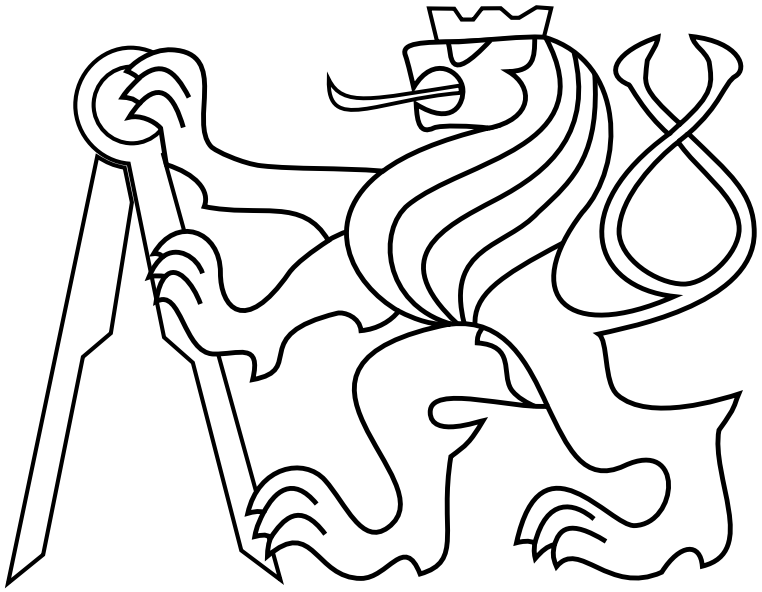
CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

BACHELOR'S THESIS



Jan-Jakub Šenkeřík

Pursuing of a moving helicopter by a compact team of micro aerial vehicles

Department of Cybernetics

Thesis supervisor: **Ing. Viktor Walter**

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..... .....

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Šenkeřík Jan-Jakub** Personal ID number: **457229**

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:

Pursuing of a moving helicopter by a compact team of micro aerial vehicles

Bachelor's thesis title in Czech:

Sledování letící helikoptéry kompaktním týmem malých bezpilotních prostředků

Guidelines:

An algorithm for pursuing a moving helicopter by a team of relatively localized micro aerial vehicles (MAVs) will be implemented and adapted for deployment with the multi-MAV system of MRS group at CTU. Motivation of this work is to enable a robust following of an unauthorized unmanned aerial system (UAS) detected in no-fly zones for its possible elimination (physical elimination of the unauthorized UAS is not a part of the thesis).
The following tasks will be solved:
1. To implement a method for cooperative following a flying object by a group of MAVs and integrate it into the ROS system used for control teams of helicopters at the MRS group at CTU.
2. To focus on aspects of following an object (the UAS) moving in 3D space.
3. To integrate a method of mutual localization of team members using onboard sensors being developed in parallel to this thesis (a technique using UV LEDs and/or convolutional neural network will be used based on availability ? design of object detection and localization algorithms is not part of the thesis)
4. To verify the method in a realistic Gazebo simulator.
5. To prepare the system for experimental verification with the multi-MAV platform of the MRS group [4] (the real experiment will be realized in case of available HW based on decision of the thesis advisor)

Bibliography / sources:

[1] S. Minaeian, J. Liu and Y. J. Son, "Vision-Based Target Detection and Localization via a Team of Cooperative UAV and UGVs," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 46, no. 7, pp. 1005-1016, 2016.
[2] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs and C. Bouman, "Multi-target detection and tracking from a single camera in Unmanned Aerial Vehicles (UAVs)," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, 2016.
[3] Hausman K., Müller J., Hariharan A., Ayanian N., Sukhatme G.S. Cooperative Control for Target Tracking with Onboard Sensing. In: Hsieh M., Khatib O., Kumar V. (eds) Experimental Robotics. Springer Tracts in Advanced Robotics, vol 109: 879-892, 2016
[4] 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.
[5] Robin, C. & Lacroix, S. Auton Robot, 40(4): 729-760, 2016.

Name and workplace of bachelor's thesis supervisor:

Ing. Viktor Walter, Multi-robot Systems, FEE

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

Date of bachelor's thesis assignment: **14.02.2018** Deadline for bachelor thesis submission: **25.05.2018**

Assignment valid until: **30.09.2019**

Ing. Viktor Walter Supervisor's signature	prof. Ing. Michael Šebek, DrSc. Head of department's signature	prof. Ing. Pavel Ripka, CSc. Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

Acknowledgements

Firstly, I would like to thank my parents and my girlfriend for all their loving support during my studies, and secondly, I would like to express my gratitude to Ing. Viktor Walter who dedicated a lot of time to help me with this thesis.

Abstract

The goal of this thesis is to implement an algorithm for pursuing a moving helicopter by a team of unmanned aerial vehicles localized relative to each other and the target. This algorithm aims to control a formation of unmanned aerial vehicles using only data obtained from cameras mounted on each member. The algorithm is implemented and adapted for deployment with the multi-MAV system of MRS group at CTU. The algorithm is verified in a realistic Gazebo simulator and then partially tested in a real-world experiment.

Abstrakt

Cílem této bakalářské práce je implementovat algoritmus, jehož úkolem je pronáslednování pohybující se helikoptéry formací relativně lokalizovaných bezpilotních helikoptér. Tento algoritmus má za úkol řídit formaci bezpilotních helikoptér pouze pomocí dat obržených z kamer, které jsou připevněny na každé bezpilotní helikoptéře. Algoritmus je implementován a přizpůsoben použití v rámci skupiny MRS na ČVUT. Algoritmus je následně otestován v realistickém simulátoru Gazebo a částečně vyzkoušen ve skutečném experimentu.

Contents

List of Figures

1 Introduction

An unmanned aerial vehicle (UAV) or unmanned micro aerial vehicle (MAV) is arguably one of the most discussed topics in the field of autonomous system control. There is currently a great demand for this technology in a considerable number of application, such as surveillance ([1] and [2]), transportation of objects [3], military applications [4], rescue missions [5], cooperative swarm measurement [6] or even applications like documentation of historic buildings [7].

1.1 Motivation and goals

Let us imagine a situation when an unauthorized MAV is detected in a no-fly zone. Thit can be highly dangerous in certain locations. Let us assume, for example, a civilian flying with a toy helicopter in the area of an airport runway. This happens relatively often as testified [8] by Earl Lawrence, the director of Federal Aviation Administration. According to director Lawrence, there have been around 1800 reports of sighting of MAVs in 2016, so far without any collisions.

In a situation like this, there has to be a system that can easily find, follow and eliminate the threat. The aim of this thesis is to implement an algorithm, that could be an essential part of said system. The main purpose of the algorithm is to pursue a moving object by a formation of micro aerial vehicles (MAVs). An example of a formation of several MAVs following the target can be seen in the Fig. 1.

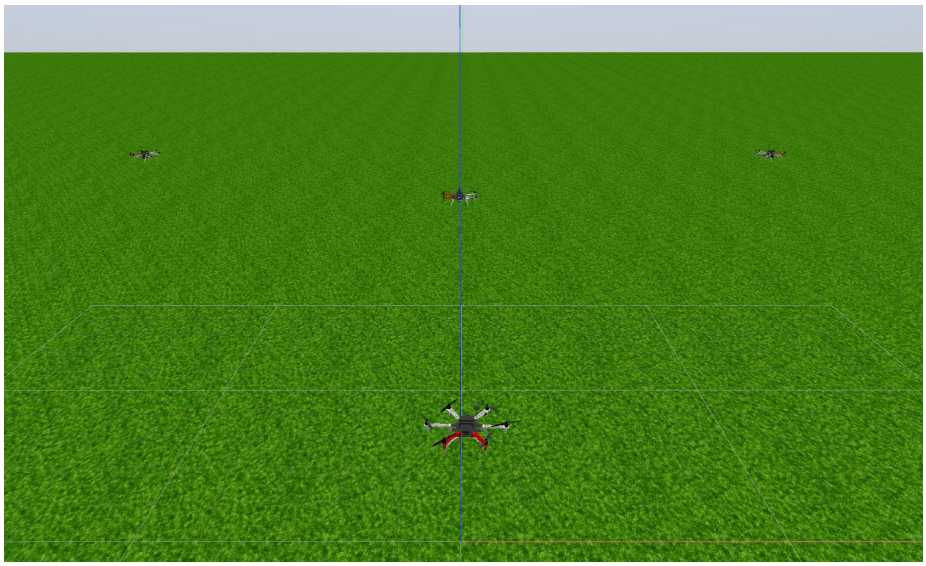


Figure 1: Several MAVs in a compact formation following a target

The algorithm used in this project was described in the article [9] with the goal of having a compact formation of several MAVs localized relative to each other and the target, that will follow the target. When combined with reliable detecting system and a way to eliminate the target, this method could be an ideal solution for dealing with unauthorized MAVs.

1.2 Main tasks and criteria

As mentioned in the previous section, the main goal is to implement a target following algorithm and to create a robust following system.

The algorithm described in [9] is to be integrated into the Robot Operating System (ROS)-based system used by the helicopters of the MRS group at the CTU and altered for following in 3D space. For mutual localization, a method developed in parallel by other members of MRS group, in this case, a convolutional neural network, will be utilized and integrated. The correctness of the system will be verified in the Gazebo simulator and prepared for an experimental verification, if the needed hardware is available.

The objective of the algorithm is to maintain a circular formation containing n MAVs with the target in their middle. Every MAV can only use the data obtained from the previously mentioned neural network and from its neighboring MAVs, thus it can share its own data only with its neighbors. The formation of MAVs is to follow the target through 3D space without obstacles. The target can be any object, but for the purposes of this thesis and its experiments, the target was always another MAV.

1.3 Related work

This subsection is dedicated to mentioning papers that are presenting approaches to the vision-based following or tracking, similar to the one used in this thesis. Some of these papers can be a source material for further improvements of the algorithm described in this thesis, in the future.

1.3.1 MAV in GPS-denied environment

The condition described in the paper [10] are very similar to the conditions in the experiments conducted in this thesis. The ideal use of the method described by F. Poiesi is to follow a target using only data obtained from the camera view to stabilize the formation and to follow the target. The paper [10] proposes a way of a self-stabilization of the multi-MAV groups withou an external positioning system. The proposed approach to this scenario is a vision-based stabilization.

1.3.2 Target tracking with onboard sensing

In the paper [11] an approach to the cooperative control of a team of robots is considered. The main assumption is that the robot positions are unknown and therefore it is important to estimate their positions and the position of a moving target using only the onboard sensors. This scenario is very similar to the one in this thesis, but instead of the camera view, it uses the onboard sensors.

1.3.3 Single camera tracking

The paper [12] presents an approach to detect and track an unmanned aerial vehicle from a single camera. System like this would be a great addition to the method described in this thesis. The method described by F. Poiesi does assume a ground target and use of this method creates some challenges when used for an aerial target. These challenges are described in the section 4.

2 Description of the method

This thesis uses the article by F. Poiesi as a base for the development of the system. The method described in the article can be divided into two parts.

The first part of the algorithm focuses on the vision-based thrust control. The control is based on the position of the target in the camera view of each individual MAV. The second part is used to find a consensus among all MAVs, so that every MAV is moving at the same speed and in the same direction.

2.1 Vision-based thrust control

Let us assume a formation of n MAVs. The goal of this part of the method is to control the velocity of the i^{th} MAV only by using the position of the target in the camera image. The camera image is of size $[-\frac{W}{2}\ \frac{W}{2}] \times [-\frac{H}{2}\ \frac{H}{2}]$, where W is the width and H is the height of the camera image. Therefore, the origin of this coordinate system is at the center of the camera view as seen in the Fig. 2.

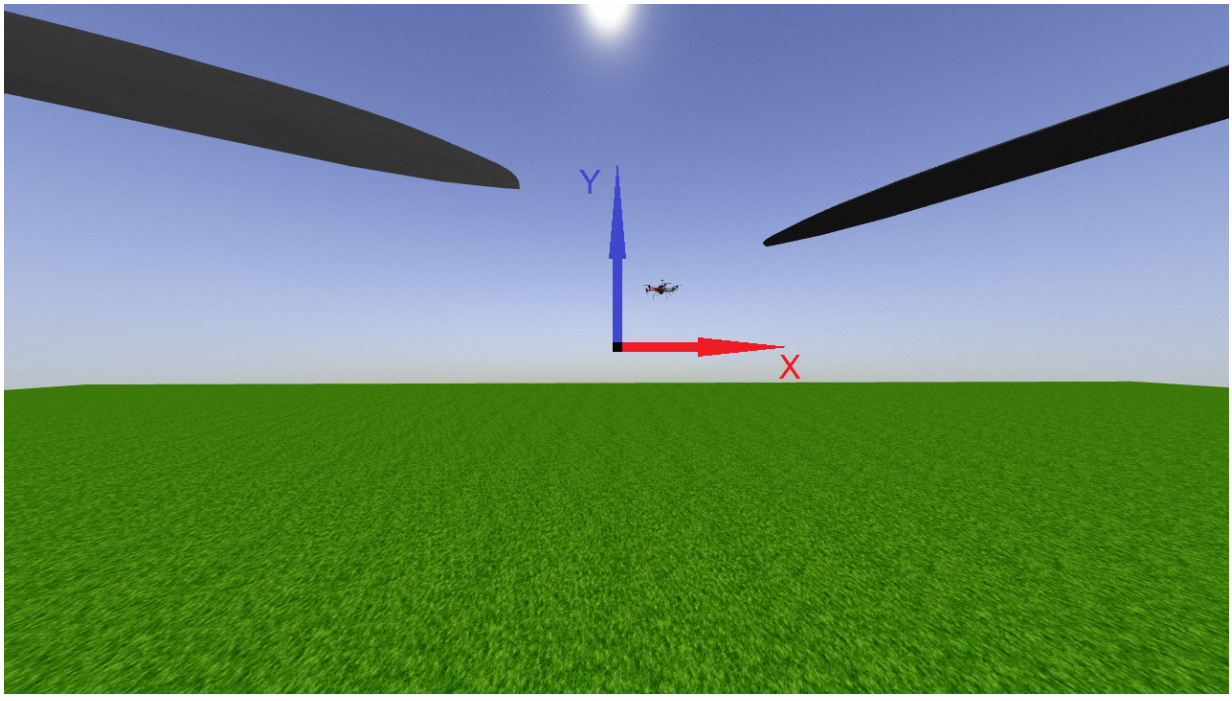


Figure 2: Image showing the coordinate system of the camera view

The method is using the data about the position of the target in the picture to compute the correct speed and direction.

