CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

BACHELOR'S THESIS

Michal Němec

ARFID chips localization by a group of helicopters using a principle of multilateration

Department of Control Engineering

Thesis supervisor: **Ing. Vojtěch Spurný**

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne ....

Podpis autora práce

CTU CZECH TECHNICAL UNIVERSITY IN PRAGUE

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Němec Michal** Personal ID number: **457202**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Control Engineering**

Study program: **Cybernetics and Robotics**

Branch of study: **Systems and Control**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

ARFID chips localization by a group of helicopters using a principle of Multilateration

Bachelor’s thesis title in Czech:

Lokalizace ARFID čipů skupinou helikoptér využívajících principu Multilaterace

Guidelines:

A principle of multilateration will be used for localization of ARFID tags by a group of Unmanned Aerial Vehicles (UAVs). The following tasks will be solved:
- To understand and implement the multilateration technique [3,4,5] for using onboard of UAVs in real time.
- To integrate the method into the ROS-based system designed at MRS group for control of groups of UAVs [1,2].
- To verify system functionalities in the Gazebo and with real platforms in outdoor conditions.
- To statistically evaluate and discuss the influence of the number of applied vehicles on achieved results (precision and reliability).
- To implement a model of the localization sensor into the Gazebo with parameters identified experimentally using real UAVs.

Bibliography / sources:

[1] T. Baca, P. Stepan and M. Saska. Autonomous Landing On A Moving Car With Unmanned Aerial Vehicle. In The European Conference on Mobile Robotics (ECMR), 2017.
[2] G. Loianno, V. Spurny, J. Thomas, T. Baca, D. Thakur, D. Hert, R. Penicka, T. Krajnik, A. Zhou, A. Cho, M. Saska, and V. Kumar. Localization, Grasping, and Transportation of Magnetic Objects by a team of MAVs in Challenging Desert like Environments. IEEE ICRA and RAL, 2018.
[3] Á. Szüllő, R. Seller, D. Rohács and P. Renner, 'Multilateration based UAV detection and localization,' 2017 18th International Radar Symposium (IRS), 2017.
[4] E. Mazidi, 'Introducing New Localization and Positioning System for Aerial Vehicles,' in IEEE Embedded Systems Letters, vol. 5, no. 4, pp. 57-60, Dec. 2013.
[5] G. Greco, C. Lucianaz, S. Bertoldo and M. Allegretti, 'Localization of RFID tags for environmental monitoring using UAV,' 2015 IEEE 1st International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2015

Name and workplace of bachelor’s thesis supervisor:

Ing. Vojtěch Spurný, Multi-robot Systems FEL

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **31.01.2018** Deadline for bachelor thesis submission: **25.05.2018**

Assignment valid until: **30.09.2019**

Ing. Vojtěch Spurný Supervisor’s signature	prof. Ing. Michael Šebek, DrSc. Head of department’s signature	prof. Ing. Pavel Ripka, CSc. Dean’s signature

Acknowledgements

I would like to thank my supervisor Ing. Vojtěch Spurný for his help and guidance over the writing of this thesis. Special thanks goes to Multi-robot Systems group for their help during the experiments. Without their assistance and dedicated involvement, this thesis would never been accomplished. I would also like to thank my family, partner and friends who always supported me.

Abstract

This thesis deals with localization of unknown ARFID devices by a group of unmanned aerial vehicles (UAVs) in a space using multilateration methods and Bluetooth low energy (BLE) technology. Methods described in this work are formulated in a general way, therefore they can be used with any technology that allows distance measurements. At the beginning of this work we are discussing how to measure distance using BLE and then how to apply this knowledge for multilateration algorithms. Multilateration algorithms are presented from general perspective for arbitrary dimensional space and arbitrary number of deployed UAVs. Furthermore these algorithms are extended by a knowledge about uncertainty of a measurements and their functionality is verified in simulations and by real-world deployment.

Keywords: multilateration, bluetooth low energy, unmanned aerial vehicle, radio frequency identification, localization, least squared method, received signal strength indicators

Abstrakt

V této práci je řešen problém lokalizace neznámého ARFID čipu na bázi Bluetooth low energy technologie. K lokalizaci jsou použity metody využívající principu multilaterace. V práci jsou všechny multilaterační algoritmy popsány obecně a jsou použitelné s libovolnou technologií která umožňuje měřit vzdálenosti mezi objekty. Nejdříve jsou popsány postupy měření vzdáleností pomocí technologie BLE a získané výsledky jsou použity v multilateračních metodách. Všechny popsané algoritmy pracují v libovolném dimenzionálním světě a nejsou limitovány počtem bezpilotních helikoptér. Lokalizovaná poloha je poté zpřesněna zavedením nepřesnosti měření vzdálenosti.

Keywords: multilaterace, bluetooth low energy, bezpilotní helikoptéra, radio frequency identification, lokalizace, metoda nejmenších čtverců, received signal strength indicators

Contents

List of Figures

Scripts

1 Introduction

With rise of mobile devices new bluetooth standard Bluetooth low energy (BLE) was presented. This technology enables creation of low cost and low powered devices that are able to tell proximity distance between each other. Localization techniques are sometimes imeplemented in indoor environment where mobile device is measuring it's distance to static points and then tries to estimate itself inside building. The same principle could be reversed, a group of unmanned aerial vehicles (UAVs) with known positions could measure distance to the unknown object and based on that estimate it's position. All mentioned methods requires distance measurements and thus they are call multilateration methods.

This thesis is continuation of previous research conducted at the Czech Technical University in Prague, Department of Cybernetics, where methods of active RFID chips localization were implemented [1]. Multilateration methods have their use in large variety of problems where measurement of distances between objects is available. One of the most known example is GPS system where data from satellites are used to estimate location of a user on the surface of the Earth. Another known examples are airports [2] that are using multilateration techniques to estimate location of airplane in the air and thus they are able to plan their path according to it. The same methods are also possible to use for a group of UAVs equipped with devices that are able to measure distances to an unknown object [3], [4]. Based on observations from these devices, location of this unknown object can be estimated. Furthermore, UAVs are able to dynamically react that can be helpful for example for checking previously marked spot in open fields or even for finding moving objects in proximity radius [5], [6].

There are multiple formulations of multilateration problems. They could be formulated as linear problem, having closed form solution (Section (7.3), or as non-linear problem, requiring toolset of optimization techniques to estimate unknown location (Section (7.4)). The close form solution has complexity of $\mathcal{O}(M^3)$, where M is number of deployed UAVs. Simulations of our proposed non-linear solution (see Section 8.0.1) shows time complexity of $\mathcal{O}(M)$ which is big improvement compare to linear case.

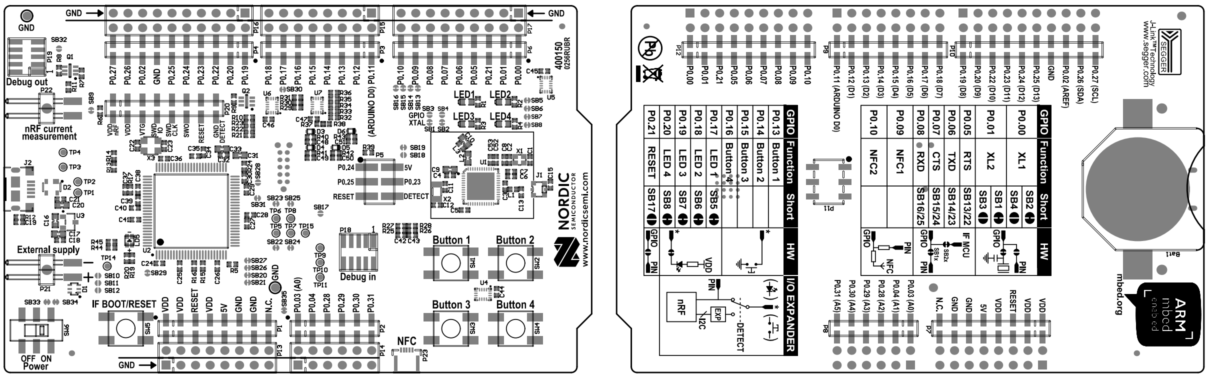
Furthermore, uncertainty of distance measurements can be considered during processing of measured signal. This issue is further referenced and solved by introducing covariance matrix (Section (7.6)) which contains uncertainty information about error propagation in the measurement.

2 Technology

Radio Frequency Identification (RFID) is a collection of devices that are using a radio frequency signal to communicate with others. These devices are widely used as identification tags or security cards. One of their big advantages is low-cost manufacturing and availability on the market. They are generally operating in the free frequency bands such as 868MHz or 2.4GHz. Bluetooth Low Energy(BLE) is one example of such technology [7]. Today they are widely used as a proximity sensor and advertising devices that can notify other devices in a close radius. Giving us information about a distance from the devices inspires the idea of using BLE protocol for a localization in space.

2.1 Bluetooth hardware

For an implementation of transmitter and receiver of BLE signal, we used in this work development board **Nordic nRF52DK**. A transmitter is a device that will be localized. Receiver boards are placed onboard of UAVs that implement algorithms discussed in this thesis. Nordic supplies well documented Software Development Kit (SDK) for development of Bluetooth enabled devices. The board contains ARM microprocessor which can be programmed and flashed over standard USB from Unix/Windows operating system. Board also can be used in further development of an overall solution for location estimation. The device is capable of being extended with other hardware through General-purpose input/output (GPIO) ports.



(a) Front of a **nRF52DK**

(b) Back of a **nRF52DK**

Figure 2.1: Reference layouts of bluetooth development board[a].

[a] https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF52-DK

3 Bluetooth Low Energy (BLE) specification

Standard BLE is defined in Bluetooth Core specification[1]. BLE uses radio frequency the 2.4 GHz band for communication and divides this band into 40 channels with 2 MHz spacing, starting at 2402 MHz. Complete set of 40 channels is divided into 3 advertising channels (37, 38, 39), and 37 data channels (0-36). All these channels are visualized in Figure 5.1a. This figure also shows channels of Wifi that can interfere with BLE signal. These channels are then used for different part of communication:

Advertising channels
Device Discovery
Connection Establishment
Broadcast Transmissions

Data channels
Bidirectional communication between connected devices
Adaptive frequency hopping used for subsequent connection events

Onboard nRF52DK board is used for measuring Received Signal Strength Indicators (RSSI) values from the transmitter points overall advertising channels. This value is then processed using signal propagation model to give resultant distance measurement.

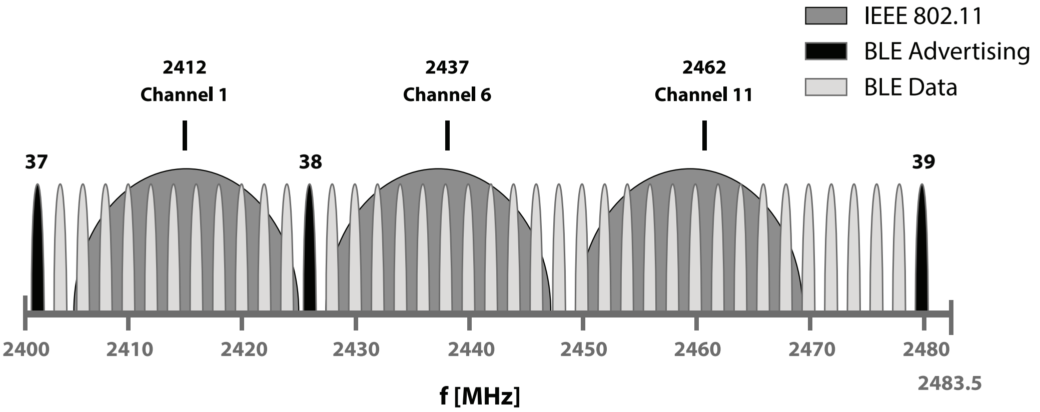


Figure 3.1: Comparison between WiFi and BLE frequency range. BLE spectrum spans enough space to be able work outside and inside spectrum of WiFi signal. Advertising channels are positioned on the edges of the spectrum. This provides better noise toleration against Wifi signal.

[1] https://www.bluetooth.com/specifications/bluetooth-core-specification

4 Distance computation

BLE protocol specifies information about transmitting TxPower and device identifiers. Receiver device is measuring RSSI value for a distance estimations from a transmitter point. For a further use, we are going to call UAV carrying receiver device as an anchor point.

