CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

BACHELOR'S THESIS

Dan Šuster

Reactive motion planning of a formation of helicopters for mapping of transmission sources position

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: **Šuster Dan** Personal ID number: **457158**

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:

Reactive motion planning of a formation of helicopters for mapping of transmission sources position

Bachelor’s thesis title in Czech:

Reaktivní plánování pohybu formace helikoptér v úloze mapování pozic zdrojů vysílání

Guidelines:

A motion planning algorithm will be designed that enables to react on currently estimated positions of transmission sources to improve a map quality in the next measurement step. The following tasks will be solved:
- To design and implement different coverage and reactive methods for control a formation of Micro Aerial Vehicles (MAVs) in the task of cooperative mapping locations of transmission sources by measuring the intensity of transmission in multiple locations.
- To design and implement a motion prediction algorithm (using Kalman filter) to estimate the position of moving transmission sources.
- To integrate the methods into the ROS system being designed at MRS group, CTU in Prague [1,2].
- To verify system functionalities in Gazebo and with real outdoor UAVs.
- To statistically compare the performance of different motion planning approaches.

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] A. Buffi, P. Nepa and R. Cioni, 'SARFID on drone: Drone-based UHF-RFID tag localization,' 2017 IEEE International Conference on RFID Technology & Application (RFID-TA), 2017.
[4] M. Lehner, M. Eberhardt, A. Ascher and E. Biebl, 'An active UHF RFID transponder for fawn saving during pasture mowing,' 2015 German Microwave Conference, 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: **30.01.2018** 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 throughout the thesis. Furthermore, I would like to thank Ing. Tomáš Báča and the whole Multi-robot Systems group for great advice during the experiments. I would also like to thank my family for their continuous support throughout my studies.

Abstract

This thesis deals with the design, implementation, simulation, and experimental verification of two approaches to motion planning and formation design for localization of an unknown transmission source by a group of unmanned helicopters. The mentioned localization is based on the principle of multilateration. The first method proposes a fast-reacting motion planning for following of the moving target, where a design of Kalman filter is proposed for its tracking. The second method waits for more estimations of the position of the target and then moves towards the estimated position. The formation consists of fully autonomous helicopters with Bluetooth Low Energy development boards attached to them. Furthermore, the precision of the target localization for both moving and static target during a real-world experiment is presented.

Keywords: UAV, motion planning, unmanned helicopter, localization, target tracking, multi-robot, formation, Kalman filter, RSSI, Bluetooth Low Energy, multilateration

Abstrakt

Tato práce se zabývá návrhem, implementací, simulací a experimentálním ověřením dvou přístupů plánování pohybu formace bezpilotních helikoptér v úloze lokalizace neznámého zdroje vysílání. Zmiňovaná lokalizace je postavena na principu multilaterace. První metoda spočívá v návrhu rychle reagujícího plánovacího algoritmu pro sledování pohybujícího se cíle, kde odhad jeho pozice je počítám pomocí Kalmanova filtru. Druhá metoda vyčkává několika měření odhadu pozice neznámého cíle a poté se přiblíží k jeho odhadu. Formaci tvoří skupina autonomních bezpilotních helikoptér, které na sobě mají uchycené vývojové desky podporující Bluetooth Low Energy. V této práci je dále diskutována přesnost lokalizace z dat získaných z realného experimentu při lokalizaci pohyblivého a statického cíle.

Klíčová slova: UAV, plánování pohybu helikoptér, bezpilotní helikoptéra, lokalizace, sledování cíle, multi-robot, formace, Kalman filter, RSSI, Bluetooth Low Energy, multilaterace

Contents

List of Figures

1 Introduction

In recent years, both software and hardware development have moved towards the mobile systems, small robotics, and embedded systems. This led to the development of relatively low-cost Unmanned Aerial Vehicles (UAVs), which are today widely used in the industry and research. The UAVs could be used, for example, to perform security checks, to track assets based on proximity, and to map areas or objects that are hidden from the satellites or that are in hardly accessible places [1].

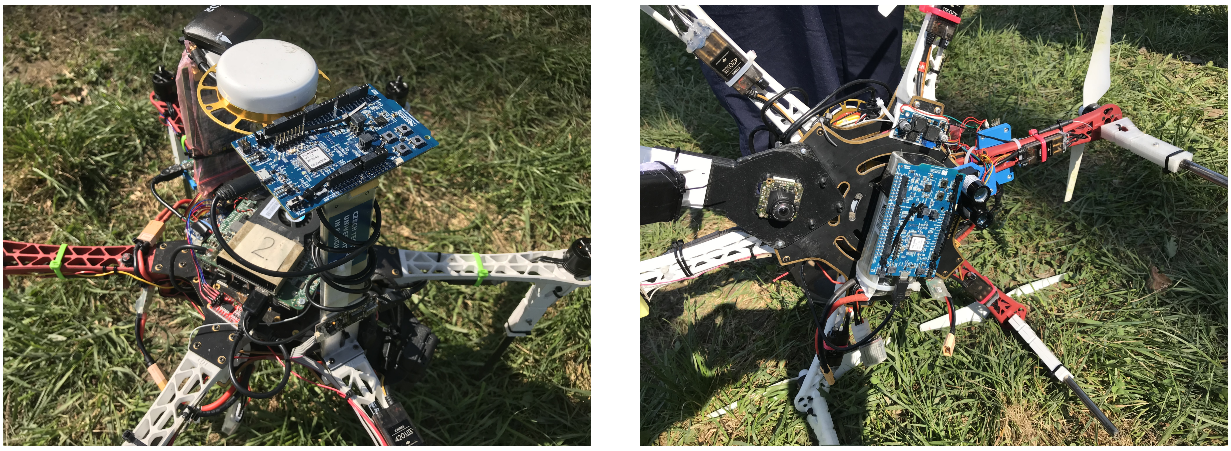
Another use-case is localization and mapping of the unknown transmission sources. The unknown sources could be assets, animals, or another UAVs. For example in [2], the source was a real radiation source and the localization method was based on measurements from a particle detector. Another possibility is to localize the mobile phones based on Bluetooth Low Energy (BLE) received signal strength indicator measurements. Moreover, the same BLE technology is utilized in this thesis for localization of a moving or static target using a team of UAVs. Therefore, two motion planning methods for a formation of UAVs are proposed. The localization uses multilateration principle, which is employed for estimation of the position of the target based on measured distances from the target. The obtained measurements of a low power signal from BLE are usually noisy. Therefore, a Kalman filter is designed to estimate a position of the target, while taking the UAV dynamics into account. The functionality of the proposed methods are verified by the simulations and a real-world experiment.

The real-world experiments were conducted thanks to the long-term research of the Multi-robot Systems group that focuses on formation flying [3] [4] [5] [6], stabilization of the helicopters [7] [8] [9] [10], relative localization [3] [11], and navigation of swarms of UAVs [12] [13].

2 Preliminaries

2.1 Bluetooth low energy (BLE)

The BLE technology, as designed, is a low-power solution for short-range communication [14]. As the development continued, low-power consuming transmitters became an attractive choice for localization applications. Nowadays, it is widely available in mobile phones, notebooks, and low-cost development boards. There are different approaches for localization, but most of them are based on Time Difference Of Arrival (TDOA) or Received Signal Strength Indicator (RSSI) measurement [15] [16]. Throughout the thesis, the experiments (the BLE parts) will be based on nRF52 BLE development board, manufactured by Nordic Semiconductors [17]. There are two versions of the firmware necessary. The first one, denoted as the transmitting board, has to be flashed as a beacon advertising at 10 Hz and the second one, denoted as the receiver board, has to be flashed to scan only for the given major/minor advertisement combination. The major/minor are two numbers that can be set in the firmware code and that are part of the standard BLE payload. The transmitter UAV had the board attached to the top construction part and the receiver UAVs had the board attached to the battery on the bottom side of the construction as shown in the Figure 1.



(a) Photo - Transmitter UAV

(b) Photo - Receiver UAV

Figure 1: Bluetooth low energy development board attached to UAV. The transmitter UAV has the BLE board attached to the top part and the receiver UAV has the board attached to the battery on the bottom side. This placement is so that the line of sight could be preserved and thus the loss of the signal strength loss caused by the metal parts of the construction is minimized. It is assumed that the receiving UAV flies in higher height than the transmitting UAV.

2.1.1 Conversion of Received Signal Strength Indicator (RSSI) to distance

Aforementioned, the receiver board in scan mode discovers advertisements. Along with the base BLE protocol data, the advertisement returns a number of the channel where the message was advertised on and the RSSI value. Now, the distance from the transmitting source can be computed using the formula

$$d(rssi) = ax^b + c, \tag{1}$$

where a,b,$c \in \mathbb{R}$ are constants, d is computed distance and x is called *ratio*. This *ratio* is given as

$$x(rssi) = \frac{rssi}{TXpower}, \tag{2}$$

where $TXpower$ is calibrated RSSI value at 1 m distance and $rssi$ is actual measured value.

The obtained measurements contain noise and many times the fitted curve does not correspond enough to the reality [18]. To enhance the process, distance computations in all the experiments discussed within this thesis will be based on the line-segmented function shown in Figure 2, which is based on real experiments. The RSSI values that are lower than the minimum, or greater than the maximum, are saturated to the minimum, respectively to the maximum value. The lower bound is given by the minimum distance between two UAVs (approx. 3 m) and the upper bound is given as maximum experiment distance estimation.

