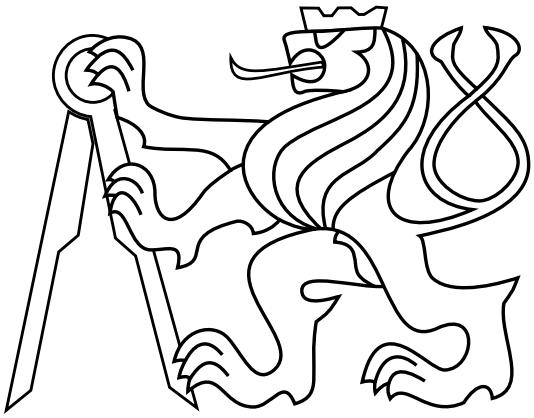
Bachelor thesis

Decentralized Model of a Swarm Behavior Boids in ROS

Pavel Petráček



May 2017

Thesis supervisor: Ing. Martin Saska, Dr. rer. nat.

Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics

Acknowledgement

Firstly, I would like to thank my family for their eternal support throughout my life. Special acknowledgment belongs to Dr. Martin Saska, whose advice and guidance allowed me to complete this thesis. Furthermore, I would also like to thank to all the members of the MRS group, whom were always willing to provide their professional assistance. I also thank the Faculty of Electrical Engineering for the interesting studies, knowledge and experience it gave me during the three years. Final thanks belongs to the Czech nation for the suppression of the communist party, providing freedom to study, and for its great tradition in brewery.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, May 25, 2017

.....

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

BACHELOR PROJECT ASSIGNMENT

Student: Pavel P e t r á č e k

Study programme: Cybernetics and Robotics

Specialisation: Robotics

Title of Bachelor Project: Decentralized Model of a Swarm Behavior Boids in ROS

Guidelines:

The goal of the thesis is to design, implement in ROS (Robot Operating System) and experimentally verify in Gazebo simulator a decentralized algorithm for control a swarm of unmanned aerial vehicles (UAV) in an environment with obstacles. The following tasks will be solved:

- To understand the system of Multi-Robot Systems group at CTU being developed for the MBZIRC competition in ROS and to integrate the Boids model of a swarm behavior [2,1] with low-level MPC controller [3] in a decentralized fashion.
- To implement an obstacle avoidance function [5] enabling to stabilize the swarm in environment with obstacles.
- To verify the implementation in the Gazebo simulator in 2D and 3D scenarios in a forest like environment.
- To study influence of reliability and precision of relative localization of neighboring particles on the swarming behavior.
- To prepare the system for an experimental verification with the multi-UAV platform of the Multi-Robot Systems group (thesis advisor will decide whether the real experiment or a complex statistical analysis in the simulator will be realized based on availability of the system).

Bibliography/Sources:

[1] Alexander G. Madey and Gregory R. Madey. 2013. Design and evaluation of UAV swarm command and control strategies. In Proceedings of the Agent-Directed Simulation Symposium (ADSS 13), 2013.
[2] Reynolds, C. Flocks, Herds, and Schools: A Distributed Behavioral Model. Computer Graphics 1987; 21(4), 25-34.
[3] T. Baca, G. Loianno and M. Saska. Embedded Model Predictive Control of Unmanned Micro Aerial Vehicles. In 21st International Conference on Methods and Models in Automation and Robotics (MMAR). 2016.
[4] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," IEEE Transactions on Automatic Control, vol. 51, no. 3, pp. 401-420, 2006.
[5] 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.

Bachelor Project Supervisor: Ing. Martin Saska, Dr. rer. nat.

Valid until: the end of the summer semester of academic year 2017/2018

L.S.

prof. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, February 14, 2017

Abstrakt

Tato práce se zabývá návrhem, implementací a experimentálním ověřením systému pro stabilizaci roje bezpilotních a plně autonomních helikoptér s využítím rozšířeného rojového modelu Boids. Za cíl si klade návrh robustního a decentralizovaného systému pro nasazení v komplexních prostředích s vysokou hustotou překážek, a jeho integraci do roje reálných helikoptér. Práce bere v úvahu omezení vyplívající z využití vícerobotických systémů pracujících v reálném čase. Shlukovací a navigační schopnosti roje bezpilotních helikoptér v komplexních prostředích byli ověřeny mnohými simulacemi a reálnými experimenty.

Klíčová slova

bakalářská práce, bezpilotní, autonomní, helikoptéra, uav, roj, boids, ros, gazebo

Abstract

This thesis deals with design, implementation and experimental verification of system proposed for stabilization of a swarm of unmanned and fully autonomous helicopters, using an expanded swarming model Boids. The main focus of this work lies in a proposal of robust and decentralized swarming behavior suited for complex environments with high density of obstacles, and its integration to a swarm of real helicopters. Constraints of multi-robot systems working in real time are considered. The capability of the swarm of unmanned helicopters to cluster and navigate in complex environments was verified in various simulations and real experiments.

Keywords

bachelor thesis, unmanned, autonomous, uav, swarm, flocking, boids, ros, gazebo

Contents

Abbreviations

FEE	Faculty of Electrical Engineering
CTU	Czech Technical University in Prague
MAV	Micro aerial vehicle
UAV	Unmanned aerial vehicle
UUV	Unmanned underwater vehicle
MRS	Multi-Robot Systems group at FEE CTU
Boids	Flocking algorithm presented by Craig Reynolds in 1986
ROS	Robot Operating System
MPC	Model Predictive Control
V-REP	Virtual Robot Experimentation Platform
SyRoTek	System for robotic e-learning
SPOF	Single Point of Failure
NASA	National Aeronautics and Space Administration
GPS	Global Positioning System
SITL	Simulation in the Loop

1 Introduction

In the last few years, software and hardware equipment for embedded systems have extremely developed and led to an enormous advance in aerial vehicles technology. Such technology led to the creation of relatively low-cost aerial vehicles without a human pilot aboard called Unmanned Aerial Vehicles (UAV) or Micro Aerial Vehicles (MAV). Many types of these vehicles had been developed in recent years, differing mainly in design, number of used rotors and onboard hardware. Teams around the world started to make UAVs autonomous allowing to control bigger groups of UAVs, where human factor is suppressed.

Creation of autonomous robots led to a new robotics development branch, which focuses on completely autonomous robot behavior, including UAVs. Single UAV can be applied to many different applications such as localization, monitoring, package delivery or many other tasks. However, using multiple autonomous helicopters increases efficiency, speed and performance in performing of specific task.

There are two main approaches for deployment of groups of autonomous helicopters. Both approaches are based on real-time onboard computing and coordination of individual helicopters. The first one is based on formations of helicopters using direct communication, deterministic behavior and defined shapes of the formation. The second approach, described in this thesis, is built on a decentralized control of autonomous, self-organizing groups of UAVs called *swarms*, which use relative onboard localization of their neighborhood.

The biggest advantage of a swarm of UAVs is its decentralization. Decentralized swarm model will not disintegrate, in case of an individual failure. It also reduces hardware demands for individual UAVs as there is no need for uniformed computational power, which is decentralized onto all of the swarm members. However, a necessary preliminary for an UAV to be part of a swarm is a capability of relative localization of its neighborhood.

Core of this thesis is the analysis and presentation of a simple, yet robust, swarming model, based on *Boids* [1], extended with obstacle avoidance and navigational functions. Furthermore, the thesis focuses on design, implementation and verification of theoretical basis on simulations in complex environments, using Robot Operating System (ROS) and Gazebo. Also, an important part of the thesis is overcoming of challenges, brought up by the use of swarming behavior on robotic systems and to ensure its safety and reliability in real applications. Research provided by this thesis is part of grant *Stabilization and control of teams of relatively-localized micro aerial vehicles in high obstacle density areas* provided by Czech Science Foundation[1].

[1] Further information at `https://gacr.cz/en/`.

1.1 State of the art

Swarm behavior was primarily researched for non-robotic systems, but with recent development in robotics, this behavior is being developed and optimized for robust multi-robot stabilization. Non-robotic swarm behavior was designed for computer simulations of a flock of birds [1] to be used in movies and games [2] or intelligent motion of groups in strategy games [3].

The main problem with robotic systems is that mainly theoretical models with verification on swarms of dimensionless particles were developed [4][5]. Research for those models lack further designed approaches and experiments with real hardware and software, which brings up many challenges to overcome. Overcoming these challenges and deployment of a swarm of helicopters in a real-world environment is the main goal of this thesis.

Multi-robot systems implementing swarming behavior are being developed for public and military applications. Those applications include development of an architecture for control of a swarm of Unmanned Aerial Vehicles by a single operator [6], autonomous search and rescue systems [7][8], land mine detection [9], cooperative surveillance [10], detection and tracking of a contaminant plume [11] or patrolling of a defined body of water for detection and tracking of vessels of interest [11].

The easiest way to verify functionality of swarming behavior in real world is by using of low-cost micro robots. With their development in recent years, many projects tried to create micro-robots and apply swarming behavior to them, starting with *LIBOT Robotic System* [12] and *Colias* [13]. The largest documented swarm, consisting of 1024 individual robots, was demonstrated by Harvard University in 2014 [14]. Micro-robot swarm systems are also planned to be used for exploration of the solar system by NASA[2] [15].

MRS[3] group have published many papers about development of the swarming behavior. The previous work of the group was primarily focused on formation flying [16][17][18][19][20][21][22][23], surveillance by compact UAV groups [10][24][25] and stabilization and navigation of swarms of UAVs [26][27][28]. Further swarm development was focused on swarm motion planning [29], swarm shape optimization [30] and escape behavior in swarms of unmanned helicopters [31]. However, even though all of the MRS group papers took into consideration constraints of real helicopters, most of them had been verified only by computer simulations and not by a real deployment of swarms of helicopters.

Primary focus of this thesis is to present real applications of swarming helicopters while taking real hardware and software limitations into consideration. The thesis introduces a possibility to utilize a simple swarming model, based on Boids [1], for real multi-robot system of UAVs in an environment, where the GPS[4] is a reliable positioning system. Usage of the GPS is necessary, due to lack of any relative onboard localization, which is not based on the GPS and is reliable and accurate enough to be used. Nevertheless, such relative onboard localization is in a development process by the MRS group and its integration to the swarming model presented in this thesis is a part of additional work in the future. The GPS can be classified as a simplistic positioning system, which is reliable and accurate enough in an open environment for the first deployment of a swarm of helicopters.

[2] National Aeronautics and Space Administration
[3] Multi-Robot Systems
[4] Global Positioning System

Furthermore, this thesis is focused on keeping optimal formation in plane, where all UAVs maintain a constant altitude, or 3D space. Research of influence of accuracy and precision of relative onboard localization is also performed. Safety of autonomous swarms and its surroundings is an important part of the development and should be perfectly examined before its usage in real applications [32].

1.2 Problem definition

We define the swarm as a group of decentralized, homogeneous, autonomous and self-organized UAVs[5]. The main goal of this thesis is to present a swarming model for autonomous UAVs. To do so, low-level controllers [33] developed for MBZIRC 2017 competition by MRS group at Faculty of Electrical Engineering at CTU in Prague, University of Pennsylvania and University of Lincoln will be used.[6] Utilization of these UAV controllers is a necessary prerequisite for development of the swarming model.

A decentralized model suggests, that the swarming algorithm needs to run on the processing unit of each UAV. Movement of an UAV shall be solely computed from onboard information about its relatively localized neighbor UAVs and obstacles, predefined in the controller. In further description of the model, we define a controlled UAV as h_0 and UAVs relatively localized by h_0 as set $\vec{h}_0 = [h_1 \dots h_n]^T$, where n is number of localized UAVs by UAV h_0. Furthermore, presenting of the flocking forces in this thesis is attached with figures, whose template is illustrated on Fig. 1, where r (m) is a radius of sphere, whose center is a center of gravity of UAV h_0, and $n = 2$, therefore $\vec{h}_0 = [h_1, h_2]^T$. Such radius indicates the range of spherical space, where the UAV h_0 (marked by the blue point) is capable of localization of other UAVs (marked by the green points). UAVs unable to be localized by UAV h_0 are marked with the red point. This scheme is prepared for integration of onboard relative localization (such as the system [34][35], which was used in our group for stabilization of UAV formations), although in hardware experiments in this thesis the GPS[7] emulates the relative localization.

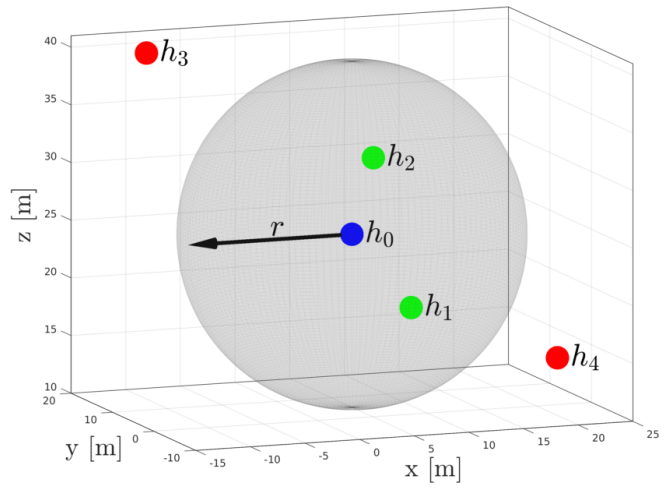


Figure 1 A visualization of a relative localization by UAV h_0 in 3D space. Number of localized UAVs $n = 2$.

[5] Unmanned Aerial Vehicles
[6] Hereinafter MBZIRC system of the MRS group, see `http://mrs.felk.cvut.cz/projects/mbzirc`.
[7] Global Positioning System

A swarm must be capable of a movement in a complex environment with obstacles, presented as cylinders, trees, lamps, pillars and walls. A localization and type recognition of the obstacles is not part of this thesis, therefore the obstacle types, their position and other required parameters will be given to the swarm algorithm as an input.

Proposed swarming model must ensure sufficient safety distance during the swarm deployment between all UAVs and obstacles. Also, it must be capable to attract UAVs to an eventual motion goal, and provide a self-optimizing and autonomous behavior, which can be found in nature, e.g. a flock of birds or a school of fish.

2 Preliminaries

In the previous research of the MRS group, swarming simulations and system verifications were performed using tools or platforms like *Matlab* [36][37], *SyRoTek*[1] [31][38] or *V-REP*[2] [29][39]. However, implementation of methods presented in this thesis are performed with modern, progressive tools - Robot Operating System and Gazebo simulator. These tools are nowadays hugely popular in the robotic community for their rapidly evolving functionality. Combined, they allow us take our model, implement it, test it, simulate it and apply it to real hardware with relatively ease to use, robust and reliable behavior. A huge advantage of these tools is also their open-source status, which allows the best teams around to world to contribute their work and provide better, more expertise robotic development.

2.1 Robot Operating System

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.[3]

Source code in ROS is organized via *packages*. A package is an atomic unit of ROS software and can represent ROS nodes, a ROS-independent library, a dataset, configuration files, a third-party piece of software, or anything else that logically constitutes a useful module.[4] Packages also work great with low-level build system infrastructure tool *catkin*.[5]

ROS developers can choose from an extensive set of modern programming languages. Supported development languages are primarily Python, C/C++ and Lisp, but libraries for Java, JavaScript or Lua are currently being tested as well.

Running ROS starts with launching of a ROS Master, which provides naming and registration for ROS nodes. The ROS node is a computational process usually representing one part of a robotic system, e.g. software for a simple robot manipulator could contain one ROS node for control of its joint motors and second one for a sensor data evaluation. The block diagram on Fig. 2 illustrates the core of ROS naming and communication.

Concept of ROS allows every node to transfer messages to other nodes with usage of so called *topics* or *services*. A topic represents an unidirectional message bus, defined by its name and message type. Nodes are not aware, who they are communicating with, therefore they are broadcasting the data through the topics. The topics are divided to subscribers, which listen to the topic with a specific name and acquire data published to it by the second type - publishers. There can be multiple publishers and subscribers to a topic. Block diagram on Fig. 3 illustrates one publisher, publishing to a topic defined by its name "/example" and message type "std_msgs/String", which is subscribed by

[1] System for robotic e-learning `https://syrotek.felk.cvut.cz/`
[2] Virtual Robot Experimentation Platform `http://www.coppeliarobotics.com/`
[3] Definition from official site `http://wiki.ros.org/`.
[4] Further information at `http://wiki.ros.org/Packages`
[5] Further information at `http://wiki.ros.org/catkin`

at least two subscribers. On the other hand, services are used for communication with a request/reply interactions.

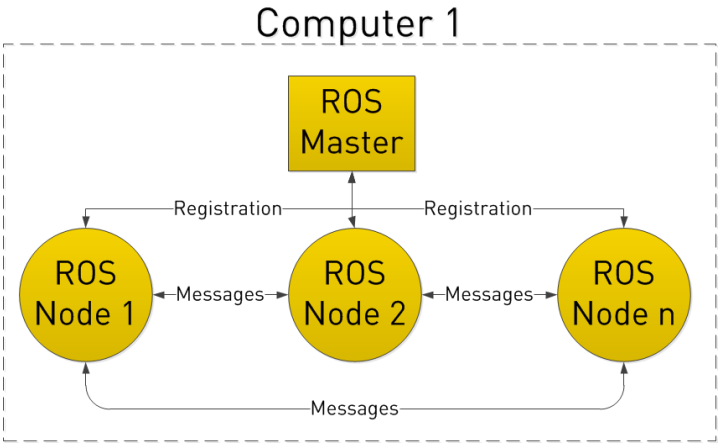


Figure 2 Fundamental communication between ROS Master and nodes [40].

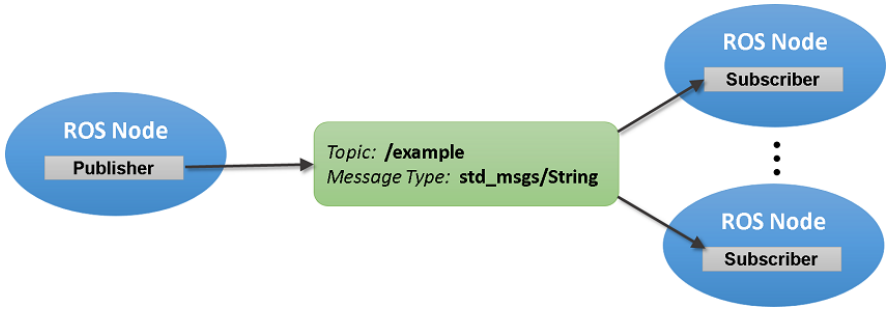


Figure 3 Concept of publishers and subscribers [41].

The MRS group at FEE CTU in Prague, with the help of University of Pennsylvania and University of Lincoln, had developed UAV controllers for the MBZIRC 2017 competition in ROS, version *Indigo*. These controllers are preliminary for implementation and simulation of proposed swarming approaches as it represents base structures to be build on.

The topics are part of a communication model used for simulations and real experiments in this thesis. The communication model, illustrated on Fig. 4, represents an UAV running its controllers, which send and acquire messages via topics. Grey boxes in the communication model on Fig. 4 represent parts of SITL[6], which can be classified as black boxes for the swarming controller proposed in this thesis. The blocks illustrated on Fig 4 represent

- PennController - A controller developed by University of Pennsylvania, which utilizes PixHawk autopilot, presented in section 2.3. The controller utilizes the kinematic and dynamical model of the UAV to provide motion commands to control the flight of the helicopter.
- MPC[7] - A lightweight embedded system for stabilization and control of UAVs. The system utilizes predictive controllers to find optimal control actions for the aircraft using only onboard computational resources [33].

[6]Software in the Loop
[7]Model Predictive Control

- Odometry - Estimation of current position in a coordination system by usage of the data from onboard sensors and the MPC.
- Swarming controller - A controller of a swarming behavior of an UAV, proposed in this thesis. The controller is based on fusion of data from its own Odometry and relatively localized neighborhood. The detailed model is illustrated on Fig. 27.
- Other UAVs - All other UAVs in the coordination system, capable to be part of a swarm. The relative localization is used for an estimation of their relative position by the swarming controller.
- Obstacles - All the obstacles in the coordination system, which are given to all UAVs as an input. The definition of an obstacle is proposed in section 3.2.

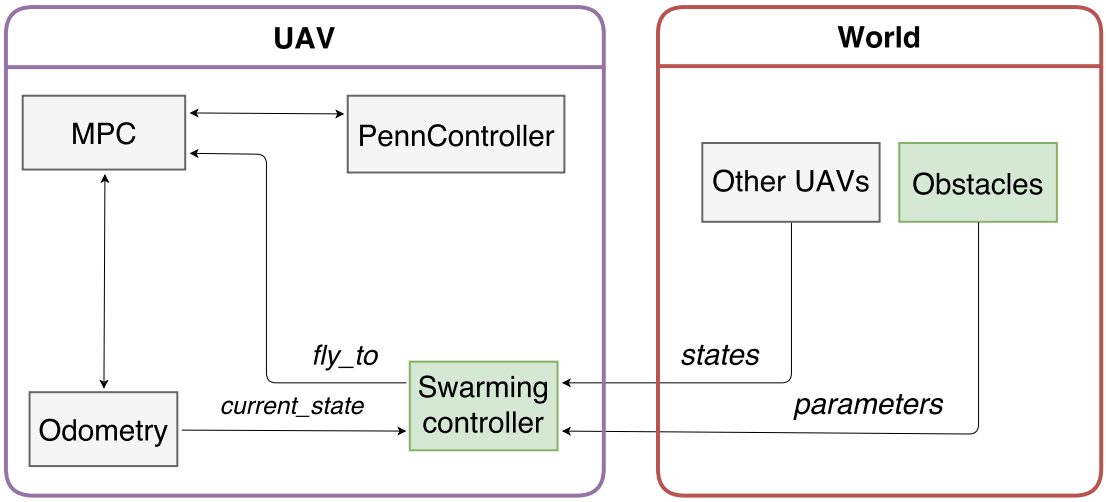


Figure 4 Block diagram of message flow of an UAV in a swarm.

Huge advantage of ROS is also its independence on the machine it is running on or even system architecture. Therefore communication is possible between different platforms, e.g. UNIX system, Arduino and Android. A simulation performed in ROS and Gazebo simulator with a higher number of UAVs demands bigger computational resources. By sharing the communication in a local network, the computational power can be divided onto more computers.[8] By doing so, it is possible for simulations to have parts of the simulated system spread to more machines, thus providing computational resources for more complex simulations. Using this functionality, simulations in sections 4 and 6 had been performed.

[8]Further information at `http://wiki.ros.org/ROS/NetworkSetup`.

2.2 Gazebo

Robot simulation is an essential tool in every roboticist's toolbox. A well-designed simulator makes it possible to rapidly test algorithms, design robots, perform regression testing, and train AI system using realistic scenarios. Gazebo offers the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. At your fingertips is a robust physics engine, high-quality graphics, and convenient programmatic and graphical interfaces. Best of all, Gazebo is free with a vibrant community.[9]

Figure 5 Logo of the Gazebo simulator.

Gazebo simulator has been developed since 2002 at the University of Southern California, USA. Over the years, its development led to integration of ROS and PR2 into it, which has since become one of the primary tools used in the ROS community.

It allows its users to start an interface to verify systems functionality, develop its own worlds and advanced multi-component models, record data and many more. All with usage of physics engines *ode*, *bullet*, *dart* or *simbody*.

Figure 6 Screenshot of an UAV flying in front of a model of house in Gazebo.

[9]Definition from official site `http://gazebosim.org/`.

2.3 UAV platform

Simulations and real experiments require a set of helicopters. A platform of helicopters have again been developed by the MRS group of CTU in Prague Faculty of Electrical Engineering, with help of University of Pennsylvania and University of Lincoln. Fig. 7 shows the hardware equipment of an helicopter of the MRS group.

- Frame - Commercial DJI F550 hexacopter.
- PixHawk - Industry standard autopilot (control unit).
- Intel NUCi7 - Powerful onboard PC running the ROS.
- 3DR uBlox GPS + Tersus GNSS system - Absolute localization system for outdoor environment.
- Terraranger One - External altitude sensor for application, where a knowledge of precise distance to surface is required.
- Mobius ActionCam - Horizontal camera, which could be in a future work used for relative visual onboard localization, described in section 3.4, to localize the neighborhood of the UAV.
- mvBlueFOX - Vertical camera used for capturing of surface below the UAV.

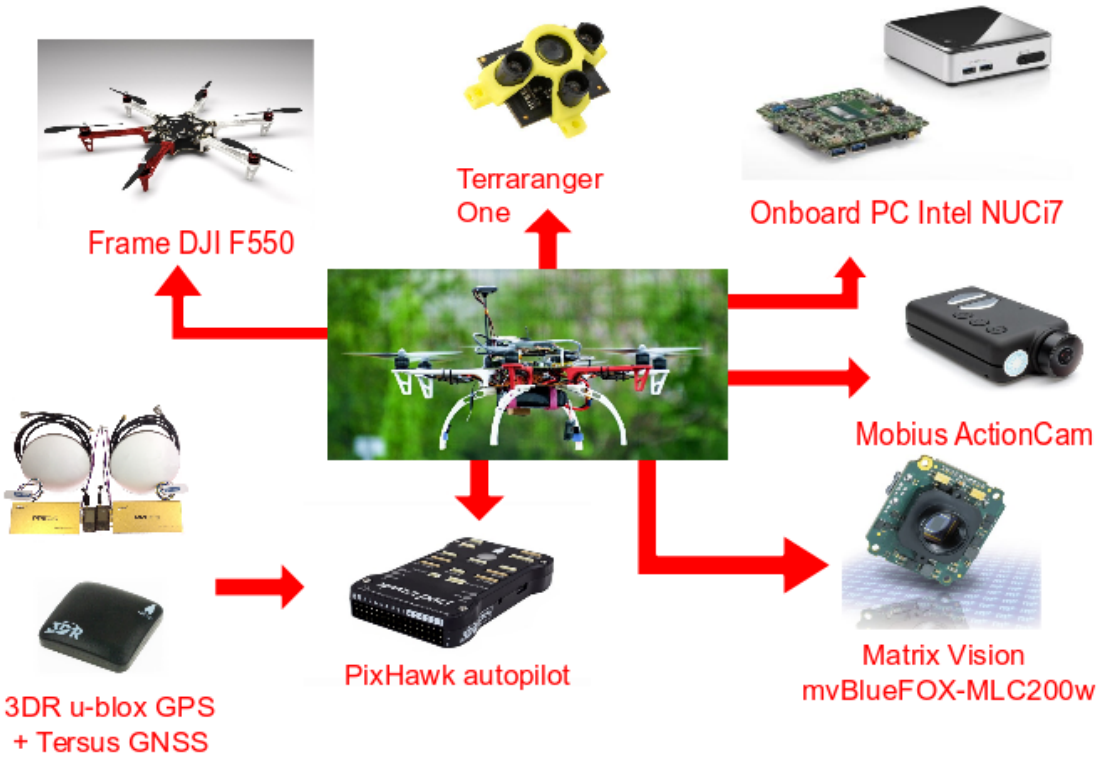


Figure 7 Hardware of a single helicopter of the MRS group at FEE CTU.

3 Flocking

In artificial intelligence, the term *flocking* describes a nature inspired behavior found among species of birds, fish, insects or even among the human crowds. This behavior might seem to be random, yet scientists have developed flocking models by analyzing individual movements and reactions to other individuals in a flock, school, swarm, herd or crowd. Further in this thesis, a group of helicopters or commonly a group of any individuals will be called a *swarm*.

Figure 8 A common starling (Sturnus vulgaris) flock in Lauwersmeer, the Netherlands [42].

The main attribute of a swarm is uniformed behavior for any number of individuals. This limitlessness is observed in nature, since fish schools with several kilometers of length and millions of individuals can be tracked.

The flocking behavior is based on movements and individual reactions of at least two subjects, each representing one member of a swarm. In the robotic context, swarming algorithms are used for stabilization and control of groups of

- Unmanned Ground Vehicles (UGV),
- Unmanned Aerial Vehicles (UAV) and
- Unmanned Underwater Vehicles (UUV).