The *thrust gain* $m_i(k)$ for the i^{th} MAV in the time step k is computed as

$$m_i(k) = 1 - exp\left(-\frac{1}{2}\tilde{x}_{t,i}(k)^T(\sum_m)^{-1}\tilde{x}_{t,i}(k)\right), \tag{1}$$

where $\tilde{x}_{t,i}(k) = \begin{bmatrix} x_{t,i}(k) \\ y_{t,i}(k) \end{bmatrix}$ is the position of the target in the camera image of i^{th} MAV and $\sum_m \in \mathbb{R}^{2\times 2}$ is a covariance matrix. As shown in the article by F. Poiesi, it is apparent that the further the target is from the center of the camera view, the larger $m_i(k)$ is. For example if the image of the target is in the origin of the camera view, then $m_i(k) = 0$, whereas if the target is moving out of the origin of the camera view, then $m_i(k)$ is increasing up to the value of $m_i(k) = 1$.

The *thrust direction* is obtained by projecting $m_i(k)$ onto the two virtual axes on the camera plane of the i^{th} MAV. The thrust direction is then computed as

$$\begin{aligned} a_{x,i}(k) &= sgn(\tilde{x}_{t,i}(k))m_i(k) \\ a_{y,i}(k) &= sgn(\tilde{x}_{t,i}(k))m_i(k), \end{aligned} \tag{2}$$

where *sgn()* is the *signum* function.

When the i^{th} MAV changes its position based on the output of the equations (2), the position of the target $\tilde{x}_{t,i}(k) = \begin{bmatrix} x_{t,i}(k) \\ y_{t,i}(k) \end{bmatrix}$ changes in the camera view of the i^{th} MAV and the thrust control will change its action. In order to do this it is essential to estimate the derivative of $a_{x,i}$ and $a_{y,i}$, as follows:

$$\begin{aligned} \dot{a}_{x,i}(k) &= |a_{x,i}(k)| - |a_x(k - \tau_a\Delta_k)| \\ \dot{a}_{y,i}(k) &= |a_{y,i}(k)| - |a_y(k - \tau_a\Delta_k)|, \end{aligned} \tag{3}$$

where Δ_k is the sampling period and τ_a is the interval regulation constant. The larger τ_a is, the more robust is the control, while the smaller τ_a is, the faster the reaction to a change is.

Once $\dot{a}$ is known the i^{th} MAV can set its velocity using equations (3):

$$\begin{cases} v'_{x,i}(k) = \alpha a_{x,i}(k) + v_{x,i}(k - \Delta_k) & \text{if } \dot{a}_{x,i}(k) > 0 \\ v'_{x,i}(k) = v_{x,i}(k - \Delta_k) & \text{otherwise} \\ \\ v'_{y,i}(k) = \alpha a_{y,i}(k) + v_{y,i}(k - \Delta_k) & \text{if } \dot{a}_{y,i}(k) > 0 \\ v'_{y,i}(k) = v_{y,i}(k - \Delta_k) & \text{otherwise,} \end{cases} \tag{4}$$

where constant α controls the sensitivity to target's change of movement, with a larger α meaning a more sensitive control.

The output of the set of equations (4) controls the velocity of the i^{th} MAV. These values are further used in the i^{th} MAV and its neighbors.

2.2 Distributed formation control

In the previous section, all the computation led to setting the velocity and direction of a single MAV in the whole formation. If the formation is to stay compact, there needs to be an agreement on the velocity and direction, but according to F. Poiesi, the whole formation must not use the same velocity and direction for every MAV, because each MAV should be able to slightly maneuver out of bounds of the formation. This is needed in several cases such as with loss of line-of-sight of the target or aggressive maneuvers of the target.

To account for this, each MAV communicates with its neighbors. The i^{th} MAV sends the information about its velocity and direction of flight to its neighbors and it also receives same type of data from its neighbors. The final velocity for the i^{th} MAV is computed as

$$\begin{aligned} v_{x,i}(k) &= \frac{1}{d} \sum_{j \in D_i} v'_{x,j}(k) \\ v_{y,i}(k) &= \frac{1}{d} \sum_{j \in D_i} v'_{y,j}(k), \end{aligned} \tag{5}$$

where d is the number of MAVs taken into account and D_i is the set the neighboring MAVs.

Once there is a consensus among the neighbors, it is also important to ensure that the formation will stay within some geometric constraints. This can be achieved by using the *intersection rule*. The goal of the intersection rule is to keep each MAV on the intersection of two circles. Each circle has a MAV in the center. The radius r_0 of the circle is the same for every MAV and is set at the beginning of the flight. Once a MAV is out of the formation it either continues to fly with its current computed velocity and direction or it flies to the nearest intersection. The conditions for this decision are explained below, in equation (8).

In order for the intersection rule to work there must at least 3 MAVs, so that each MAV has two neighbors. An example of a formation following this rule is seen in the Fig. 3.

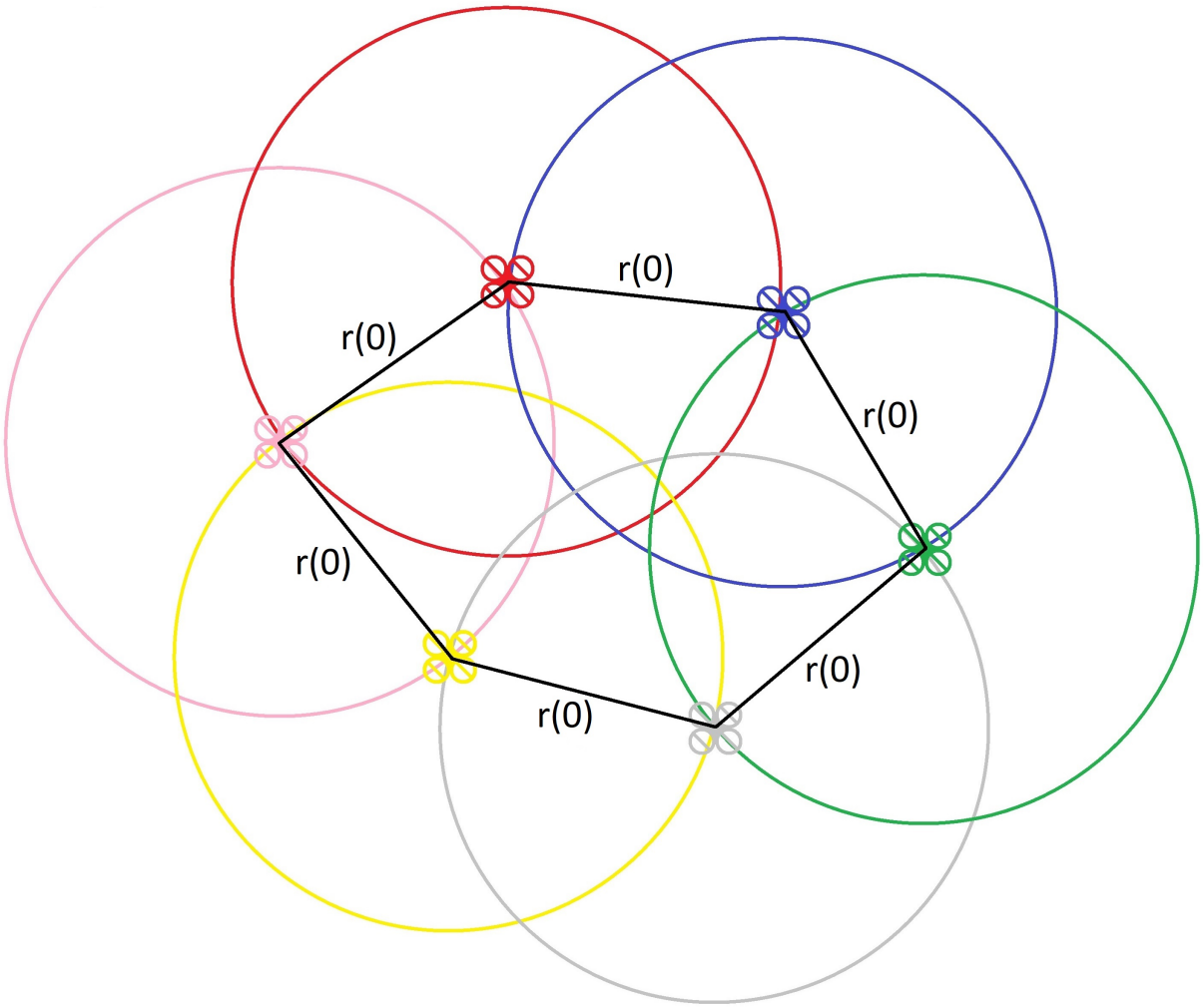


Figure 3: Basic diagram illustrating the intersection rule

The final part of this method is to maintain the formation using the combination of the output of the set of equations (4) and (5)) and of the intersection rule. In each time step, the i^{th} MAV has to decide if it is either going to fly to the nearest intersection or if it is going to continue its flight using the computed velocities.

At the beginning of each time step, the algorithm first computes a set point $x_{g,i}$ based on the previously demonstrated equations. The desired set point is calculated as

$$x^c_{g,i}(k) = x_i(k - \Delta_k) + v_i(k)\Delta_k, \tag{6}$$

where $x_i(k - \Delta_k)$ is the i^{th} position of the MAV in a previous time step and $v_i(k)$ is the currently computed velocity.

At this point the algorithm must confirm that the candidate for set point calculated in equation (6) is within the boundaries of the formation shape. Firstly, the algorithm finds

the nearest intersection as

$$\begin{aligned} p_j^c = \arg\min_{\mathrm{x}} & |x - x_{g,i}^c(k)| \ s.t. \\ & x \in \Gamma(x_{g,j}^c(k), r_{i,j}(0)) \cap \Gamma(x_{g,q}^c(k), r_{i,q}(0)), \\ & j, q \in D_i, j \neq q, \end{aligned} \tag{7}$$

where $\Gamma(x, r)$ defines the circle centered on a certain MAV with the center x and radius r, $r_{i,j}(0)$ and $r_{i,q}(0)$ are the initial distances between i^{th} and j^{th}, and i^{th} and q^{th} MAV respectively. It is assumed that $\Gamma(x_{g,j}^c(k), d_{i,j}(0)) \cap \Gamma(x_{g,q}^c(k), d_{i,q}(0))$ in not empty. Paper [13] is devoted to this case.

The algorithm then decides which set point to use. It can either be the set point $x_{g,i}^c$ calculated using the equation (6) or the nearest intersection p_j^c calculated in equation (7). The final set point is chosen as

$$x_{g,i}(k) = \begin{cases} x_{g,i}^c & \text{if } \epsilon_{i,j} > \varepsilon \text{ or } \epsilon_{i,q} > \varepsilon \\ p_j^c & \text{otherwise,} \end{cases} \tag{8}$$

with

$$\begin{aligned} \epsilon_{i,j} &= ||x_{g,i}^c(k) - x_{g,j}^c(k)|| - d_{i,j}(k) \\ \epsilon_{i,q} &= ||x_{g,i}^c(k) - x_{g,q}^c(k)|| - d_{i,q}(k), \end{aligned} \tag{9}$$

where $||\ ||$ is the ℓ^2 norm and ε is a constant that defines how big are the constrains for the formation to maintain the distances between neighboring MAVs. The larger ε is, the more constrained the MAVs are in maintaining the initial distances defined at time $k = 0$.

3 Camera vision

For this algorithm to work, it is essential to have a way to obtain the position of the target in the camera view of each MAV. While conducting the experiments with this algorithm two ways of obtaining the position were used. The first way was computing the position in the camera view using the knowledge of the absolute position of the following MAV and its target. The second way was using a convolutional neural network that was developed in parallel by other MRS team member at the CTU. In the initial stages, all experiments were conducted using only the computation of the position in the camera view. Later, when the neural network was available, several experiments were conducted using this system.

3.1 Computing positions in the virtual camera view

Before the neural network was available, a placeholder to obtain the position of the target in the camera view was needed. This was done by calculating this position using the known absolute position of the target and its follower using the camera projection matrix.

Let $\overline{x}' = [x_1, x_2, x_3]^T$ be a relative position of the target MAV to the camera and $\overline{y} = [y_1, y_2, y_3]^T$ a point in the camera view, then

$$\overline{y} = M \cdot \overline{x}', \tag{10}$$

where M is the camera projection matrix defined in OpenCV as

$$\begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{11}$$

where f_x and f_y are the focal lengths of the camera in pixels for x and y axis and c_x and c_y are the coordinates of the optical center of the camera also in pixels. The relative position $\overline{x}'$ is obtained as

$$\overline{x}' = (TR)^{-1}\overline{x}, \tag{12}$$

where $\overline{x}$ is the global position of the target, T is the transformation matrix and R is the rotation matrix. The transformation matrix T is defined as

$$\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{13}$$

where t_x, t_y and t_z are the global coordinates of the camera, and the rotation matrix R is defined as

$$\begin{pmatrix} cos\phi & -sin\phi & 0 & 0 \\ sin\phi & cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{14}$$

where ϕ is the yaw of the camera.

3.2 Neural network

In some of the experiments that were conducted for the purposes of this thesis, the MAVs used a neural network instead of the approach described in the previous section. The neural network-based detector was developed in parallel by other MRS team member at CTU. It was based on the YOLO (You Only Look Once) detection system [14].

The name *You Only Look Once* comes from the characteristic, that it processes the received image only once. For the detection and bounding box estimation, it uses a single multi-layer convolutional neural network. Because of this, YOLO is a very fast detection system that can simultaneously predict all the bounding boxes and compute probabilities expressing how likely is the target in a certain bounding box. Thanks to this, it can be used for an "on-the-fly" detection.

Before the detection starts, YOLO scales down the received image to 448 $\times$ 448 pixels. After that, the image is divided into an S $\times$S array of cells. Several bounding boxes are predicted and each is given the confidence score. YOLO them decides on the position of the object with accuracy, that reflects the confidence score.

