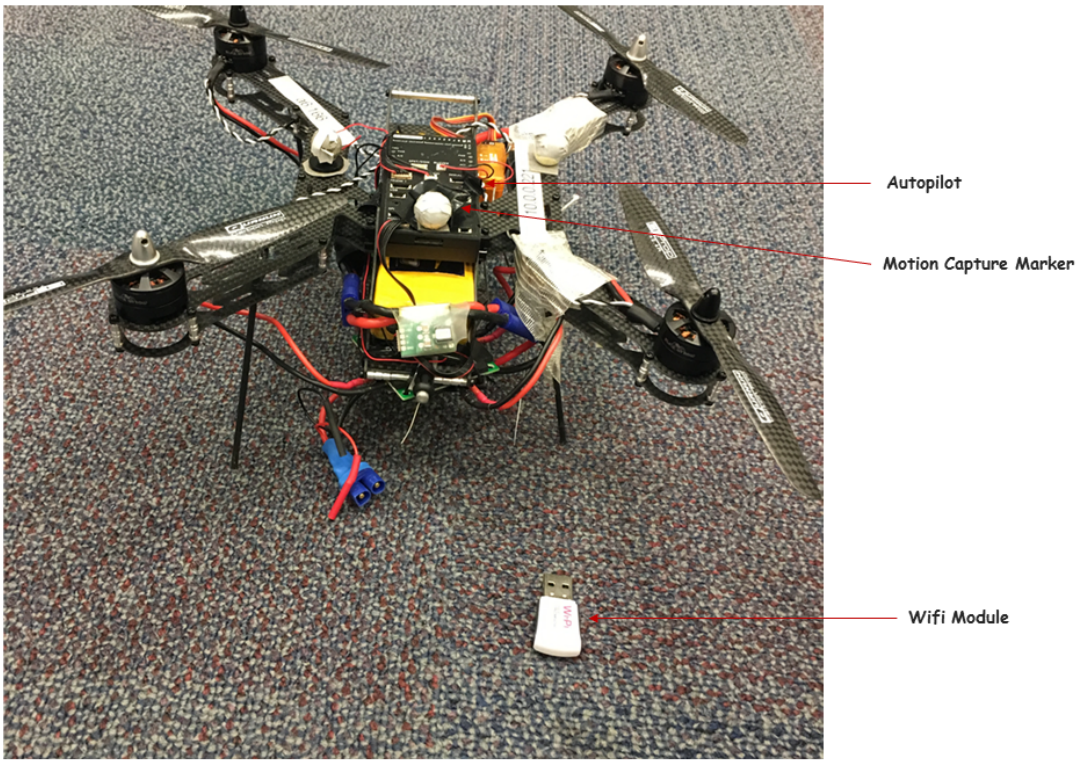


Figure 6.5: Overview of quadrotor UAV used in this experiment with important components highlighted

position and velocities in offboard mode namely

- */mavros/setpoint/attitude*
- */mavros/setpoint/position_local*
- */mavros/setpoint/velocity*

where velocities can be both linear and rotational. The offboard mode is fully compatible with the PX4 autopilot stack. It can be activated by a switch which can be set up on a transmitter using QGroundControl. With offboard mode disabled, the transmitter has full control over the vehicle. While enabled, the offboard mode allows the vehicle to be controlled through one of the topics mentioned above. As an added safety feature, if the offboard mode switch is enabled and no information is provided on any of the control topics, or if the message rate is too low, offboard mode will not engage. Such an approach means that the pilot can always take over control if a tested algorithm does not behave in a desired or safe manner. Two files used to launch mavros can be seen in Listings 6.1 and 6.2.

Wi-Fi Receiver **-** A Wi-Fi interface is used to provide connectivity between the aircraft and other systems. The specific adapter used here is based on the rt5780 chipset to ensure the best compatibility with the 802.11s standard.

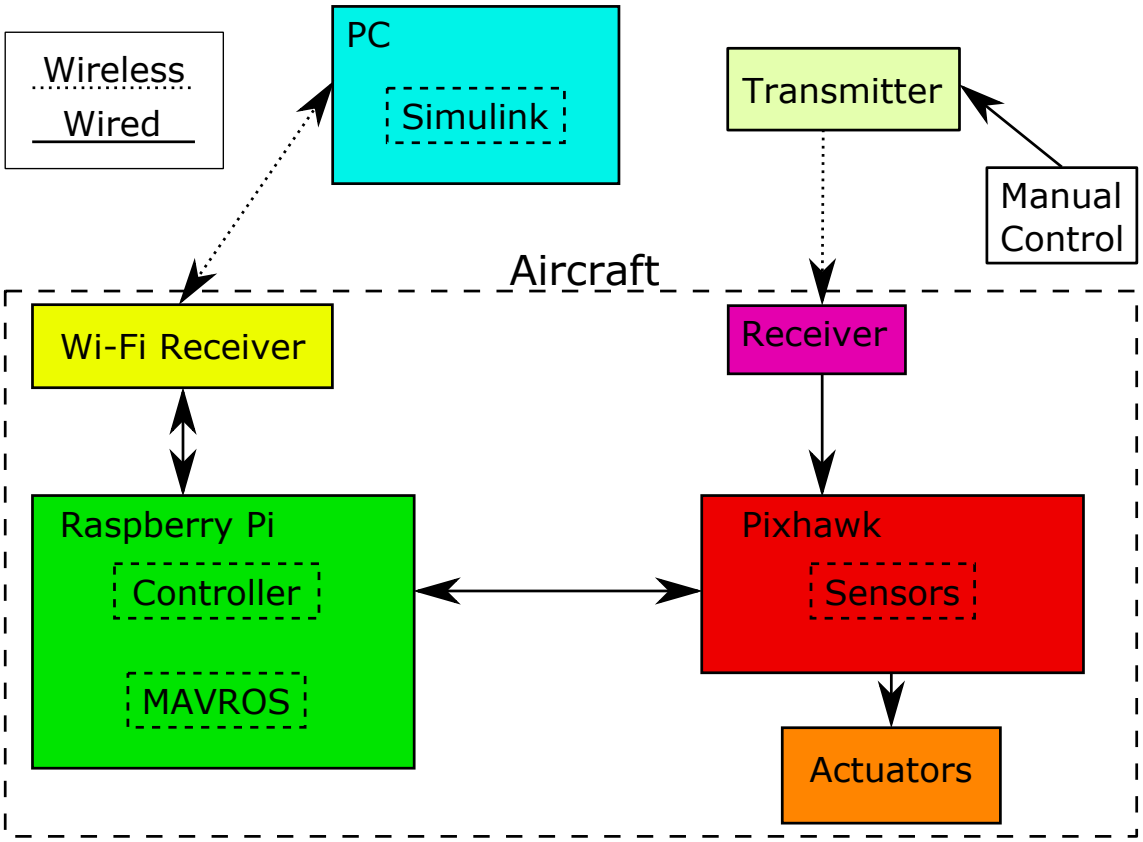


Figure 6.6: System overview for rotary wing aircraft.

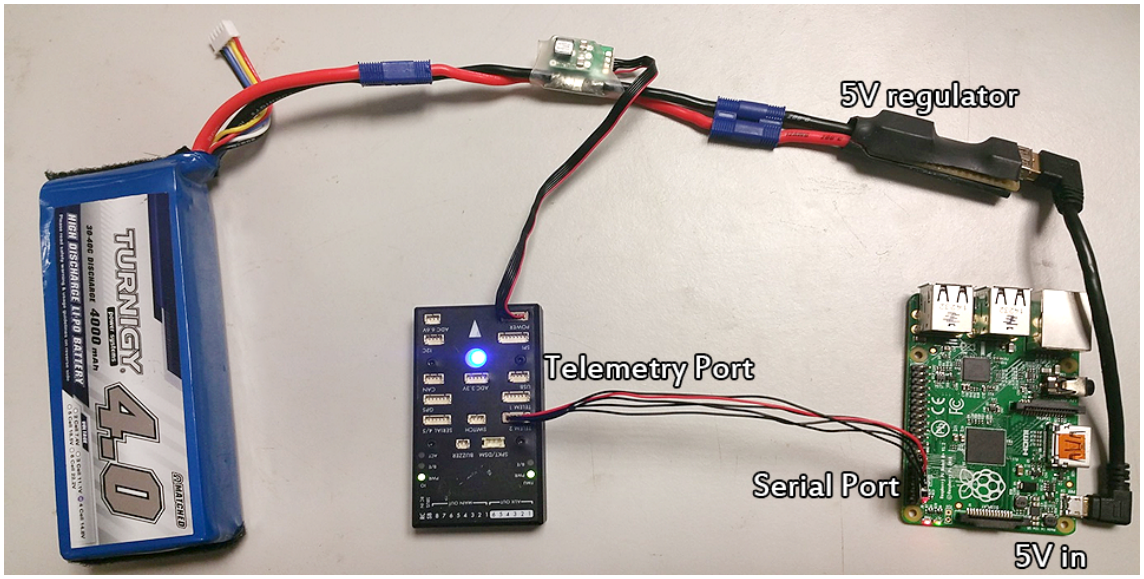


Figure 6.7: Hardware overview of common ROS/Autopilot system components.

Pixhawk **-** The Pixhawk is a COTS autopilot which is popular in the small UAV area. In terms of hardware, the Pixhawk provides a broad range of sensors including a 6 DOF IMU, pitot static airspeed sensor, GPS and barometer. On the software side, the PX4 flight stack is used. With PX4 and the MAVlink protocol, most of the PX4 data inputs and outputs are available over the serial port. MAVROS interprets these on

the Pi and made available to the controller. One of the key advantages of using ROS to test experimental controllers is safety. Adding custom controllers into the Pixhawk firmware is possible through source code modification. However, as well as requiring good programming knowledge, this also introduces risk to the aircraft. A bug in custom code can lead to crashing of the Pixhawk software, which would likely lead to the crashing of the aircraft. Placing the custom code on the separate Raspberry Pi and not modifying Pixhawk source code means there is a permanent and reliable fail-safe if an issue occurs with the custom code. Control can be handed back to the Pixhawk with a single button toggle on the transmitter.

Transmitter/Receiver **-** The transmitter and receiver provide a long-range wireless control link between the aircraft and the pilot. The transmitter interprets manual control inputs and sends them using radio signals to the receiver, which passes these on to the Pixhawk. The primary consideration for the transmitter and receiver is signal quality, which dictates maximum operating range. Onboard the aircraft, multiple "satellite" (auxiliary) receivers are connected to the primary receiver to improve signal quality by removing dead zones which can arise from electrical and signal interference. As so many electrical components are contained within a small area, interference is likely to occur. The use of multiple receivers in different locations and orientations mitigates this risk. In the case of loss of control link, the Pixhawk is configured to perform an emergency landing.

Actuators **-** these are the components which affect the physical motion of the test platform. For a multirotor, the motors are considered as actuators.

For more details regarding the system the reader is invited to study recently publish ROS overview [122].

6.3.3 City Model in Experimental Area

Performing an experiment involving UAV communication relay in the realistic urban environment is challenging. Regulations heavily restrict flights in an urban environment. It could be possible to utilise military or firefighters urban training ground; however, such an experiment would be limited in repeatability. The indoor test can solve both of those limitations. As of the time of writing this thesis, flying indoors is mostly not sanctioned by civil aviation authority (CAA). With the correct setup, the experiment can be repeated as many times as necessary in the indoor area to obtain a meaningful number of experiments.

Indoor area poses one challenge of effectively emulating effects on wireless communication by the urban environment. First of all, due to the limited size of the buildings in the scaled environment, typical buildings materials such as bricks and concrete are not thick enough to attenuate signal significantly. This is mainly an issue considering the low cost of wi-fi dongles used in this experiment, which are even less likely to detect small differences in signal strength. The second challenge is making the indoor emulated urban environment, easily adaptable and modifiable to test the performance of GP extensively.

Listing 6.1: mavros node launch file

```
<launch>
        <!-- vim: set ft=xml noet : -->
        <!-- example launch script for PX4 based FCU's -->

        <arg name="fcu_url" default="/dev/ttyAMA0:921600" />
        <arg name="gcs_url" default="" />
        <arg name="tgt_system" default="1" />
        <arg name="tgt_component" default="1" />
        <arg name="log_output" default="screen" />

        <include file="$(find mavros)/launch/node.launch">
                <arg name="pluginlists_yaml" value="$(find
                    mavros)/launch/px4_pluginlists.yaml" />
                <arg name="config_yaml" value="$(find mavros)/
                    launch/px4_config.yaml" />
                <arg name="fcu_url" value="$(arg fcu_url)" />
                <arg name="gcs_url" value="$(arg gcs_url)" />
                <arg name="tgt_system" value="$(arg tgt_system)"
                     />
                <arg name="tgt_component" value="$(arg
                    tgt_component)" />
                <arg name="log_output" value="$(arg log_output)"
                     />
        </include>
</launch>
```

To solve those challenges usage of water containers as a material for buildings is proposed. First water is excellent at absorbing 2.4 GHz wireless network [123] used by 802.11s standard even with small obstruction. With such a significant change, cheap wi-fi dongles are capable of detecting the difference quickly. Second, water containers can be easily moved and stacked to create different city layouts as desired.

6.4 Empirical Communication Model

To accurately asses communication performance by GP, it is important to compare its performance against model-based approaches. However, the communication model developed in previous chapters could not be used in this experimental setting. The previously defined models were intended for a much larger scale than that of an experiment. Additionally,

Listing 6.2: mavros node launch file

```
<launch>
        <!-- vim: set ft=xml noet : -->
        <!-- base node launch file -->

        <arg name="fcu_url" />
        <arg name="gcs_url" />
        <arg name="tgt_system" />
        <arg name="tgt_component" />
        <arg name="pluginlists_yaml" />
        <arg name="config_yaml" />
        <arg name="log_output" default="screen" />

        <node pkg="mavros" type="mavros_node" name="mavros"
            required="true" clear_params="true" output="$(arg
            log_output)">
                <param name="fcu_url" value="/dev/ttyAMA0
                    :921600" />
                <param name="gcs_url" value="udp
                    ://10.0.0.121:14551@10.0.0.130:14551" />
                <param name="target_system_id" value="$(arg
                    tgt_system)" />
                <param name="target_component_id" value="$(arg
                    tgt_component)" />
                <!--<param name="use_sim_time" value="true"
                    />-->
                <!--remap from="/mavros/vision_pose/pose" to="/
                    vicon/miniQuad/miniQuad"/-->


                <!-- load blacklist, config -->
                <rosparam command="load" file="$(arg
                    pluginlists_yaml)" />
                <rosparam command="load" file="$(arg config_yaml
                    )" />
        </node>
</launch>
```

the buildings considered in those models are composed of different materials than water, causing further inaccuracies. Thus the need for a communication model suitable for these scenarios arises. It is worth noting that it is not the intention to claim that this is the best possible model for the circumstances, nor that developing this model forms a substantial contribution to the area. The measurements performed here are bound to have an error due to not using anechoic chamber and poor accuracy of measurement equipment. However it is believed that this model is better than a general multi-purpose model in the experimental area. To develop correct model, two experiments are performed; first to determine the distance based model, and second to model effects of buildings on wireless communication.

6.4.1 Distance Based Model

To compute the distance based model, the following procedure was followed:

1. quadrotor UAV was placed in one end of the room, but still within Vicon range. Turtlebot UGV was positioned on the other end.
2. UAV was moved back and forth to and from turtlebot five times to collect signal strength from UGV over the length of the room.
3. Turtlebot was rotated by 90 degrees. Step 2 was repeated until data was available for all four sides of turtlebot. This step was necessary to account for possible small directional differences antennas may have.
4. steps 2 and 3 were repeated for two different turtlebots. This step was performed to account for small differences between dongles caused by an imperfection in the manufacturing process.
5. The data for four sides of two turtlebots were combined into a single dataset.
6. Polynomial fit for the combined data was obtained
7. Polynomial equation can be used to predict signal strength versus distance relationship.

With those steps completed, a new distance-based model suitable for our experimental area Fig. 6.8 was created. While collected data is limited to 5 meters, it is sufficient for our experiment as the experimental area is limited to 5m by 5m due to Vicon limitations. The model is formulated as follows:

$$W_i = -46.05d^{0.1376} \tag{6.1}$$

where W_i is signal strength between node i and the UAV and d is the distance.