Function $\Omega(x_{rssi})$ characterises relation between measured RSSI value and distance. We shall define it as strictly positive function. We do not have any physical interpretation of a negative distance and strictly positive eliminates the case of being localized directly inside anchor point

$$\Omega : \mathbb{R} \to (0, +\infty) \, . \tag{4.1}$$

4.1 Signal propagation model

There are many signal propagation models used for computing distance from RSSI values. Model should be chosen based on requirements of a final application. The unknown object is chosen to be in open field, hence this ensures there are minimal reflections and multi path noise in the signal. For computation reasons, we need our model to give us the function of a distance. If model gives function of a receiver power than it needs to be an invertible function, otherwise we cannot use this function for a unique computation from a distance. Given that we are in an open field simple model such as Free space signal propagation model (FSM). This model could be derived from more general Friis transmission equation [8]. Free space model assumes several assumptions about receiver and transmitter sides. That are mainly these:

Isotropic transmission from the antenna

Development board Nordic nRF52DK has its antenna by default facing in one direction. Therefore we cannot expect it to be isotropically radiating signal. This can be solved by attaching our own antenna and then use Friis equation that would take advantage of geometric properties of that antenna.

Line of sight propagation of the signal

Receiver at any given moment of measurement should see transmitter without any obstacle between them. BLE technology is expected to have range up to 100 meters according to documentation. In this range, we can ensure that only obstacle between transmitter and receiver are bodies of the UAVs itself. Methods presented here are generally working in arbitrary distances. A different technology that would allow to measure distance even longer than is distance to the horizon on the earth surface can be used. In that case, the ability how to quantify when the signal is received in the line of sight manner with account for a earth surface curvature needs to be presented (discussed more in Section 11.2).

Free space model relation between RSSI and distance is specified as

$$\text{RSSI}\ [\,\text{dB}\,] = -10n\log_{10}(d) + \text{TxPower} + \nu\,, \qquad \nu \sim \mathcal{N}(\mu,\, \sigma)\,, \tag{4.2}$$

therefore we can compute distance as

$$d\ [\,\text{m}\,] = 10^{\dfrac{\text{TxPower} - \text{RSSI} + \nu}{10n}}\,, \tag{4.3}$$

where n is environment constant used for calibration in non ideal environment (free space has $n = 2$). TxPower [dB] is constant of the transmitter module that represents RSSI value received in 1 meter distance from transmitter.

5 RSSI characteristics

BLE signals are in a range of 2.4GHz frequency which means signal is travelling in straight line path and there are no bending or reflection from a stratosphere. We are planning to run an algorithm onboard of the UAV in an outdoor field thus to minimize reflections from the ground we should measure the characteristics in the air.

Placement of a BLE transmitter on the target is crucial for a correct RSSI readings. UAV carries other devices that can work on the same frequency range and also further part of the construction is shielded so that radio waves cannot penetrate the UAVs body that easily. Due to that, we need to make an assumptions about target relative location against UAV for partially correct transmitter placement:

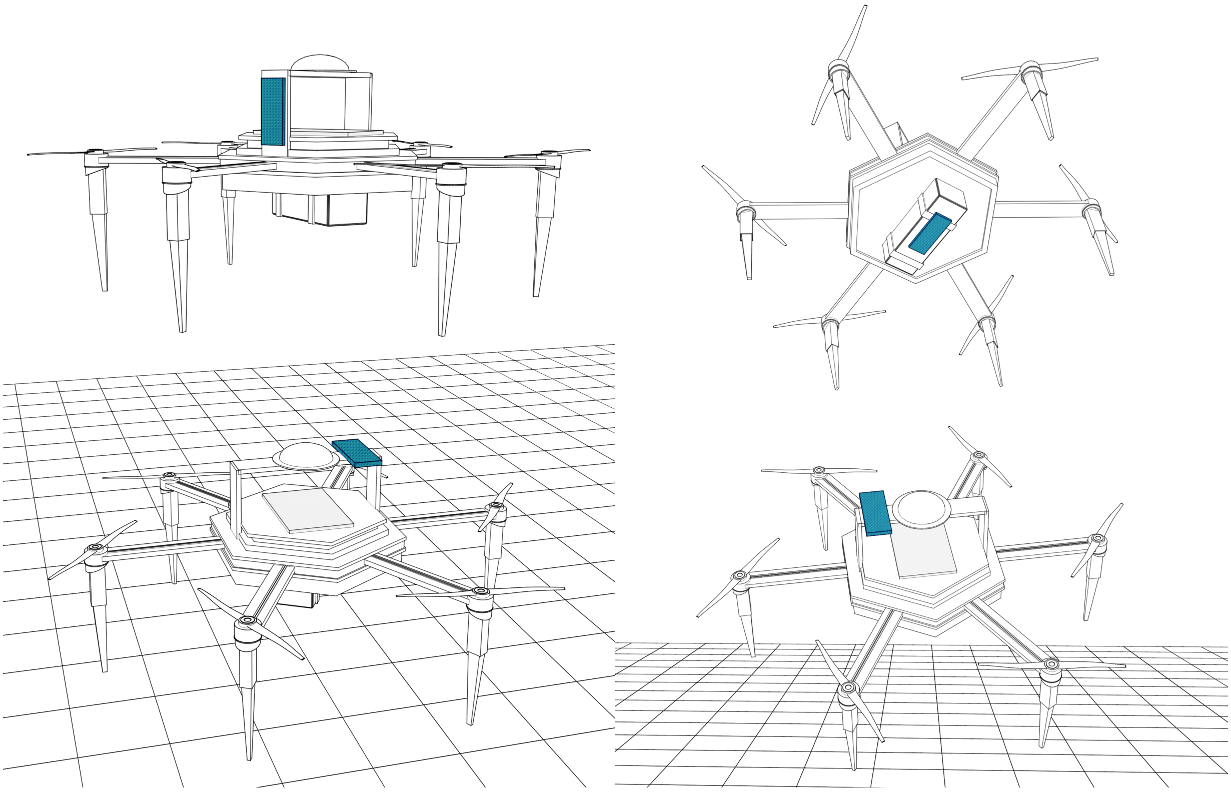
- Target movement is restricted to be under the location of UAV (eq. UAVs are estimating location of the target moving on the ground)

5.1 Measurements

Experiments are designed to measure signal decay over a absolute distance between transmitter and receiver. For a further use we are going to call signal decay characteristic as characteristic.

Characteristics are measured in two main configurations. First one is measured with the placement of a transmitter and receiver in 90° rotated to each other. Transmitter is placed on the one side of the UAV and receiver is placed flat on top of the second UAV (see Fig. 5.1a). In Second configuration, transmitter and receiver are facing each other (see Fig. 5.1b). Both configurations are used for measuring types of characteristics. Figure 5.2 shows placement of Nordic devices and their orientation.

. The first configuration has longer running times because each characteristic is measured on a new battery pack. Second configuration has measured both characteristics in one run.



(a) Configuration 1

(b) Configuration 2

Figure 5.1: Device placement onboard of UAV. For both transmitter UAV at the bottom has device facing upwards. Transmitter UAV (Fig. 5.1a) has Nordic device facing to the side. Latter one has receiver device facing bottom. This has been shown to be better configuration (Fig. 5.6) than a first one. Goal of the experiments is to gather information about directivity of an Nordic board antenna. Second measurement (Fig. 5.4) shows that by using onboard devices facing to each other (Configuration 2).

5.1.1 All directional characteristic

Characteristics are measured using spherical trajectory. Transmitter UAV is placed in one point and receiver UAV is following non-collision[2] trajectory around transmitter. The UAVs first configuration is obtained in 7 minute flight and is measuring samples from top hemisphere with radius 12 meters. Characteristic for the second configuration is measured

[2] UAVs have collision avoidance system that is activated whenever another UAV is inside safety cylinder $\begin{bmatrix} r & h \end{bmatrix} = \begin{bmatrix} 5 & 6 \end{bmatrix}$ [meters].

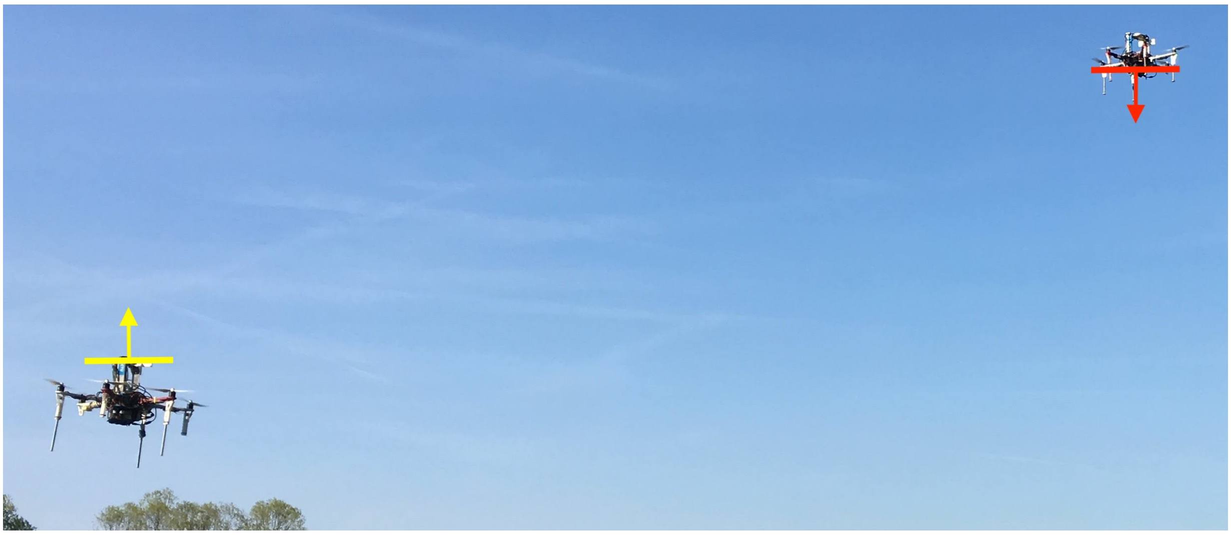
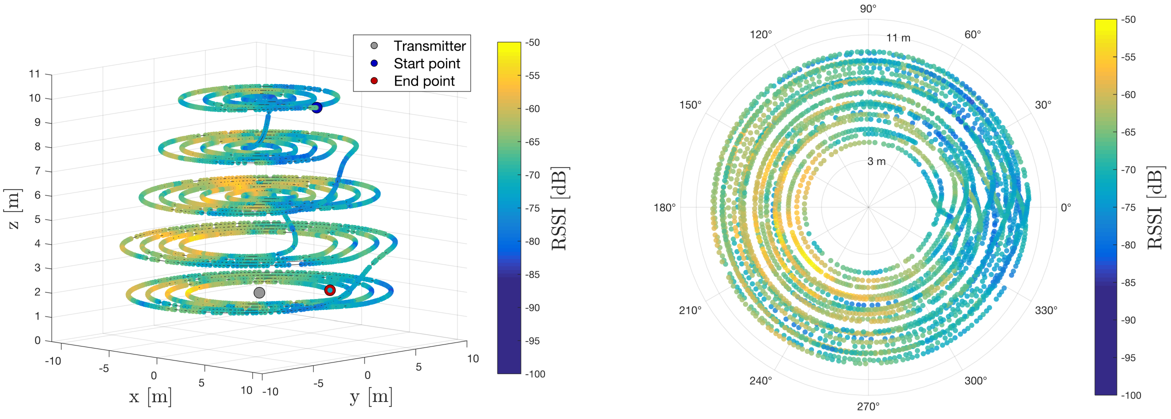


Figure 5.2: RSSI characteristics experiments conducted with respect to the configuration 2. Transmitter (yellow arrow) and receiver (red arrow) are facing according to the arrows.

in 3 minute 30 seconds flight measuring top hemisphere with less details. Trajectories are generated with respect to the maximal time flight on one fully charged battery pack[3]

Experiment data shows that using configuration 2 (Fig. 5.4) ensures better symmetry or received signal than first (Fig. 5.3). We do not have any prior knowledge about rotation and relative position of transmitter in relation to the receiver thus the second configuration is chosen for experiments presented further in the work.

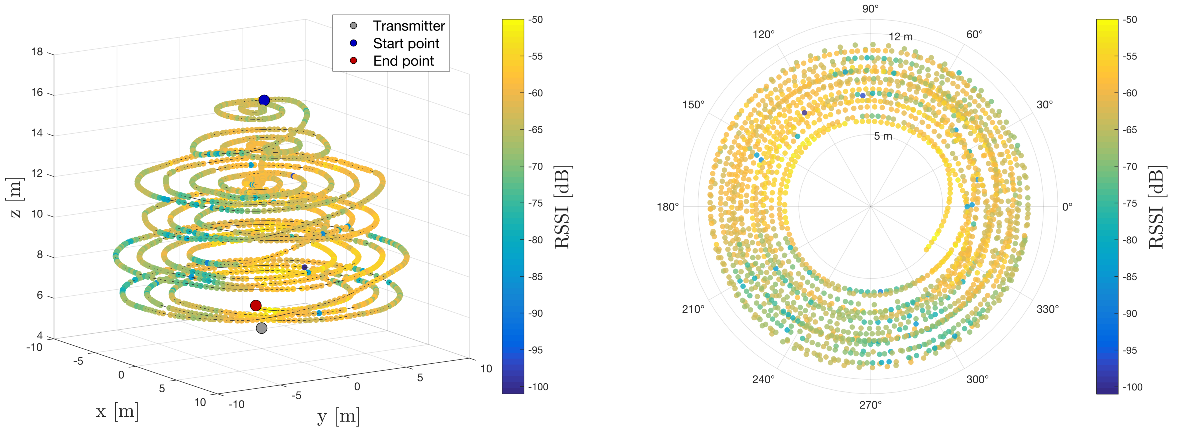


(a) UAV trajectory

(b) RSSI values mapped to polar coordinates

Figure 5.3: Spherical characteristic taken with the configuration 1. Video available at `http://mrs.felk.cvut.cz/nemec2018thesis`

[3] Fully charged battery pack has flight expectance ∼10 minutes.



