Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Control Engineering

A system for autonomous grasping and carrying of objects by a pair of helicopters

Bc. Petr Kohout

Supervisor: Ing. Martin Saska, Dr. rer. nat.
May 2017

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Control Engineering

DIPLOMA THESIS ASSIGNMENT

Student: **Kohout Petr**

Study programme: Cybernetics and Robotics
Specialisation: Systems and Control

Title of Diploma Thesis: **A system for autonomous grasping and carrying of objects by a pair of helicopters**

Guidelines:

The goal of the thesis is to design, implement in ROS (Robot Operating System), and experimentally verify in Gazebo simulator and in real experiments a predictive-control-based algorithm for cooperative carrying of large objects by a pair of unmanned aerial vehicles (UAV). An important part of the thesis is a mechanical design of a gripper for autonomous grasping of the objects. The following tasks will be solved:

1. To design, implement and experimentally test a system for grasping of objects with a metal surface using a permanent magnet and a servo mechanism for objects release after the desired location is achieved. To integrate a sensor that indicates a proper attachment of the object to the UAV.
2. To integrate the gripper into the UAV system being designed for the MBZIRC competition.
3. To design, integrate and verify in the Gazebo simulator a mechanism for synchronization of two model predictive controllers [3] being run in parallel on two UAVs grasping and carrying the object.
4. To design, integrate and verify in the Gazebo simulator a motion planning technique based on a unique combination of TRRT [1,2] and RRT-path [4] algorithms, which ensures a feasible solution in environments with narrow passages.
5. To prepare the system for experimental verification with the multi-UAV platform of the Multi-Robot Systems group [3] (the real experiment of the motion planning will be realized in a future work after the MBZIRC competition).

Bibliography/Sources:

[1] M. Manubens, D. Devaurs, L. Ros and J. Cortés, Motion Planning for 6-D Manipulation with Aerial Towed-cable Systems, Proceedings of Robotics: Science and Systems, 2013.
[2] L. Jaillet, J. Cortés and T. Siméon, Sampling-Based Path Planning on Configuration-Space Costmaps, IEEE Transactions on Robotics, vol. 26, no. 4, pp. 635-646, Aug. 2010.
[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] V Vonasek, J Faigl, T Krajnik and L Preucil. RRT-Path: a guided Rapidly exploring Random Tree. In Robot Motion and Control, 2009.

Diploma Thesis Supervisor: Ing. Martin Saska, Dr. rer. nat.

Valid until the summer semester 2017/2018

L.S.

prof. Ing. Michael Šebek, DrSc.
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, February 21, 2017

Acknowledgements

I would like to thank to Ing. Martin Saska, Dr. rer. nat. for his supervision, advices, and for his patience. I would also like to thank to Ing. Tomáš Báča and Ing Vojtěch Spurný for their advices. Finally, my thanks goes to my friends, family and my girlfriend for their support and patience during studies.

Declaration

I declare that I worked out the presented thesis independently and I quoted all used sources of information in accord with the Methodical instructions about ethical principles for writing academic thesis.

Prague, May 26, 2017

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

V Praze, 26. května 2017

Abstract

This thesis deals with developing a system for autonomous pick-up and carry of the object by a pair of robotic helicopters. Part of the work is a gripper design, synchronization of predictive regulators (MPC) and developing of trajectory planning method based on rapidly-exploring random trees. The proposed planning method ensures feasible solution in environments with narrow passages and minimizing some cost function. The algorithm is compared with few other algorithms based on RRT. The developed system is tested in Gazebo simulator and also on real UAVs.

Keywords: master thesis, autonomous object carrying, formations, motion planning, rapidly-exploring random trees, unmanned aerial vehicles

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

Abstrakt

Tato práce se zabývá vývojem systému pro autonomní uchopení a nesení předmětu párem robotických helikoptér. Součástí navrhovaného systému je návrh gripperu, synchronizace prediktivních regulátorů během letu a vývoj plánovacího algoritmu založeného na rychle rostoucích náhodných stromech (RRT). Navržený plánovací algoritmus zajišťuje nalezení dosažitelné trajektorie v prostředí s úzkými průchody a minimalizuje zadanou funkci. Algoritmus je porovnán s několika modifikacemi původního RRT. Navržený systém je testován v simulátoru Gazebo i na reálných dronech.

Klíčová slova: diplomová práce, autonomní nesení předmětu, plánování pohybu, formace, rychle rostoucí náhodný strom, bezpilotní helikoptéry

Překlad názvu: Systém pro autonomní sběr a přesun objektů párem helikoptér

Contents

Figures

Tables

Chapter 1

Introduction

Mobile robots are becoming widely used technology in various applications for example in dangerous or difficult accessible areas. Mobile robots which are used on the ground, in the air, and in the water. Robots used in the air are called unnamed aerial vehicles (UAVs) or drones. Nowadays are most popular UAVs are helicopters capable of vertical takeoff and landing (VTOL). This UAVs usually have two or more pairs of propellers with fixed pitch (Figure 1.1).

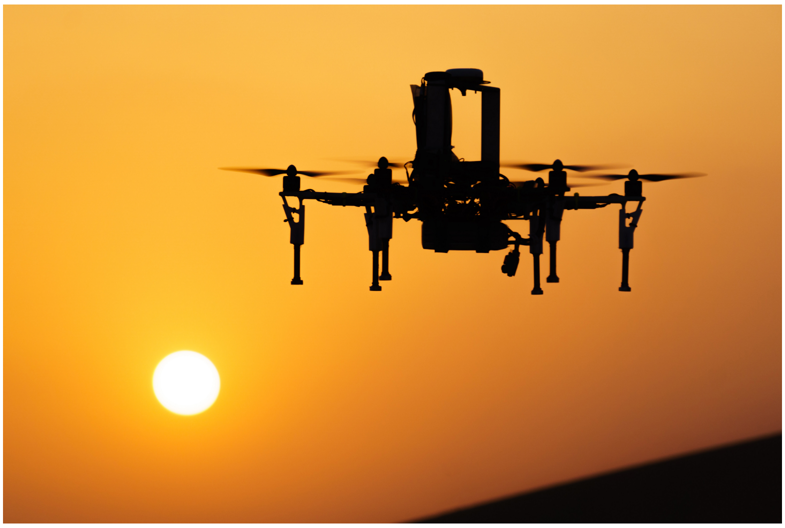


Figure 1.1: An example of six rotor UAV (hexacopter) used for experiments in this thesis

A mechanical solution of these UAVs is usually simple, just brushless motors mounted on a fixed frame. Propellers are mounted directly on motors and direction of the flight is controlled by propellers velocity change. Enough space for flight controller and other equipment is in the UAV body. Thanks to the simple design, these UAVs are more durable and cheaper than a conventional helicopter. Low cost and high durability, along with the great maneuverability, make them an ideal platform for research.

UAVs are popular for both civilian and military use. They are mainly used for photography and filming in the civilian sector. Filming by drones is especially suitable for getting aerial pictures of factories, concerts videos and nowadays also recording sports performances, providing a unique view that would not be possible without drones. Newly drones are tested for autonomous delivering objects. Autonomous delivering could be the only way to deliver medicines and vaccines into inaccessible areas. Some e-shops successfully tested autonomous orders delivered by UAVs. Autonomous delivering can reduce transport cost and speed up the delivery time. Delivering orders by UAVs is limited by size and weight of the delivered object due to the limited payload of UAVs.

This thesis contributes to the development of a system for one of challenges in the prestigious Mohamed Bin Zayed International Robotics Challenge (MBZIRC) [1]. Task solved in this challenge is search, locate, track, pick and place a set of static and moving objects by three collaboration UAVs. Some of the objects are too big and heavy for carrying by just single drone, therefore, two coordinated UAVs are needed to move these "big" objects.

This thesis deals with autonomous carrying object by the pair of UAVs. This technique could help with heavy object manipulating and partly remove the limits of UAVs.

For solving this problem, we take advantage of a long-term research on formation flying [2, 3, 4, 5, 6, 7, 8], stabilization of compact swarms of UAV [9, 10] and motion planning [11, 12, 13] at Multi-robot systems group of CTU in Prague. We aim to apply these basic research approaches of UAVs control in this unique application and to combine the expertise in formation flying and motion planning similarly as in our work on surveillance scenarios [14, 15, 16].

The first task solved in this thesis is to design and build a grasping mechanism capable of holding an object from a ferromagnetic material. The designed gripper has to be able to detect whether the object is attached. The gripper should be implemented into the UAV system. Design of the gripper is described in chapter 4.

Another task is to develop and verify a mechanism for synchronization of two model predictive controllers running parallel on two UAVs. The reason why synchronization is necessary and the solution of this problem are described in chapter 5.

Finally, motion planning methods are introduced in chapter 6. A unique motion planning technique which ensures a feasible solution in environments with narrow passages is designed. The performance of designed planning method is compared with known methods at chapter 6.6.

The experimental verification results are shown at chapter 7, where the system is verified in Gazebo simulator and results from simulator are compared with results from the real experiment.

1.1 Problem statement

The task is trajectory planning and carrying object by two UAVs in an environment with obstacles like trees, walls, etc. The obstacles may have the shape of a general polygon. In planning part of this thesis, the map of the environment is required. The map has to contain all obstacles, starting position and orientation of the object, the goal position, and orientation of the object and map borders. The output of planning is two trajectories, starting from object starting position and ending at object goal position. The generated trajectories have to be tracked simultaneously and precisely.

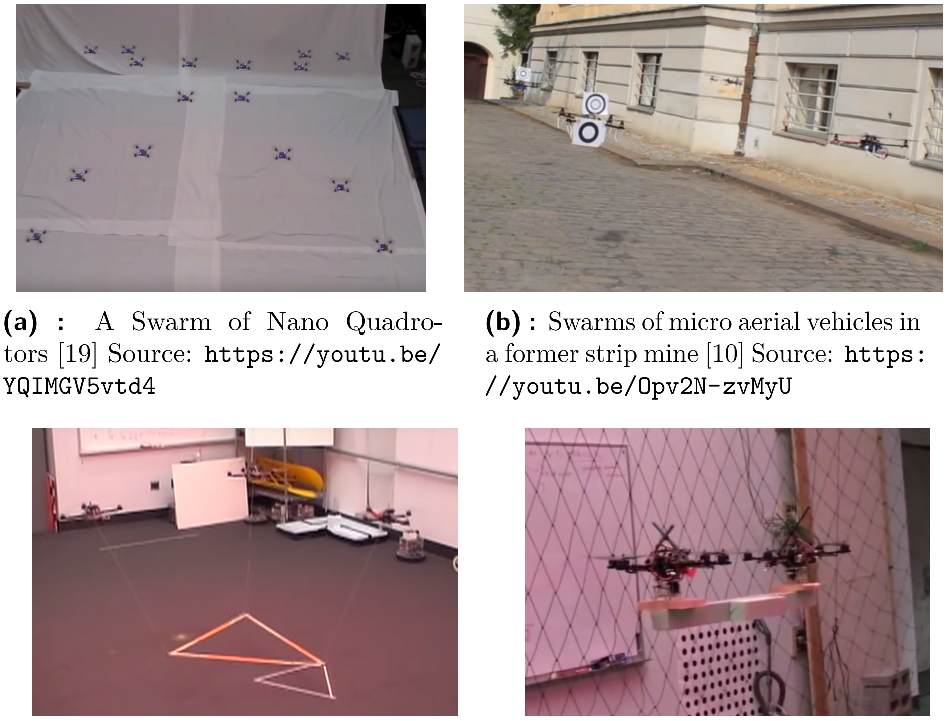
The basic requirement for UAV is sufficient payload at least as three-quarters of the object's weight and low-level stabilization. The object has to be reliably attached to UAVs, and it means some gripping device is required. Designing, production, and testing of the gripping device with feedback is part of this thesis. The UAVs used for carrying object must be able to localize itself with an error no more than 15 cm and determine the actual height above ground with an accuracy of 5 cm.

Synchronized trajectory tracking requires communication between the UAVs. Communication must be able to exchange actual position and setpoint between UAVs every 0,2 s.

Chapter 2

State of the art

There are many techniques and methodologies to control a swarm of UAVs, but only several attempts of some object being carried by two or more helicopters can be found in history. In 1972, the US Army commissioned a study to determine the feasibility of joining two helicopters to increase the payload. The study [17] found that it is possible but dangerous. Many other studies that combined the combinations of helicopters and airships were processed before the test flight. Many of these designs are summarized in [18]. The test flight of prototype *PA-97: Multiple Helicopter Heavy Lift System*, which was a combination of four helicopters and airship, ended in disaster.



(a) : A Swarm of Nano Quadrotors [19] Source: `https://youtu.be/YQIMGV5vtd4`

(b) : Swarms of micro aerial vehicles in a former strip mine [10] Source: `https://youtu.be/Opv2N-zvMyU`

(c) : Cooperative Manipulation and Transportation with Aerial Robots [20]

(d) : Cooperative Grasping and Transport using Quadrotors [21]

Figure 2.1: Examples of related works

Meanwhile, researchers in the Soviet Union studied using two helicopters to transport spaceship Buran, tank and other heavy components tied on a long cable to Baikonur cosmodrome [22]. During one of test flights, weak turbulence appeared, and helicopters began to be unstable, and the object was subsequently dropped. After this incident, the research was stopped, and aircraft AN-225 Mrija with sufficient payload was developed.

In the literature, there is a basis of many written documents preoccupied with formations of flying and trajectory planning. There are several approaches to formation behavior [23, 24], usually based on a social behavior of animals, leader, and follower approach or it relies on virtual structure following [25]. The virtual structure tracking is very similar to cooperative object carrying. Example of virtual structure following can be found in [19], where 20 MAVs localized by VICON[1] flying in various formations. Another example of formation can be found in [10], where formation with relatively stabilized UAVs is showed. The formation flying is a complex discipline, but because there is no solid connection between the members of the formation, there is not necessarily such high precision positioning like cooperative carrying.

Path planning of formation in the virtual structure is usually done by creating virtual rigid body object and path planning for this object. The individual path for each UAV is generated from the position of UAV in virtual object. Path planning for UAVs manipulating the object in the complex environment can be defined as piano movers problem. Many approaches can be found in the literature. In [26] the grid-based algorithm A* is used for planning trajectory of the mobile robot. The potential field is used for path planning in [27] and can be combined with other approaches. A geometric algorithm like visibility graph can also be used for path planning, in [28] is combined visibility graph algorithm with A* algorithm. Very popular is sampling-based algorithms like Rapidly-exploring random trees (RRT) [29]. An example of formation path planning by RRT is in [30]. The RRT algorithm has many enhancements like RRT* introduced in [31], RRT-Path [11] or Transition based RRT [32]. In [33] is introduced path planning method for manipulating with cable crane system by three UAVs. A transition based RRT algorithm is used for motion planning of three UAVs connected to crane platform.

The autonomous carrying by a couple of robotic helicopters is not new. In [34] methods of stabilizing and controlling system of two helicopters carrying the load, are examined and simulated. The stability of the multi-helicopter system is also investigated in [35]. The another approach is shown in [21, 20], where complete control and trajectory flying with UAVs carrying object is shown. The position of UAVs is observed by VICON and control loop is not computed off-board and sent to UAVs by a wireless network. In contrast to these approaches, where the locations of UAVs are observed by external

[1] Vicon is multiple cameras based capture system, capable of providing position with millimeter accuracy. `www.vicon.com`

devices this thesis deal with the object carrying using only on-board sensors for localization.