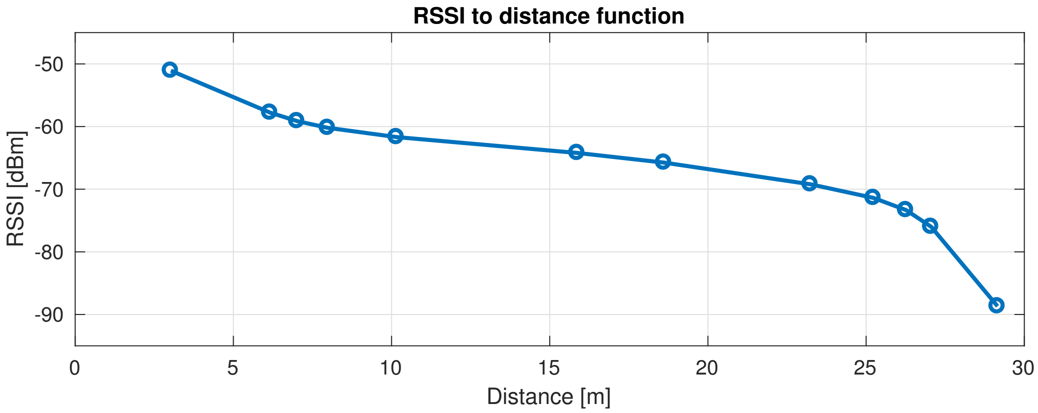


Figure 2: Received signal strength indicator (RSSI) to distance function. The function is used to transfer RSSI obtained by the BLE chip to distance. It is given as line segments to best fit the real experiments behavior. The exact values are shown in Table 1.

RSSI [dBm]	Distance [m]
-51.08	3.02
-57.77	6.16
-59.12	7.01
-60.20	7.98
-61.67	10.15
-64.20	15.87
-65.75	18.61
-69.21	23.24
-71.39	25.23
-73.33	26.26
-75.99	27.05
-88.68	29.15

Table 1: RSSI to distance data table corresponding to the function shown in Figure 2. The first column represents RSSI values in [dBm] and the second column represents distance values in [m].

2.1.2 BLE Channels

The BLE development board scans for all the advertisement channels, indexed as 37, 38 and 39. The RSSI values differ a lot among the channels at the same time, therefore to bring more stability to the RSSI signal, an average of all three channels will be passed as input to distance function described in section 2.1.1. Figure 3 shows the frequencies of the channels. The frequencies are the same as some of the Wi-fi channels, that makes the measurements noisier when interfered with Wi-fi signals.

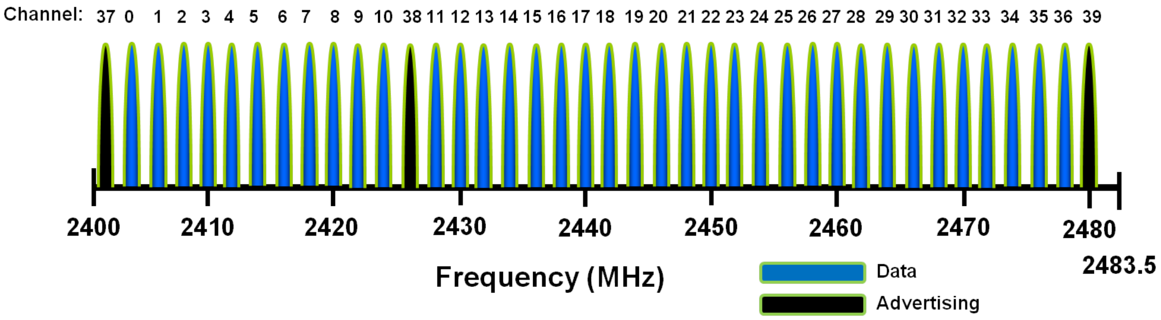

Channel: 37 0 1 2 3 4 5 6 7 8 9 10 38 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 39
2400
2410
2420
2430
2440
2450
2460
2470
2480
2483.5
Frequency (MHz)
Data
Advertising

Figure 3: BLE frequency channels.[1]

[1] Image source: https://microchip.wdfiles.com/local–files/wireless:ble-link-layer-channels/ble-phy-channel-assignment.png

2.2 Multilateration

Multilateration is a technique to determine or estimate a position of the target based on distance measurements from anchors with a known location. The anchors are flying UAVs, therefore the input of multilateration are the last known locations and the last measured distances. The problem is formulated as a non-linear least squares problem, which can be solved using iterative non-linear optimizers, for example by Levenberg–Marquardt algorithm. The problem can be solved in n dimensions, but the aim of this thesis is to localize UAVs in 2D or 3D environments. The 2D version requires a minimum of three anchors and for the 3D version, at least four anchors are required.

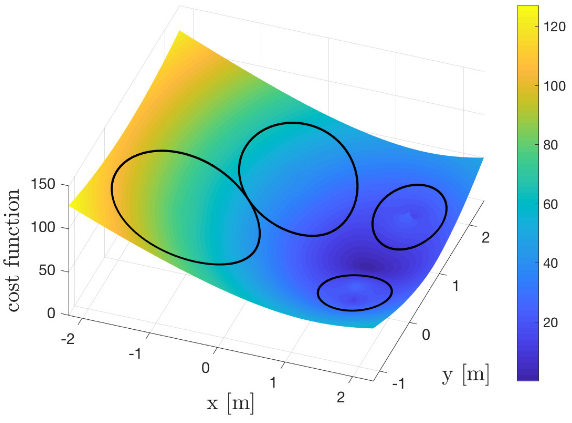


Figure 4: Multilateration - cost function

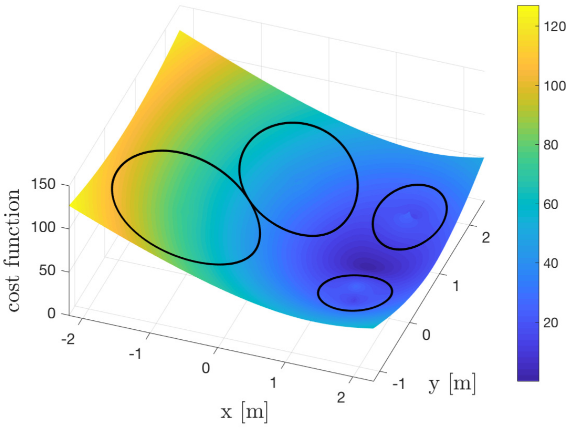


Figure 5: Multilateration - gradient of the cost function

All the experiments and simulations within this thesis are built on top of the weighted non-linear least squares variant described in [18]. The example of the cost function and its gradient for four UAVs are shown as examples in Figures 4 and 5. It is assumed that the anchor location is error free, which is not true in general. The UAVs' location, used for verification of the proposed method, is provided by the Real Time Kinematics (RTK) module and the stated error is ± 15 mm when the RTK is in "RTK FIX" state [19]. This is approximately 100 times more precise compared to distance measuring error, therefore the error can be neglected.

2.3 Robot operating system (ROS)

> *The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.* [2]

ROS has its own filesystem management with a defined structure for ROS package. Every package has to contain basic information and a list of dependencies. The packages can be built by low-level build tool *catkin*. ROS widely supports C++ and Python development. The main purpose of ROS is to enable communication between various modules and to set a standard, which all the robotic system can easily adapt. ROS is also suited for multi-robot systems since it allows connecting to remote master and to share data with no need to implement the communication side.

The fundament of everything is the ROS Master. It allows registering nodes and set up communication between nodes through topics or services. A topic represents a typical producer-consumer scheme, where a node can publish messages with a predefined type, but with no assurance of message delivery and with zero knowledge of registered listeners. Any node is able to register a listener and receive the messages. A service represents a request-response pattern. One node starts a service server and another one opens a service client. The client node sends a message of predefined type to the server and expects a returned message [20].

2.4 Gazebo simulator

Gazebo is a robot simulation tool that enables fast testing of algorithms and provides a robust physics engine, high-quality graphics, and convenient programmatic and graphical interfaces. It is compatible with ROS, making the testing easier and open for any robotic system. The MRS group at FEE CTU provides a simulation environment with UAV visualization, which is almost identical with real experiment environment [21].

[2] Definition from official site http://www.ros.org/about-ros/

3 Location filtering

The output from multilateration algorithm discussed in section 2.2 contains noise since the input of the multilateration is noisy measurements of RSSI (transformed to distance). There are two approaches how to handle this. The RSSI values could be filtered by 1D Kalman filter (KF) discussed in [18] or a 3D position KF, which takes the UAV dynamics into account. The former option is easier to develop and the results are comparable to position KF (experiment results in section 6). On the other hand, the main disadvantage is the time delay, which is caused by the heavy RSSI filtering. The latter option takes the output from multilateration computed from unfiltered RSSI (distances) and predicts the position based on UAV dynamics. This section deals with the position filtering and suggests a basic Kalman filter solution. Table 2 describes symbols used in context with KF.

Symbol	Description
k	$k \in 0, 1, ..., n$
$\mathbf{x_k}$	system state vector at time k
$\mathbf{\hat{x}_k}$	system state prediction vector at time k
$\mathbf{z_k}$	observation vector at time k
$\mathbf{F}$	system transition matrix
$\mathbf{H}$	output transition matrix
$\mathbf{P_k}$	covariance matrix at time k
$\mathbf{\hat{P}_k}$	covariance matrix prediction at time k
$\mathbf{w_k}$	process noise vector at time k
$\mathbf{v_k}$	measurement noise vector at time k
$\mathbf{Q}$	process covariance matrix
$\mathbf{R}$	measurement covariance matrix
$\mathbf{K_k}$	Kalman gain matrix at time k
$\boldsymbol{\nu}_\mathbf{k}$	innovation vector at time k
$\mathbf{S_k}$	innovation covariance matrix at time k

Table 2: Symbols used for Kalman filter description.