(a) UAV trajectory

(b) RSSI values mapped to polar coordinates

Figure 5.4: Spherical characteristic taken with the configuration 2. Video available at `http://mrs.felk.cvut.cz/nemec2018thesis`

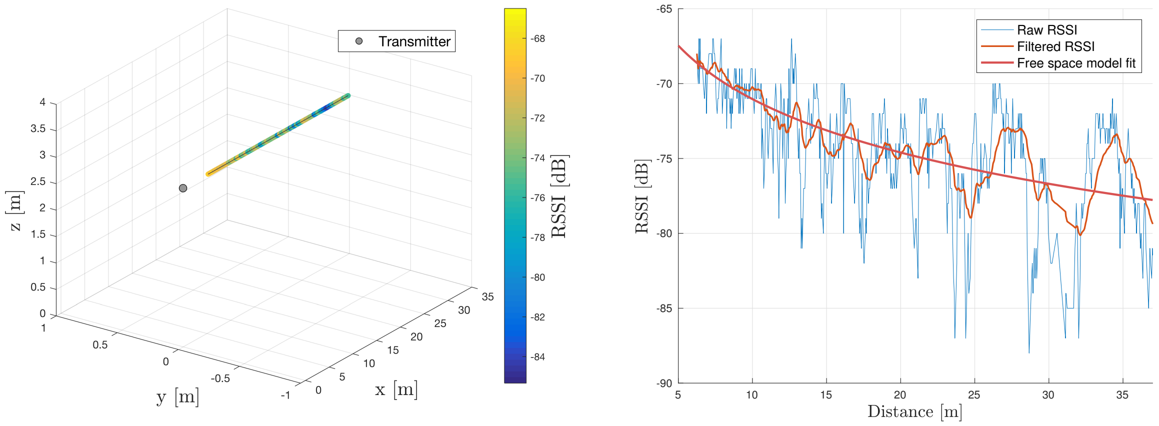
5.1.2 One directional characteristic

Characteristics are measured using straight line trajectory. Reason for this measurement is ability to measure further distances from the transmitter. First configuration is measured in 7 minutes on the distance interval $\langle 5, 40 \rangle$ meters (Fig. 5.5a). Second configuration is measured in 3 minutes and 30 seconds in interval $\langle 4, 30 \rangle$ meters (Fig. 5.6a). Experiments shown in Figure 5.7 display difference between two configurations. Free space model does not really account for instability of BLE signal and needs to be adjusted for a further use.

For that we first introduce saturation range of a signal. We are assuming distance from transmitter to receiver in a range $\langle\, 1,\ 42\, \rangle$ meters in all experiments. Multilateration experiments are conducted in the configuration 2 (Fig. 5.1b), That way we define our distance function to follow second measurement of a directional characteristic. Measured characteristics started to overlap for distances greater than 30 meters. We chose our distance function to overlap these to measurements and than saturate at final distance (Fig 5.7). Final function is taken to be defined segment-wise and evaluated by linear interpolation between specified points. Results are summarized in Table 1. Parameters estimated from measured data. Root square mean error (RMSE) is computed for a raw and kalman filtered values.

Table 1: Free space model data fitting

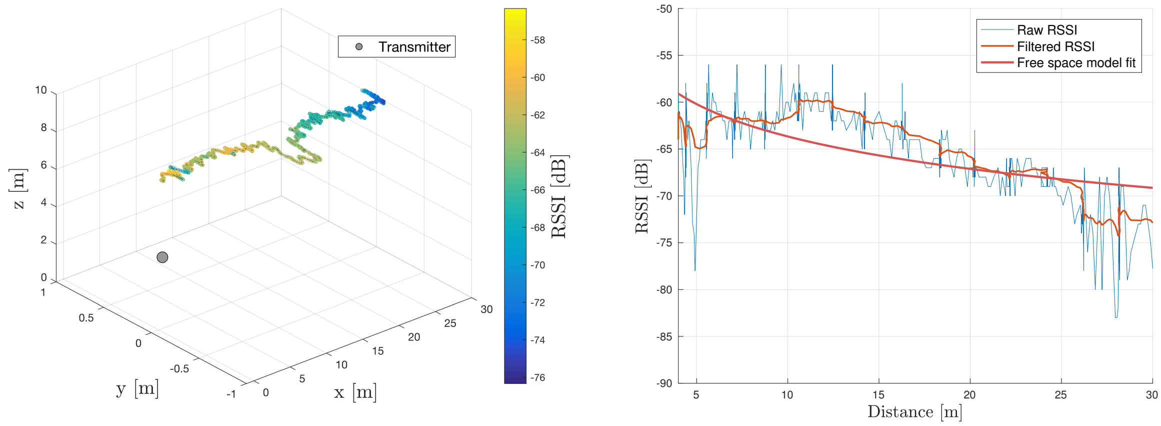
Free space model	n	TxPower [dB]	$RMSE_{\text{fit,raw}}$ [dB]	$RMSE_{\text{fit,filter}}$ [dB]
Configuration 1	1.186	-59.18	3.8015	1.9842
Configuration 2	1.149	-52.17	3.6188	2.8685



(a) UAV trajectory

(b) RSSI characteristic

Figure 5.5: One directional measurement of RSSI characteristic in the UAV configuration 1. RSSI values are filtered using kalman filter with parameter ($R = 0.01$). Free space model fit (Eq. 4.2): $n = 1.186$, TxPower $= -59.18$ [dB]. Video available at `http://mrs.felk.cvut.cz/nemec2018thesis`



(a) UAV trajectory

(b) RSSI characteristic

Figure 5.6: One directional measurement of RSSI characteristic in UAV configuration 2. RSSI values are filtered using kalman filter with parameter ($R = 0.01$). Free space model fit (Eq. 4.2): $n = 1.149$, TxPower $= -52.17$ [dB]. This measurement (Fig. 5.6b) shows great improvenent of overall stability of the signal in comparison to (Fig. 5.5b), mainly due to the fact that transmitter and receiver weren't blocked by UAV body. Video available at `http://mrs.felk.cvut.cz/nemec2018thesis`

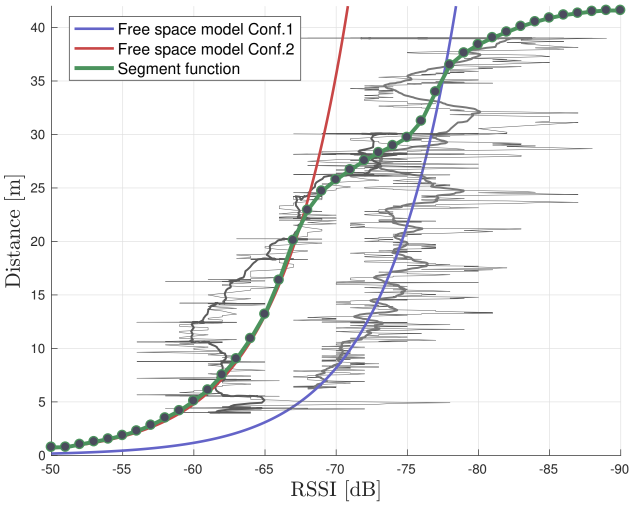


Figure 5.7: Custom distance function created from both experiments. Curve is manually created from both datasets by specifying keypoints of the characteristics and interpolated in between using spline curve. Beginning of the curve closely follows second configuration because following experiments were using multilateration techniques had their transmitters and receiver installed in that manner.

5.2 Filter

RRSI values are filtered using one dimensional kalman filter. Nordic development board has been programmed in a way that we are getting measurements in fixed time interval. Due to fact that we do not have complex model for behaviour of RSSI values in the environment. For that we propose to use kalman filter of one variable with static model. Kalman filter algorithm can be simplified to as

$$\bar{x} = x_n \,, \tag{5.1}$$

$$\bar{c} = c_n + R \,, \tag{5.2}$$

$$K = 1 + \bar{c} \,, \tag{5.3}$$

$$x_{n+1} = \bar{x} + K(z - \bar{x}) \,, \tag{5.4}$$

$$c_{n+1} = \bar{c}(1 - K) \,, \tag{5.5}$$

where z is actual measured sampled. Initial value of x_0 is taken to be first sample of rssi value and inital value c_0 is set to 1. Tuning parameters R will be estimated heuristically for having balance between response time and filtered value. Based on the experiments measuring RSSI values in one direction (Fig. 5.5) such constant has been chosen to be $R = 0.01$.

5.3 Module implementation

Transmitter uses only Nordic hardware and works as standalone device. Device is programmed using reference Nordic program for BLE beacon taken from reference development page[4]. BLE beacon is transmitting advertising data with identifiers major/minor (10/10).

Receiver board is programmed as BLE receiver using code from manufacturer[5]. Device is listening for advertising data with identifiers same as transmitter board in 100ms fixed intervals. ROS module is filtering raw RSSI values from board readings using kalman filter. Signal model fit (Fig. 5.7) transforms filtered RSSI values to their distance equivalents. Distance measurements are then published using ROS system in the same intervals as for nordic board.

Location module is implemented and tested in Gazebo system used for UAV simulations. Module is able to work with free space model function (Eq. 4.3) as well as with piece wise defined characteristic (Fig. 5.7).

[4] https://github.com/NordicPlayground/nrf51-ble-tutorial-advertising

[5] https://developer.nordicsemi.com/nRF5_SDK/

6 Least square method (LSM)

. Least square method is generally defined as optimization problem where we are trying to find extremas of a scalar cost function in a form

$$J(\vec{x}) = \frac{1}{2}\vec{e}(\vec{x})^T H \vec{e}(\vec{x})\,, \tag{6.1}$$

where $\vec{x}$ is vector of unknown variables, $\vec{e}$ is a column error vector that defines relation between expected and sampled values. We are trying to find argument in which function has it's lowest value

$$\vec{x} = \arg\min_{\vec{x}} J(\vec{x})\,. \tag{6.2}$$

Square matrix H could be used for additional weights and parameters. If matrix H is different from identity matrix method is also called Weighted least square method (WLSM) Generally we define elements of a error vector as difference between expected and sampled value

$$e_i(\vec{x}) = \beta_i - \gamma_i(\vec{x})\,. \tag{6.3}$$

Special type of a matrix H is positive diagonal matrix. We can rewrite expression (Eq. 6.1) with coeficients from diagonal as

$$J(\vec{x}) = \frac{1}{2}\vec{e}(\vec{x})^T H \vec{e}(\vec{x}) = \frac{1}{2}\sum_i H_{ii} e_i^2 = \frac{1}{2}\sum_i \left(\sqrt{H_{ii}} e_i\right)^2\,, \tag{6.4}$$

which is the same as setting matrix H to identity matrix and rewrite our error vector as a new vector

$$e'_i(\vec{x}) = \sqrt{H_{ii}}\, e_i(\vec{x})\,, \tag{6.5}$$

and our modified cost function

$$J(\vec{x}) = \frac{1}{2}\vec{e'}(\vec{x})^T \vec{e'}(\vec{x})\,. \tag{6.6}$$

Scalar β_i is sampled value and scalar function $\gamma_i(\vec{x})$ of unknown variables vector. Function γ_i represents relation of unknown variable vector $\vec{x}$ with respect to sampled value β_i

6.1 Derivative of cost function

Most of algorithms used for finding extrema of a function requires computing gradient of function. Assuming that in expression (Eq. 6.1) is matrix H only positive diagonal matrix[6] with scalar values and (Eq. 6.3) used as error vector then computation of a gradient can be simplified to computing gradient of a unknown function γ_i as

$$\frac{\partial J(\vec{x})}{\partial x_j} = \frac{1}{2}\frac{\partial}{\partial x_j}||\vec{e'}(\vec{x})||^2 = \sum_i H_{ii} e_i \frac{\partial e_i(\vec{x})}{\partial x_j} = -\sum_i H_{ii} e_i \frac{\partial \gamma_i(\vec{x})}{\partial x_j}\,. \tag{6.7}$$