Chapter 3

Preliminaries

In this chapter, standard techniques and preliminaries on which we build in this thesis are introduced.

3.1 Model predictive control

Model predictive control (MPC) is an advanced technique of process control, widely used in process industries where time constants are relatively high, like chemical plants, and where computing power of available computers has been sufficient for decades already. With more powerful hardware, its popularity is growing even in faster systems control. Using of estimation of the system model, MPC can predict the future behavior of the system and can choose the right control actions to achieve desired behavior. The MPC solves a finite horizon optimization control problem starting from the current state with constant sampling time.

This thesis builds on the existing MPC controller [36], which was designed at MRS group of CTU, for a micro aerial vehicle (MAV). The MPC controller is capable trajectory tracking and disturbance rejection by solving a finite horizon optimization control problem starting from the current state with a constant sampling time. A sequence of control steps that minimize the specified cost function is a result of the optimization process, with respecting system constraints. Only the first step of control steps is implemented, and the optimization problem is solved again from the new state.

The MPC controller for trajectory tracking [36], with constant sampling time, between trajectory points, is not the sufficient solution for carrying some object by the pair of helicopters. Due to the disturbance, synchronization between helicopters may be lost, and the distance between them may change beyond an acceptable limit. At higher speeds, points in the trajectory are too far apart, and optimization may cause poor tracking accuracy. Some improvements are proposed in Chapter 5 where the solution with variable sampling time is proposed. The variable sampling time depends on the state of the second MPC, and it is used for restore synchronization between UAVs.

3.2 Kinematic model

For some path planning methods, some kinematic model is necessary. The kinematic model used for path planning can be less accurate than the model used in MPC [36]. This model describes the behavior of a system state depending on the control parameters but, ignores accelerations and energies associated with this motion. The UAV is considered as a dimensionless particle in this model (Figure 3.1).

The kinematic model of a UAV in the 3-dimensional space is described by three state variables which represent its position in space and position behavior on three input parameters as

$$\begin{aligned} \dot{x}(t) &= v \cdot \cos\phi \cos\alpha \\ \dot{y}(t) &= v \cdot \sin\phi \cos\alpha \\ \dot{z}(t) &= v \cdot \sin\alpha \\ \dot{\varphi}(t) &= \gamma, \end{aligned} \tag{3.1}$$

where v is a velocity of the UAV, ϕ is an angle in the plane that determines the direction of movement. Angle α determines to ascend or descending of the point, and γ is an angular speed of yaw.

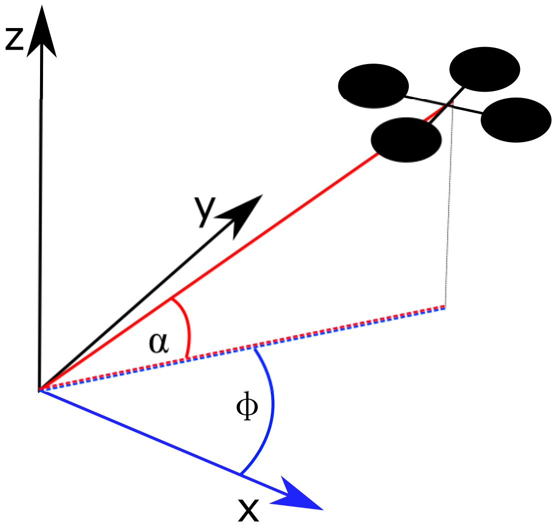


Figure 3.1: Model of UAV

A two-dimensional model is sufficient for the purpose of this thesis. The model from equation 3.1 can be simplified to two-dimensional case by an assumption constant α with value 0 as

$$
\begin{aligned}
\dot{x}(t) &= v \cdot \cos\phi \\
\dot{y}(t) &= v \cdot \sin\phi \\
\dot{\varphi}(t) &= \gamma.
\end{aligned} \tag{3.2}
$$

When the object is attached to a flexible mount (sufficient mount is designed in chapter 4), changing of yaw is not necessary. Angular speed γ is 0, and it means yaw stay constant.

3.3 Feasibility conditions

The configurations of the system are reachable in case of accomplished conditions given by degrees of freedom of grasping system between drones and object. The relative position of the UAV is limited from one side so that the position of the second UAV is not in collision. The opposite direction is constrained by the thrust of drone motors. The carried object is more stable when it is in tension.

Chapter 4
Magnetic gripper

Design some device to hold the object under the UAV is necessary for successful compliance of this task. Requirements for this device is:

- Lightweight
- Low energy consumption
- High hold-force
- Switchable between 0 and 2 DOF mount for ensure stable attachment when carrying object by single UAV and free attachment when cooperative carry.
- Reliable dropping
- Grip status feedback for recognize proper attachment of the object to the UAV

The object is made from the ferromagnetic material so that some magnetic gripper can be used. The easiest solution could be an electromagnet, but there is a problem with energy consumption. The force of electromagnet (Equation 4.1) is directly proportional to the current in the winding of electromagnet (I), a number of turns in the winding (N), cross-sectional area of core (A) and inversely proportional to the length of flux path (L). This means that we need a relatively high current or a large number of turns to achieve sufficient strength. For our use, mechanical solution is designed because tested electromagnet has not suitable payload for an object according to the MBZIRC specifications.

$$F = \frac{\mu^2 \cdot N^2 \cdot I^2 \cdot A}{2 \cdot \mu_0 \cdot L^2} \tag{4.1}$$

The strongest permanent magnet available to buy is neodymium magnet (NdFeB). It is a type of rare-earth magnet made from an alloy of neodymium, iron, and boron. This alloy is vulnerable to corrosion hence the surface is covered by nickel, zinc or epoxy.

4.1 Basic idea

Magnetic field surrounds space around permanent magnet (Figure 4.1). Magnetic field lines, starting from one magnetic pole and ending in the other, describe a magnetic field

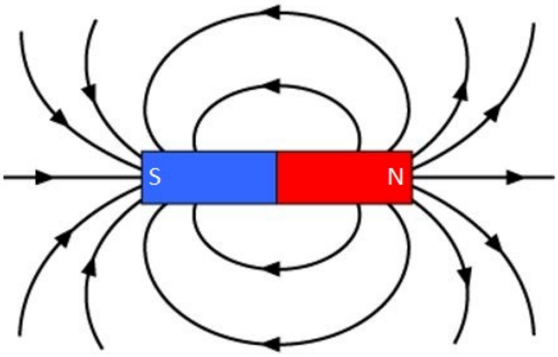


Figure 4.1: Magnetic field of cylindrical magnet [37]

Ferromagnetic materials very well lead the magnetic field. A pole extenders from the ferromagnetic material can be used to lead magnetic field directly to carried object (gray on figure 4.2). Object dropping is realized by a magnet rotation of 90 degrees. The magnet rotation creates an air gap between magnet and pole extenders and the magnetic circuit is interrupted. The magnet has to be greater than its diameter to avoid contact with pole extenders when the gripper is turned off. Some remaining magnetic fields still affect the object despite the fact that the magnet is not in touch with the pole extenders. This remaining magnetic field can be strong enough to hold lightweight object attached. Shorting magnetic field can be used for shielding the remaining magnetic field (Figure 4.3), magnetic short is illustrated by yellow color. There is still a weak remaining magnetic field, but too weak for hold the object attached.

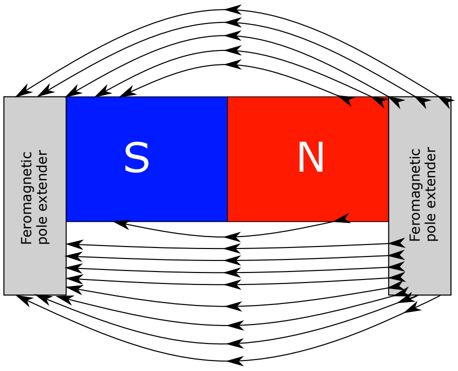


Figure 4.2: Magnetic field with pole extenders

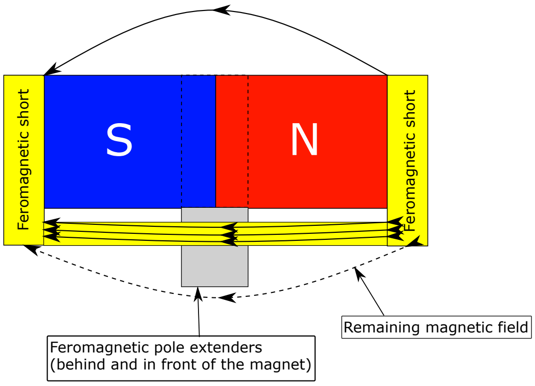

Feromagnetic short
S
N
Feromagnetic short
Remaining magnetic field
Feromagnetic pole extenders
(behind and in front of the magnet)

Figure 4.3: Shorted magnetic field

The behavior of a magnetic field can be used to detect the state of the gripper. Magnetic field cover the whole magnet uniformly when an object is not attached (Figure 4.2). If the object is attached, the magnetic field is conducted by the pole extenders to the object and the magnetic field, on the opposite side of the magnet is reduced (Figure 4.4). Hall effect sensor can be used to detect the change of magnetic field, so the UAV can recognize if the object is attached or not.

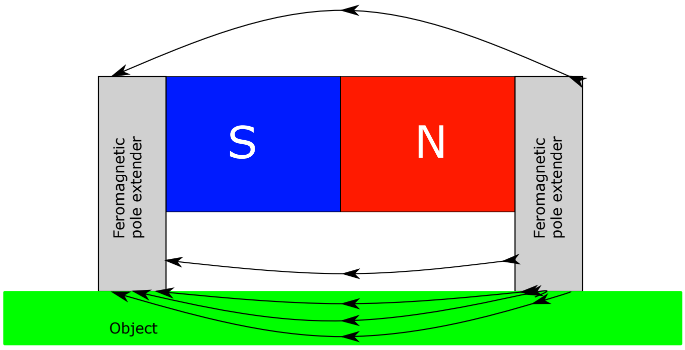


Figure 4.4: Magnetic field with object attached

4.2 Solution

The magnetic gripper is designed using the basic idea. The design is created suitable for 3D printing. Almost all mechanical components are 3D printed. The shape of the gripper is square with side length 76.5 mm. Two, rod type neodymium magnets are used. Both magnets have the same size. The datasheet holding force is 71 N with 12 mm diameter and 20 mm in height. The magnetic flux conducting is done by pole extenders made of iron. The iron is used because its permeability μ has very good value, that means it has excellent magnetic conductivity. Extenders are mounted on springs to ensure the best possible contact with the magnet and smooth magnet rotation. The standard servo is used for magnet rotation. The servo torque is transmitted by gear with ratio 1:2 to increase the force.

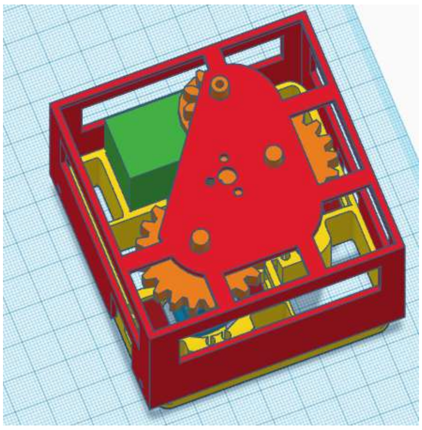


Figure 4.5: Model of the magnetic gripper with the cover

Each magnet is built in its holder (blue in the figure 4.6) with Hall effect sensor. This means that the feedback is separate for each magnet, so it is possible to recognize which magnet holds the object. The magnet holders are directly connected with gears. The whole device is covered (red on the figure 4.5). The cover has holes for hold the gears in their desired positions.

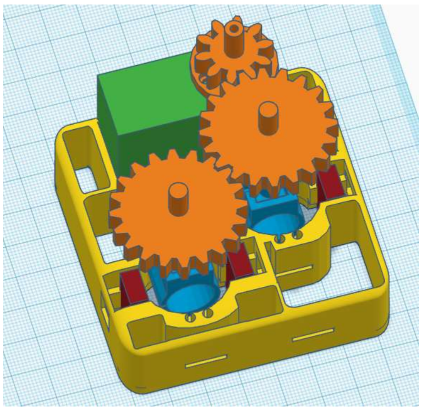


Figure 4.6: Model of the magnetic gripper without the cover

The grippers weight is 200 g. More than half of the total weight consists of servo and magnets. The holding force is 50 N on 1mm iron sheet, which is sufficient for this thesis. Remaining holding force produced by the turned off gripper is less then 1 N.

4.3 UAV mount

For the object carrying by two UAVs, the free joint between the Uav and the object is better. Something like a ball joint or rope hanging can be used. For transferring an object using only one UAV may be useful to lock the joint in fixed position to prevent oscillation. The mount with a servo is designed. The servo has a winch with string. On the top of the gripper cover cone with the end of the string is mounted. The string goes through a tube with the same inner diameter as the base of the cone. The joint is locked when the string is completely wrapped around the winch, and the cone is in the tube. When the string is not wrapped, the cone is under the tube, and the gripper is hanging on the string. It is shown in the figure 4.7.

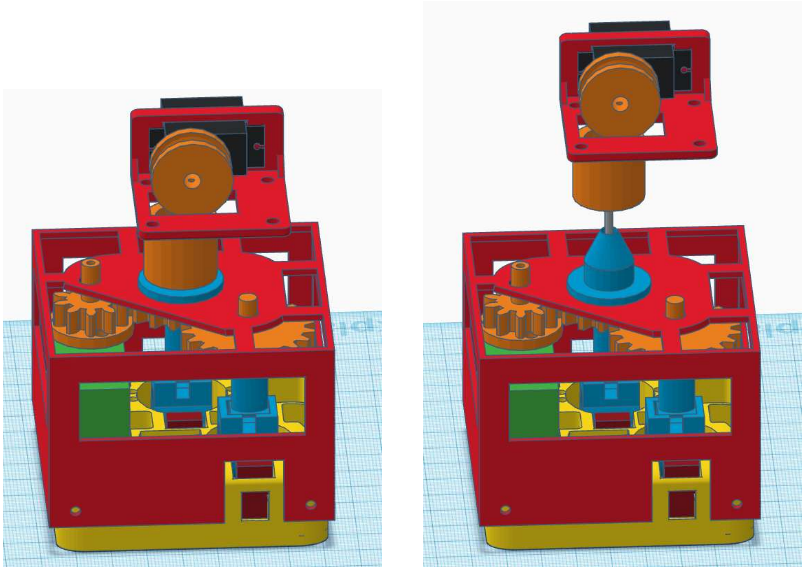


Figure 4.7: Locked and unlocked joint

4.4 Gripper control

The magnetic gripper is controlled from the main computer by the electronic board developed by MRS group of CTU. The communication between the main computer and gripper electronic is ensured by the UART bus with UART to USB converter. The communication protocol is simple and contains checksum to detect communication errors.

```
A/B length code data CS
```

Where **A/B** determines direction of communication, **length** is length of message including checksum. **Code** identify type of message (Table 4.1) and **data** is data associated with this type of message. The checksum **CS** is just one byte number an it is sum of all bytes in the message.