The version of the YOLO detection system used in the experiments was *YOLOv2*. The description of this newer version of YOLO can be found in [15]. The newer YOLOv2 for example allows the user to set how the received image is scaled down.

4 Adaptation of the algorithm

In the case of this thesis, the implementation of the algorithm differs from the original description by F. Poiesi. The main focus of this section is to summarize the main differences and to describe the adjustments and modifications that are necessary in order to address challenges these changes presented.

4.1 From 2D to 3D space

The most significant difference between the original method described in the previous sections and the adaptation of said method which is the goal of this thesis, is the fact that the original method is designed for following a ground target by a formation of MAVs, that do not change their altitude and every MAV has its own camera that is tilted down. This way, all MAVs are able to see the ground vehicle. This means that the method allows for control of the position of each MAVs in the x and y coordinate, but not z, as demonstrated on Fig. 4.

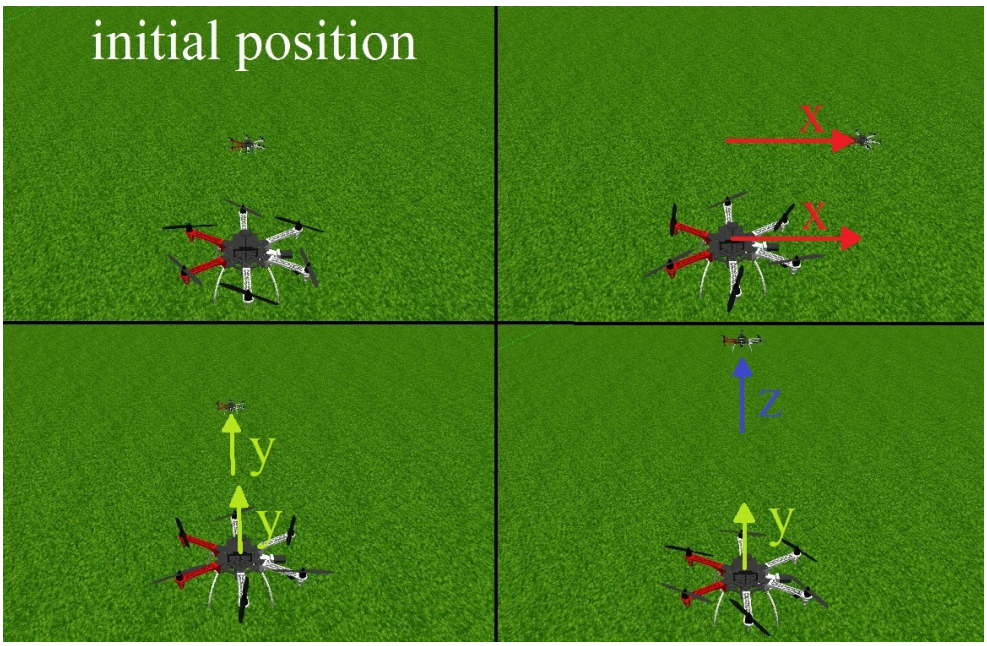


Figure 4: MAV following the target flying in fixed height above ground

Once this method is used for following an aerial vehicle that can move not only in x and y coordinate, but also in z, each individual MAV can not follow the target correctly in all three coordinates by itself and at the same time. It should also be pointed out, that cameras should be mounted horizontally on each MAV when following an aerial vehicle.

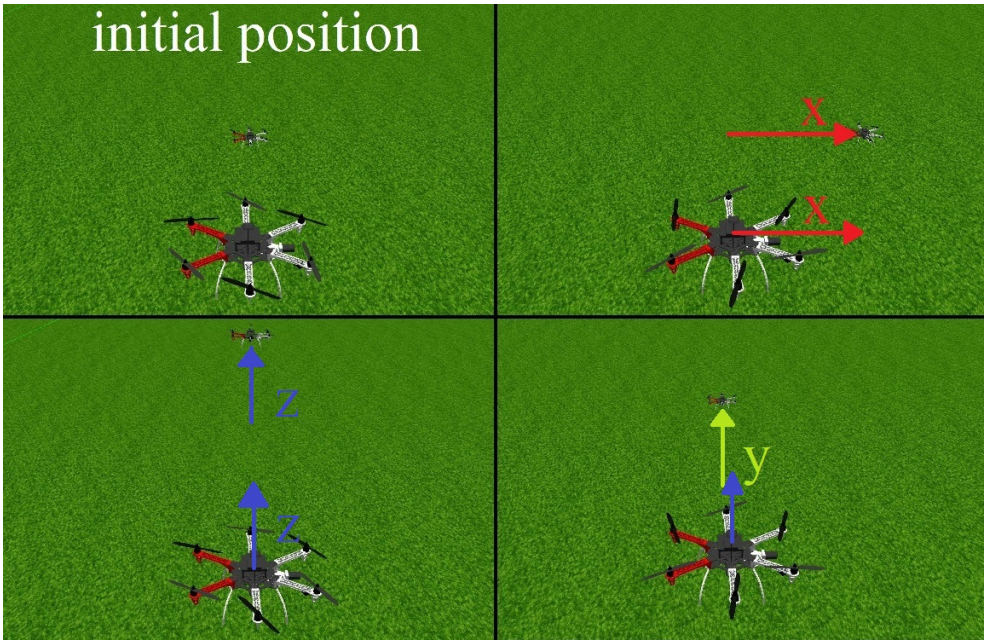


Figure 5: MAV following the target flying in all three coordinates

For a better understanding let us take this basic scenario seen on Fig. 5. The target MAV will start its flight at the point $x_t \in \mathbb{R}^3$, $x_t = [0, 0, 5]$. The follower MAV that is going to follow the target starts its flight at the point $x_f \in \mathbb{R}^3$, $x_f = [0, -5, 5]$. Whenever the target changes its position in x or z coordinate, the follower will adjust its position accordingly without any problem, but if the target changes position in y coordinate, the follower is not going to be able to interpret this change correctly, which will cause the wrong adjustment of the position of the follower and could eventually lead to losing the target from sight.

The loss of ability the to navigate correctly through the whole 3D space according to the movement of the target prevents neighboring MAVs within the formation from finding the consensus in the way described in the set of equations (5). Once the MAV has the target in the field of view of the camera, the change of the position of the target $\tilde{x}_{t,i}(k) = \begin{bmatrix} x_{t,i}(k) \\ y_{t,i}(k) \end{bmatrix}$ in the camera image, can only be translated as a change of altitude difference (change in the z coordinate), based on the change of $y_{t,i}(k)$, and combination of changes in the x and y coordinates, based on the change of $x_{t,i}(k)$, depending on the rotation of the MAV within the 3D space as well as the relative position with the target.

4.2 Intersection rule in 3D space

In the original algorithm, every single MAV maintains the same altitude the whole time. The intersection rule takes advantage of this, since once the altitude is omitted, the computation outputs the maximum of the points in space (the situation of two MAVs being in the same point in space, thus creating the infinite number of intersections, is neglected). Unfortunately, in 3D space where MAVs are able to move in all three dimensions, the sets $\Gamma(x^c_{g,j}(k), d_{i,j}(0))$ and $\Gamma(x^c_{g,q}(k), d_{i,q}(0))$ from equation (7) are not circles, but the spherical surfaces. This implies, that the intersection of said sets are not single points, but rather infinite set of points in the shape of a circle (the rare case of single point intersection is not taken into an account) as can be seen in the Fig. 6.

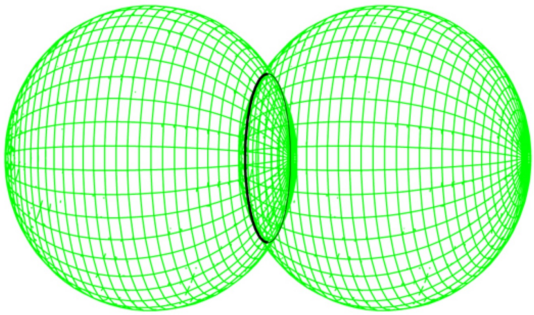


Figure 6: Two spheres intersect to form a circle [1]

There are two possible solutions for this. The first option is to find the nearest point in the set of the infinite number of points as accurately as possible, but that could have a potential impact on the performance of MAVs processing power during the flight. The second, faster solution is to omit the altitude in this computation. This leads to the same scenario, as is described in the original algorithm. The MAV finds the x and y coordinates of the intersection and computes the z coordinate using the equation (6), then it flies towards this computed point or adjusts it as explained in equation (8).

4.3 Distribution of control in the formation

For the whole formation to behave as intended, the conclusions of the previous sections 4.1 and 4.2 must be put together. The original way of finding consensus among the MAVs is not applicable, as explained above, so a different solution is called for. To ensure that the formation is able to stay within the bounds of the geometrical constraints in the 3D space without hindering the ability to follow the target correctly, the consensus on the velocity and direction computed in the equation (5) must be altered. The calculations for

[1] Source: `https://www.ck12.org/book/CK-12-Math-Analysis/section/6.1/`

each MAV must be split between calculation for the x and y coordinates, and calculation for the z coordinate. For the x and y coordinates the calculations will remain similar to the original calculations described in section 2, but the final velocities v_x for the x axis and v_y for the y axis must be calculated solely from the output of the equation

$$\begin{cases} v'_{x,i}(k) = \alpha a_{x,i}(k) + v_{x,i}(k - \Delta_k) & \text{if } \dot{a}_{y,i}(k) > 0 \\ v'_{x,i}(k) = v_{x,i}(k - \Delta_k) & \text{otherwise,} \end{cases} \tag{15}$$

as

$$\begin{aligned} v_{x,i}(k) &= abs(sin(\varphi))v'_{x,i}(k) \\ v_{y,i}(k) &= abs(cos(\varphi))v'_{x,i}(k), \end{aligned} \tag{16}$$

where φ is the yaw of the camera (the yaw of the camera is zero when the camera is facing towards the positive direction of the x axis of the local frame of the MAV).

For the z coordinate the intersection rule will not be applied, instead each MAV will be controlling its altitude based on the consensus about the altitude created from the data received from the neighbors of the MAV and from the output of the equation

$$\begin{cases} v'_{y,i}(k) = \alpha a_{y,i}(k) + v_{y,i}(k - \Delta_k) & \text{if } \dot{a}_{y,i}(k) > 0 \\ v'_{y,i}(k) = v_{y,i}(k - \Delta_k) & \text{otherwise.} \end{cases} \tag{17}$$

5 Simulation in the Gazebo simulator

Before any experiment can be performed in real-world, a thorough of testing and simulation is necessary in order to ensure that everything is working as intended. The main focus of this section is to demonstrate all the major experiments that were conducted in the Gazebo simulator.

For the purposes of understanding and transparency of each experiment, it is important to properly define the simulation environment. All the simulations were conducted in a simple environment without any obstacles. The ground was covered with a simple grass texture and the sky-box was a simple blue sky texture. The coordinate system of this simulation is shown in the Fig. 7.

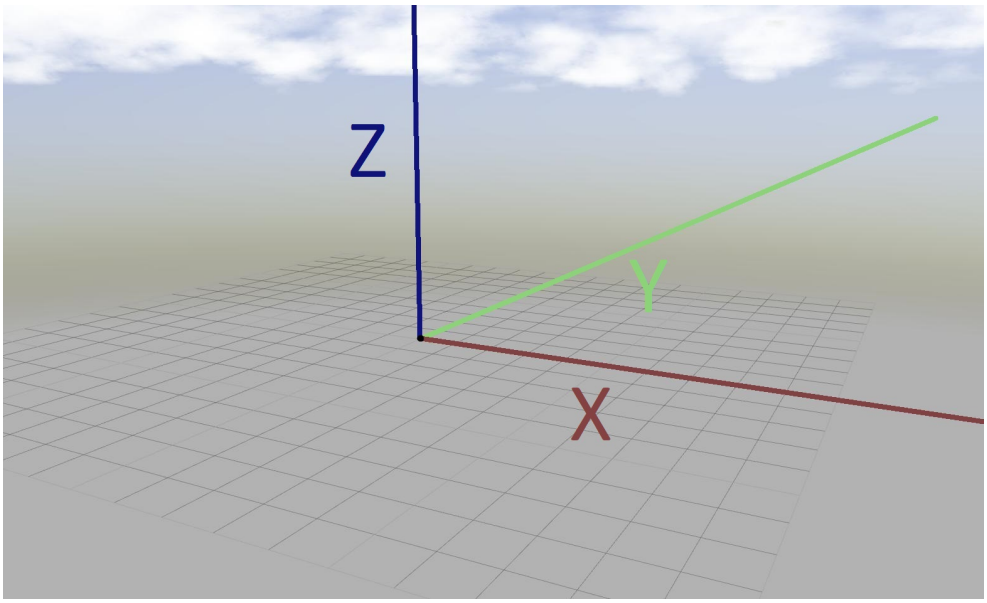


Figure 7: Coordinate system of the Gazebo simulation

5.1 One following MAV

The first experiment was designed to verify the functionality of the thrust control. Because there is only one follower in this simulation, it is not possible to take into an account any form of consensus. The initial coordinates for the target were $[0, 0, 5]$ and for the follower $[0, -5, 5]$. The following MAVs are always rotated in a way so that the camera mounted on the MAV is facing towards the target at the beginning of the simulation. The control of the following MAV was based solely on the output of the set of equations (2). In this scenario, the follower was able to change only its x and z coordinate. The x coordinate was controlled by $a_{x,i}(k)$ and z coordinate by $a_{y,i}(k)$. The experiment is shown in Fig. 8.