Higher order optimization algorithms could also require to specify second order derivative of (Eq. 6.1). For direct computation of second derivative we would need to differentiate again (Eq. 6.7) that is

$$\frac{\partial^2 J(\vec{x})}{\partial x_j \partial x_k} = \sum_i H_{ii} \left(\frac{\partial \gamma_i(\vec{x})}{\partial x_j}\frac{\partial \gamma_i(\vec{x})}{\partial x_k} - e_i \frac{\partial^2 \gamma_i(\vec{x})}{\partial x_j \partial x_k} \right)\,. \tag{6.8}$$

6.2 Solutions of optimization problem

LSM formulated in previous section cam be solved analytically if function γ_i is linear in parameters of unknown vector $\vec{x}$. meaning γ_i can be written as

$$\gamma_i(\vec{x}) = \sum_j c_{ij} x_j\,, \qquad \frac{\partial \gamma_i(\vec{x})}{\partial x_j} = c_{ij}\,, \qquad \frac{\partial \gamma_i(\vec{x})}{\partial x_j \partial x_k} = 0\,. \tag{6.9}$$

Linear version of (Eq. 6.9) can also be expressed as matrix multiplication $\mathbf{C} = [c_{ij}]$. We would solve extrema of a function with error vector $\vec{e}(\vec{x}) = \vec{\beta} - C\vec{x}$ is then

$$J(\vec{x}) = \frac{1}{2}||\vec{\beta} - C\vec{x}||^2 = \frac{1}{2}\left[\vec{x}^T C^T C \vec{x} - 2\vec{x}^T C^T \vec{\beta} + \vec{\beta}^T \vec{\beta}\right]\,. \tag{6.10}$$

Solving for a $\vec{x}$ where $\nabla J(\vec{x}) = \vec{0}$ gives matrix form solution to linear LSM problem.

$$\nabla J(\vec{x}) = C^T C\vec{x} - C^T\vec{\beta} \xRightarrow{\nabla J(\vec{x})=\vec{0}} C^T C\vec{x} = C^T\vec{\beta}\,, \tag{6.11}$$

$$\vec{x} = (C^T C)^{-1} C^T \vec{\beta}\,. \tag{6.12}$$

Final formula (Eq. 6.12) assumes that $C^T C$ is nonsingular matrix which is requirement for unique solution to exists. Problems with this solution is also numerical instability that arise in $C^T C$ when matrix is almost singular or coefficients are too big or too small. Linear problem can slo be solved for a general weight matrix H

$$\vec{x} = (C^T H^{-1} C)^{-1} C^T H^{-1} \vec{\beta}\,. \tag{6.13}$$

More complex function cannot be solved analytically and techniques from numerical analysis needs to be used.

[6] Differentiating cost function when weight matrix H is non-diagonal we can expand matrix multiplication as $J = \frac{1}{2}\sum_i \sum_j H_{ij} e_i e_j$ and consequently compute derivative $\frac{\partial J}{\partial x_k} = -\frac{1}{2}\sum_i \sum_j H_{ij}\left[e_i \frac{\gamma_j}{\partial x_k} + e_j \frac{\gamma_i}{\partial x_k}\right]$ and second order derivative $\frac{\partial^2 J}{\partial x_l \partial x_k} = \frac{1}{2}\sum_i \sum_j H_{ij}\left[\frac{\partial \gamma_i}{\partial x_l}\frac{\partial \gamma_j}{\partial x_k} + \frac{\partial \gamma_j}{\partial x_l}\frac{\partial \gamma_i}{\partial x_k} - e_i \frac{\partial^2 \gamma_j}{\partial x_l \partial x_k} - e_j \frac{\partial^2 \gamma_i}{\partial x_l \partial x_k}\right]$.

Iterative nonlinear optimizers

Nonlinear optimizers needs to be used when problem cannot solve directly by closed form solution. In real world application they are also used in many linear optimization problems where state space is too huge for a solution using explicit formula or number of operations needed for a solution is higher then number of iterations in nonlinear optimizer.

Gradient descent (GD) method is used for finding local minimum of a unknown vector function $F(x)$. It's iterative method which requires to specify initial point and known gradient of a function $F(x)$. This method has lowest memory overhead and is better suited in environments with big data structures(for example when training neural networks)

Gauss-Newton (GN) is second order iterative method. It additionally uses second order derivative to estimate correct iteration step. Because of that algorithm needs more function evaluations then GD method, but has higher change of finding extrema point faster[7].

Levenberg marquardt (LM) method behaves as combination of Gradient descent method and Newton-Gauss method. When point is further from extrema it iterates in a direction of gradient and when closer to the extrema uses approximation of a second derivative and behaves as Newton-Gauss method with quadratic convergence. Overall expected convergence is expected to be linear and slow. Method is expected to have biggest memory and function evaluation overhead from the three mentioned here, but it provides more robust solver that is better suited in optimizing smaller problems.

[7] Faster mean it will converge to the right solution in less iterations then other method.

7 Localization methods

Many algorithms exists using different approaches[9]. We are going to show methods that are using distances for localization estimation (lateration methods). Proposed algorithms are working with M distance measurements from anchor points. Anchor point is actual position of the UAV in local space with respect to the origin. We will define unknown position of the target in N-dim space $\vec{x}$ as coordinates to standard basis of Euclidean space

$$\vec{x} \in \mathbb{R}^N \,, \tag{7.1}$$

and our distance measurements $\vec{\xi}$ vector as follows

$$\vec{\xi} \in (0, +\infty)^M \,. \tag{7.2}$$

Further lets define matrix L where row represents position vector of the anchor and column represents coordinate in N-dim space. As the last we define L_i with one index as row vector of matrix L for a representation of anchor point position

$$\mathbf{L} = \begin{bmatrix} \vec{L_1} \\ \vdots \\ \vec{L_M} \end{bmatrix} \,, \qquad L \in \mathbb{R}^{M \times N} \,. \tag{7.3}$$

Goal of these algorithms is to find a unknown position $\vec{x}$ in N-dim space using information about position of the anchor points and their distances to the target object.

7.1 Existence of a solution

Method has a unique solution in N dimensional space if $rank(L) = N$ and $M \geq N+1$, which gives us set of $N+1$ independent equations. For a $rank(L) < N$ equations we could run an algorithm in $rank(L)$ dimensional space [8]. When creating new base vectors in subspace $rank(L)$ dimensions it's generally recommended to make sure that our base vectors are orthonormal. These conditions can be ensured by using Gram-Schmidt Orthogonalization process when creating new basis.

7.2 Naive method

We could try to localize unknown object using solution given by set of equation for N-dimensional sphere (Eq. 7.4). This would require to specify M (minimum of N+1) anchor

[8] example of using 3 non collinear anchor points in 3D space can be simplified to solving same problem in 2D where space is defined by plane constructed from 3 anchor points

points with their corresponded distance measurements. For less than N+1 points set of equations has multiple solutions.

$$\sum_{j=1}^{N}(x_j - L_{ij})^2 = \xi_i^2 \,, \qquad i \in \{1, \ldots, M\} \,. \tag{7.4}$$

Disadvantage of this method is that we need to know exact distance measurement ξ_i without any measurement errors. Any slight offset from ideal values ξ_i would result in non solvable set of equations. For example if we had perfectly calibrated set of 3 anchor points for localization, adding another anchor point would possibly result in unsolvable set of equations.

7.3 Hyperbolic algorithm

Naive method can be extended in a way that we are solving linear system using LSM method. Back in (Eq. 7.4) subtracting k-th equation from others and expanding squared elements gives set of $M-1$ linear equations

$$\sum_{j=1}^{N}(x_j - L_{ij})^2 - \sum_{j=1}^{N}(x_j - L_{kj})^2 = \xi_i^2 - \xi_k^2 \,, \tag{7.5}$$

$$\sum_{j=1}^{N} x_j^2 - 2x_jL_{ij} + L_{ij}^2 - (x_j^2 - 2x_jL_{kj} + L_{kj}^2) = \xi_i^2 - \xi_k^2 \,, \tag{7.6}$$

$$\sum_{j=1}^{N} -2x_jL_{ij} + L_{ij}^2 + 2x_jL_{kj} - L_{kj}^2 = \xi_i^2 - \xi_k^2 \,, \tag{7.7}$$

$$\sum_{j=1}^{N} 2(L_{kj} - L_{ij})x_j = \xi_i^2 - \xi_k^2 - \sum_{j=1}^{N}\left(L_{ij}^2 - L_{kj}^2\right) \,, \tag{7.8}$$

where $k \in \{1, \ldots, M\}$. Other equations are constructed from remaining indexes $i \in \{1, \ldots, M\} \,/\, \{k\}$. Linear system of equations (Eq. 7.8) can be rewritten in a matrix form $A\vec{x} = \vec{b}$, where matrix A and vector $\vec{b}$ are specified by elements

$$A = 2\left[L_{kj} - L_{ij}\right] \,, \qquad A \in \mathbb{R}^{(M-1)\times N} \,, \tag{7.9}$$

$$\vec{b} = \left[\xi_i^2 - \xi_k^2 - \sum_{j=1}^{N}\left(L_{ij}^2 - L_{kj}^2\right)\right] = \left[\left(\xi_i^2 - \sum_{j=1}^{N} L_{ij}^2\right) - \left(\xi_k^2 - \sum_{j=1}^{N} L_{kj}^2\right)\right] \,. \tag{7.10}$$

Solving matrix equation with (Eq. 7.9) and (Eq. 7.10) can be done directly using formula (Eq. 6.12). From index notation we see that selection of equation that will be subtracted from others doesn't influence solution, because i is complementary index to k, meaning choosing different values will yield in a same set of equations.

Name of the method comes from it's use in Time of Arrival (TDOA) and Time Difference of Arrival (TDOA)[10] systems where distance measurement ξ_i is a function of time arrival on the receiver $\xi_i \propto \tau_i$ (dimensions typically used $N \in \{2, 3\}$) thus (Eq. 7.5) is set of hyperbola/hyperboloid equations.

Example of solving two dimensional location of a unknown object using 3 anchor points will lead to solving system of 2 equation (Eq. 7.12). Parameter k can be any number from $\{1, 2, 3\}$. Choosing $k = 1$ gives set of equations constructed from equations $\{2, 3\}$.

$$L = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix}^T \qquad \vec{\xi} = \begin{bmatrix} d_1 & d_2 \end{bmatrix}^T \tag{7.11}$$

$$2 \begin{bmatrix} x_1 - x_2 & y_1 - y_2 \\ x_1 - x_3 & y_1 - y_3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (d_2^2 - x_2^2 - y_2^2) - (d_1^2 - x_1^2 - y_1^2) \\ (d_3^2 - x_3^2 - y_3^2) - (d_1^2 - x_1^2 - y_1^2) \end{bmatrix} \tag{7.12}$$

```
function [A,b] = hyperbolic_linear_params(xi,L)
    M = length(xi);
    A = L(2:M, :);
    b = zeros(M-1,1);
    for i=1:M-1
      A(i,:) = 2.*(L(1,:) - A(i,:));
      b(i) = xi(i+1)^2 - xi(1)^2 - sum(L(i+1,:).^2-L(1,:).^2);
    end
end
```

Script 1: Matlab implementation of linear parameters (Eq. 7.9),(Eq. 7.10) in hyperbolic localization method (parameter $k = 1$). Script is able to work in arbitrary number of receiver points in arbitrary dimensional space.

7.4 Nonlinear spherical method

In this localization method we define cost function which has its minimum in possible point where we are trying to estimate our object. First we specify error distance column vector $\vec{e}(\vec{x})$ of every measured distance against computed distance. We can write final error vector element wise as

$$e_i(\vec{x}) = \xi_i - \Lambda_i(\vec{x}) \,, \qquad i \in \{1, \ldots, M\} \,. \tag{7.13}$$

Scalar function $\mathbf{\Lambda_i}$ defines computing analytical distance of target from each anchor point. We are going to define our $\mathbf{\Lambda_i}$ function as distance between two points in flat Euclidean space

$$\Lambda_i : \mathbb{R}^N \to \langle 0, +\infty) \ , \qquad \Lambda_i(\vec{x}) = ||\vec{x} - \vec{L}_i|| = \sqrt{\sum_{j=1}^{N} (x_j - L_{ij})^2} \, , \qquad i \in \{1, \ldots, M\} \, . \quad (7.14)$$

This method is useful if we know that our distance measurement vector $\vec{\xi}$ has same accuracy in every distance from receiver point. In many applications we might be in position where accuracy of distance estimations is different with respect to the distance measurements. This would require us to specify weight matrix H in (Eq. 6.1). We could analyze behaviour around one anchor point (Eq. 7.15) with configuration

$$L = \left[\ 0 \quad 0 \ \right] \, , \qquad \vec{\xi} = \left[\ 1 \ \right] \, . \qquad (7.15)$$

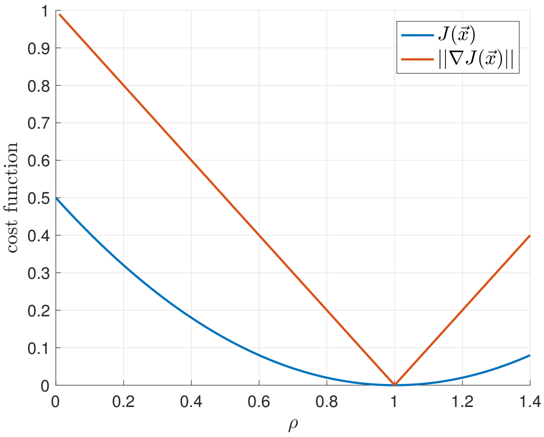


Figure 7.1: Visualization of a cost function evaluated for a one anchor (Eq. 7.15), and it's derivative in two dimensions. Function is evaluated with parameter $\alpha = 1$ (Eq. 7.17). Cost function with one anchor point is symmetric around it's z-axis we can also use substitution in cylindrical coordinates $\rho(x, y) = \sqrt{x^2 + y^2}$ and plot function $J(\vec{x})$ as function of one variable $J(\rho)$.