3.1 Discrete Kalman Filter

Kalman filter is a linear optimal estimator that minimizes the mean square error of the estimated states if it is provided with an accurate system model and all the noise is Gaussian. The optimization process is described in Algorithm 1.

```
Algorithm 1 Kalman filter process
Initialization
  1: Called only in the first iteration. Initialize the filter state with first measured value, skip the other phases.
Predict
  1: Predict next state based on the model
        $\mathbf{\hat{x}_{k+1}} \leftarrow \mathbf{F x_k} \qquad (3)$
  2: Predict estimate covariance
        $\mathbf{\hat{P}_{k+1}} \leftarrow \mathbf{F P_k F}^T + \mathbf{Q} \qquad (4)$
Update
  1: Update innovation vector
        $\boldsymbol{\nu}_{\boldsymbol{k}\mathbf{+1}} \leftarrow \mathbf{z}_k - \mathbf{H\hat{x}_{k+1}} \qquad (5)$
  2: Update innovation covariance matrix
        $\mathbf{S_{k+1}} \leftarrow \mathbf{H\hat{P}_{k+1}H}^T + \mathbf{R} \qquad (6)$
  3: Compute Kalman gain
        $\mathbf{K_{k+1}} \leftarrow \mathbf{\hat{P}_{k+1}H}^T\mathbf{S_{k+1}}^{-}1 \qquad (7)$
  4: Set new state
        $\mathbf{x_{k+1}} \leftarrow \mathbf{\hat{x}_{k+1}} + \mathbf{K_{k+1}}\boldsymbol{\nu}_{\boldsymbol{k}\mathbf{+1}} \qquad (8)$
  5: Update estimate covariance
        $\mathbf{P_{k+1}} \leftarrow (\mathbf{I} - \mathbf{K_{k+1}H_{k+1}})\mathbf{\hat{P}_{k+1}} \qquad (9)$
```

State space representation of the system with no input is generally described as

$$\begin{aligned} \mathbf{x_{k+1}} &= \mathbf{F}\,\mathbf{x_k} + \mathbf{w_k} \\ \mathbf{z_k} &= \mathbf{H}\,\mathbf{x_k} + \mathbf{v_k}, \end{aligned} \tag{10}$$

where $\mathbf{F}$ is the state transition matrix, $\mathbf{H}$ is the output transition matrix, $\mathbf{x_{k+1}}$ is the system state in the next step $k+1$, and $\mathbf{x_k}$ is the system state in actual step. Furthermore, $\mathbf{w_k}$ is the process noise, $\mathbf{v_k}$ is the measurement noise, and $\mathbf{z_k}$ is the measurement made in actual step. The system is linear and time-invariant (LTI).

3.1.1 System model

The model of the system is derived from point-mass dynamics in three-dimensional space. The model omits the gravity acceleration in the z direction because the UAV movement is controlled by the onboard regulators and does not fall freely to the ground. Therefore, the equations are stated as

$$\begin{aligned} x_{k+1} &= x_k + v_{x_k}\Delta t, \\ v_{x_{k+1}} &= v_{x_k}, \\ y_{k+1} &= y_k + v_{y_k}\Delta t, \\ v_{y_{k+1}} &= v_{y_k}, \\ z_{k+1} &= z_k + v_{z_k}\Delta t, \\ v_{z_{k+1}} &= v_{z_k}, \end{aligned} \tag{11}$$

where x, y and z are the cartesian coordinates, v_x, v_y and v_z are the velocities, and Δt is constant time step. The state variables are selected as

$$\mathbf{x} = \begin{bmatrix} x & v_x & y & v_y & z & v_z \end{bmatrix}^T. \tag{12}$$

Now, the state space model can be completed. The state transition matrix $\mathbf{F}$ and output transition matrix $\mathbf{H}$ are derived in [22] as

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \tag{13}$$

where the matrix $\mathbf{H}$ is selected to output state variables x, y and z.

The variables that are missing for Kalman filter are the matrices $\mathbf{Q}$ and $\mathbf{R}$. The matrices for constant velocity particle have been derived in [23]. The matrices are then defined as

$$\mathbf{Q} = q \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 & 0 & 0 & 0 \\ \frac{\Delta t^2}{2} & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ 0 & 0 & 0 & 0 & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix}, \quad \mathbf{R} = r \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{14}$$

where q and r are parameters that can be used for tuning. The pair of the parameters q and r was empirically tuned on three trajectories. The first was linear translation, the second took a shape of prabola, and the third was circular. The aim was to have smoother trajectories as well as fast enough filter to handle the turning segments of trajectories. The best performing parameters were found as $q = 0.03$ and $r = 3$, because the filtered trajectory was not drifting during the turning segment in parabola trajectory while still smoothing the line trajectory. Time step Δt was set to 1 s, which corresponds to real experiments (more in 6). The trajectories were generated for UAV moving at $1\,\mathrm{m\,s^{-1}}$. Afterwards, the Gaussian noise was added to the trajectories to simulate a multilateration output using C++ normal distribution from *std* library with mean set to 0 m and standard deviation set to 1 m. The results of the test trajectories are shown in Figures 6,7 and 8.

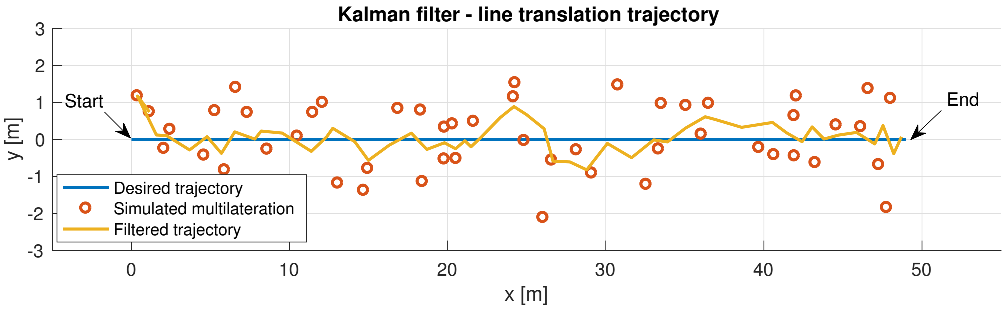
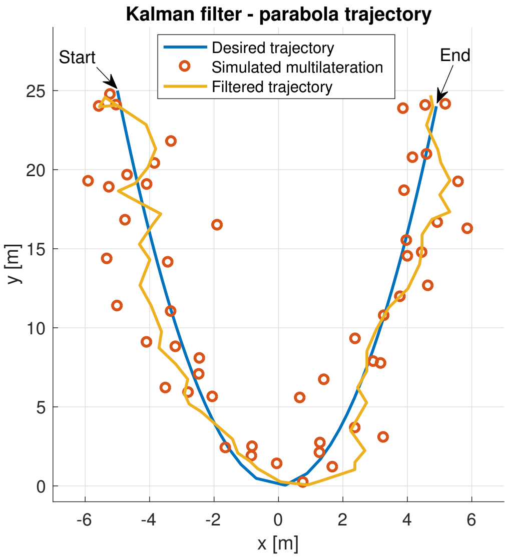


Figure 6: The multilateration ouput simulated for a UAV moving at speed of $1\,\mathrm{m\,s^{-1}}$ and following a line trajectory.



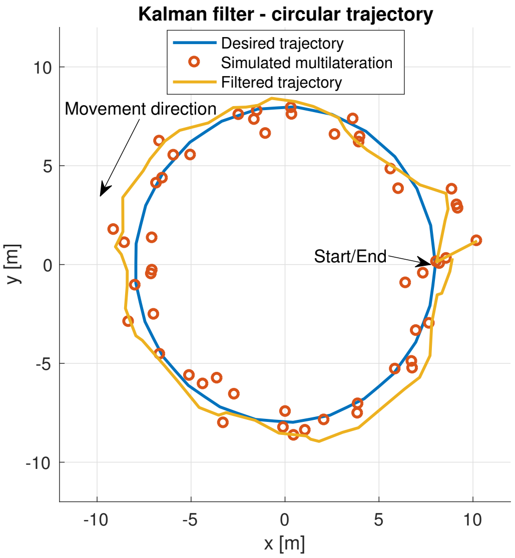


Figure 7: The multilateration ouput simulated for a UAV moving at speed of $1\,\mathrm{m\,s^{-1}}$ and following a prabola trajectory.

Figure 8: The multilateration ouput simulated for a UAV moving at speed of $1\,\mathrm{m\,s^{-1}}$ and following a circular trajectory.

4 Motion planning for formations

The aim of this thesis is to develop two different planning approaches for object localization problem. In the best case scenario, the formation should be able to minimize the localization error over time. This section shows the implementation of two different formations with different planning methods. The first method's idea is to design an immediately reacting planning function, while the other one would take more measurements into account a proceed based on the standard deviation of the measurements. Both of the methods are implemented in C++ and used by the planning modules.

4.1 Fast-reacting formation

The formation in this method takes the shape of a circle with one UAV in the center, which flies higher than the others. The center of the circle is set to be the estimated position of the target. The circle is designed for any number of drones, but for example, with four drones, the shape would be a triangle. This method reacts immediately to the new position estimation event and sends the formation to the estimated position of the target. The algorithm is written in Algorithm 2.

The function *plan_positions* described in this algorithm computes the positions to which should the UAVs fly. The input parameters are the list of UAVs, the UAV that will be on top of the formation, and the estimated position of the target. Along with those, two constants need to be set. The MIN_HEIGHT parameter offsets the formation by x meters above the estimated z coordinate of the target. This constant is mainly, but not necessarily, set for simulation and experiment purposes, where the formation is moving above the target. Therefore, no collisions should happen even when the formation would be delayed behind the moving target. The FORM_RADIUS constant sets the radius of the circle shape.

First of all, the height of the top UAV is set as the z coordinate of the target plus both of the constants. The next step is to find a UAV with the lowest Euclidean distance from the target and put a line that goes through that UAV and the center of the circle. The point at which the line and circle intersect is called the *base_point*. The angle of the base point is computed as $atan2$, which returns the angle of the point on the circle.

In the first *for loop*, vectors from the position of the target to the position of the UAVs are created. After that, angles between these vectors and the vector from the position of the target to the *base_point* are computed. The angles lie within the range of $< 0, \pi >$ and to get the angles to the range of $< 0, 2\pi >$, the positions of the UAVs are inserted into the equation of the line formed by *base_point* and the position of the target. If the value is greater than 0, the angle is shifted as $uav.angle \leftarrow (2\pi - uav.angle)$. The last step is to sort the UAVs by the computed angle and start assigning them the positions. The UAVs are equally distributed on the circle, starting from the base angle.