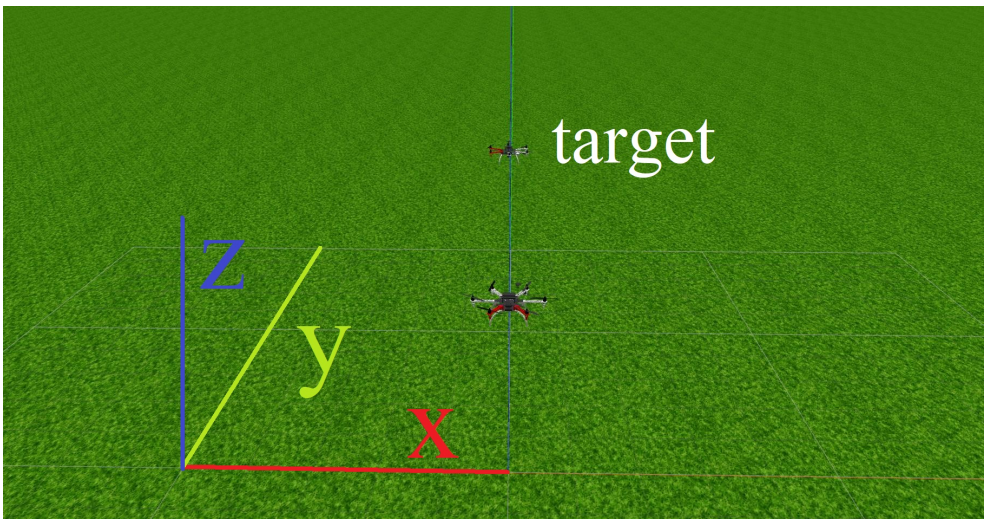
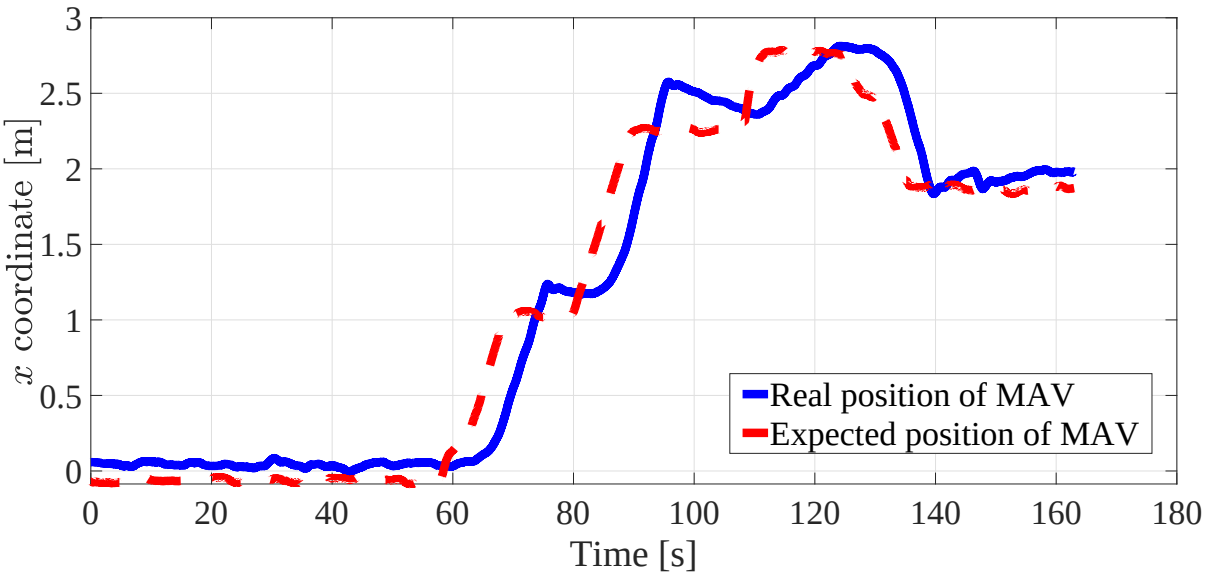


Figure 8: Simulation with one following MAV

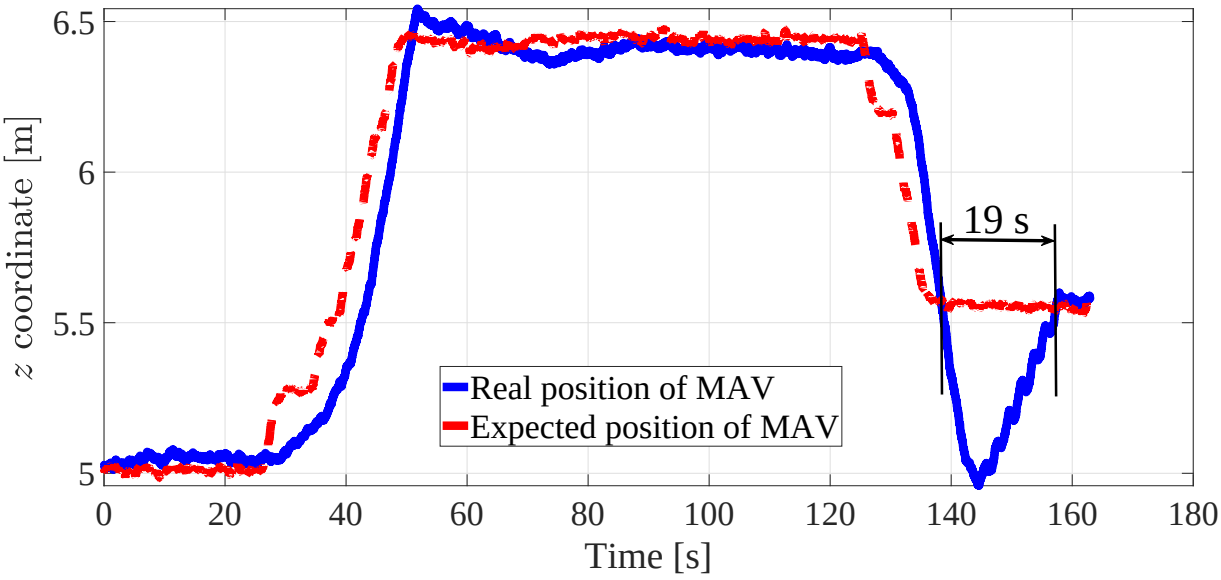
The results of this simulation are shown in the Fig. 9. How the MAV reacted to the change of the x coordinate of the target is shown in the Fig. 9a. In all the plots the blue line shows the real position of the follower during the whole simulation and the red line shows where the follower would be if the change was perfect and instantaneous, according to the position of the target. This position is obtained as

$$x_{e,i}(k) = x_i(k) + x_t(k), \tag{18}$$

where $x_{e,i}(k)$ is the expected position of the i^{th} MAV, $x_i(k)$ is the real position of the i^{th} MAV and $x_t(k)$ is the position of the target MAV. From the plot in the Fig. 9a it is apparent that the following MAV was changing its x coordinate with an acceptable precision for the application. There is a visible constant delay between the moment when the target changes its position and when the follower reacts, which is to be expected. The change in z coordinate over time can be seen in the Fig. 9b. During this experiment, there have been only two changes in this coordinate. The reaction of the follower to the first change was fast and accurate, but during the second change of z coordinate, there was an overshoot. This overshoot was most likely caused by the fact, that the target changed its position quickly in both coordinates, so the next computed set point for the follower forced the MAV to change its position too rapidly. The MAV was able to compensate for this in the x coordinate, but not in the z coordinate. This overshoot was eventually corrected in 19 seconds.



(a) Change of the x coordinate with time



(b) Change of the z coordinate with time

Figure 9: Plots showing the change of the position of the MAV in x and z coordinates

5.2 Two following MAVs

The second experiment was designed to check the same functionality as the previous experiment, with the addition of communication between the two following MAVs. Initial coordinates for the target, the first and the second follower were $[0, 0, 5]$, $[0, -5, 5]$ and $[5, 0, 5]$, respectively. The control of the following MAVs was based only on the output of the set of equations (4) and on the data received from the neighbor. In this scenario, the first follower was controlling the coordinate x for both following MAVs and the second follower was controlling the coordinate y for both following MAVs. The experiment is shown in the Fig. 10.

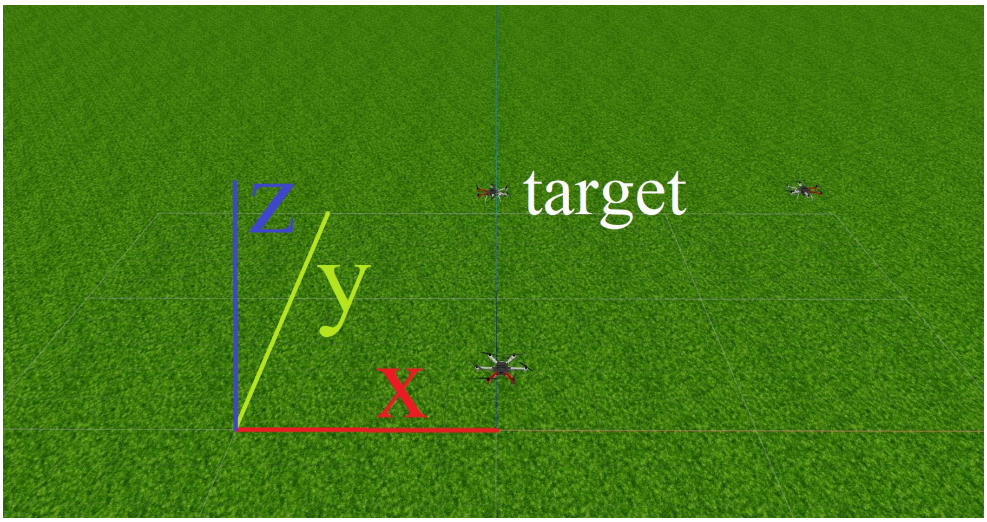


Figure 10: Simulation with two following MAVs

The Fig. 11 shows the set of plots showing the change in x coordinate over time for both followers. It is apparent that the first MAV correctly started to change its position based on the movement of the target and that it also correctly sent the data to the second MAV, since the next plot shows that the second following MAV changed its position in the same manner as the first following MAV. The plots in the next Fig. 12 show the same behavior as in the Fig. 11, but for the y coordinate. The Fig. 13 shows how MAVs reacted to change of the z coordinate of the target. In all three sets of plots are seen delays between the moment when the target changes its position and the moment when the following MAVs react. This happened, because the target changed its position only by a small distance. The thrust gain for the followers was therefore relatively low and it took a longer time to fly to the set point, but both MAVs still followed the target with a sufficient precision for this application.

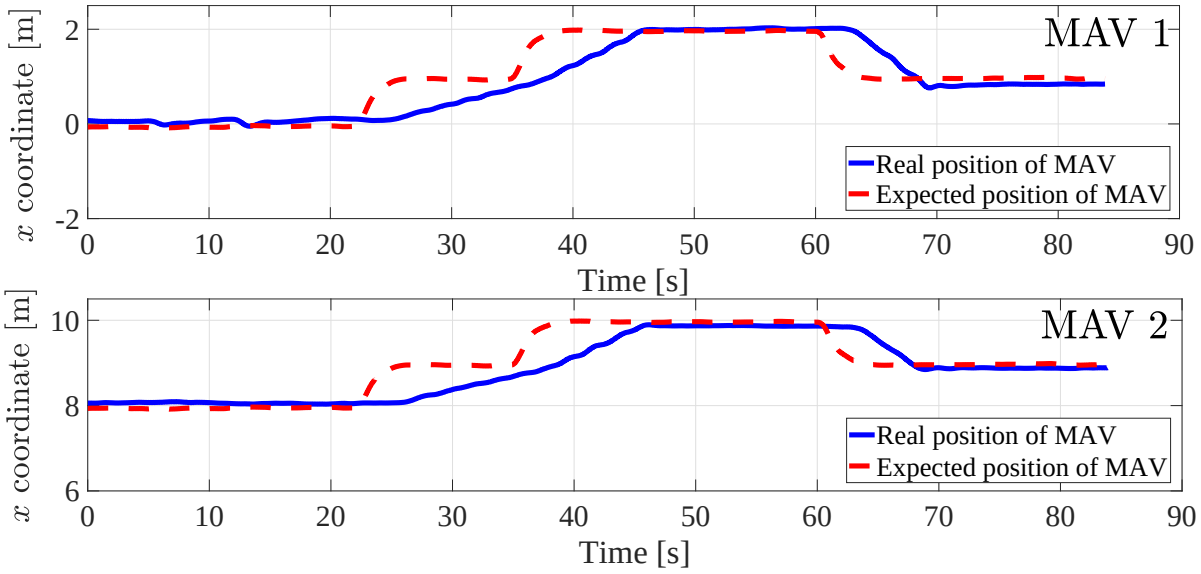


Figure 11: Plots showing the change in the x coordinate for both following MAVs in the formation

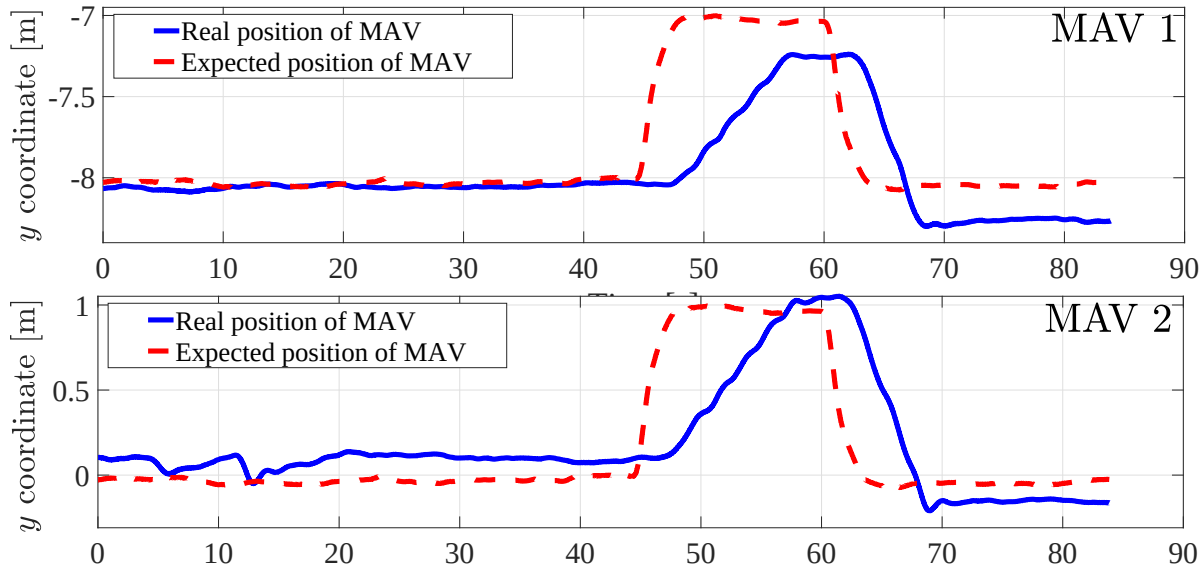


Figure 12: Plots showing the change in the y coordinate for both following MAVs in the formation