First order derivative Λ_i

For computation of there methods we use nonlinear solver that uses Levenberg Marquardt algorithm for finding local minimum of the function. Optimization method also requires our derivative of cost function, for a faster computation. LM method is able to numerically estimate Jacobian from function alone but that is in cost of a computation time and doesn't guarantee right results. Following formula

$$\frac{\partial \Lambda_i(\vec{x})}{\partial x_j} = \frac{x_j - L_{ij}}{\Lambda_i(\vec{x})} \, , \qquad (7.16)$$

is computing derivative with respect to every anchor point. We can immediately see problem around the points $\vec{x} = \vec{L}_i$ where function is undefined. Solution is to redefine our error vector $e_i(\vec{x})$ used in cost function using power α of a ξ_i values and $\Lambda_i(\vec{x})$ function

$$e_i(\vec{x}) = \xi'_i - \Lambda'_i(\vec{x}) \,, \qquad \xi'_i = \xi_i^\alpha \,, \qquad \Lambda'_i(\vec{x}) = \Lambda_i^\alpha(\vec{x}) \,. \tag{7.17}$$

Parameter α could be used in range $\mathbb{R}$, but it makes sense to restrict it's value on the interval $(0, +\infty)$. If we want to make derivative defined in a point $\vec{x} = \vec{L}_i$ we should choose $\alpha \geq 2$. Choosing $\alpha = 2$ (see Fig. 7.2) also benefits from simplifying computation of a square root in $\Lambda_i(\vec{x})$ function. Derivative of a modified Λ_i function is written as

$$\frac{\partial \Lambda'_i(\vec{x})}{\partial x_j} = \alpha \Lambda_i^{\alpha-2}(\vec{x})(x_j - L_{ij}) \,. \tag{7.18}$$

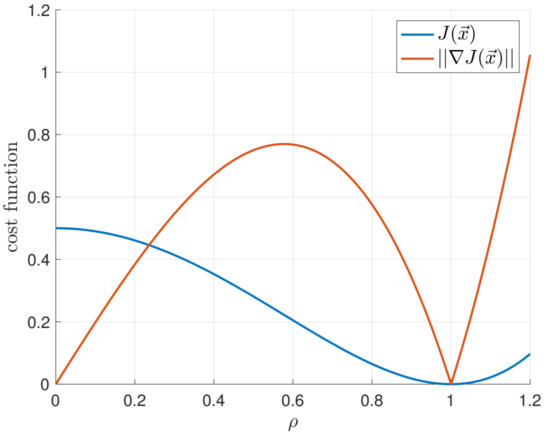


Figure 7.2: Visualization of a cost function evaluated for a one anchor point (Eq. 7.15), and it's derivative in two dimensions. Function is evaluated using (Eq. 7.17) with parameter $\alpha = 2$. Cost function with one anchor point is symmetric around it's z-axis we can also use substitution in cylindrical coordinates $\rho(x, y) = \sqrt{x^2 + y^2}$ and plot function $J(\vec{x})$ as function of one variable $J(\rho)$.

Second order derivative Λ_i

Optimization method like Newton-Gauss method requires computation of Hessian. We could approximate second order derivative by numeric approximation, or we could take second derivative directly by differentiating (Eq. 7.19). Implementation of second derivative can be found in appendix (Script. 8).

$$\frac{\partial^2 \Lambda'_i(\vec{x})}{\partial x_j \partial x_k} = \alpha(\alpha - 2)\Lambda_i^{\alpha-4}(\vec{x})(x_j - L_{ij})(x_k - L_{ik}) + \alpha \Lambda_i^{\alpha-2} \delta_{jk}) \,, \tag{7.19}$$

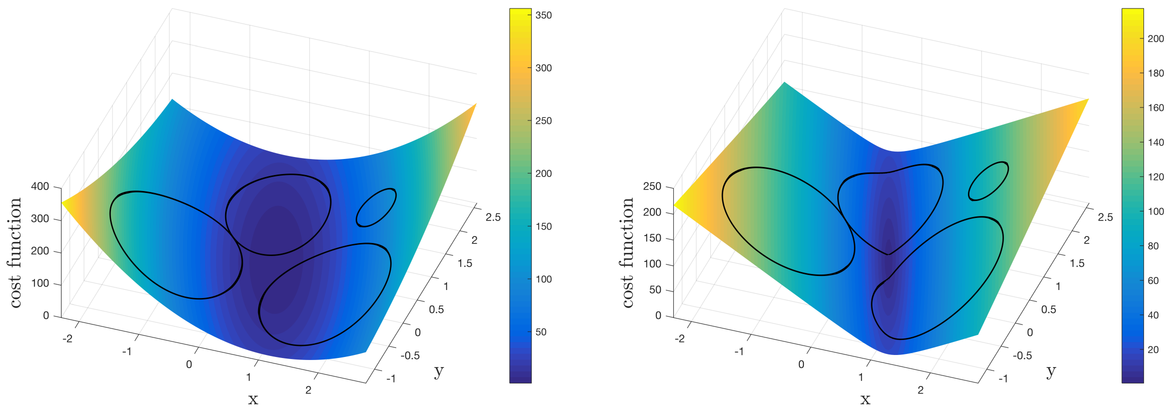
where δ_{ij} is Kronecker delta

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \,. \tag{7.20}$$

If we want second derivative to be defined for every $\vec{x} \in \mathbb{R}^N$ it's necessary to chose $\alpha \geq 4$. Second derivative could be also used in first order optimizers for estimating upper bound of one step.

Differences are shown by plotting cost function (see Fig. 7.3 and Fig. 7.4) evaluated in two dimensions with configuration

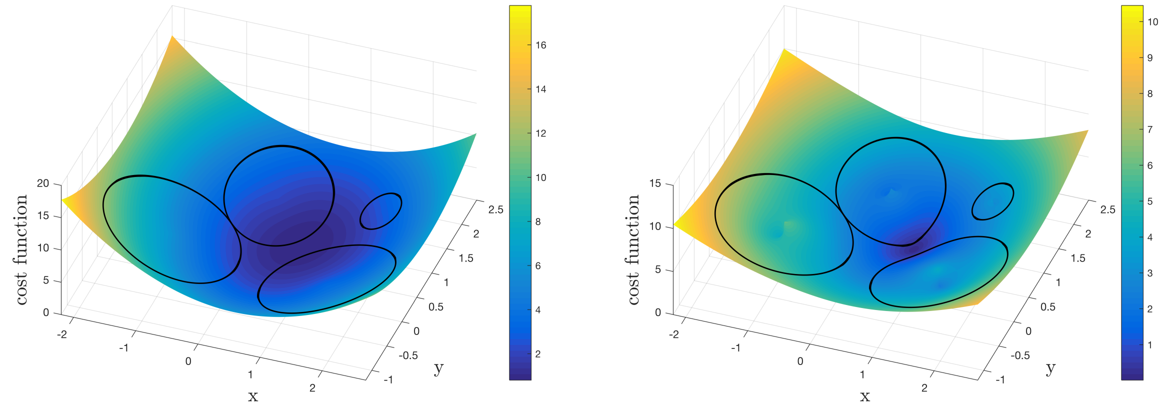
$$L = \begin{bmatrix} -1 & 0 & 1.5 & 1.6 \\ 0 & 1.5 & 0 & 1.6 \end{bmatrix}^T, \qquad \vec{\xi} = \begin{bmatrix} 1 & 0.8 & 1 & 0.3 \end{bmatrix}^T. \tag{7.21}$$



(a) Cost function $J(\vec{x})$

(b) Gradient of cost function $||\nabla J(\vec{x})||$

Figure 7.3: Visualization of hyperbolic cost function. Function is evaluated using four anchor points in configuration (Eq. 7.21).



(a) Cost function $J(\vec{x})$

(b) Gradient of cost function $||\nabla J(\vec{x})||$

Figure 7.4: Visualization of spherical cost function defined by (Eq. 7.13) using $\alpha = 1$ without covariance matrix. Function is evaluated using four anchor points in configuration (Eq. 7.21).

7.5 Initial location problem

Nonlinear iterative solvers requires initial point from which algorithm estimates next steps. Generally we could express working of iterative methods as

$$x_{n+1} = f(x_n, ...) \,, \tag{7.22}$$

where $f(x_{\mathrm{n}}, ...)$ is a function of a previous step and optionally other parameters. Hence for a first iteration we need to specify initial point x_0. One method is to estimate location by hyperbolic algorithm which doesn't require initial location to work and use it as initial point in nonlinear solver. Hyperbolic algorithm could be computationally expensive due to matrix multiplications (complexity on average $\mathcal{O}(n^3)$). Therefore we assume that unknown location lies in close radius to UAVs we could guess our initial point as weighted location of all UAVs

$$\vec{x}_0 = \frac{1}{M} \sum_{i=1}^{M} \vec{L}_i \,. \tag{7.23}$$

We simulated dataset of 100 randomly distributed points for a different number of anchor points and non of them showed advantage of using hyperbolic solution over average location (Eq. 7.23).

7.6 Accuracy improvements using measurement uncertainty

In general case where real error distribution is not known we could try to use normal distribution as our model. Let Λ_i be measurement function of random variable $\vec{x}$. Λ_i specifies one measurement of one anchor point as taking sample from normal distribution $\mathcal{N}(\vec{\xi}, \Sigma)$. Likelihood of location $\vec{x}$ with respect to the measurement $\vec{\xi}$ is than defined as

$$\mathcal{L}(\vec{x}; \vec{\xi}, \Sigma) = \frac{1}{\sqrt{\det|2\pi\Sigma|}} \exp\left[-\frac{1}{2}\left(\vec{\xi} - \vec{\Lambda}(\vec{x})\right)^T \Sigma^{-1} \left(\vec{\xi} - \vec{\Lambda}(\vec{x})\right)\right] \,, \tag{7.24}$$

where Σ is covariance matrix for measurements $\vec{\xi}$. Assuming that individual measurements are not correlated to each other[9], covariance matrix is taken to be diagonal matrix $\Sigma = [\sigma_i^2 \delta_{ij}]$. Case of error being correlated is addressed in [11] Optimization problem can be formulated as finding point $\vec{x}$ in which has likelihood function $\mathcal{L}$ it's maximum. This imply relation[10]

$$\vec{x} = \arg\max_{\vec{x}} \mathcal{L} = \arg\min_{\vec{x}} \left[\frac{1}{2}\left(\vec{\xi} - \vec{\Lambda}(\vec{x})\right)^T \Sigma^{-1} \left(\vec{\xi} - \vec{\Lambda}(\vec{x})\right)\right] \,. \tag{7.25}$$

[9] Each UAV is taking measurement ξ without influence of other UAVs. Synchronization of the measurements that are taken into final computation is done independently of others.

[10] Function extreme $\nabla_{\vec{x}} \mathcal{L} = -\mathcal{L} \nabla_{\vec{x}} \left[\frac{1}{2}\left(\vec{\xi} - \vec{\Lambda}(\vec{x})\right)^T \Sigma^{-1} \left(\vec{\xi} - \vec{\Lambda}(\vec{x})\right)\right]$.

Method is sometimes referred as Maximum Likelihood Estimation (MLE) method[12]. Comparing it directly with cost function definition (Eq. 6.1) we can interpret general weight matrix H as being directly inverse of covariance matrix Σ. Based on that we can easily extend our localization method with information about uncertainty of distance measurement $\vec{\xi}$. More complex method for estimating actual accuracy limitations are analysed in a work [13].