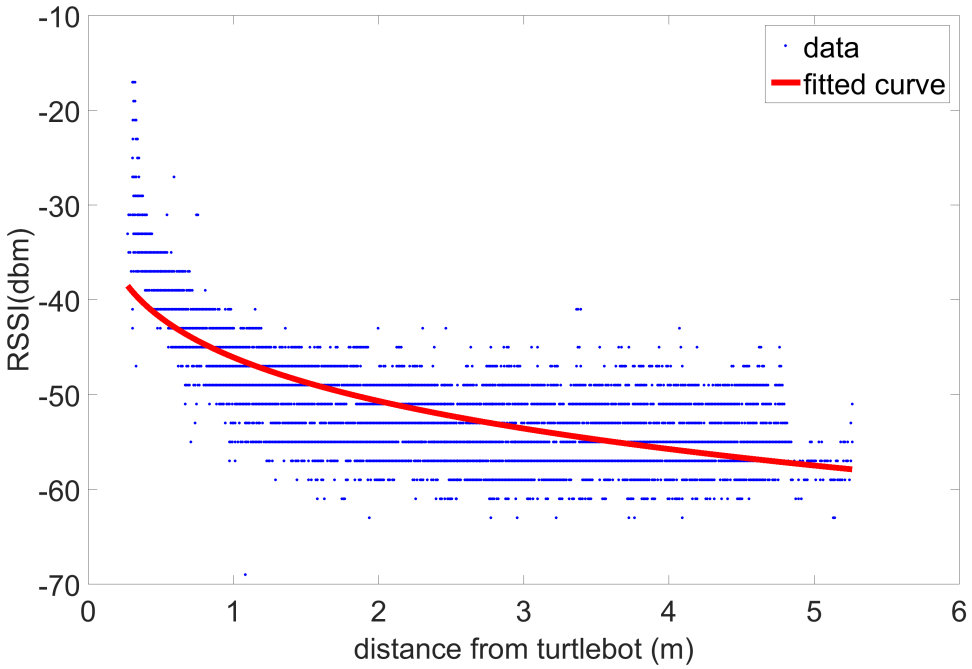


Figure 6.8: Signal strength versus distance relationship between turtlebot and quadrotor. Due to relatively small distance RSSI reduction is small, but still noticable.

6.4.2 Effects of Buildings

To generate the effects of buildings for the empirical communication model, the following procedure was employed:

1. The quadrotor and the turtlebot were placed with four sets of bottle waters between them.
2. Signal strength from the turtlebot was collected over two minutes.
3. Buildings were removed one by one and step 2 was repeated after each removal, to collect data with reducing the number of buildings.
4. Polynomial fit to the combined data was obtained.
5. Polynomial equation can be used to predict signal attenuation by a given length of obstruction by building.

Fig. 6.9 shows the fit of a polynomial equation to the obtained data. It can be noted that the fit is not a linear fit but a polynomial. First layers of water containers stop the majority of the electromagnetic waves. Upon entering the next layer of water containers, there are not as many electromagnetic waves; thus the probability any of them will meet a water particle to be absorbed is significantly smaller. Thus the overall reduction in signal strength will be smaller. The model can be formulated as:

$$W_{bi} = -2.204d^{0.2352} + 0.2642 \tag{6.2}$$

where W_{bi} is the strength of obstructions i and the UAV, d is the distance of the obstructions.

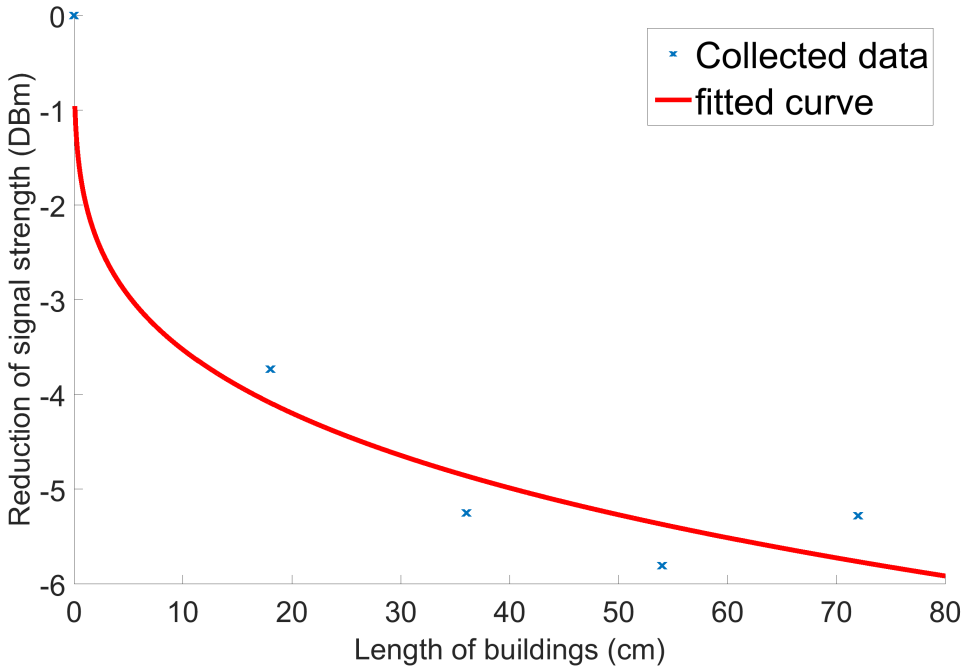


Figure 6.9: Distance of LoS obstruction and its effects on signal strength.

6.5 Experimental Setup

6.5.1 Experimental Overview

This section describes how each separate component from the previous section combines together to form overall experiment.

Outline of vehicles and connection in this experiment is shown in Fig. 6.10. There are one or two turtlebots and one quadrotor UAV connected using the 802.11s protocol in a partial mesh topology. Additionally, one laptop is provided to act as a mesh station to attach the mesh with the rest of the indoor network in the lab. Within the indoor network, there are two additional desktop PCs, one to run the algorithm and other to compute and propagate positional data from Vicon. Note the algorithm applied in experimental settings is the same algorithm as described in details in Chapter 4. Positional data is then transferred to the quadrotor which is running vrpn_client_ros ROS package. This package is tasked with interpreting Vicon positional data into a ROS topic. The quadrotor is running the ROS Vicon client package to reduce the time transfer of positional data to the quadrotor.

It is also worth mentioning of some limitation caused by the driver for the rt5870 chipset. Namely, to measure the signal strength from a node at high frequency, the node for which signal strength is measured needs to be sending data at high frequency.

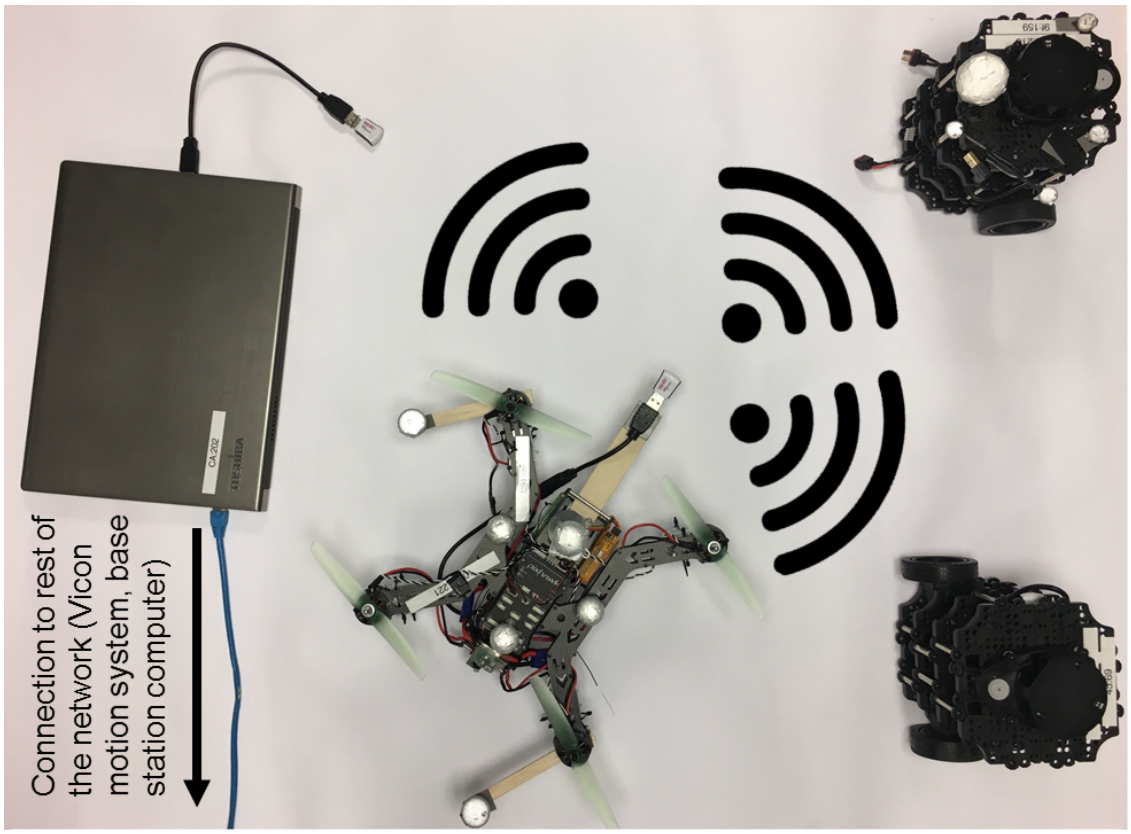


Figure 6.10: Diagram showing experiment setup. Turtlebots are only connected to quadrotor. Quadrotor is additionally connected to laptop. Laptop is responsible for obtaining and propagating data from rest of the network such as vicon positional and control command data

Otherwise, the power saving feature causes the signal strength measurement rate to drop significantly. Such a low rate is particularly a problem when the quadrotor UAV moves relatively fast; thus misalignment between received signal strength and position occurs. For this reason, the turtlebot needs to be continually sending data out to allow constant update of signal strength.

6.5.2 Experimental Procedure

Each of the tests performed during the experiment followed the same procedure outlined below

1. UAV takes off to the preprogrammed launch point.
2. UAV performs back and forth scan pattern flight to collect signal strength signal from ground vehicles.
3. UAV returns to initial starting position while communication strength between arbitrary positions in the air and fixed positions on the ground is predicted using Gaussian Process.

4. (where applicable) Based on GP prediction, the optimal point is found to be where the sum of all signals from the ground vehicle is maximum. UAV moves to the optimal point as predicted by GP

5. (where applicable) UAV stays in optimal point for 15 seconds to collect signal strength data. This data is later used to compare performance between GP and empirical model networking prediction.

6. (where applicable) methodology from chapter 3 is used to find optimal position using communication model from 6.4. UAV gets to that position and again measures signal strength for 15 seconds to collect signal strength data

7. UAV lands, data is saved, and the run concludes.

6.6 Results

Several types of experiments were performed to validate the proposed approach. The first experiment with the single ground vehicle and UAV flying in a back and forth scan pattern was performed. With this type of test, the initial validation of a simple case was performed. For the second experiment, the second UGV was included, to compare the performance of the GP and model in a simple case. For the third experiment type, one building was added between two UGVs, to compare the performance of the GP and model-based network prediction on the slightly more complicated scenario. Finally, two UGVs and fully built-up, randomly generated city with the GP and model-based approach were tested. In each of the experiment, half of the collected data points was used for the GP prediction, while rest was used for error comparison and performance evaluation.

The number of data points collected varied slightly from experiment to experiment and oscillated around 1800 per UGV. Computational time for GP was on average 11.4 seconds on quad-core Intel Core I7-5775 processor with Matlab with two UGVs. The average was computed across all experiments involving more than one UGV. This time includes both optimising the hyperparameters (described in Section 4.2.2) and generating the grid points map (described in Section 4.2.3). This computational time is faster than what is indicated in Fig. 4.12. There are two main reasons for this. Firstly, the CPU used in the experiment has much faster per core performance compared to the one used in the simulation. Secondly, in the simulation, only one core was used per simulation run, while in the experiment Matlab is able to utilise multiple cores and threads for GP computation. The distance between each grid point (d_g) was set for 15 cm. The computational time of 11.4 seconds shows that the proposed approach is sufficiently fast for a periodical near real-time updates of predictions during the relay mission. Particularly as relay missions are expected to last tens of minutes, 11.4 seconds for GP update is an acceptable time. Moreover, it could be improved with implementation in a much faster language such as

C++. Also, the GP map update can happen on a separate core of the CPU, thus not affecting the computation of trajectory.

6.6.1 Single UGV

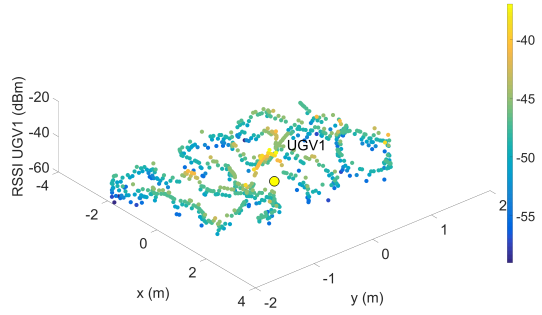
For the experiment of single UGV, it was placed in the centre of the room so that the USB wifi dongle on trutlebot was at the origin $(0, 0)$ x and y coordinates. Examining experimental trials from Fig. 6.11, it can be seen that across the tests, maximum signal strength prediction occurs on or close to the origin $(0, 0)$. This is expected behaviour; signal strength should be the strongest closest to the dongle. It can be noted that sometimes on the edge of the scenario GP predicts high signal strength value. This is caused by random one or two high signal strength values on the sides during the back and forth scan pattern, which creates peak prediction further down.

Error in prediction is analysed in Fig. 6.12. The number of points in an error bin deceases with the increase of the error bounds. This implies that the prediction error generally satisfiers a normal distribution with less points having a large error. The relatively high error of very few data points can be explained by smoothing tendencies of GP. GP prediction will never fully match actual signal strength close to wi-fi dongles. In GP such a high reading is smoothed out by relatively low signal strength reading nearby. This is one of the well-known limitations of GP and can be fixed by usage of Gaussian Process mixture techniques such as one described in [124].

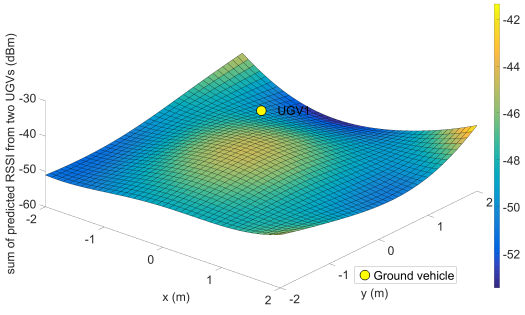
6.6.2 Two UGVs

In this section two UGVs are introduced for the first time in the experimental settings. The primary purpose of this experiment is to compare the performance of GP against the created model. For the sake of space saving, hereafter only GP prediction results are presented, while raw data figures can be found in Appendix A for reference.

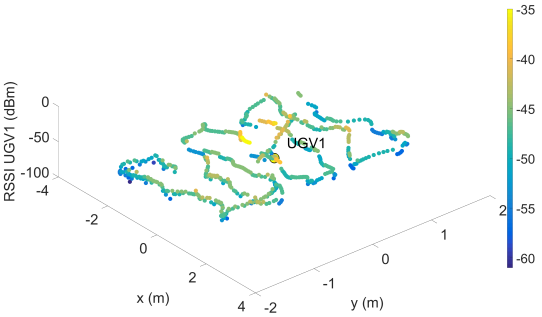
With this simple scenario without buildings, the performance of both communication models can be compared. For model-based communication prediction, the optimal point is in the middle between two ground nodes. For measurement based GP, optimal position varies (see Fig. A.3 for details). In this case, the variation can be either due to the error of prediction or due to a different characteristic of the wireless network at the time of the experiment. Those two behaviours can be clearly distinguished by examining Table 6.1 for each of the scenarios which show a 15-second average of signal strength from two ground nodes, at both optimal positions. For scenario ID 4 and 5, GP performance is better despite not being close to the middle. The reasons for such behaviour are unknown and are assumed to be caused by the stochastic nature of the wireless signal. Besides here the interest is not to explain those effect but to compare the performance of a simple model to GP. On average, however, the GP has a slight performance advantage compared to the model-based approaches. Fig. 6.14 shows very similar tendency as for the single UGV



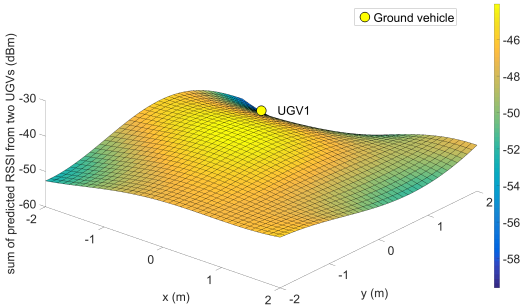
(a) Collected signal strength raw data trial 1



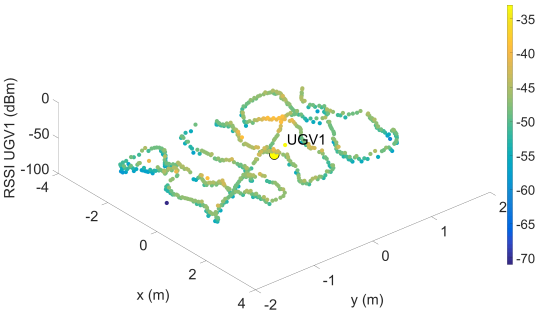
(b) GP prediction for trial 1



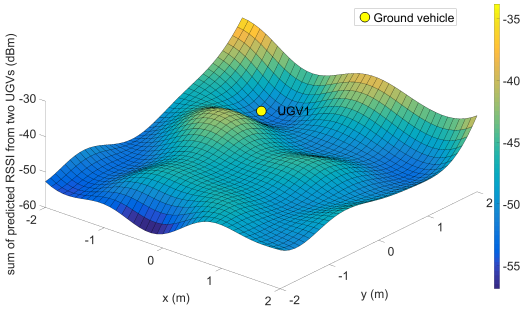
(c) Collected signal strength raw data trial 2