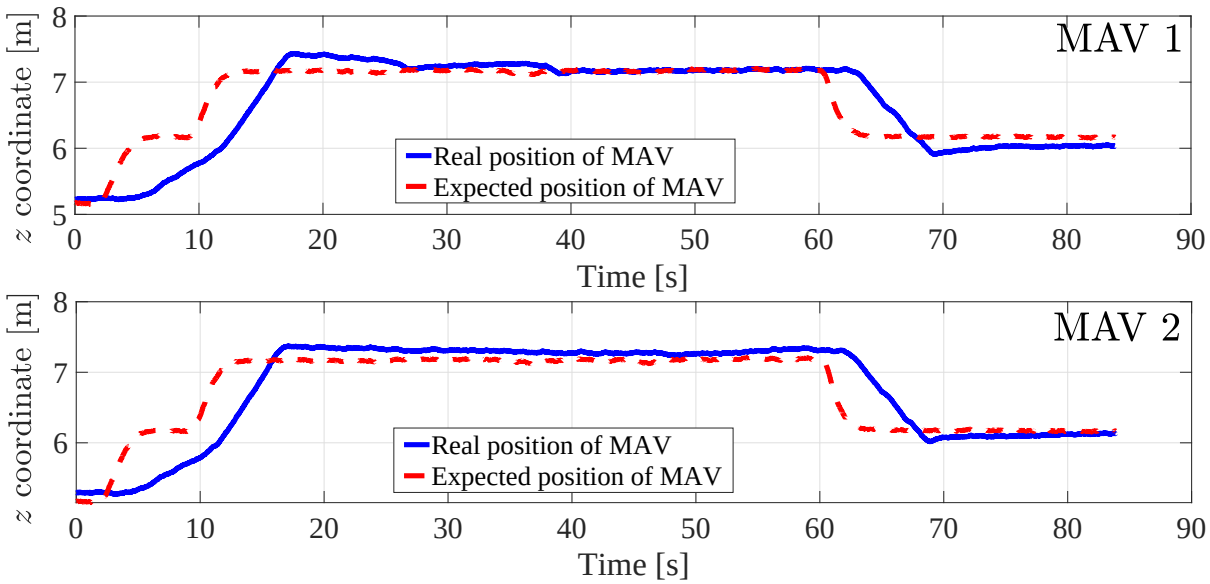


Figure 13: Plots showing the change in the z coordinate for both following MAVs in the formation

5.3 Three following MAVs

The third experiment was designed to verify the functionality of the whole method. With three following MAVs, all the parts of the algorithm were tested. Initial coordinates for the target, the first, the second and the third follower were $[0, 1.733, 5]$, $[0, -5.2, 5]$, $[6, 5.2, 5]$ and $[-6, 5.2, 5]$, respectively. The initial coordinates were chosen in this manner so that the formation would take the shape of an equilateral triangle with the target at its center. The first follower was primarily controlling the coordinate x, while the second and the third follower were primarily controlling the coordinate y, as described in section 4.3. The experiment is shown in the Fig. 14. This simulated experiment was recorded. The video recording is provided on the CD.

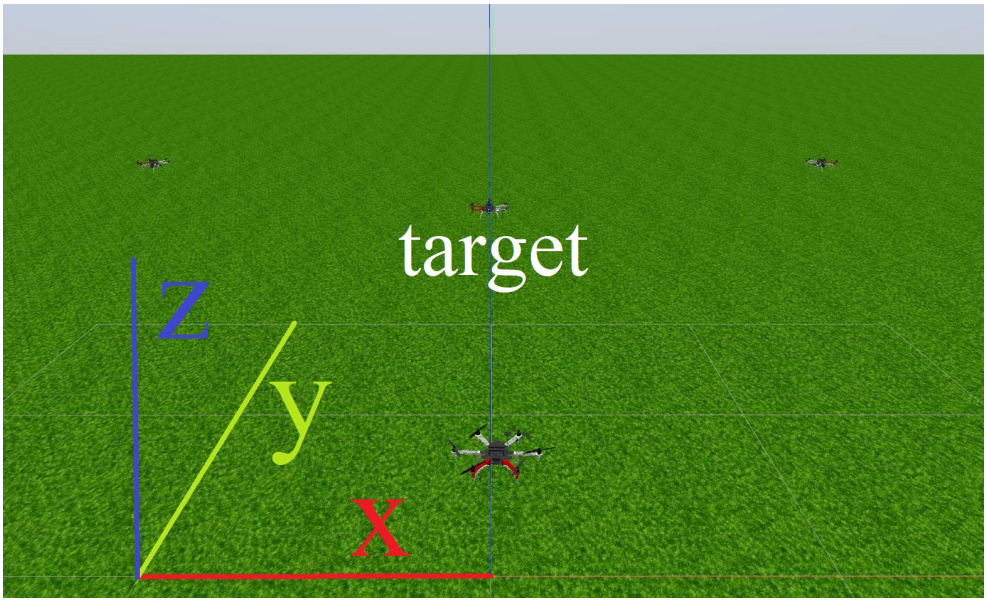


Figure 14: Simulation with three following MAVs

This simulation was somewhat more complex than the rest because second and third following MAVs had to compute the change in the targets y coordinate together. The changes in x coordinate were computed primarily by the first following MAV. The Fig. 15 and Fig. 16 show how each MAV behaved. Some deviation from the correct position can be found, but the following of the target was consistent. The Fig. 13 shows a set of plots showing the changes in the z coordinate for each MAV. There is a notable large overshoot, that happened at the beginning of the simulation with the MAV 1. The cause of this overshoot is difficult to pinpoint. There is a small change of the x coordinate at the same time as the overshoot happened. It is possible that the change of altitude have rotated the MAV 1 slightly to the right. This is a plausible explanation for the small change of the x coordinate. Because of that the MAV 1 had to adjust not only the z coordinate, but also the x coordinate, which could have caused the overshoot.

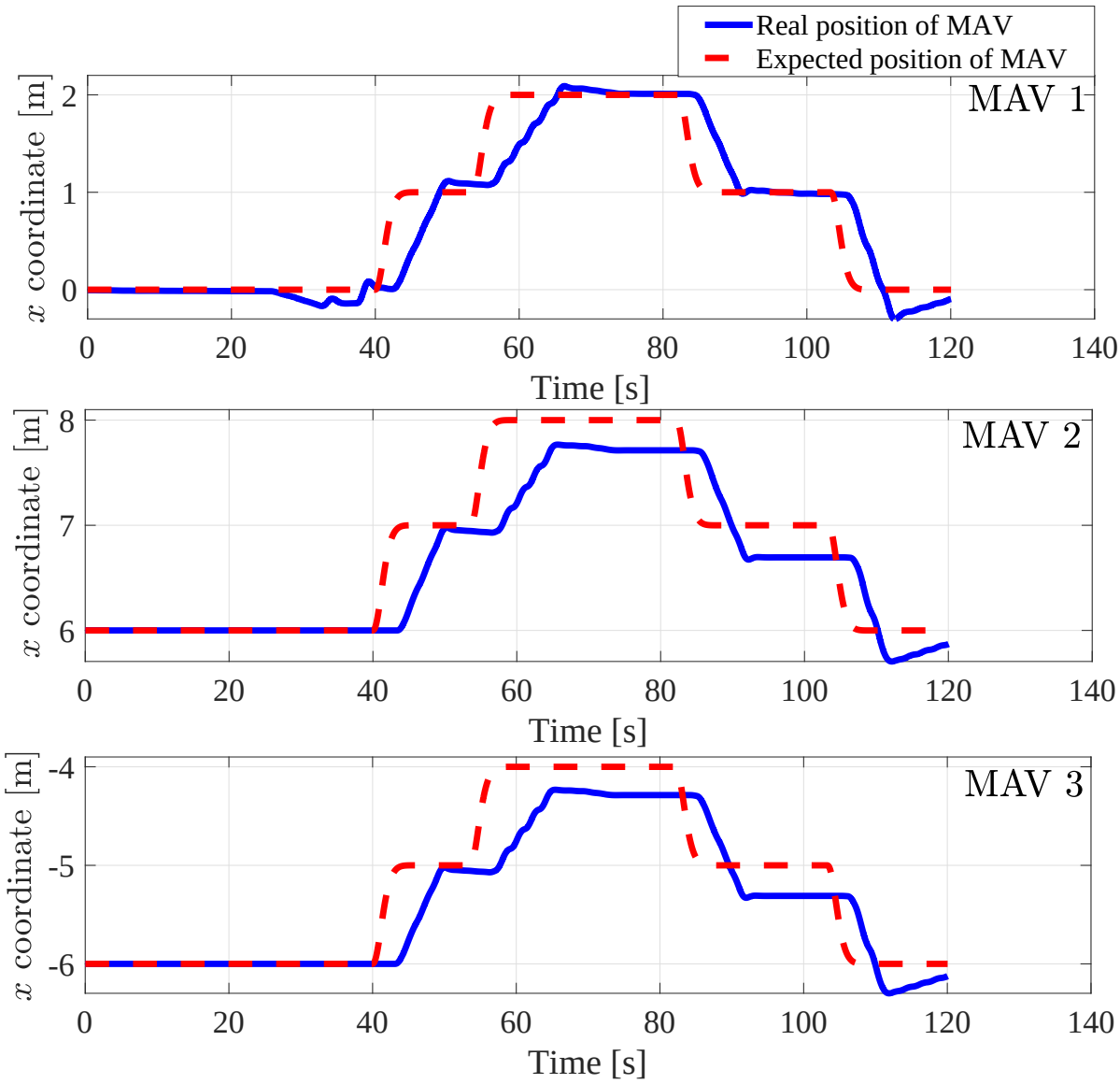


Figure 15: Plots showing the change in the x coordinate for all three following MAVs in the formation

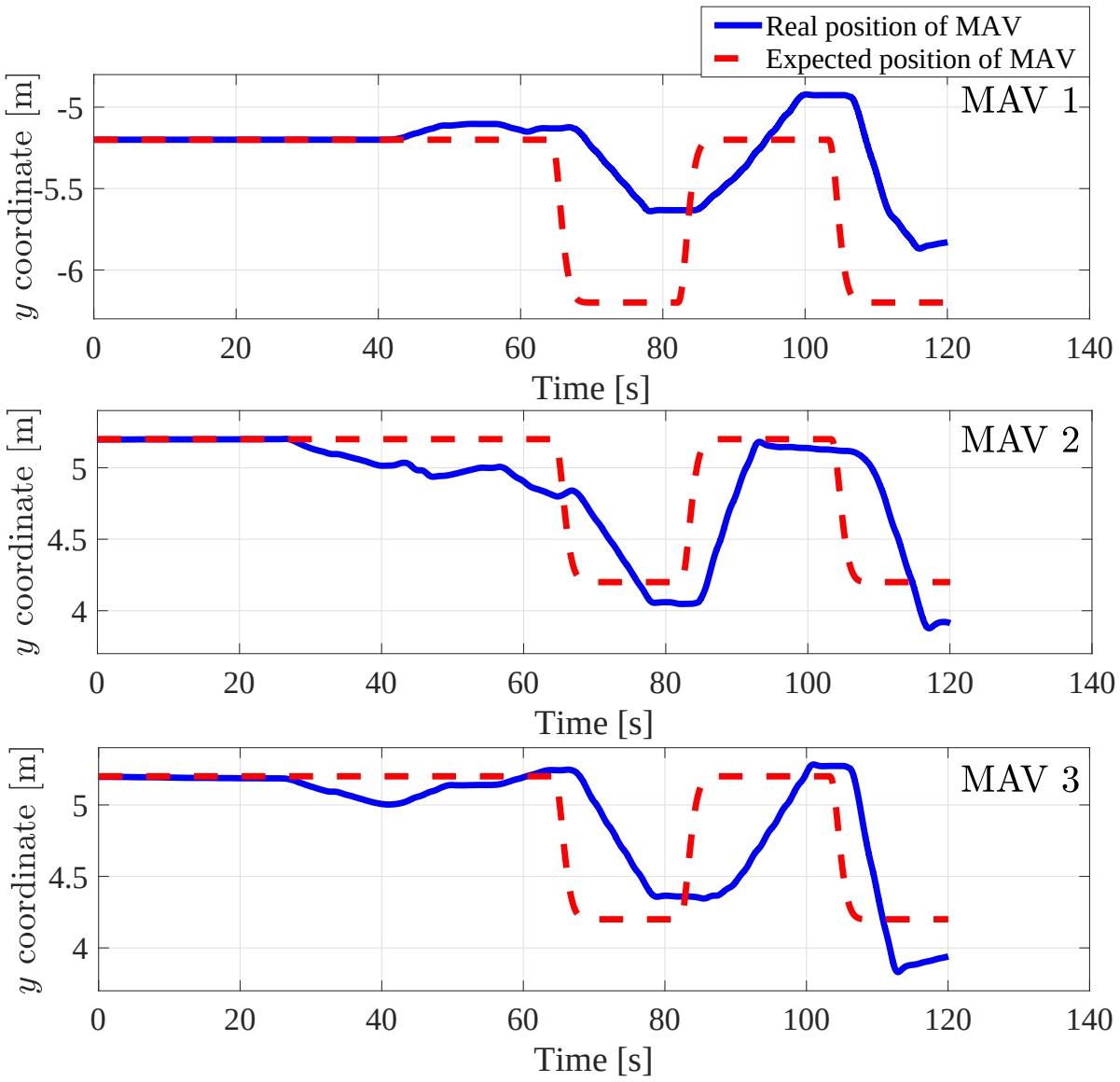


Figure 16: Plots showing the change in the y coordinate for all three following MAVs in the formation