This thesis analyses behavior for Unmanned Aerial Vehicles in planar (a flight with a constant altitude) and 3D space.

Usefulness for swarms of helicopters can be found in relative control and safety assurance for higher number of helicopters, each independently acting on its own by processing data from build-in sensors and without publishing any information to its surroundings.

Huge advantage for a controlled swarm in robotic context is decentralization, which is a process of dividing the computing power to each individual. The decentralization makes the swarm robust, immune to central control unit failure, independent on a

SPOF[1] of the swarm members and mainly eliminates the requirement of communication between the central control unit and other members of the swarm. The swarm control brings up more advantages such as safe motion of the whole swarm in space, where there could be free or unspecified obstacles present, or an ability to react on dangerous subjects invading their nearby space, called *predators* in [31] and [39]. On the other hand, disadvantages of the swarm control are mainly nondeterministic behavior and unoptimized motion, while following a trajectory or flying to a specified goal.

Control of a swarm without the requirement to publish any data is an ideal state of the application, where the vehicles do not require communication modules. This ideal state is achievable with relative onboard localization of neighbors, described in 3.4.

Many of the flocking algorithms are based on three *flocking rules* (or further called *flocking forces*) defined by Craig W. Reynolds in 1987 [1]. The *flocking rules* were introduced in [1] as

- Collision Avoidance: avoid collisions with nearby flockmates,
- Velocity Matching: attempt to match velocity with nearby flockmates,
- Flock Centering: attempt to stay close to nearby flockmates.

These forces are also known as *Separation*, *Alignment* and *Cohesion*.

3.1 Boids

The expression *Boids* is originally a name of simulation algorithm enclosed to the Reynold's article [1]. The name was inherited and nowadays it represents a core of platform-independent flocking algorithms, which then expand the Boids with other behavioral patterns, like obstacle avoidance.

The problem with Boids is its non-clear interpretation, where only the flocking rules and their natural origin is defined. Given these definitions, the rules can be interpreted numerous ways and with or without the restrictions of the relative localization.

Before the closer look at the flocking rules, we define $\vec{s}_0$ to be the state of a helicopter h_0, then

$$\vec{s}_0 = [\vec{p}_0,\ \vec{v}_0,\ \vec{a}_0]^T, \tag{1}$$

$$\vec{p}_0 = [x_0,\ y_0,\ z_0]^T, \tag{2}$$

$$\vec{v}_0 = \dot{\vec{p}}_0 = [v_0^x,\ v_0^y,\ v_0^z]^T, \tag{3}$$

$$\vec{a}_0 = \dot{\vec{v}}_0 = [a_0^x,\ a_0^y,\ a_0^z]^T, \tag{4}$$

where $\vec{p}_0$ is absolute position of the center of mass, $\vec{v}_0$ is the velocity, and $\vec{a}_0$ is the acceleration of helicopter h_0.

In section 1.2 vector $\vec{h}_0$ was defined. Lets extend the definition with an index $j \in \{1, \ldots, n\}$ specifying j-th UAV from vector $\vec{h}_0$ of relatively localized UAVs by the UAV h_0.

[1] Single Point of Failure

3.1.1 Collision Avoidance

Animals separate from each other for clear purposes, to not hurt them or other individual, therefore the collision avoidance force, known as the *separation*, is necessary to maintain a safe service of the swarm.

A simple form of the separation force, as defined in [1], is described as

$$\vec{f}_s = \frac{1}{n}\sum_{j=1}^{n}\vec{f}_{sj} = \frac{1}{n}\sum_{j=1}^{n}\Omega_j\frac{\vec{p}_0 - \vec{p}_j}{\|\vec{p}_0 - \vec{p}_j\|}, \tag{5}$$

where $\Omega_j = r - \|\vec{p}_0 - \vec{p}_j\|$.

To ensure safe, fluent and variable motion of the swarm, the separation force $\vec{f}_s$ from Eq. (5) must be weighted accordingly to ensure that no helicopters collide. The separation force with weighted function, which assures that a distance between helicopters is maintained, is in this work designed as

$$\vec{f}_s = \frac{1}{n}\sum_{j=1}^{n}\vec{f}_{sj} = \frac{1}{n}\sum_{j=1}^{n}\epsilon_j(\vec{p}_0 - \vec{p}_j), \tag{6}$$

where ϵ_j is coefficient defined as

$$\epsilon_j = \begin{cases} k_1\left(\dfrac{\sqrt{\|\vec{p}_0 - \vec{p}_j\|}}{\|\vec{p}_0 - \vec{p}_j\|} - \dfrac{\sqrt{d}}{d}\right), & if \ \ \|\vec{p}_0 - \vec{p}_j\| <= d \\ 0, & if \ \ \|\vec{p}_0 - \vec{p}_j\| > d. \end{cases} \tag{7}$$

where $k_1 > 0$ is coefficient affecting the steepness of the separation curve and $d \in (0,\ r\rangle$ (m) defines a distance, where the separation force stops affecting the behavior. Fig. 9 illustrates dependence of the two coefficients k_1 and d on the weighting coefficient ϵ.

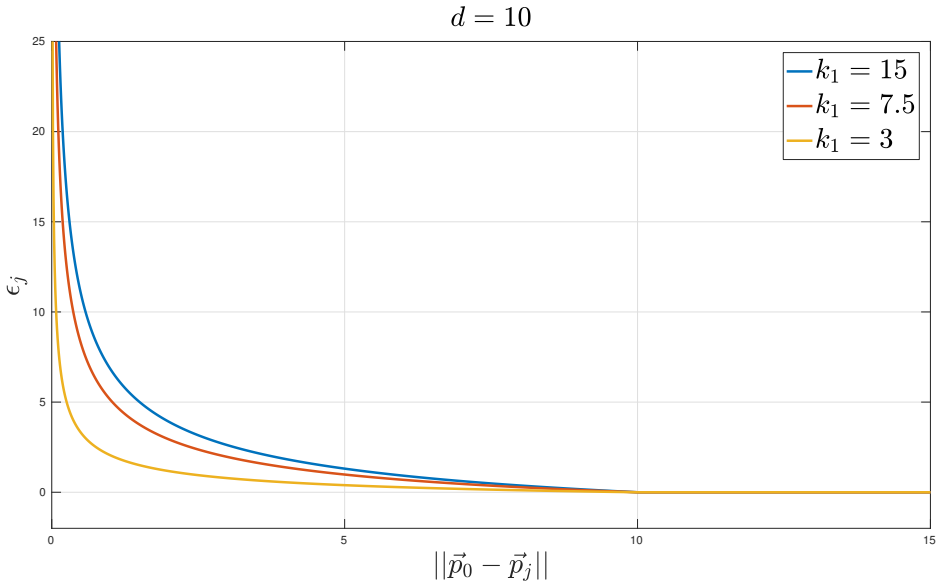
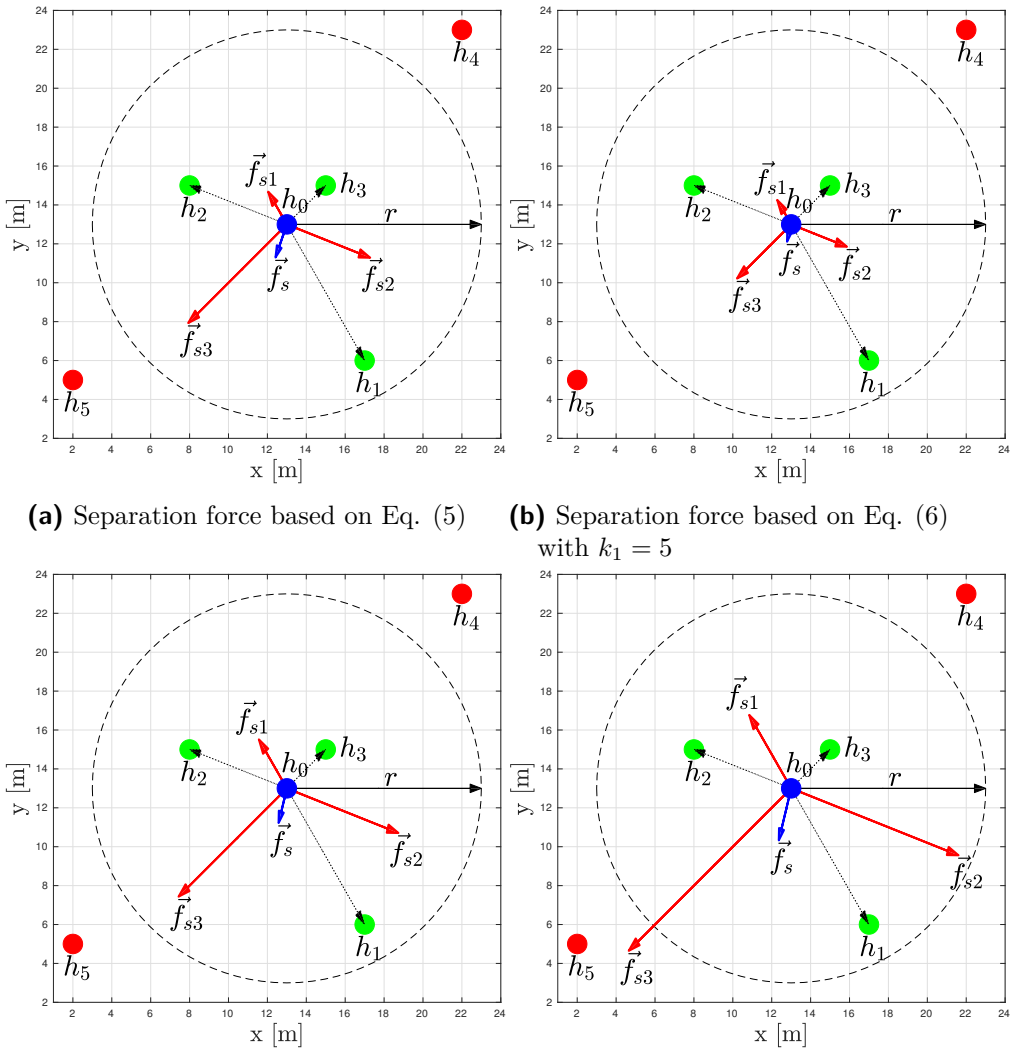


Figure 9 Dependency of the coefficient ϵ_j on a distance between two UAVs h_0 and h_j.

Fig. 10a shows the separation force $\vec{f}_s$ defined in Eq. (5). Separation forces from Fig. 10b, 10c and 10d are based on Eq. (6), where $d = r$, and show the dependence on the coefficient k_1.



(a) Separation force based on Eq. (5)

(b) Separation force based on Eq. (6) with $k_1 = 5$

(c) Separation force based on Eq. (6) with $k_1 = 10$

(d) Separation force based on Eq. (6) with $k_1 = 15$

Figure 10 The comparison of the separation forces from Eq. (5) and (6).

3.1.2 Velocity Matching

The velocity matching, known as the *alignment*, ensures fluent motion of the swarm by navigating the individual to the average velocity of its neighbors. The most simple and sufficient form of the alignment force is given by formula

$$\vec{f}_a = \frac{1}{n}\sum_{j=1}^{n}\vec{f}_{aj} = \frac{1}{n}\sum_{j=1}^{n}\vec{v}_j. \tag{8}$$

The velocity matching is interpreted as the arithmetic mean of the velocities of the localized UAVs. Fig. 11 displays, how the UAV control mechanism uses the velocities of the localized neighbor UAVs to calculate the alignment force. Using relative onboard localization, the velocity of an UAV h_j in a simulation step t could be obtained as

$$\vec{v}_j[t-1] = \vec{p}_j[t] - \vec{p}_j[t-1]. \tag{9}$$

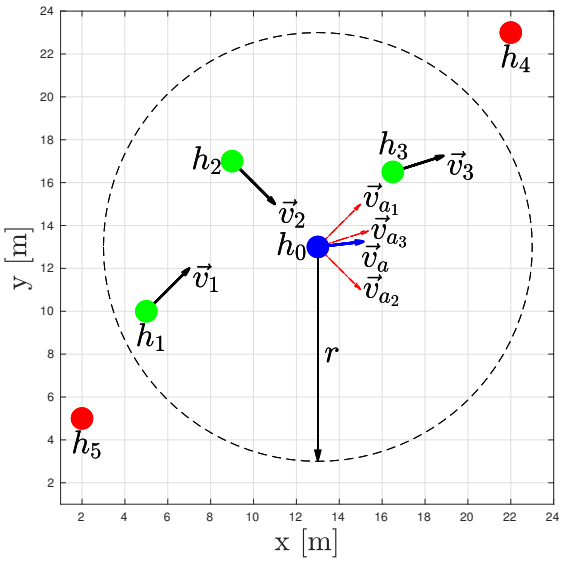


Figure 11 Illustration of the alignment force from Eq. (8).

3.1.3 Flock Centering

The flock centering, known as the *cohesion*, of an animal flock was proven to statistically increase chance of surviving for an individual in situations, where the flock is being attacked by a predator. Also it helps the animals socialize and to find a mating partner. The cohesion force provides a way to keep the swarm united, by heading the helicopter to the center of the nearby swarm members.

The most simple and sufficient form of the cohesion force is given by formula

$$\vec{f_c} = \frac{1}{n}\sum_{j=1}^{n}\vec{f_{cj}} = \frac{1}{n}\sum_{j=1}^{n}(\vec{p_j} - \vec{p_0}). \tag{10}$$

Fig. 12 shows, that the cohesion force is heading into the center of gravity of a sub-swarm formed from the relatively localized neighbors by the UAV h_0.

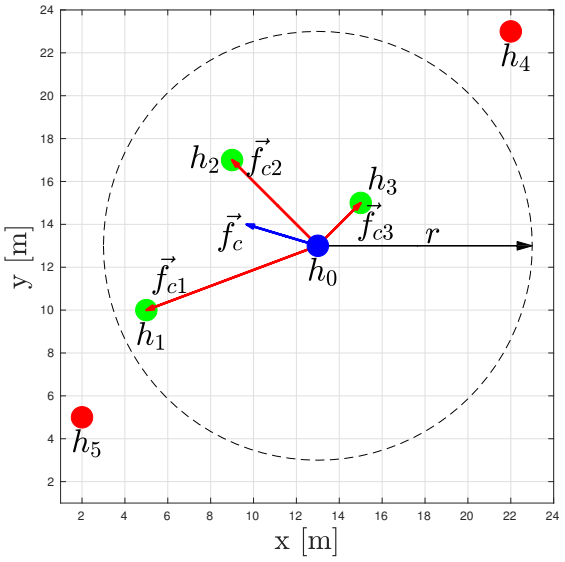


Figure 12 Illustration of the cohesion force from Eq. (10).

3.1.4 Combined forces

Specified flocking behavior is based on the combination of the flocking rules. For fluent motion of the swarm, we consider the velocity of the controlled UAV h_0. Boids behavior in one simulation step t for each helicopter in the swarm is given as

$$\vec{v}_0[t] = \vec{f}_{steer} + \vec{v}_0[t-1], \tag{11}$$

$$\vec{p}_0[t] = \vec{p}_0[t-1] + \vec{v}_0[t], \tag{12}$$

where

$$\vec{f}_{steer} = \vec{f}_s + \vec{f}_a + \vec{f}_c - \vec{v}_0[t-1]. \tag{13}$$

Using this model, $\vec{f}_{steer}$ represents a steering force, which is applied on the current velocity $\vec{v}_0[t-1]$ to acquire position $\vec{p}_0[t]$. The current velocity is respected, therefore motions are more fluent and the flock motion seems more natural.

Final example of the combination of the flocking rules as presented by Reynolds in [1], with edited equation for the separation force is displayed on Fig. 13. The velocity $\vec{v}_0[t]$ of the UAV h_0 in step t on Fig. 13 suggests that the algorithm may tend to $\alpha - lattices$ defined in [43].

The $\alpha - lattices$ are an arrangement of the swarm, where the UAVs tend to form an optimal formation in an environment without obstacles due to effort to minimize the distance between the closest UAVs. In planar space, the $\alpha - lattices$ form a grid formed from an equilateral triangles with an edge-length ξ. In free 3D space, the $\alpha - lattices$ form a "crystal" structure from a triangular prisms with an edge-length ξ. The $\alpha - lattices$ in a planar space are illustrated on Fig. 14a and in free 3D space on Fig. 14b. The edge-length ξ expresses an equilibrium point between the separation and cohesion force and its value is dependent on the parameters of the swarming behavior (primarily parameters of the separation force).

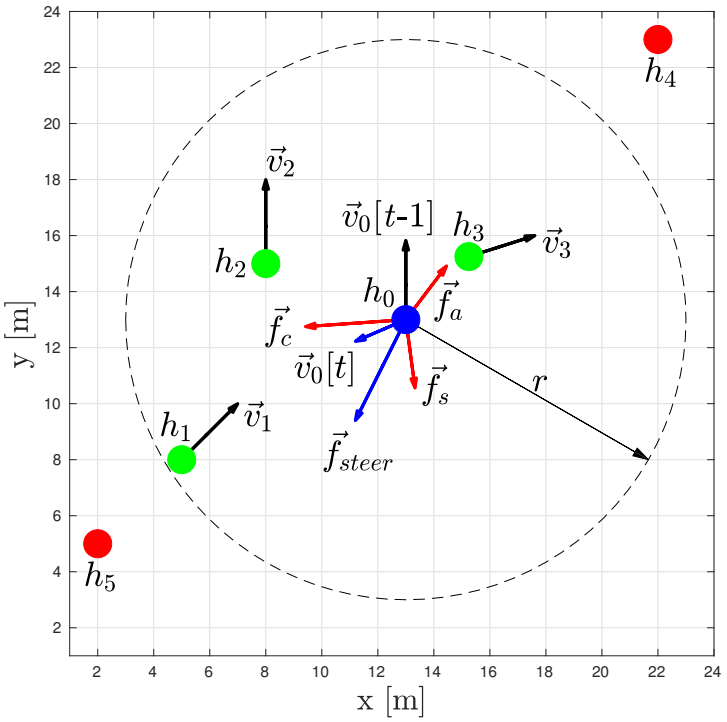
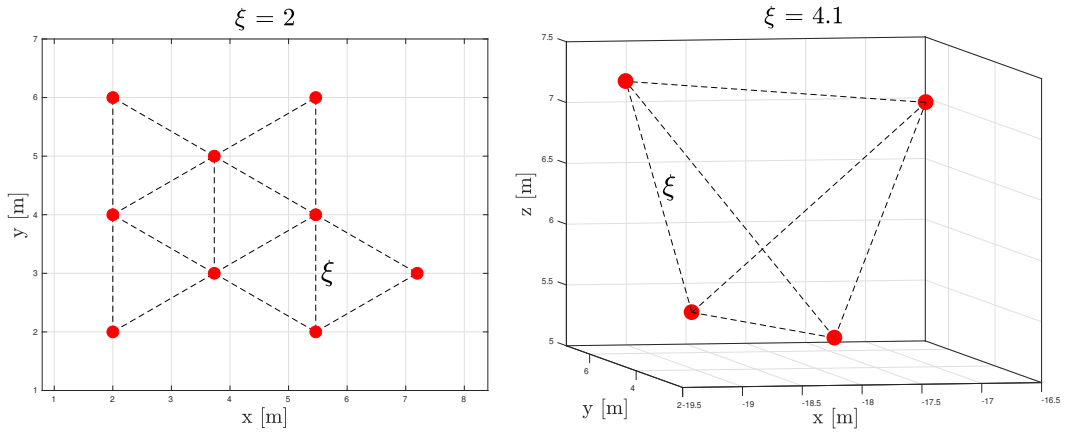


Figure 13 Calculating of the velocity of the UAV h_0 in a simulation loop t according to Eq. (11).



(a) Free planar space. **(b)** Free 3D space.

Figure 14 Illustration of the optimal formations ($\alpha - lattices$) for a swarm of UAVs in planar and 3D space.

3.2 Obstacle avoidance

Obstacle avoidance ensures safe movement of a swarm through an environment with obstacles. For real UAVs, different types of obstacles need to be avoided - trees, walls, fences, lamps, buildings, wires, etc. This thesis is restricted on cylindrical obstacle, representing i.e. a tree or a lamp, and walls. Those two obstacle types were chosen, because they are the base for modeling of further complex environments. Such complex environments could be e.g. a forest or an office like environment. Obstacle avoidance can be designed using virtual UAVs, so called $\beta - agents$ in [43]. These virtual UAVs are created, whenever an obstacle is localized by an UAV using relative onboard localization. Each individual of a swarm localizes the obstacles around him, determines their type and for each of them creates a virtual UAV.

Define $\vec{O}_0 = [O_1, \ldots, O_m]$, where m is number of localized obstacles, to be a vector of localized obstacles by the UAV h_0. Also define $\vec{q}_0 = [q_1, \ldots, q_m]$ to be a vector of virtual UAVs for all obstacles from $\vec{O}_0$. Thus, in each simulation step a virtual UAV is computed $\forall O_j,\ j \in \{1, \ldots, m\}$. A virtual UAV q_j is defined by its state

$$\vec{s}_{q_j} = [\vec{p}_{q_j},\ \vec{v}_{q_j}]^T, \tag{14}$$

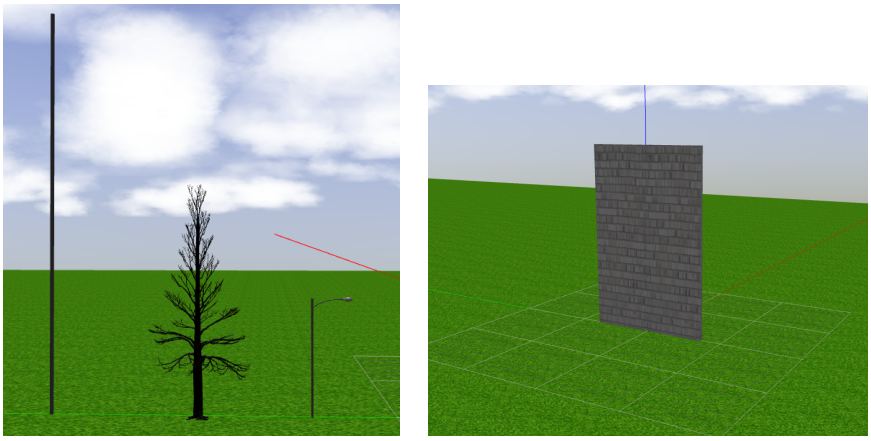
where $\vec{p}_{q_j}$ is position

$$\vec{p}_{q_j} = [x_{q_j},\ y_{q_j},\ z_{q_j}]^T, \tag{15}$$

and $\vec{v}_{q_j}$ velocity

$$\vec{v}_{q_j} = [v^x_{q_j},\ v^y_{q_j},\ v^z_{q_j}]^T. \tag{16}$$

Computing the state of each virtual UAV depends on the obstacle type, therefore the UAV must be capable of recognizing the type of the localized obstacle. After determining the virtual UAVs states, a similar approach, as for computing the flocking rules (specifically the separation and alignment forces defined in sections 3.1.1 and 3.1.2), is used to evaluate the obstacle avoidance force. The obtaining method for a virtual UAV state is based on [43], where the method is supported with a mathematical proof. Fig. 15 illustrates a model of a cylindrical and a wall obstacle in the Gazebo simulator, and Fig. 16 illustrates those two types designed for experiments in a real environment.



(a) Cylinder **(b)** Wall

Figure 15 Illustration of common obstacle types in Gazebo simulator.

(a) Cylinder

(b) Wall

Figure 16 Illustration of the obstacle models used for real experiments.

3.2.1 Cylindrical obstacle

Cylindrical obstacle O_c is defined by its base center $\vec{c}_c = [c_c^x, c_c^y]^T$, radius R_c and height H_c. The state of a virtual UAV $q_j \in \vec{q}_0$ for cylindrical obstacle O_c and UAV h_0 is illustrated on Fig. 17. If condition

$$H_c + z_{off} < z_0 \tag{17}$$

is fulfilled, then the obstacle O_c is ignored, since the UAV overflies it. The height offset z_{off} ensures that the UAV is truly located over the obstacle and depends on the UAV dimensions, accuracy of relative localization of the obstacle, accuracy of the altitude sensor, oscillation of the motor regulators and control delays. Otherwise, the state is computed for a virtual UAV $q_j \in \vec{q}_0$ and the UAV h_0 as

$$\vec{p}_{q_j} = \mu\vec{p}_0 + (1-\mu)\vec{c}_c, \tag{18}$$
$$\vec{v}_{q_j} = b_1\mu\mathbf{P}\,\vec{v}_0, \tag{19}$$

where

$$\mu = \frac{R_c}{\|\vec{p}_0 - \vec{c}_c\|}, \tag{20}$$
$$\mathbf{P} = \mathbf{I} - \vec{n}\,\vec{n}^T, \tag{21}$$
$$\vec{n} = \frac{\vec{p}_0 - \vec{c}_c}{\|\vec{p}_0 - \vec{c}_c\|}, \tag{22}$$

$b_1 > 0$ is a weighting coefficient, $\vec{n}$ is a *normal unit*, $\mathbf{I}$ is an *identity matrix* and $\mathbf{P}$ is a *projection matrix*.

The position, defined in Eq. (18), of the virtual UAV is the closest point on the obstacle border between the UAV and the obstacle center. The velocity, defined in Eq. (19), is then provided as rhe perpendicular vector to the directional vector between the UAV and the obstacle to ensure a smooth obstacle bypass.

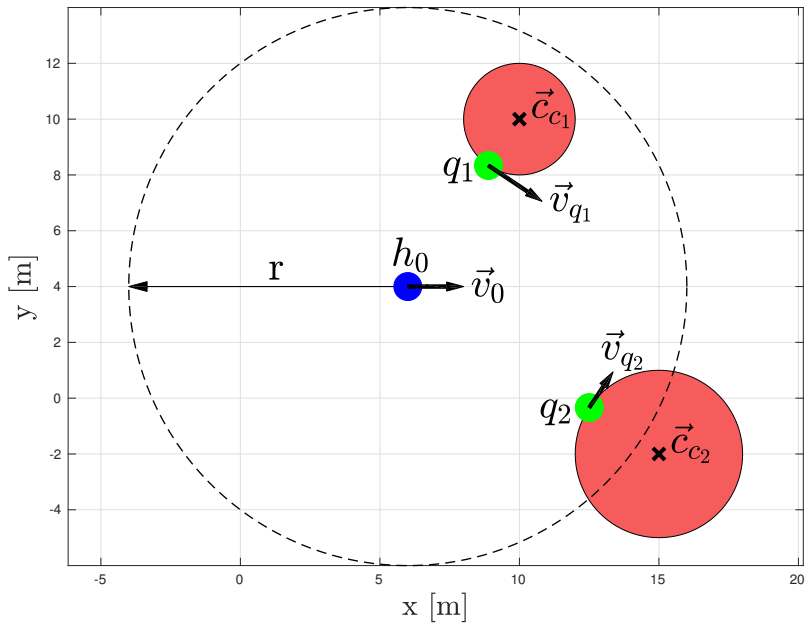


Figure 17 Illustration of the virtual agents for two cylindrical obstacles.

3.2.2 Wall obstacle

Assume a homogeneous wall obstacle O_w with negligible width d_w. Such wall obstacle is defined by its unit normal vector $\vec{n}_w = [n_w^x, n_w^y]^T$, its center of gravity $\vec{c}_w = [c_w^x, c_w^y, c_w^z]^T$, its length $l_w \gg d_w$ and finally height H_w. The state of a virtual UAV $q_j \in \vec{q}_0$ for a wall obstacle O_w and UAV h_0 is illustrated on Fig. 19.