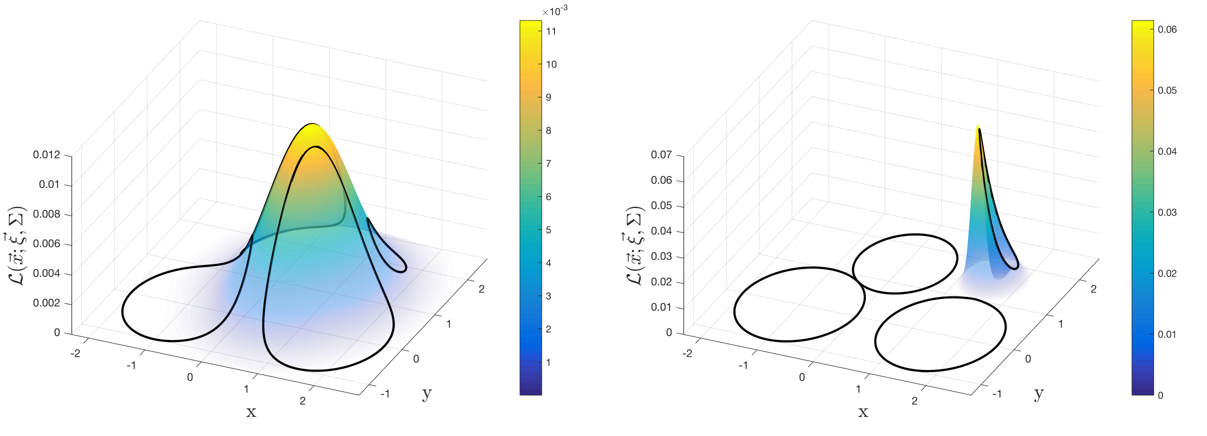
7.7 Correction for BLE distance uncertainty

Given a free space signal propagation model. We can see that distance relation is same as taking sample from log-normal distribution[14]. It can be shown [15] that covariance matrix Σ is equal to

$$\Sigma = [\,\xi_i^4 \delta_{ij}\,]\,. \tag{7.26}$$

Applying this covariance matrix as weight matrix in least square problem can greatly improve overall localization accuracy[14] (for comparison see Fig. 7.5). Covariance stated here cannot be used in hyperbolic algorithm because of different covariance configuration. In [14] it is shown that general covariance matrix for a hyperbolic linear problem has a form (Eq. 7.27). We could use same nonlinear method or solve problem directly using formula (Eq. 6.13) where matrix H is covariance matrix Σ.

$$\Sigma_{ij} = \begin{cases} \sigma_k^2 + \sigma_i^2 & i = j \\ \sigma_k^2 & i \neq j \end{cases}, \qquad k \in \{1, \dots, M\}\,, \qquad i, j \in \{1, \dots, M\}/\{k\}\,. \tag{7.27}$$



(a) Covariance matrix $\Sigma = I$ (b) Covariance matrix $\Sigma = [\xi_i^4 \delta_{ij}]$

Figure 7.5: Likelihood function evaluated for a normal and log-normal model in configuration (Eq. 7.21).

8 Implementation

Multilateration methods described in sections (7.7) and (7.4) are implemented as ROS module onboard of UAVs. One UAV is chose to be control unit and is synchronizing bluetooth topics over ROS. Module is integrated with system Gazebo for UAV simulations. From collected measurements is then computed location of unknown target in local cartesian coordinates as x_t, y_t, z_t and broadcasted in fixed time interval. Broadcasting is done on separate topic on which can other module be subscribed. Because bluetooth devices Nordic nRF52DK were limited to broadcasting RSSI measurements every 100ms we chose sampling period of multilateration to be 1 second.

8.0.1 Performance

We compared run time of the multilateration algorithm for $\boldsymbol{N}$ drones. Every test is run with anchor point positions generated from normal distribution. Simulation are performed on computer with processor Intel Core i7-6660U@2.4GHz and 16GB@1866MHz DDR3 RAM memory. For each anchor point configuration we compute running time of an algorithm over 100 randomly generated sample points.

$$\xi_i = ||\vec{p}_{\text{target}} - \vec{L}_i|| + \nu \,, \qquad \nu \sim \mathcal{N}(0, k)\,, \tag{8.1}$$

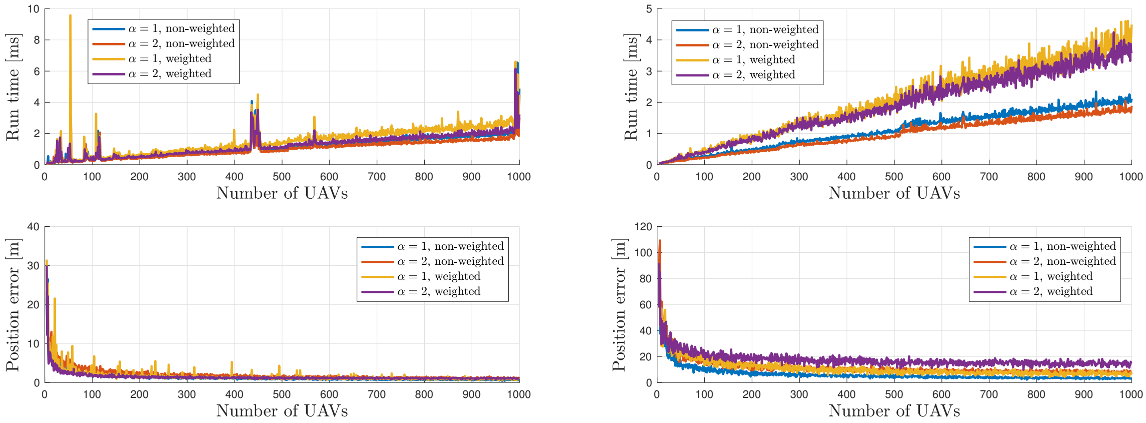
where k is uncertainty parameter used for testing sensitivity of algorithm. Average times are shown in table 2. Conducted experiments were using up to 4 UAVs. Average running time 37 μs introduces very little overhead and method can be used onboard of an UAVs in real time.

Table 2: C++ implementation performance

	$\Sigma = I$		$\Sigma = [\,\xi_i^4 \delta_{ij}\,]$	
UAVs	$\alpha = 1$	$\alpha = 2$	$\alpha = 1$	$\alpha = 2$
4	37 μs	34 μs	32 μs	30 μs
5	65 μs	61 μs	55 μs	56 μs
6	67 μs	49 μs	45 μs	54 μs
7	59 μs	40 μs	39 μs	48 μs
8	57 μs	40 μs	40 μs	41 μs
9	58 μs	58 μs	48 μs	48 μs
10	60 μs	51 μs	50 μs	52 μs

Comparison of different methods form previous sections. Parameter α is an exponent defined in (Eq. 7.17). Covariance matrix is evaluated as BLE correction discussed is section (7.7)

Since we cannot make experiments with more than 4 UAVs (due to hardware unavailability) we will compare influence of many UAVs on accuracy and run-time only by simulation. Simulations shows that our implementation has linear time complexity $\mathcal{O}(M)$ with respect to number of UAVs used for multilateration as shown in (Fig. 8.1). Average boundary of run-time 1ms is crossed for $\sim$ 400 UAVs. Closed form solution of hyperbolic method has disadvantage of general time complexity of $\mathcal{O}(M^3)$. In case of many anchor points it's recommended to use non-linear solver for solving linear problem.



(a) Low signal uncertainty

(b) High signal uncertainty

Figure 8.1: Comparison of non-linear solvers. Left figure shows behaviour when distance measurements are stable and reliable, meaning have variance from real value around 5% (eg. 1.00 ± 0.05 m.). Second figure shows variance around 50%. Spikes represent situations during which is matrix L near singular (discussed in section (7.1)). Therefore optimizer needs more steps to correctly find local minimum.

9 Experiments

We have already discussed experiments measuring RSSI values in section 5. This section presents summary of other multilateration experiments. We have prepared four experiments to test different localization techniques. Due to technical error during the recording of log files we could not conclude any results from the first and the second experiment.

Mean and variance values are estimated from measurement samples by formulas

$$\bar{e} = \frac{1}{M} \sum_{i=1}^{M} e_i \,, \tag{9.1}$$

and

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{M} (e_i - \bar{e})^2 \ . \tag{9.2}$$

9.1 Static anchor points

Experiment has layout of 4 UAVs as receivers and one UAV as transmitter. Receiver positions are static represented by matrix L

$$L = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 10 & 1.5 \\ 10 & 0 & 1.7 \\ 10 & 10 & 0 \end{bmatrix} . \tag{9.3}$$

Receivers formed a rectangle of side length 10 meters and transmitter was moved along predefined path in constant height of 2 meters (Fig. 9.1).

Average localization errors during the experiment indicates instability of a signal resulting in approximate location information within the area formed by UAVs (Fig. 9.2). Table (3) represents coordinates and absolute error of compute location of unknown object. Distance and locations error were measured at given time as differences of real and computed value.

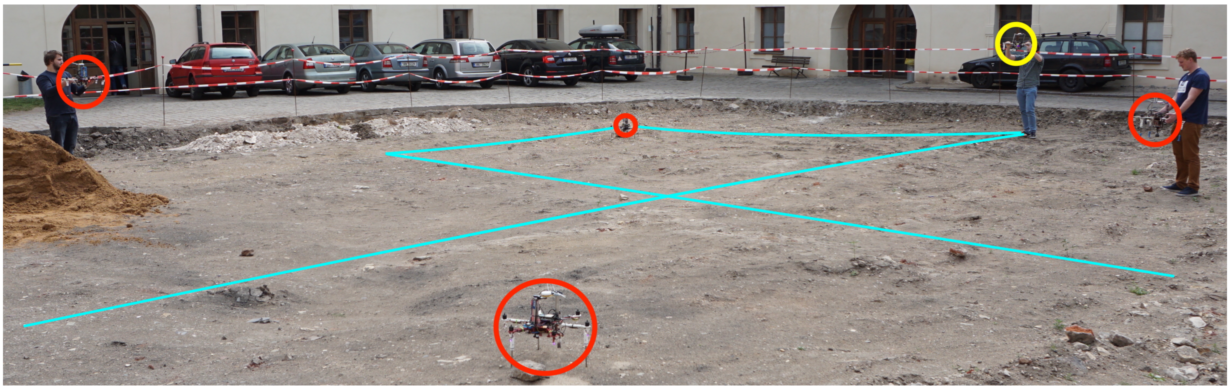
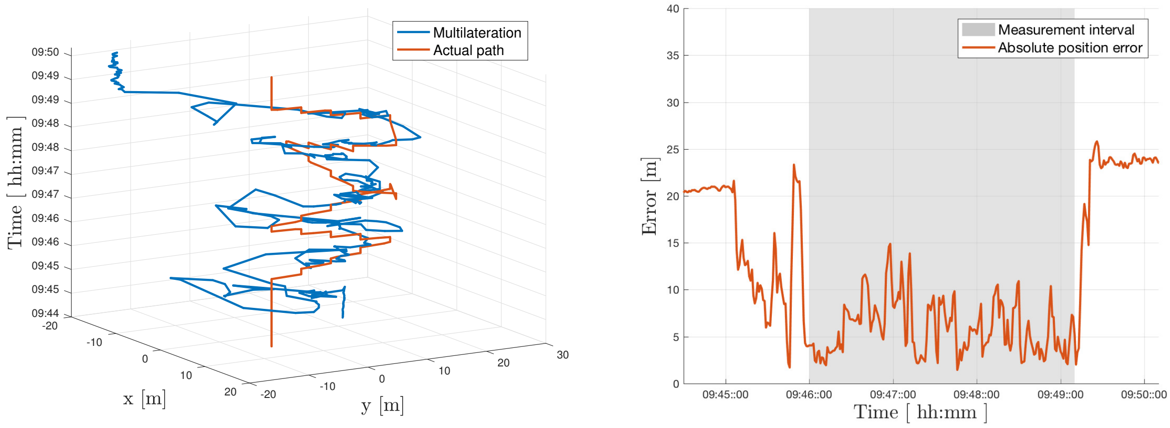


Figure 9.1: Experiment overview. Receivers (red circles) are placed on static positions. Transmitter (yellow circle) moves along blue line in constant height.



(a) Time series of XY trajectory plane

(b) Absolute position error during experiment

Figure 9.2: Experiment overview. Multilateration algorithm with BLE covariance matrix using 4 UAVs as receiver points and one UAV as transmitter.

Table 3: Unknown target location error

coordinate	$\bar{e}$ [m]	σ [m]
x	3.15	2.95
y	3.66	2.87
z	1.70	0.80
r*	5.95	2.90

$$^{*}\mathbf{r} = \sqrt{x^2 + y^2 + z^2}$$

9.2 Moving anchor points

The last experiment was conducted with collaboration on UAVs formation planning [16]. Data presented here are part of multilateration algorithm used during the experiment. Measurements Figure 9.4 and Table 4 shows similar results as previous experiment.