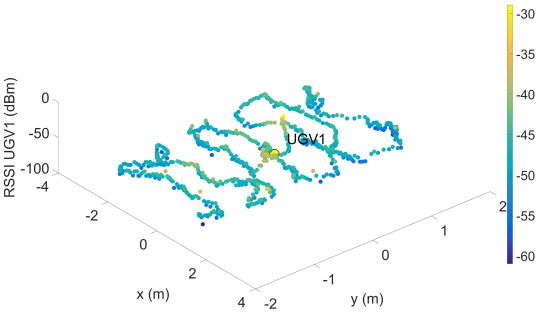
(d) GP prediction for trial 2



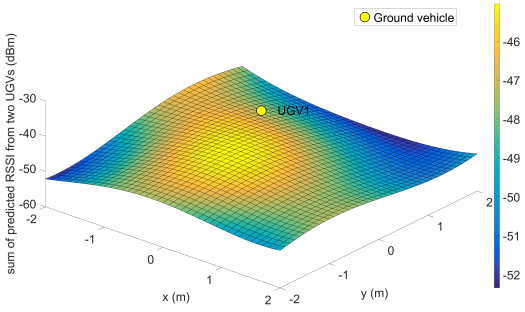
(e) Collected signal strength raw data trial 3



(f) GP prediction for trial 3

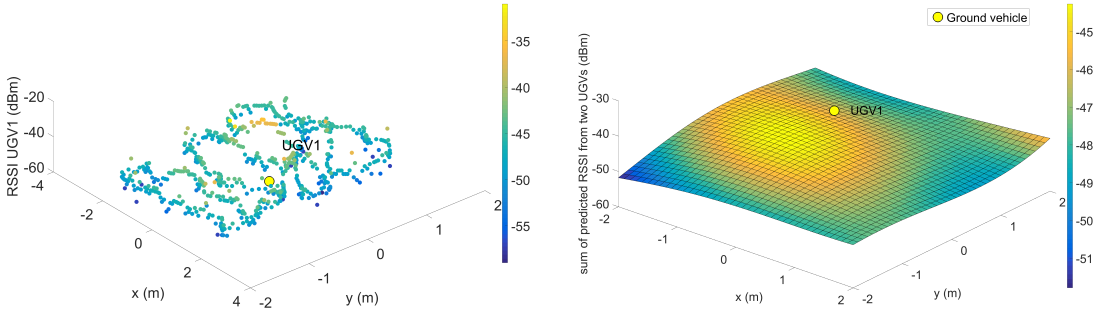


(g) Collected signal strength raw data trial 4



(h) GP prediction for trial 4

Figure 6.11: Summary of experimental trials. Each trial is represented as raw data and GP prediction. About half data points in raw data were used for prediction



(i) collected signal strength raw data trial 5

(j) GP prediction for trial 5

Figure 6.11: Summary of experimental trials. Each trial is represented as raw data and GP prediction. About half data points in raw data were used for prediction s (cont.)

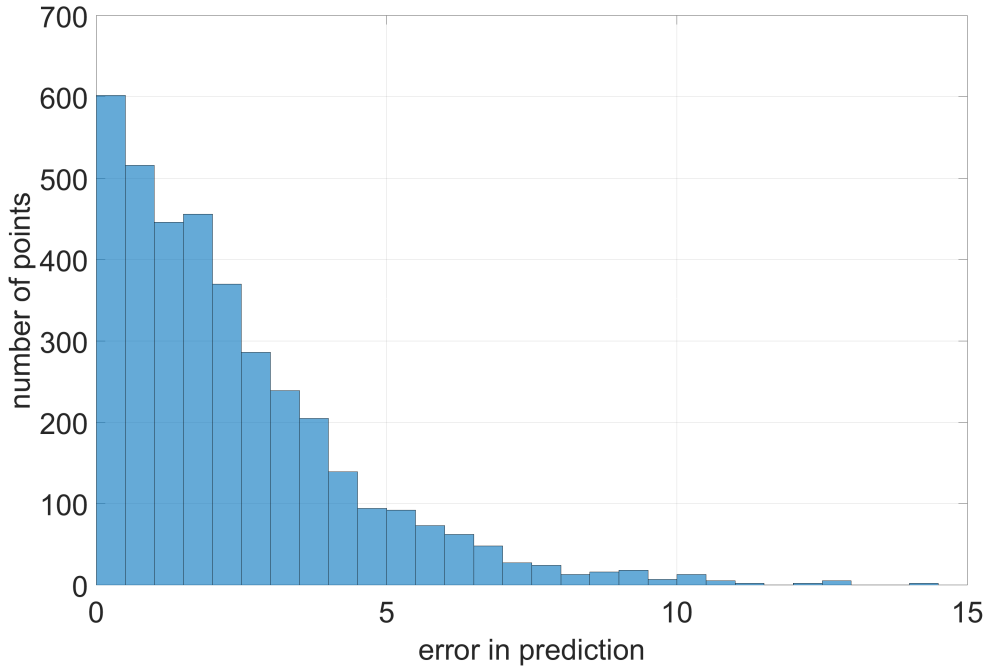
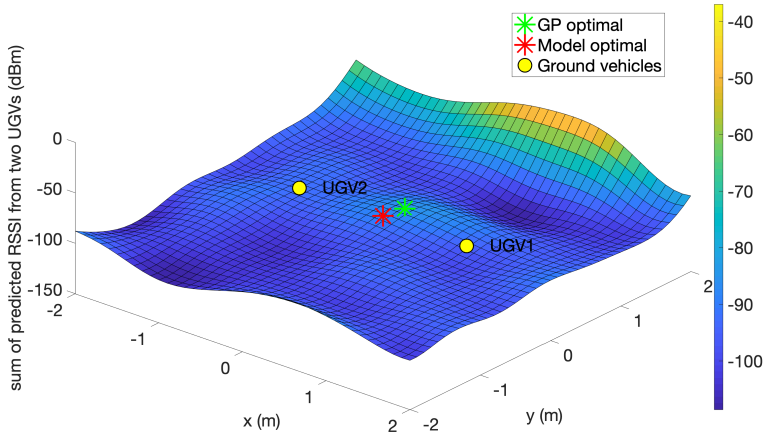


Figure 6.12: Error histogram between five trials. Half of the data was used for developing the model for prediction, while the other half was used for calculating the error.

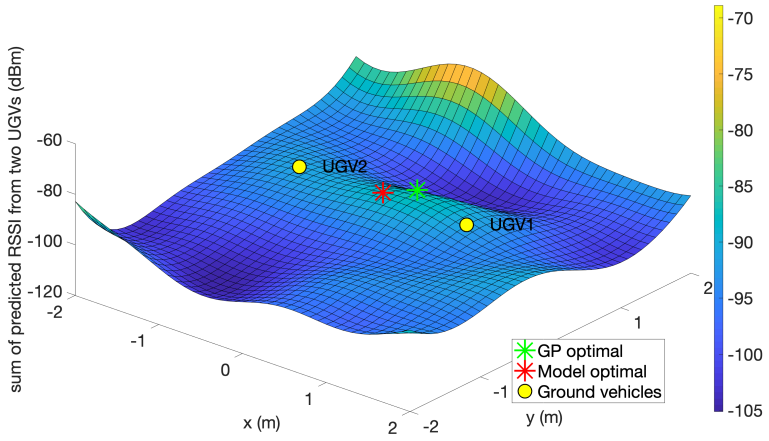
case.

6.6.3 Two UGVs and One Building

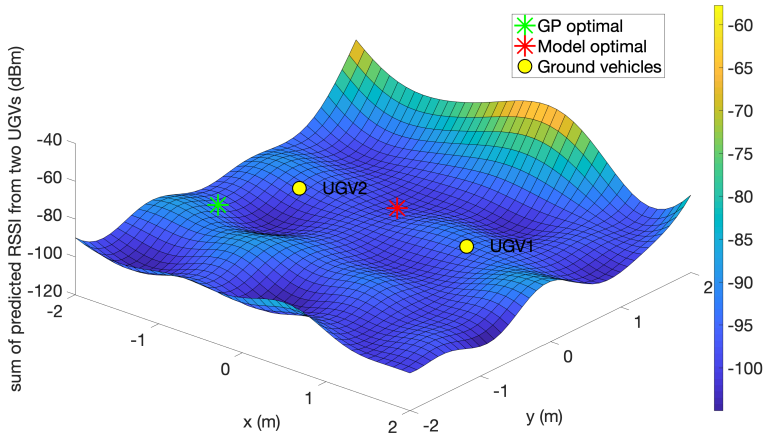
In this part, one additional building is placed close to one of the turtlebots as a line of sight obstruction between the two. Initial comparison of GP and empirical model in a more complex setting can be performed. From Fig. 6.15, it can be noted that predicted position for the UAV with the empirical model and GP are very different. However based on Table 6.2, using the two solutions are on average very similar with a slight advantage of GP based approach. This is a promising result, implying in a realistic setting GP can have



(a) GP prediction for trial 1



(b) GP prediction for trial 2



(c) GP prediction for trial 3

Figure 6.13: Summary of experimental trials. Each trial is represented as a sum between two GP prediction for two UGVs.

Table 6.1: Summary of results across multiple runs for two UAVs

scenario id	signal strength for GP optimal position (dBm)	signal strength for model optimal position (dBm)
1	−91.15	−92.79
2	−86.44	−92.87
3	−95.46	−99.66
4	−96.53	−99.33
5	−94.93	−100.86
mean	−92.90	−97.10

performance very similar to that of a good model. Unlike the model, the GP can achieve this without relying on a number of correctly estimated parameters. Error histogram (Fig. 6.16) shows correct trends that the number of points in an error bin decreases with the the bounds of the bin.

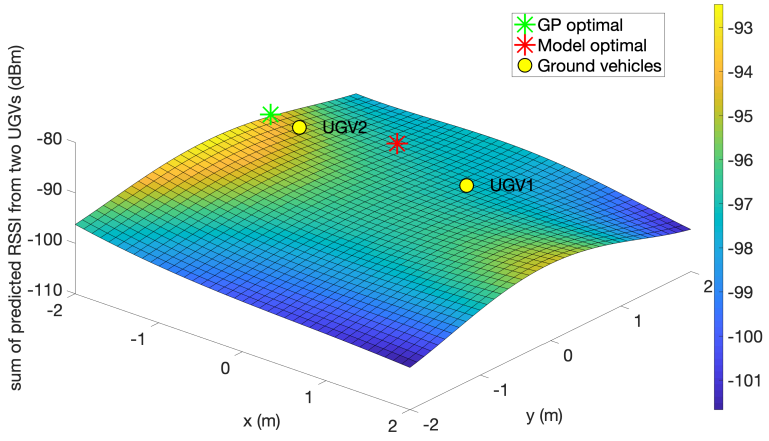
Table 6.2: Summary of results across multiple runs for two UAVs with single building

scenario id	signal strength for GP optimal position (dBm)	signal strength for model optimal position (dBm)
1	−95.90	−98.91
2	−96.74	−95.33
3	−93.4	−98.5
4	−96.08	−97.94
5	−98.06	−95.97
mean	−96.04	−97.34

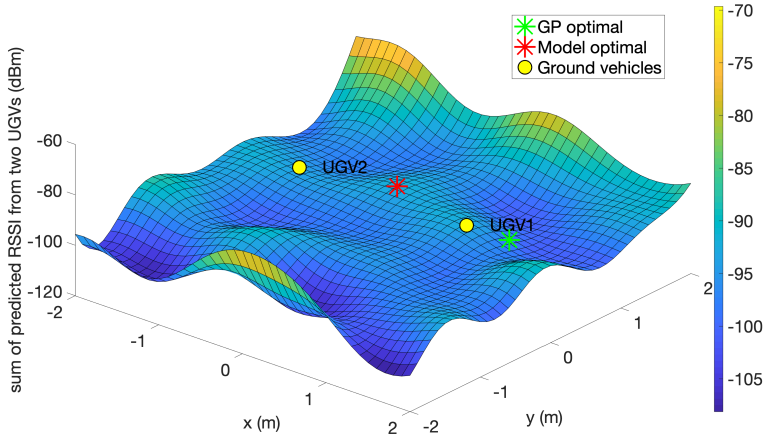
6.6.4 Two UGVs, Full City

Finally, a full city is introduced to the experimental setting. Nine random and one pre-defined city were considered. Random cities consisted of the same height of buildings but different positions and rotations for each run. Random cities were developed to show the robustness of the proposed algorithm in different scenarios, while the pre-defined city was designed to show performance in more realistic city-like environment. All cities layout are shown in Fig. 6.17. The cities for each scenario alongside the results are presented below, while raw data is shown in Appendix A.

Comparing the results from Fig. 6.17, it can be noted that best positions varied widely between the empirical model and the GP learned model. Table 6.3 reflects this difference



(d) GP prediction for trial 4



(e) GP prediction for trial 5

Figure 6.13: Summary of experimental trials. Each trial is represented as a sum between two GP prediction for two UGVs.

by showing vastly different performance results across the runs. On average, GP has a very similar performance as the empirical model approach, with a slight advantage for GP, which is consistent with earlier results. It can be noted that unlike the other two approaches, in a few scenarios model based is significantly better than the GP approach. In such a complex scenario, the stochastic nature of wireless communication is particularly prominent, meaning it is easier for GP to make an erroneous prediction. Error histogram (Fig. 6.18) is also following same trend as the previous two sections.

6.6.5 Conclusions

In this chapter, experiment to assess the performance of Gaussian Process channel prediction in an urban environment was performed. For this assessment, one UAV was utilised

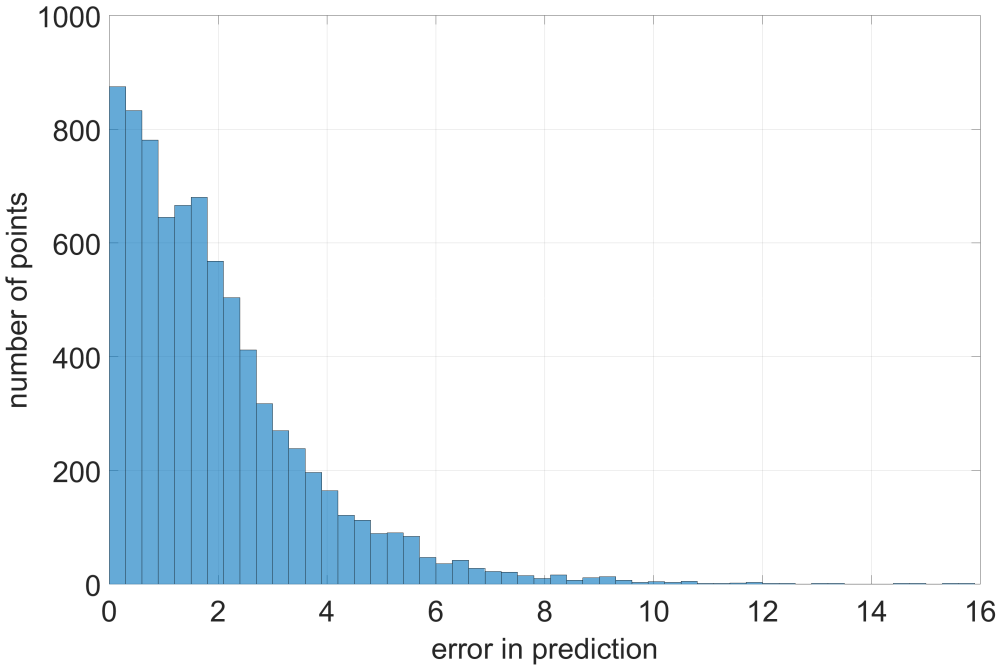
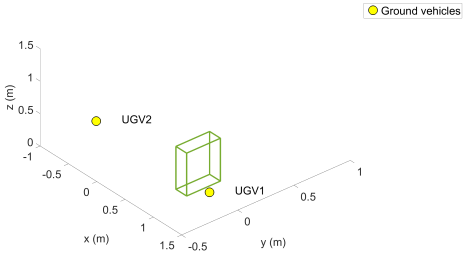
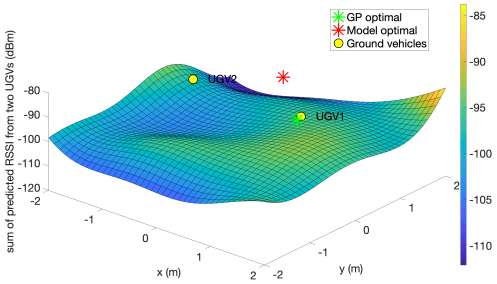


Figure 6.14: Error histogram between five trials. Half of the data was used for prediction, while the other half was used for calculating the error.

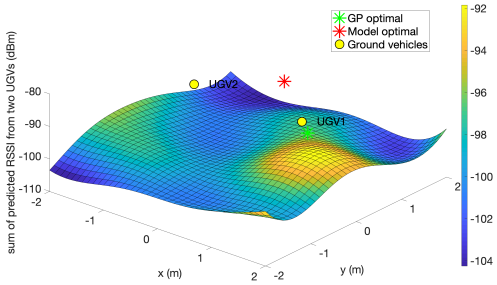
to fly at a constant height and try to optimise communication relay between two ground vehicles. The urban environment consisted of buildings made out of water due to its excellent signal absorption properties. Four different experiments were performed: i)with a single ground node, ii)with two ground nodes, iii)with two ground nodes and one building, and finally, iv) the fully simulated city. It was shown that prediction from GP is consistent across the runs using a single ground node. The first experiment served to show general performance and consistency of GP prediction. On the other side, the other three has demonstrated that regardless of scenario complexity, GP based approach can achieve similar performance as a model-based approach. Although for given indoor environments, the performance of them are identical; for unknown environments, it is difficult to make the experimental communication model accurately, in the first place. For example, a empirical model created is unlikely to work correctly in any other environment due to the stochastic nature of wireless communication. That's why the GP approach was advocated to learn the communication loss model in real-time. The experiment should be extended in the future to include scenarios with mobile ground nodes both with known and unknown trajectory. This would require significant improvement in computational power, for example by using GPU. GP can benefit from parallel execution offered by the GPUs. Also, GP should be extended to sparse GP approach to further improve computational speed in an experimental setting. Finally, for unknown trajectory experiment, Kalman filter would need to be implemented to predict future positions of ground nodes.