To gripper		From gripper	
Code	Description	Code	Description
1	Set gripper on/off	1	Attached state
2	Eeprom write to address	2	Eeprom read result
3	Eeprom read from address	3	Hall sensors values
4	Set joint on/off	4	Threshold values
5	Set thresholds		

Table 4.1: Table of communication codes

4.5 Feedback

The feedback is critical for this task. Thanks to feedback the UAV can recognize when the object is attached and can fly to the dropping zone. Without the feedback, the UAV can stop landing too high while the object is not attached yet, or can continue landing despite the fact that the object is attached. This behavior can cause crash of the helicopter.

Two Hall effect sensors are used for recognition of attached state. Three thresholds are used for the decision. Two thresholds are absolute values of the sum of values from both sensors for on and off state with hysteresis. The third threshold is a relative value, when the value of one of the sensors changes more than the threshold, the state changes.

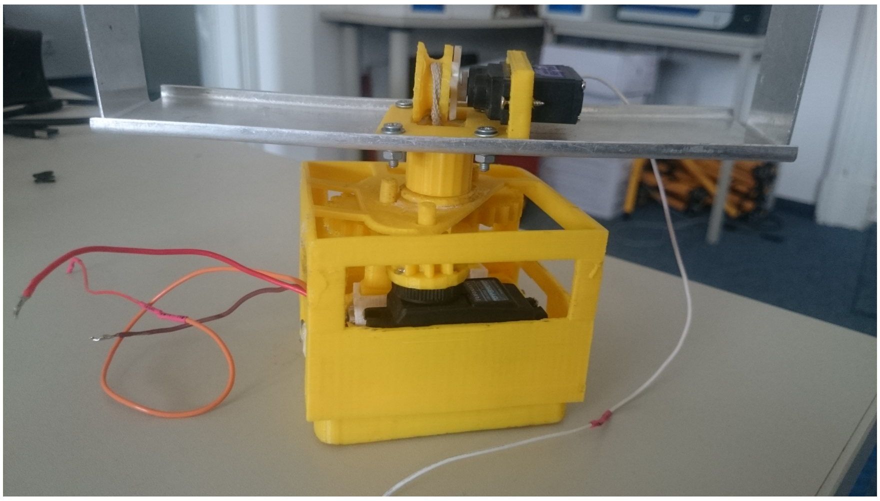


Figure 4.8: Photo of 1 of the 6 grippers made for the MBZIRC competition [38]

Chapter 5

Object carrying

5.1 Attaching and detaching object

The object attaching and detaching is a tough part, and it requires very precise positioning and synchronization during takeoff. Synchronization of trajectory tracking is described below in chapter 5.2.2. The controller is designed like a state machine (Figure 5.1). When the object is detected, the endpoints are determined and assigned to a specific UAV. The endpoints are aligned with the UAV. Landing on the object is done by tracking the landing trajectory. When both UAVs are aligned with the object, the magnetic gripper is turned on, and landing is started on both UAVs simultaneously. During the landing, the gripper feedback is checked, and when the gripper feedback shows attached state on both UAVs the landing is stopped, the trajectories for take-off are generated, loaded and started again simultaneously. After the take-off, the system is ready for the flight to the dropping zone.

Detaching is quite similar like attaching. Again, landing and takeoff trajectories are generated, and landing is started simultaneously. The landing trajectory ends at the height where the object is few centimeters over the ground. The both grippers are turned off after landing is done. The object should fall to the floor, which should confirm the gripper feedback. The object transportation is done after take-off.

5.2 Flying with object

There are two ways how to solve the cooperative carrying problem.

1. We can model both UAVs and attached object as one object and control it centralized from one of the UAVs.
2. Implement a formation control, where both UAVs have an independent controller which respects possible configurations of formation, while the object is carried.

In this thesis, the second option is used. Two MPC controllers being run in parallel are used for tracking trajectories.

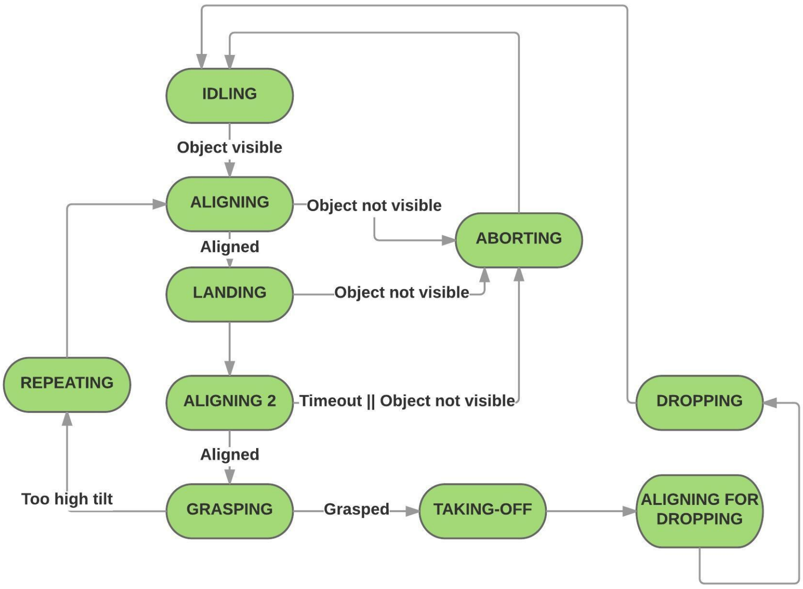


Figure 5.1: Illustration of state machine used for the object attaching and detaching

5.2.1 Without orientation change

For the purpose of MBZIRC competition, no synchronization between MPC trackers is necessary. In competition, the assignment is getting the object to the drop zone. Drop zone is big enough to detach the object in any pose. In this case, we can simply let the orientation of object constant and just flight shifted trajectories directly to the drop area, without any synchronization of MPCs. If some inaccuracy appears, it should be same on both MPCs.

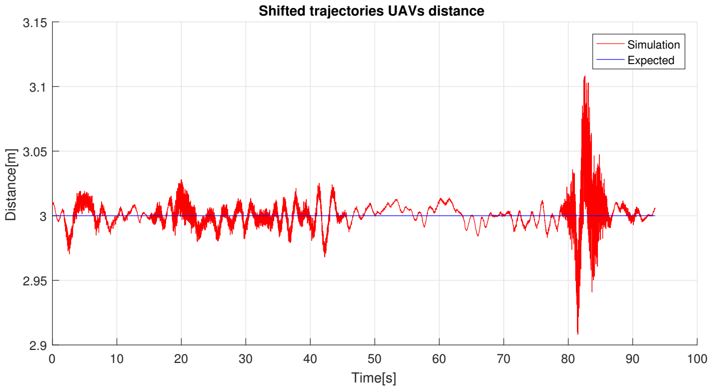


Figure 5.2: Graph of the distance between UAVs during flying shifted trajectories

As we can see in the figure 5.2, when the trajectories are started in exactly same time, the distance of UAVs is almost constant, and synchronization is not necessary.

5.2.2 With orientation change

The more approach case is needed when we add an orientation change. Orientation change is needed when avoiding collision with obstacles. We can also try to take two trajectories with the same duration, but the length of the trajectories will be different. Even though the paths tracking are started simultaneously, synchronization can be lost during orientation change move as shown in figure 5.3. Some communication between MPC trackers should be added.

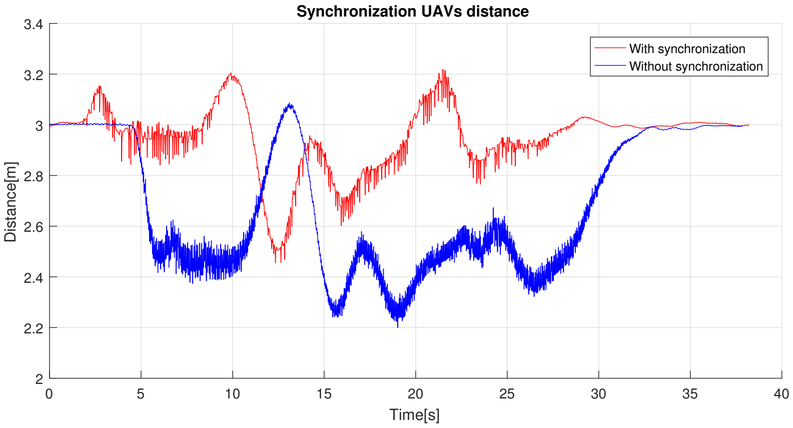


Figure 5.3: Comparison of Distance between UAVs with and without synchronization

Setpoint synchronization is done by slowing down the sampling time of "faster" MPC which is waiting for delayed second MPC. Thanks to this technique, it is not necessary to start trajectory tracking exactly at the same time. When MPC put setpoint to prediction horizon, a message to another MPC is sent. Next setpoint is added to prediction horizon only when another MPC reached the set point with the same index. This synchronization is not only required when starting the trajectory following. During the trajectory, tracking can also cause loss of synchronization, for example, because of an overloaded control computer, or process scheduling on not real-time operating systems.

Chapter 6

Trajectory planning

Path planning is the essential part of this thesis. The resulting path has to respect all constraints of both UAVs and respective position between them. Many methods of trajectory planning can be found in the literature. There are grid-based algorithms, geometric algorithms, sampling based and others. Each category has advantages and disadvantages. Sampling based algorithm is used in this thesis.

Sampling based path planning is suitable for complex problems in high-dimensional spaces. These techniques aim to find a collision-free path rather than the quality path.

6.1 Rapidly exploring random tree (RRT)

The rapidly-exploring random tree introduced in [29] is the random based algorithm. It starts from initial configuration and uses random points to determine the direction of tree growing. The random point is generated, and the nearest node is found. The tree is expanded from the nearest node in random node guidance and checking if the new configuration is feasible. The algorithm ends when the final configuration is reached, or a maximum number of nodes is reached. The number of nodes has to be limited, because sampling based algorithms are not able to recognize, whether target configuration is reachable.

The expansion is done by the commanding model by generated inputs. The generated inputs have to satisfy constraints (maximal speed, acceleration, etc.) of the real system. When these constraints are met, resulting trajectory is feasible.

In Algorithm 1 is visualized structure of RRT algorithm.

- **distance** - Returns distance between two points.

- **randomPoint** - Select random point.

Algorithm 1: RRT

```
input  : $n_{Start}$ - Starting node
         $n_{Goal}$ - Goal node
output : Trajectory from $n_{Start}$ to $n_{Goal}$
initTree;
$n_{New} = n_{Start}$;
while distance($n_{New}$,$n_{Goal}$)<Tolerance and numOfIterations<MaxIterations do
    $n_{Random}$=randomPoint();
    $n_{Nearest}$=findNearestNeighbor(Tree,$n_{Random}$);
    $n_{New}$=expand($n_{Near}$, $n_{Random}$);
    if isFeasible($n_{New}$) then
        Tree.add($n_{New}$);
    end
end
if numOfIterations<MaxIterations then
    return reconstructTrajectory(Tree,$n_{New}$,$n_{Start}$);
else
    return null;
end
```

- **findNearestNeighbor** - Returns node with minimal distance from n_{Random}.

- **expand** - Expand tree from node n_{Near} in direction n_{Random}. Expanding is done by setting input to kinematic model.

- **isFeasible** - Returns true if node is feasible configuration in free configuration space C_{free}.

- **reconstructTrajectory** - Returns trajectory from n_{Start} to n_{New}. Trajectory is obtained by recursively picking parent node, until parent node is not null.

In this case, the simple model is used. Model is described in chapter 3.2. The formation is composed of two UAVs, described by model, and object, represented as rectangle between UAVs. Both UAV models are commanded simultaneously, and every tree node contains coordinates of both UAVs.

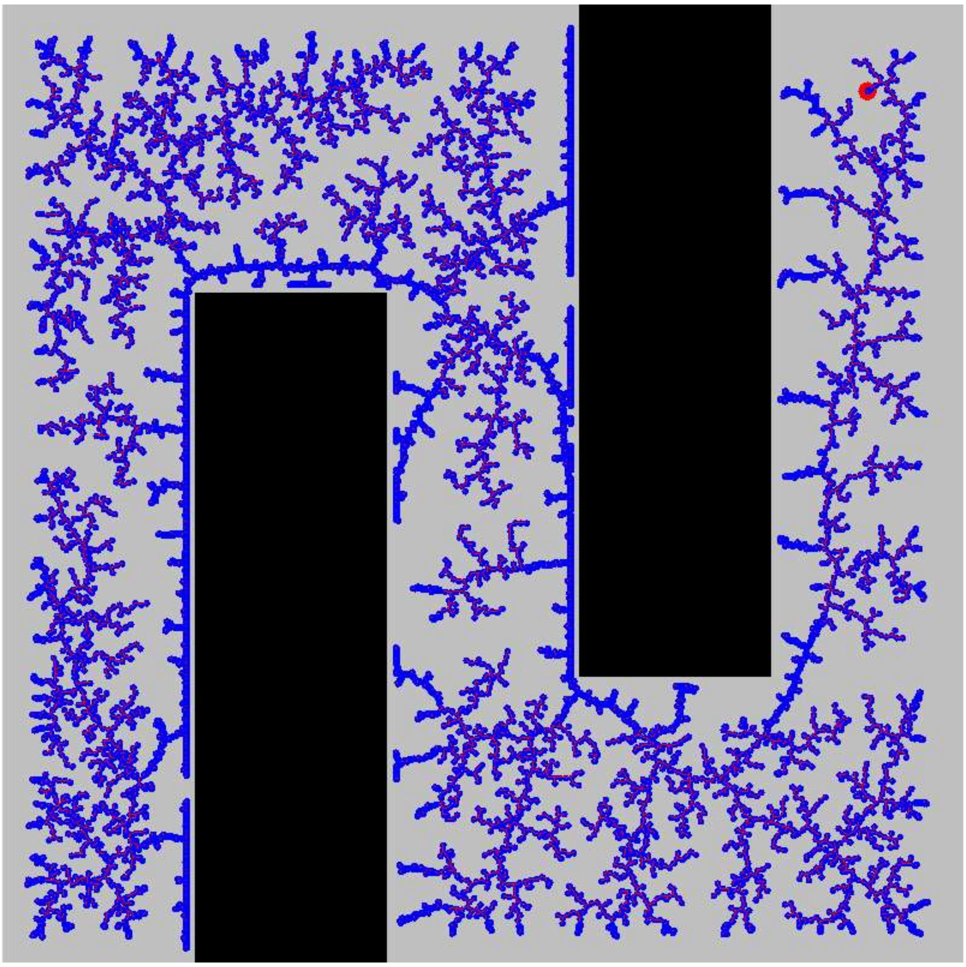


Figure 6.1: RRT growing example

6.2 RRT-Path

The significant disadvantage of RRT is poor efficiency in the environment with narrow passages. The tree grows randomly, and there is a low probability of growing directly to narrow passage. The RRT-Path [11] is a modification of RRT, which deals with growing tree in the environment with narrow passages. A guiding path is found by some finite algorithm(A*, Dijkstra, etc.). Point for tree expansion is not arbitrary, but it is a point on the guiding path. This ensures tree growing only in direction which leads to goal.