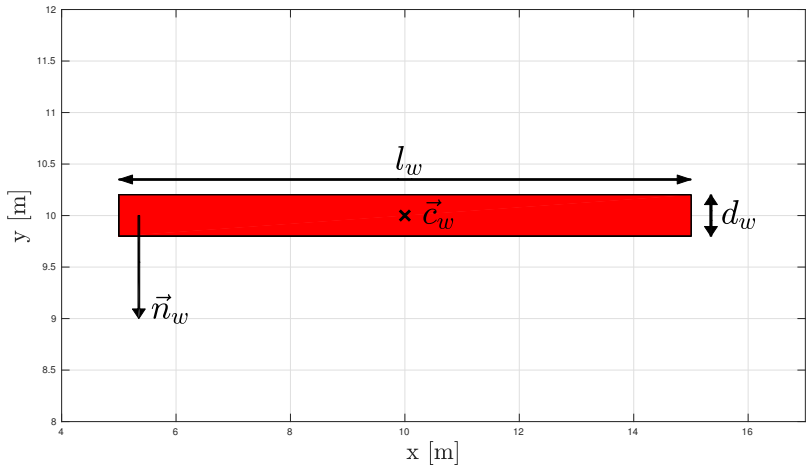


Figure 18 Definition of a planar wall obstacle.

As for the cylindrical obstacle, the wall obstacle is ignored, if the UAV altitude is higher than the wall height. Therefore, the wall obstacle is ignored if condition

$$H_w + z_{off} < z_0 \tag{23}$$

is met.

The state of a virtual UAV $q_j \in \vec{q}_0$ for wall obstacle O_w and UAV h_0, illustrated on Fig. 19, is computed as

$$\vec{p}_{q_j} = \mathbf{P}(\vec{p}_0 + \vec{p}_{off}) + (\mathbf{I} - \mathbf{P})\vec{c}_w, \tag{24}$$

$$\vec{v}_{q_j} = b_2 \mathbf{P} \vec{v}_0, \tag{25}$$

where $b_2 > 0$ is a weighting coefficient, $\mathbf{I}$ is an *identity matrix*, $\mathbf{P} = \mathbf{I} - \vec{n}_w \vec{n}_w^T$ is a *projection matrix* and $\vec{p}_{off}$ is an offset in the position of the virtual UAV ensuring that its position does not exceed the obstacle dimensions. The figure 19 also colors a strip defined as a set of points in a plane $z = z_0$, which are located between two parallel lines, both with a normal unit $\vec{n} = \vec{u}_w$, where $\vec{u}_w$ is the directional vector ($\vec{n} \cdot \vec{u}_w = 0$) of the wall O_w. Those lines pass, as illustrated on Fig. 19, through the edges of the wall, while the distance between those lines is equal to l_w. The vector $\vec{p}_{off}$ value is defined by the position of the UAV and is designed as

$$\vec{p}_{off} = \begin{cases} \vec{0}, & \text{if } \vec{p}_0 \text{ is in the colored stripe,} \\ \\ \vec{p}_w - \vec{p}_0, & \text{otherwise.} \end{cases} \tag{26}$$

The vector $\vec{p}_w = min\{distance(h_0,\ O_w)\}$ represents the closest point on the wall border to the UAV h_0.

The position of the virtual UAV, defined in Eq. (24), is the closest point on the wall border to the UAV. The velocity, defined in Eq. (25), is then provided as the perpendicular vector to the normal vector of the wall obstacle to ensure a smooth obstacle bypass using the alignment force.

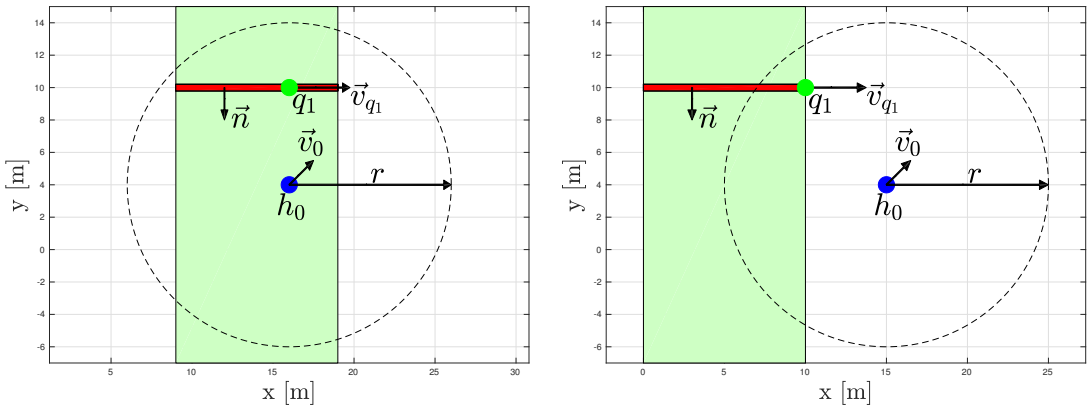


Figure 19 Illustration of the virtual agents for a wall obstacle.

3.2.3 Obstacle avoidance force

The obstacle avoidance force is a combination of previously defined flocking rules - separation (3.1.1) and alignment (3.1.2), however with slight modifications. The obstacle avoidance force is defined for an UAV h_0 as

$$\vec{f}_{oa} = \vec{f}^{\,s}_{oa} + \vec{f}^{\,a}_{oa} = \frac{1}{n}\sum_{j=1}^{n}(\vec{f}^{\,sj}_{oa} + \vec{f}^{\,aj}_{oa}), \tag{27}$$

$$\vec{f}_{oa} = \frac{1}{n}\sum_{j=1}^{n}\sigma_j(\vec{p}_0 - \vec{p}_{q_j}) + \frac{1}{n}\sum_{j=1}^{n}\tau_j\vec{v}_{q_j}, \tag{28}$$

where n is the number of localized obstacles by the UAV h_0, σ_j is the weighting coefficient and $\tau_j \in \{0, 1\}$ is defined further. The weighting coefficient σ_j is defined as

$$\sigma_j = \begin{cases} k_2\left(\dfrac{\sqrt{\|\vec{p}_0 - \vec{p}_{q_j}\|}}{\|\vec{p}_0 - \vec{p}_{q_j}\|} - \dfrac{\sqrt{w}}{w}\right), & if \;\; \|\vec{p}_0 - \vec{p}_{q_j}\| <= w, \\ 0, & if \;\; \|\vec{p}_0 - \vec{p}_{q_j}\| > w. \end{cases} \tag{29}$$

where $k_2 > 0$ is the steepness coefficient and $w \in (0,\ r\rangle$ (m) defines a distance, where the obstacle avoidance separation force stops affecting the behavior. The coefficient w, whose significance is illustrated on Fig. 22, is defined to differentiate safe distance in relationship UAV-obstacle and UAV-UAV. The coefficients w and k_2 were defined to distinguish the safety distance between two UAVs and between an UAV and an obstacle.

In the next section 3.3 a motion goal will be presented. The motion goal modifies behavior of any UAV by removing the alignment force from the Eq. (28) in a certain case. In case of an UAV h_0 has a motion goal G_0 set, the $\tau_j = 0\ \forall j \in \{1, \ldots, m\}$ if and only if

$$k(\vec{p}_0, O_j) = k(\vec{p}_{G_0}, O_j), \tag{30}$$

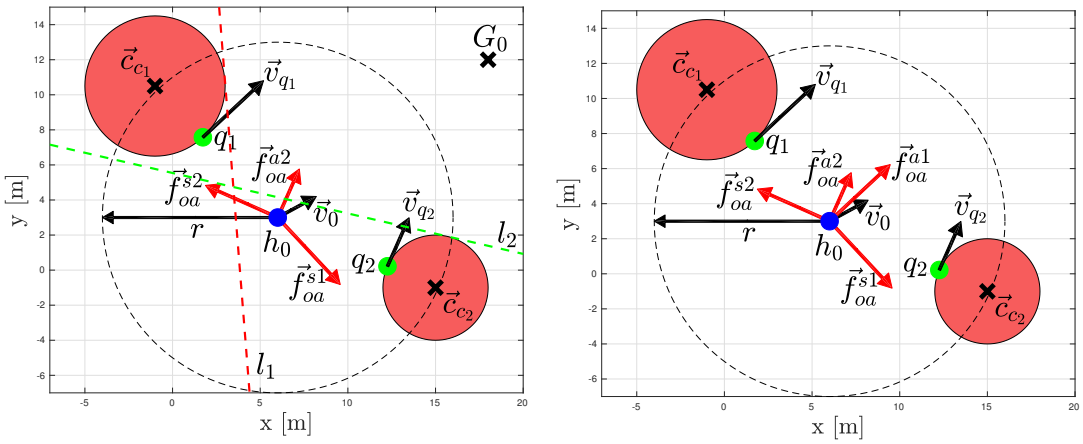
otherwise $\tau_j = 1$. Function k in the Eq. (30) is defined as

$$k(\vec{p}, O_j) = (x_B - x_A)(y_p - y_A) - (y_B - y_A)(x_p - x_A), \tag{31}$$

where $\vec{p} = [x_p, y_p, z_p]^T$ and $\vec{A} = [x_A, y_A]^T$, $\vec{B} = [x_B, y_B]^T$ are two different points laying on a line l_j. The line l_j is defined for the types of obstacle as

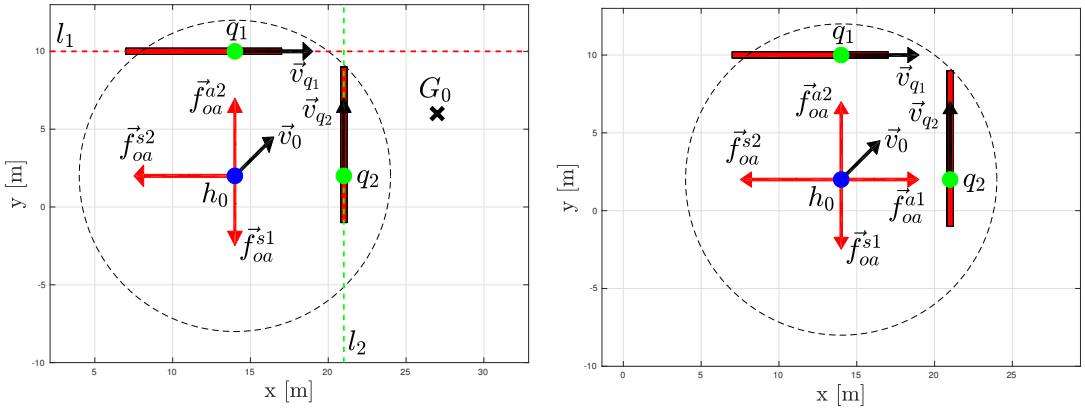
- **cylindrical**: a perpendicular line to vector $\overrightarrow{c_c G_0}$ going through the closest point to the goal on the cylindrical obstacle border (illustrated on Fig. 20a),
- **wall**: a line passing through the whole length of the wall obstacle with its directional vector equal to the normal vector n_w of the wall (illustrated on Fig. 21a).

Equation (30) verifies if the position of the UAV h_0 and its motion goal G_0 lay on the same side of the line l_j in a plane $z = z_0$. Result is, as illustrated on Fig. 20 and 21, that the velocity of a virtual UAV is not taken into account, when an obstacle is already bypassed. Therefore the UAV does not falsely evaluates obstacle avoidance velocity as valid. In situations illustrated on Fig. 20a and 21a, the coefficient $\tau_1 = 0$ and $\tau_2 = 1$, since the UAV h_0 is located on the same side of the line l_1 as the motion goal G_0.



(a) UAV h_0 has a motion goal G_0 ($\vec{f}^{a1}_{oa} = \vec{0}$). **(b)** UAV h_0 has no motion goal set.

Figure 20 Illustration of the obstacle avoidance forces for a cylindrical obstacle.



(a) UAV h_0 has a motion goal G_0 ($\vec{f}^{a1}_{oa} = \vec{0}$). **(b)** UAV h_0 has no motion goal set.

Figure 21 Illustration of the obstacle avoidance forces for a wall obstacle.

Addition of the obstacle avoidance force to the flocking rules, as defined in section 3.1.4, results to calculating of the final force of an UAV h_0 in time (or simulation step) t as

$$\vec{v}_0[t] = \vec{f}_{steer} + \vec{v}_0[t-1], \tag{32}$$

$$\vec{p}_0[t] = \vec{p}_0[t-1] + \vec{v}_0[t], \tag{33}$$

where

$$\vec{f}_{steer} = \vec{f}_s + \vec{f}_c + \vec{f}_a + \vec{f}_{oa} - \vec{v}_0[t-1]. \tag{34}$$

The steering force $\vec{f}_{steer}$ needs to be applied on current velocity $\vec{v}_0[t-1]$ to acquire position $\vec{p}_0[t]$.

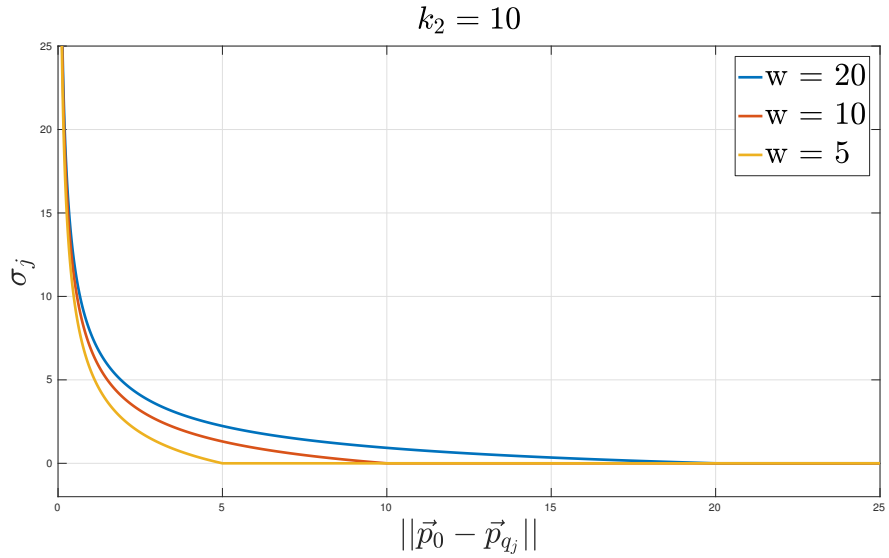


Figure 22 Dependence of the coefficient σ_j on distance between the UAV h_0 and a virtual UAV q_j.

3.3 Navigational force

The purpose of the swarm behavior is to ensure a safe movement of the swarm. A navigational force ensures, that the individuals navigate their movement to a motion goal. Without the navigational force the swarm flies out to a random direction. This flying off to a random direction is fully nondeterministic and is a result of the velocity matching (the alignment force). A state of a swarm in an environment without obstacles converges to a state, where the velocity of each swarm member is alike and the rest of the flocking rules cancel each other out. The force pushes each individual to a motion goal, which could differ for each individual or be common for the whole swarm.

Let define a motion goal G_0 of an UAV h_0 by its state

$$\vec{s}_{G_0} = [\vec{p}_{G_0},\ \vec{v}_{G_0}]^T, \tag{35}$$

$$\vec{p}_{G_0} = [x_{G_0},\ y_{G_0},\ z_{G_0}]^T, \tag{36}$$

$$\vec{v}_{G_0} = \dot{\vec{p}}_{G_0} = [v^x_{G_0},\ v^y_{G_0},\ v^z_{G_0}]^T, \tag{37}$$

where $\vec{p}_{G_0}$ is the position of the motion goal and $\vec{v}_{G_0}$ its velocity. We define motion goal to be either *static* or *dynamic*. The state of a static goal has a constant position and zero velocity over time. On the other hand, the states of a dynamic goal are defined by a trajectory $\vec{f}_{G_0}(t)$ dependent on time (or simulation step) t. Function $\vec{f}_{G_0}(t)$ is arbitrary as it represents any trajectory, which the helicopter should follow. It is expected to choose this dynamic function for applications of an object following or a space sweeping.

The navigational force should consider current state of an UAV h_0. Therefore we define the force, as presented in [43], for *static* goal as

$$\vec{f}_{n_0} = c_1(\vec{p}_{G_0} - \vec{p}_0) - c_2\vec{v}_0, \tag{38}$$

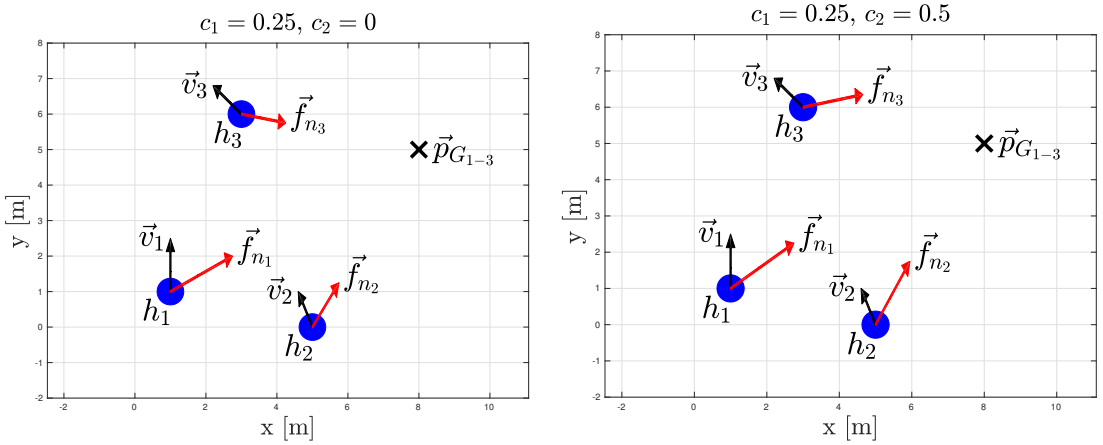
and for *dynamic* goal

$$\vec{f}_{n_0}(t) = c_1(\vec{p}_{G_0}(t) - \vec{p}_0) + c_2(\vec{v}_{G_0}(t) - \vec{v}_0), \tag{39}$$

where weighting coefficients c_1, $c_2 > 0$. Those coefficients should be chosen empirically to ensure that navigational force does not over-pushes other (importantly separation) forces and provides balanced navigation to the motion goal.

As defined, this force differs to previously presented forces. The force does not consider navigational forces of other UAVs in the swarm and is defined purely on the state of the UAV. The navigational functionality of the swarm should be researched as a trajectory planning for UAV swarms. The swarm navigation could be useful in many applications, however it is not part of this thesis.

The static motion goal is illustrated on Fig. 23. Fig. 23a illustrates navigational force from the UAV to the motion goal. It is clear that with constants $c_1 > 0$ and $c_2 = 0$ the navigational force is clearly weighted directional vector. For the static motion goal, it is not suitable to choose $c_2 > 0$, since the velocity of the UAV will change the direction of the navigational force outside of the motion goal.



(a) Zero velocity coefficient. **(b)** Nonzero velocity coefficient.

Figure 23 Illustration of the navigational force to the static motion goal.

However, for the dynamic motion goal, choosing $c_2 > 0$ is appropriate. In such case, an UAV tries to catch the motion goal by directing to the position of the goal in the next step. This functionality is illustrated on Fig. 24. Choosing valid coefficient c_2 is essential as too small c_2 results in a behavior similar to static goal and too big could result in a big navigational errors, when the trajectory of the dynamic motion goal is nonlinear.

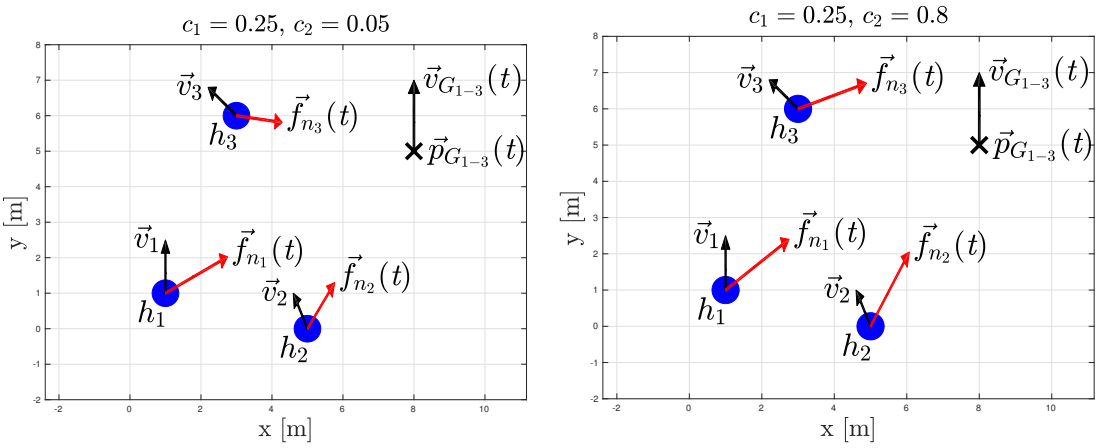
Addition of the navigational force to the flocking forces, as defined in sections 3.1.4 and 3.2.3 results to the final force of an UAV h_0 in time (or simulation step) t as

$$\vec{v}_0[t] = \vec{f}_{steer} + \vec{v}_0[t-1], \tag{40}$$

$$\vec{p}_0[t] = \vec{p}_0[t-1] + \vec{v}_0[t], \tag{41}$$

where

$$\vec{f}_{steer} = \vec{f}_s + \vec{f}_c + \vec{f}_a + \vec{f}_{oa} + \vec{f}_{n_0} - \vec{v}_0[t-1]. \tag{42}$$



(a) Small velocity coefficient.

(b) Big velocity coefficient.

Figure 24 Illustration of navigation force to dynamic motion goal in time (or computational loop) t.

3.4 Relative onboard localization

To be able to control an UAV, a navigational system of the UAV is needed. The two main approaches for navigational systems with usage of merely onboard hardware are visual navigation based on data from onboard cameras, and fuse and evaluation of data from sensors [33] with utilization of the GPS. The main difference between these two approaches is their usability in various environments. The utilization of the GPS is less complicated, however is only possible in an environment, where the GPS is reliable and accurate localization system. Nevertheless for swarms of UAVs, usage of the GPS arises a requirement of communication between the UAVs. The mutual communication is required between the UAVs to provide information about other UAVs state.

On the other hand, the visual navigation can be divided into two approaches as well. The first one is based on a pattern recognition, where after detection of the pattern in the image data, a relative position can be estimated. The estimation is performed according to the detected position of the pattern in the image, previously known parameters of the pattern and known camera orientation. Such patterns used for relative visual onboard localization procedure in [24], [27] and [44] are shown on Fig. 25. The second approach is based on optical data flow, where changes in optical data are evaluated to estimate the position of the UAV. The optical flow approach is more robust, however its utilization is not possible while flying over consistent surface where no significant objects can be found, e.g. snow. Both visual localization procedures are being developed by the MRS group and could be in the future deployed on the swarming model proposed in this thesis.

For any kind of relative onboard localization, the relative position is estimated with an unknown error. This error depends on camera resolution, ambient light, reliability and traffic delays of mutual communication, accuracy of the GPS and many other factors.

The relative visual onboard localization procedure is being researched by the MRS group, however it is complicated, problematic and has never been applied to swarms of UAVs in a complex environment. This thesis proposes a robust swarming model for a forest-like environment, therefore a precise relative onboard localization is required. Because of availability and simplicity of relative localization using the onboard GPS module, the visual based approach of localization was not used for the first deployment

of a swarm of UAVs in a forest-like environment. However, in a real forest, the GPS cannot be classified as reliable positioning system and other relative localization will have to be integrated into the swarming model, which is a part of a future research. Without usage of any cameras, there is no capability for an UAV to localize obstacles. Therefore all obstacles and their parameters must be given to the swarm as an input.

Figure 25 The pattern used for relative visual onboard localization by the MRS group.

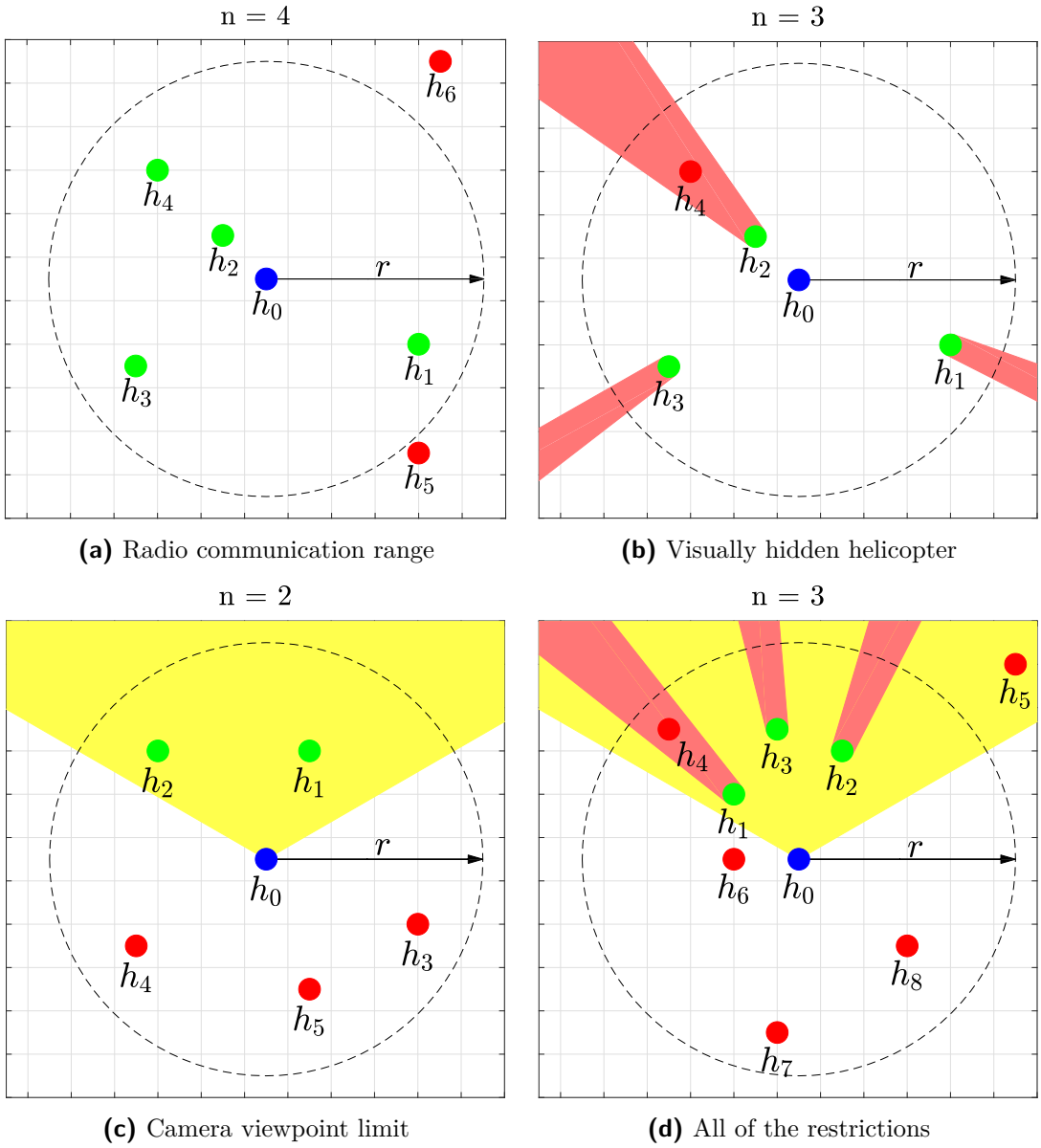
3.4.1 Restrictions of relative visual onboard localization

Behavior restrictions appeared while studying swarms and they differ for each kind of swarm. For example, a bird in a flock is restricted by its aerodynamic possibilities (lift, drag, the force of gravity, roll rate), number of interaction partners, weight or eyesight capabilities (most of bird species have eyes on the side of their heads) [45][46]. Another restriction for various species is reaction time, visibility, physical prerequisites or extra non-visual communication (ant's chemical path).

In the robotic context, we are restricted by hardware and software limitations. The relative onboard localization is limited by a camera viewpoint (similarly as the limit of bird's eyes), pattern recognition reliability, inability to localize visually hidden objects or individuals, camera resolution, accuracy of the GPS, communication delays or a size of environment where the UAV is capable to localize its surroundings.

A visual localization brings up many challenges to overcome for a swarm of helicopters. Thus as mentioned earlier, we did not use it for a first deployment of a swarm of UAVs in a complex environment. The primary restrictions of a relative visual onboard localization by an UAV are presented on Fig. 26.