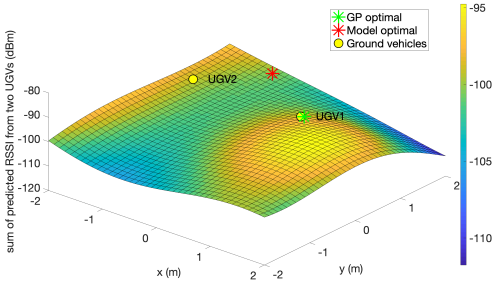
(a) map for two ground nodes with one building



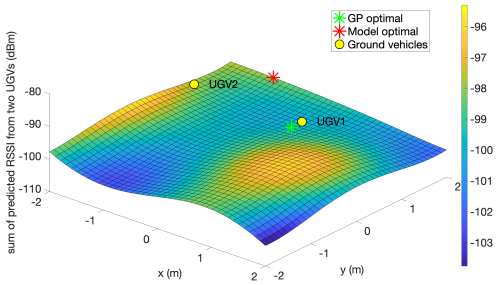
(b) GP prediction for trial 1



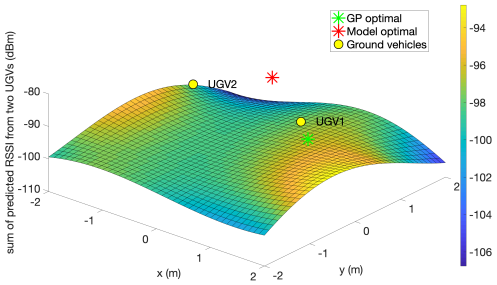
(c) GP prediction for trial 2



(d) GP prediction for trial 3



(e) GP prediction for trial 4



(f) GP prediction for trial 5

Figure 6.15: Summary of experimental trials. Each trial is represented as a sum between two GP prediction for two UGVs. In each of the trials single building was used with position depicted in 6.15a

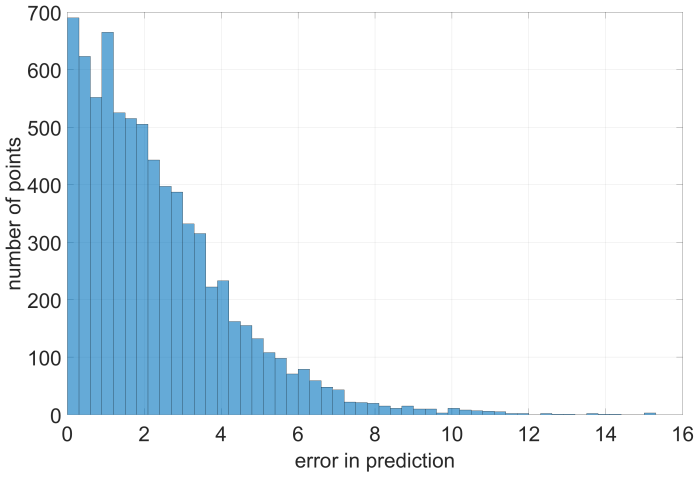
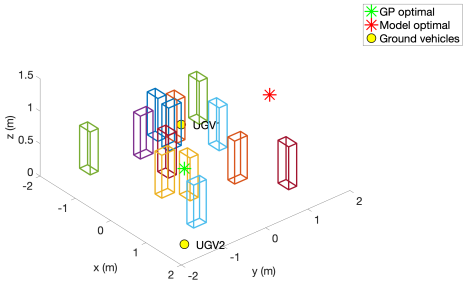


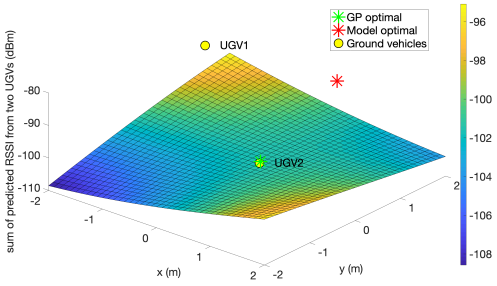
Figure 6.16: Error histogram between five trials. Half of the data was used for prediction, while the other half was used for calculating the error.

Table 6.3: Summary of results across multiple runs for two UAVs with full city

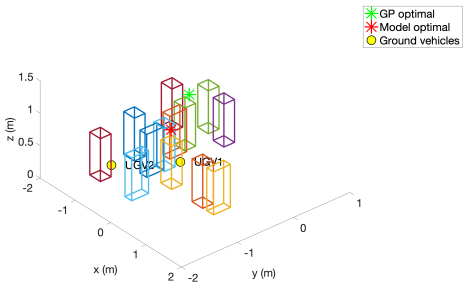
scenario id	signal strength for GP optimal position (dBm)	signal strength for model optimal position (dBm)
1	−101.33	−97.20
2	−94.70	−111.62
3	−91.27	−98.30
4	−99.57	−99.09
5	−98.51	−100.63
6	−97.81	−103.98
7	−95.56	−92.68
8	−92.57	−94.73
9	−95.26	−93.23
10	−101.48	−94.17
mean	−97.10	−98.56



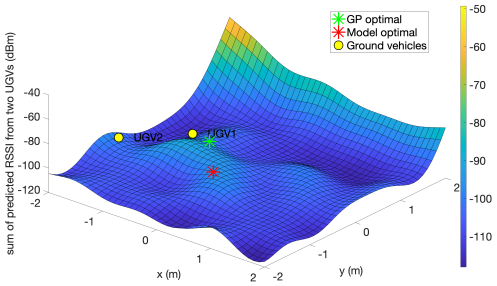
(a) city outline for trial 1



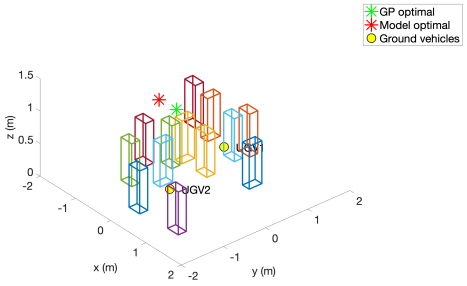
(b) GP prediction for trial 1]



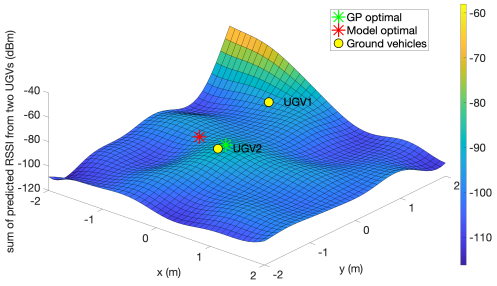
(c) city outline for trial 2



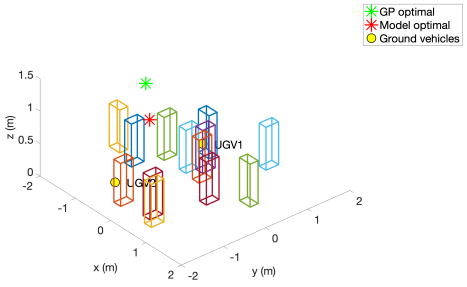
(d) GP prediction for trial 2



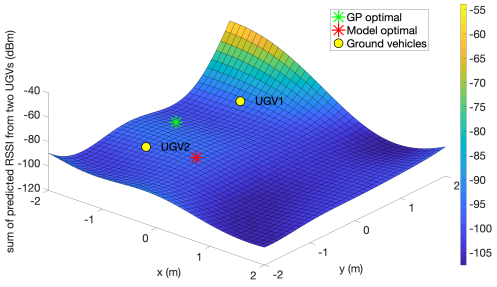
(e) city outline for trial 3



(f) GP prediction for trial 3

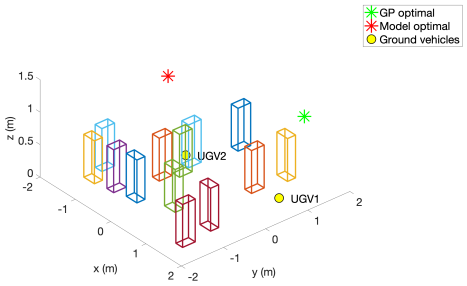


(g) city outline for trial 4

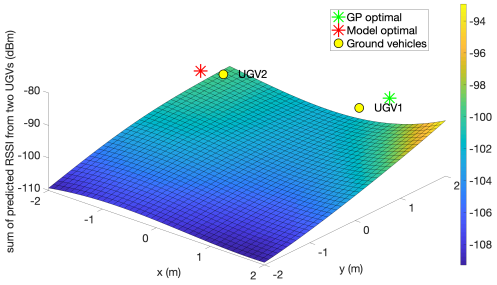


(h) GP prediction for trial 4

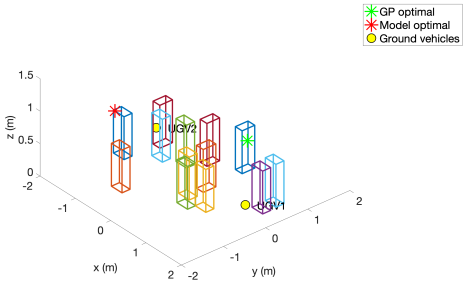
Figure 6.17: Summary of experimental trials. Each trail is represented as a raw data and GP prediction. From raw data about half, data points were used for prediction



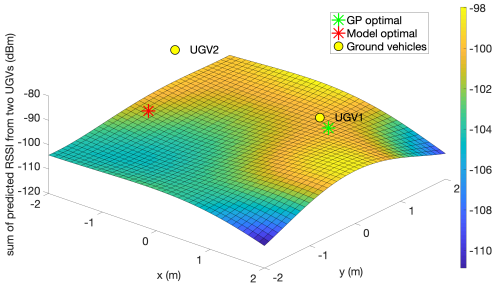
(i) city outline for trial 5



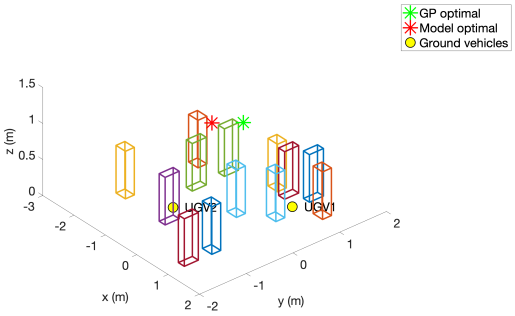
(j) GP prediction for trial 5



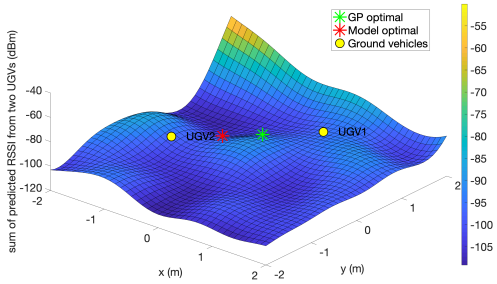
(k) city outline for trial 6



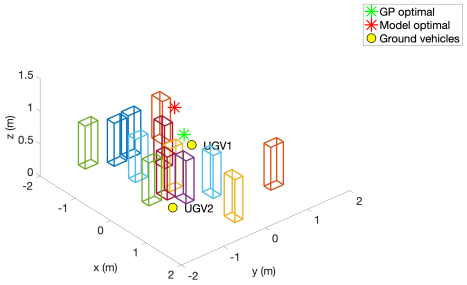
(l) GP prediction for trial 6



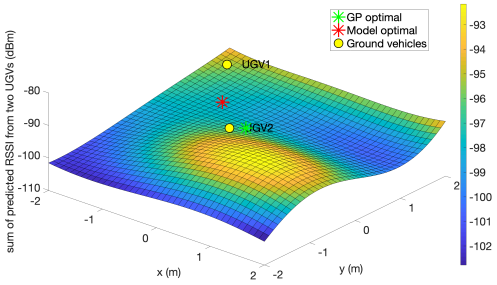
(m) city outline for trial 7



(n) GP prediction for trial 7

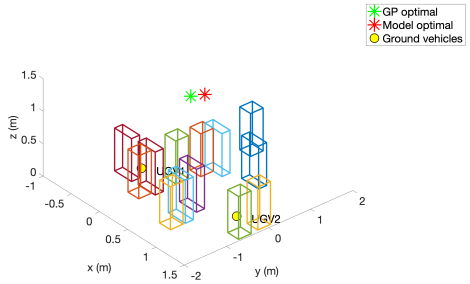


(o) city outline for trial 8

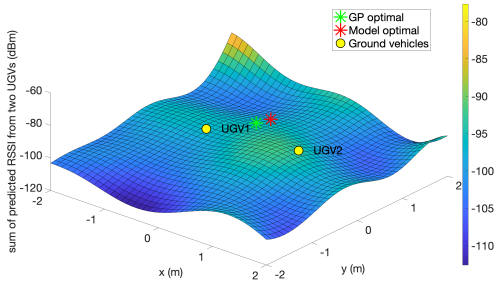


(p) GP prediction for trial 8

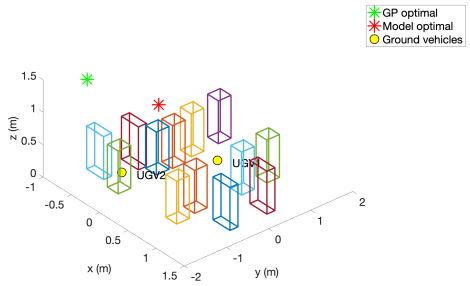
Figure 6.17: Summary of experimental trials. Each trial is represented as a raw data and GP prediction. From raw data about half, data points were used for prediction (cont.)



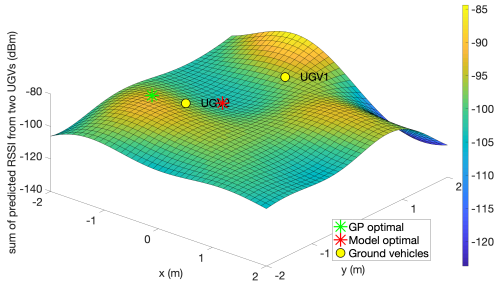
(q) city outline for trial 9



(r) GP prediction for trial 9



(s) city outline for trial 10



(t) GP prediction for trial 10

Figure 6.17: Summary of experimental trials. Each trail is represented as a raw data and GP prediction. From raw data about half, data points were used for prediction (cont.)

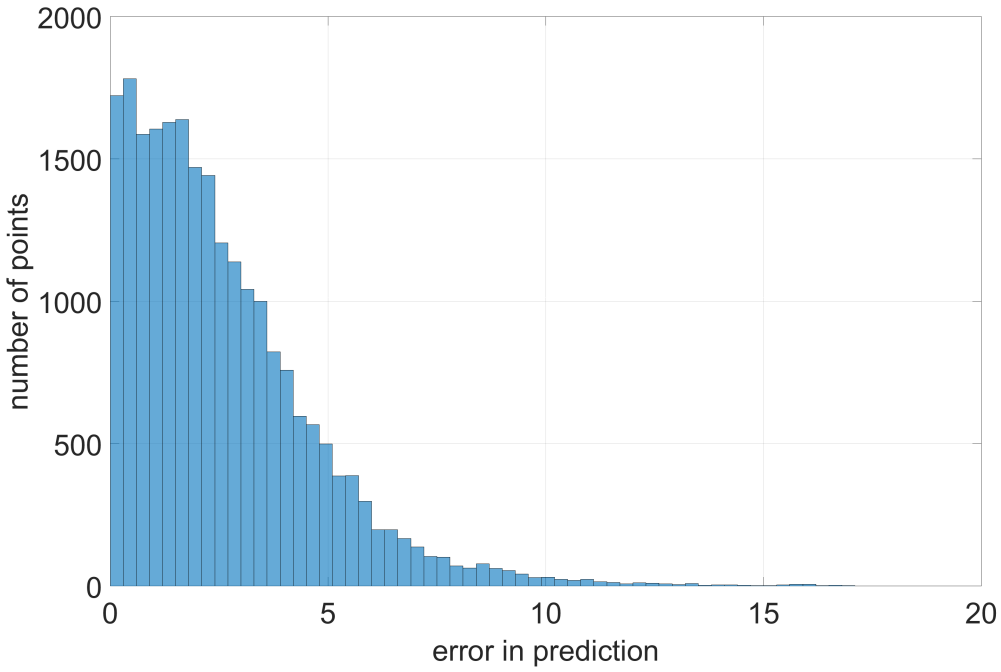


Figure 6.18: Error histogram between five trials. Half of the data was used for prediction, while the other half was used for calculating the error.