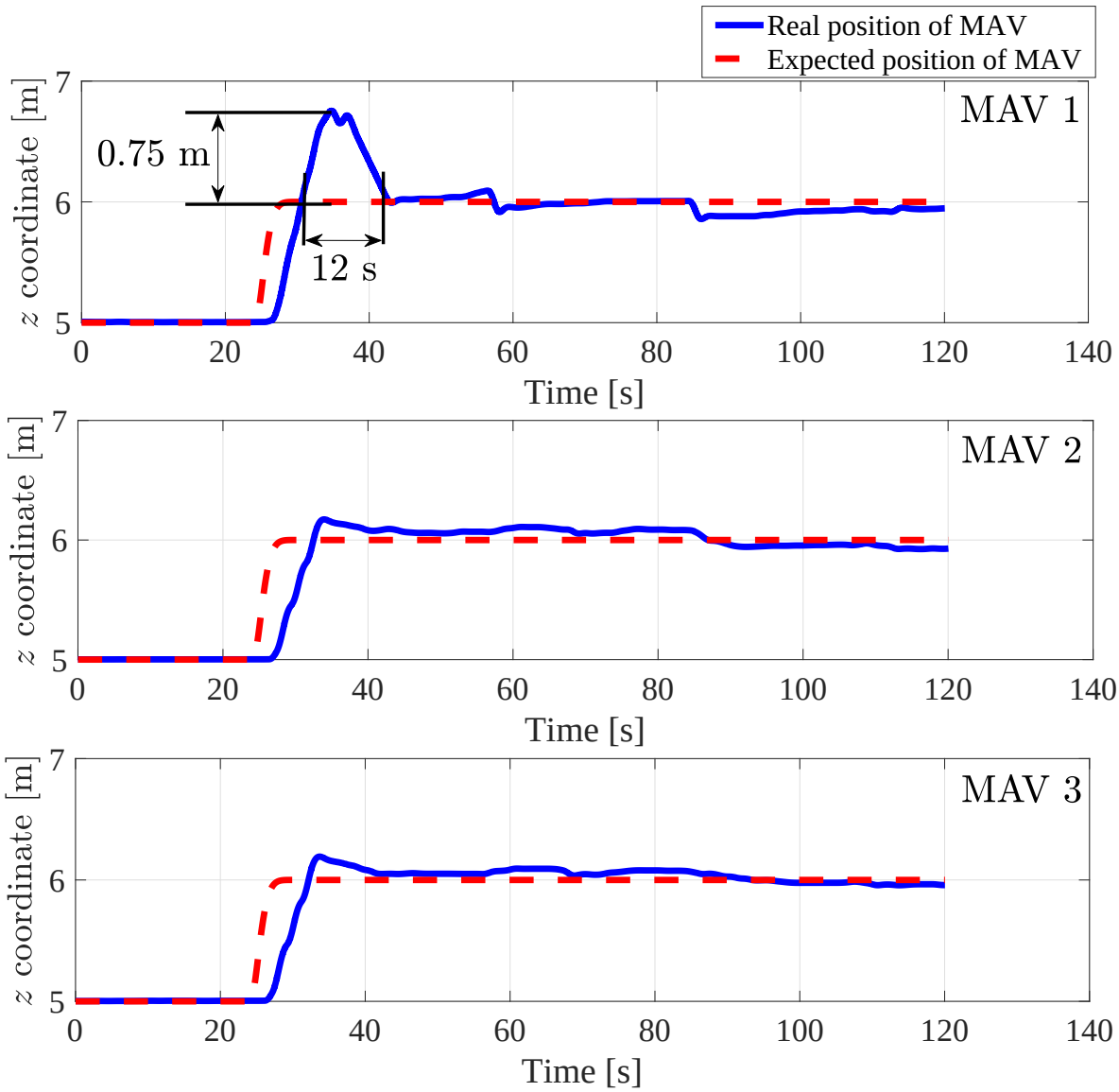


Figure 17: Plots showing the change in the z coordinate for all three following MAVs in the formation

5.4 Four following MAVs

The fourth and final experiment conducted in the Gazebo simulator was designed to verify the functionality of the whole algorithm with a formation of 4 MAVs. The initial coordinates for the target, the first, the second, the third and the fourth follower were $[0, 0, 5]$, $[0, -5, 5]$, $[5, 0, 5]$, $[0, 5, 5]$ and $[-5, 0, 5]$, respectively. The experiment is shown in the Fig. 18.

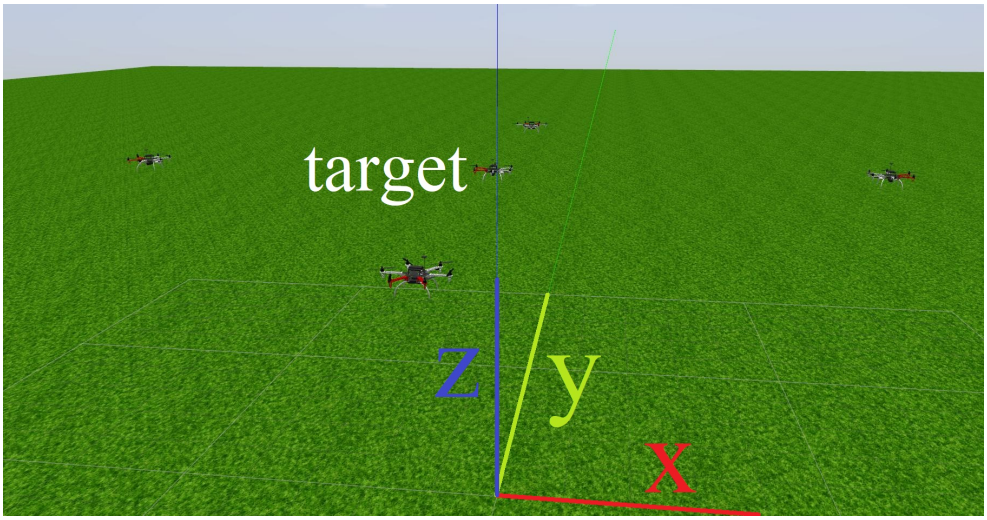


Figure 18: Simulation with four following MAVs

This is the largest simulation that was conducted for this thesis in the Gazebo simulator. All the changes of the x, y and z coordinates for all four following MAVs are shown in the Fig. 19, Fig. 20 and Fig. 21. Upon closer inspection, same situations seen in the previous simulations can be found. Sometimes, the following MAV is close to where it is supposed to be (the MAV 1 and the MAV 2 plot in the Fig. 19, the MAV 1 and MAV 4 plot in the Fig. 20,the MAV 2 and MAV 3 plot in the Fig. 21) with some small deviations that are corrected in a few seconds (the MAV 3 plot in the Fig. 19, the MAV 2 plot in the Fig. 20) as well as same more significant overshoots that are eventually corrected after a few seconds (the MAV 4 plot in the Fig. 19, the MAV 3 plot in the Fig. 20, the MAV 1 and the MAV 4 plot in the Fig. 21).

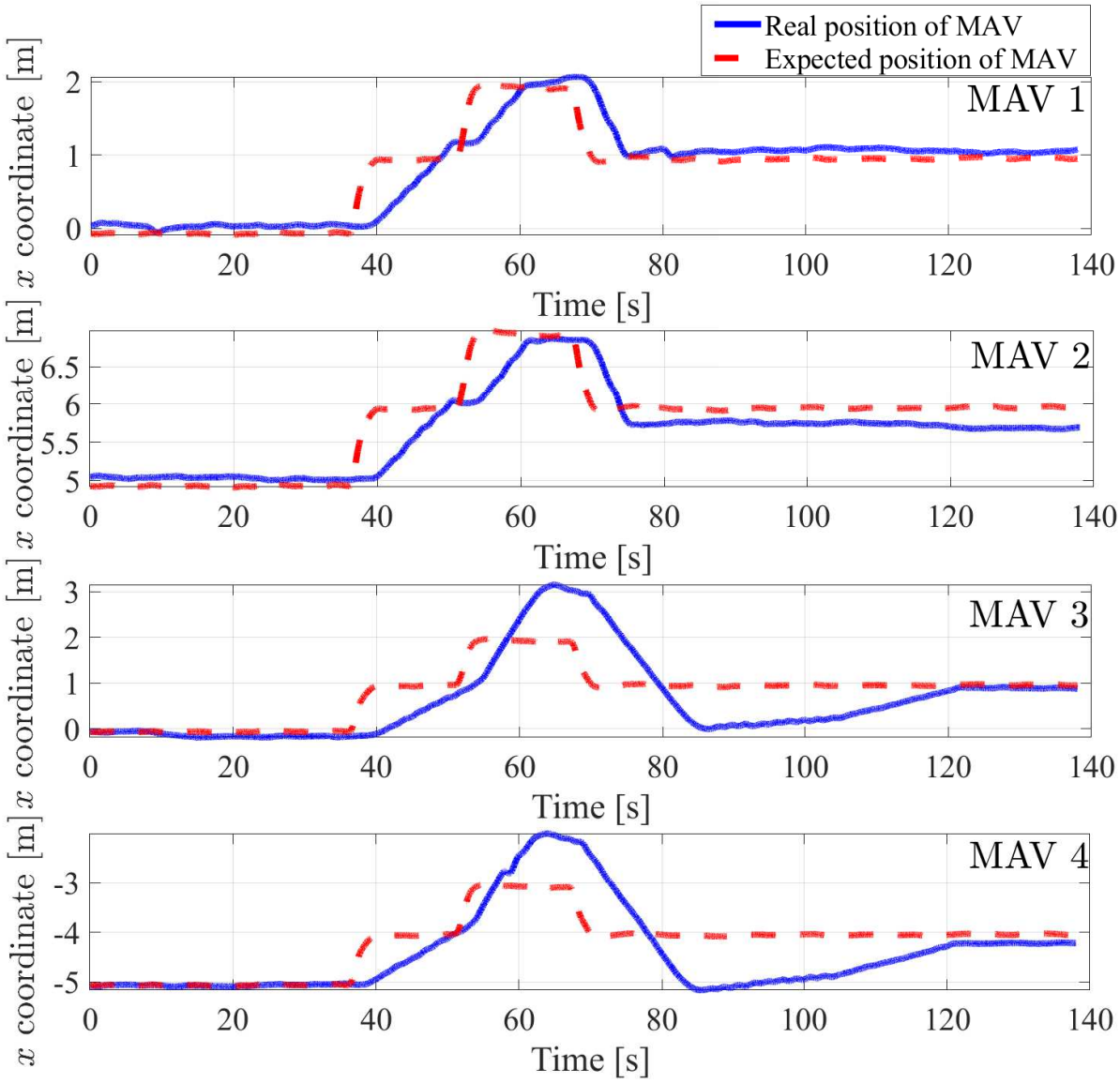


Figure 19: Plots showing the change in the x coordinate for all four following MAVs in the formation

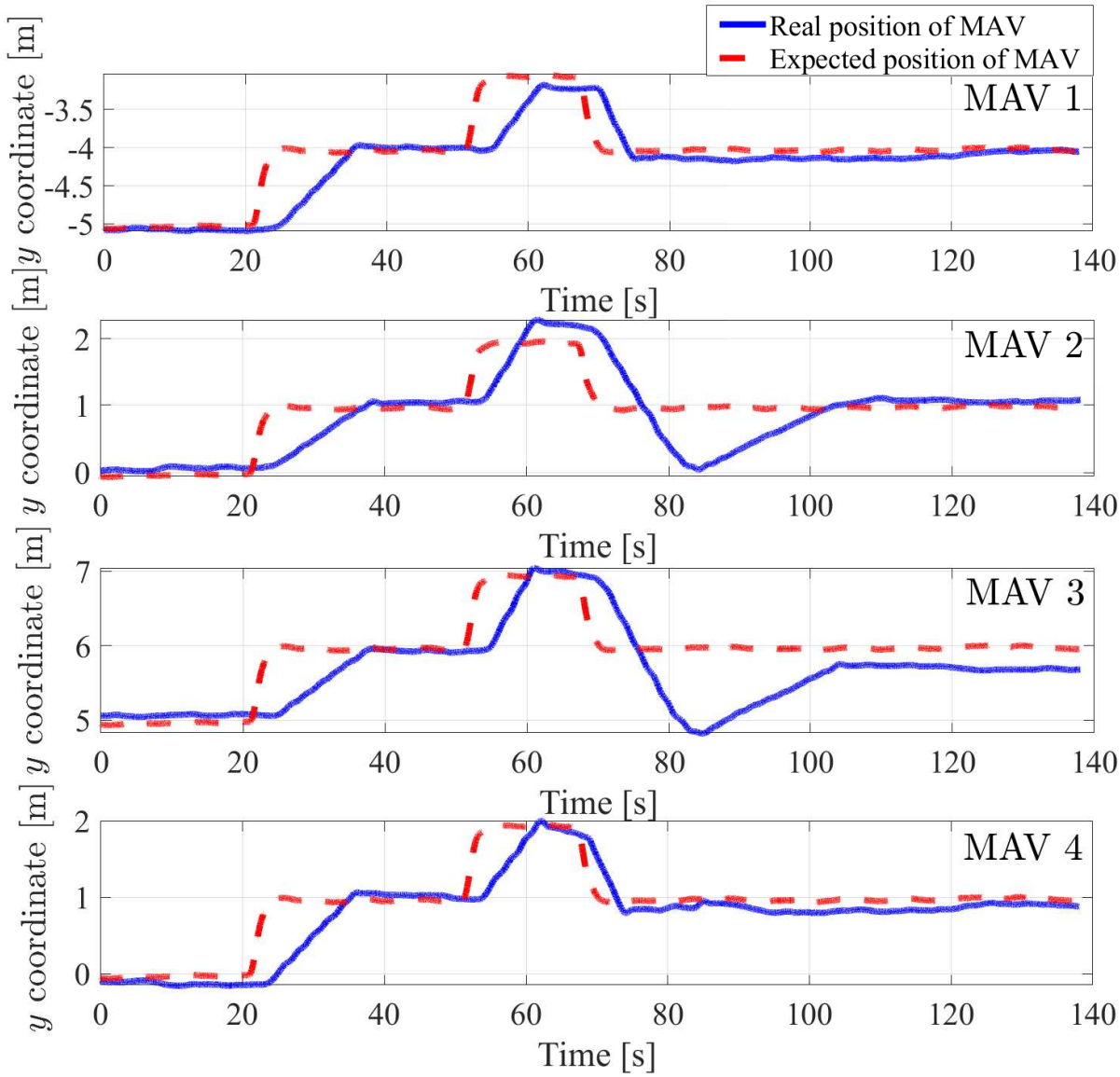


Figure 20: Plots showing the change in the y coordinate for all four following MAVs in the formation