- Fig. 26a expresses a distance r (m), in which successful and precise localization of other individuals can be performed.
- Fig. 26b shows an issue, where the closest objects create visual "shadow" behind them, and objects located in that shadow loose a localization potential.
- Fig. 26c displays a viewpoint restriction of the onboard camera, which is used to obtain image data to process the localization algorithms.
- Fig. 26d displays a model with relative onboard localization, where all the visual restrictions mentioned above must be considered.



(a) Radio communication range

(b) Visually hidden helicopter

(c) Camera viewpoint limit

(d) All of the restrictions

Figure 26 Visual limitations of an UAV h_0 with a single camera used for relative visual onboard localization of up to 8 neighbors.

3.5 UAV constraints

Application of the proposed model has to be supported with real system constraints. Since swarming algorithm runs as SITL[2], an efficient loop rate must be chosen. The loop rate should be maximized to a value in which the UAV is capable to relatively localize its surroundings and compute the swarming algorithm. Maximization of the loop rate leads to more fluent and precise swarming behavior. While using the GPS localization, a speed of mutual communication between the UAVs must be considered as well, because too slow transmission could corrupt the system functionality.

The loop rate is connected with a loop distance. The loop distance specifies the maximal change in the position of the UAV in one loop. Setting of the loop distance to a small value leads to an increase of probability to suppress a possible localization

[2]Software in the Loop

disturbance (the GPS inaccuracy), because such error would had to be present for numerous loops. However, disadvantage of a low loop distance is a degradation of the swarm movement speed. On the other hand, setting of the loop distance to a larger value could lead to collisions, loss of self-optimizing behavior and non-efficient movement. Therefore in applications, where the motion speed of the swarm is not the main priority, this value should be chosen wisely, according to the used UAV platform, localization reliability and accuracy, connection network and the environment.

The swarm is restricted to maintain a constant altitude, if it is flying in a planar environment. That is for an UAV h_0, a constant altitude z_c and simulation step t provided in the swarming controller as

$$z_0[t+1] = z_c - z_0[t] \tag{43}$$

before flying and publishing its state to the ROS network. In a 3D environment, this compensation is not necessary, however the altitude must be saturated between critical limits. These altitude limitations are defined by a flyable zone. A lower limit is primarily chosen as a safe altitude to overfly all objects on the ground, not classified as obstacles e.g. high grass or people. An upper limit is necessary for majority of applications, since micro aerial vehicles are not constructed for flying in high altitude, where the air is dispersed and the vehicles malfunction. The upper limit is also often defined by the legislative i.e. maximal allowed altitude in the airspace class G in the Czech Republic is 300 m.[3]

A model of the proposed swarming behavior, with the loop distance and altitude saturation is illustrated on Fig. 27. This diagram extends the diagram on Fig. 4. The inputs of the swarming controller are own state of the controlled UAV, the states of other UAVs in the swarm (obtained via a relative localization), and parameters of the obstacles.

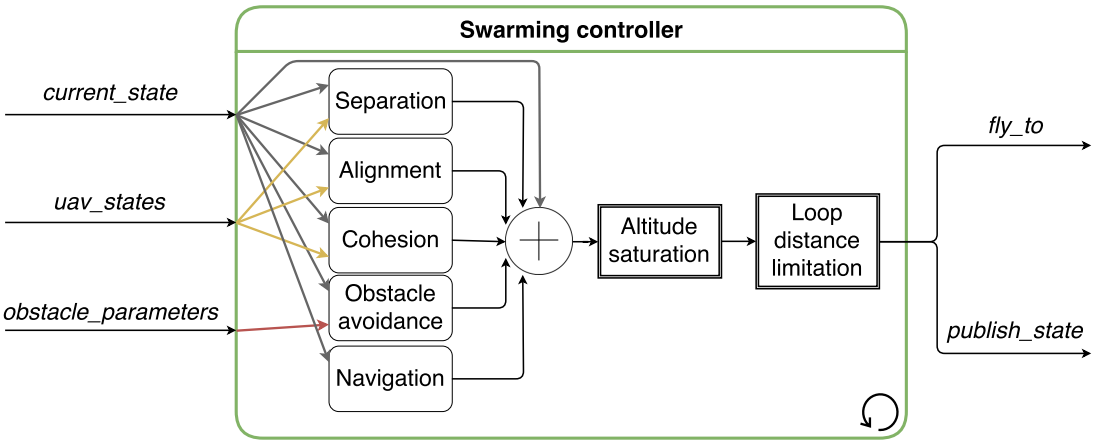


Figure 27 A process diagram of the swarming controller.

[3]More information in legislative addendum from CIVIL AVIATION AUTHORITY `http://lis.rlp.cz/predpisy/predpisy/dokumenty/L/L-2/data/effective/doplX.pdf`.

3.6 Pseudocode

The swarming algorithm for one UAV, defined by the flocking forces in this section can be rewritten to a pseudocode in Alg. 1. The algorithm starts with an UAV taking off, that is done by initializing of the MBZIRC system described in section 2.1. The swarming controller runs in a loop, which starts with a relative localization of nearby UAVs and obstacles, and continues with computing of the flocking forces. A new state is determined with usage of current position and the flocking forces. The determined state velocity is then limited to the loop distance and altitude is saturated between the altitude limits. Note, that when the swarm should maintain a constant altitude, the lower and upper altitude limits equal the constant altitude. At last, the new desired position is sent to the MPC, which ensures the flight to the computed position is performed.

```
# UAV take off
Run controllers;
Take off;
# Swarming controller
while flock do
    # Perform a relative localization of the neighborhood
    Localize UAVs;
    Localize obstacles;
    # Compute the desired position to fly to according Eq. 41
    # The size of the UAV velocity (speed) must be limited to the value of the
    loop distance
    steering_force = separation + alignment + cohesion + obstacle avoidance +
                     + navigation - current_velocity;
    new_velocity = current_velocity + steering_force;
    new_velocity = limit(new_velocity);
    new_position = current_position + new_velocity;
    # Ensure, that the altitude does not exceed the altitude limits
    if new_position.altitude < altitude_lower_limit then
        new_position.altitude = altitude_lower_limit - new_position.altitude;
    else if new_position.altitude > altitude_upper_limit then
        new_position.altitude = altitude_upper_limit - new_position.altitude;
    # Perform a flight to the new position
    fly_to(new_position);
    # Ensure the loop is running with given loop rate
    sleep(1/loop_rate);
end
```

Algorithm 1: Pseudocode of the decentralized swarming algorithm described in section 3.

4 Simulation

The proposed swarm model was implemented in ROS[1] and verified in Gazebo simulator. To be able to control and stabilize a swarm of real UAVs, the MBZIRC system developed by the MRS group, with help of University of Pennsylvania and University of Lincoln, for the MBZIRC 2017 competition was used. Controllers of the MBZIRC system, described in section 2.1, provide low-level control of the helicopter rotors, a model prediction with a variable length of control horizon and accurate flight of the controlled UAV. The development in ROS allows to simulate real robotic systems in simulations and in a short time to transfer it to real helicopters.

This chapter presents basic simulation results of swarming behavior formerly described. It focuses on forming of a swarm, usage of the navigational force and mainly on flying in environments with single and multiple obstacles. It also presents results of flights in a forest-like environment being our target motivation.

Simulations are hardware demanding, therefore more computers were connected together to communicate via topics, described in section 2.1. The connection results to a state, where each connected machine serves as a computational base for z helicopters, where z is number of UAVs a single machine is able to control. For example, 64-bit machine with $CPU\ Intel® Core^{TM}\ i7-2600 @ 8 \times 3.40 GHz$, SSD memory disk and $4 \times 4\ GiB$ DDR3 RAM memory is capable to run graphical UI of Gazebo simulator and 6 UAVs each running the MBZIRC system controllers illustrated on Fig. 4. The number of available machines to the author of this thesis, which were used for the simulations, limits number of UAVs in the simulations to less than 10. However, connecting more machines together could result in the verification of swarms with a much bigger number of UAVs. Such expansion is part of a future work.

Localization of neighboring UAVs and obstacles in the simulations was not performed by relative visual onboard localization, defined in section 3.4. Such localization brings up new challenges and to keep the simulation easy, relative onboard localization with no usage of visual data processing was used. Non-visual localization is based on the broadcasting of its odometry data via radio signal, followed by its subscribing and processing by the rest of the UAVs. Given its own position and the odometry data of others, the UAV is capable to determine relative position of other UAVs. Position and parameters of the obstacles are predefined in the UAV controller prior the mission.

Parameter t entitling simulation figures 28-54 of the swarming behavior defines number of computational loops. The arrows indicate the UAVs velocity direction at the current loop t. All of the simulations were performed in R^3 and divided onto *planar* (2D - constant altitude of all UAVs) and *3D* experiments. At last, some simulation figures do not contain a legend to maintain readability of these figures. Each of these legend-less figures contain trajectories of various number of UAVs, where the description of the UAVs is insignificant.

[1] Robot Operating System

4.1 Boids

Pure Boids model, thus only flocking rules defined in section 3.1, were applied to a set of UAVs in this subsection. Expected behavior for such simulation in an environment without obstacles is firstly a start of an attraction of all mutually localized UAVs to form a swarm. Next, stabilization of the formation is expected, followed by a motion of the whole swarm to a nondeterministic direction, which is caused by the alignment force defined in section 3.1.2. This behavior is truly expected for flocking forces alone as theoretically shown in [1]. Craig Reynolds has presented in [1] a method for graphical simulations to represent a flock of flying birds, which randomly fly around to induce a realistic feeling.

Fig. 28 displays the expected behavior in planar environment. Initial UAVs position, together with the flocking forces, represented by velocity computed for loop $t = 1$ is illustrated on the first sub-figure. Initial velocity of all UAVs is always zero, therefore the velocities in the first loop illustrate merely the flocking forces without an influence of the UAVs velocity. On the next sub-figures, we can gradually describe forming of a swarm, its stabilization, and finally obtaining of a common velocity for the whole swarm and flying out in that direction.

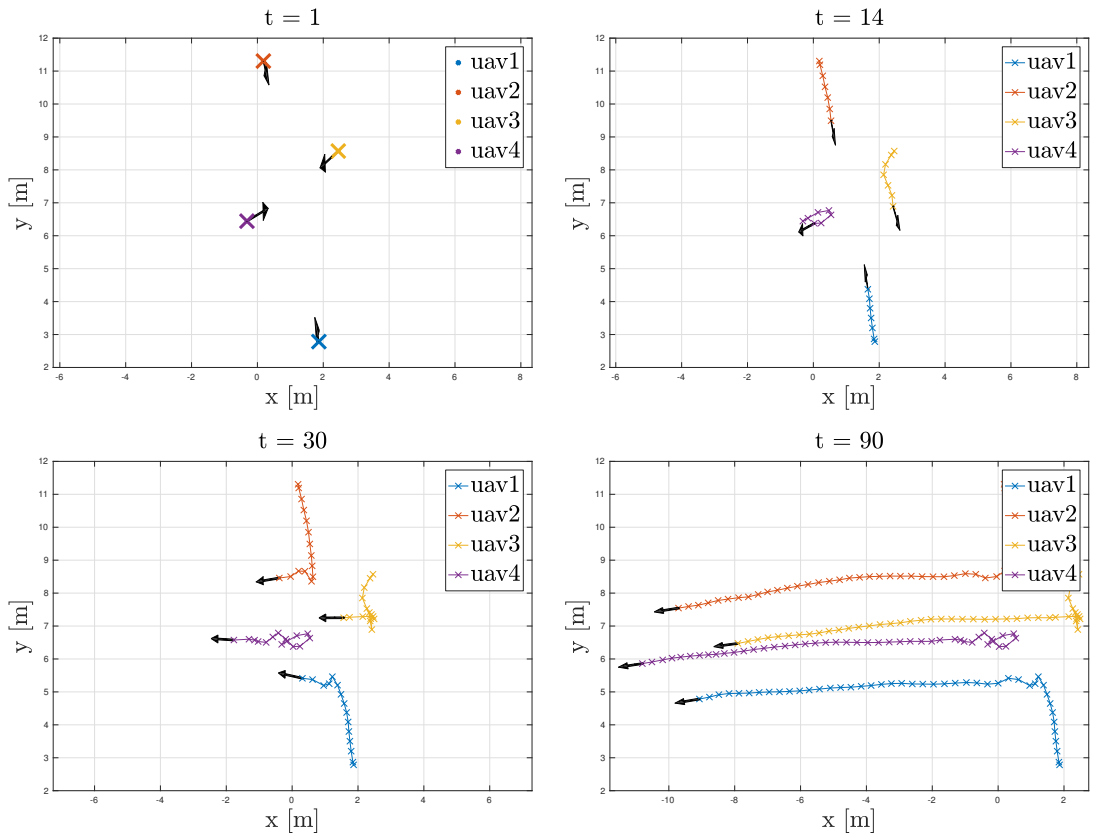


Figure 28 Forming of a planar *Boids* swarm composed of 4 UAVs.

Executing of the same simulation in 3D space is shown on Fig. 29. We can see again that after forming the swarm, it flies to a nondeterministic direction. The direction is similar to the plane (x, y) used in Fig. 28, due to the same initial positions of the UAVs.

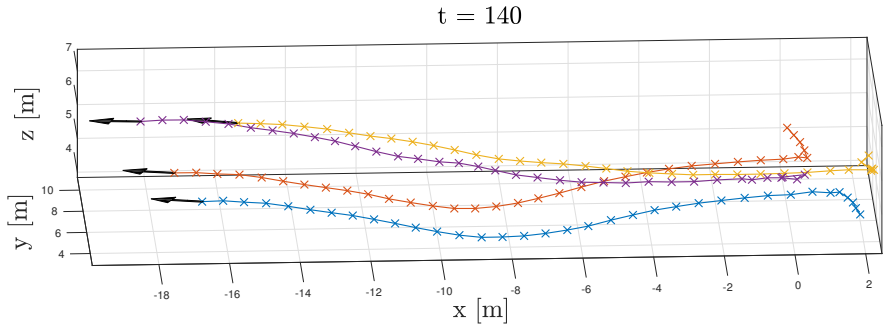


Figure 29 Forming of a 3D *Boids* swarm composed of 4 UAVs.

Forming process of the $\alpha - lattices$ defined in section 3.1.4 is illustrated on Fig. 30 and 31. Presence of the $\alpha - lattices$ confirms correct implementation of the model as it converges to an optimal formation. Individual distances between the UAVs during the simulation illustrated on Fig. 29 and 31 are plotted on Fig. 32. Note that even though the distance, where the separation force starts affecting the behavior is $d = 4\,\text{m}$, the edge-length ξ in a free 3D space for the $\alpha - lattices$ stabilizes on $\xi = 2.80 \pm 0.15\,\text{m}$. The edge-length ξ is an equilibrium point between the separation and cohesion force, which operate against each other, and the difference between d and ξ can be adjusted by change of the separation force parameters. Also, the edge-length ξ is not equal for all the UAVs distances. The inaccuracy is given by the controllers oscillations and relative localization inaccuracy. These constraints are expected, since the model has been designed and simulated on controllers intended for use on real UAVs, where factors like wind or mutual communication delays affect negatively the swarming behavior.

Fig. 33 displays the same forming of a swarm as Fig. 29, but with 7 UAVs. Plotting individual distances of the simulation on Fig. 34 results in extensive inaccuracy of the edge-length ξ between the UAVs. This interval increment is given by bigger number of UAVs in the swarm, where the swarm needs more loops to be able to fully develop optimal formation. Also, the individual distances between the UAVs are, with the same simulation parameters, bigger than the ones on Fig. 32. That is caused by increase of number of localized swarm members, therefore in an increase of the separation force, defined in section 3.1.1, which shifts the equilibrium point to a higher value.

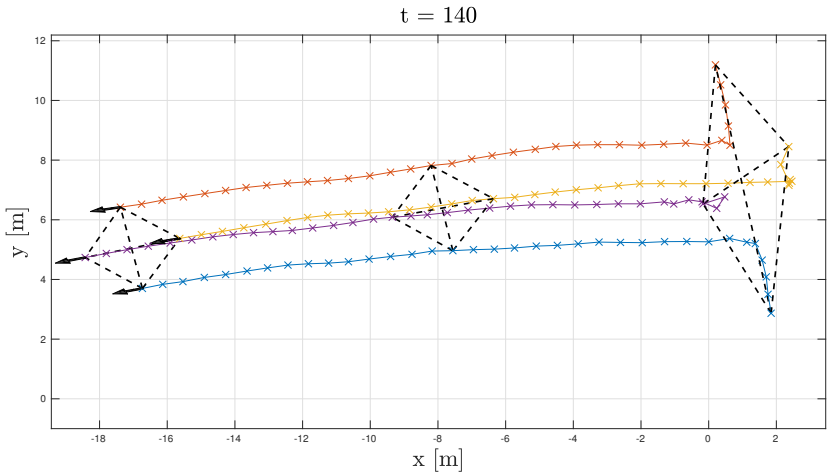


Figure 30 Illustration of the $\alpha - lattices$ in a swarm composed of 4 UAVs in a free planar environment.

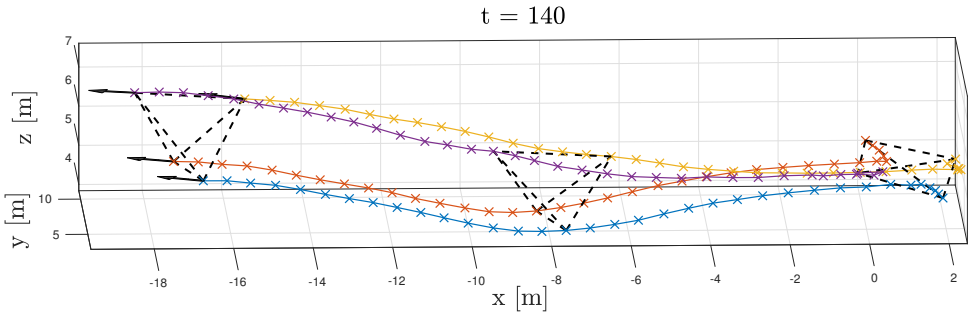


Figure 31 Illustration of the $\alpha - lattices$ in a swarm composed of 4 UAVs in a free 3D environment.

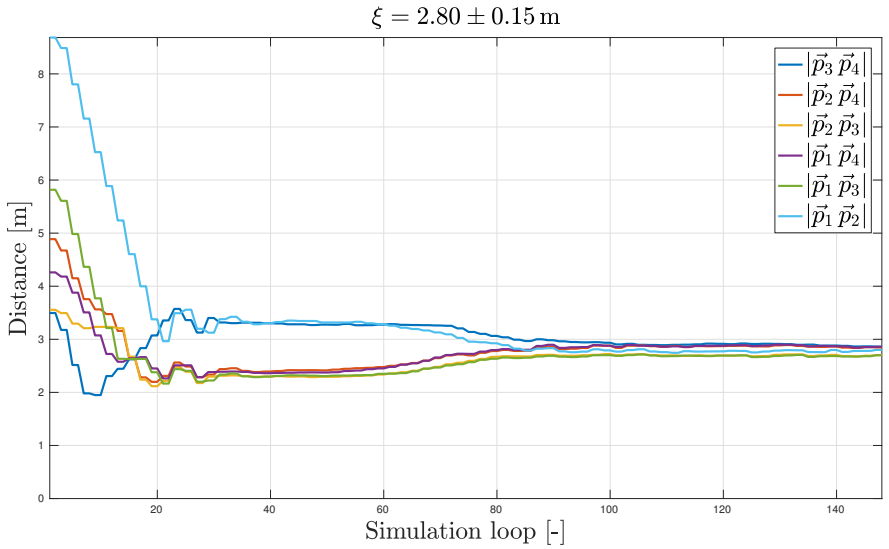


Figure 32 Individual distances between all the UAVs during the simulation illustrated on Fig. 29.

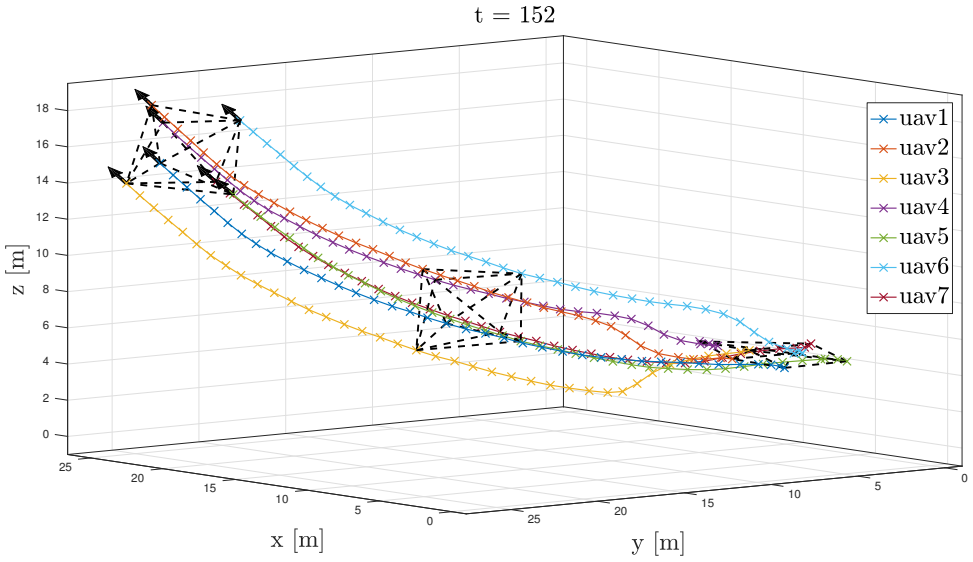


Figure 33 Forming of a 3D *Boids* swarm composed of 7 UAVs.

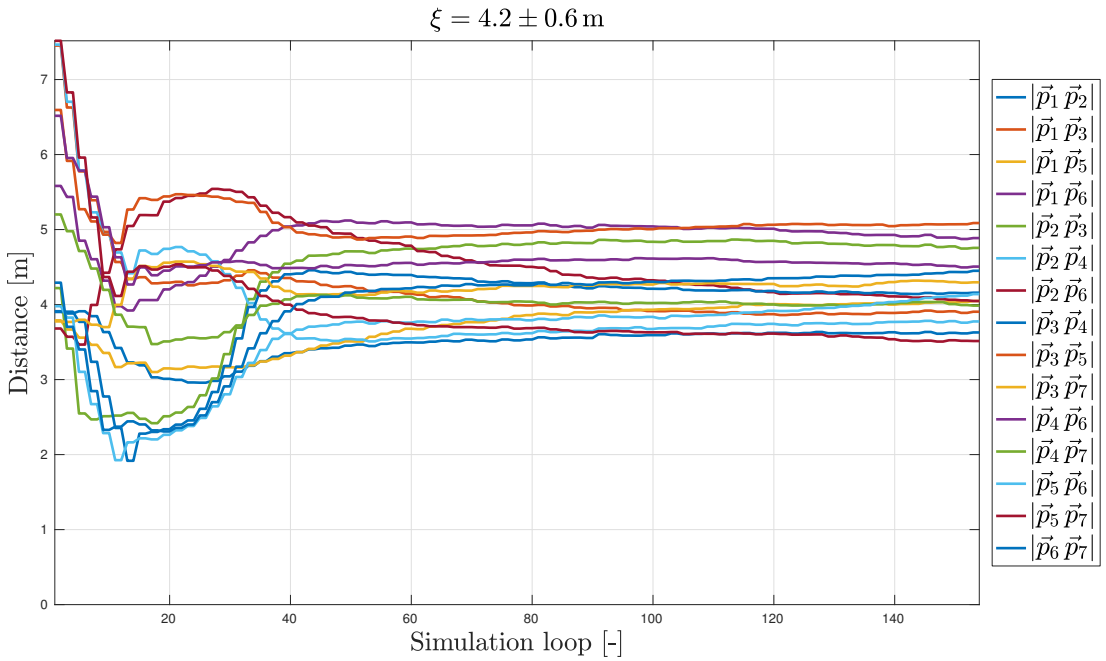


Figure 34 Individual distances between the UAVs during the simulation illustrated on Fig. 33.

Having two distant swarms, not capable to localize each other shows Fig. 35. The result of this simulation is that both the swarms behave individually on their own. Therefore each of those swarms is independent without performing any interaction with the second one.

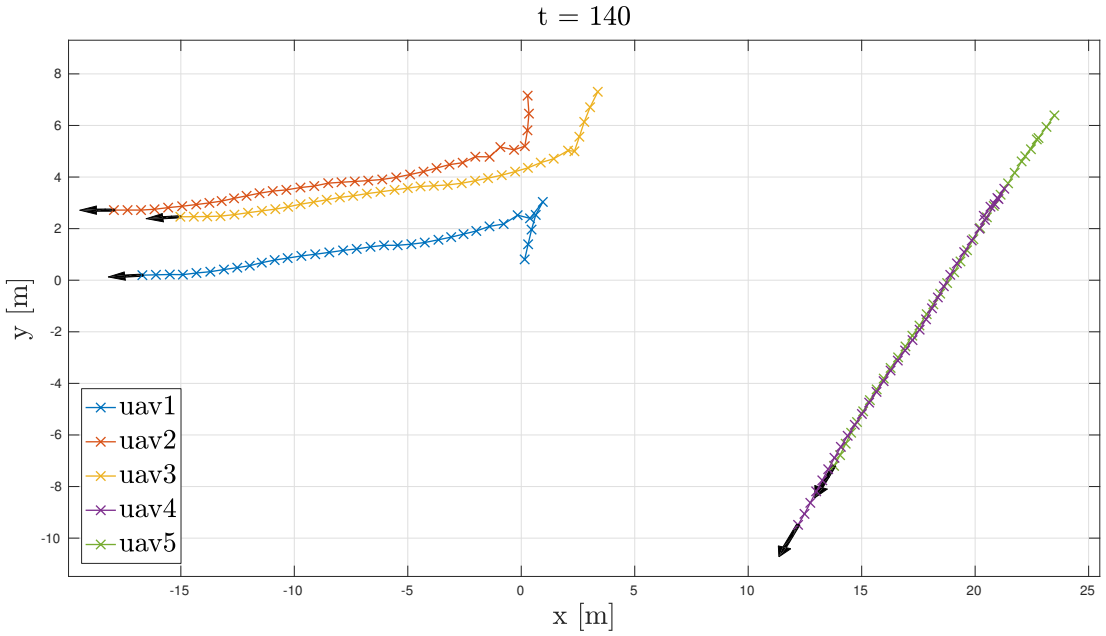


Figure 35 Illustration of two distant swarms not localized by each other.

4.2 Navigational force

Adding the navigational force, defined in section 3.3, results in controlled swarm motion capability. Such motion in a planar environment without obstacles is illustrated on Fig. 36. After forming of the swarm, it is attracted towards the common static motion goal. For a swarm composed of 5 autonomous UAVs on Fig. 36, the swarm forms a pentagon, where every UAV has similar distance to the common motion goal, as plotted on Fig. 37. The difference in the distance of all the UAVs to the common goal for the stabilized formation is less than 1 m.

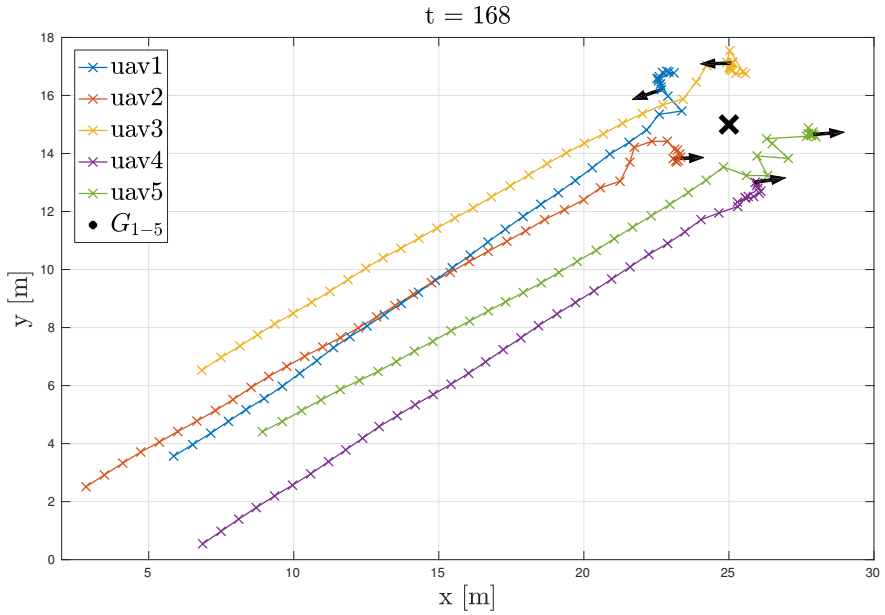


Figure 36 An influence of the navigational force in free planar environment on a swarm composed of 5 UAVs.