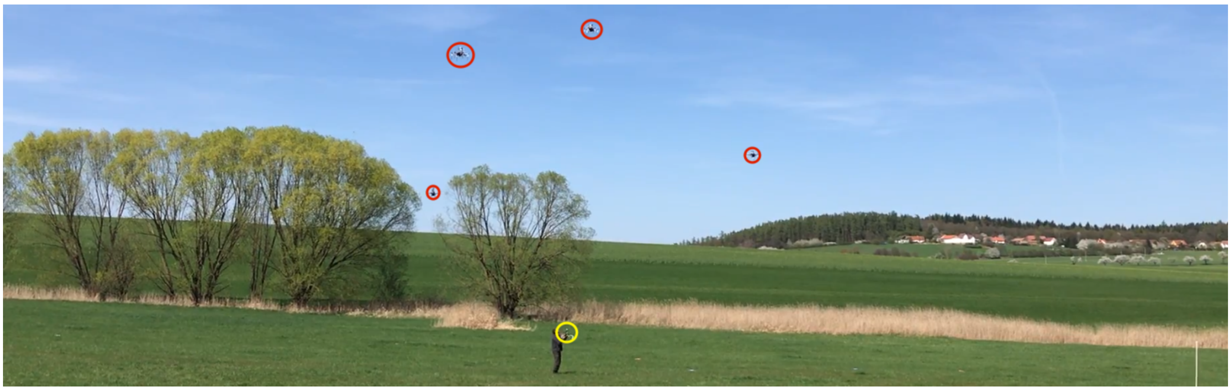
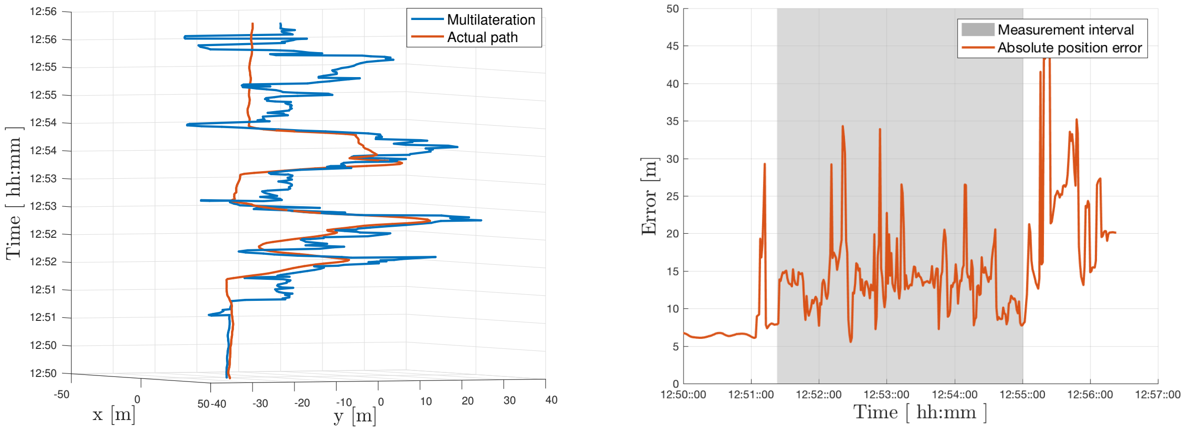


Figure 9.3: Experiment overview. Receivers (red circles) are moving in formation based on the estimated location [16] of the transmitter (yellow circle).



(a) Time series of XY trajectory plane

(b) Absolute position error during experiment

Figure 9.4: Experiment overview. Multilateration algorithm with BLE covariance matrix is used using 4 UAVs as receiver points and one UAV as transmitter. UAVs with transmitter are moving according to trajectory planning algorithm described in [16].

Table 4: Unknown target location error

coordinate	$\bar{e}$ [m]	σ [m]
x	5.91	4.46
y	6.14	3.60
z	4.93	4.13
r*	11.158	4.7431

$$^*\mathbf{r} = \sqrt{x^2 + y^2 + z^2}$$

Overall localization error during the experiment shows more uncertainty then static case. This was mainly due to dynamically moving target which resulted in slower response time of the filter because error is computed with respect to actual location in space and computed location.

10 Conclusion

This thesis presented implementations and overview of variety of multilateration techniques. First we conducted experiments designed to measure RSSI characteristics (Section (5)). These experiments provided ground for further use of multilateration methods.

We have presented linear (Section 7.3) and non-linear (Section (7.4)) algorithms used in multilateration system to estimate location of an unknown object. Signal propagation model, used for determining RSSI characteristic, was used for estimating covariance matrix in order to estimate location more accurately then without it. Localization module was integrated with Gazebo system and used to test each experiment by simulation.

Average localization error during the outdoor experiment was computed to be 8.4 meters. Such accuracy does not corresponds with other experiments conducted by others [1],[17],[18] showing accuracy up to 1m. To further confirm our observations we would need more experiments to be conducted in a way of measuring localization with BLE technology. Despite experiment results such accuracy can be still used for proximity localization of objects in open field where higher accuracy is not needed.

11 Future work

This thesis provides method of calculating unknown location of a transmitter based on the measurements of RSSI values. Improvements to this system could be made via extending proposed method with more information about the environment, such as Earth location, geographical data or other uncertainties that are appliable in real world applications.

11.1 Uncertainty of UAV location

In section (7.6) we discussed how to improve localization algorthms using covariance matrix of distance measurement uncertainty. During that we assumed location of UAVs to be correct without any error of measurement. Conducted experiments used differential GPS location with position accuracy up to 15mm and UAV position error was negligible in comparison to measured distances which were in range 1 to 42 meters. Other applications that are not using accurate positing would benefit from localization method capable of using such information.

11.2 Localization on Earth

Method discussed in this work were working in orthonormal cartesian space. During the experiments we have created such space by defining origin in one specific place and related positions of UAVs with respect to it from GPS readings. In this section we discuss possible creation of such coordinate system without known reference point. Since main positioning system used in outdoor is GPS, we are going to discuss methods used for a creating so called Local cartesian coordinates on the surface of a earth. GPS system uses reference ellipsoid WGS84. Earth could be characterized by a 3 coordinates, latitude, longitude and height from reference ellipsoid. We can represent surface of a reference ellipsoid by parametric equations

$$\vec{r}_e(u,v) = \begin{bmatrix} a\cos(u)\cos(v) \\ b\cos(u)\sin(v) \\ c\sin(u) \end{bmatrix}, \qquad u \in \left\langle -\frac{\pi}{2}, \frac{\pi}{2} \right\rangle, \qquad v \in \langle -\pi, \pi \rangle, \tag{11.1}$$

where r_e is position vector in cartesian coordinates on surface of an ellipsoid, u is latitude and v is longitude coordinate. Constants a,b and c are defined by standard WGS84. Location on the surface of the Standalone UAV could than approximate it's local coordinate system by taking normal plane tangent to the ellipsoid in it's latitude and longitude coordinates. Local cartesian coordinate system is than defined by 3 basis vectors $\vec{e}_x$,$\vec{e}_y$ and $\vec{e}_z$ created on the surface

$$\vec{e}_x = \frac{\frac{\partial \vec{r}_e}{\partial v}}{||\frac{\partial \vec{r}_e}{\partial v}||}, \qquad \vec{e}_y = \frac{\frac{\partial \vec{r}_e}{\partial u}}{||\frac{\partial \vec{r}_e}{\partial u}||}, \qquad \vec{e}_z = \frac{\vec{e_x} \times \vec{e_y}}{||\vec{e_x} \times \vec{e_y}||}. \tag{11.2}$$

Location of other UAVs in such system would be determined by other localization method. Namely we could use relative visual localization [19] or techniques that uses Light Detection And Ranging (LIDAR) devices [20].

11.3 Horizon boundary

Computation of ellipsoid horizon boundary can be used for determining if our received signal is bent over the surface and thus we need to use correction in our estimation to account for a curved surface.

```
function [curve] = ellipsoid_horizon(azi, lat, lng, h, a, b, c)
    syms u v
    Kes = [1/a 0 0; 0 1/b 0; 0 0 1/c];
    re = @(u,v) [a.*cos(u).*cos(v);
                 b.*cos(u).*sin(v);
                 c.*sin(u)];
    diff_re = @(du,dv) [eval(subs(diff(re, u),[u v],[du dv]))
                        eval(subs(diff(re, v),[u v],[du dv]))];
    diff_rs = @(du,dv) Kes*diff_re(du,dv);
```

Script 2: Matlab implementation of (Eq. 11.1),(Eq. 11.3)

Computation of a horizon can be done on the surface of the sphere and then transformed by transformation matrix from sphere surface back to ellipsoid surface. We can specify such transformation matrix T_e^s in a form

$$\vec{r_s}(u,v) = \begin{bmatrix} \cos(u)\cos(v) \\ \cos(u)\sin(v) \\ \sin(u) \end{bmatrix}, \quad T_e^s = \begin{bmatrix} 1/a & 0 & 0 \\ 0 & 1/b & 0 \\ 0 & 0 & 1/c \end{bmatrix}, \quad \vec{r_s}(u,v) = T_e^s \vec{r_e}(u,v)\,. \tag{11.3}$$

Goal is to compute boundary of horizon that is visible from geographical location $\vec{p} = \begin{bmatrix} u_p & v_p & h_p \end{bmatrix}$, where u is latitude, v is longitude and h is orthogonal height from geographical position on the surface. Because normal surface vector $\vec{r_n}$ on the ellipsoid is not generally collinear with position vector $\vec{r_s}$ we need to recompute spherical angles of $\vec{R}$

$$\vec{r_n}(u,v) = \frac{\frac{\partial \vec{r_e}}{\partial v} \times \frac{\partial \vec{r_e}}{\partial u}}{||\frac{\partial \vec{r_e}}{\partial v} \times \frac{\partial \vec{r_e}}{\partial u}||}, \qquad \vec{R}(\vec{p}) = \vec{r_e}(u_p, v_p) + h_p \vec{r_n}(u_p, v_p)\,. \tag{11.4}$$

```
    R = re(lat,lng);
    dR = diff_re(lat,lng);
    dredu = dR(:, 1)/norm(dR(:, 1));
    dredv = dR(:, 2)/norm(dR(:, 2));
    enorm = cross(dredv,dredu);
    enorm = enorm./norm(enorm);
    Re = (R+h*enorm);
    Rs = Kes*Re;
```

Script 3: Matlab implementation of (Eq. 11.4)

First step is to compute absolute position $\vec{p}$ in reference cartesian coordinates. Because we are specifying normal height above the ellipsoid we need to construct absolute position vector with respect to normal vector in $\begin{bmatrix} u & v \end{bmatrix}$ surface coordinates. Next step is to rescale ellipsoid to unit sphere together with our reference location $\vec{R}$. Height above unit sphere is than used for computing horizon boundary in spherical space

$$h = ||T\vec{R}|| - 1\,. \tag{11.5}$$

Horizon boundary can be solved as two dimensional problem (Fig. 11.1) where

$$l^2 = h^2 + 2hr\,, \qquad d = \frac{l^2}{r+h}\,, \qquad o = h + r - d\,, \qquad k = \sqrt{r^2 - o^2}\,. \tag{11.6}$$

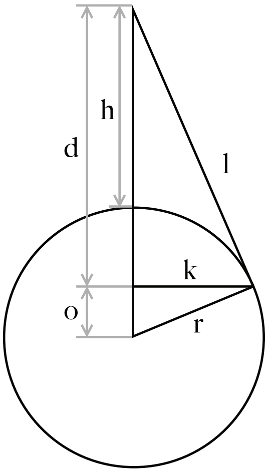


Figure 11.1: Horizon boundary on the sphere.

Transforming parameters to boundary curve can be done by specifying position vector $\vec{r_c}$ with respect to basis of a sphere where z-axis base is in direction of our target point $\vec{R}$, xy-plane basis can be than computed from spherical basis at point u_{sp}, v_{sp}. Inverse parameters could be obtained from inverse spherical transformation

$$\Phi(\vec{r}) = \begin{bmatrix} u(\vec{r}) \\ v(\vec{r}) \end{bmatrix} = \begin{bmatrix} \sin^{-1}(z/||\vec{r}||) \\ \tan^{-1}(y/x) \end{bmatrix}\,. \tag{11.7}$$

```
    h_scaled = norm(Rs) - 1;
    l2 = h_scaled^2 + 2*h_scaled;
    d_scaled = l2/(1 + h_scaled);
    x_scaled = h_scaled + 1 - d_scaled;
    k_scaled = sqrt(l2-d_scaled^2);
```

Script 4: Matlab implementation of (Eq. 11.5),(Eq. 11.6).