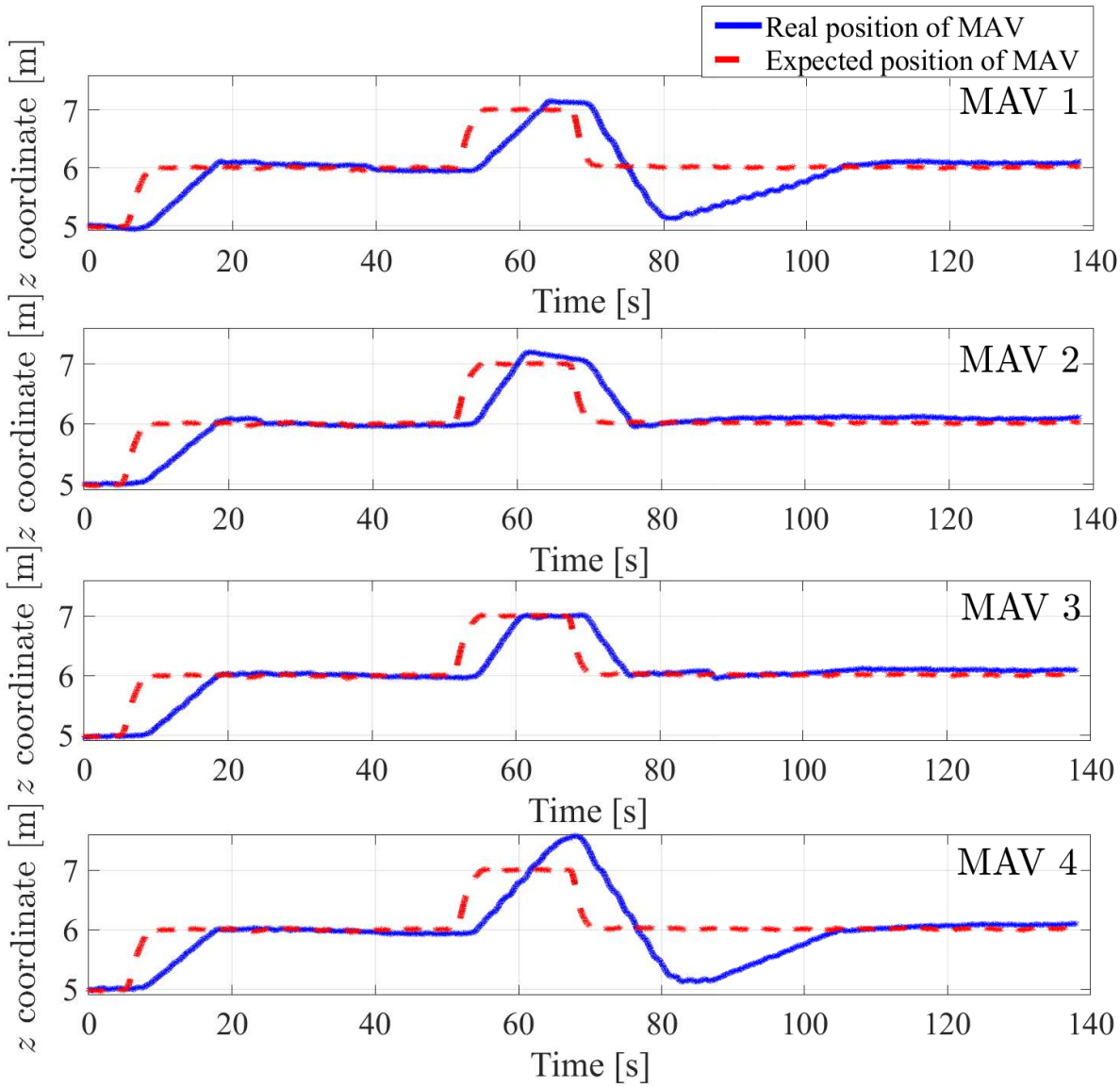


Figure 21: Plots showing the change in the z coordinate for all four following MAVs in the formation

6 Real-world experimentation

This section describes the real-world experiments and analyzes the outcome of these experiments. Two real-world experiments were conducted using the algorithm described in this thesis. One was conducted using the computation of the position in the virtual camera view as described in the section 3.1 and one was conducted using the convolutional neural network described in the section 3.2.

Even though the MAVs used in the simulation are not too different in term of behavior from how the real MAVs of the MRS group at the CTU behave, there are some obvious and less obvious differences that need to be taken into account when transferring the experiment from simulator into reality. For example, in the simulator the exchange of data between MAVs is instant and without limitations while in the real-world experiment this communication is in fact limited and the bandwidth of the Wi-Fi connection that is used for communication may not be high enough to transfer all the data if the broadcast is too fast, which can lead to an overload of the Wi-Fi communication.

Figure 22: Real-world experiment with one following MAV

6.1 Experiment with one MAV follower

The first real-world experiment was designed in a similar way to how the simulated experiment from the section 5.1 was designed. The only part of the algorithm that could be verified in this experiment was the thrust control. The initial coordinates for the target were $[20, -10, 4]$ and for the follower $[20, -15, 4]$. The control of the following MAV was based solely on the output of the set of equations (2). The scenario was the same as in the section 5.1. The follower was able to change only its x and z coordinate appropriately. A photograph of the experiment is shown in the Fig. 22. This experiment was conducted using the convolutional neural network described in the section 3.2. It is important to point out that this experiment was conducted at a time when the YOLO-based detector was not set up correctly and was outputting the data with a delay of several seconds. Additionally, during the experiment the YOLO-based detector transmitted some false detections, as seen in the Fig. 23 and Fig. 24.

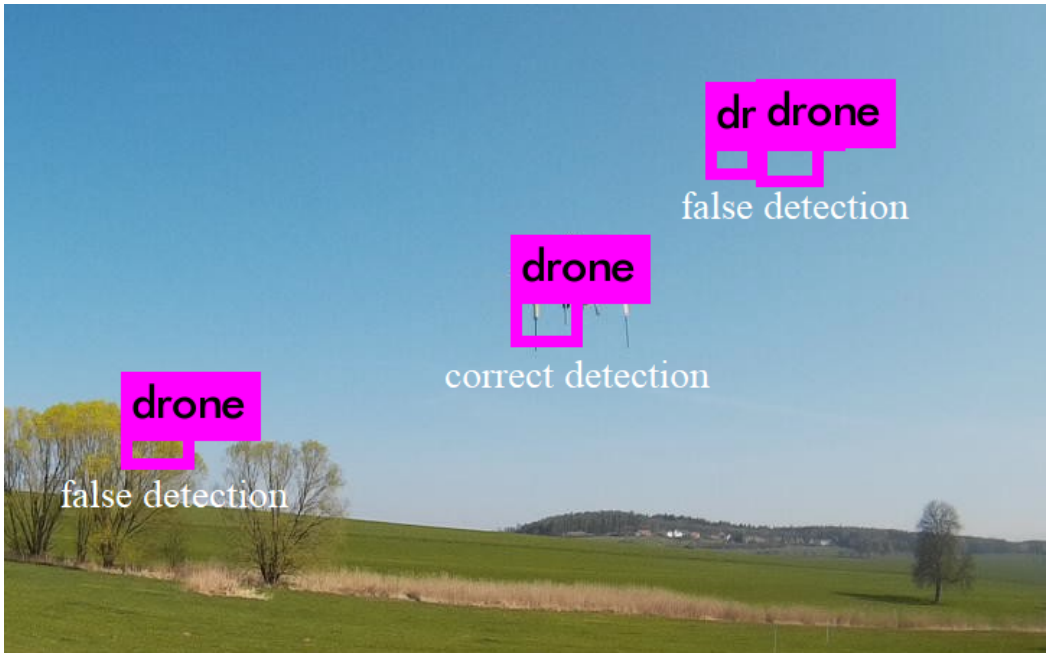


Figure 23: Two false detections

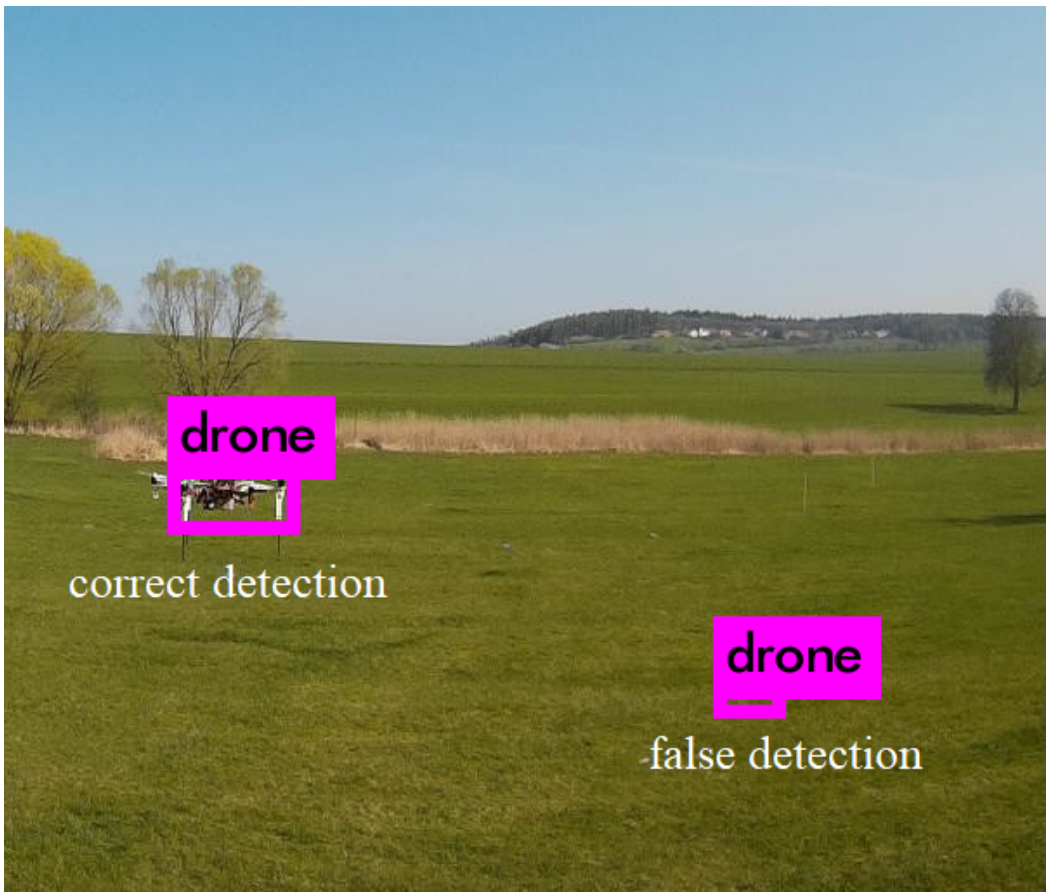
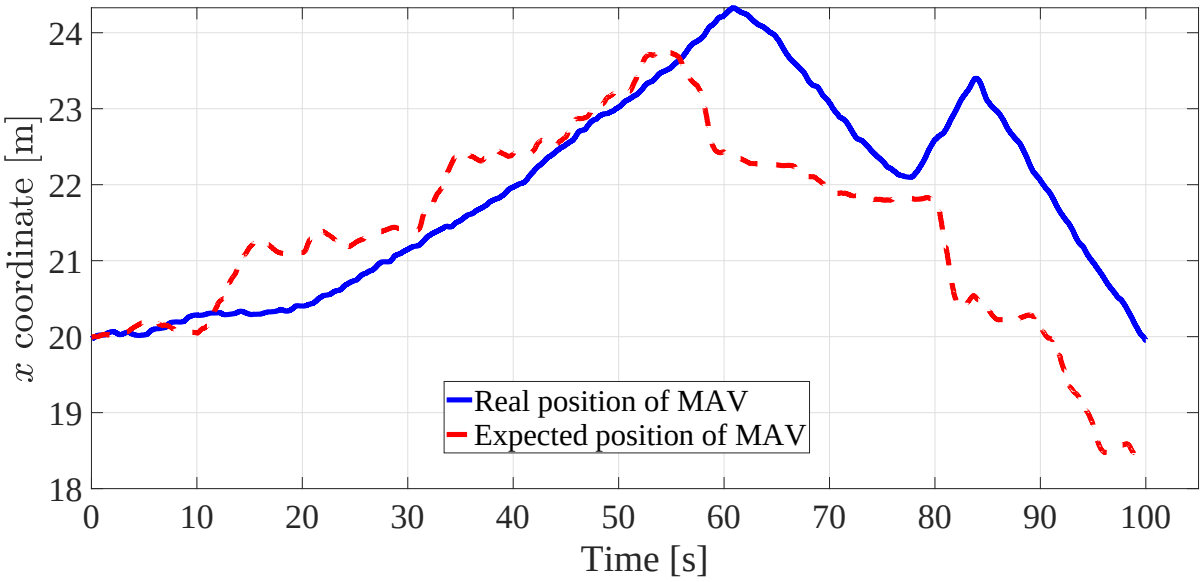


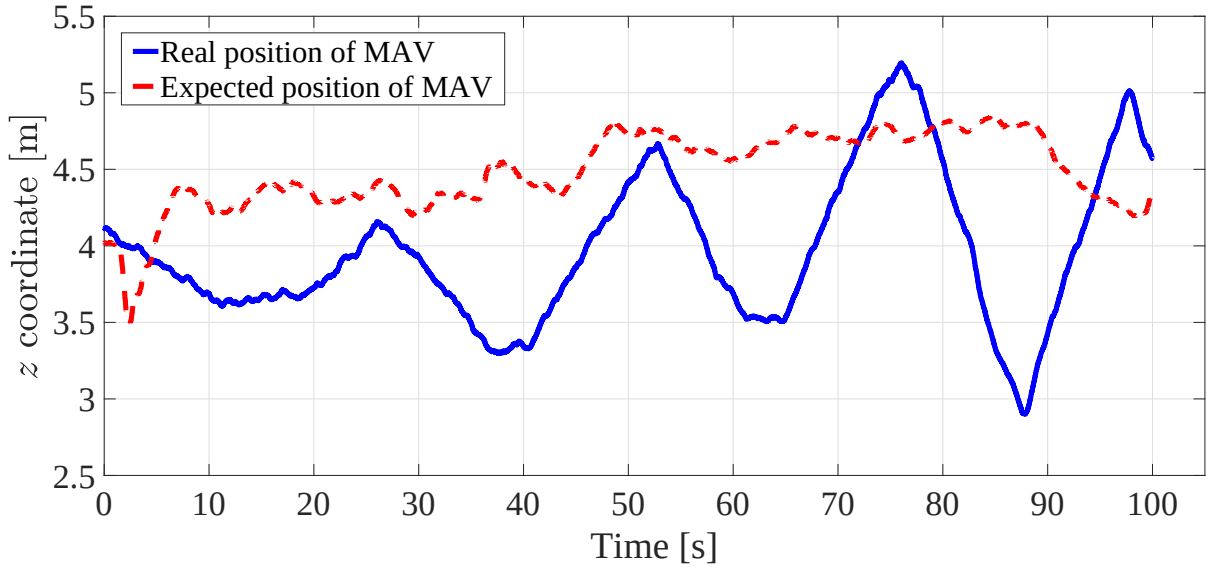
Figure 24: One false detections

6.2 Results of the first experiment

For the experiment, there had been some adjustments made in the algorithm to compensate for the delay of the YOLO-based detection system. Specifically, the speed of the reaction on the change of the position of the target was significantly reduced. This change resulted in a considerable reduction of the size of the overshoots. After an analysis of the Fig. 25a and Fig. 25b it can be said, that the follower MAV was changing its direction of flight with acceptable precision. However the delay of the YOLO-based detection system still influenced the experiment negatively. The following MAV was still overshooting the position of the target, especially in the z coordinate, as seen in the Fig. 25b.