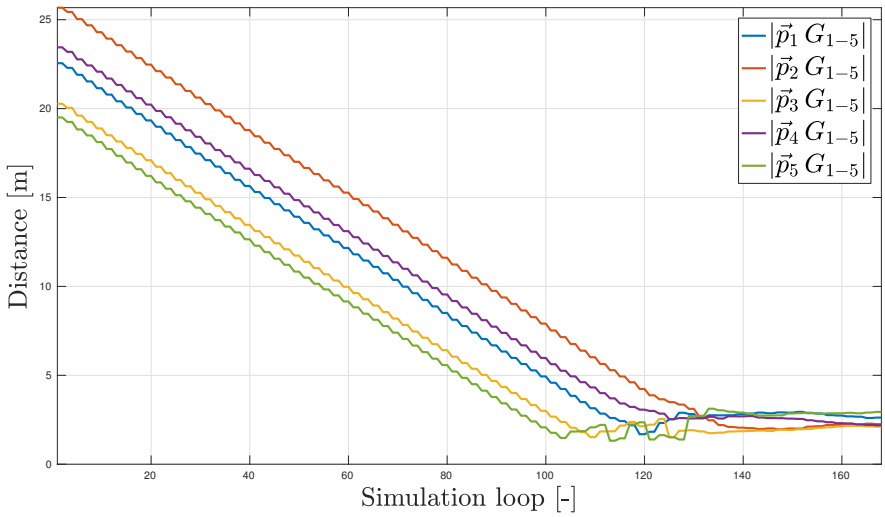


Figure 37 Individual distances between the UAVs and the common static goal during the simulation illustrated on Fig. 36.

.

In free 3D space the swarm again converges to the $\alpha - lattices$, as illustrated on Fig. 38, and its distances between the UAVs and the common motion goal for stabilized formation again differ with a slight error, as in planar environment. The behavior of pure Boids model or with the navigational force addition differs for planar and 3D space only in formation it consolidates.

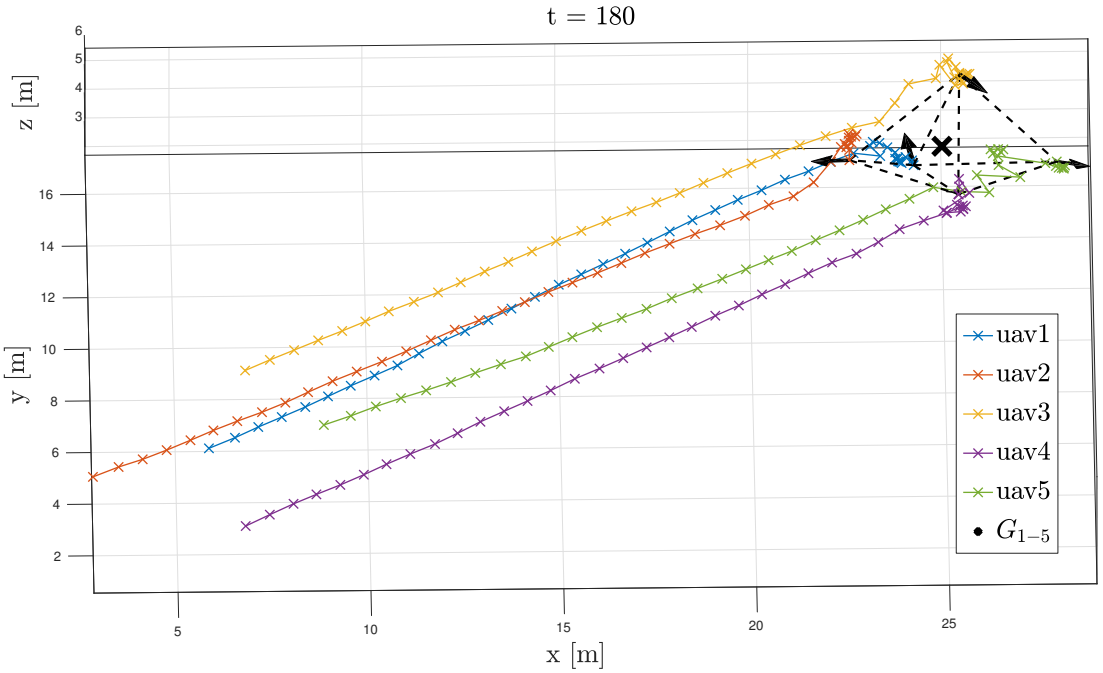


Figure 38 An influence of the navigational force in a free 3D environment on a swarm composed of 5 UAVs.

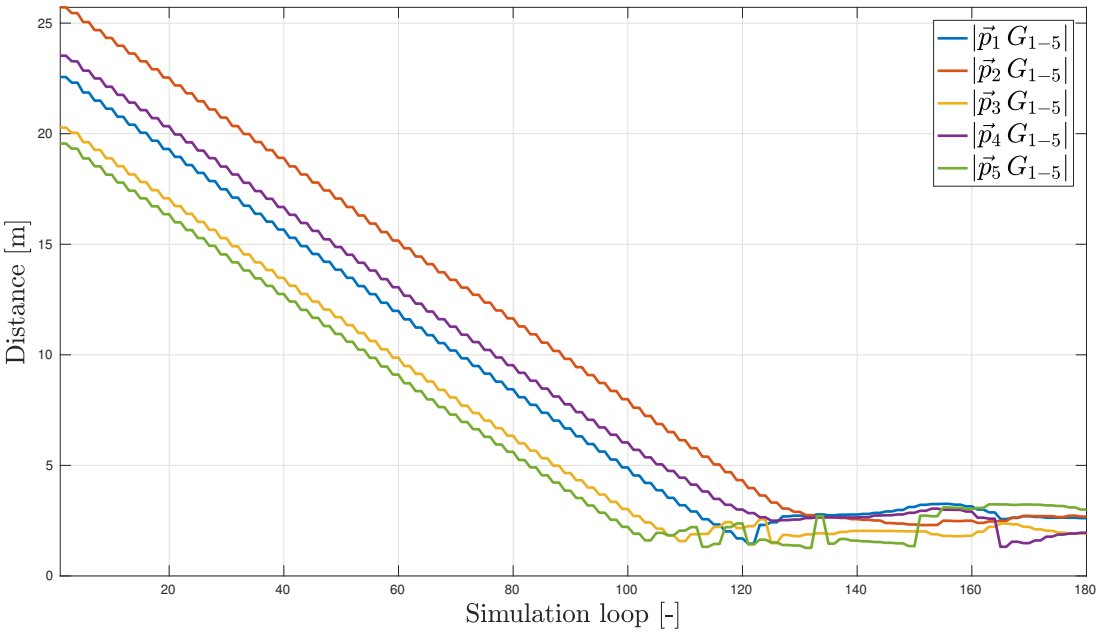


Figure 39 Individual distances between the UAVs and the common static goal during the simulation illustrated on Fig. 38.

.

4.3 Obstacle avoidance

Final and the most tricky force to be added. This section tries to cover many simulations for both types of defined obstacles and their combination. Bypassing of single cylindrical obstacle is illustrated on Fig. 40 and 41. It is clear, that the swarm is capable of self-disintegrating to be able to optimally bypass such obstacle.

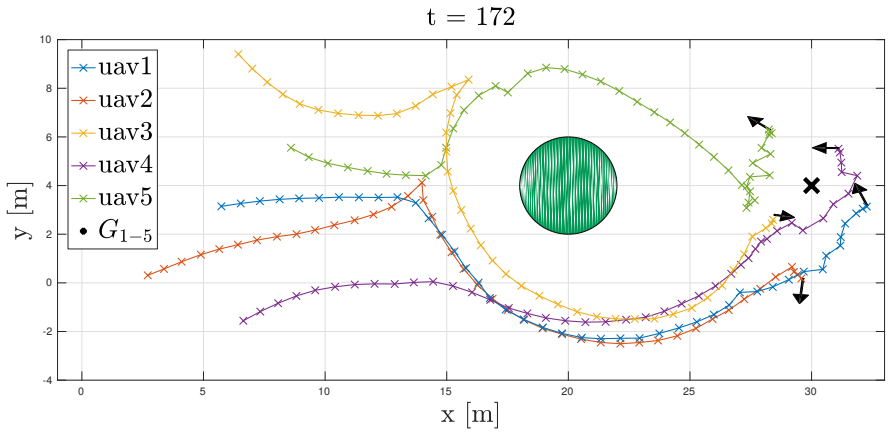


Figure 40 Process of bypassing a cylindrical obstacles by a swarm composed of 5 UAVs in planar environment.

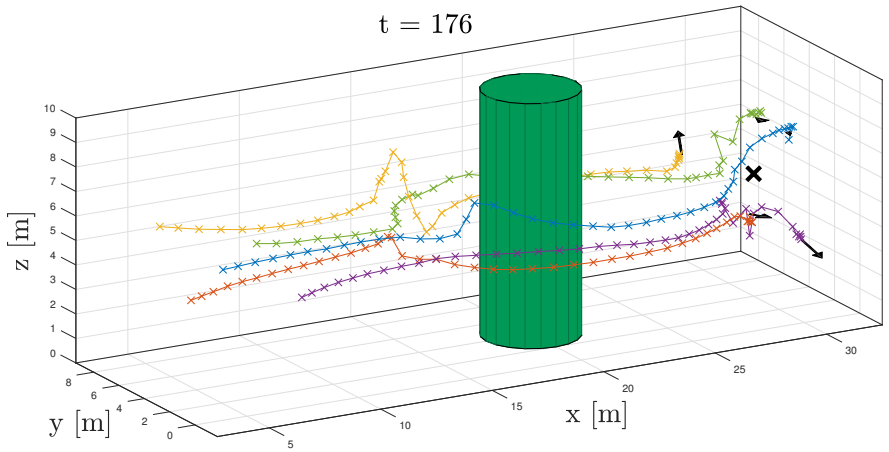


Figure 41 Process of bypassing a cylindrical obstacles by a swarm composed of 5 UAVs in 3D space.

Fig. 42 and 43 illustrate bypassing of a single wall obstacle, where the swarm does not disintegrate as for the cylindrical obstacle. The disintegration is dependent on the dimensions of the obstacle, where the wall on these figures is much larger for the swarm point of view then the cylindrical obstacle.

The UAVs trajectories while flying through a gap between two cylindrical obstacles in planar environment highly depends on width of the gap. If the gap is wide enough, so the navigational force overcomes the separation forces of the obstacles, the swarm is capable of going through the gap, as illustrated on Fig. 44.

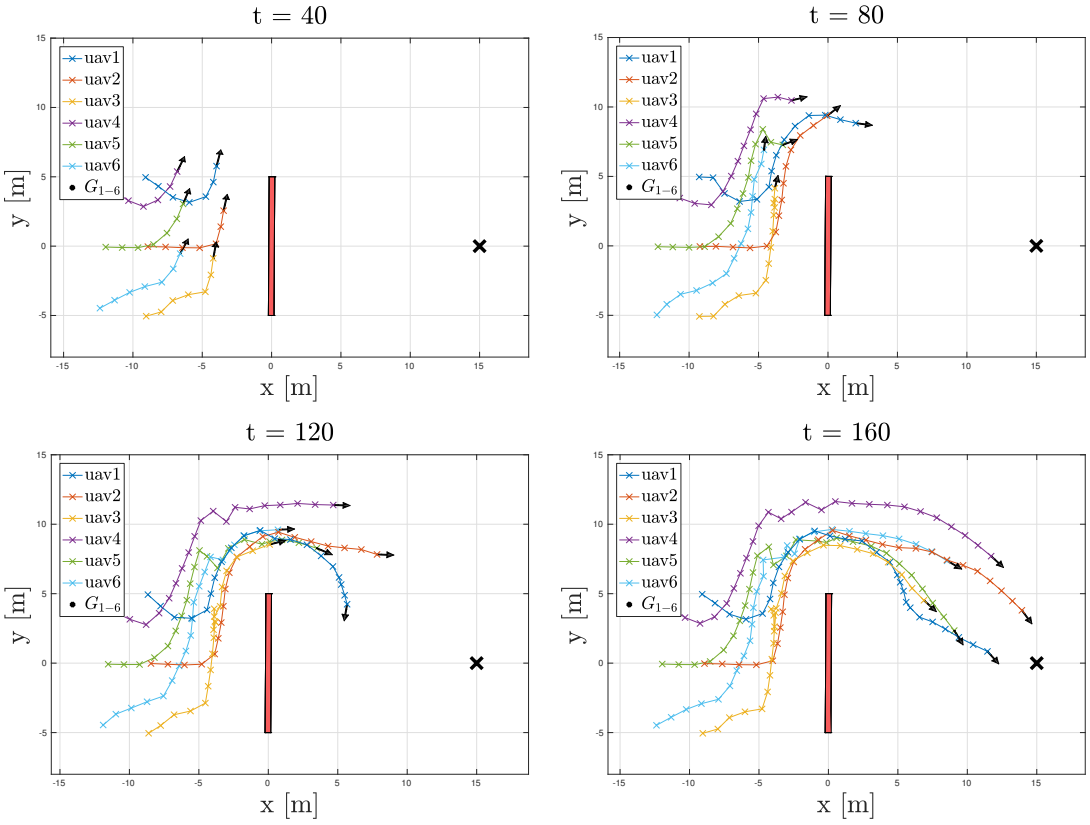


Figure 42 Process of bypassing a wall obstacle by a swarm composed of 6 UAVs in planar environment.

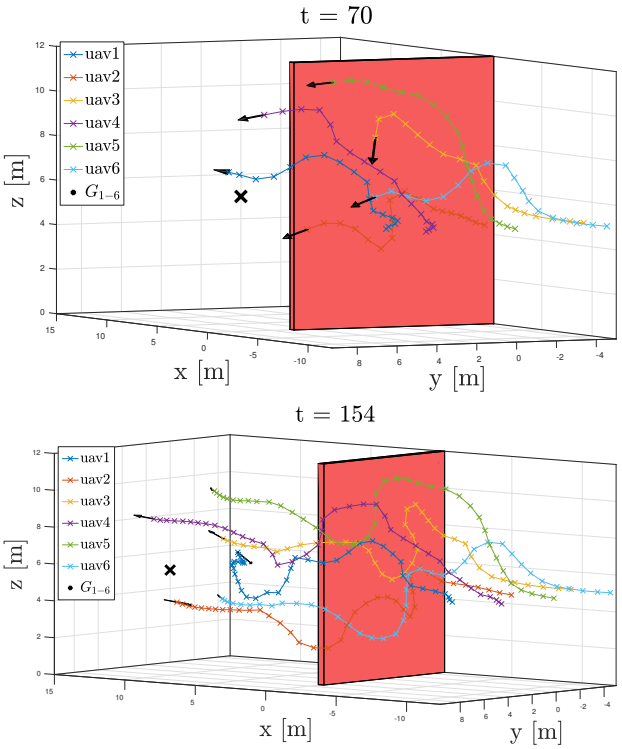


Figure 43 Process of bypassing a wall obstacle by a swarm composed of 6 UAVs in 3D space.

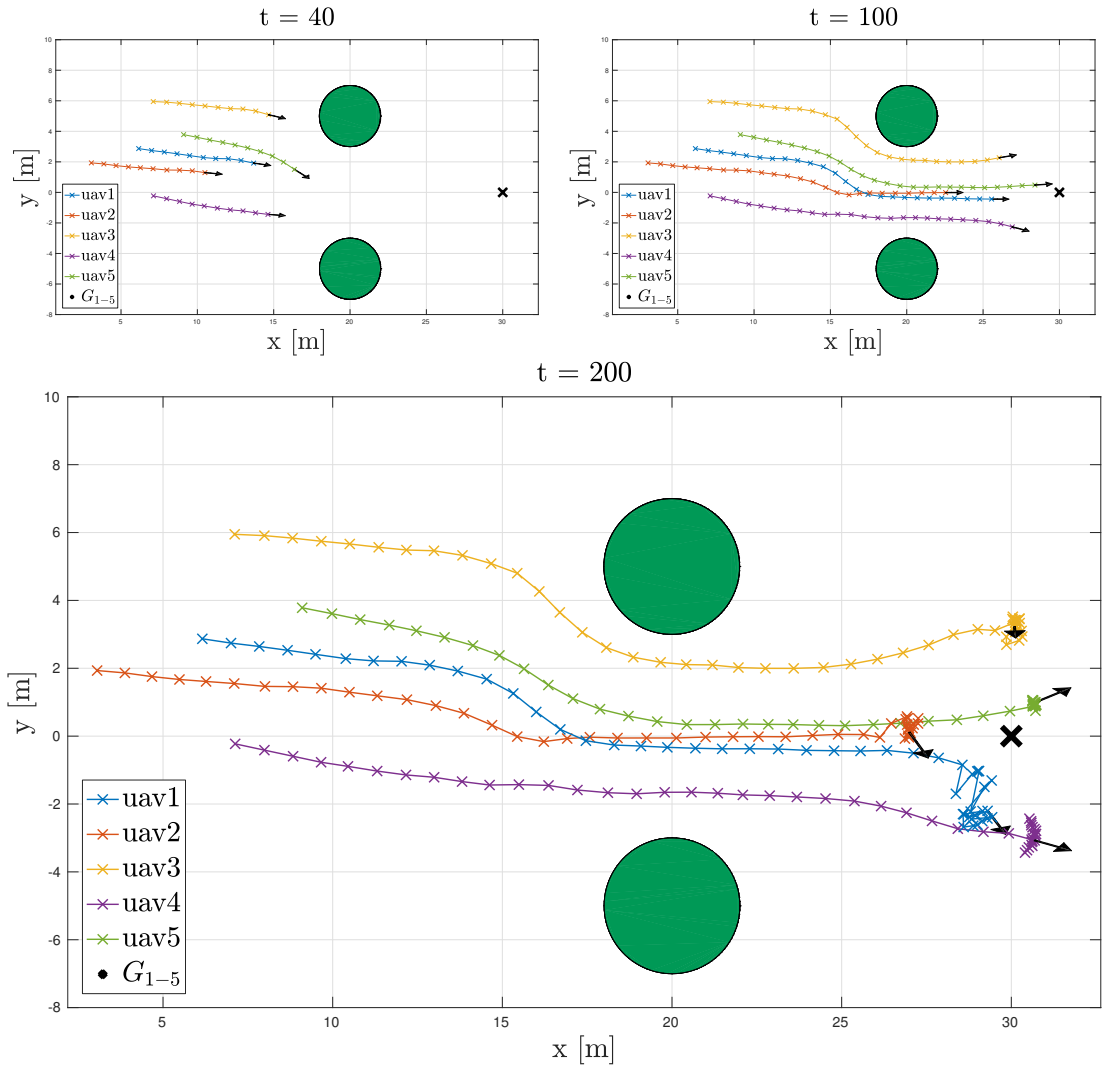
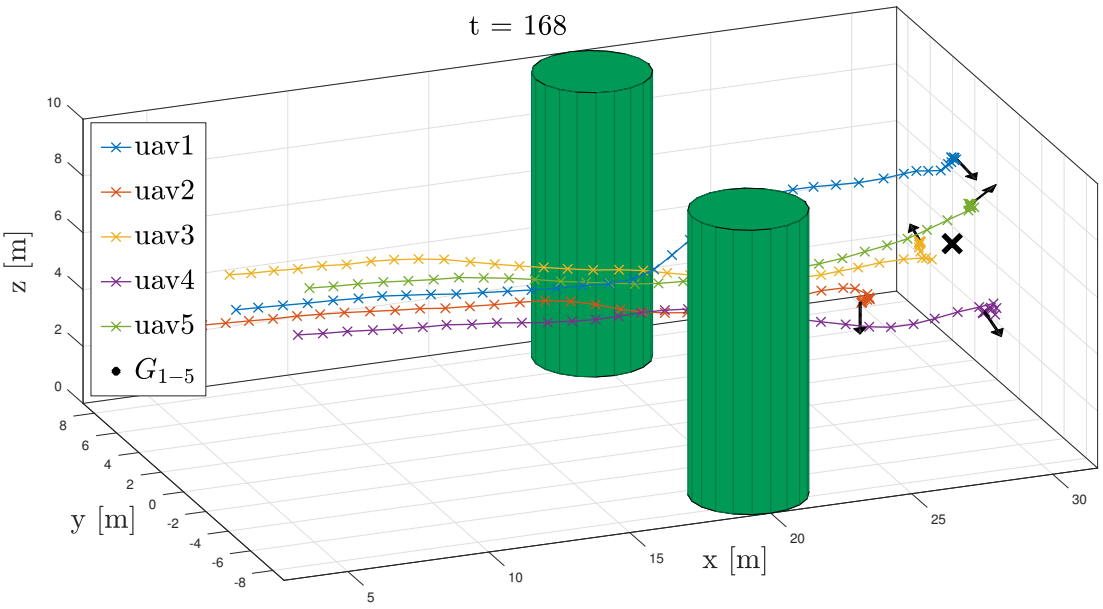


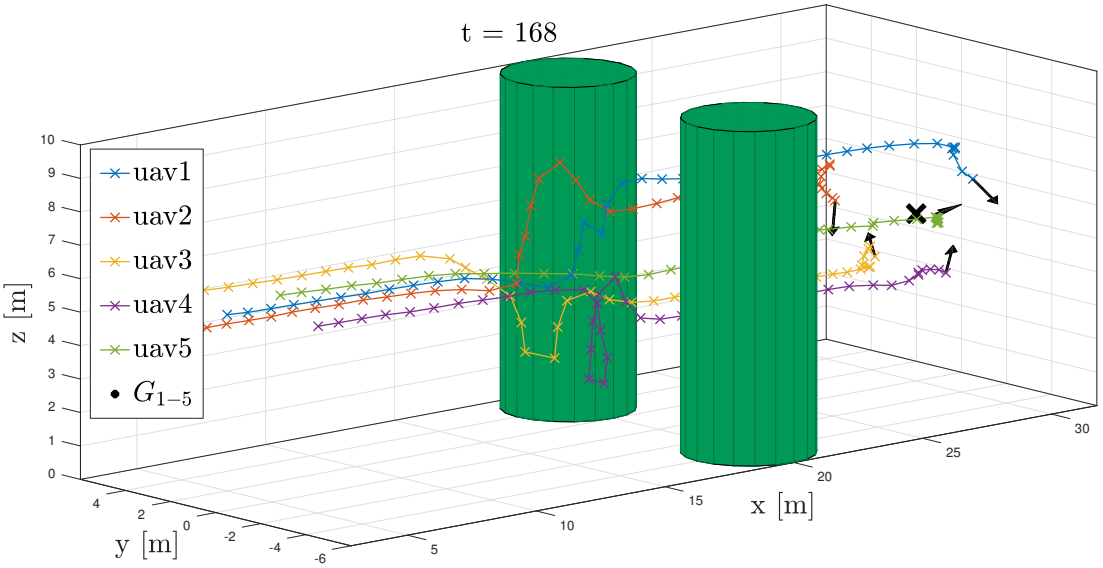
Figure 44 A flight of a swarm composed of 5 UAVs through a gap between two cylindrical obstacles in planar environment.

In 3D space, the gap can be narrowed, because the swarm forms a formation capable to fly through without flying one by one. Such behavior is illustrated on Fig. 45, where the gap widths are 10 m and 4 m. It is important to note, that the swarm in the 3D space independently adapts itself to the environment while maintaining safe distances between the UAVs and the obstacles for the whole time. This behavior ensures clearer and faster swarm motion in obstacles-filled environment. Therefore, it is recommended to perform simulations and experiments in a complex environment without setting of a constant altitude.

A full process of forming a swarm formation, illustrated on Fig. 45b, capable of a flight through a narrow gap between two cylindrical obstacles is illustrated on Fig. 46. The process is based on the localization of the obstacles, and navigating the swarm movement into the narrow gap, while forming an optimal formation allowing the swarm to go through as a whole and not one by one.



(a) Wide gap - 10 m.



(b) Narrow gap - 4 m.

Figure 45 Forming of an optimal formation by a swarm composed of 5 UAVs before a flight through a gap between two cylindrical obstacles.

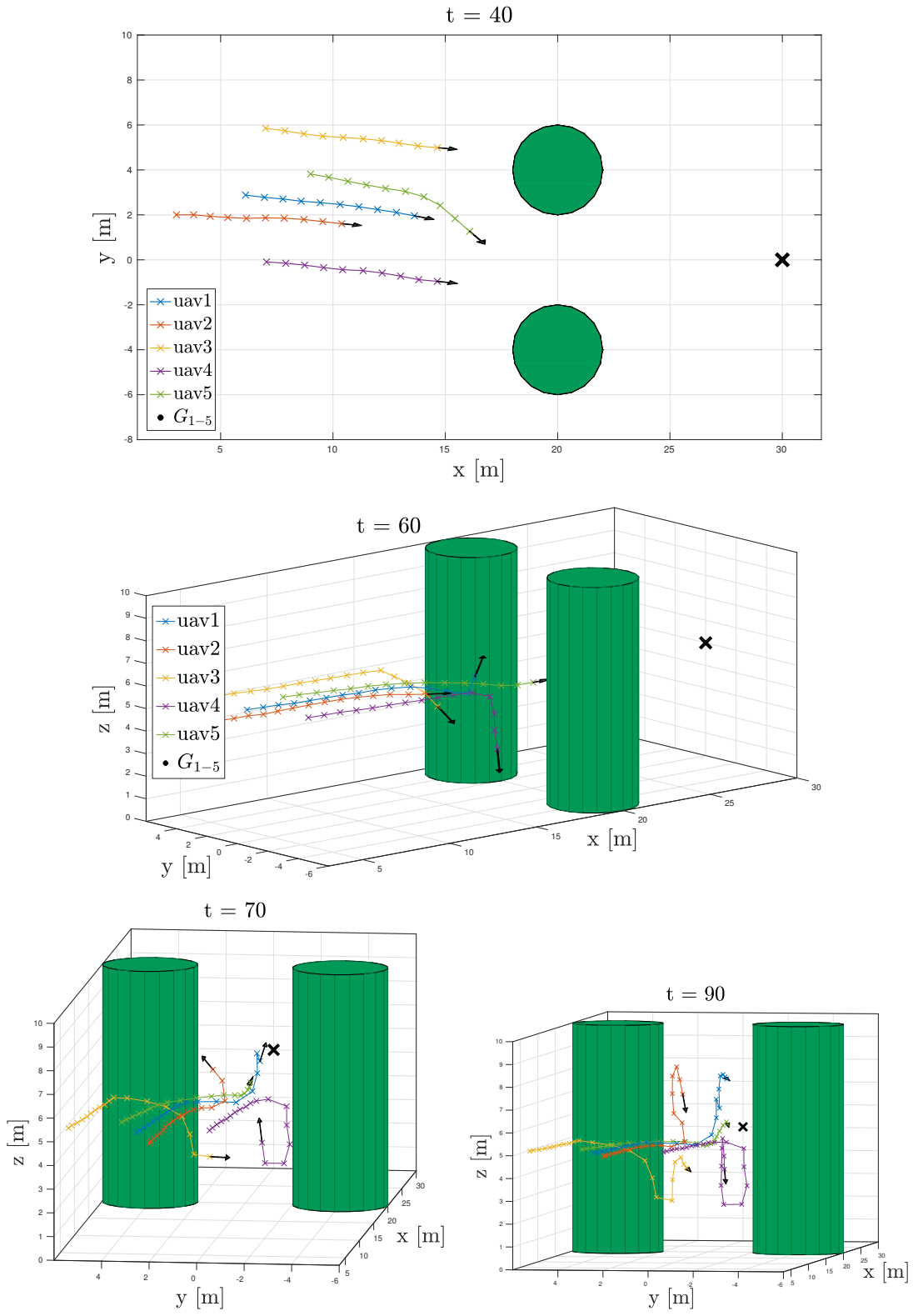


Figure 46 Detailed look on the simulation illustrated on Fig. 45b.