The guiding path could be found by many algorithms, which is finite and complete. The complete algorithm is an algorithm which found the path if exists and it should end in finite time when the path does not exists. In our case, two algorithms are used and compared. The first algorithm is A*, which find the shortest way from the start point to the goal point. The second algorithm is Dijkstra, which finds the optimal path regarding distance from

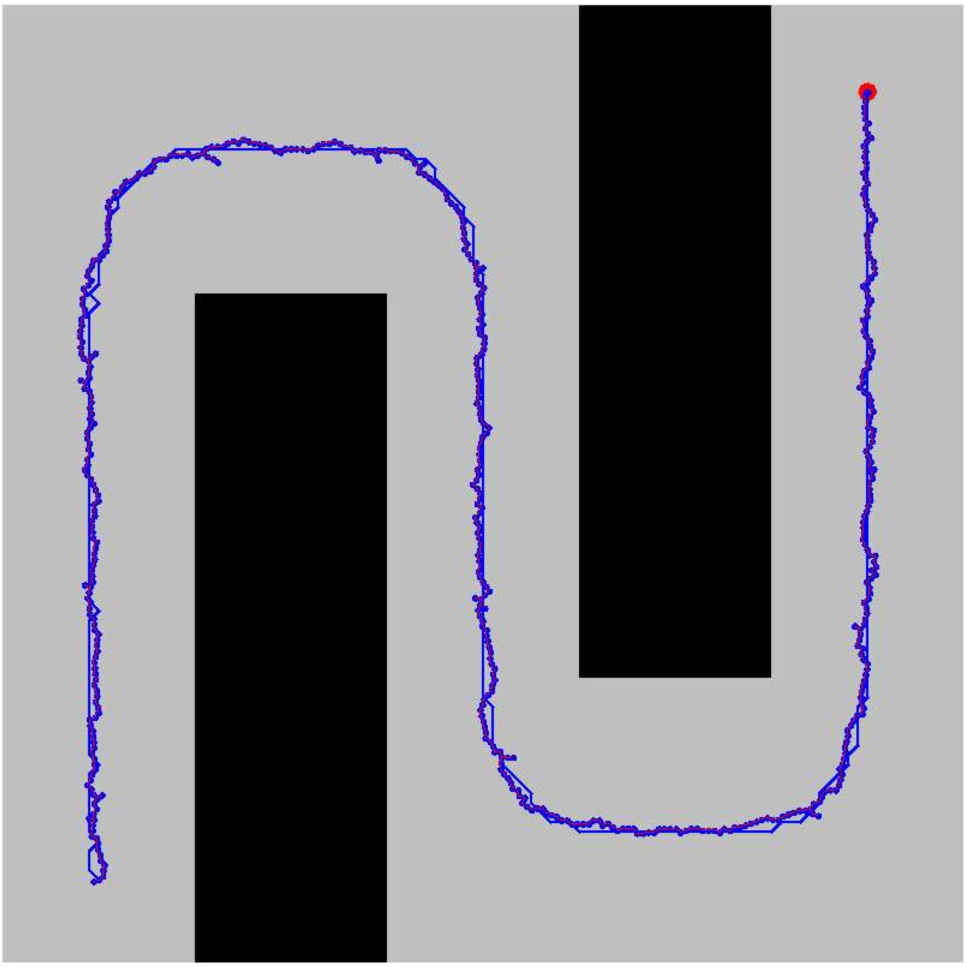


Figure 6.2: RRT-Path growing example

the nearest obstacle. The found path in not shortest, but it is relatively safe.

- **findGuidingPath** - Find path from node n_{Start} to n_{Goal} by some algorithm regardless constrains of real robot.

- **randomPointGuided** - Select random point in specified radius around point on guiding path

6.3 Cost function

For the following two algorithms a cost function must to defined. The cost function measures a quality of configuration. The conditions of feasible configuration are defined in chapter 3.3. For a given configuration q, we

```
Algorithm 2: RRT-Path
input  : $n_{Start}$ - Starting node
         $n_{Goal}$ - Goal node
output : Trajectory from $n_{Start}$ to $n_{Goal}$
initTree;
$n_{New} = n_{Start}$;
guidingIndex=0;
guidingPath=findGuidingPath($n_{New}$,$n_{Goal}$);
while distance($n_{New}$,$n_{Goal}$)<Tolerance and
  numOfIterations<MaxIterations do
    $n_{Random}$=randomPointGuided(guidingIndex, radius);
    $n_{Nearest}$=findNearestNeighbor(Tree,$n_{Random}$);
    $n_{New}$=expand($n_{Near}$, $n_{Random}$);
    if isFeasible($n_{New}$) then
        Tree.add($n_{New}$);
    end
    if distance($n_{New}$,guidingPath(guidingIndex)<Tolerance and
      guidingIndex<guidingPath.length then
        guidingIndex++;
    end
end
if numOfIterations<MaxIterations then
    return reconstructTrajectory(Tree,$n_{New}$,$n_{Start}$);
else
    return null;
end
```

define the cost $c(q)$. Many different cost functions can be defined based on different optimization problem. Our optimization problem is relative position between UAVs.

6.3.1 Optimal distance

An optimal distance between UAVs is defined based on the length of the carried object and length of the hinge. The ideal distance is such that the curtain is stretched but not so much that the drones have a sufficient thrust. The tension between UAVs prevents oscillation of the object while carrying. The cost function has to penalize deviation from the ideal position. The deviation in the direction of another UAV should be penalized less than in the opposite direction.

We define optimal distance cost function as

$$c(q) = ((d - l) \cdot 10)^4 \tag{6.1}$$

where d is distance between UAVs and l is an ideal distance between UAVs.

It is a simple power function of deviation from the ideal distance multiplied by 10. The power function ensures sharp rise in prices at a higher deviation.

6.4 Transition based RRT (T-RRT)

Transition-based RRT [32] is another modification of RRT algorithm. The algorithm was developed for improving the quality of the solution, generated by sampling based algorithm. Transition test is integrated, enabling to bias the tree growing toward the low-cost region. This technique can be used for many improvements of result path like smoothing or maximizing distance from obstacles. The structure of T-RRT is shown at algorithm 3. There is one change in condition for node adding. Function **transitionTest** checking if cost of new node is suitable for add node.

Algorithm 3: T-RRT

```
input  : $n_{Start}$ - Starting node
         $n_{Goal}$ - Goal node
output : Trajectory from $n_{Start}$ to $n_{Goal}$
initTree;
$n_{New} = n_{Start}$;
while $distance(n_{New},n_{Goal})<Tolerance$ and
 $numOfIterations<MaxIterations$ do
    $n_{Random}$=randomPoint();
    $n_{Nearest}$=findNearestNeighbor(Tree,$n_{Random}$);
    $n_{New}$=expand($n_{Near}$, $n_{Random}$);
    if $isFeasible(n_{New})$ and $transitionTest(T,C_{Near},C_{New})$ then
        Tree.add($n_{New}$);
    end
end
if $numOfIterations<MaxIterations$ then
    return reconstructTrajectory(Tree,$n_{New}$,$n_{Start}$);
else
    return null;
end
```

A cost function, used for transition test, can be defined in many ways. In motion planning is useful define the cost function based on terrain, to avoid high-slope regions. For UAV swarm is important relative location of each UAV. The distance between UAVs can be used in the cost function.

In Algorithm 4 is shown structure of function **transitionTest**. There are two special cases. The first special case is when the new cost C_{New} is higher than C_{Max}, the result is automatically false. The second case is when C_{New} is smaller than C_{Near}. In this case, is result true. The Monte Carlo method

Simulated annealing is used in the last case, to prevent deadlock in local minimal. This technique does not guarantee to find global minimal.

Algorithm 4: transitionTest(T,C_{Near},C_{New})

```
input  : T - Temperature
         $C_{Near}$ - Cost of nearest neighbor
         $C_{New}$ - Cost of new node
if $C_{New} > C_{Max}$ then
    return False;
end
if $C_{New} < C_{Near}$ then
    return True;
end
if $C_{New} < T$ then
    $T = C_{New}$;
    return True;
else
    $T = T \cdot 2^{0.8}$;
    return False;
end
```

6.5 T-RRT-Path

The T-RRT can find the path with better quality then RRT, but there is the same problem with narrow passages like RRT. This is a direct challenge to combine it with RRT-Path. This method should produce a relatively high-quality path even in a complex environment with narrow passages. The point for expansion is the point on the guiding path, like at RRT-Path. The new node is checked by transition test (Algorithm 4) before it is added to the tree. When the node is accepted, the temperature T is recalculated and decreased to tighten the decision-making level. At the opposite side, the temperature is increased when the node is accepted. This approach is trying to keep ideal distance between UAVs, but if it is not possible, the distance can be changed in given limits. The example why change ideal distance is shown in figure 6.3. There is a corner with the acute angle. When the UAVs keep ideal distance with the tensioned hinge, the angle of the corner is too acute to go through, but when the UAVs get closer, the object will be less stable, but UAVs can go through the corner.

The ratio for changing temperature should be chosen so that it is not too high or low. If the ratio is too high, the temperature increases too quickly and path quality is not good. If the ratio is low, it takes a long time to raise the temperature to the suitable value, when it is deadlocked in local minimal

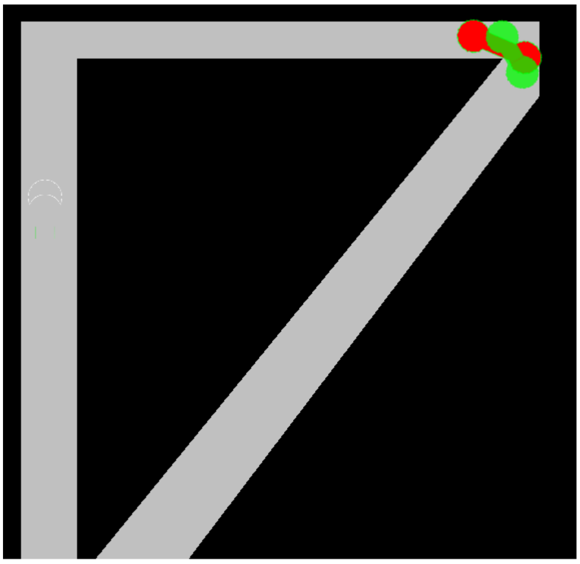


Figure 6.3: Acute angle corner with marked UAVs with ideal distance (red) and in closer distance (green)

and algorithm is slow. The increasing ratio is experimentally determined as $2^{0,8}$.

6.6 Performance comparison

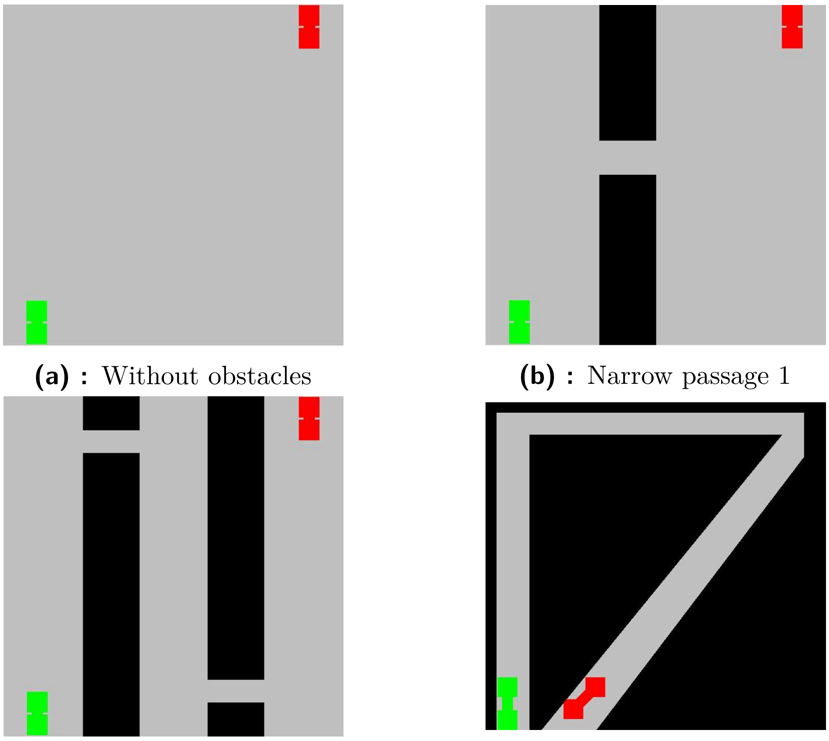
The performance of RRT-Path is compared with original RRT in [11]. There are shown a comparison of tree sizes in the environment with narrow passages. Algorithm T-RRT is compared with original RRT in [32], where the cost of resulting path generated by both algorithms is compared. In this chapter, the T-RRT-Path is compared with RRT-Path and T-RRT algorithms.

In the table 6.1 is shown time, the size of the tree, the size of resulting path and cost for all four algorithms mentioned above. The maximum node count is limited to 10 000 and mark **-** indicates that the value is not available because the algorithm has not ended. It is tested on one map without obstacles (Figure 6.5a), two maps with narrow passages (Figure 6.5b, 6.5c) and one map with passage with two corners (Figure 6.5d). The starting configuration is marked by green color and goal configuration by red color. The tree in T-RRT-Path has as few nodes as RRT-Path, but the path cost is much smaller.

In the figure 6.6 is shown a comparison of deviations from the ideal distance between UAVs computed by algorithms RRT-Path and T-RRT-Path for all maps.