Algorithm 2 Planning of the positions in the formation

```
function PLAN_POSITIONS(UAVS, TOP_UAV, TARGET_POS)
    top_uav.z ← target_pos.z + MIN_HEIGHT + FORM_RADIUS          ▷ constants
    nearest ← find_nearest(target_pos, uavs)
    base_point ← line_intersect_circle(nearest, target_pos, FORM_RADIUS)
    base_angle ← atan2(target_pos, base_point)          ▷ normed to $< 0, 2\pi >$
    for uav in uavs do
        vec_uav ← Vector(target_pos, uav.position)
        vec_base ← Vector(target_pos, base_point)
        uav.angle ← angle_between_vectors(vec_uav, vec_base)
    end for
    uavs ← sort(uavs.angle)
    angle_step ← $2\pi$/uavs.size
    i ← 0
    for uav in uavs do
        angle_pos ← base_angle + i * angle_step
        uav.position ← point_on_circle(angle_pos, target_pos, FORM_RADIUS)
        i++
    end for
end function
```

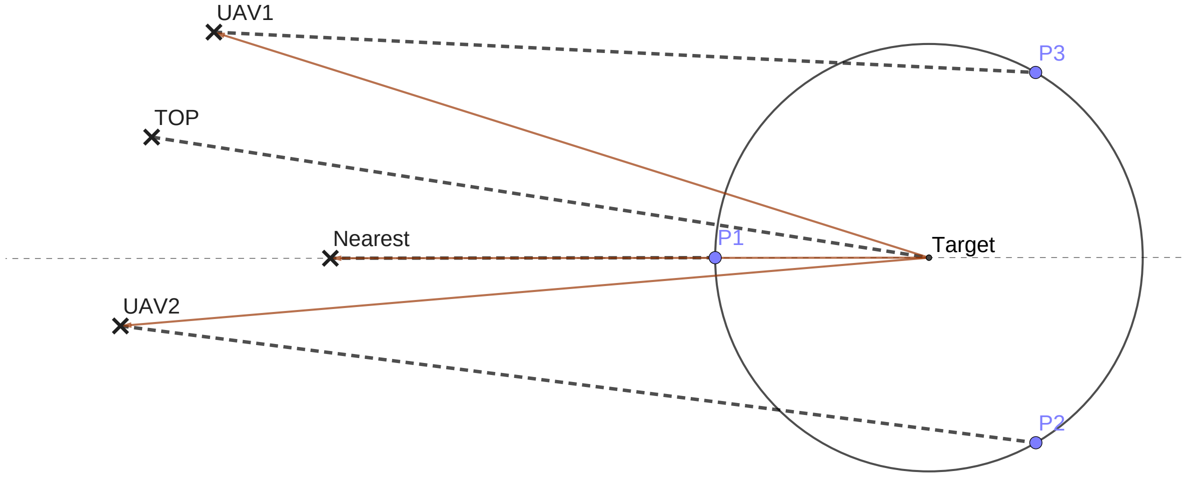


Figure 9: This figure shows an example of position planning algorithm for 4 UAVs, visualized from the top. The line from the target to the nearest UAV intersects with the circle and sets the first point P1. The thin full lines represent vectors aiming from the position of the target. The UAVs are sorted by the angles between their vectors and the Target-Nearest vector. Afterwards, the UAVs are equally placed on the circle. In this example, the UAVs are placed with a step of $2\pi/3$.

4.2 Stepping formation

This formation was primarily designed to track a static (slowly moving) target. The method described in the previous section reacts immediately to a new estimated position of the target. This method, on the other hand, waits for more estimations of the target position and then moves the formation towards the estimated position. This method is designed for 4 UAVs, where the shape of the formation is asymmetrical, but it is based on the shape of a square.

The algorithm starts by selecting a center point. The method initially sets the shape of the formation as the square, where the z axis of each UAV is set as a parameter. After the initialization, the algorithm waits for n estimations of the target position. When it is done, mean position is computed along with errors defined as Euclidean distance from the mean of the estimated positions. Furthermore, the standard deviation (std) of the errors is computed.

In the next step of the method, a vector from the center to the position of the target (the mean value) is obtained and also the distance between those points is computed. If the distance together with std are lower than the desired precision, the algorithm has localized the target (it is in proximity of the target) and waits for its further movement. If the value was higher, then the center of the formation is shifted in the direction to the target by a half of the estimated distance.

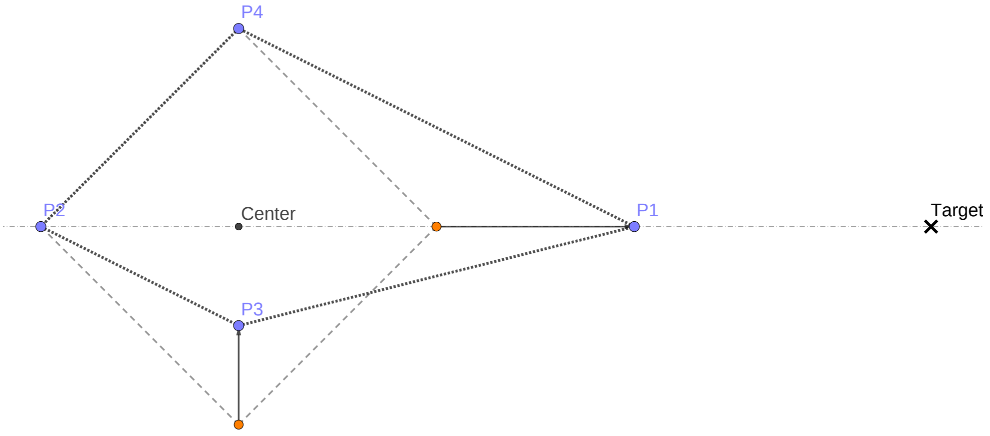
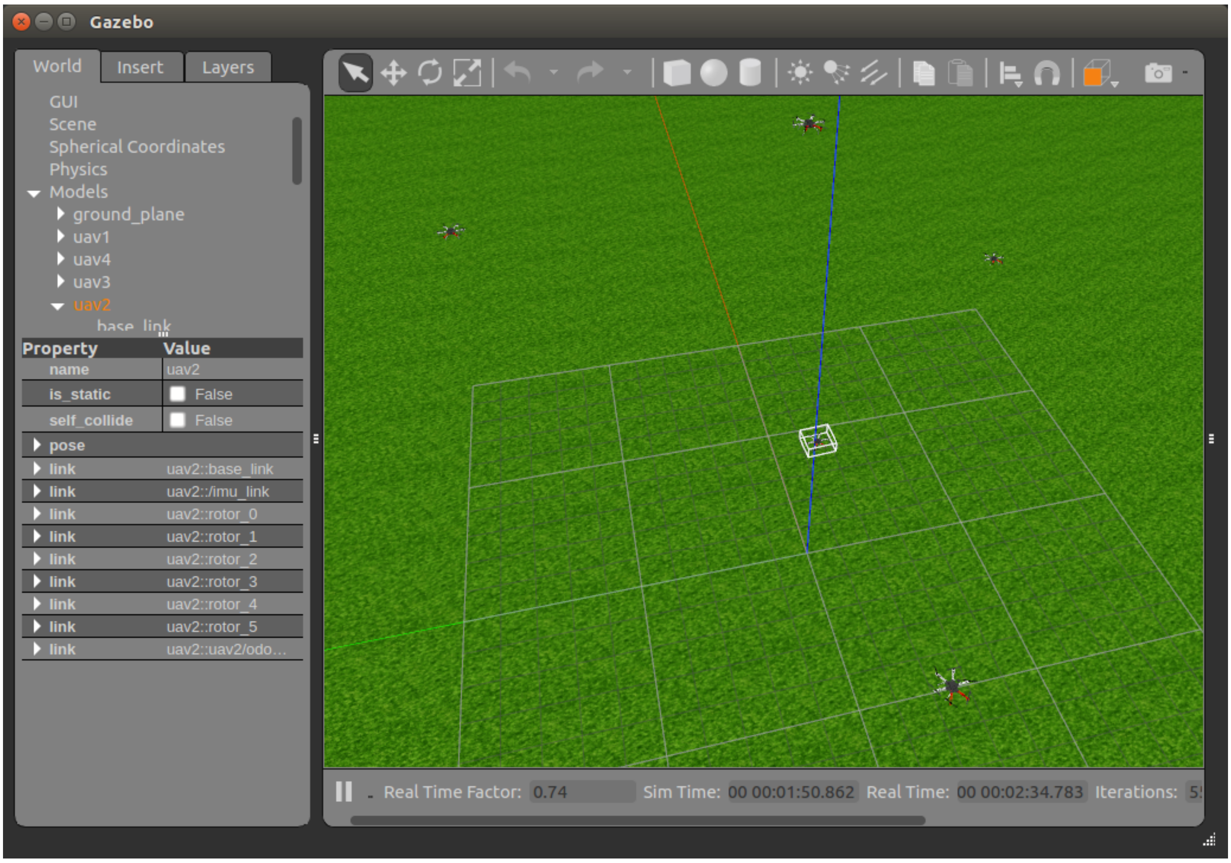


Figure 10: This figure shows the formation positions, viewed from the top. Every step, the planning algorithm waits for n measurements to be obtained. Then, the planner shifts the center to the next step.

The shape of the formation for the next steps starts by computing the positions of the UAVs using a square that is centered at the new planned position and rotated to the target as it is shown in the Figure 10. Afterwards, the UAVs that are assigned to the corners of the square marked by orange dots in the figure are shifted. One of these UAVs is translated towards the target (position P1) and the remaining one is shifted a half way to the center (position P3). This is more explained in section 5.3. After this, the target is measured again and the process is repeated. During the flight, the algorithm waits and does not recompute the trajectories.