A forest-like environment in the Gazebo simulator is illustrated on Fig. 47. However, using textures of the trees has negative effect on the simulation speed, thus the trees in forest-like environments were replaced by cylindrical obstacles as shown on Fig. 56. Triggering of the swarming behavior with a swarm situated in a forest-like environment is illustrated on Fig. 48 for planar and Fig. 49 for 3D space. These figures show simulations, where no motion goal is set for the swarm. The results are similar as in section 4.1, with a difference, that the UAVs are capable to maintain safe distance from the obstacles. Initially, the swarm is formed, cumulates its members and finds it way out of the obstacles-filled environment.

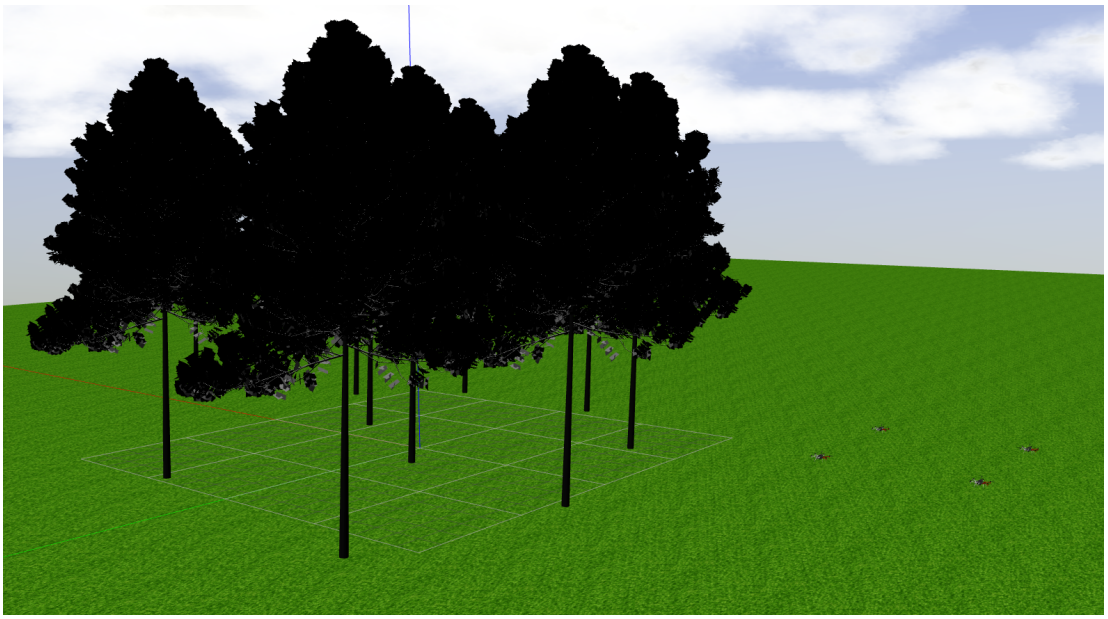


Figure 47 Visualization of a forest with textures of trees in the Gazebo simulator.

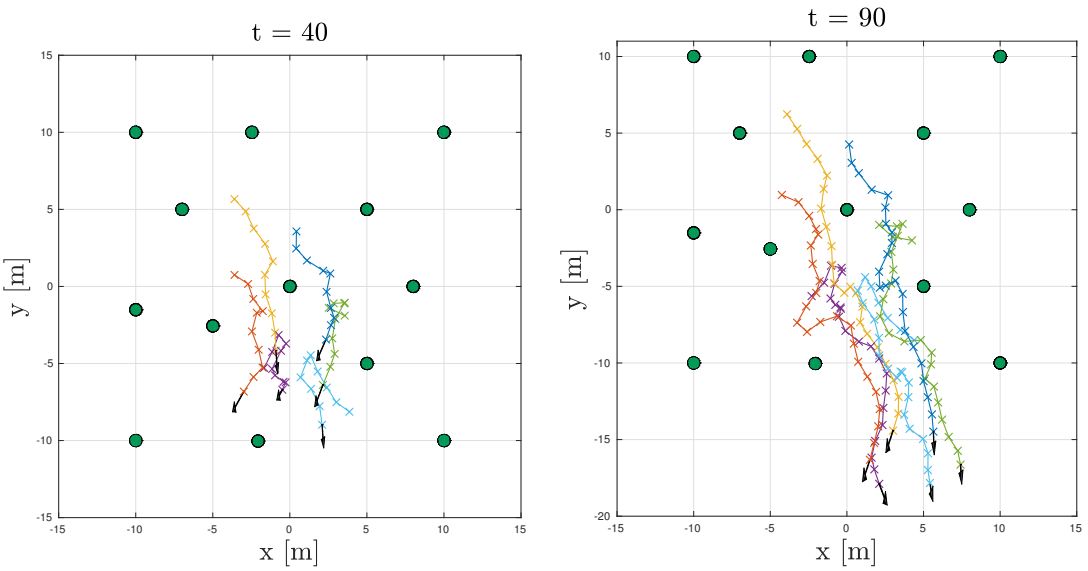


Figure 48 The swarming behavior in a forest-like planar environment.

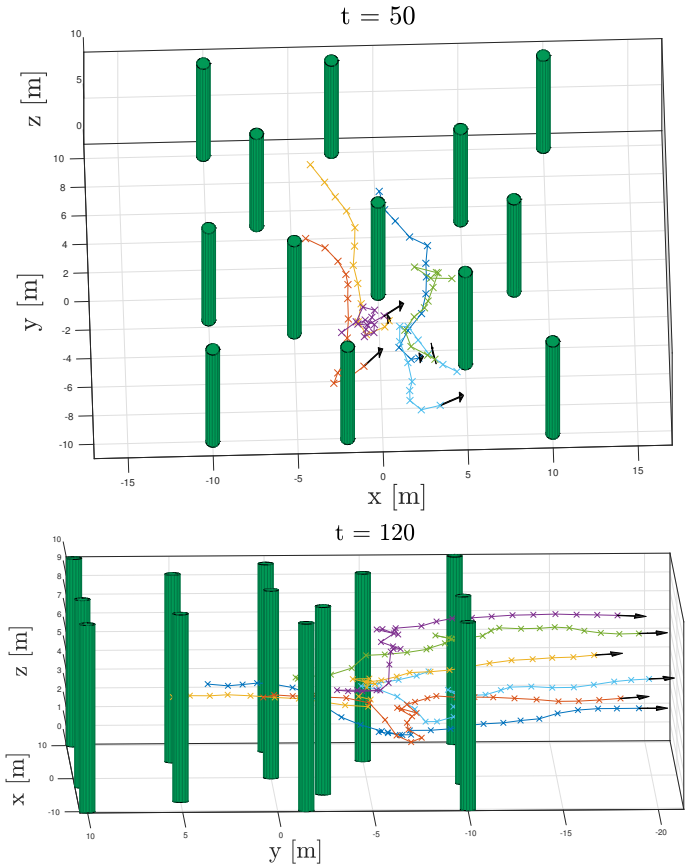


Figure 49 The swarming behavior in a forest-like 3D environment.

Setting a common static motion goal to a swarm of UAVs in a forest-like environment, illustrated on Fig. 50 for planar and Fig. 51 for 3D environment, leads to the expected behavior. The navigational force attracts the swarm towards the motion goal, while the swarm safely moves out of the complex environment.

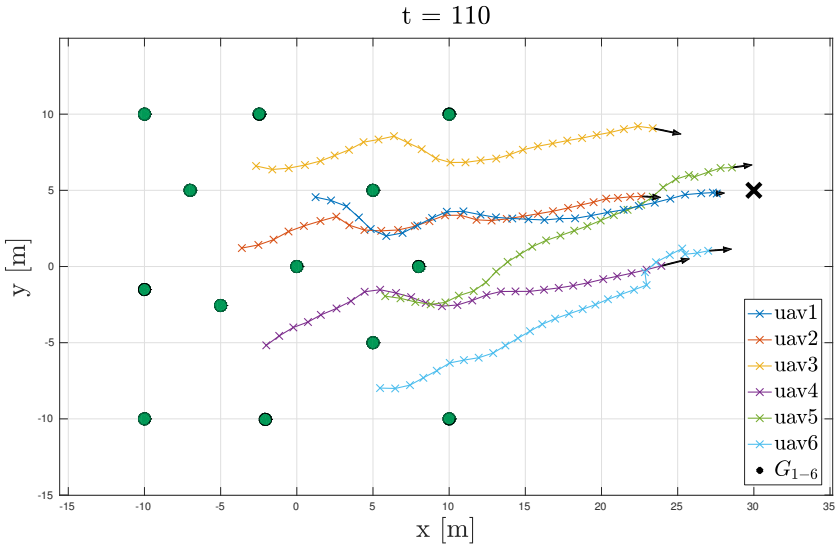


Figure 50 The swarming behavior triggered in a forest-like planar environment with a common static motion goal.

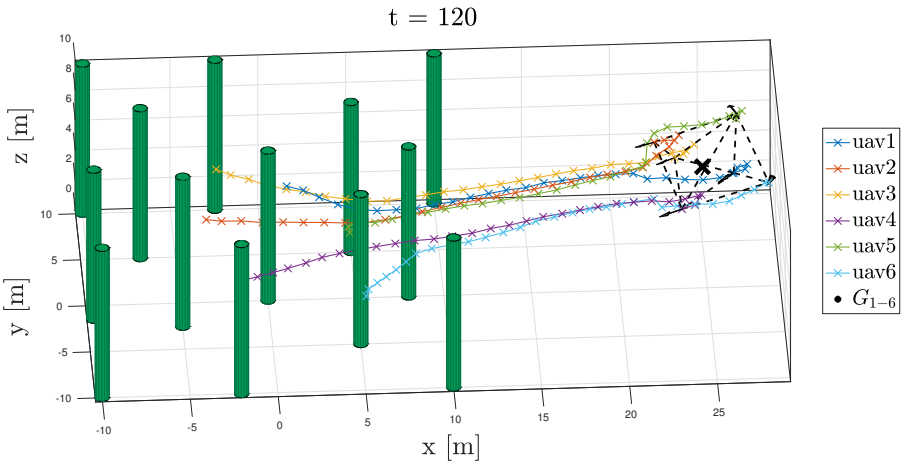


Figure 51 The swarming behavior triggered in a forest-like 3D environment with a common static motion goal.

A simulation of 9 UAVs navigating through a forest-like environment is illustrated on Fig. 52 for planar, Fig. 54 and 56 for 3D space. These simulations illustrate capabilities of the proposed system at its best. The swarm is capable to disintegrate when needed, thus providing safety of its individual UAVs and navigates to the common motion goal. Fig. 53 and 55 illustrate distance to the closest obstacle of each UAV in these simulations. A video of a Gazebo simulation of a swarm composed of 6 UAVs in world with the same obstacle distribution as on Fig. 52 can be found at `https://youtu.be/yDyiwVx0xB4`.

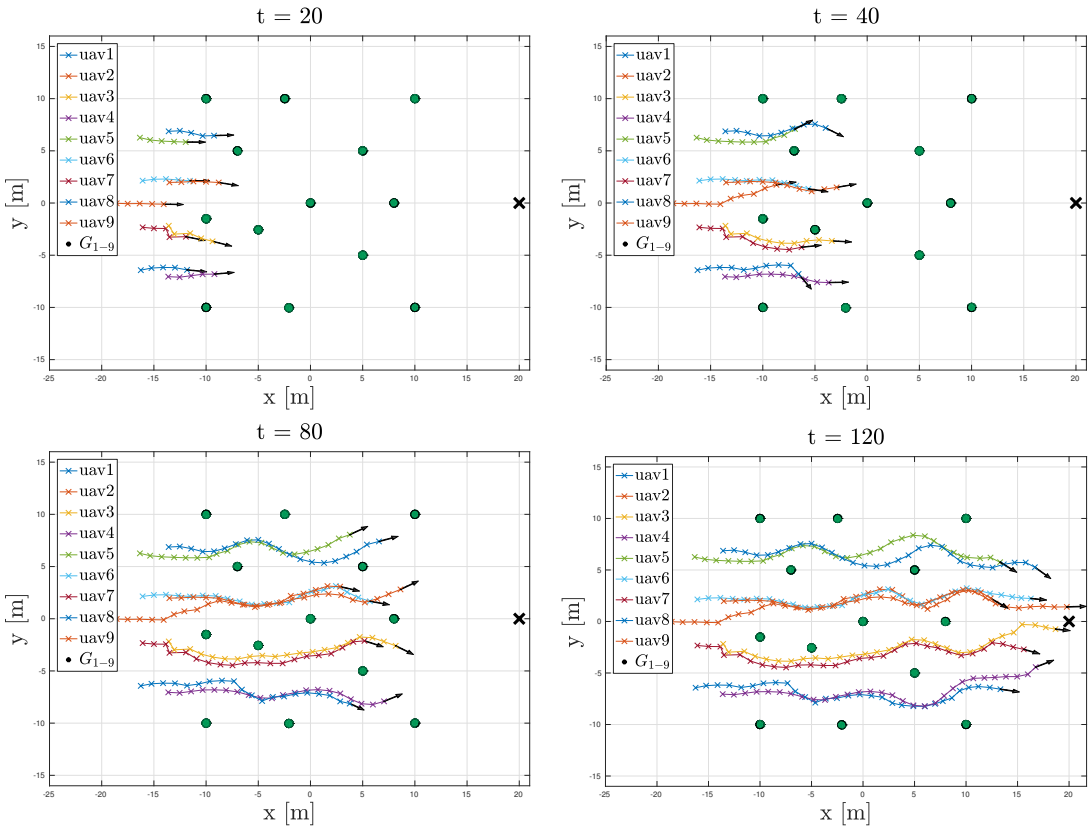


Figure 52 A swarm composed of 9 UAVs navigating through a forest-like planar environment.

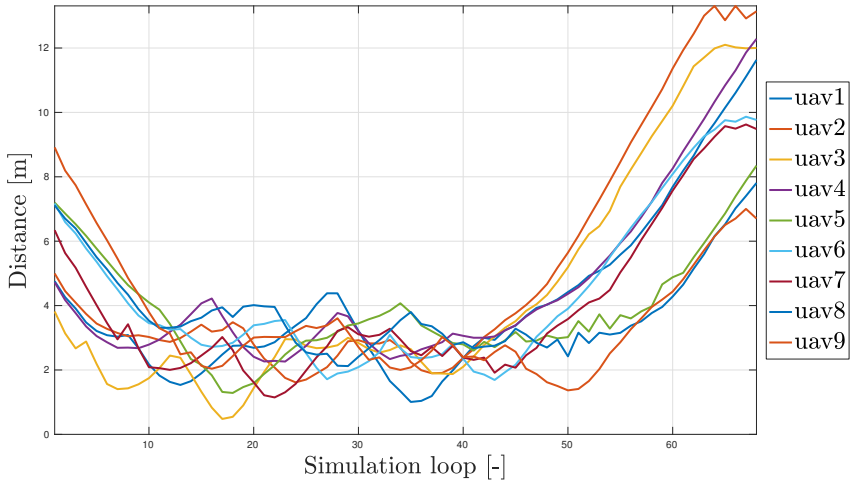


Figure 53 Distance to the closest obstacle during simulation on Fig. 52.

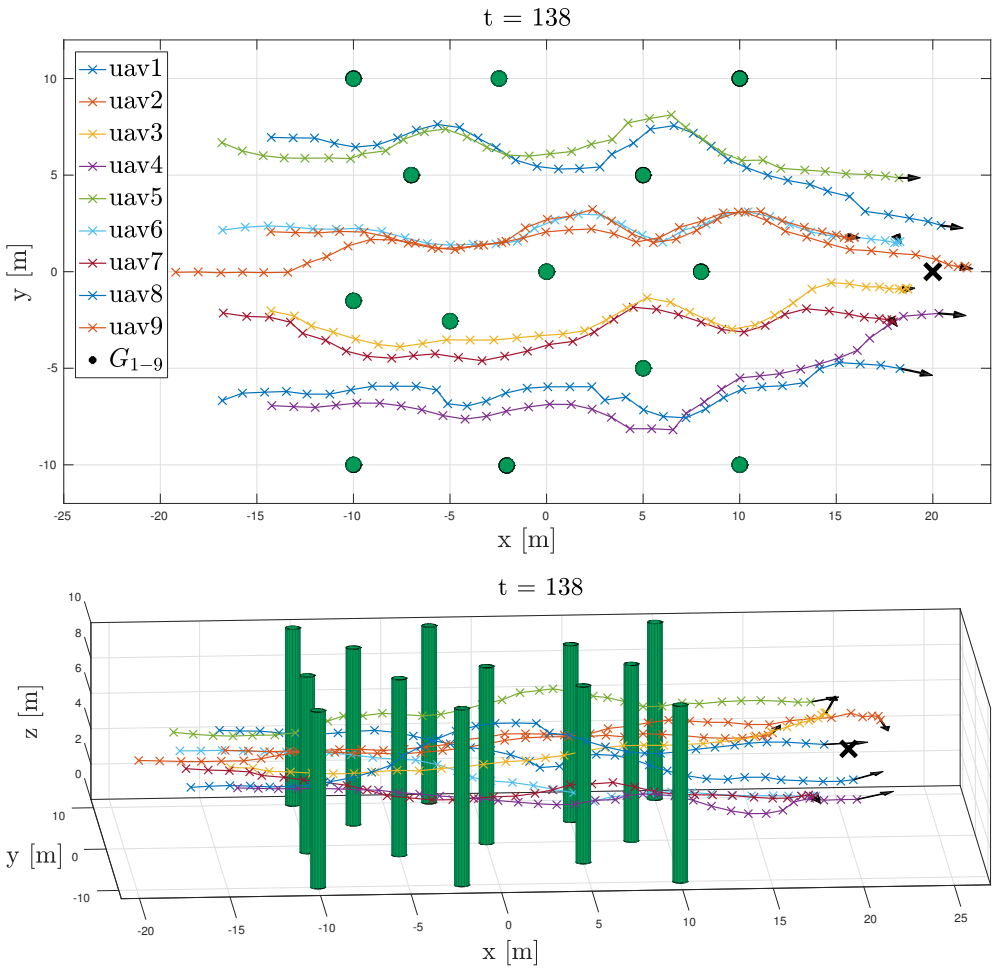


Figure 54 A swarm composed of 9 UAVs navigating through a forest-like environment in 3D space. A video of a Gazebo simulation of a swarm composed of 6 UAVs in world with the same obstacle distribution can be found at `https://youtu.be/yDyiwVx0xB4`.

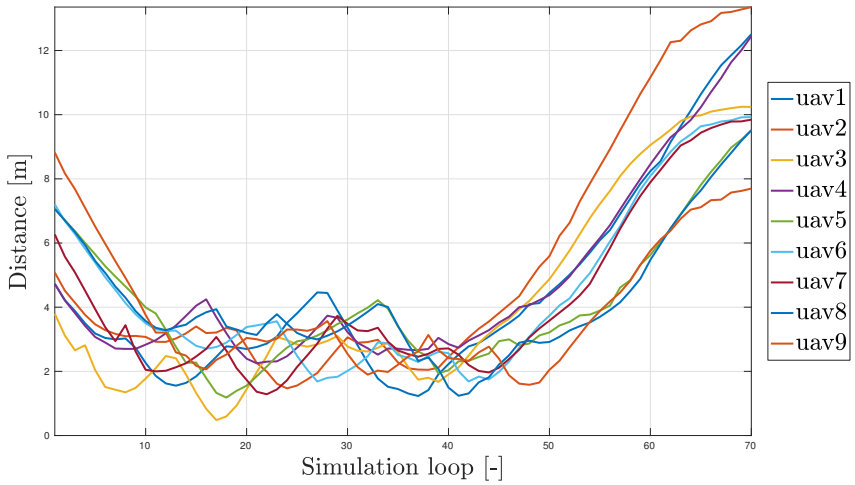


Figure 55 Distance to the closest obstacle during simulation on Fig. 54.

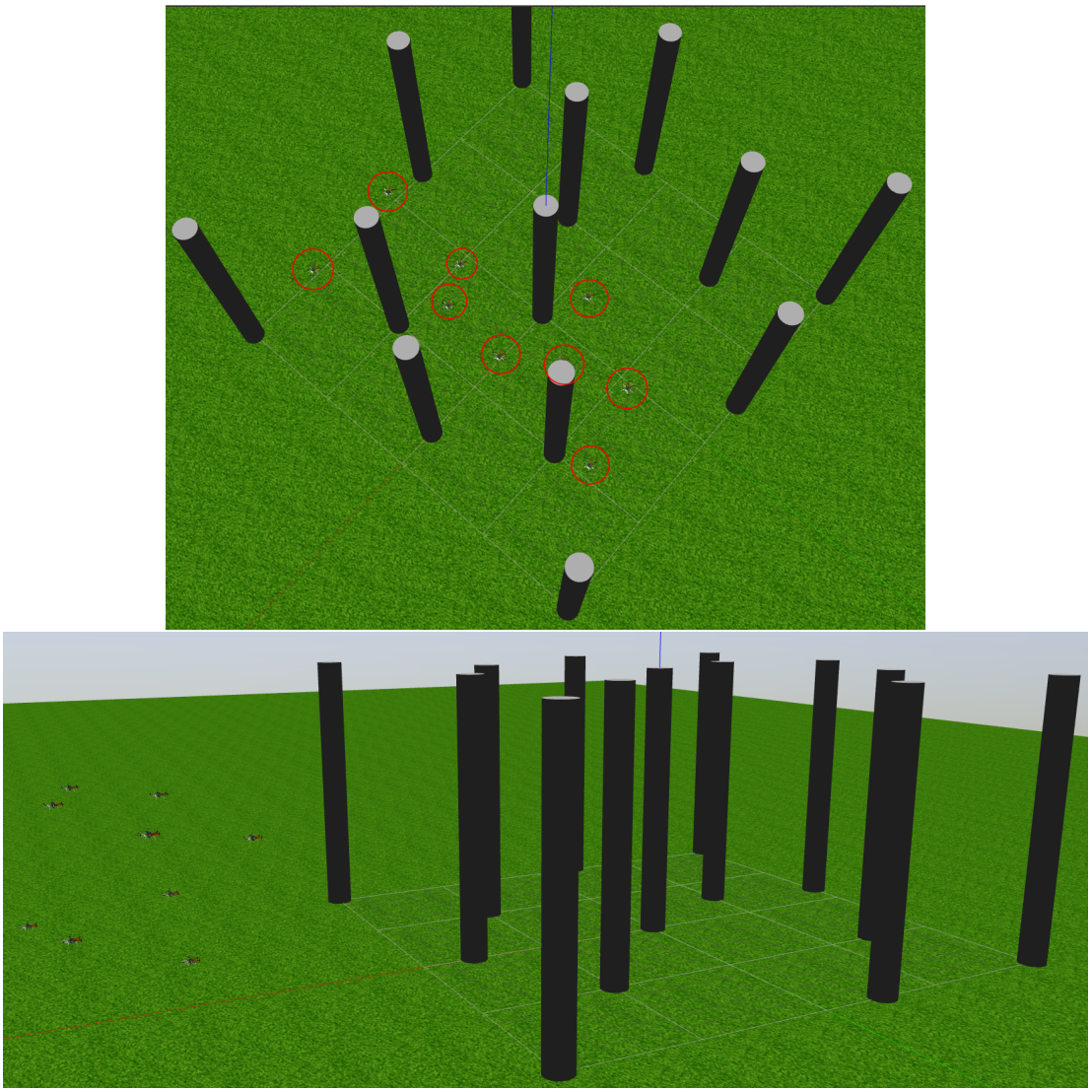


Figure 56 Visualization of the simulation with the simplified forest-like environment on Fig. 54 in the Gazebo simulator.

4.4 Table of simulation parameters

The simulations in the section 4 were performed with parameters described in Table 1. A conclusion of this table is, that for an environment filled with obstacles it is necessary to choose constants k_1 and k_2 wisely to primarily ensure safety, but also functionality of the swarm. The loop rate was set to 5 Hz and loop distance to 0.75 m in all the simulations.

Simulation figure	r [m]	k_1 [-]	k_2 [-]	c_1 [-]	c_2 [-]	d [m]	w [m]	b_1 [-]	b_2 [-]
28	10	10	X	X	X	4	X	10	4
29	10	10	X	X	X	4	X	10	4
33	15	10	X	X	X	8	X	10	4
35	10	10	X	X	X	4	X	10	4
36	15	10	X	0.25	1.15	8	X	10	4
38	15	10	X	0.25	1.15	8	X	10	4
40	10	12.5	15	0.25	1.15	8	6	10	4
41	10	12.5	15	0.25	1.15	8	6	10	4
42	10	15	12.5	0.15	1.05	8	6	10	4
43	15	10	17.5	0.15	1.05	8	8	10	4
44	10	12.5	15	0.25	1.15	8	6	10	4
45	10	12.5	15	0.25	1.15	8	6	10	4
46	10	10	17.5	0.25	1.15	8	6	10	4
48	10	10	12.5	0.15	1.05	8	6	10	4
49	10	10	12.5	0.15	1.05	8	6	10	4
50	10	10	12.5	0.15	1.05	8	6	10	4
51	10	10	12.5	0.15	1.05	8	6	10	4
52	7.5	14	25	0.15	0.9	8	4	10	4
54	7.5	14	25	0.15	0.9	8	4	10	4

Table 1 Parameters of the swarming model used in simulations.

5 Swarm deployment

After successful numerical verification of the algorithms in section 4, the system proposed in section 3 was verified by a deployment of a real swarm of UAVs. A forest-like environment showed on Fig. 57 was created from a set of cylindrical obstacles. Parameters of the cylindrical obstacles, showed on Fig. 16a, are $R_c = 0.3\,\mathrm{m}$ and $H_c = 1.8\,\mathrm{m}$. These parameters, together with absolute position of the obstacles, were given to the system as an input. Further, the system was integrated to a platform of UAVs provided by the MRS group and released to a flight through the forest-like environment.

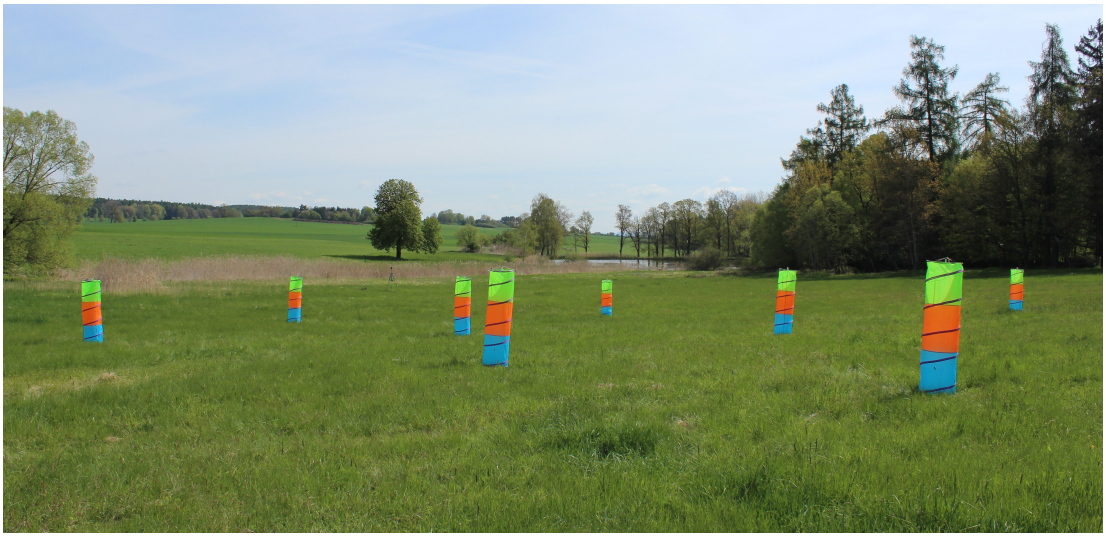


Figure 57 A forest-like environment modeled for the first deployment of a swarm of UAVs.