Chapter 7

Conclusion and Future Work

7.1 Summary

This thesis explored the trajectory planning and positioning of Unmanned Aerial Vehicles (UAVs) communication relays in urban environments in three different scenarios with varying knowledge levels about the environment. Chapter 3 analysed scenario where wireless communication between air and ground can be fully predicted using a model with relatively good knowledge about the urban environment. Chapter 4 showed approach where wireless communication strength can be predicted based on one of the four pre-defined urban environments types. Chapter 5 explored a scenario where wireless signal strength between air and ground is initially unknown and needs to be fully learned. A final experiment to confirm the approach from chapter 5 in experimental settings was also presented in Chapter 6.

7.1.1 Scenario I: Known Urban Environments

Chapter 3 discussed a model-based approach for wireless communication strength prediction with the assumption of good knowledge about the effects of the urban environment. This chapter explored two aspects: improving the trajectory planning and positioning of UAVs relays and enhancing the method of communication improvement amongst ground nodes.

In this work, the model-based approach for wireless communication strength prediction was combined with a non-linear model predictive control (NMPC) based trajectory planner for path planning and particle swarm optimisation (PSO) for positioning. For the first time, these techniques were used in an urban environment.

A new communication metric, modified global message connectivity (mGMC), was also proposed to help with communication improvement amongst ground nodes. Unlike approaches from the past which focused on either single weakest connection or sum of all connection, an mGMC approach focused on a certain number of connections which are weaker than the acceptable threshold. With a sum of all connections, UAVs could improve

connections which already have sufficiently high bandwidth good, thus not providing real benefit to the group of ground nodes. On the other hand, focusing on only weakest connection could result in UAV focusing on single weak connection, rather than improving a group of better, but still not that good connections, resulting in an overall worse improvement to the group. MGMC addresses those issues by ignoring connections of sufficient quality, while considering improvements to all weak connections at the same time. Additionally mGMC in the future, could be used to determine a sufficient number of UAVs for relay mission.

7.1.2 Scenario II: Unknown Urban Environments

The model used in Chapter 4 still required that an urban environment fits into one of the predefined types. This limited implementation of UAV relay in a realistic urban environment. The work in Chapter 5 tried to fully predict the communication model by using a machine learning technique, a Gaussian Process (GP).

The GP is a non-parametric machine learning technique which is useful for regression when the input is a random variable. With the GP technique, signal strength between the air and the ground could be fully predicted as a continuous function of a position of the UAV. However, the mobility of the ground nodes had to be sacrificed due to the high computational and data collection time demands. It was shown that the GP does not reach as good a performance as model-based networking. However, it was better than flying with the wrong model or no communication model at all. With GP approach it was possible for the UAV to perform relay missions with no knowledge about urban environments. In real life, UAV relay missions can often be performed at last minute notice, and good knowledge about mission environment may not be available. GP approach can mitigate that thus it is possible for UAVs relays to perform missions where previously not possible. Implementation and analysis of the GP in the urban environment was the main contribution of this chapter.

7.1.3 Scenario III: Partially Known Urban Environments

Fully predictable communication strength required a good knowledge of the urban environment, which is usually not the case. In Chapter 4, the level of the knowledge requirement was reduced by using a probabilistic-based approach with a learning-based approach.

The parameters of the probabilistic model were different depending on one of the four urban environment types. Depending on the parameters, the prediction of the signal strength between arbitrary positions in the air and on the ground changed. The neural network (NN) learning based approach was used to predict one of the four of urban environment types based on the signal strength received by the UAV and angle between air and ground vehicles. Neural Network was adopted to reduce data volume and computational time required. By combining this model with the NMPC controller, it was

shown that performance could quickly reach a similar level to model-based wireless signal strength prediction. It is worth noting that, this approach can achieve similar level performance with significantly reduced required knowledge of the urban environment and reduced computational cost. Both advantages mean UAV relay missions can be performed in more environments than ever before. Moreover, this is a completely novel and unique approach to predicting communication strength.

7.1.4 Experiment in Unknown Simulated Urban Environment

To prove the practicality of the GP approach proposed in Chapter 5, the performance of the experiment was reported in Chapter 6.

In this experiment, a quadrotor UAV was acting as a communication relay between two stationary ground nodes in an urban environment. The UAV would first fly in a back and forth pattern and collect RSSI signal to create a communication map. Based on this map, an optimal UAV position was chosen where the sum of RSSI was the highest. To simulate the city, a set of water containers was used as water is a great 2.4 GHz wireless signal absorber. For the need of the chapter, a new communication model was created to compare against the GP approach. It was shown that position created based on the GP communication map is as good or better than the one based on the model. it was also shown that the GP communication map could be created relatively fast, in about 11.4 seconds. With such a small computational time the practicality of the GP based approach is proven.

This experiment was the first attempt to simulate an urban environment realistically in the field with stationary ground nodes.

7.2 Future Work

Three challenges should be addressed in the future: i) computational time and quality of prediction with the GP, ii) the efficiency of the NMPC based path planner, iii) and improving the prediction of Neural Network.

The GP is a promising technique to solve this problem, however, it requires high computational time and sometimes outputs a low quality of prediction.

Computational time is dependant on the number of collected data points, i.e. the more data points collected, the slower the computation. There exist solutions which should be explored to address this problem. For example, sparse GP approach attempts to group points with similar values into a single entry to GP, thus reducing the computational load significantly [125].

Another issue stems from how the GP fits the data. Sudden changes in predictions are treated as errors and are either over or underestimated. To solve this issue, a GP mixtures approach could be used [126], where multiple GPs are computed at the same time and combined via separate covariance functions.

The performance of NMPC based path planners could also be improved. The NMPC based path planner only plans paths within a certain time horizon, which allows it to be more efficient. This often sacrifices optimality of the proposed path, due to insufficient horizon length. Approaches like a rapidly-exploring random tree (RRT) and RRT* also plan within a certain horizon time; however, they can plan paths with much longer time horizon with the same computational time. RRT approaches should be tested to determine whether they offer better performance.

Also, the NN has shown about 70 % accuracy in prediction of the urban environment type. Deep learning approaches (particularly long short-term memory (LSTMs) networks [127]) can offer much higher accuracy. LSTMs are particularly suitable to this problem as it is a time series problem, where inputs at the very beginning can hold significant information about prediction and should be remembered for future usage.

Finally, mobile grounds with the unknown urban environment should be considered. This would allow full measurement-based approach in a dynamic urban environment. This could be achieved with extending of GP to utilise more computational power or using other machine learning technique.

Bibliography

[1] Cory Dixon and Eric W. Frew. Optimizing cascaded chains of unmanned aircraft acting as communication relays. *IEEE Journal on Selected Areas in Communications*, 30(5):883–898, 2012.

[2] Daryl G. Press. Urban warfare: Options, problems and the future, Janaury 1999.

[3] Ilker Bekmezci, Ozgur Koray Sahingoz, and Samil Temel. Flying Ad-Hoc Networks (FANETs): A survey. *Ad Hoc Networks*, 11(3):1254–1270, 2013.

[4] Hyo-Sang Shin and Pau Segui-Gasco. UAV swarms: decision-making paradigms. *Encyclopedia of Aerospace Engineering*, pages 1–13, 2014.

[5] Edison Pignaton De Freitas, Tales Heimfarth, Ivayr Farah Netto, Carlos Eduardo Lino, Carlos Eduardo Pereira, Armando Morado Ferreira, Flávio Rech Wagner, and Tony Larsson. UAV relay network to support WSN connectivity. *2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, pages 309–314, 2010.

[6] Adrian Agogino, Chris HolmesParker, and Kagan Tumer. Evolving large scale UAV communication system. *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference - GECCO '12*, pages 1023–1300, 2012.

[7] Seungkeun Kim, Hyondong Oh, Jinyoung Suk, and Antonios Tsourdos. Coordinated trajectory planning for efficient communication relay using multiple UAVs. *Control Engineering Practice*, 29:42–49, 2014.

[8] Zhu Han, A Lee Swindlehurst, and K J Ray Liu. Optimization of MANET connectivity via smart deployment / movement of unmanned air vehicles. *IEEE Transactions on Vehicular Technology*, 58(7):3533–3546, 2009.

[9] Pawel Ladosz, Hyondong Oh, and Wen-Hua Chen. Trajectory planning for communication relay unmanned aerial vehicles in urban dynamic environments. *Journal of Intelligent & Robotic Systems*, 89(1):7–25, Jan 2018.

[10] Alexandra Grancharova, Esten Ingar Grotli, Dac Tu Ho, and Tor Arne Johansen. UAVs Trajectory Planning by Distributed MPC under Radio Communication Path Loss Constraints. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 79(1):115–134, 2014.

[11] Akrama Al-Hourani, Sithamparanathan Kandeepan, and Simon Lardner. Optimal LAP altitude for maximum coverage. *IEEE Wireless Communications Letters*, 3(6):569–572, 2014.

[12] Anthony J. Carfang, Neeti Wagle, and Eric W. Frew. Improving data ferrying by iteratively learning the radio frequency environment. *IEEE International Conference on Intelligent Robots and Systems*, pages 1182–1188, 2014.

[13] Pawel Ladosz, Hyondong Oh, and Wen-Hua Chen. Prediction of air-to-ground communication strength for relay uav trajectory planner in urban environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems 2017, IROS 2017*, 2017.

[14] Hang Zhao, Rimma Mayzus, Shu Sun, Mathew Samimi, Jocelyn K. Schulz, Yaniv Azar, Kevin Wang, George N. Wong, Felix Gutierrez, and Theodore S. Rappaport. 28 GHz millimeter wave cellular communication measurements for reflection and penetration loss in and around buildings in New York city. *IEEE International Conference on Communications*, pages 5163–5167, 2013.

[15] G Durgin, Ts Rappaport, and H Xu. Measurements and models for radio path loss and penetration loss in and around homes and Trees at 5.85 GHz. *IEEE Transactions on Communications*, (11):1484–1496, 1998.

[16] Kate A Remley, Senior Member, Galen Koepke, Christopher L Holloway, Chriss A Grosvenor, Dennis Camell, John Ladbury, Robert T Johnk, and William F Young. Radio-Wave Propagation Into Large Building Structures Part 2 : Characterization of Multipath. *IEEE Transactions on Antennas and Propagation*, 58(April):1290–1301, 2010.

[17] William J. Tanis and Glenn J. Pilato. BUILDING PENETRATION CHARACTERISTICS 0F 880 MHz AND 1922 MHz RADIO WAVES. 1993.

[18] A. Davidson. Measurement of building penetration into medium buildings at 900 and 1500 mhz. *IEEE Transactions on Vehicular Technology*, 46(1):161–168, 1997.

[19] David W. Matolak, Kate A. Remley, Camillo Gentile, Christopher L. Holloway, Qiong Wu, and Qian Zhang. Peer-to-peer urban channel characteristics for two public-safety frequency bands. *IEEE Antennas and Propagation Magazine*, 56(5):101–115, 2014.

[20] C.L. Holloway, G. Koepke, D. Camell, W.F. Young, and K.A. Remley. Propagation measurements before, during, and after the collapse of three large public buildings. *Antennas and Propagation Magazine, IEEE*, 56(3):16–36, 2014.

[21] Christopher L. Holloway, Galen H. Koepke, Dennis G. Camell, Catherine A. Remley, Dylan F. Williams, Susan Schima, Seturnino Canales, and Douglas T. Tamura. Propagation and Detection of Radio Signals Before, During, and After the Implosion of a Large Convention Center. *Technical Note (NIST TN)*, 2005.

[22] Christopher L. Holloway, Galen H. Koepke, Dennis G. Camell, Catherine A. Remley, Dylan F. Williams, Susan Schima, Seturnino Canales, and Douglas T. Tamura. Propagation and detection of radio signals before, during, and after the implosion of a large sports stadium (veterans stadium in philadelphia). *Technical Note (NIST TN)*, 2005.

[23] Christopher L Holloway, Galen Koepke, Dennis Camell, Kate A Remley, Dylan F Williams, Susan A Schima, Seturnino Canales, and Douglas T Tamura. Propagation and detection of radio signals before, during, and after the implosion of a 13-story apartment building. *Technical Note (NIST TN)*, 2005.

[24] George R. Maccartney, Junhong Zhang, Shuai Nie, and Theodore S. Rappaport. Path loss models for 5G millimeter wave propagation channels in urban microcells. *GLOBECOM - IEEE Global Telecommunications Conference*, pages 3948–3953, 2013.

[25] James N. Murdock, Eshar Ben-Dor, Yijun Qiao, Jonathan I. Tamir, and Theodore S. Rappaport. A 38 GHz cellular outage study for an urban outdoor campus environment. *IEEE Wireless Communications and Networking Conference, WCNC*, pages 3085–3090, 2012.

[26] Jorgen B Andersen and Theodore S Rappaport. Propagation measurements and models for wireless communications channels. *IEEE Communications Magazine*, 33(1):42–49, 1995.

[27] Theodore S. Rappaport, George MacCartney, Mathew Samimi, and Shu Sun. Wideband Millimeter-Wave Propagation Measurements and Channel Models for Future Wireless Communication System Design. *IEEE Transactions on Communications*, PP(99):1–1, 2015.

[28] Yaniv Azar, George N. Wong, Kevin Wang, Rimma Mayzus, Jocelyn K. Schulz, Hang Zhao, Felix Gutierrez, Duckdong Hwang, and Theodore S. Rappaport. 28 GHz propagation measurements for outdoor cellular communications using steerable beam antennas in New York city. *IEEE International Conference on Communications*, pages 5143–5147, 2013.

[29] Juha Laurila, Kimmo Kalliola, Martin Toeltsch, Klaus Hugl, Pertti Vainikainen, and Ernst Bonek. Wide-band 3-D characterization of mobile radio channels in urban environment. *IEEE Transactions on Antennas and Propagation*, 50(2):233–243, 2002.

[30] John S. Otto, Fabián E. Bustamante, and Randall A. Berry. Down the block and around the corner: The impact of radio propagation on inter-vehicle wireless communication. *Proceedings - International Conference on Distributed Computing Systems*, pages 605–614, 2009.

[31] Johan Karedal, Nicolai Czink, Alexander Paier, Fredrik Tufvesson, and Andreas F. Molisch. Path loss modeling for vehicle-to-vehicle communications. *IEEE Transactions on Vehicular Technology*, 60(1):323–328, 2011.

[32] Jianhui Wu and Dongfeng Yuan. Propagation measurements and modeling in Jinan city. *Personal, Indoor and Mobile Radio Communications, 1998. The Ninth IEEE International Symposium on*, 3:1157–1160 vol.3, 1998.

[33] V.S. Abhayawardhana, I.J. Wassell, D. Crosby, M.P. Sellars, and M.G. Brown. Comparison of Empirical Propagation Path Loss Models for Fixed Wireless Access Systems. *2005 IEEE 61st Vehicular Technology Conference*, 1(c):73–77, 2005.

[34] Roberto Magán-Carrión, Rafael A. Rodríguez-Gómez, José Camacho, and Pedro García-Teodoro. Optimal relay placement in multi-hop wireless networks. *Ad Hoc Networks*, 46:23–36, 2014.

[35] Oleg Burdakov, Patrick Doherty, Kaj Holmberg, and Per Magnus Olsson. Optimal placement of UV-based communications relay nodes. *Journal of Global Optimization*, 48(4):511–531, 2010.

[36] R. Irem Bor-Yaliniz, Amr El-Keyi, and Halim Yanikomeroglu. Efficient 3-D placement of an aerial base station in next generation cellular networks. *2016 IEEE International Conference on Communications, ICC 2016*, 2016.

[37] Mason Thammawichai, Sujit P. Baliyarasimhuni, Eric C. Kerrigan, and Joao B. Sousa. Optimizing Communication and Computation for Multi-UAV Information Gathering Applications. *IEEE Transactions on Aerospace and Electronic Systems*, 54(2):601–615, 2017.

[38] S. Wu, C. X. Wang, e. H. M. Aggoune, M. M. Alwakeel, and X. H. You. A general 3d non-stationary 5g wireless channel model. *IEEE Transactions on Communications*, pages 3065–3078, 2017.

[39] D.H. Choi, S.H Kim, and D.K. Sung. Energy-Efficient Maneuvering and Communication of a Single UAV-Based Relay. *Aerospace and Electronic Systems, IEEE Transactions on*, 50(3):2320/2327, 2014.

[40] Pengcheng Zhan, Kai Yu, and a. Lee Swindlehurst. Wireless relay communications with unmanned aerial vehicles: Performance and optimization. *IEEE Transactions on Aerospace and Electronic Systems*, 47(3):2068–2085, 2011.

[41] Jesús Pérez, Jesús Ibáñez, Luis Vielva, David J. Pérez-Blanco, and Ignacio Santamaría. Tight closed-form approximation for the ergodic capacity of orthogonal STBC. *IEEE Transactions on Wireless Communications*, 6(2):452–457, 2007.

[42] John Magliacane. Splat!, 2018.

[43] Esten Ingar Grotli and Tor Arne Johansen. Path planning for UAVs under communication constraints using SPLAT! and MILP. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 65(1-4):265–282, 2012.

[44] E. I. Grtli and T. A. Johansen. Task assignment for cooperating uavs under radio propagation path loss constraints. In *2012 American Control Conference (ACC)*, pages 3278–3283, June 2012.

[45] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Merouane Debbah. Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage. *IEEE Communications Letters*, pages 1647–1650, 2016.

[46] Akram Al-Hourani, Sithamparanathan Kandeepan, and Abbas Jamalipour. Modeling air-to-ground path loss for low altitude platforms in urban environments. *2014 IEEE Global Communications Conference*, pages 2898–2904, 2014.