5 Simulations

This section contains the simulation results and examples, and also describes the communication model used both in the simulations and later for the real-world experiment. The simulations were conducted to validate the functionality of the motion planning algorithms. The simulations were implemented using the ROS packages build by the MRS group and visualized in Gazebo. The example of the environment and the formation is shown in the Figure 11.



Property	Value
name	uav2
is_static	□ False
self_collide	□ False
pose	
link	uav2::base_link
link	uav2::/imu_link
link	uav2::rotor_0
link	uav2::rotor_1
link	uav2::rotor_2
link	uav2::rotor_3
link	uav2::rotor_4
link	uav2::rotor_5
link	uav2::uav2/odo…

Real Time Factor: 0.74 Sim Time: 00 00:01:50.862 Real Time: 00 00:02:34.783 Iterations: 5

Figure 11: The environment used for testing in Gazebo simulator. The figure shows the UAVs forming above the estimated position of the target.

5.1 Communication model

A stable communication is necessary for the object localization and planning of the formation movement, because all the UAVs except for the target are communicating with the Master. Any UAV can be selected as the Master. The Master runs the Multilaterator and Scheduler modules and subscribes to the Odometry (estimated states of the UAV) and BLE Scanner modules. The *slow_odom* topic provides the information about the estimated state of the UAV at 2 Hz frequency. The BLE Scanner module reads the RSSI values from the BLE development board and publishes the converted distance (see section 2.1.1) at approx. 5 Hz. The Multilaterator reads the location of the receivers along with the distances from the target object, periodically (on 1 Hz) estimates the location of the target, and publishes the filtered output on the *multilateration* topic. The Scheduler module is subscribed to that topic and reacts to this message from the Multilaterator module. The trajectories are then planned for all the UAVs in formation and published on the *desired_trajectory* topic. The MPC tracker handles the trajectory input and starts following the desired path. The complete communication model is visualized in Figure 12.

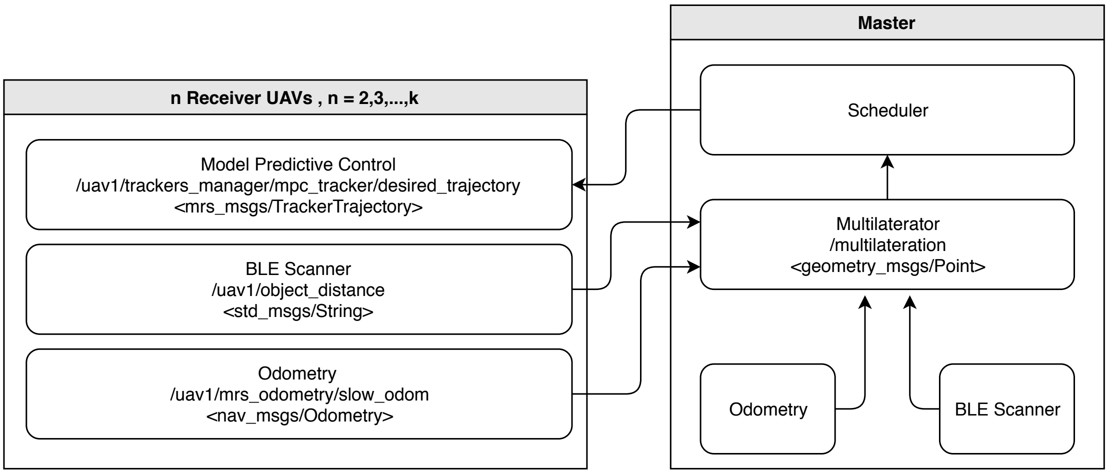
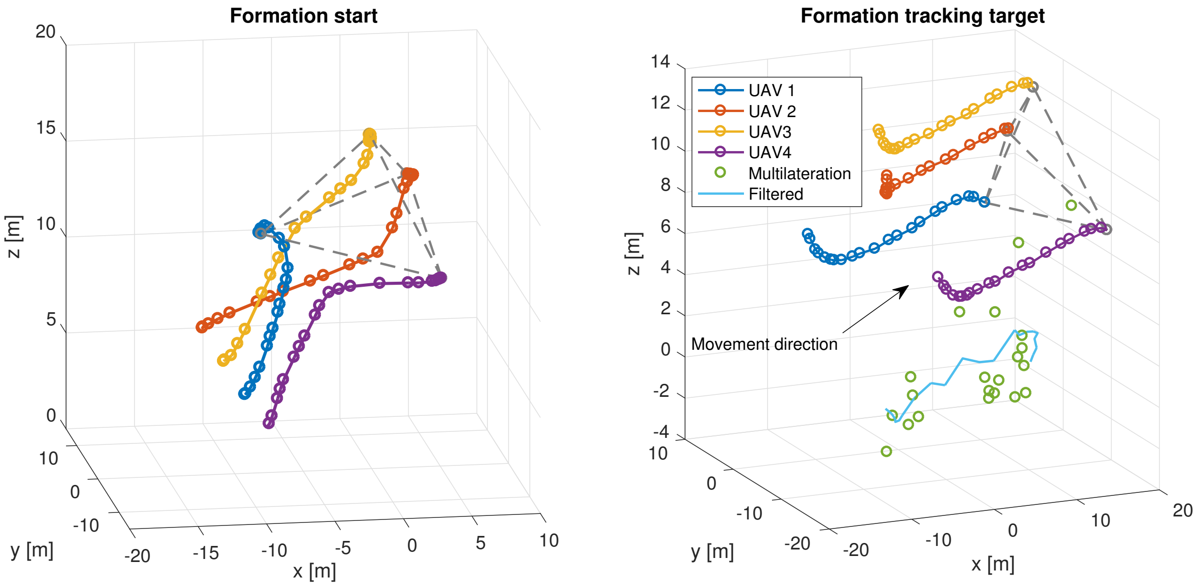


Figure 12: The communication model that is used during the simulations and the experiment. One UAV is selected as a Master. Furthermore, there has to be at least two more receiver UAVs to track the target in 2D. The Master is subscribed to the Odometry and BLE Scanner modules of all the UAVs assigned as receivers. The Multilaterator gathers all the data and periodically sends the filtered output to the Scheduler module. The Scheduler then plans the trajectories for all the UAVs and informs their trajectory tracking managers.

5.2 Fast-reacting formation

In the simulation, one UAV was selected as a target and was randomly flying in given area. The formation of 4 UAVs had to localize the target and keep flying in the formation above its position. This is the same scenario as the real-world experiment. The example of the UAVs getting to the formation is shown in the Figure 13a. The UAVs started forming line 5 m above the ground and then the UAVs formed the shape of the proposed formation.

The Figure 13b shows a few seconds of the target localization simulation. The multilateration input was simulated as a distance from the position of the UAV to the position of the ground-truth plus Gaussian noise with mean 0 and std 3 m. The formation was able to track the target and follow its position for several minutes of simulation. On average, it took the formation approx. 20 s to track a static target.



(a) The figure shows the UAVs moving towards their planned position to hold a formation shape, and what trajectories did the UAVs took. The UAVs started forming a line. Afterwards, the planning algorithm computed trajectories to reach the formation positions.

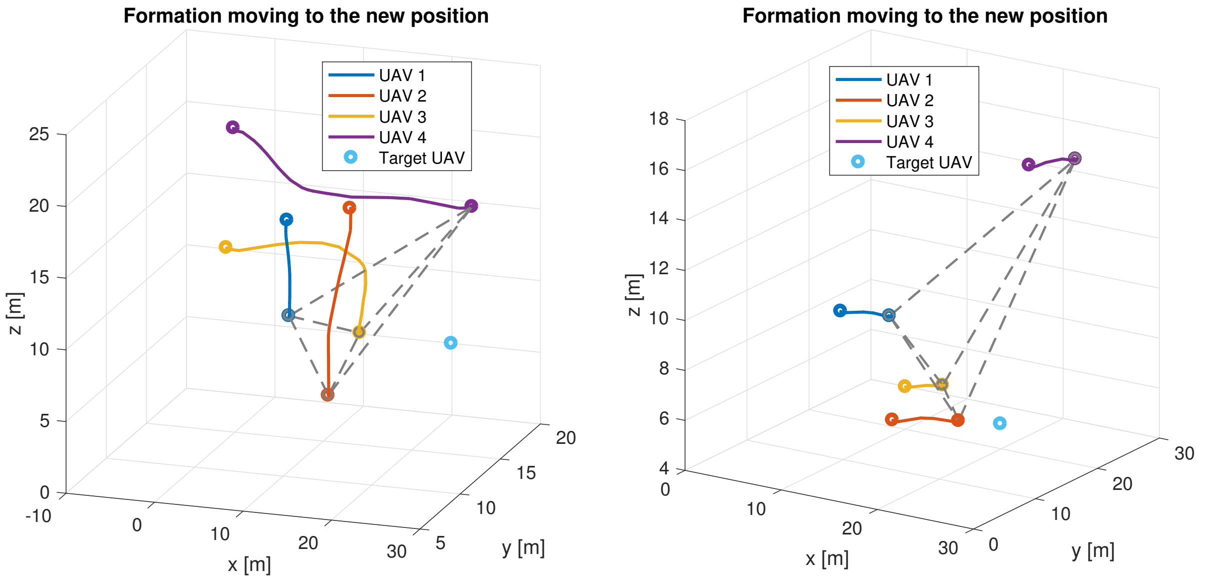
(b) Visualization of the target localization simulation. The formation is following the filtered multilateration output. The target was randomly flying and formation of UAVs was estimating the target's position.

Figure 13: The figures show how the formation is constructed and how the formations moves when tracking a target. Once the formation is formed, the shape stays the same.

5.3 Stepping formation

The simulations for this method were designed to determine in how many steps is the formation able to localize the static target, because in every next step, the distance between the formation and the target should be half of the value. The measure count (n constant) constant was to 15. At first, the formation took the shape of a rotated square, but in most of the simulations, the formation could not successfully localize the target, because when the formation got closer, the estimated position of the target started converging to other minimums, and the formation, even though it waited for more measurements, started drifting away from the target. After many tries, a better solution was to make the shape asymmetric with more different distances from the target in each axis.