Algorithm 5: T-RRT-Path

```
input  : $n_{Start}$ - Starting node
         $n_{Goal}$ - Goal node
output : Trajectory from $n_{Start}$ to $n_{Goal}$
initTree;
$n_{New} = n_{Start}$;
guidingIndex=0;
guidingPath=findGuidingPath($n_{New}$,$n_{Goal}$);
while distance($n_{New}$,$n_{Goal}$)<Tolerance and
  numOfIterations<MaxIterations do
    $n_{Random}$=randomPointGuided(guidingIndex, radius);
    $n_{Nearest}$=findNearestNeighbor(Tree,$n_{Random}$);
    $n_{New}$=expand($n_{Near}$, $n_{Random}$);
    if isFeasible($n_{New}$) and transitionTest($C_{Near}$,$C_{New}$) then
        Tree.add($n_{New}$);
    end
    if distance($n_{New}$,guidingPath(guidingIndex)<Tolerance and
      guidingIndex<guidingPath.length then
        guidingIndex++;
    end
end
if numOfIterations<MaxIterations then
    return reconstructTrajectory(Tree,$n_{New}$,$n_{Start}$);
else
    return null;
end
```



(a) : Without obstacles

(b) : Narrow passage 1

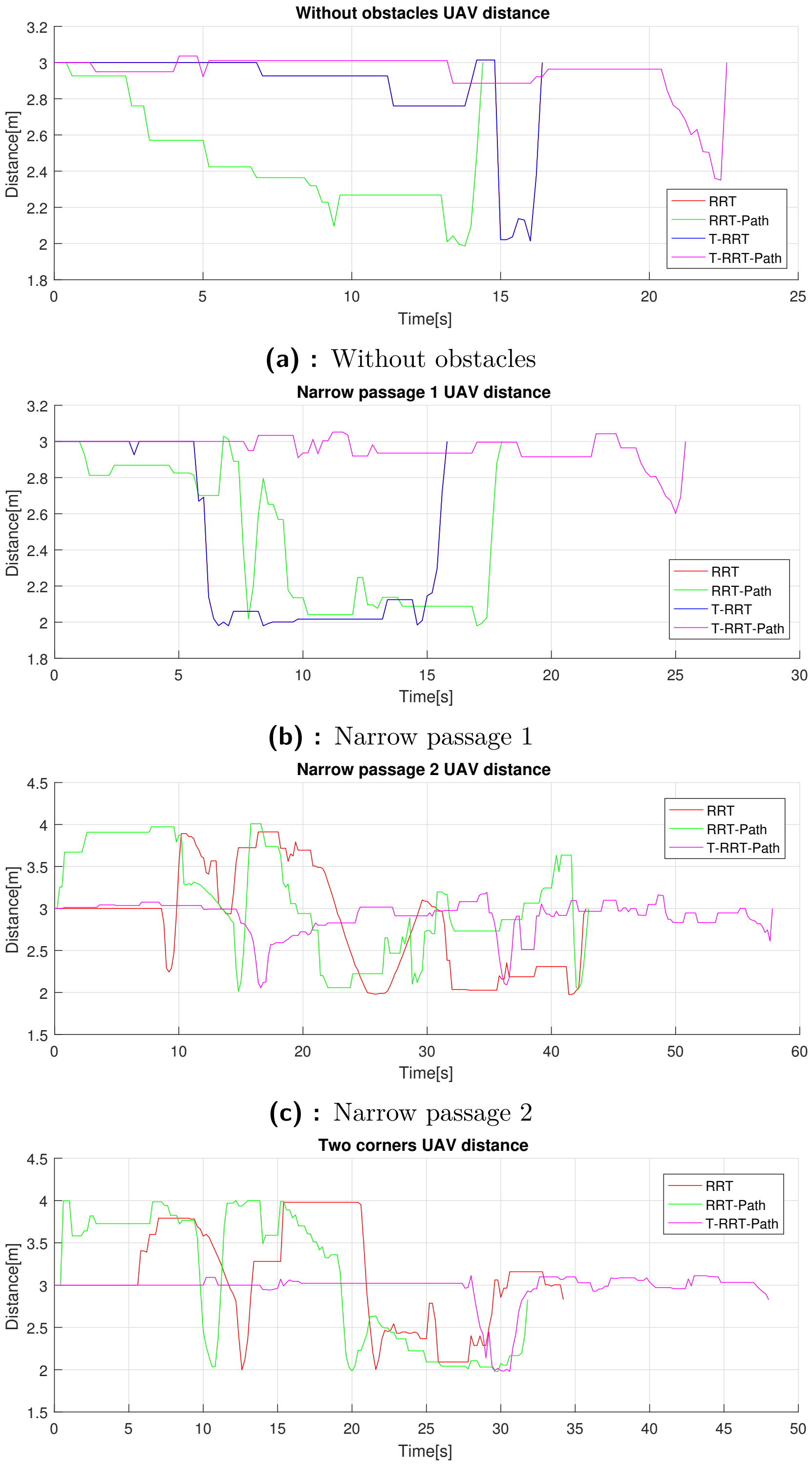
(c) : Narrow passage 2

(d) : Two corners

Figure 6.4: Testing maps

	Without obstacles	Narrow passage 1	Narrow passage 2	Two corners
RRT				
Prepare time	0,03 s	0,08	0,06 s	0,00 s
Total time	4,03 s	19,09 s	1365,28 s	961,75 s
Tree nodes	141,00	166,50	924,00	2928,50
Path nodes	114,00	129,50	283,50	241,00
Path cost	440,02	552,80	2015,89	1421,44
RRT-Path				
Prepare time	0,15 s	0,08 s	0,07 s	0,02 s
Total time	1,18 s	2,72 s	17,78 s	19,60 s
Tree nodes	82,00	125,00	321,50	226,00
Path nodes	73,00	90,50	217,50	172,00
Path cost	1360,24	1930,52	2242,99	2676,86
T-RRT				
Prepare time	0,00 s	0,00 s	-	-
Total time	10,96 s	77,57 s	-	-
Tree nodes	489,50	595,00	-	-
Path nodes	78,50	83,00	-	-
Path cost	0,05	0,17	-	-
T-RRT-Path				
Prepare time	0.16	0.16	0.07	0.02
Total time	4.69	44.86	481.90	216.31
Tree nodes	134.75	166.0	502.0	265.75
Path nodes	113.00	125.50	285.00	233.25
Path cost	11.65	10.07	60.02	31.21

Table 6.1: Comparison of planning algorithms



(a) : Without obstacles

(b) : Narrow passage 1

(c) : Narrow passage 2

(d) : Two corners

Figure 6.6: Graphs of deviation from ideal position

Chapter 7

Experiments

The path planning method proposed above and MPCs synchronization is experimentally verified in the Gazebo simulator and on the real UAVs. The real experiments are also tested in the simulator, and the results are compared. Some complex experiments are just simulated.

7.1 Robot Operation System (ROS)

The Robot Operation System is open source set of libraries and tools, called middleware that aim to simplify creating complex robots. Middleware is software that provides services for sending messages between applications. ROS itself is not a real-time system. It is set of Linux frameworks and libraries, which is typically operated on Ubuntu distribution. There is a big community, and so it is possible to obtain a many of libraries and packages.

ROS is designed like modular system, consisting of so-called nodes. The ROS Master provides communication between nodes. Each node can provide services and call services of another node. There is also a possibility to exchange data by publishing to a topic. Each node that subscribes topic is announced when the new message appeareds on this topic. The topic can have an arbitrary data structure.

The next useful service provided by ROS is data logging. This service is provided by a tool called rosbag. The rosbag can log communication between nodes and replay it.

A parameter server is provided by ROS. It is used for setting parameters of active nodes. Every active node can save and load data from parameter server.

Graphical utilities are provided by ROS. Provided utilities like **RViz** or **RQT** can be used for data visualization in a real-time.

There are many plugins and integrations like OpenCV for image processing or Gazebo simulator. Gazebo simulator is graphical simulator with a physical engine. The connection with ROS provides the unique way to test the system in the simulator before putting it on the real hardware.

7.2 Real UAVs

The real experiments are carried out on relatively simple maps. Videos of the experiments with real UAVs can be found at `https://youtu.be/nVWqOCK6x24`.

7.2.1 Hardware description

The last part of this thesis is testing these methods on a real UAVs. The testing platform is based on DJI f550 frame (Figure 7.1). The drone has six arms, which gives him a relatively high load capacity. This frame construction is called a hexacopter.

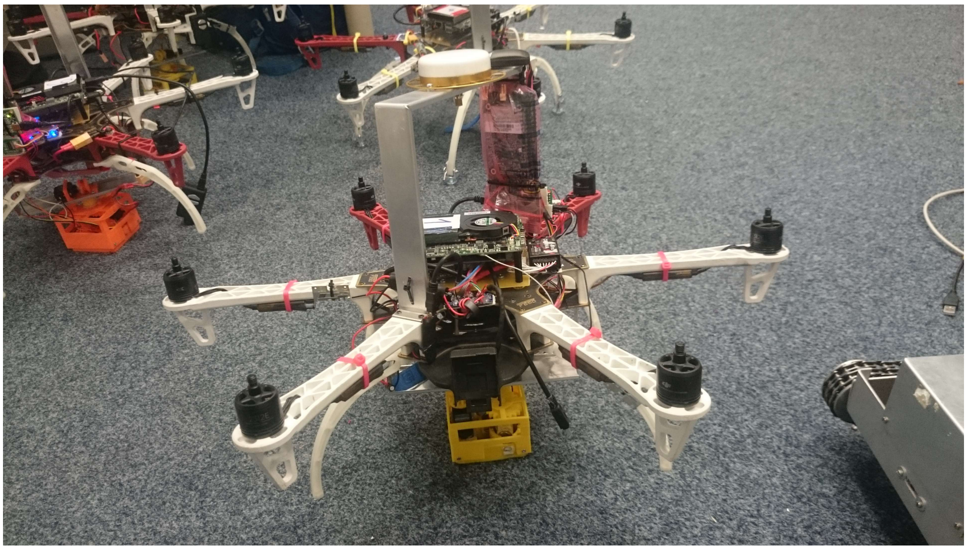
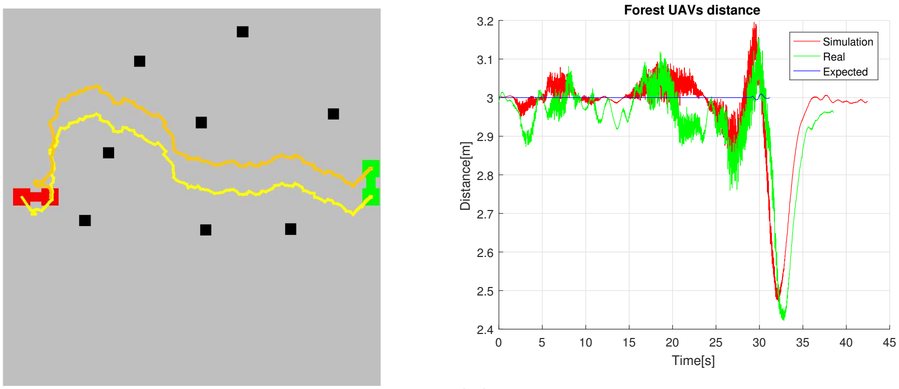


Figure 7.1: Testing platform with gripper designed within this thesis.

Basic stabilization of UAV is done by Pixhawk control unit. The powerful computer with Robot Operating System (ROS) is used for commanding the Pixhawk. UAV is also equipped with a laser distance meter (TeraRanger) for precise height measurement, two cameras providing vision feedback and Real Time Kinematic (RTK) satellite navigation that provides a precise position without unwanted drift. The communication between drones is solved using wifi in the 5 GHz band.

7.2.2 Flight in forest

The first of real experiments is a flight in the environment with few obstacles and without narrow passages. The end orientation of carried object is different from starting orientation. The map of experiment with generated trajectories and photo from the experiment is shown on figure 7.2a. There are shown starting configuration (green), goal configuration (red) and generated trajectories (orange and yellow). On the figure 7.2b is shown the distance between UAVs during real experiment compared with the same experiment in Gazebo simulator and desired values.



(a) : Map of environment

(b) : Graph of distance between UAVs during experiment

Figure 7.2: Flight in the forest experiment

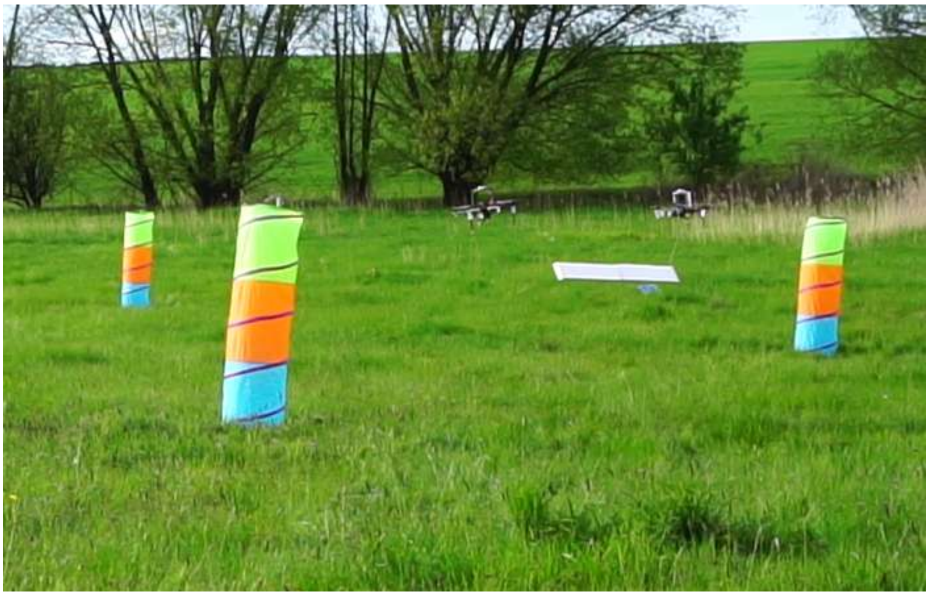


Figure 7.4: Photo from the experiment

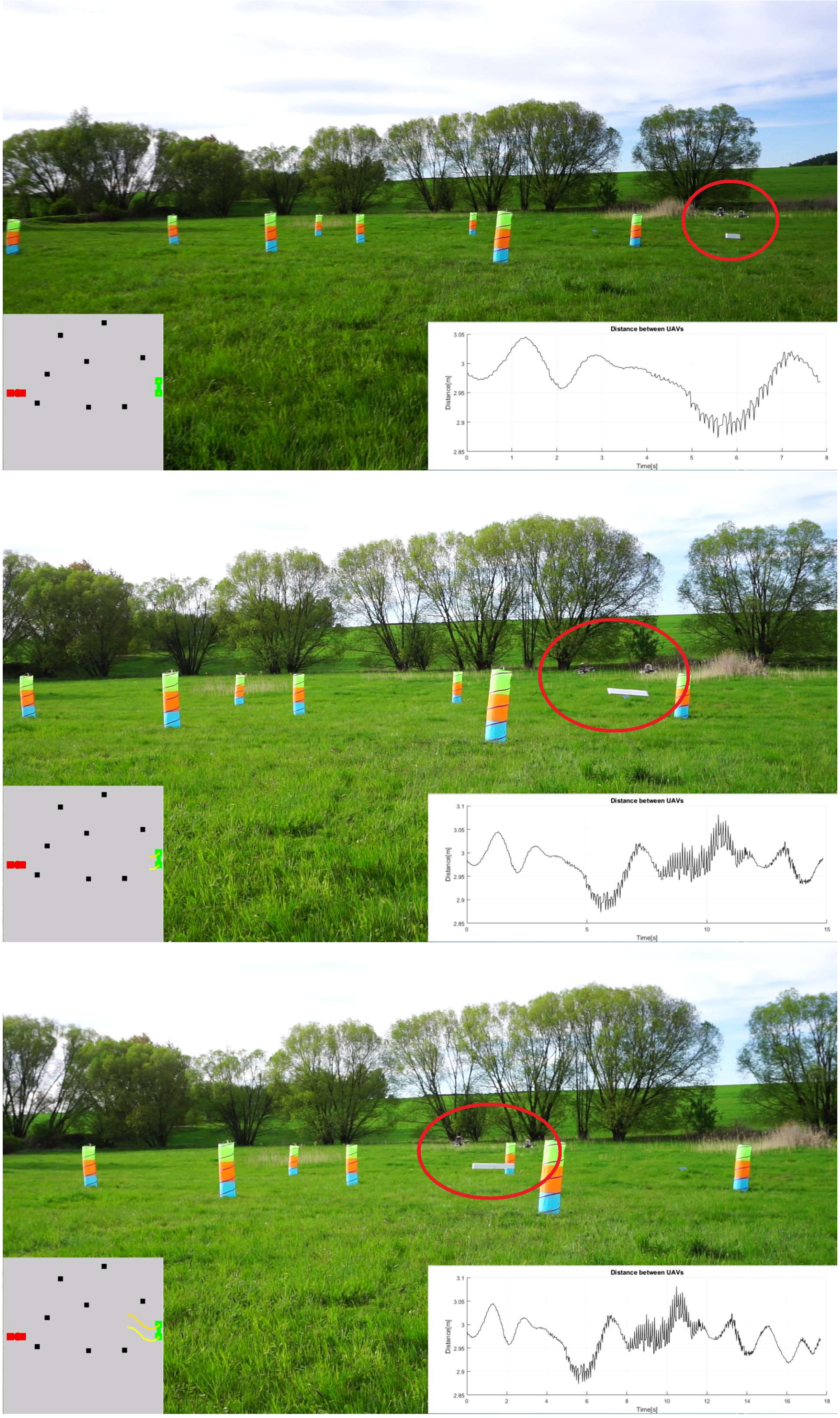


Figure 7.5: Video of the experiment can be found at `https://youtu.be/nVWqOCK6x24`