[47] Bishwarup Mondal, Timothy A. Thomas, Eugene Visotsky, Frederick W. Vook, Amitava Ghosh, Young Han Nam, Yang Li, Jianzhong Charlie Zhang, Min Zhang, Qinglin Luo, Yuichi Kakishima, and Koshiro Kitao. 3D channel model in 3GPP. *IEEE Communications Magazine*, 53(3):16–23, 2015.

[48] Deok Jin Lee and Klas Andersson. Hybrid control of long-endurance aerial robotic vehicles for wireless sensor networks. *International Journal of Advanced Robotic Systems*, 8(2):101–113, 2011.

[49] Abbas Chamseddine, Guillaume Charland-Arcand, Ouassima Akhrif, Samuel Gagne, Francois Gagnon, and Denis Couillard. Optimal position seeking for unmanned aerial vehicle communication relay using only signal strength and angle of arrival. *Proceedings of the IEEE Conference on Decision and Control*, 2015-Febru(February):976–981, 2015.

[50] Michael M. Zavlanos, Alejandro Ribeiro, and George J. Pappas. Network integrity in mobile robotic networks. *IEEE Transactions on Automatic Control*, 58(1):3–18, 2013.

[51] Stephanie Gil, Swarun Kumar, Dina Katabi, and Daniela Rus. Adaptive communication in multi-robot systems using directionality of signal strength. *The International Journal of Robotics Research*, 34(7):946–968, 2015.

[52] Mong-ying A Hsieh, Vijay Kumar, and Camillo J Taylor. Constructing Radio Signal Strength Maps with Multiple Robot. *Constructing radio signal strength maps with multiple robots*, pages 4184–4189, 2004.

[53] Anthony J. Carfang and Eric W. Frew. Real-time estimation of wireless ground-to-air communication parameters. *2012 International Conference on Computing, Networking and Communications, ICNC'12*, pages 975–979, 2012.

[54] Spencer Watza and Eric Frew. Hybrid RF propagation model using ITM and gaussian processes for communication-aware planning. *RSS 2017 RCW Workshop*, 2017.

[55] Jonathan Fink, Nathan Michael, Alex Kushleyev, and Vijay Kumar. Experimental characterization of radio signal propagation in indoor environments with application to estimation and control. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pages 2834–2839, 2009.

[56] Jonathan Fink and Vijay Kumar. Online methods for radio signal mapping with mobile robots. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1940–1945, 2010.

[57] Jonathan Fink, Alejandro Ribeiro, and Vijay Kumar. Robust Cyber-Physical Control of Mobility and Communication in Autonomous Robot Teams. *Proceedings of the IEEE*, 100(1):164–178, 2012.

[58] Sergio Caccamo, Ramviyas Parasuraman, Luigi Freda, Mario Gianni, and Petter Ögren. RCAMP: a resilient communication-Aware motion planner for mobile robots with autonomous repair of wireless connectivity. *IEEE/RSJ International Conference on Intelligent Robots and Systems 2017, IROS 2017*, pages 2010–2017, 2017.

[59] Yajun Gao, Haoyao Chen, Yanjie Li, Congyi Lyu, and Yunhui Liu. Autonomous Wi-Fi relay placement with mobile robots. *IEEE/ASME Transactions on Mechatronics*, 4435(c), 2017.

[60] Jacopo Banfi, Alberto Quattrini Li, Nicola Basilico, Ioannis Rekleitis, and Francesco Amigoni. Multirobot online construction of communication maps. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2577–2583, 2017.

[61] Yasamin Mostofi, Mehrzad Malmirchegini, and Alireza Ghaffarkhah. Estimation of communication signal strength in robotic networks. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1946–1951, 2010.

[62] Yasamin Mostofi and Pradeep Sen. Compressed mapping of communication signal strength. *Proceedings - IEEE Military Communications Conference MILCOM*, 2008.

[63] Yuan Yan and Yasamin Mostofi. Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments. *IEEE Transactions on Wireless Communications*, 12(4):1562–1572, 2013.

[64] Mehrzad Malmirchegini and Yasamin Mostofi. On the spatial predictability of communication channels. *IEEE Transactions on Wireless Communications*, 11(3):964–978, 2012.

[65] Y. Mostofi, a. Gonzalez-Ruiz, a. Gaffarkhah, and Ding Li. Characterization and modeling of wireless channels for networked robotic and control systems - a comprehensive overview. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4849–4854, 2009.

[66] Junting Chen, Uday Yatnalli, and David Gesbert. Learning radio maps for UAV-aided wireless networks: a segmented regression approach. *IEEE International Conference on Communications*, 2017.

[67] Evsen Yanmaz, Robert Kuschnig, and Christian Bettstetter. Achieving air-ground communications in 802.11 networks with three-dimensional aerial mobility. *Proceedings - IEEE INFOCOM*, pages 120–124, 2013.

[68] Bertold Van Der Bergh, Alessandro Chiumento, and Sofie Pollin. LTE in the sky: Trading off propagation benefits with interference costs for aerial nodes. *IEEE Communications Magazine*, 54(5):44–50, 2016.

[69] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[70] Eduardo Feo Flushing and Gianni a. Di Caro. A flow-based optimization model for throughput-oriented relay node placement in wireless sensor networks. *Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13*, page 632, 2013.

[71] Ahmed S. Ibrahim, Karim G. Seddik, and K. J. Ray Liu. Improving connectivity via relays deployment in wireless sensor networks. *GLOBECOM - IEEE Global Telecommunications Conference*, pages 1159–1163, 2007.

[72] Stephanie Gil, Mac Schwager, Brian J. Julian, and Daniela Rus. Optimizing communication in air-ground robot networks using decentralized control. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1964–1971, 2010.

[73] Byung Cheol Min, Yongho Kim, Sangjun Lee, Jin Woo Jung, and Eric T. Matson. Finding the optimal location and allocation of relay robots for building a rapid end-to-end wireless communication. *Ad Hoc Networks*, 39:23–44, 2015.

[74] D. G. Reina, H. Tawfik, and S. L. Toral. Multi-subpopulation evolutionary algorithms for coverage deployment of UAV-networks. *Ad Hoc Networks*, 68:16–32, 2018.

[75] K. Chandrashekar, M.R. Dekhordi, and J.S. Baras. Providing full connectivity in large ad-hoc networks by dynamic placement of aerial platforms. *IEEE MILCOM 2004. Military Communications Conference, 2004.*, 3:1429–1436, 2004.

[76] Arvind Merwaday and Ismail Guvenc. UAV assisted heterogeneous networks for public safety communications. *2015 IEEE Wireless Communications and Networking Conference Workshops, WCNC 2015*, pages 329–334, 2015.

[77] Junting Chen and David Gesbert. Optimal positioning of flying relays for wireless networks: A LOS map approach. *IEEE International Conference on Communications*, 2017.

[78] Jiangbin Lyu, Yong Zeng, Rui Zhang, and Teng Joon Lim. Placement Optimization of UAV-Mounted Mobile Base Stations. *IEEE Communications Letters*, 21(3):604–607, 2017.

[79] Milen Nikolov and Zygmunt J. Haas. Relay placement in wireless networks: Minimizing communication cost. *IEEE Transactions on Wireless Communications*, 15(5):3587–3602, 2016.

[80] Wang Xu, Wu, Daneshmand, Liu. A data privacy protective mechanism for WBAN. *Wireless Communications and Mobile Computing*, pages 421–430, 2015.

[81] Eduardo Feo Flushing, Luca M. Gambardella, and Gianni A. Di Caro. Simultaneous task allocation, data routing, and transmission scheduling in mobile multi-robot teams. *IEEE International Conference on Intelligent Robots and Systems*, 2017-September:1861–1868, 2017.

[82] Yong Zeng and Rui Zhang. Energy-Efficient UAV Communication with Trajectory Optimization. *IEEE Transactions on Wireless Communications*, pages 1–30, 2016.

[83] Dae Hyung Choi, Byoung Hoon Jung, and Dan Keun Sung. Low-complexity Maneuvering Control of a UAV-based Relay without Location Information of Mobile Ground Nodes. *Computers and Communication (ISCC), 2014 IEEE Symposium on*, pages 1 – 6, 2014.

[84] L. Singh and J. Fuller. Trajectory generation for a uav in urban terrain, using nonlinear mpc. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, volume 3, pages 2301–2308 vol.3, June 2001.

[85] Y. Kang and J. K. Hedrick. Linear tracking for a fixed-wing uav using nonlinear model predictive control. *IEEE Transactions on Control Systems Technology*, 17(5):1202–1210, Sep. 2009.

[86] Vitaly Shaferman and Tal Shima. Unmanned Aerial Vehicles Cooperative Tracking of Moving Ground Target in Urban Environments. *Journal of Guidance, Control, and Dynamics*, 31(5):1360–1371, 2008.

[87] T. Templeton, D. H. Shim, C. Geyer, and S. S. Sastry. Autonomous vision-based landing and terrain mapping using an mpc-controlled unmanned rotorcraft. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 1349–1356, April 2007.

[88] X. Wang, V. Yadav, and S. N. Balakrishnan. Cooperative uav formation flying with obstacle/collision avoidance. *IEEE Transactions on Control Systems Technology*, 15(4):672–679, July 2007.

[89] M. Rosalie, J. E. Dentler, G. Danoy, P. Bouvry, S. Kannan, M. A. Olivares-Mendez, and H. Voos. Area exploration with a swarm of uavs combining deterministic chaotic ant colony mobility with position mpc. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1392–1397, June 2017.

[90] Dac-Tu Ho, Esten Ingar Grotli, P. B. Sujit, Tor Arne Johansen, and Joao Borges De Sousa. Performance evaluation of cooperative relay and particle swarm optimization path planning for UAV and wireless sensor network. *2013 IEEE Globecom Workshops*, pages 1403–1408, 2013.

[91] Philip B. Charlesworth. A non-cooperative game to coordinate the coverage of two communications UAVs. *Proceedings - IEEE Military Communications Conference MILCOM*, pages 668–673, 2013.

[92] Alexandros Giagkos, Myra S. Wilson, Elio Tuci, and Philip B. Charlesworth. Comparing approaches for coordination of autonomous communications UAVs. *2016 International Conference on Unmanned Aircraft Systems, ICUAS 2016*, pages 1131–1139, 2016.

[93] Hee-Tae Roh and Jang-Won Lee. Communication-Aware Position Control for Mobile Nodes in Vehicular Networks. *IEEE Journal on Selected Areas in Communications*, 29(1):173–186, 2011.

[94] Vinod Ramaswamy, Sangwoo Moon, Eric W. Frew, and Nisar Ahmed. Mutual Information based communication aware path planning: A game theoretic perspective. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1823–1828, 2016.

[95] Youngdong Choi, Meir Pachter, and David Jacques. Optimal relay UAV guidance-A new differential game. *Proceedings of the IEEE Conference on Decision and Control*, pages 1024–1029, 2011.

[96] P. Basu, J. Redi, and V. Shurbanov. Coordinated flocking of UAVs for improved connectivity of mobile ground nodes. *IEEE MILCOM 2004. Military Communications Conference, 2004.*, 3:1628–1634, 2004.

[97] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Merouane Debbah. Mobile Internet of Things: Can UAVs Provide an Energy-Efficient Mobile Architecture? 2016.

[98] S Kim, P Silson, A Tsourdos, and M Shanmugavel. Dubins path planning of multiple unmanned airborne vehicles for communication relay. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 225(1):12–25, 2011.

[99] Stephanie Gil, Dan Feldman, and Daniela Rus. Communication coverage for independently moving robots. *IEEE International Conference on Intelligent Robots and Systems*, 1:4865–4872, 2012.

[100] Guangchi Zhang, Qingqing Wu, Miao Cui, and Rui Zhang. Securing UAV Communications via Trajectory Optimization. *2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings*, 2018-January:1–6, 2018.

[101] Sameera S. Ponda, Luke B. Johnson, Andrew N. Kopeikin, Han Lim Choi, and Jonathan P. How. Distributed planning strategies to ensure network connectivity for dynamic heterogeneous teams. *IEEE Journal on Selected Areas in Communications*, 30(5):861–869, 2012.

[102] John Chinneck. Practical optimization: A gentle introduction. [online], 12 2015. http://www.sce.carleton.ca/faculty/chinneck/po.html.

[103] Pieter Tjerk De Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, 2005.

[104] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, Dec 1943.

[105] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[106] Martin Fodslette Mller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525 – 533, 1993.

[107] David Legland. geom3d, June 2009.

[108] R. Sedgewick and K. Wayne. Minimum spanning trees, 2015.

[109] J Kennedy and R Eberhart. Particle swarm optimization. *Neural Networks, 1995. Proceedings., IEEE International Conference on*, 4:1942–1948 vol.4, 1995.

[110] Seungkeun Kim, Hyondong Oh, and Antonios Tsourdos. Nonlinear model predictive coordinated standoff tracking of a moving ground vehicle. *Journal of Guidance Control and Dynamics*, 36:557–566, August 2013.

[111] Kishore Mehrotra and Pravas R. Mahapatra. A jerk model for tracking highly maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, 33(4):1094–1105, 1997.

[112] ITU-R. Rec. p.1410-2 propagation data and prediction methods for the design of terrestrial broadband millimetric radio access systems. *P Series Radiowave propagation*, 2003.

[113] Christopher J. Paciorek and Mark J. Schervish. Nonstationary covariance functions for gaussian process regression. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 273–280. MIT Press, 2004.

[114] Thomas Schwengler. Wireless & cellular communications class notes for tlen-5510 fall 2017, chapter 4. `http://morse.colorado.edu/~tlen5510/text/classweb.html`, 2017. accessed 19/12/2017.

[115] Brett Meador. A Survey of Computer Network Topology and Analysis Examples. pages 1–79, 2008.

[116] Open Mesh. B.a.t.m.a.n. advanced documentation overview.

[117] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke. Ieee 802.11s: The wlan mesh standard. *IEEE Wireless Communications*, 17(1):104–111, February 2010.

[118] IEEE. Ieee 802.15.4-2015/cor 1-2018 - ieee standard for low-rate wireless networks corrigendum 1.

[119] Iván Armuelles Voinov, AidelenChung Cedeño, Joaquín Chung, and Grace González. A Performance Analysis of Wireless Mesh Networks Implementations Based on Open Source Software. *Open Source Software: Mobile Open Source Technologies*, 427:107–110, 2014.

[120] Adam Stambler. rosserial_xbee, 2018.

[121] Vladimir Ermakov. Mavros, 2019.

[122] Pawel Ladosz, Matthew Coombes, Jean Smith, and Michael Hutchinson. *A Generic ROS Based System for Rapid Development and Testing of Algorithms for Autonomous Ground and Aerial Vehicles*, pages 113–153. Springer International Publishing, Cham, 2019.

[123] Umair Mujtaba Qureshi, Zuneera Aziz, Faisal Karim Shaikh, Zuneera Aziz, Syed M.Zafi S. Shah, Syed M.Zafi S. Shah, Adil A. Sheikh, Emad Felemban, and Saad Bin Qaisar. RF path and absorption loss estimation for underwaterwireless sensor networks in differentwater environments. *Sensors (Switzerland)*, 16(6), 2016.

[124] Carl E. Rasmussen and Zoubin Ghahramani. Infinite mixtures of gaussian process experts. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 881–888. MIT Press, 2002.

[125] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006.

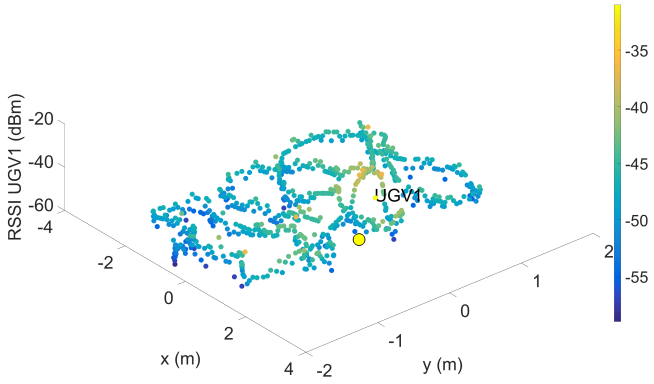
[126] Carl E Rasmussen and Zoubin Ghahramani. Infinite mixtures of gaussian process experts. In *Advances in neural information processing systems*, pages 881–888, 2002.