The Figure 14 visualizes the formation transitions between single steps. The right figure shows the formation as it takes the last step to be within the localization tolerance and then stopping the algorithm. The whole localization process took around 90 s. Therefore, this method was slower than the first proposed method.



(a) The figure shows the formation moving from the previous positions to the new positions. In the previous step, the estimated position of the target was in higher height, therefore the UAVs are descending in the z axis.

(b) In this figure, the formation continues one more step from the last one shown in the left figure. The formation center moves close enough to the estimated position of the target; thus, the planning algorithm does not set new positions and waits for the target to change its position.

Figure 14: The figures show how the formation moves between the steps when tracking a static target. The target was localized in 6 planning steps.

6 Real-world experiment

This experiment took place in the outdoor environment on the open field providing a similar environment as in the simulations. The experiment was designed to verify the localization technique, test the multi-robot formation planning, verify the proposed Kalman Filter setup and to analyze localization errors for both steady and moving target. The experiment was prepared for five UAVs, where four UAVs were selected as receivers (anchors) and the last one as a transmitter (target). Unfortunately, one of the UAVs were not able to fly due to the technical problems with the antenna used for remote control. Therefore, we decided (for safety reasons) to move on the field with the disarmed UAV in hands to have the ground truth position provided by the RTK module. Since we were only able to move on the ground forming the xy area, the whole experiment will be visualized in two dimensions (x and y). The formation was following the location of the target estimated by the RTK and not the estimated position from the proposed technique. This was due to that the aim of this experiment was to measure localization error on a long trajectory without drifts and longer time delay and also to validate the formation design.

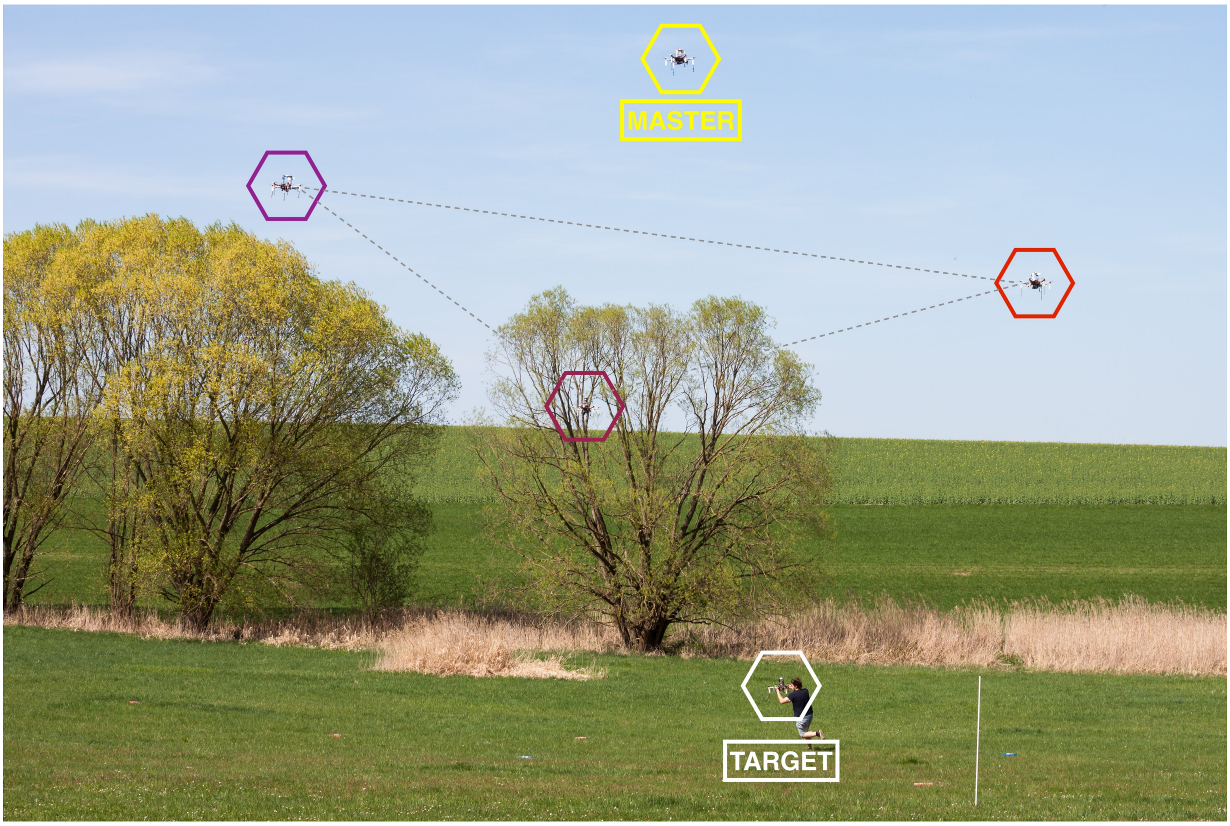


Figure 15: Photo from the real-world experiment. The target UAV was carried in hands, creating xy trajectories measured by the RTK module. The formation is tracking the estimated location from RTK to test the precision and time delay of the localization.

The UAVs used in this experiment were built by the MRS group at CTU in Prague. The UAVs are equipped with powerful computers, multiple sensors and Wi-fi communication used for data transfer among the UAVs [19]. The BLE boards were attached to to the UAVs as shown in the Figure 1.

The experiment timeframe is divided into five parts $P1$ - $P5$. The whole timeframe is shown in Figures 16 and 17. The first figure shows the localization of the whole experiment and provides an overview of the localization data. The second figure shows the velocities and provides a deeper view on the target and formation movement. A video attachment to this experiment is available at `http://mrs.felk.cvut.cz/suster2018thesis`.

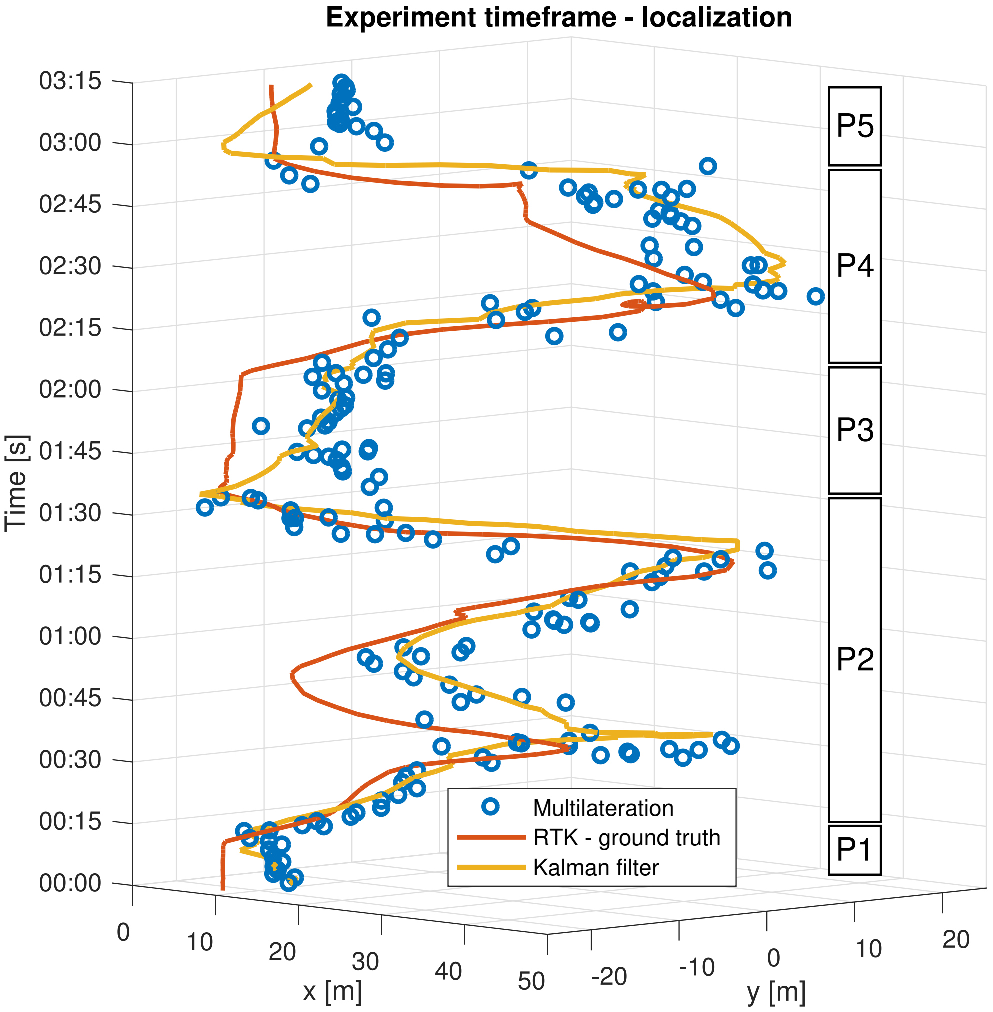


Figure 16: Experiment timeframe - localization. The chart presents the whole experiment and displays trajectory of ground-truth provided by RTK, raw multilateration output and filtered trajectory. Video available at `http://mrs.felk.cvut.cz/suster2018thesis`.