(a) Change of the x coordinate in time



(b) Change of the z coordinate in time

Figure 25: Plots showing the change in the x and z coordinates for the following MAV in the 2nd experiment

6.3 Experiment with two MAV followers

The second real-world experiment was designed in a similar way the simulated experiment from the section 5.2. The shape of the formation was the same (one target and two following MAVs) and the initial coordinates were similar. The initial coordinates for the target, the first and the second follower were $[0, 0, 5]$, $[0, -8, 5]$ and $[8, 0, 5]$, respectively. The scenario was the same as in the section 5.2. The first follower was controlling the coordinate x for both following MAVs and the second follower was controlling the coordinate y for both following MAVs. The photo of the experiment is shown in the Fig. 26. This experiment used the computation of the position in the virtual camera view described in the section 3.1, because the MAVs used in this experiment did not have the neural network integrated in them.

Figure 26: Real-world experiment with two following MAVs

6.4 Results of the second experiment

The plots showing the results of the second real-world experiment can be seen in the Figs. 27, 28 and 29. Upon closer inspection of each plot, it can be said that the formation of the two MAVs was following the target with a relatively good accuracy. For the first 80 seconds, the formation was able to change the positions of both MAVs very quickly and quite accurately. At this point of the experiment, it was decided to change the position of the target more rapidly and by a greater distance. The result of this change has resulted in the loss of the ability of the formation to follow the target as accurately as it did before. This effect can be seen in the figures, especially from 80s onward.

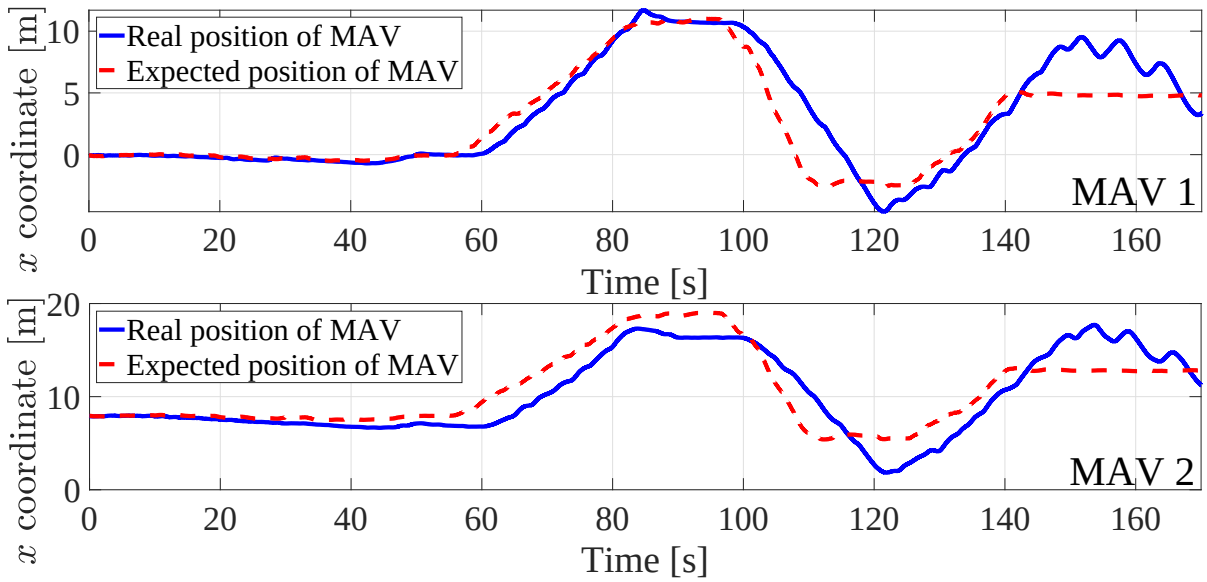


Figure 27: Plots showing the change in the X coordinate for the following MAVs in the experiment

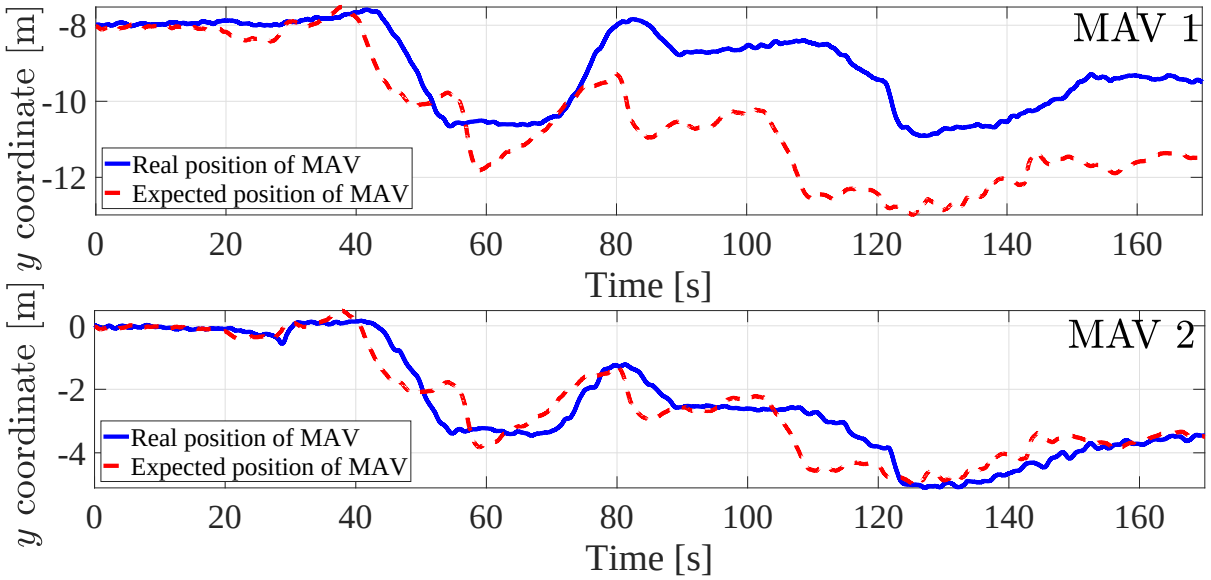


Figure 28: Plots showing the change in the Y coordinate for the following MAVs in the experiment

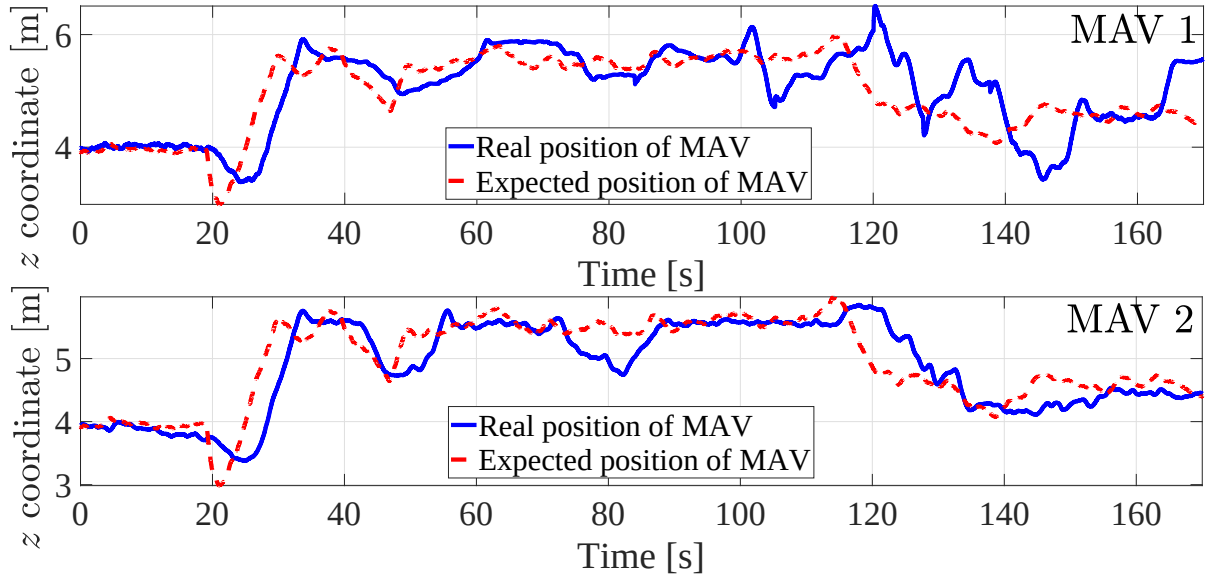


Figure 29: Plots showing the change in the Z coordinate for the following MAVs in the experiment

7 Conclusion

The goal of this thesis was to implement an algorithm for pursuing a moving helicopter by a team of unmanned aerial vehicles localized relative to each other and the target using the article [14] by F. Poiesi and A. Cavallaro. The algorithm explained in the section 2 was implemented in the C++ programming language and integrated into the ROS system. It was then adapted for the purposes of this thesis, as explained in the section 4. The implemented algorithm was successfully verified in the Gazebo simulator. The results of all the experiments conducted in the Gazebo simulator can be found in the section 5. The algorithm was then used in an experimental verification with the multi-MAV platform of the MRS group [10]. The results of the real-world experiments can be found in section 6. Based on the results of all the simulations and experiments, it can be concluded, that the implemented algorithm is working accurately if the movement of the pursued target is fluent and not too fast. With increasing speed of the target the ability of the formation to follow the target accurately decreases. This problem can be addressed in the future work on the project.

References

[1] M. Saska, J. Chudoba, L. Preucil, J. Thomas, G. Loianno, A. Tresnak, V. Vonasek, and V. Kumar. Autonomous deployment of swarms of microaerial vehicles in cooperative surveillance. *Proceedings of 2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014.

[2] M. Saska, V. Vonasek, 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*, 2016.

[3] 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, January 2018.

[4] V. Roberge, M. Tarbouchi, and G. Labonté. Fast genetic algorithm path planner for fixed-wing military uav using gpu. *IEEE Transactions on Aerospace and Electronic Systems*, 2018.

[5] F. Dell'Agnola, L. Cammoun, and D. Atienza. Physiological characterization of need for assistance in rescue missions with drones. *2018 IEEE International Conference on Consumer Electronics (ICCE)*, January 2018.

[6] M. Saska. Large sensors with adaptive shape realised by selfstabilised compact groups of micro aerial vehicles. *International Symposium on Robotic Research*, 2017.

[7] 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. *IEEE ETFA*, 2017.

[8] Woodrow Bellamy III. Drones came too close to airplanes 1,800 times in 2016. *www.aviationtoday.com*, March 2017. http://www.aviationtoday.com/2017/03/17/drones-came-close-airplanes-1800-times-2016/.

[9] F. Poiesi and A. Cavallaro. Distributed vision-based flying cameras to film a moving target. *International Conference on Intelligent Robots and Systems, Hamburg, Germany*, September 2015.

[10] 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.

[11] K. Hausman, J. Müller, A. Hariharan, N. Ayanian, and G. S. Sukhatme. Cooperative control for target tracking with onboard sensing. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[12] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and C. Bouman. Multi-target detection and tracking from a single camera in unmanned aerial vehicles (uavs). *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[13] B.D.O. Anderson, B. Fidan, C. Yu, and D. Walle. Uav formation control: Theory and application. *Recent Advances in Learning and Control*, 371:15–33, 2008. Springer.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, June 2016.

[15] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, July 2017.

[16] T. Baca, M. Platkevic, J. Jakubek, A. Inneman, V. Stehlikova, M. Urban, O. Nentvich, M. Blazek, R. McEntaffer, and V. Daniel. Miniaturized x-ray telescope for vzlusat-1 nanosatellite with timepix detector. *Journal of Instrumentation*, 11(10):C10007, 2016.

Appendix A CD Content

In Table 1 are listed names of all root directories on CD.

Directory name	**Description**
Pursuing of a moving helicopter by a compact team of micro aerial vehicles	the thesis in pdf format
Simulated experiment	Video of the simulated experiment with three MAV followers described in the section 6.3
3MAVpursuit	An example of the implementation used in the experiment described in the section 6.3
Real-world experiment	Video of the real-world experiment described in the section 5.3

Table 1: CD Content

Appendix B List of abbreviations

In Table 2 are listed abbreviations used in this thesis.

Abbreviation	**Meaning**
MAV	Micro Aerial Vehicle
MAVs	Micro Aerial Vehicles
ROS	Robot Operating System
3D	Three Dimensional
2D	Two Dimensional
YOLO	You Only Look Once

Table 2: Lists of abbreviations