[127] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

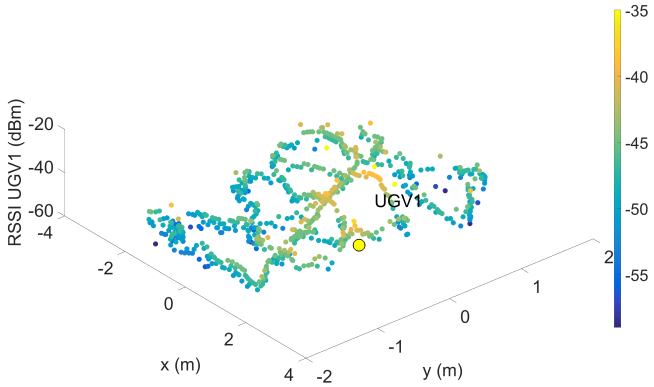
Appendix A

Appendix A

A.1 Raw Data for Two UGV Experiment

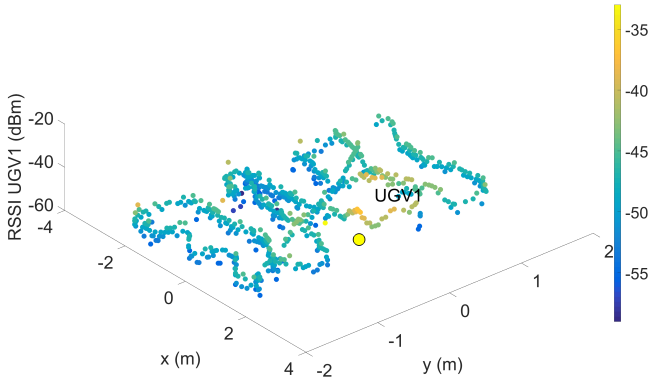


(a) Raw experimental data for UGV 1 and trial 1

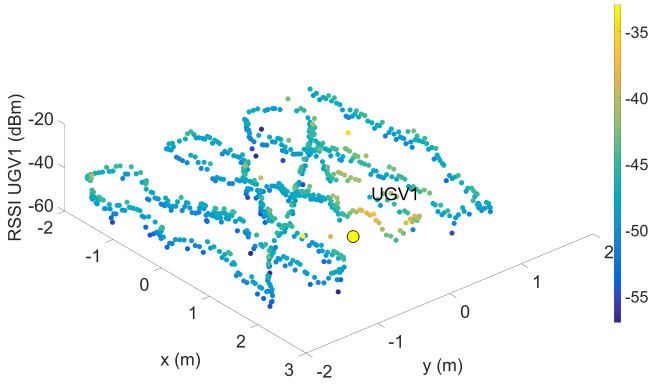


(b) Raw experimental data for UGV 1 and trial 2

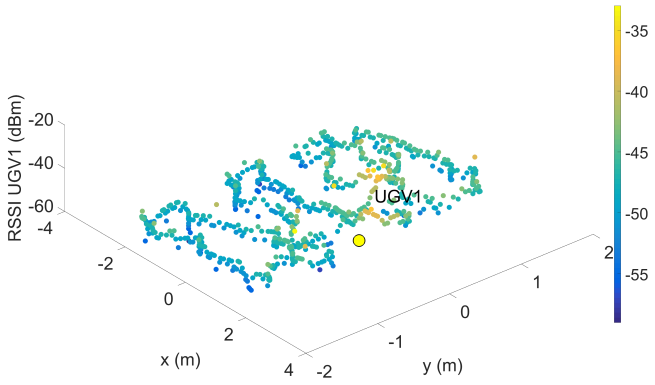
Figure A.1: Summary of raw experimental data for UGV 2



(c) Raw experimental data for UGV 1 and trial 3

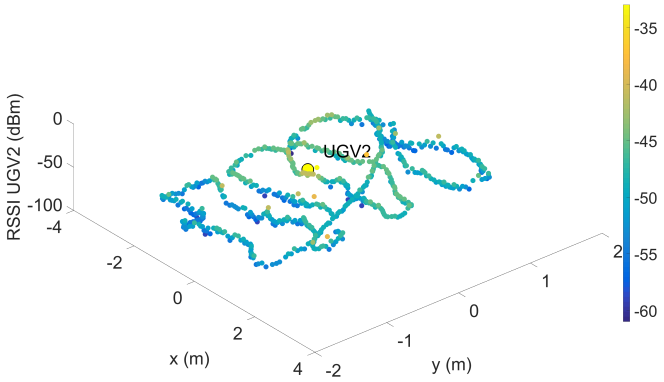


(d) Raw experimental data for UGV 1 and trial 4

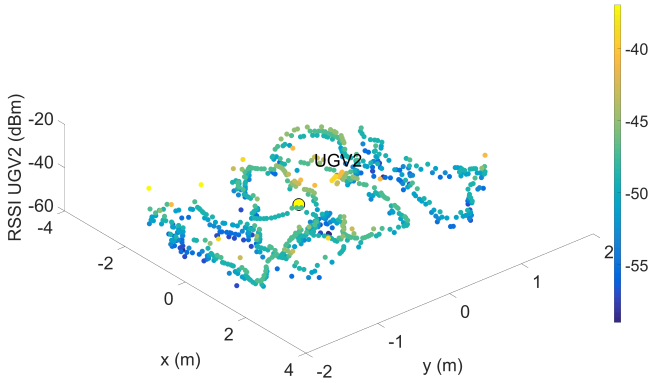


(e) Raw experimental data for UGV 1 and trial 5

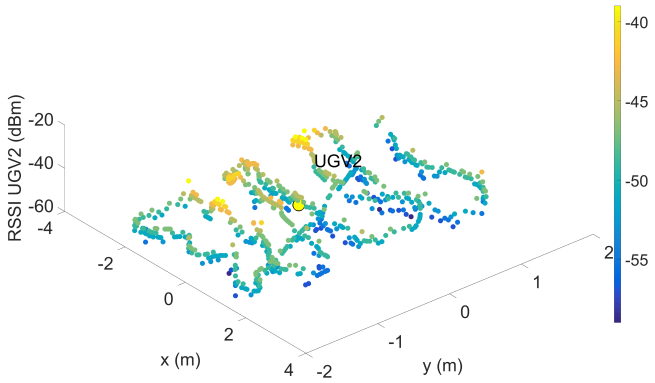
Figure A.1: Summary of raw experimental data for UGV 1



(a) Raw experimental data for UGV 2 and trial 1

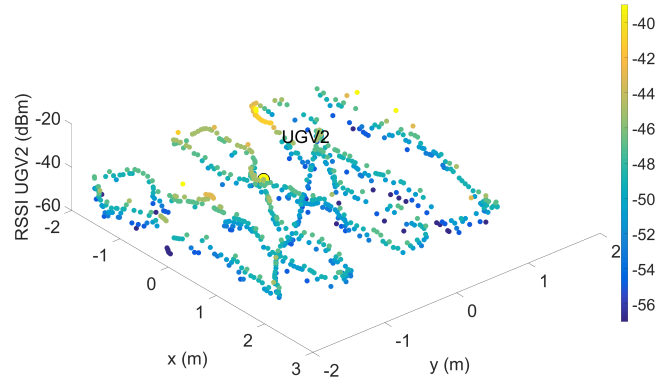


(b) Raw experimental data for UGV 2 and trial 2

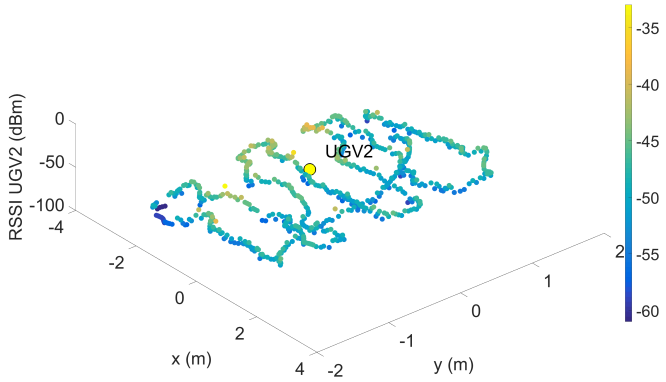


(c) Raw experimental data for UGV 2 and trial 3

Figure A.2: Summary of raw experimental data for UGV 2



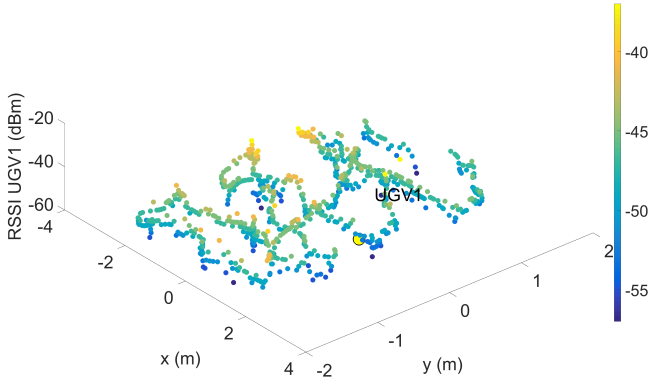
(d) Raw experimental data for UGV 2 and trial 4



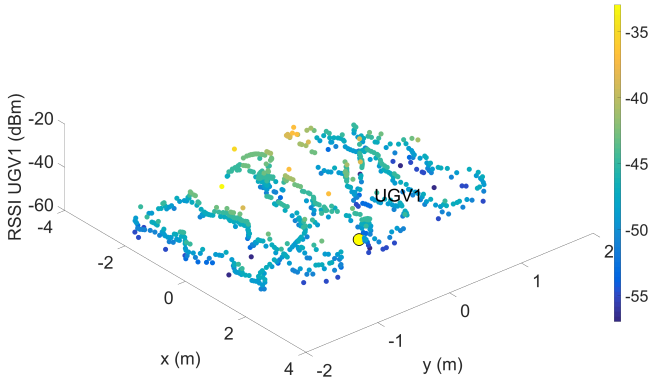
(e) Raw experimental data for UGV 2 and trial 5

Figure A.2: Summary of raw experimental data for UGV 2

A.2 Raw Data for Two UGV One Building Experiment

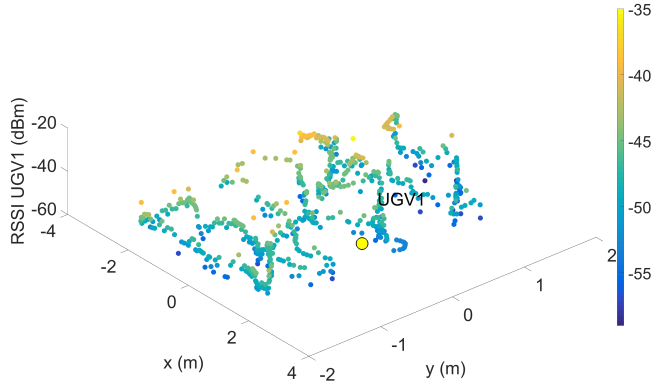


(a) Raw experimental data for UGV 2 and trial 1

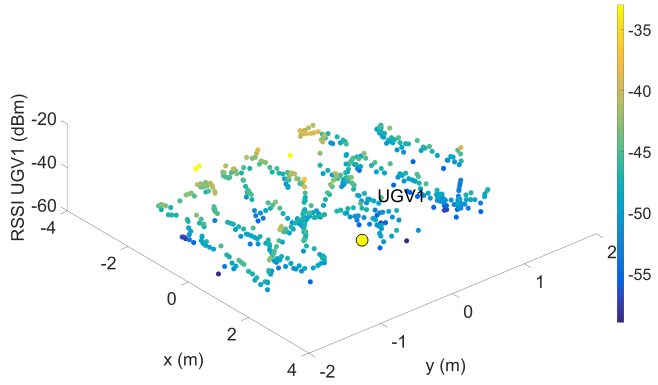


(b) Raw experimental data for UGV 1 and trial 2

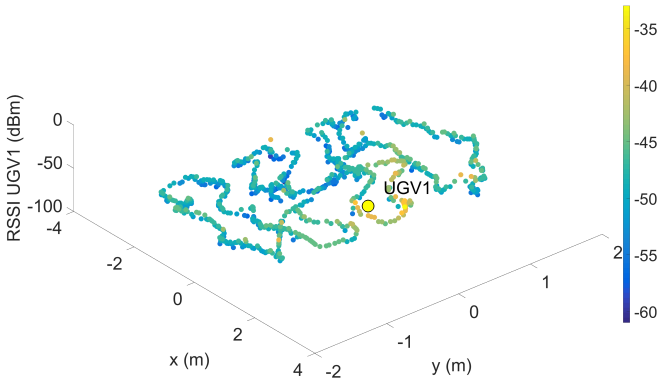
Figure A.3: Summary of raw experimental data for UGV 1 with one building



(c) Raw experimental data for UGV 1 and trial 3

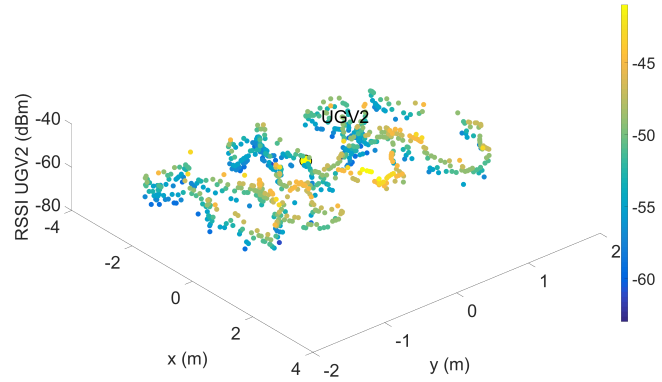


(d) Raw experimental data for UGV 1 and trial 4

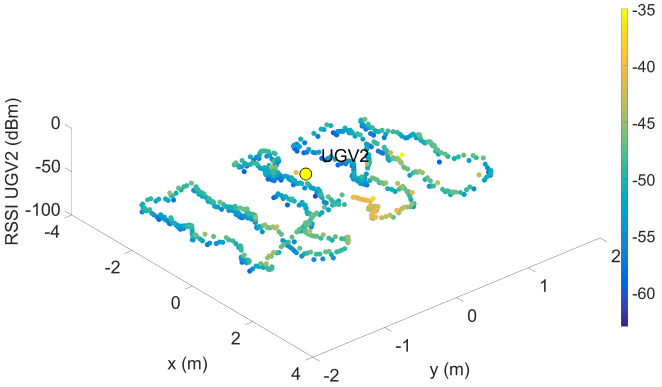


(e) Raw experimental data for UGV 1 and trial 5

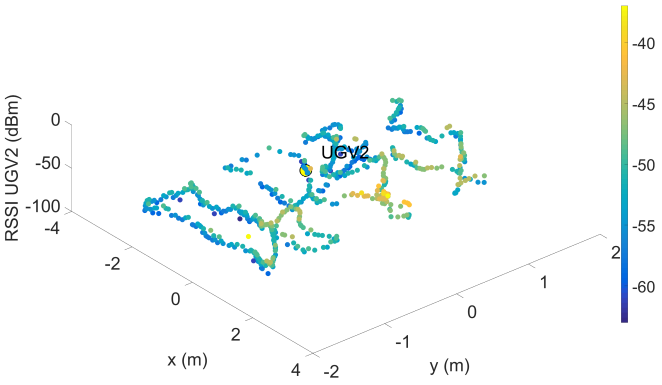
Figure A.3: Summary of raw experimental data for UGV 1 with one building



(a) Raw experimental data for UGV 2 and trial 1

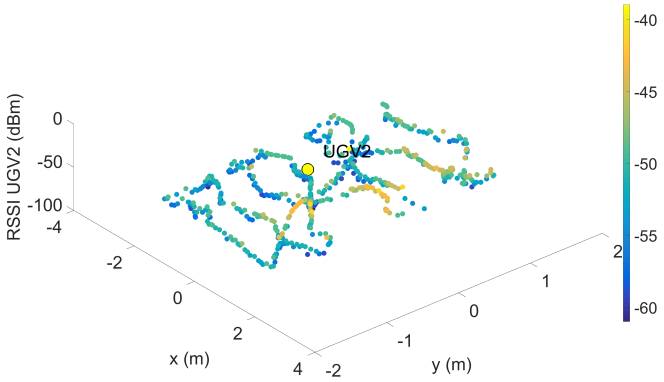


(b) Raw experimental data for UGV 2 and trial 2

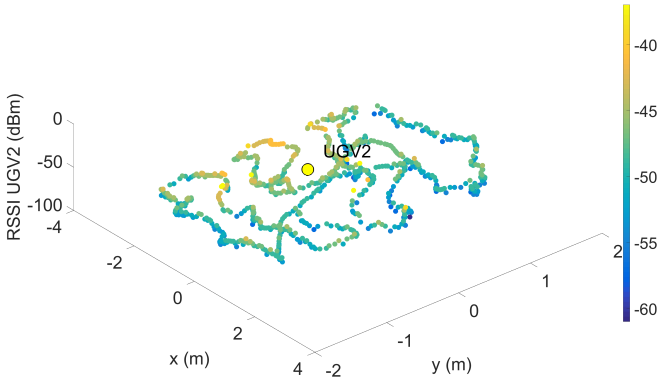


(c) Raw experimental data for UGV 2 and trial 3

Figure A.4: Summary of raw experimental data for UGV 2 with one building



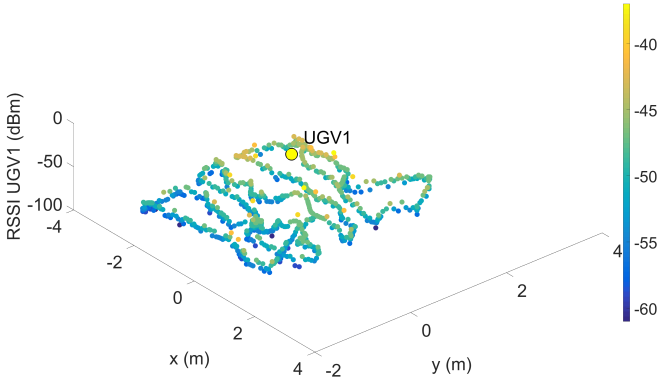
(d) Raw experimental data for UGV 2 and trial 4