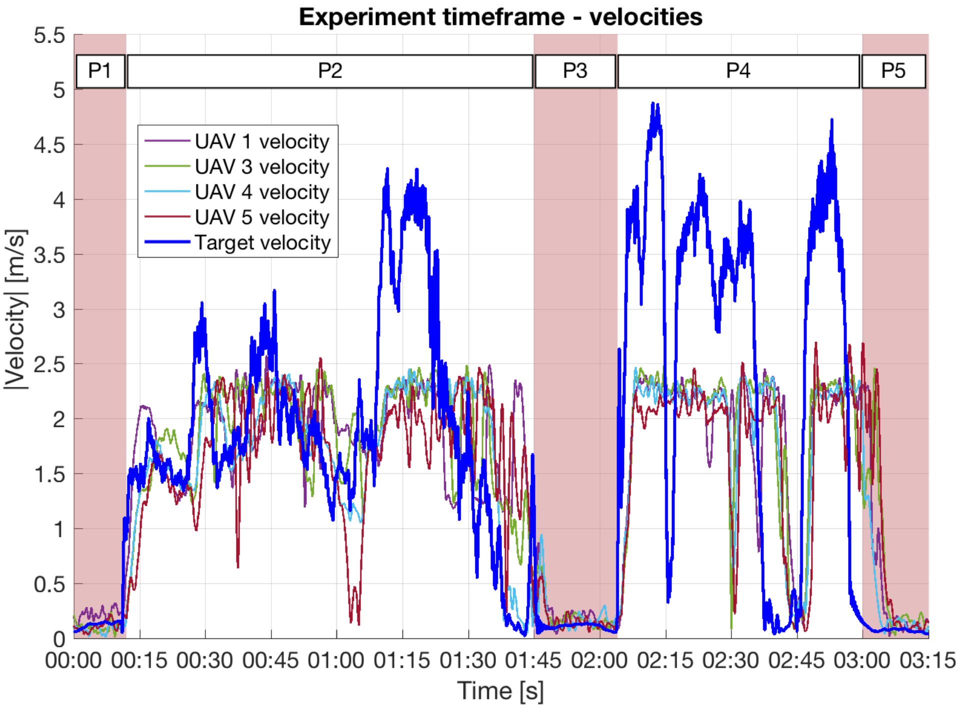
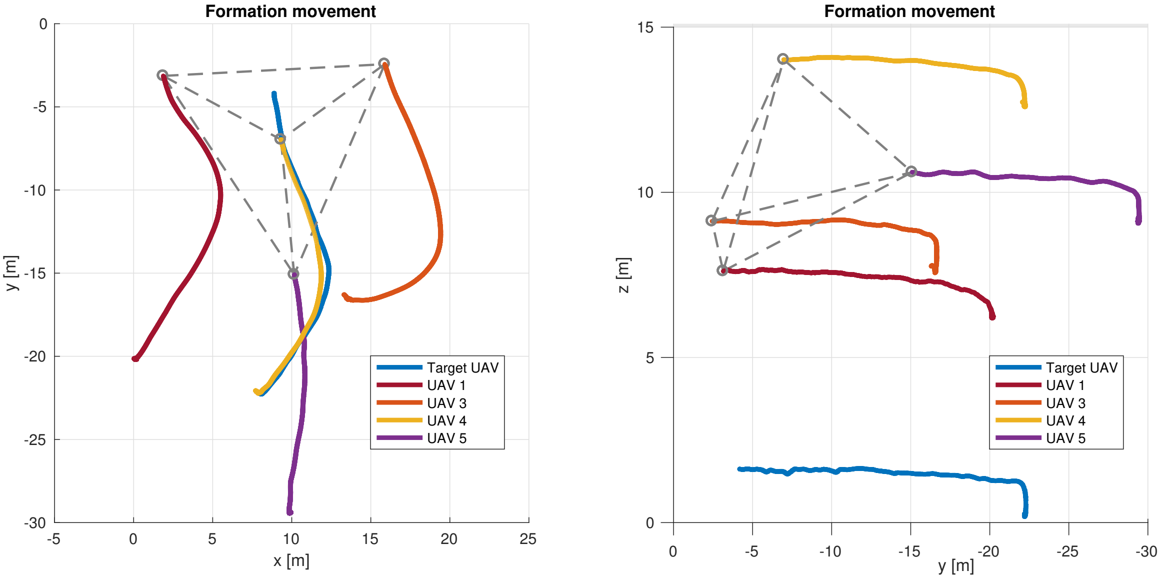


Figure 17: Experiment timeframe - velocities. The timeframe chart shows that in the parts $P1$, $P3$, and $P5$ the target and the formation were not moving. The parts $P2$ and $P4$ represent the moving parts where both the target and the formation were moving. The formation speed limit was set to $3\,\mathrm{m\,s^{-1}}$. The target was moving faster in some parts, thus causing the formation to be delayed behind the target.

The parts of this experiment, denoted in the figures as $P1$, $P3$, and $P5$ present the static target localization and the parts $P2$ and $P4$ show the moving target localization. The output from multilateration is more stable during the static target parts, but as can be seen in Figure 16, the position is not correct. That happened due to the noisy RSSI measurements. The BLE technology is good for its low energy consumptions, but the RSSI values are unstable. Hence the huge localization error.



(a) Top view

(b) Side view

Figure 18: The formation was stable during the experiment except for one moment, when the Wi-fi signal was low and the Scheduler module could not publish trajectories to the UAVs. The figures show the top and side view of the formation movement during the part P2.

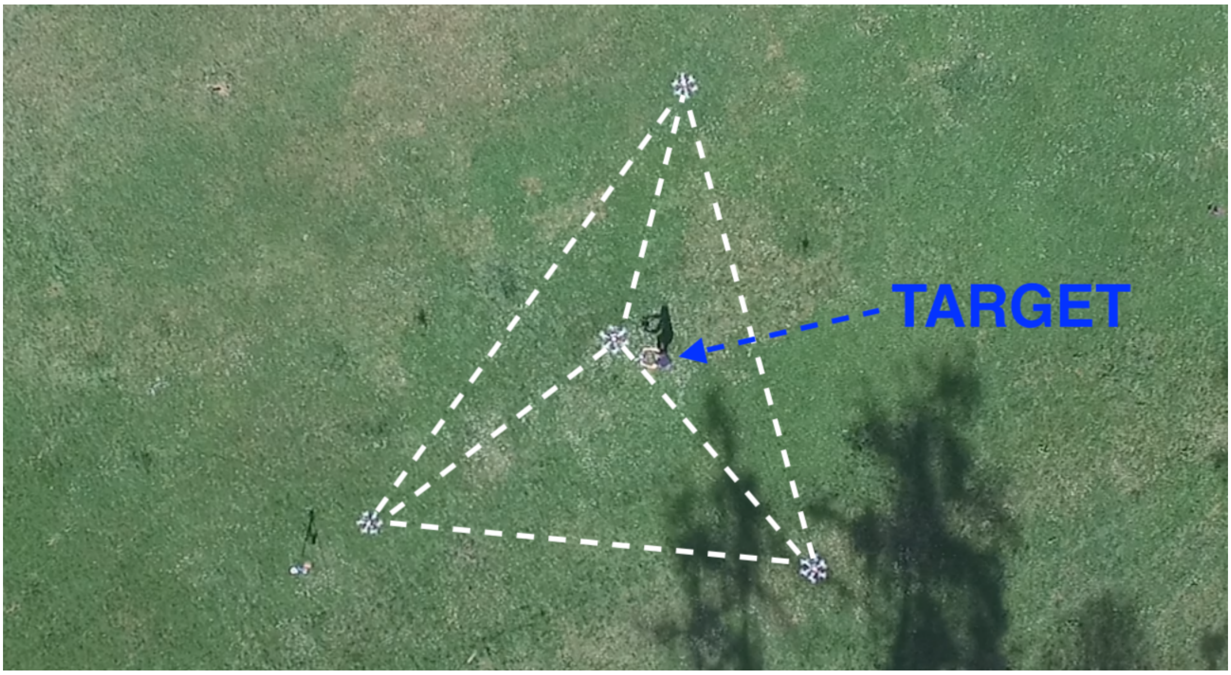
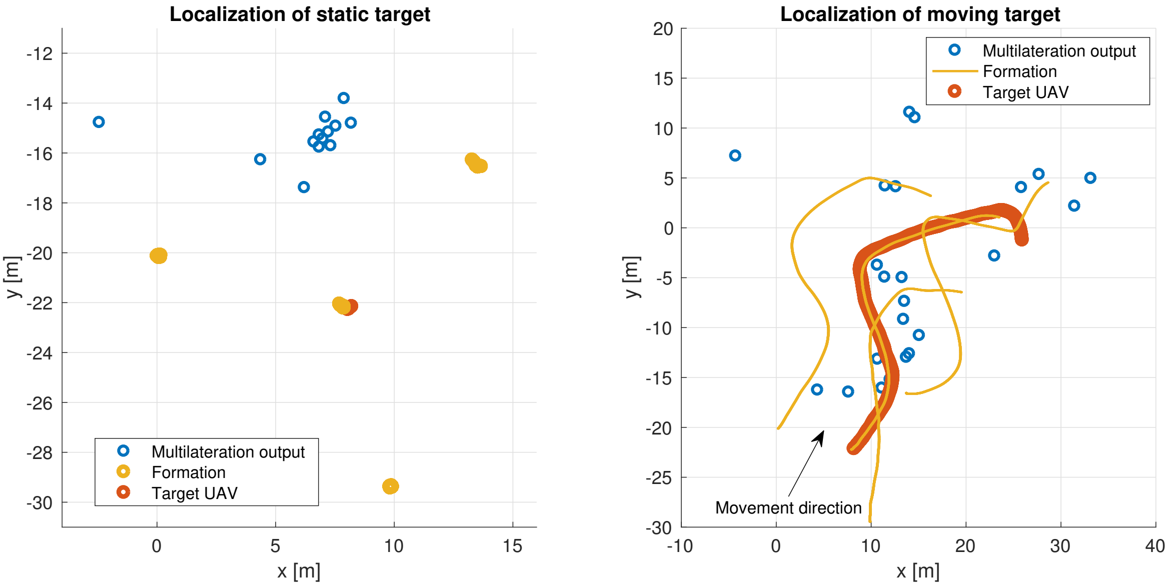


Figure 19: Photo taken by the DJI drone, showing the formation from the top.