Three UAVs were used for the deployment. The flight through the modeled forest was performed three times with the same initial positions of the UAVs, the same motion goal and the same parameters (given by Table 2). All the flights were performed with constant altitude 1.6 m and successfully resulted to a stabilized motion of the swarm through the forest-like environment with similar trajectories.

Trajectories from the first two experimental deployments, illustrated on Fig. 59, indicate functionality of the swarm in a complex environment very similar to the simulation results. A stabilized and safe motion is provided for the whole flight between the UAVs and the obstacles, as illustrated on Fig. 60 and 61. These figures show, that each UAV maintained distance safety at least 2 m from all the other UAVs and obstacles. After navigation through the forest-like environment the swarm oscillated at the motion goal, where the UAVs wrestled with each other to minimize their distance to the desired position.[1]

Fig. 58 shows four snapshots from the third experimental deployment in the forest-like environment.[2] Each corner of these snapshots contains detailed data about the

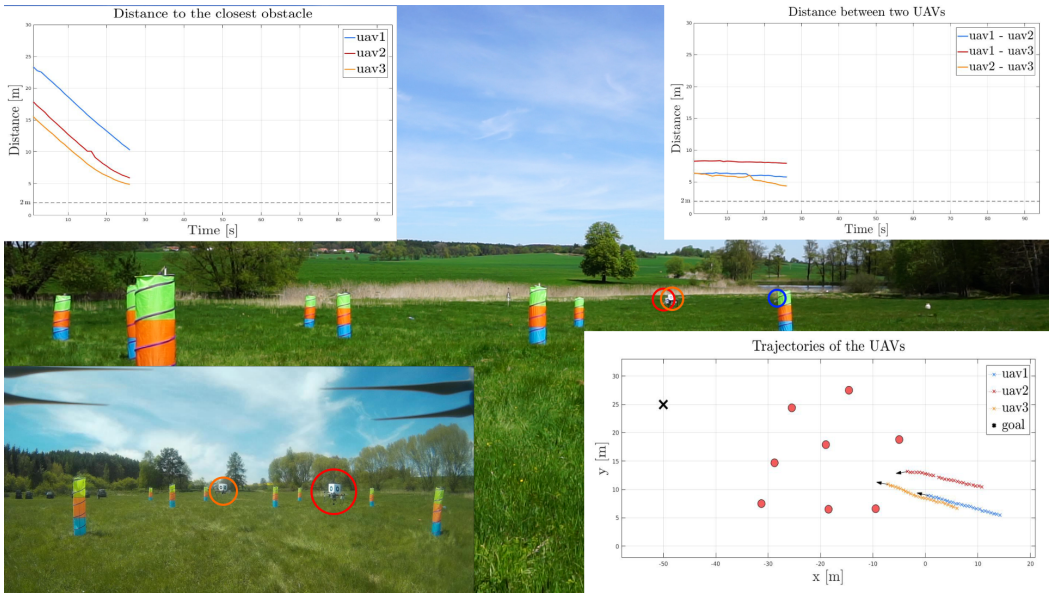
[1] Full videos from the first two experimental deployments can be found at `https://youtu.be/e53Y-JOPrDU` and `https://youtu.be/WzcLwoGwcME`.

[2] Full video of the third experimental deployment can be found at `https://youtu.be/hqHW6jYTBEY`.

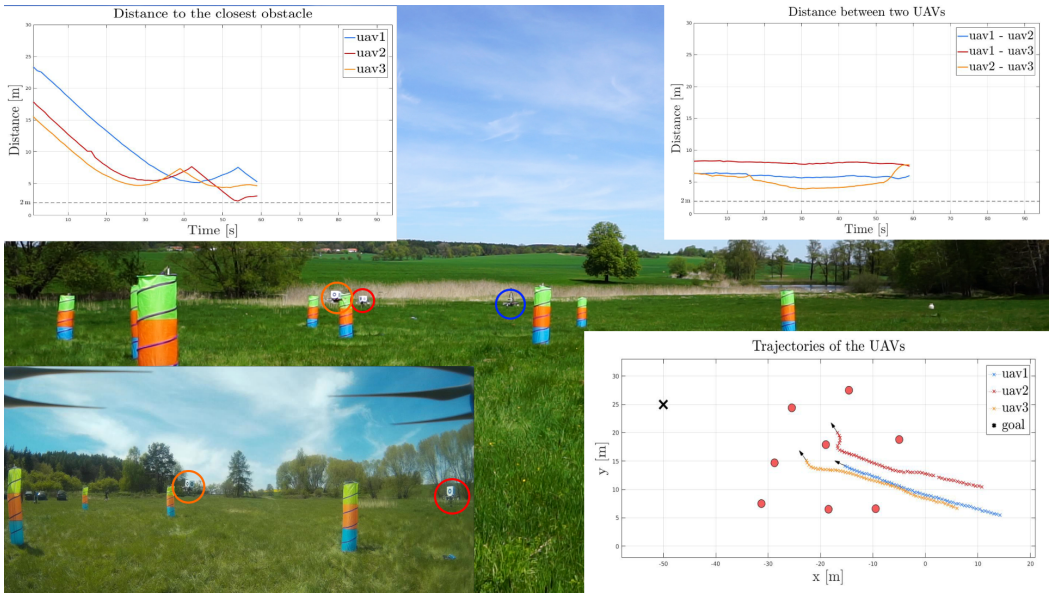
swarm:

- Bottom left corner - a snapshot from the front horizontal camera of the third UAV (highlighted by blue color). Note that the localization patterns have no effect on the swarm, since the precise GPS has been used for localization. Further onboard camera snapshots are shown on Fig. 62.
- Bottom right corner - trajectory of the swarm from start to time t (s).
- Top left corner - distance of each UAV to its closest obstacle.
- Top right corner - individual distances between all the UAVs.

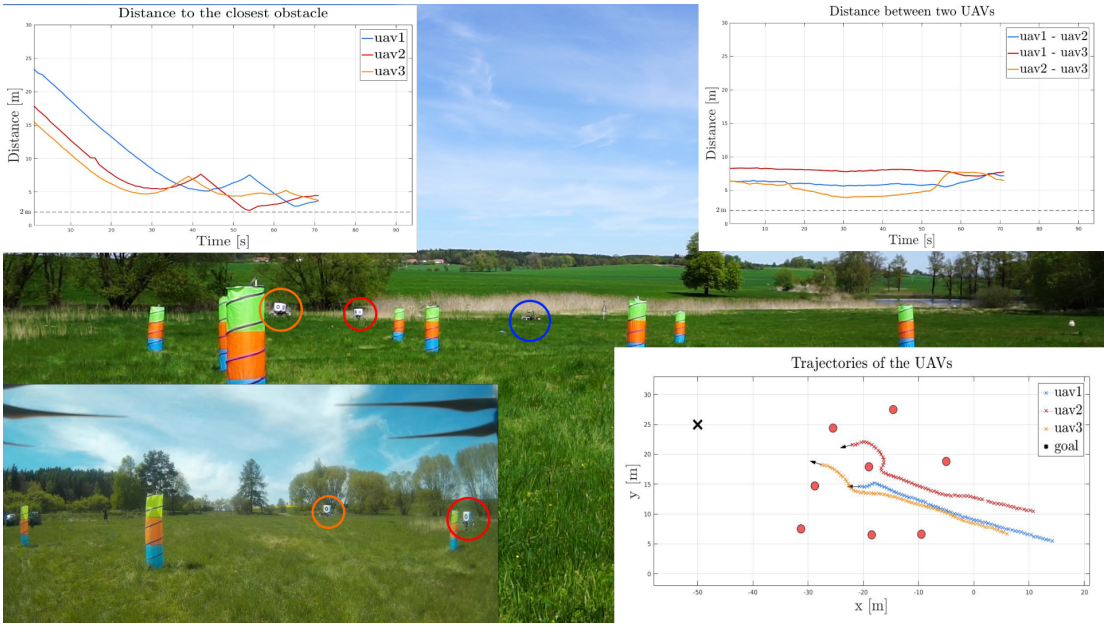
The top two corners of these snapshots show, that all the UAVs have maintained a distance safety at least 2 m to all the other UAVs and obstacles.



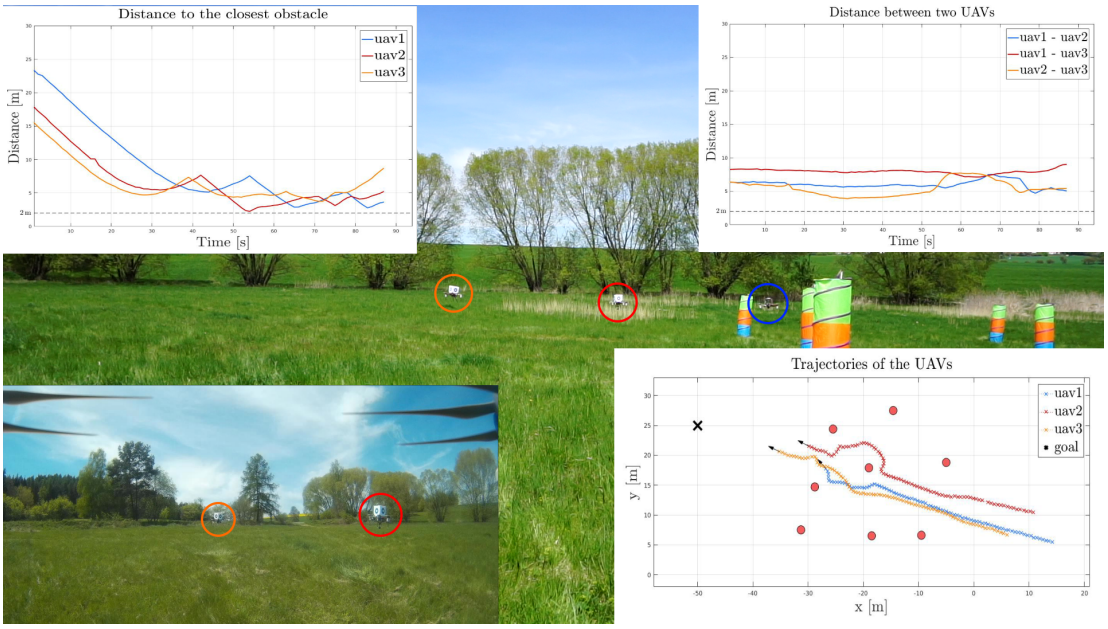
(a) $t = 25\,\mathrm{s}$



(b) $t = 59\,\mathrm{s}$

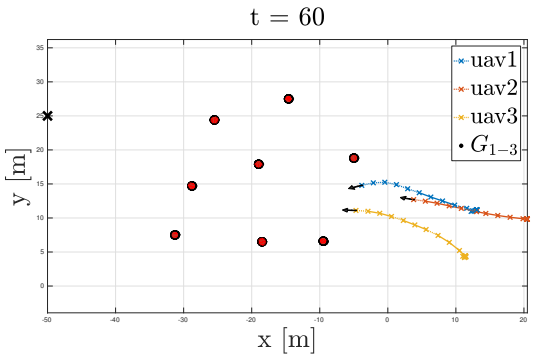


(c) $t = 71\,\mathrm{s}$

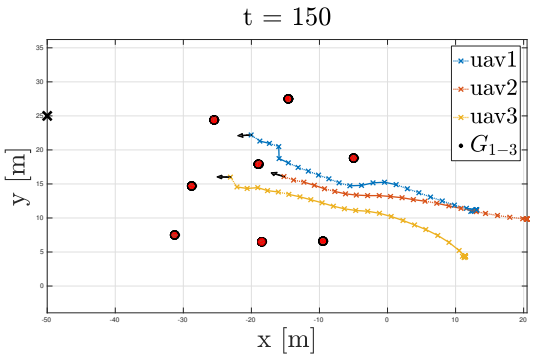


(d) $t = 87\,\mathrm{s}$

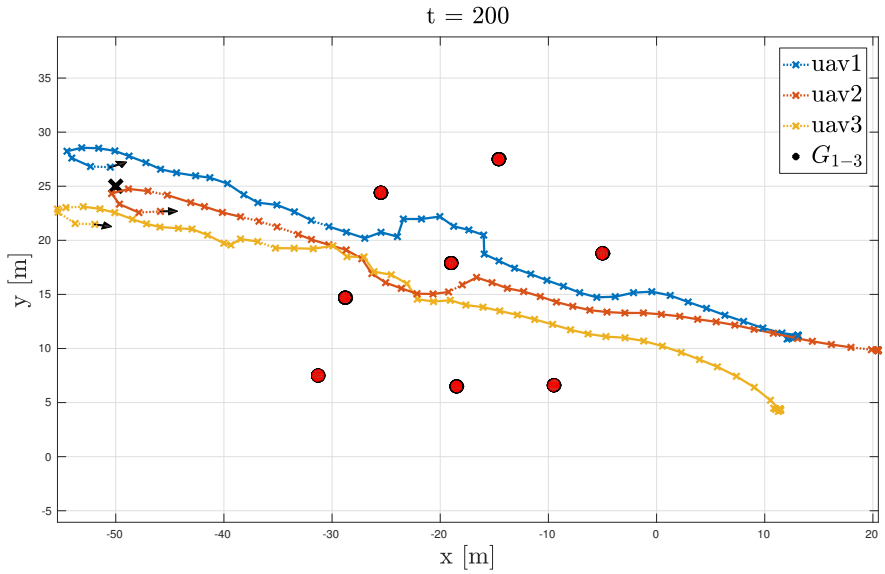
Figure 58 Snapshots from the third experimental deployment of a swarm composed of 3 unmanned and fully autonomous helicopters. Full video can be found at `https://youtu.be/hqHW6jYTBEY`.



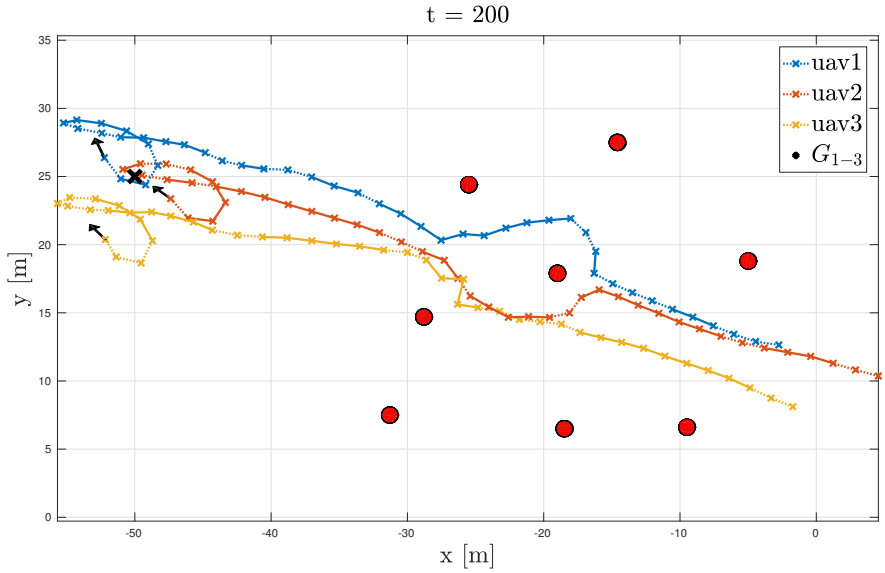
(a) First deployment after $t = 60$.



(b) First deployment after $t = 150$.

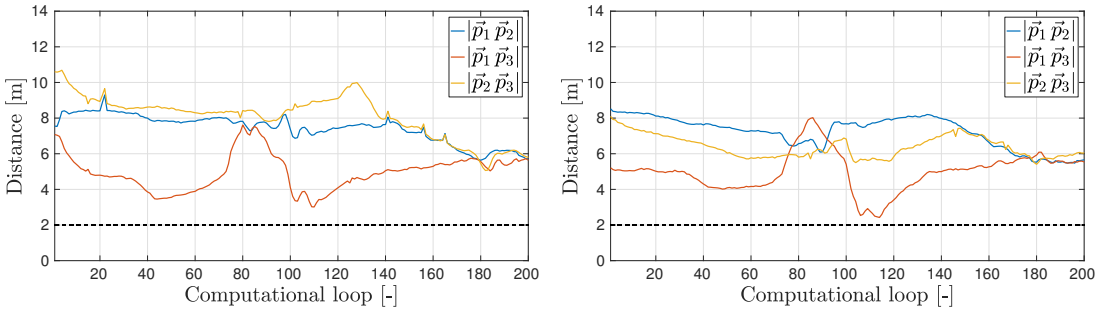


(c) First deployment. Full video can be found at `https://youtu.be/e53Y-JOPrDU`.



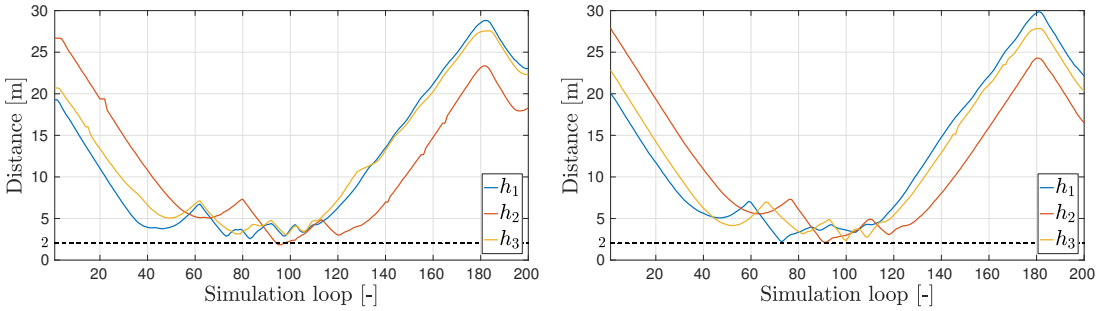
(d) Second deployment. Full video can be found at `https://youtu.be/WzcLwoGwcME`.

Figure 59 Trajectories of a swarm composed of 3 UAVs during the first two executions of the experimental deployment.



(a) First deployment. **(b)** Second deployment.

Figure 60 Distance between two UAVs during the first two experimental deployments, whose trajectories are shown on Fig. 59.



(a) First deployment. **(b)** Second deployment.

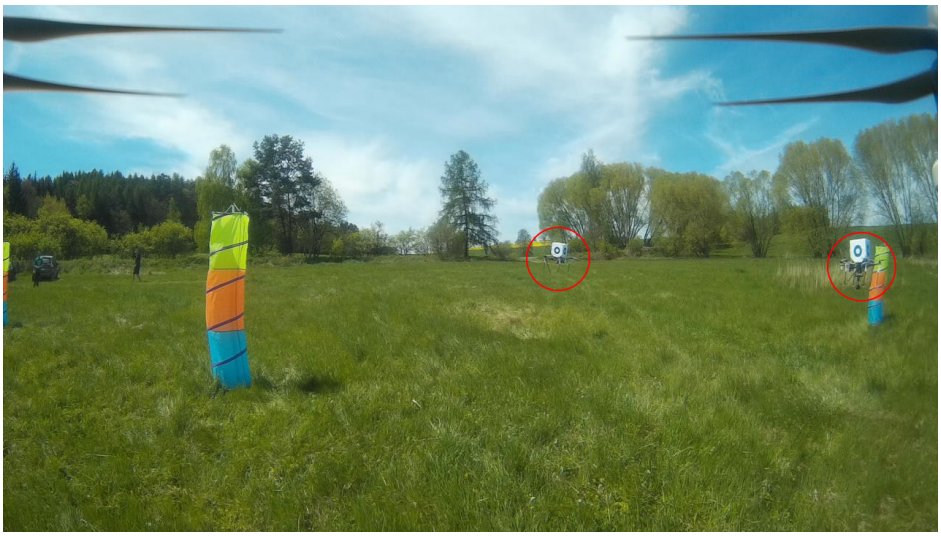
Figure 61 Distance to the closest obstacle during the first two experimental deployments, whose trajectories are shown on Fig. 59.

r [m]	k_1 [-]	k_2 [-]	c_1 [-]	c_2 [-]	d [m]	w [m]	b_1 [-]	b_2 [-]
15	13.5	16	0.2	0	8	8	10	X

Table 2 Simulation parameters.

(a) Prior running the swarming controller.

(b) Splitting due to the obstacle on the right.



(c) Merging after the center obstacle bypass.

(d) After successful flight through the forest.

Figure 62 Images from the front vertical camera on one UAV during the third experimental deployment of a swarm composed of 3 UAVs in a forest-like environment.

6 Relative localization precision and reliability

Studying the influence of the relative localization on the swarming system is an important part of the research. This section finds margins in the system behavior, which could lead to a loss of the properties of the swarming behavior. The most important margin to find is the loss of security in safe distance between individual UAVs or UAVs and obstacles. Further margins to examine are loss of self-optimizing and integrated behavior characterizing swarming system.

In section 3.4 usage of relative onboard localization utilizing the GPS was introduced. This section studies an influence of the GPS localization used for simulations and real deployment of a swarm in this thesis. Preliminary for usage of precise GPS is an outdoor environment to reduce signal reflection and inaccuracy. However, the GPS is never absolutely precise and some inaccuracy is always present in the localization data.

6.1 Inaccuracy of positioning system

The localization procedure is heavily dependent on accuracy of the GPS system. Differential GPS *Tersus GNSS* used for the UAV platform, described in section 2.3, achieves, according to the manufacturer, centimeter-level accuracy positioning [47]. With such precise localization, swarming behavior is affected only with negligible error. However, accuracy of the GPS system is affected by environment (sun, buildings, metal obstacles, etc.), weather [48] and other effects. These external affects can suddenly influence the system to corrupt its functionality. Therefore, in developed positioning system, artificial inputs of localization error to the GPS data was added to simulate conditions, where positioning system is affected by non-negligible error.

Localizing of absolute position $\vec{p}_0$ of an UAV h_0 is noised with error, therefore

$$\vec{p}_0 = \vec{\hat{p}}_0 + \vec{\delta}, \tag{44}$$

where $\vec{\hat{p}}_0$ is a precise position, which can never be truly measured, and

$$\vec{\delta} = \vec{\delta}_e + \vec{\delta}_a. \tag{45}$$

The vector $\vec{\delta}$ is a localization error, where $\vec{\delta}_e$ is a nondeterministic inaccuracy error of the GPS, which we are unable to measure and assume is negligible, and $\vec{\delta}_a$ is an artificial inaccuracy error added to the odometry data. The error $\vec{\delta}_a$ represents sudden or permanent non-negligible increase of the GPS error $\vec{\delta}_e$. The error $\vec{\delta}_a$ was distributed to the odometry data with usage of the *uniform distribution*, defined as

$$f(|\vec{\delta}_a|) = \frac{1}{\delta_a} \quad \forall\, |\vec{\delta}_a| \in \langle 0,\, \delta_a \rangle, \tag{46}$$

where $f(x)$ is a probability density function, $\vec{\delta}_a$ is an uniformly distributed error vector with its absolute value from interval $\langle 0,\, \delta_a \rangle$ and $\delta_a > 0$ is a maximal absolute value

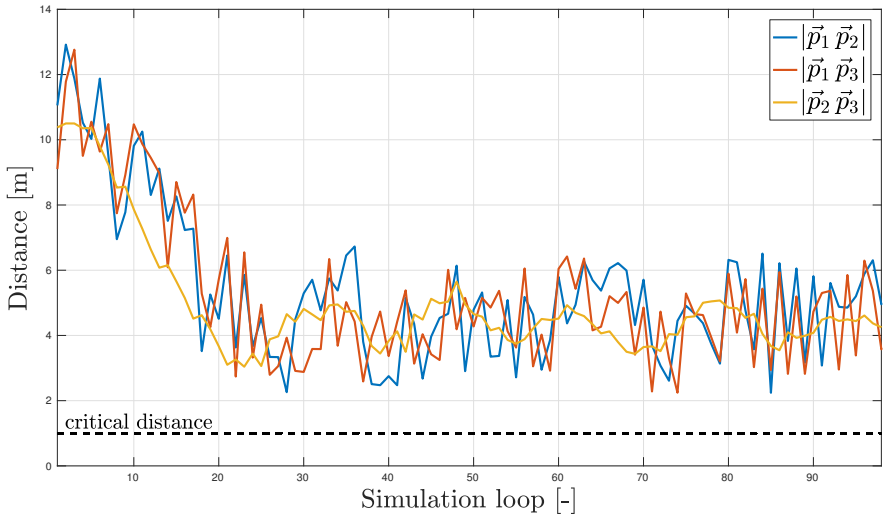
of the inaccuracy error. The value of δ_a is then a margin of changes in the swarming behavior to find. In further sections, we define a critical margin of the safety distance of an UAV to be 1 m. Thus the error δ_a is determined as marginal, if the position between the UAV and any other UAV or an obstacle drops below the 1 m critical borderline.

Further simulations with an artificial localization error of different values were performed. These simulations in a forest-like environment were performed in a world shown on Fig. 56. An important assumption for these simulations is that an attribute of a swarm - safety and collision avoidance of its members, is the primary attribute to preserve. The swarm safety is highly dependent on parameters d and w, defined in Eq. (7) and Eq. (29), which adjust distances between the UAVs and obstacles. However, the safety is also dependent on the loop rate and distance, which define, whether the swarm is capable to react in time on a sudden disturbance. The parameters for further studies were set to: $d, w = 8$ m, loop rate 30 Hz and loop distance 2 m. Presented results in further sections 6.1.1 and 6.1.2 are valid only for the predefined parameters.

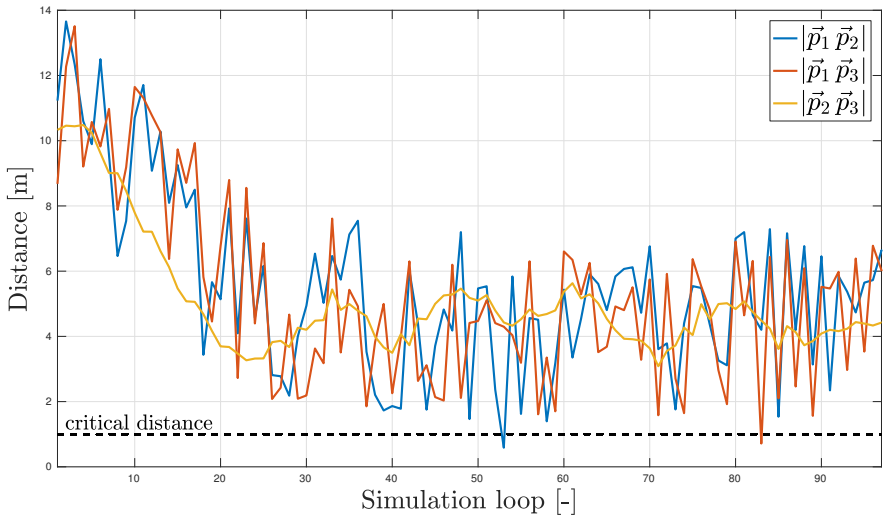
6.1.1 Permanent inaccuracy

Even the differential GPS, contained on the UAV platform of our group, has a permanent error $\vec{\delta}_e$ present. We assume that with an artificial error $\vec{\delta}_a = \vec{0}$, the error of the differential GPS $\vec{\delta}_e = \vec{\delta}$ is negligible. However in the real world, the GPS is not absolutely reliable, and disturbances or inaccuracies are always present. Permanent addition of the artificial error $\vec{\delta}_a$ simulates a condition, where $\vec{\delta}$ is non-negligible. The system was pushed to its margins, since localization errors in meters were added. Nowadays, such a high value of error is unlikely even for a commercial GPS.