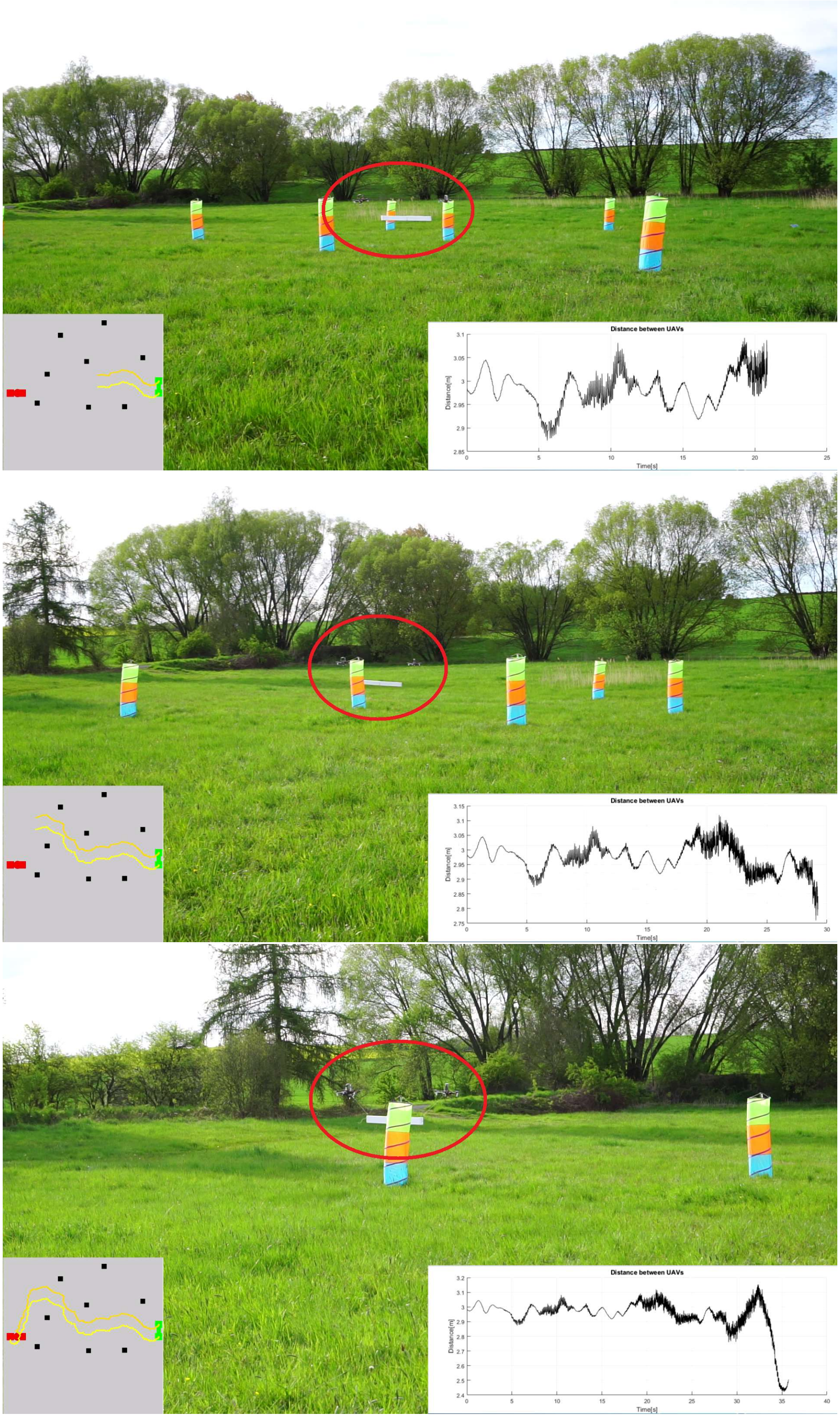
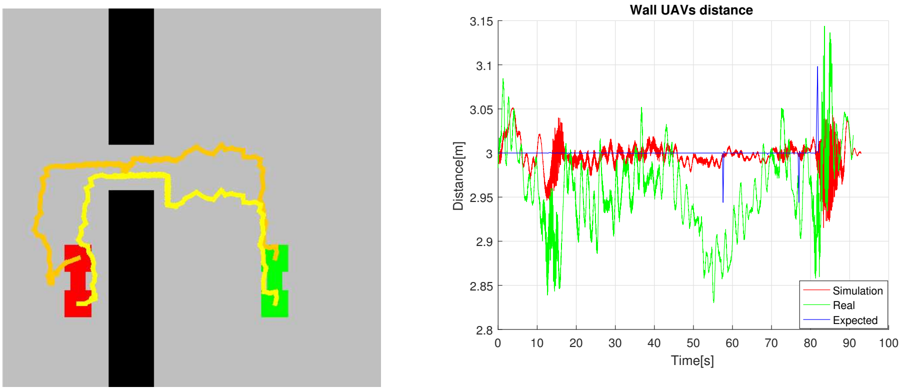


Figure 7.7: Video of the experiment can be found at `https://youtu.be/nVWqOCK6x24`

7.2.3 Flight through the narrow passage

The second experiment with real UAVs is flight through a narrow passage in the wall that is narrower than the spacing between UAVs. In the figure 7.9a is shown the map with generated trajectories and photo from the experiment. The figure 7.9b again compares the real experiment with the simulation and the expected values.



(a) : Map of environment

(b) : Graph of distance between UAVs during experiment

Figure 7.9: Flight through the narrow passage

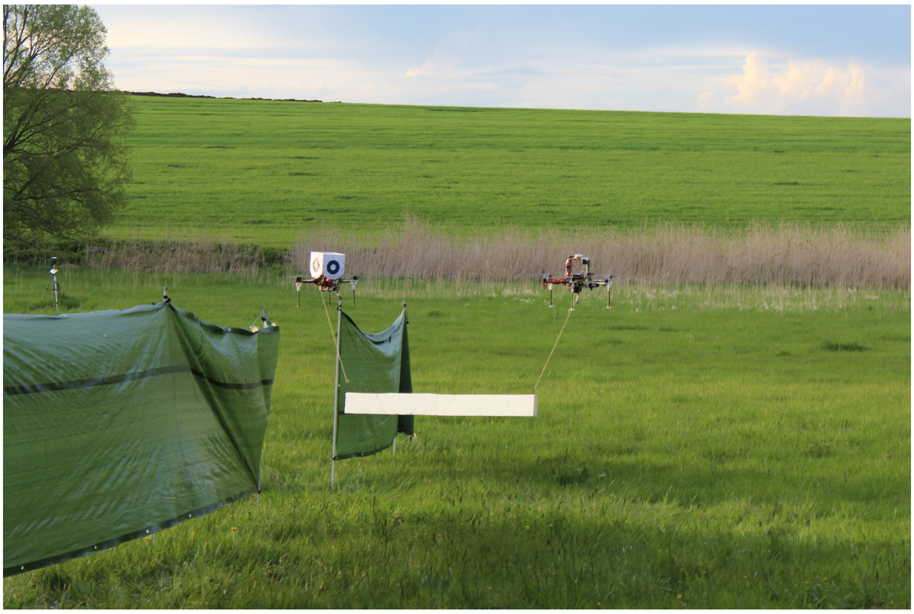


Figure 7.11: Photo from the experiment

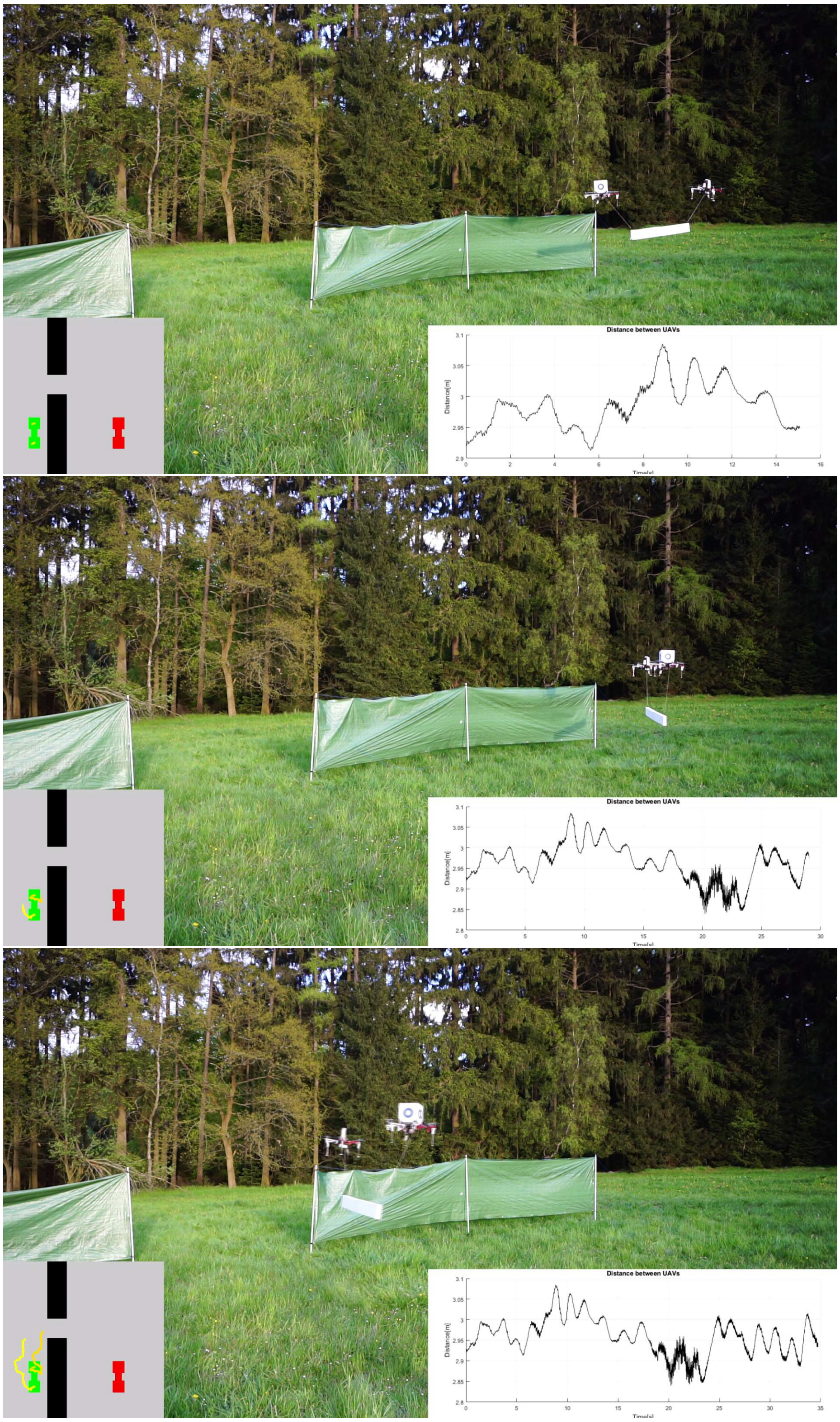


Figure 7.12: Video of the experiment can be found at `https://youtu.be/nVWqOCK6x24`

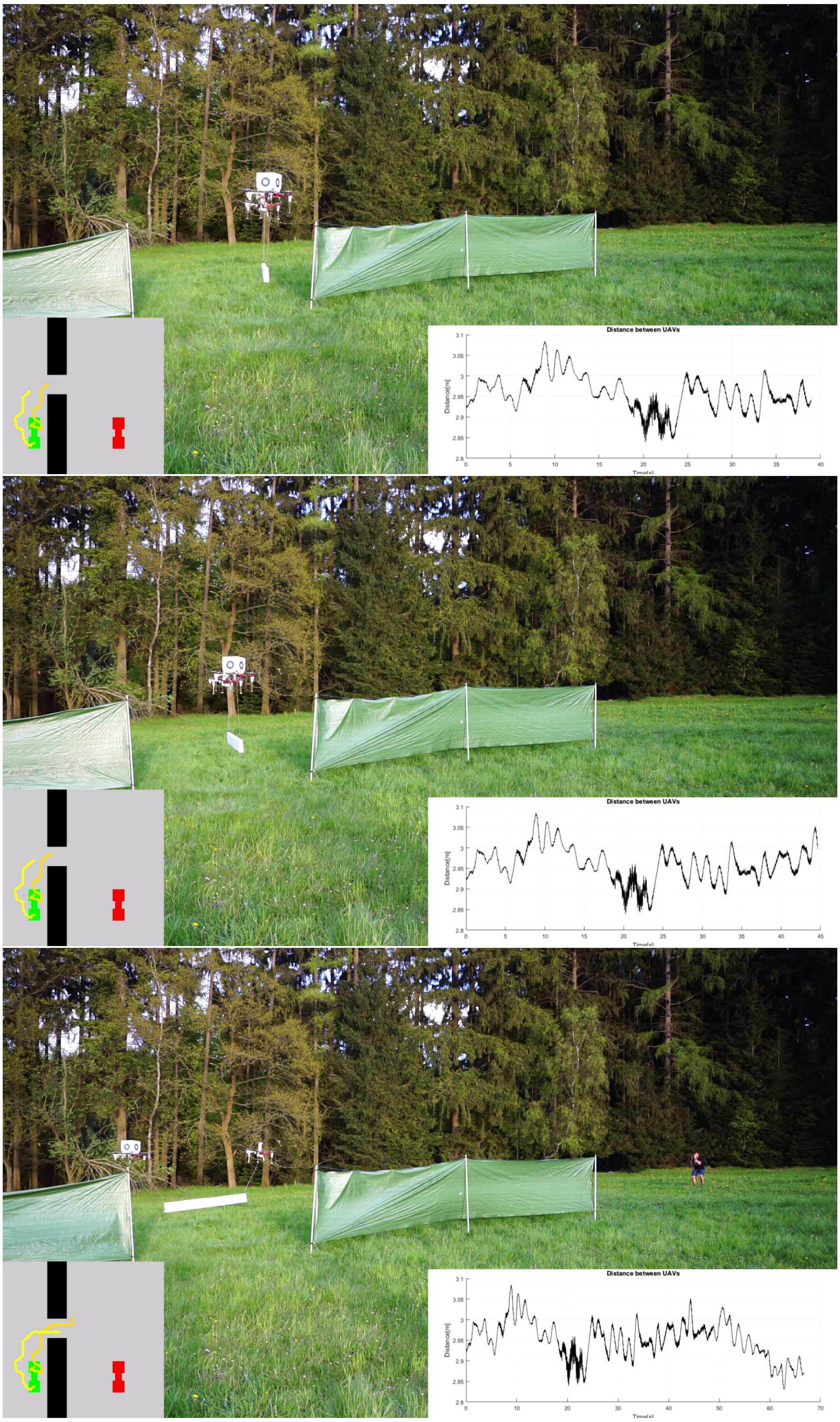


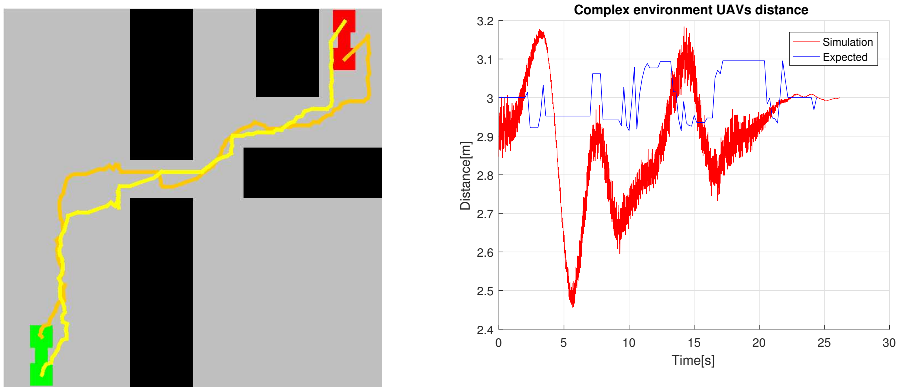
Figure 7.14: Video of the experiment can be found at `https://youtu.be/nVWqOCK6x24`

7.3 Simulation

Simulated situations are a little more complicated than those that have been tested on real drones. Videos of the simulations can be found at `https://youtu.be/lGu6IDJNJT4`.