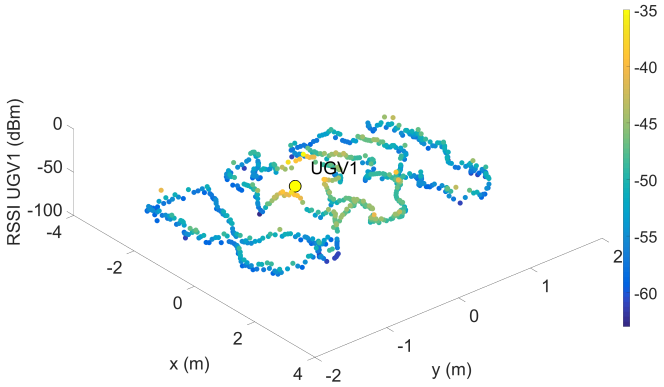
(e) Raw experimental data for UGV 2 and trial 5

Figure A.4: Summary of raw experimental data for UGV 2 with one building

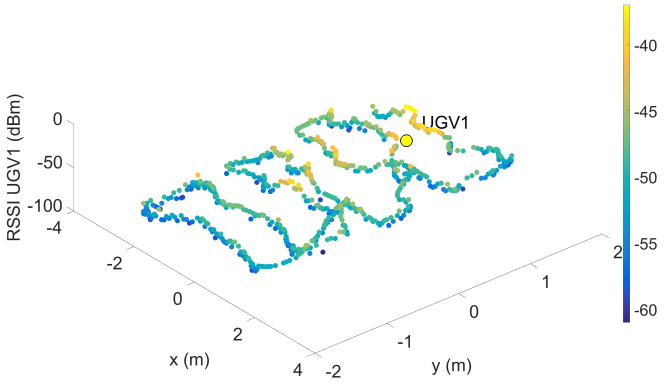
A.3 Raw Data for Two UGV Full City Experiment



(a) Raw experimental data for UGV 1 and trial 1

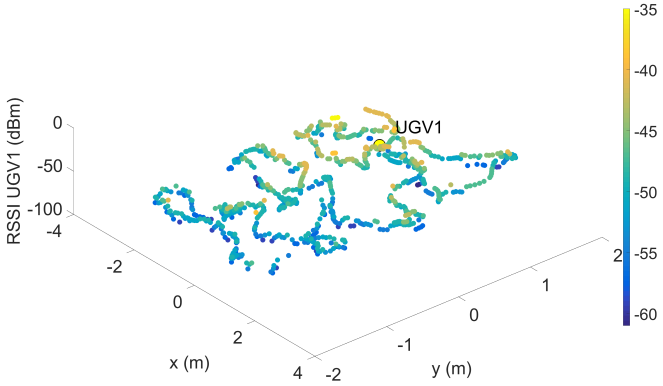


(b) Raw experimental data for UGV 1 and trial 2

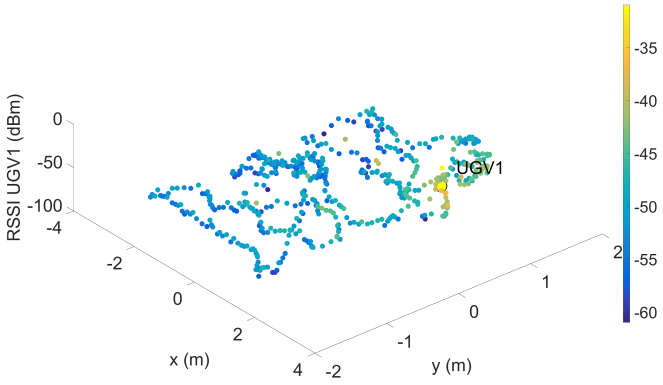


(c) Raw experimental data for UGV 1 and trial 3

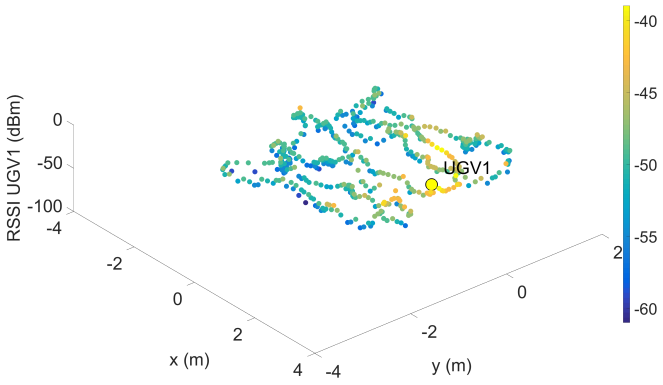
Figure A.5: Summary of raw experimental data for UGV 1 with full city



(d) Raw experimental data for UGV 1 and trial 4

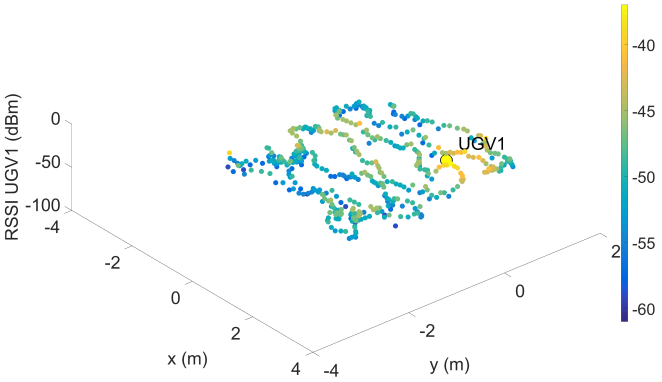


(e) Raw experimental data for UGV 1 and trial 5

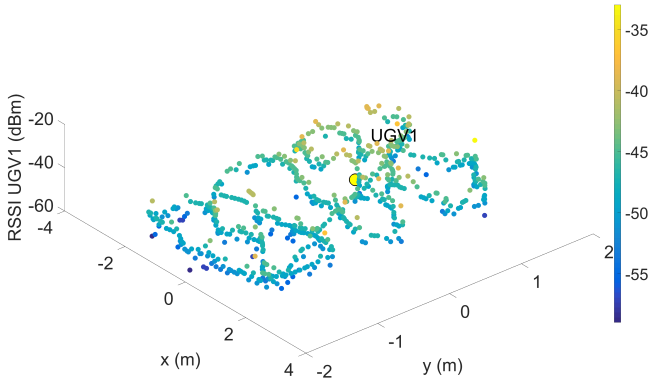


(f) Raw experimental data for UGV 1 and trial 6

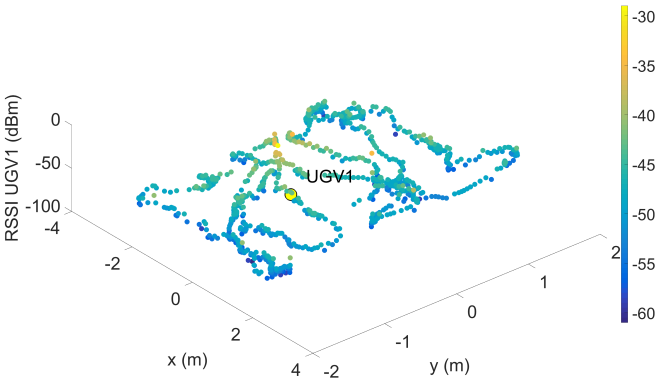
Figure A.5: Summary of raw experimental data for UGV 1 with full city



(g) Raw experimental data for UGV 2 and trial 7

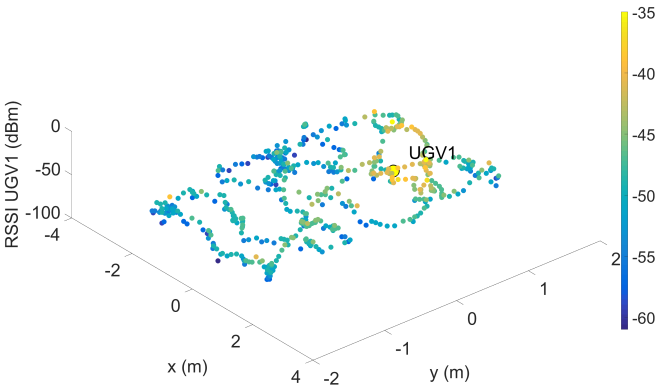


(h) Raw experimental data for UGV 1 and trial 8



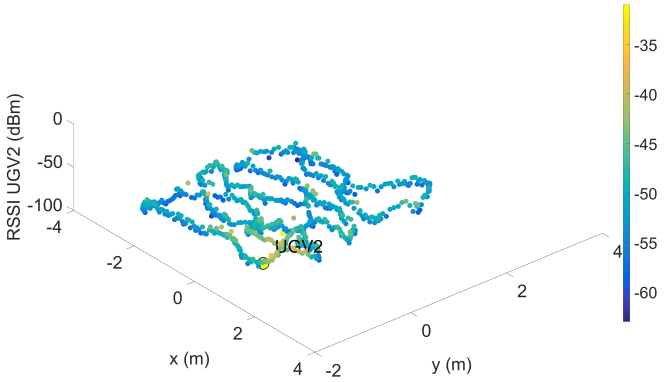
(i) Raw experimental data for UGV 1 and trial 9

Figure A.5: Summary of raw experimental data for UGV 1 with full city

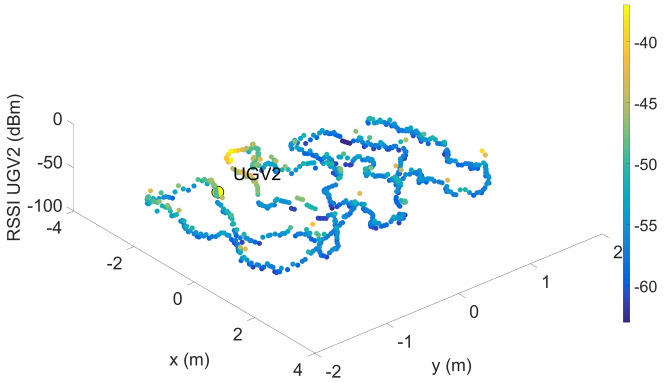


(j) Raw experimental data for UGV 1 and trial 10

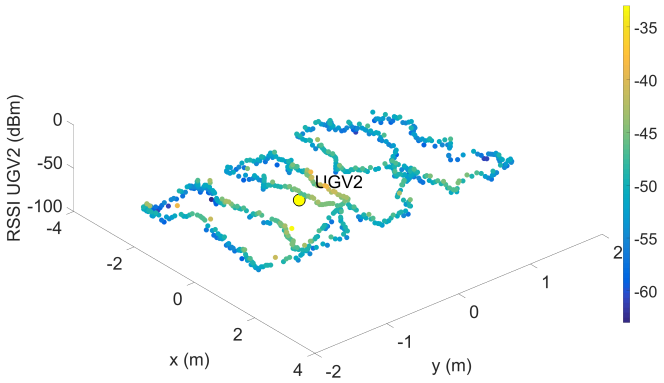
Figure A.5: Summary of raw experimental data for UGV 1 with full city



(a) Raw experimental data for UGV 2 and trial 1

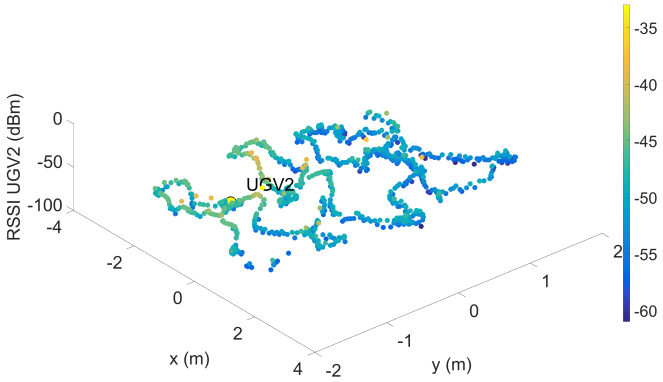


(b) Raw experimental data for UGV 2 and trial 2

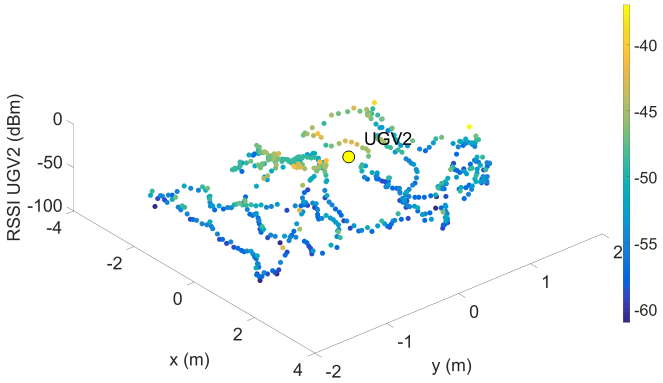


(c) Raw experimental data for UGV 2 and trial 3

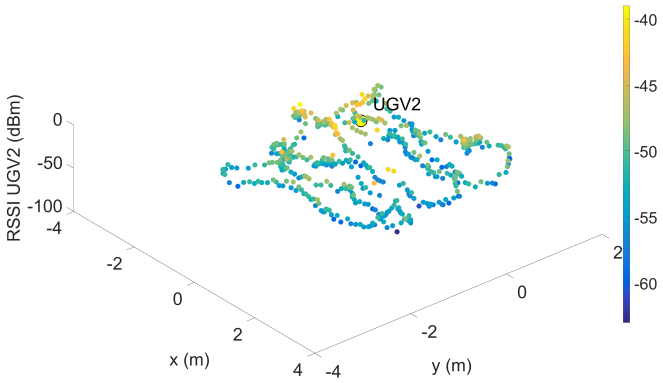
Figure A.6: Summary of raw experimental data for UGV 2 with full city



(d) Raw experimental data for UGV 2 and trial 4

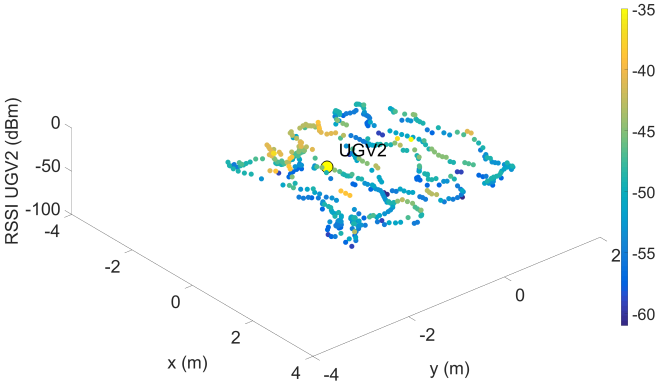


(e) Raw experimental data for UGV 2 and trial 5

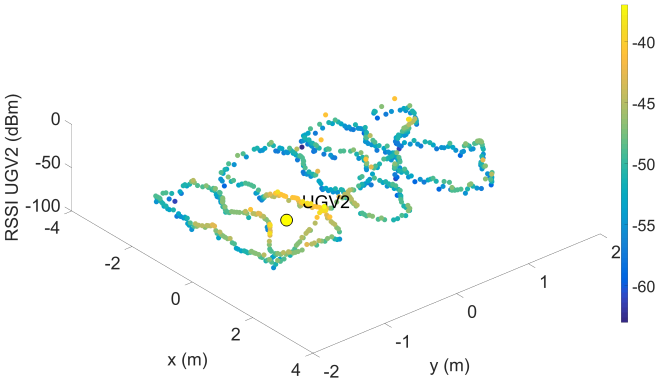


(f) Raw experimental data for UGV 2 and trial 6

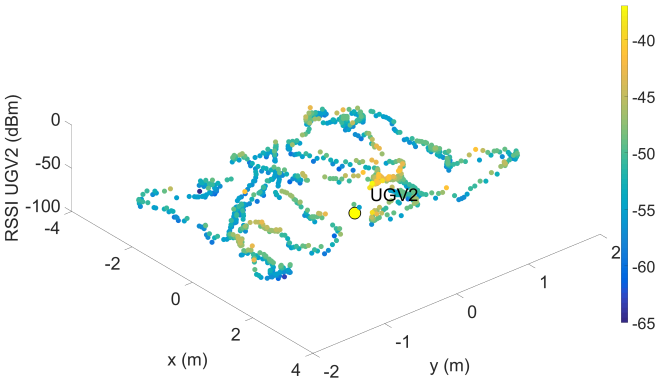
Figure A.6: Summary of raw experimental data for UGV 2 with full city



(g) Raw experimental data for UGV 2 and trial 7

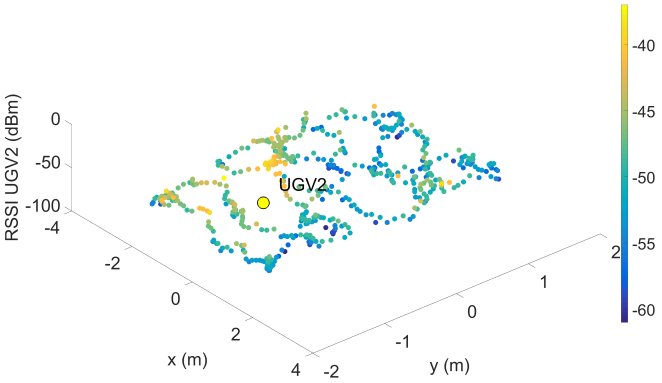


(h) Raw experimental data for UGV 2 and trial 8



(i) Raw experimental data for UGV 2 and trial 9

Figure A.6: Summary of raw experimental data for UGV 2 with full city



(j) Raw experimental data for UGV 1 and trial 10

Figure A.6: Summary of raw experimental data for UGV 2 with full city