Applying of the relative localization error δ_a only to one UAV h_1 leads to altering of the swarming behavior illustrated on Fig. 63 in a free environment and Fig. 64 in a forest-like environment. The swarming behavior is not optimal and the trajectories of the UAVs are oscillating. However, the swarm is capable to maintain safety behavior and optimal formation up to $\delta_a = 3$ m. This result is valid only for the predefined parameters. If the values of parameters d and w were lowered, the value of δ_a would decrease as well, and vice versa. With bigger localization error, probability of a collision magnificently rises, as seen on Fig. 63b, where distance between two UAVs drops under the critical distance value in the simulation loop 50-55 and 80-85.



(a) $\delta_a = 3$ m



(b) $\delta_a = 4\,\mathrm{m}$

Figure 63 Distances between 3 UAVs in free space, where the UAV h_1 is affected by a relatively high localization error δ_a.

A distance to the closest obstacle, while navigating of a swarm through a forest-like environment with only the UAV h_1 affected by an artificial error is illustrated on Fig. 64. Displayed distances show, that the trajectory of affected UAV h_1 is much more oscillating compared to the other UAVs. When navigating through such complex environment, emphasis on a relative localization is stronger, since there is more objects in the environment to stay safe from. Because of density of obstacles and UAVs, the margin value of δ_a is reduced, compared to the free space - the swarm was capable to safely navigate through the forest-like environment with error up to $\delta_a = 2\,\mathrm{m}$.

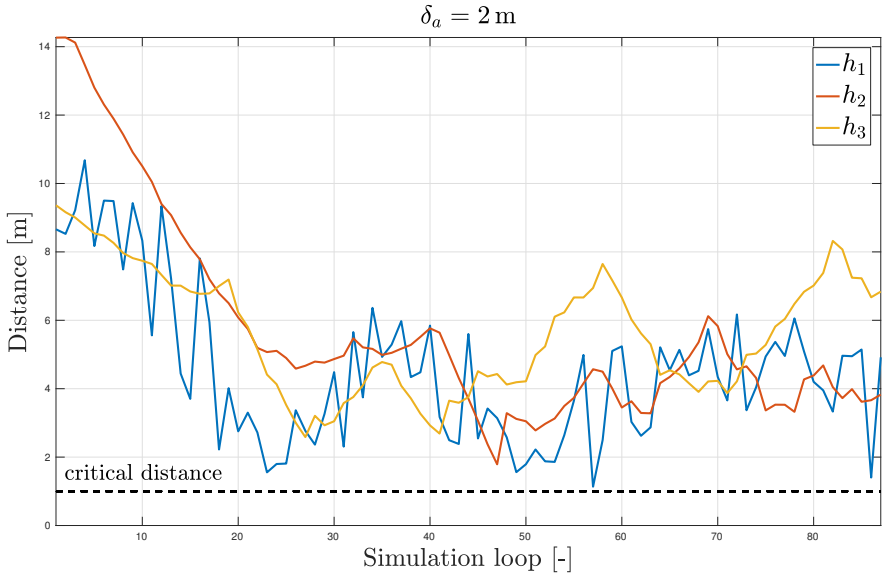
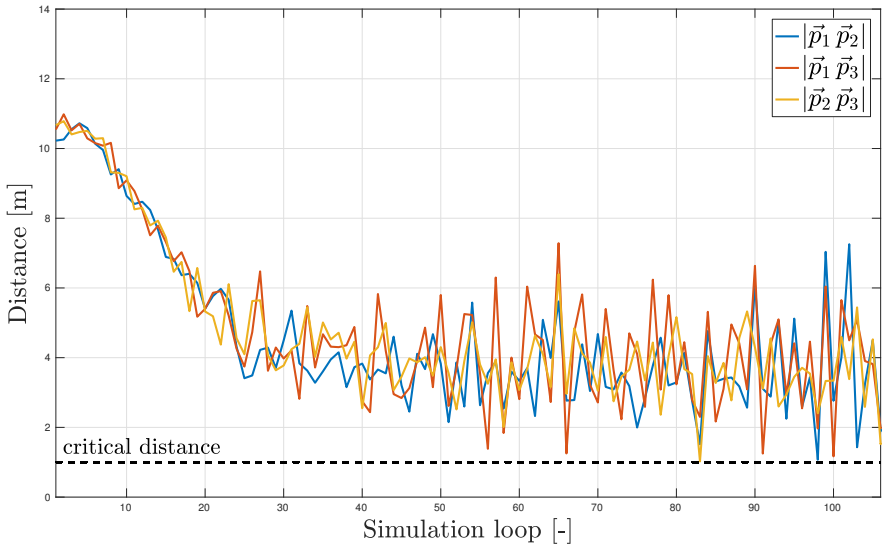
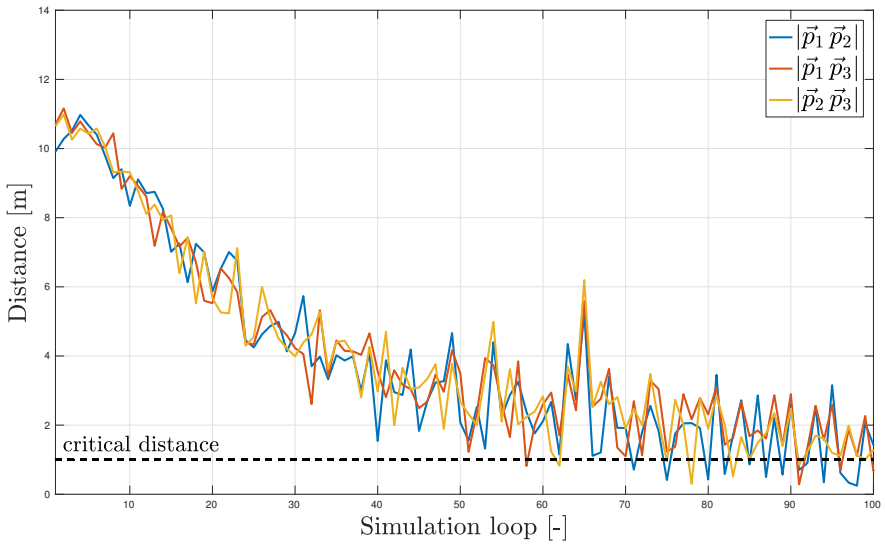


Figure 64 Distances to the closest obstacle in a forest-like environment, where the UAV h_1 is affected by a relatively high localization error $\delta_a = 2\,\mathrm{m}$.

The same experiments were performed for a swarm, where every UAV is affected by the error distribution. Inaccuracy of localization of all the UAVs is the most common for real systems. Fig. 65 illustrates distances between the 3 UAVs in the swarm. For localization error $\delta_a = 2\,\mathrm{m}$, the swarm was capable to ensure safety of its members with a relative reserve. However, the behavior is heavily oscillating, which affects the navigational speed and optimizing behavior. Therefore, as critical was determined localization error $\delta_a > 2\,\mathrm{m}$, as illustrated on Fig 65b, where the safety of the swarm cannot be assured. Compared to the simulation, where only one UAV is affected by the error distribution on Fig. 63, the margin error was determined lower. This margin value decrease, since the whole swarm has been affected by the error distribution.



(a) $\delta_a = 2\,\mathrm{m}$



(b) $\delta_a = 3\,\mathrm{m}$

Figure 65 Distances between 3 UAVs in free space, where each UAV is affected by relatively high localization error δ_a.

Navigating of a swarm through forest like environment with all the UAVs affected by the same error distribution is illustrated on Fig. 66. In a complex environment, the dependency on localization accuracy is more important. The margin error $\delta_a = 2\,\mathrm{m}$ for free space is too big for a complex environment, as seen on Fig 66. As a swarm flight in a real forest is our target motivation, we must put emphasize on the localization accuracy. The GPS in a real forest is highly unreliable to have localization error less than 2 m, therefore for such deployment, more precise relative localization has to be used.

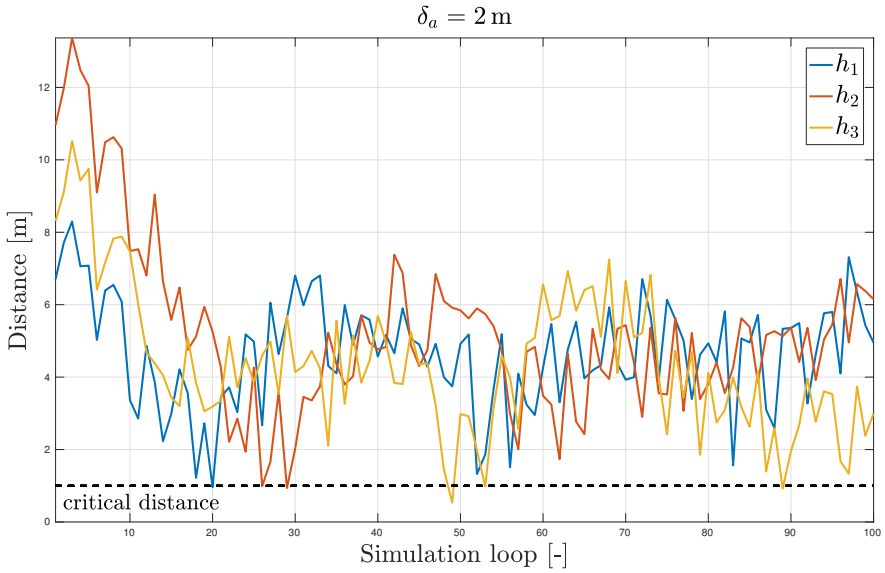


Figure 66 Distances to the closest obstacle in a forest-like environment, where each UAV is affected by a relatively high localization error $\delta_a = 2\,\mathrm{m}$.

6.1.2 Temporary inaccuracy

An important situation to study is a capability of the swarm to react on a sudden disturbance in a relative localization. Further simulations with various relative localization error δ_a were performed. Fig. 67 and 68 illustrate distances between UAVs, when one or all UAVs are affected by an localization error $\delta_a = 3\,\mathrm{m}$ between simulation loops 45 to 75. The behavior of the swarm is the same as in section 6.1.1, when the localization error influences the system. The swarm is capable to cope with a sudden localization error with a marginal error identical to the permanent inaccuracy in section 6.1.1 with a more extensive distance reserve.

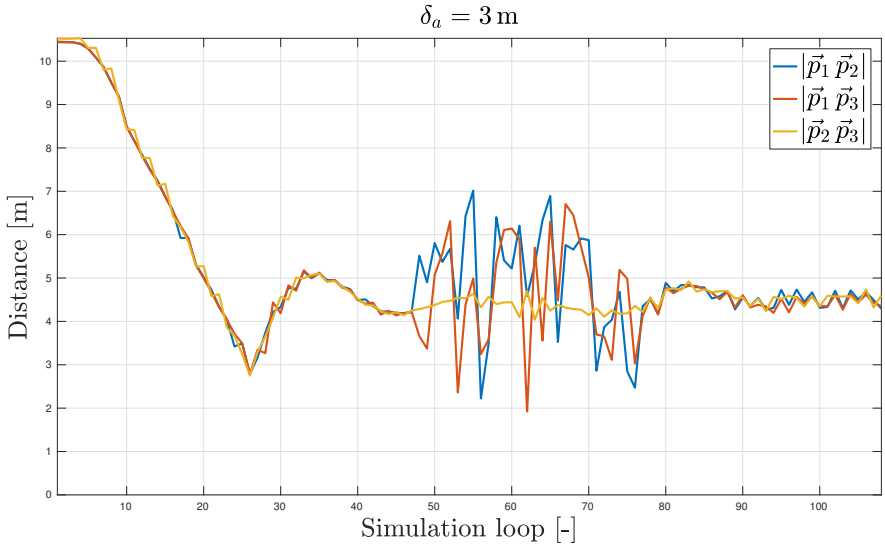


Figure 67 Distances between 3 UAVs in free space, where only the UAV h_1 is temporarily affected by a relatively high localization error $\delta_a = 3\,\mathrm{m}$ between simulation loops 45 to 75.

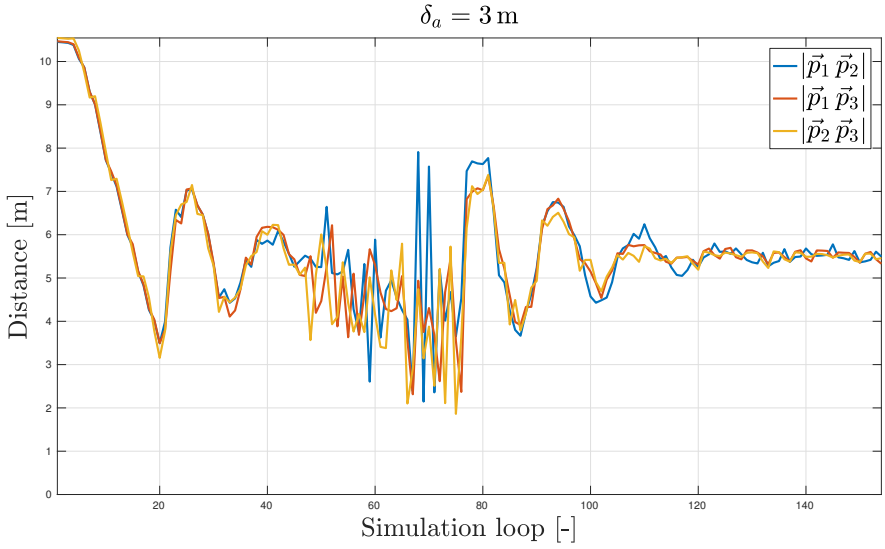


Figure 68 Distances between 3 UAVs in free space, where each UAV is temporarily affected by a relatively high localization error $\delta_a = 3\,\mathrm{m}$ between simulation loops 45 to 75.

6.2 Impact on swarming behavior

Different marginal localization errors in simulations for free and a forest-like environment have been determined in section 6.1. The simulations show, that with the predefined parameters, the swarm can be influenced by a localization error up to 3 m in a free environment and 2 m in a forest-like environment. These results are valid only for the predefined parameters. If the values of parameters d and w were lowered, the value of the localization error would decrease as well, and vice versa. Nevertheless, during a swarm deployment in an environment with unreliable and inaccurate global localization, we have two options of relative onboard localization.

The first option is to preserve usage of the global localization (as we do in this thesis), thus expand the distances between the UAVs and the obstacles by increasing the swarming parameters d and w, defined in Eq. (7) and Eq. (29). However, these parameters are limited, since both have been defined to lay in interval d, $w \in (0, r\rangle$ (m). The distance r (m) provides a size of a neighborhood, where an UAV is capable to localize other UAVs and obstacles, and is dependent on capabilities of the communication network between the UAVs. Another limitation for these parameters is density of obstacles in an obstacle-filled environment. The density limits the UAVs to fly closer together, thus blocking the requirement of larger spacing between them.

The second option is to use another relative localization, which is not dependent on a mutual communication. Such localization must provide reliable accuracy with error values less than the ones determined in section 6.1 for the same predefined parameters. Usage of such relative localization, described in section 3.4, is part of a future research.

7 Conclusion

In this thesis, we have developed a decentralized and robust system for control and stabilization of a swarm of UAVs. The system has been successfully implemented in ROS and integrated to a platform of UAVs of the MRS group at FEE CTU. The proposed system has been verified by various simulations and experiments in a complex environment.[1] The entire assignment of this thesis has been fulfilled successfully. According to the assignment, following tasks have been completed:

- A decentralized algorithm for control of a swarm of unmanned aerial vehicles in an environment with obstacles has been designed in section 3 and further implemented in ROS.
- The multi-UAV platform developed by the MRS group for the MBZIRC 2017 competition has been understood, described in section 2.3, and used for further development.
- An obstacle avoidance function has been designed in section 3.2, and further integrated to the swarming algorithm.
- A verification of the system has been performed in the Gazebo simulator in section 4.
- A study of influence of reliability and precision of relative localization has been performed in section 6.
- A successful experimental verification with the UAV platform of the MRS group has been performed and described in section 5.

7.1 Future work

This work has created a base for a swarm control during a future research. The research will contain these specific areas connected with the swarming behavior:

- Development and integration of obstacles localization method.
- Integration of the visual relative localization developed by our group [24][27][44].
- Integration of the optical flow method, described in section 3.4, for the swarm stabilization in GPS-denied environments.
- Real experiments with an extensive number of deployed UAVs.
- Swarm trajectory planning and navigation.
- Development of different swarming models and their comparison with the model proposed in this thesis.
- Flight through a real forest being our target motivation.

[1] Videos of the system simulation and experimental verification can be found at `http://yt.vu/p/PLooTKzV6hvpNF3bTfiOuMZbr2n_tGw0td`.

Bibliography

[1] Craig W. Reynolds. "Flocks, Herds, and Schools: A Distributed Behavioral Model". In: *Computer Graphics, Volume 21, Number 4, July 1987* (1987).

[2] A. P. Gerdelan. *Behavioural AI in Movies and Games: The Last 20 Years*. Tech. rep. CSTN-105. Albany, North Shore 102-904, Auckland, New Zealand: Computer Science, Massey University, Aug. 2010. URL: http://cssg.massey.ac.nz/cstn/105/cstn-105.pdf.

[3] Holger Danielsiek et al. "Intelligent Moving of Groups in Real-Time Strategy Games". IEEE Symposium on Computational Intelligence and Games. 2008.

[4] J. Pugh and A. Martinoli. "Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization". In: *2007 IEEE Swarm Intelligence Symposium*. Apr. 2007, pp. 332–339. DOI: 10.1109/SIS.2007.367956.

[5] Jim Pugh and Alcherio Martinoli. "Multi-robot Learning with Particle Swarm Optimization". In: *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*. AAMAS '06. Hakodate, Japan: ACM, 2006, pp. 441–448. ISBN: 1-59593-303-4. DOI: 10.1145/1160633.1160715. URL: http://doi.acm.org/10.1145/1160633.1160715.

[6] Gabriel T. Bugajski. "Architectural Considerations for Single Operator Management of Multiple Unmanned Aerial Vehicles". English. Master's thesis. 2010. URL: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA518058.

[7] J. S. Jennings, G. Whelan, and W. F. Evans. "Cooperative search and rescue with a team of mobile robots". In: *Advanced Robotics, 1997. ICAR '97. Proceedings., 8th International Conference on*. July 1997, pp. 193–200. DOI: 10.1109/ICAR.1997.620182. URL: http://ieeexplore.ieee.org/abstract/document/620182/.

[8] Ahmad Baranzadeh. "Decentralized Autonomous Navigation Strategies for Multi-Robot Search and Rescue". In: *CoRR* abs/1605.04368 (2016). URL: http://arxiv.org/abs/1605.04368.

[9] P. Santana et al. "A multi-robot system for landmine detection". In: *2005 IEEE Conference on Emerging Technologies and Factory Automation*. Vol. 1. Sept. 2005, pages. DOI: 10.1109/ETFA.2005.1612597. URL: http://ieeexplore.ieee.org/abstract/document/1612597.

[10] M. Saska et al. "Swarm Distribution and Deployment for Cooperative Surveillance by Micro-Aerial Vehicles". In: *Journal of Intelligent & Robotic Systems*. 84.1 (2016), pp. 469–492. URL: http://mrs.felk.cvut.cz/data/papers/SaskaJINT2016.pdf.

[11] Alexander G. Madey and Gregory R. Madey. "Design and evaluation of UAV swarm command and control strategies." In Proceedings of the Agent-Directed Simulation Symposium (ADSS 13). 2013.

[12] E. M. H. Zahugi, A. M. Shabani, and T. V. Prasad. "Libot: Design of a low cost mobile robot for outdoor swarm robotics". In: *2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. May 2012, pp. 342–347. DOI: 10.1109/CYBER.2012.6392577. URL: http://ieeexplore.ieee.org/abstract/document/6392577/.

[13] F. Arvin et al. "Development of an autonomous micro robot for swarm robotics". In: *2014 IEEE International Conference on Mechatronics and Automation*. Aug. 2014, pp. 635–640. DOI: 10.1109/ICMA.2014.6885771. URL: http://ieeexplore.ieee.org/abstract/document/6885771/.

[14] Caroline Perry. *A self-organizing thousand-robot swarm*. 2014. URL: http://www.seas.harvard.edu/news/2014/08/self-organizing-thousand-robot-swarm.

[15] W. Truszkowski et al. "NASA's swarm missions: the challenge of building autonomous software". In: *IT Professional* 6.5 (Sept. 2004), pp. 47–52. ISSN: 1520-9202. DOI: 10.1109/MITP.2004.66. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1362627.

[16] M. Saska et al. "Coordination and Navigation of Heterogeneous MAV-UGV Formations Localized by a 'hawk-eye'-like Approach Under a Model Predictive Control Scheme". In: *International Journal of Robotics Research* 33.10 (2014), pp. 1393–1412.

[17] M. Saska et al. "Fault-Tolerant Formation Driving Mechanism Designed for Heterogeneous MAVs-UGVs Groups". In: *Journal of Intelligent and Robotic Systems* 73.1-4 (2014), pp. 603–622.

[18] 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.

[19] 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.

[20] M. Saska et al. "Navigation, Localization and Stabilization of Formations of Unmanned Aerial and Ground Vehicles". In: *International Conference on Unmanned Aircraft Systems (ICUAS)*. 2013.

[21] M. Saska et al. "Coordination and Navigation of Heterogeneous UAVs-UGVs Teams Localized by a Hawk-Eye Approach". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012.

[22] M. Saska, T. Baca, and D. Hert. "Formations of Unmanned Micro Aerial Vehicles Led by Migrating Virtual Leader". In: *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 2016.

[23] M. Saska et al. "Documentation of Dark Areas of Large Historical Buildings by a Formation of Unmanned Aerial Vehicles using Model Predictive Control". In: *IEEE ETFA*. 2017.

[24] M. Saska et al. "System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization". In: *Autonomous Robots. First online.* (2016). URL: http://mrs.felk.cvut.cz/data/papers/AURO2016.pdf.

[25] M. Saska et al. “Autonomous Deployment of Swarms of Micro-Aerial Vehicles in Cooperative Surveillance”. In: *International Conference on Unmanned Aircraft Systems (ICUAS)*. 2014.

[26] 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.

[27] Martin Saska. *MAV-swarms: unmanned aerial vehicles stabilized along a given path using onboard relative localization*. Czech Technical University in Prague Faculty of Electrical Engineering, Department of Cybernetics. 2014. URL: http://mrs.felk.cvut.cz/data/papers/icuas2015.pdf.

[28] 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. URL: http://mrs.felk.cvut.cz/data/papers/ICRA2014escape_short-joined.pdf.

[29] Tomáš Brich. “Motion Planning for Swarms of Unmanned Helicopters in Complex Environment”. Bachelor’s thesis. Czech Technical University in Prague Faculty of Electrical Engineering, Department of Cybernetics, 2016. URL: http://mrs.felk.cvut.cz/data/students/brichBP.pdf.

[30] Adam Třešňák. “Shape optimization of swarm of unmanned helicopters”. Bachelor’s thesis. Czech Technical University in Prague Faculty of Electrical Engineering, Department of Cybernetics, 2013. URL: http://mrs.felk.cvut.cz/data/students/tresnakBP.pdf.

[31] Jan Vakula. “Escape behavior in swarms of unmanned helicopters”. Bachelor’s thesis. Czech Technical University in Prague Faculty of Electrical Engineering, Department of Cybernetics, 2012. URL: http://mrs.felk.cvut.cz/data/students/vakulaBP.pdf.

[32] Y.M. Zhang et al. “Development of advanced {FDD} and {FTC} techniques with application to an unmanned quadrotor helicopter testbed”. In: *Journal of the Franklin Institute* 350.9 (2013), pp. 2396–2422. ISSN: 0016-0032. DOI: http://dx.doi.org/10.1016/j.jfranklin.2013.01.009. URL: http://www.sciencedirect.com/science/article/pii/S0016003213000264.

[33] Loianno G., Baca T., and Saska M. “Embedded Model Predictive Control of Unmanned Micro Aerial Vehicles.” In 21st International Conference on Methods and Models in Automation and Robotics (MMAR). 2016.

[34] J. Faigl et al. “Low-Cost Embedded System for Relative Localization in Robotic Swarms”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2013.

[35] T. Krajnik et al. “A Practical Multirobot Localization System”. In: *Journal of Intelligent & Robotic Systems* 76.3-4 (2014), pp. 539–562.

[36] Petr Všetečka. “Navigation and Stabilization of Swarms of Micro Aerial Vehicles in Complex Environment”. Bachelor’s thesis. Czech Technical University in Prague Faculty of Electrical Engineering, Department of Cybernetics, 2015. URL: http://mrs.felk.cvut.cz/data/students/vseteckaBP.pdf.

[37] Tomáš Sekanina. "Algorithms for navigation of swarm of unmanned helicopters and following a dynamic target in a complex environment". Bachelor's thesis. Czech Technical University in Prague Faculty of Electrical Engineering, Department of Cybernetics, 2014. URL: http://mrs.felk.cvut.cz/data/students/sekaninaBP.pdf.

[38] Vojtěch Pavlík. "Swarm intelligence applied in multi-robot applications". Bachelor's thesis. Czech Technical University in Prague Faculty of Electrical Engineering, Department of Cybernetics, 2012. URL: http://mrs.felk.cvut.cz/data/students/pavlikBP.pdf.

[39] Jan Bulušek. "Active and passive escape behavior designed for groups of autonomous helicopters in dynamic environment". Bachelor's thesis. Czech Technical University in Prague Faculty of Electrical Engineering, Department of Cybernetics, 2011. URL: http://mrs.felk.cvut.cz/data/students/bulusekBP.pdf.

[40] *Intro to ROS - General concepts.* Clearpath Robotics. 2017. URL: http://www.clearpathrobotics.com/guides/ros/Intro%20to%20the%20Robot%20Operating%20System.html (visited on 25/04/2017).

[41] *Exchange Data with ROS Publishers and Subscribers.* Mathworks. URL: https://www.mathworks.com/help/robotics/examples/exchange-data-with-ros-publishers-and-subscribers.html.

[42] Mark Schuurman. *A common starling (Sturnus vulgaris) flock in Lauwersmeer, the Netherlands.* National Geographic. 2011. URL: http://voices.nationalgeographic.com/2014/09/03/animals-weird-flocks-birds-spiders-snakes-science/ (visited on 03/09/2016).

[43] Reza Olfati-Saber. "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory". Technical Report CIT-CDS 2004-005. California Institute of Technology, Control and Dynamical Systems, 2004.

[44] Jan Valeš. "Integration of relative visual localization into the system of stabilization of swarms of autonomous helicopters". Bachelor's thesis. Czech Technical University in Prague Faculty of Electrical Engineering, Department of Cybernetics, 2014. URL: http://mrs.felk.cvut.cz/data/students/valesBP.pdf.

[45] Charlotte K. Hemelrijk and Hanno Hildenbrandt. "Some Causes of the Variable Shape of Flocks of Birds". In: *PLOS ONE* 6.8 (Aug. 2011), pp. 1–13. DOI: 10.1371/journal.pone.0022479. URL: https://doi.org/10.1371/journal.pone.0022479.

[46] Michele Ballerini et al. "Empirical investigation of starling flocks: a benchmark study in collective animal behaviour". In: *Animal Behaviour* 76.1 (2008), pp. 201–215. ISSN: 0003-3472. DOI: https://doi.org/10.1016/j.anbehav.2008.02.004. URL: http://www.sciencedirect.com/science/article/pii/S0003347208001176.

[47] Tersus-GNSS. *Official site of Tersus-GNSS.* Tersus-GNSS. 2017. URL: https://www.tersus-gnss.com/ (visited on 25/04/2017).

[48] Thierry Gregorius and Geoffrey Blewitt. "The Effect of Weather Fronts on GPS Measurements." The University of Newcastle upon Tyne. 1998. URL: https://pdfs.semanticscholar.org/6769/ac7e84acd0bbf16dbd3fedadf3ed153f35f6.pdf.