Basis vectors could be then computed as column vectors from spherical coordinates.

$$\vec{e}_z = \frac{T\vec{R}}{||T\vec{R}||}\,, \qquad \vec{e}_x = \left.\frac{\partial \vec{r}_s}{\partial v}\right|_{\Phi(T\vec{R})}\,, \qquad \vec{e}_y = \left.\frac{\partial \vec{r}_s}{\partial u}\right|_{\Phi(T\vec{R})}\,. \tag{11.8}$$

Horizon curve in spherical space is than transformed back to the elliptical space by inverse transformation and change of basis $B = \left[\begin{array}{ccc} \vec{e}_x & \vec{e}_y & \vec{e}_z \end{array}\right]$

$$\vec{r}_{s,c}(\varphi) = \begin{bmatrix} k\cos(\varphi) \\ k\sin(\varphi) \\ o \end{bmatrix}\,, \qquad \vec{r}_c(\varphi) = \left(T_e^s\right)^{-1} B\vec{r}_{s,c}(\varphi)\,. \tag{11.9}$$

```
    [u_rs, v_rs, ~] = cart2sph(Rs(1), Rs(2), Rs(3));
    dRs = diff_rs(v_rs, u_rs);
    ex = dRs(:,1);
    ey = dRs(:,2);
    ez = Rs/norm(Rs);
    bx = k_scaled*cos(azi).*ex;
    by = k_scaled*sin(azi).*ey;
    bz = (zeros(1,length(azi))+x_scaled).*ez;
    curve = Kes\(bx + by + bz);
end
```

Script 5: Matlab implementation of (Eq. 11.8),(Eq. 11.9).

Function $\vec{r}_c(\varphi)$ is parametric curve in reference cartesian coordinates $\vec{r}_c : \varphi \in \langle 0, 2\pi) \to \mathbb{R}^3$, where φ is azimuthal angle measured from direction of pole. Horizon boundary forms an ellipse with major-axis $a = ||k\left(T_e^s\right)^{-1}\vec{e}_x||$ and semi-axis $b = ||k\left(T_e^s\right)^{-1}\vec{e}_y||$.

11.3.1 Orthogonality invariance under linear transformation

When solving horizon boundary in spherical space we assumed that vector $\vec{x} - \vec{r}_c(\varphi)$ remains tangent to the surface of an ellipsoid after we transform back from spherical space. Following proof shows that any linear transformation T wont effect orthogonality of vectors.

We have computed boundary curve on the sphere such that triple scalar product (Eq. 11.10) is zero.

$$\left(\frac{\partial \vec{r}_s(u,v)}{\partial u} \times \frac{\partial \vec{r}_s(u,v)}{\partial v}\right) \cdot \left.\left(\vec{x}(u,v) - \vec{x}_0\right)\right|_{\Phi(\vec{r}_{s,c}(\varphi))} = 0 \tag{11.10}$$

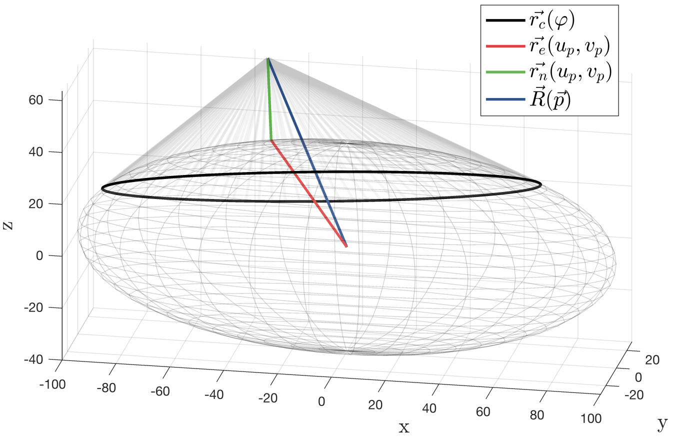


Figure 11.2: Horizon boundary computed on arbitrary ellipsoid. Boundary curve $\vec{r_c}(\varphi)$ evaluated on full interval $\langle 0, 2\pi)$. Computation of complementary vectors $\vec{r_n}$ and $\vec{r_e}$ is specified by (Eq. 11.4).

Transforming every point to the ellipsoid coordinates using transformation matrix T. This will lead for an equation (Eq. 11.11)

$$\left[\frac{\partial T\vec{r}_s(u,v)}{\partial u} \times \frac{\partial T\vec{r}_s(u,v)}{\partial v}\right] \cdot [T(\vec{x}_s - \vec{x}_0)] = 0\,, \tag{11.11}$$

$$\left[\left(T\frac{\partial \vec{r}_s(u,v)}{\partial u}\right) \times \left(T\frac{\partial \vec{r}_s(u,v)}{\partial v}\right)\right] \cdot [T(\vec{x}_s - \vec{x}_0)] = 0\,. \tag{11.12}$$

From geometric interpretation of triple scalar product (Eq. 11.12) as volume of Parallelepiped[11] we can factor out matrix multiplication as

$$\det(T)\left[\left(\frac{\partial \vec{r}_s(u,v)}{\partial u} \times \frac{\partial \vec{r}_s(u,v)}{\partial v}\right) \cdot (\vec{x}_s - \vec{x}_0)\right] = 0\,. \tag{11.13}$$

By combining (Eq. 11.10) and (Eq. 11.13) we have proved that ortogonality of vectors is preserved under linear transformation.

[11] Triple scalar product identity $\left(T\vec{a} \times T\vec{b}\right) \cdot (T\vec{c}) = \det T\left(\vec{a} \times \vec{b}\right) \cdot (\vec{c})$

References

[1] Vrba Matouš. Active searching of rfid chips by a group of relatively stabilized helicopters. Bachelor's thesis at CTU in Prague, 2016.

[2] Jiang Chaoshu, Liu Changzhong, and Wang Xuegang. Gps synchronized wide area multilateration system. In *2009 International Conference on Communications, Circuits and Systems*, pages 457–459, July 2009.

[3] F. Lazzari, A. Buffi, P. Nepa, and S. Lazzari. Numerical investigation of an uwb localization technique for unmanned aerial vehicles in outdoor scenarios. *IEEE Sensors Journal*, 17(9):2896–2903, May 2017.

[4] G. Greco, C. Lucianaz, S. Bertoldo, and M. Allegretti. Localization of rfid tags for environmental monitoring using uav. In *2015 IEEE 1st International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, pages 480–483, Sept 2015.

[5] H. Wang, B. Yan, X. Li, X. Luo, Q. Yang, and W. Yan. On optimal path planning for uav based patrolling in complex 3d topographies. In *2016 IEEE International Conference on Information and Automation (ICIA)*, pages 986–990, Aug 2016.

[6] K. S. Lee, M. Ovinis, T. Nagarajan, R. Seulin, and O. Morel. Autonomous patrol and surveillance system using unmanned aerial vehicles. In *2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*, pages 1291–1297, June 2015.

[7] B. Fennani, H. Hamam, and A. O. Dahmane. Rfid overview. In *ICM 2011 Proceeding*, pages 1–5, Dec 2011.

[8] I. Kim, S. Xu, and Y. Rahmat-Samii. Generalised correction to the friis formula: quick determination of the coupling in the fresnel region. *IET Microwaves, Antennas Propagation*, 7(13):1092–1101, October 2013.

[9] I. Ahmad, R. Asif, R. A. Abd-Alhameed, H. Alhassan, and et al. Current technologies and location based services. In *2017 Internet Technologies and Applications (ITA)*, pages 299–304, Sept 2017.

[10] D. Yildiz, S. Karagol, and O. Ozgonenel. A hyperbolic location algorithm for various distributions of a wireless sensor network. In *2017 5th International Istanbul Smart Grid and Cities Congress and Fair (ICSG)*, pages 75–79, April 2017.

[11] S. J. Julier and J. K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, volume 4, pages 2369–2373 vol.4, Jun 1997.

[12] H. Yu and Y. Jiang. Maximum likelihood network localization using range estimation and gps measurements. In *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6, Oct 2017.

[13] H. B. Lee. Accuracy limitations of hyperbolic multilateration systems. *IEEE Transactions on Aerospace and Electronic Systems*, AES-11(1):16–29, Jan 1975.

[14] P. Tarrio, A. M. Bernardos, J. A. Besada, and J. R. Casar. A new positioning technique for rss-based localization based on a weighted least squares estimator. In *2008 IEEE International Symposium on Wireless Communication Systems*, pages 633–637, Oct 2008.

[15] Ana M. Bernardos Tarrío, Paula and José R. Casar. Weighted least squares techniques for improved received signal strength based localization. *Molecular Diversity Preservation International (MDPI)*, 11(9):8569–8592, 2011.

[16] Dan Šuster. Reactive motion planning of a formation of helicopters for mapping of transmission sources position. Bachelor's thesis at CTU in Prague, 2018.

[17] M. E. Rida, F. Liu, Y. Jadi, A. A. A. Algawhari, and A. Askourih. Indoor location position based on bluetooth signal strength. In *2015 2nd International Conference on Information Science and Control Engineering*, pages 769–773, April 2015.

[18] Y. Wang, Xu Yang, Yutian Zhao, Yue Liu, and L. Cuthbert. Bluetooth positioning using rssi and triangulation methods. In *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, pages 837–842, Jan 2013.

[19] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajnik, J. Faigl, G. Loianno, and V. Kumar. System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization. *Autonomous Robots*, 41(4):919–944, 2017.

[20] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, and A. Savvaris. Lidar-inertial integration for uav localization and mapping in complex environments. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 649–656, June 2016.

Appendix A List of abbreviations

In Table 5 are listed abbreviations used in this thesis.

Abbreviation	Meaning
LM	Levenberg Marquardt
LSM	Least squared method
WLSM	Weighted least squared method
BLE	Bluetooth low energy
RSSI	Received Signal Strength Indication
ROS	Robot Operating System

Table 5: Lists of abbreviations

Appendix B CD content

In Table 6 are listed names of all root directories saved on the CD. Directories contains README with information about its content.

Directory	Description
cpp	c++ source code
firmware	Nordic nRF52DK firmware payloads
matlab	matlab implementations and experiment processing code
photos	photos from experiments
thesis	Bachelor's thesis source files
trajectory	trajectories for RSSI expetiments

Table 6: Lists of root directories

Appendix C MATLAB code

C.1 Spherical cost functions

Computation of cost function (Eq. 6.1) specified by error vector from spherical localization algorithm (7.17) and it's respective derivatives $\nabla\vec{J}$ and $\nabla^2\vec{J}$. Function is evaluated using identity H matrix and is implementing parameter α (7.19)

```matlab
function [J] = spherical_cost_fn(x,xi,L,alpha)
    J = 0;
    [M,N] = size(L);
    for i=1:M
        lambda = 0;
        for j=1:N
            lambda = lambda + (x(j)-L(i,j))^2;
        end
        e = xi(i)^alpha - lambda^(alpha/2);
        J = J + e.^2;
    end
    J = J/2;
end
```

Script 6: Matlab implementation of spherical cost function

```matlab
function [dJ] = spherical_cost_fn_grad(x,xi,L,alpha)
    [M,N] = size(L);
    dJ = zeros(1,N);
    for i=1:M
        lambda = 0;
        for j=1:N
            lambda = lambda + (x(j)-L(i,j))^2;
        end
        e = xi(i)^alpha - lambda^(alpha/2);
        f = -alpha*e*lambda^(alpha/2-1);
        for j=1:N
            dJ(j) = dJ(j) + f.*(x(j)-L(i,j));
        end
    end
end
```

Script 7: Matlab implementation of gradient $\nabla J = \left[\frac{\partial J}{\partial x_i}\right]$, see (Eq. 6.7).

```
function [H] = spherical_cost_fn_hessian(x,xi,L,alpha)
    [M,N] = size(L);
    H = zeros(N,N);
    for j=1:N
        for k=1:N
            for i=1:M
                lambda = 0;
                for li=1:N
                    lambda = lambda + (x(li)-L(i,li))^2;
                end
                e = xi(i)^alpha - lambda^(alpha/2);
                dxj = (x(j)-L(i,j));
                dxk = (x(k)-L(i,k));
                dl_dxj = alpha*lambda^(alpha/2-1)*dxj;
                dl_dxk = alpha*lambda^(alpha/2-1)*dxk;
                d2l_dxj_dxk = alpha*(alpha-2)*lambda^(alpha/2 - 2)*dxj*dxk;
                if j == k
                    d2l_dxj_dxk = d2l_dxj_dxk + alpha*lambda^(alpha/2-1);
                end
                H(j,k) = H(j,k) + dl_dxj*dl_dxk - e*d2l_dxj_dxk;
            end
        end
    end
end
```

Script 8: Hessian of spherical cost function $H_{jk}(\vec{x}) = \frac{\partial^2 J(\vec{x})}{\partial x_j \partial x_k}$, see (Eq. 6.8). Covariance matrix is taken to be identity matrix. In case of diagonal matrix we can extend algorithm applying relation (Eq. 6.5).