7.3.1 Flight through narrow passages

First, simulated complicated situations build on one of the real experiments. It is flight through narrow passages, but a little more complex than in the real experiment. The map of the environment and generated trajectories is shown in the figure 7.16a and graph of distance between UAVs in the figure 7.16b.



(a) : Map of environment

(b) : Graph of distance between UAVs during experiment

Figure 7.16: Flight in the complex environment with narrow passages

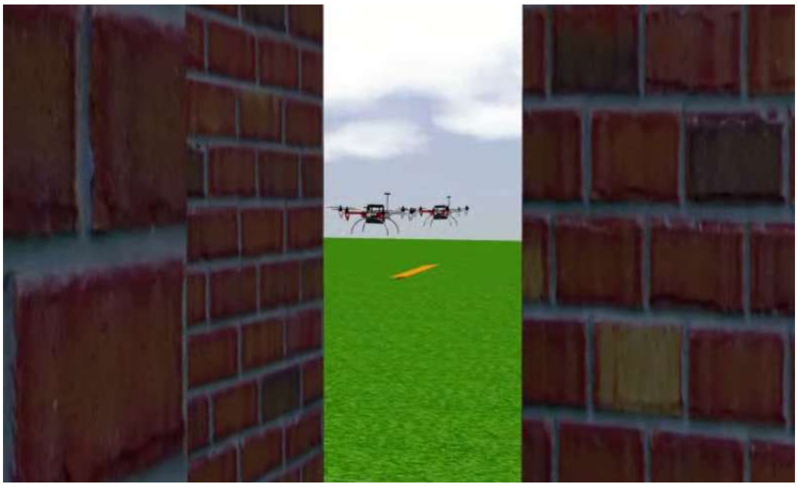


Figure 7.18: Photo from the experiment

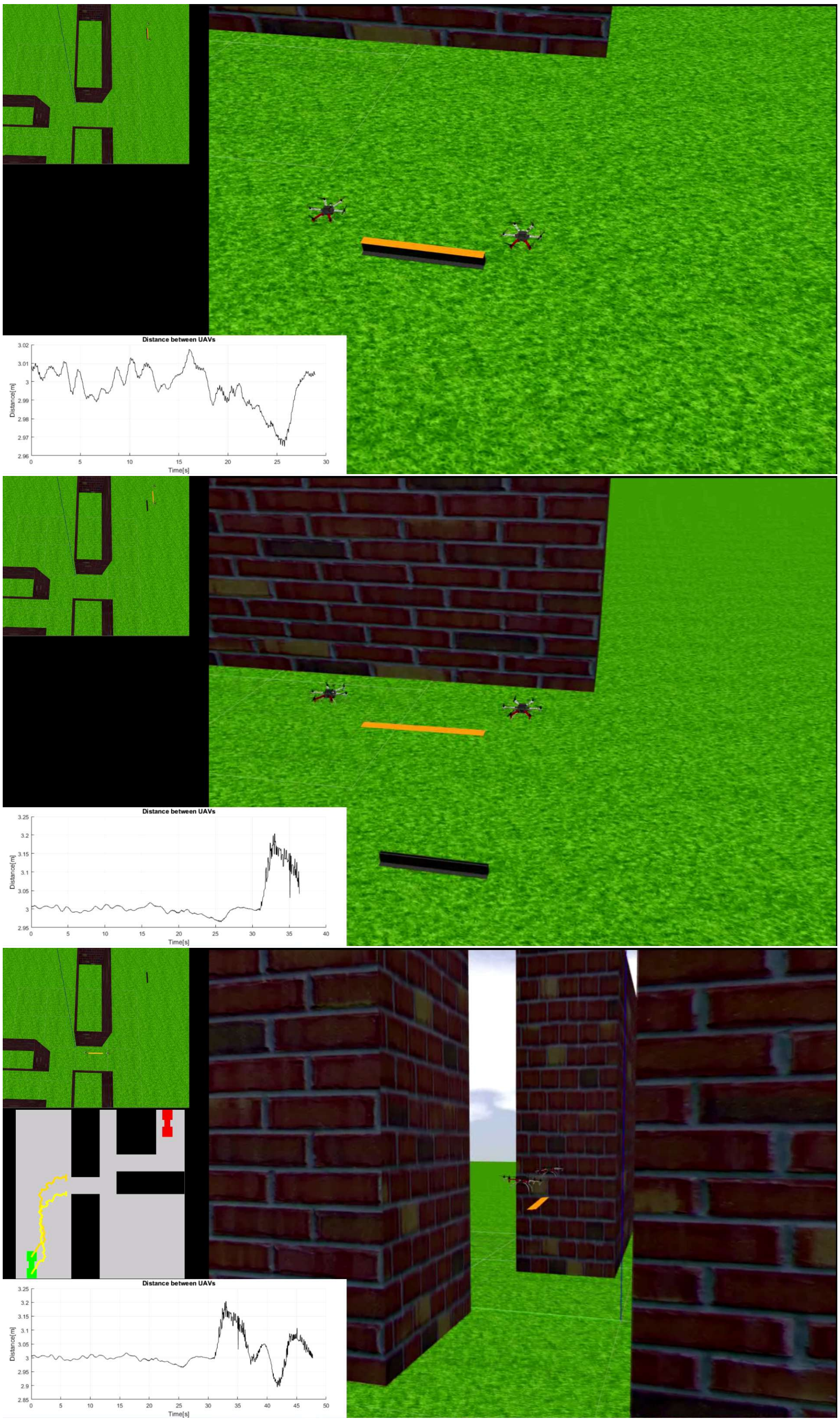


Figure 7.19: Video of the experiment can be found at `https://youtu.be/1Gu6IDJNJT4`

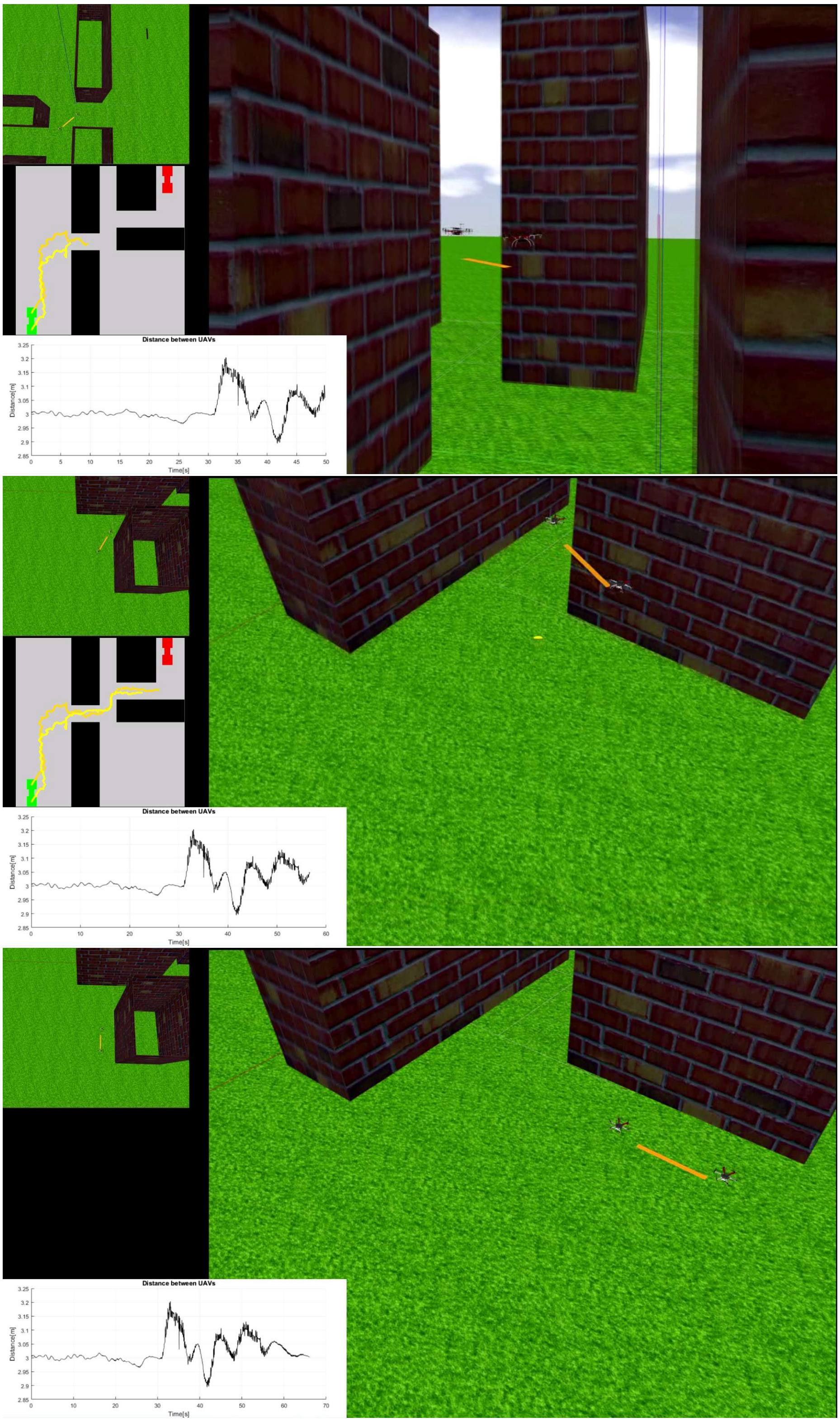
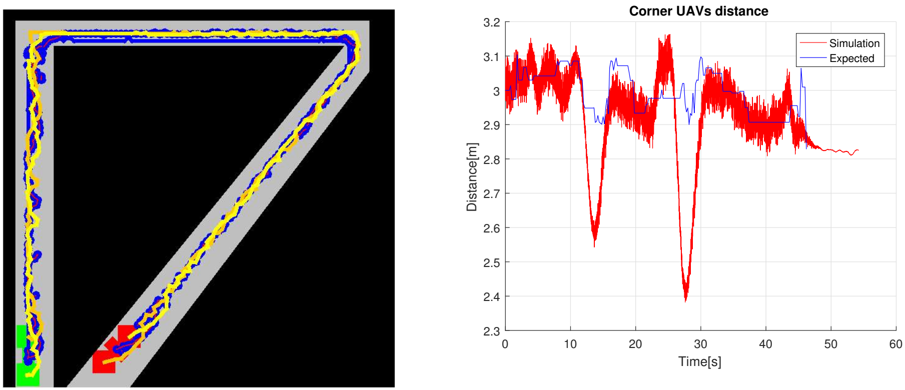


Figure 7.21: Video of the experiment can be found at `https://youtu.be/lGu6IDJNJT4`

7.3.2 Flight through the corners

The last and most complicated experiment is a flying through a narrow passage with two corners. One of these corners has the angle 90° and the other is sharp. As with previous experiments in the figure 7.23a is shown the map of the environment with generated trajectories for both UAVs and in figure 7.23b is compared the distance between UAVs with expected values.