6.1 Localization accuracy

Two approaches to the error comparison will be presented. The first approach is computing the error as the Euclidean distance between the ground-truth location of the target and the estimated location by the proposed method at the same time step. The second approach is designed to ignore the time delay and to provide an overview of the error between two trajectories. The process is to take a point from the estimated trajectory, and find the nearest point on the ground-truth trajectory (the nearest point will be called *snapped* throughout this section). Afterwards, the error is computed as the Euclidean distance between the snapped point and the selected point from the estimated trajectory. The trajectory is be limited by a time constant T_c to prevent errors during the segments where the ground-truth trajectory crosses itself. Therefore, the point can be snapped only to the trajectory part of the actual step time $\pm T_c$.



(a) Localization of the static target. This chart shows 15 seconds of the part P1. The formation and the target were not moving. During this time, the average error is approx. 7 m with std approx. 1.5 m.

(b) Localization of the moving target. This chart shows 23 seconds of the part P2. The lines represent the movement of the formation and the target. During this time, the average error is approx. 9 m with std approx. 4 m.

Figure 20: The figures show main difference between the localization of static target and the moving target. The blue circles represent the multilateration output. The standard deviation of the localization error is lower during the static position estimation. Nevertheless, the estimated position is shifted away from the ground-truth position.

The means of the errors ($\bar{e}$) and the standard deviations (std) are shown in the Table 3. The errors are computed for every part of the experiment to point out the differences between the static and moving target tracking. For the whole dataset, the Kalman filter reduced the mean error by 5% in the time-dependent comparison. The error was lower than the non-filtered variant, even though the filter increased the time delay. Furthermore, the snapped comparison shows that the filter reduced 20% of the mean error. The biggest difference was during the part P5, where the filter reduced 54% of the mean error.

The errors in this experiment are generally higher than in the other experiments presented in [24],[25], where the accuracy is stated up to 1 m. The difference is that the other experiments use static anchors, which are placed in the same height and that they use only 2D localization. The lowest mean error from this experiment has snapped Kalman filter variant.

Value in [m]	P1	P2	P3	P4	P5	Complete experiment
$\bar{e}$ multilateration	7.41	9.11	8.88	10.79	6.72	9.28
std multilateration	1.76	4.96	1.83	4.12	1.93	4.38
$\bar{e}$ multilateation filter	7.22	9.61	9.02	9.92	7.52	9.35
std multilateration filter	0.60	4.92	1.09	4.07	3.80	4.26
$\bar{e}$ KF	7.29	8.57	7.25	11.67	3.12	8.85
std KF	0.66	6.07	1.20	4.06	1.94	5.30
$\bar{e}$ snapped	6.40	6.65	6.92	8.08	7.32	6.61
std snapped	1.12	4.50	3.20	4.44	3.70	3.97
$\bar{e}$ KF snapped	7.01	4.85	5.22	6.51	2.96	5.36
std KF snapped	0.80	3.37	2.57	3.34	1.99	3.27

Table 3: The table compares the mean errors ($\bar{e}$) and the standard deviations (std) of the errors. The data shows that the Kalman filtered trajectory contained less error than the non-filtered and RSSI filtered trajectories in both snapped and time-dependent comparisons. Moreover, the parts where the target was at static position contained less error than the parts with the moving target.

7 Conclusion

This thesis dealt with the motion planning, formation design, and Kalman filter design in the task of an unknown transmission source localization for a team of unmanned aerial vehicles (UAVs). The concept of a transmission source was built on the Bluetooth Low Energy technology. The localization was based on the principle of multilateration, which required the conversion of RSSI values to distances.

The Kalman filter was designed to smooth out the raw multilateration output, because the motion planning algorithms react to the estimated position. Otherwise, the estimated position is affected by noise from the RSSI measurements.

There were two motion planning approaches presented. The first algorithm reacts immediately on a new estimation of the target position by movement of the formation of UAVs. The formation takes the shape of a circle, where *Master* UAV is above the other members of the formation that are equally distributed on the circle. The second algorithm waits for n estimations of the position of the target and then moves the formation towards this estimated position. In this method, the formation forms an asymmetrical shape, which was designed through a series of simulations. Both of the methods were simulated in Gazebo, where they were able to localize the target. By comparison of behaviour of these methods, the first method localized the moving and static target faster than the second one. On the other hand, once the second mentioned method moved the formation near the static target, the estimated position was more stable.

The proposed system has been also experimentally verified by a real-world deployment. The experiment was conducted on the open field, where the fast-reacting method was tested. The experiment was designed for 5 UAVs, where one of them was designed to be the target. The other UAVs were flying in the formation, and they were localizing the target. The formation was stable throughout the experiment except for one moment when the Wi-fi signal was low and *Master* UAV could not publish new trajectory plans. The mean error and the standard deviations were mostly lower with the use of the Kalman filter. For example, the Kalman filter reduced the mean error by 5% in the time-dependent comparison and 20% in the time-independent comparison. The mean error of the presented localization technique was 8.85 m during the whole experiment. Therefore from these observations, the proposed methods cannot be used for applications, where high accuracy is fundamental. The BLE based source mapping could still be used in applications like target localization using visual recognition, where the proposed methods could estimate the initial location for further mapping by UAVs with cameras. The proposed methods in this thesis are general and thus any other technologies than BLE, where distances from a target are measured, could adopt the same localization and planning techniques.

References

[1] M. Saska, V. Kratky, V. Spurny, and T. Baca. Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control. In *IEEE ETFA*, 2017.

[2] Petr Štibinger. Arfid chips localization by a group of helicopters using a principle of multilateration. Bachelors thesis at CTU, 2017.

[3] M. Saska, V. Vonasek, T. Krajnik, and L. Preucil. Coordination and Navigation of Heterogeneous MAV-UGV Formations Localized by a 'hawk-eye'-like Approach Under a Model Predictive Control Scheme. *International Journal of Robotics Research*, 33(10):1393–1412, 2014.

[4] M. Saska, T. Krajnik, V. Vonasek, Z. Kasl, V. Spurny, and L. Preucil. Fault-Tolerant Formation Driving Mechanism Designed for Heterogeneous MAVs-UGVs Groups. *Journal of Intelligent and Robotic Systems*, 73(1-4):603–622, 2014.

[5] V. Spurny, T. Baca, and M. Saska. Complex manoeuvres of heterogeneous mav-ugv formations using a model predictive control. In *21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2016.

[6] M. Saska, Z. Kasl, and L. Preucil. Motion Planning and Control of Formations of Micro Aerial Vehicles. In *19th World Congress of the International Federation of Automatic Control (IFAC)*. IFAC, 2014.

[7] M. Saska, J. Langr, and L. Preucil. Plume Tracking by a Self-stabilized Group of Micro Aerial Vehicles. In *Modelling and Simulation for Autonomous Systems*, 2014.

[8] M. Saska, J. Vakula, and L. Preucil. Swarms of Micro Aerial Vehicles Stabilized Under a Visual Relative Localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[9] T. Baca, G. Loianno, and M. Saska. Embedded model predictive control of unmanned micro aerial vehicles. In *2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, Miedzyzdroje, Poland, 2016.

[10] M. Saska, V. Spurny, and V. Vonasek. Predictive control and stabilization of nonholonomic formations with integrated spline-path planning. *Robotics and Autonomous Systems*, 75(Part B):379–397, 2016.

[11] 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.

[12] M. Saska, V. Vonásek, J. Chudoba, J. Thomas, G. Loianno, and V. Kumar. Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles. *Journal of Intelligent & Robotic Systems.*, 84(1):469–492, 2016.

[13] M. Saska. MAV-swarms: unmanned aerial vehicles stabilized along a given path using onboard relative localization. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015.

[14] E. Mackensen, M. Lai, and T. M. Wendt. Bluetooth low energy (ble) based wireless sensors. In *2012 IEEE Sensors*, pages 1–4, Oct 2012.

[15] A. Thaljaoui, T. Val, N. Nasri, and D. Brulin. Ble localization using rssi measurements and iringla. In *2015 IEEE International Conference on Industrial Technology (ICIT)*, pages 2178–2183, March 2015.

[16] M. Kolakowski. Kalman filter based localization in hybrid ble-uwb positioning system. In *2017 IEEE International Conference on RFID Technology Application (RFID-TA)*, pages 290–293, Sept 2017.

[17] Nordic Semiconductors. *nRF52832, Multiprotocol Bluetooth low energy (Bluetooth 5), ANT/ANT+ and 2.4GHz proprietary System-on-Chip.*

[18] Michal Němec. Arfid chips localization by a group of helicopters using a principle of multilateration. Bachelors thesis at CTU, 2018.

[19] 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 Robotics and Automation Letters*, 3(3):1576–1583, July 2018.

[20] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[21] Vojtěch Spurný, Tomáš Báča, Martin Saska, Robert Pěnička, Tomáš Krajník, Giuseppe Loianno, Justin Thomas, Dinesh Thakur, and Vijay Kumar. Cooperative autonomous search, grasping and delivering in a treasure hunt scenario by a team of uavs, 2017. (submitted to the special issue on "MBZIRC 2017 - Challenges in Autonomous Field Robotics").

[22] Roger Labbe. Kalman and bayesian filters in python. `https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python`, 2014.

[23] Ian Reid. Estimation ii lecture notes. `http://www.robots.ox.ac.uk/~ian/Teaching/Estimation/LectureNotes2.pdf`, 2001.

[24] 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.

[25] 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.

Appendix A CD Content

In Table 4 are listed names of all root directories on CD.

Directory name	**Description**
thesis	the thesis in pdf format
src	source codes
matlab	matlab source files

Table 4: CD Content

Appendix B List of abbreviations

In Table 5 are listed abbreviations used in this thesis.

Abbreviation	**Meaning**
UAV	unmanned aerial vehicle
KF	Kalman filter
RSSI	received signal strength indicator
UAV	unmanned aerial vehicle
MRS	Multi-robot systems
RTK	real-time kinematics

Table 5: Lists of abbreviations