(a) : Map of environment

(b) : Graph of distance between UAVs during experiment

Figure 7.23: Flight through passage with corners

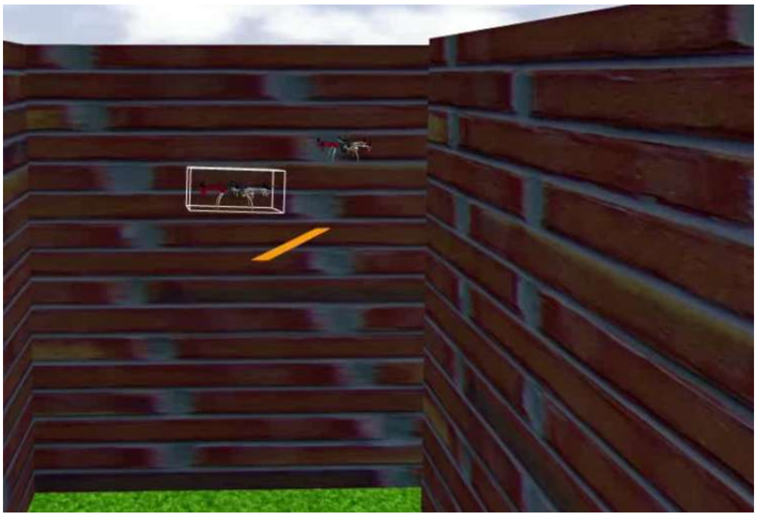


Figure 7.25: Photo from the experiment

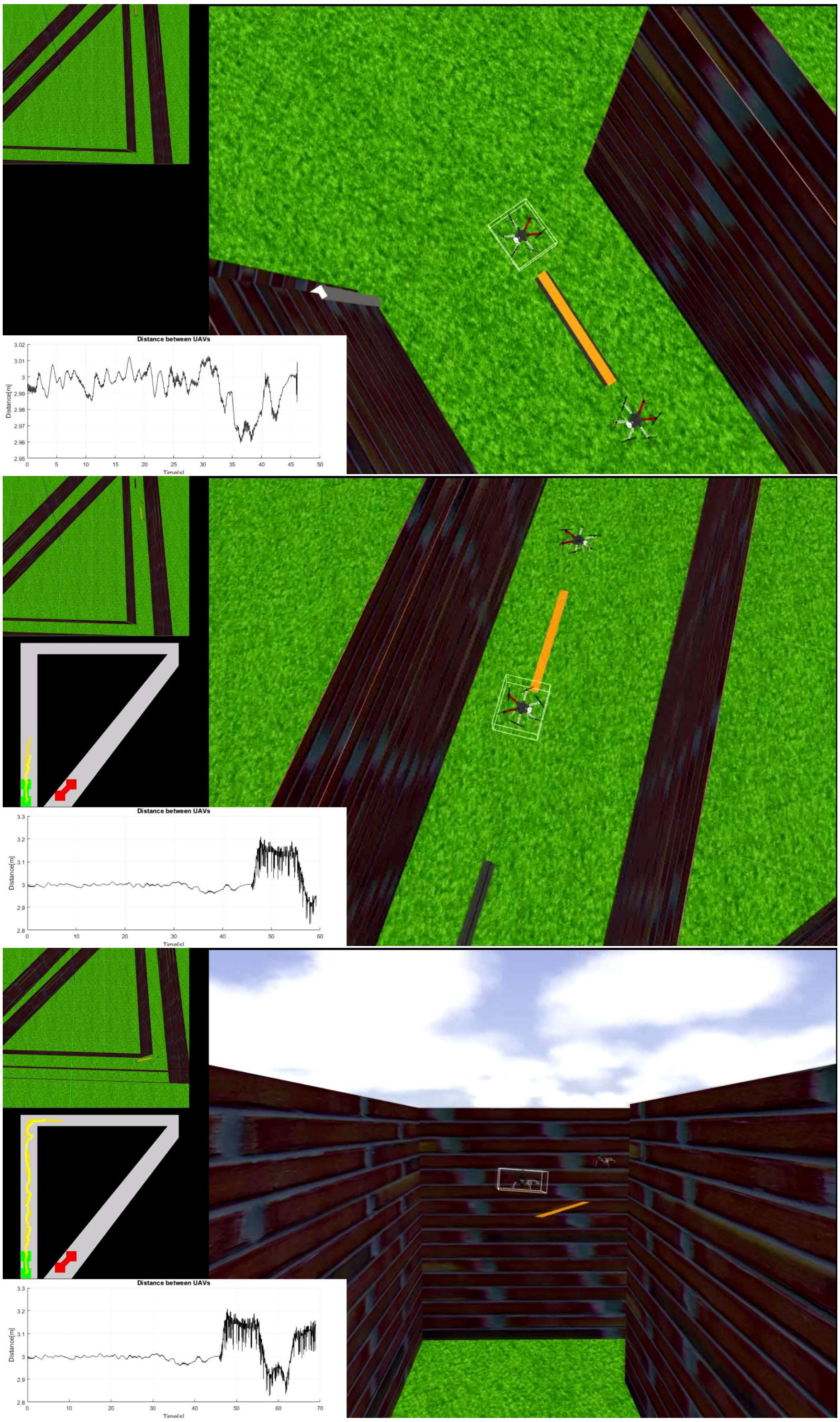


Figure 7.26: Video of the experiment can be found at `https://youtu.be/1Gu6IDJNJT4`

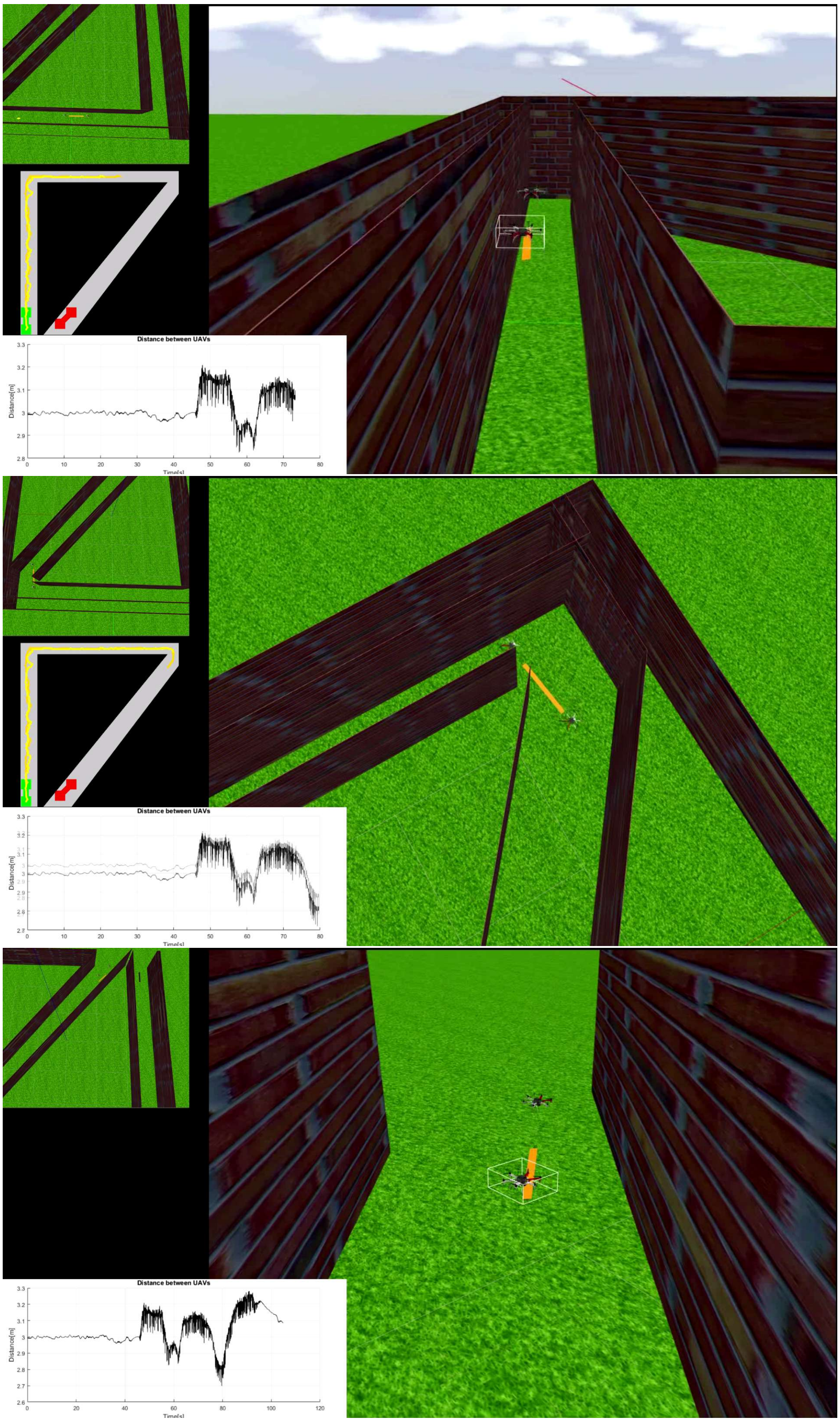


Figure 7.28: Video of the experiment can be found at `https://youtu.be/lGu6IDJNJT4`

Chapter 8

Technical details

8.1 Distance transformation

Distance transformation is a widely used method in digital image processing. The image is composed of feature and non-feature pixels. The distance transformation can convert the binary image into the gray-level image, where all pixels have a value of the distance from the nearest feature pixel like on Figure 8.1. This technique can be used for path planning, by creating distance map from obstacles.

The distance map can be created by many ways. Some methods are slow, or inaccurately and not suitable for path planning. The method introduced in [39] is relatively fast and accurate. This approach uses only nearest neighborhood for computing the distance. Computed distance is propagated over the whole map.

The map is parsed in pixels, where obstacle pixels have value 0, and other pixels have infinity value. The distance is propagated using "mask" which is applied to each pixel first from left to right and from up to down and then apply another "mask" in the opposite direction. The mask pixel is added to the corresponding pixel on map and minimum of this sums is the new value of the pixel as

$$v_{i,j} = \min_{(k,l) \in mask} \left(v_{i+k,j+l} + c(k,l) \right), \tag{8.1}$$

where $v_{i,j}$ is the value in the position (i,j), (k,l) is position in the mask and $c(k,l)$ is the value in the mask on position (k,l).

Various masks can be used, depending on used metric Figure 8.2. The numbers in mask mean distance in the specified direction at transformed image. Positions without number and marked by "-" is not used. Euclidean metric is used in this thesis.

```
∗ ∗ ∗ ∗ ∗ ∗ ∗ ∗ ∗ ∗ ∗ ∗   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
∗ − − − − − − − − − − ∗   0.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.0
∗ − − − − − − − − − − ∗   0.0 1.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 1.0 0.0
∗ − − − − − − − − − − ∗   0.0 1.0 2.0 2.8 2.4 2.0 2.0 2.0 2.4 2.0 1.0 0.0
∗ − − − − − − − − − − ∗   0.0 1.0 2.0 2.4 1.4 1.0 1.0 1.0 1.4 2.0 1.0 0.0
∗ − − − − ∗ ∗ ∗ − − − ∗   0.0 1.0 2.0 2.0 1.0 0.0 0.0 0.0 1.0 2.0 1.0 0.0
∗ − − − − ∗ ∗ ∗ − − − ∗   0.0 1.0 2.0 2.0 1.0 0.0 0.0 0.0 1.0 2.0 1.0 0.0
∗ − − − − ∗ ∗ ∗ − − − ∗   0.0 1.0 2.0 2.0 1.0 0.0 0.0 0.0 1.0 2.0 1.0 0.0
∗ − − − − − − − − − − ∗   0.0 1.0 2.0 2.4 1.4 1.0 1.0 1.0 1.4 2.0 1.0 0.0
∗ − − − − − − − − − − ∗   0.0 1.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 1.0 0.0
∗ − − − − − − − − − − ∗   0.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.0
∗ ∗ ∗ ∗ ∗ ∗ ∗ ∗ ∗ ∗ ∗ ∗   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

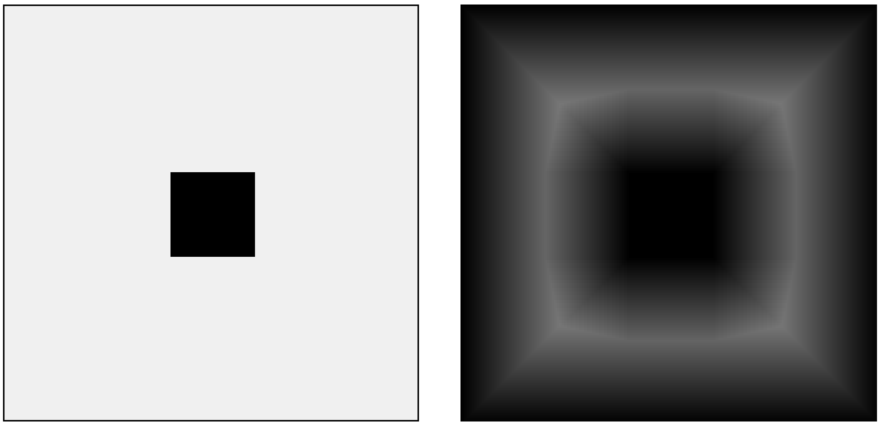


Figure 8.1: Example of distance transformation. Before distance transformation (left) and after distance transformation (right)

$$\textbf{Forward pass} \quad \begin{pmatrix} \sqrt{2} & 1 & \sqrt{2} \\ 1 & 0 & - \\ - & - & - \end{pmatrix} \qquad \begin{pmatrix} - & 1 & - \\ 1 & 0 & - \\ - & - & - \end{pmatrix}$$

$$\textbf{Backward pass} \quad \begin{pmatrix} - & - & - \\ - & 0 & 1 \\ \sqrt{2} & 1 & \sqrt{2} \end{pmatrix} \qquad \begin{pmatrix} - & - & - \\ - & 0 & 1 \\ - & 1 & - \end{pmatrix}$$

Figure 8.2: Mask examples[40]. Euclidean distance (left). Manhattan distance (right)

8.2 Path planing

Path planning is an important discipline in mobile robotic. A basic motion planning problem is to produce a collision-free trajectory between start and goal configuration. There are many path planning methods. Each method has advantages and disadvantages. Two grid-based algorithms are introduced bellow. Grid based algorithms require setting grid resolution. Search is faster with the smaller resolution, but the algorithm can not find the path through a narrow passage. However, the number of points in grid grows exponentially in high-dimension problems.

8.2.1 A*

A* is an informed search algorithm based on minimizing a cost function. It produces an optimal path in a weighted graph. An example of found path is shown on Figure 8.3. The map is passed to the grid, using chosen grid resolution. The grid is converted to the weighted graph.

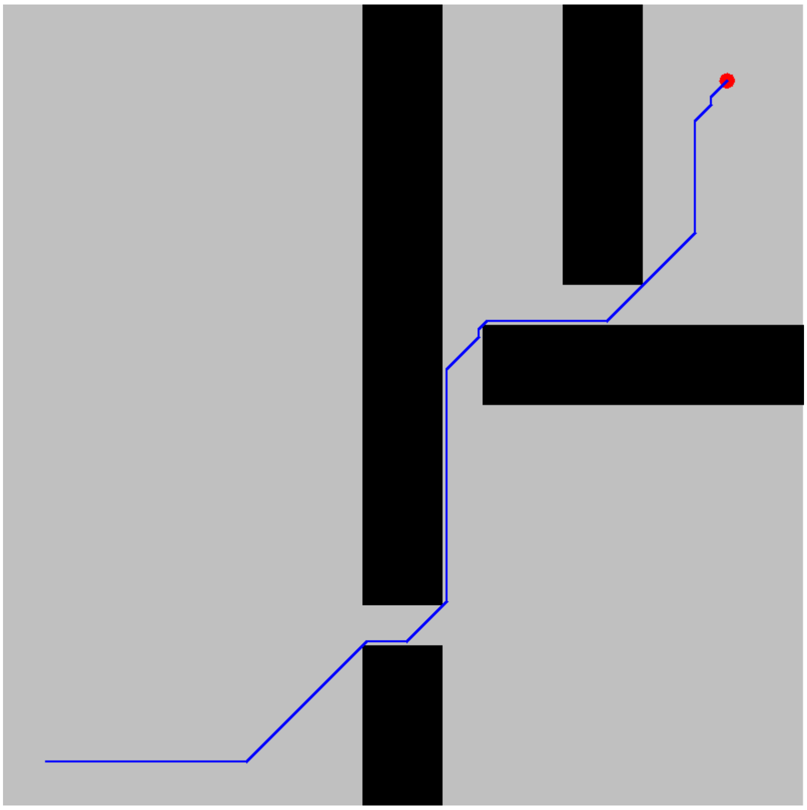


Figure 8.3: Path found by A* algorithm

In each iteration is expanded node with the lowest cost. The tree is constructed from starting node. When a shorter path is found to some node, the original part of the path is replaced by the shorter one. Node priority is given by cost function:

$$f(n) = g(n) + h(n), \tag{8.2}$$

where $g(n)$ is the real distance from the start node to node n, and $h(n)$ is a heuristic function that estimates remaining distance. The heuristic function $h(n)$ must be admissible, that means that it never overestimates the real remaining cost.

8.2.2 Dijkstra algorithm

Dijkstra algorithm is also informed searching algorithm. It is used for searching the shortest path in the weighted graph. The algorithm creates the shortest path tree from starting to each node. Compared to A*, Dijkstra algorithm does not use the heuristic function for estimated remaining distance. Only real distance from start node is used for sorting queue and select right node for expanding.

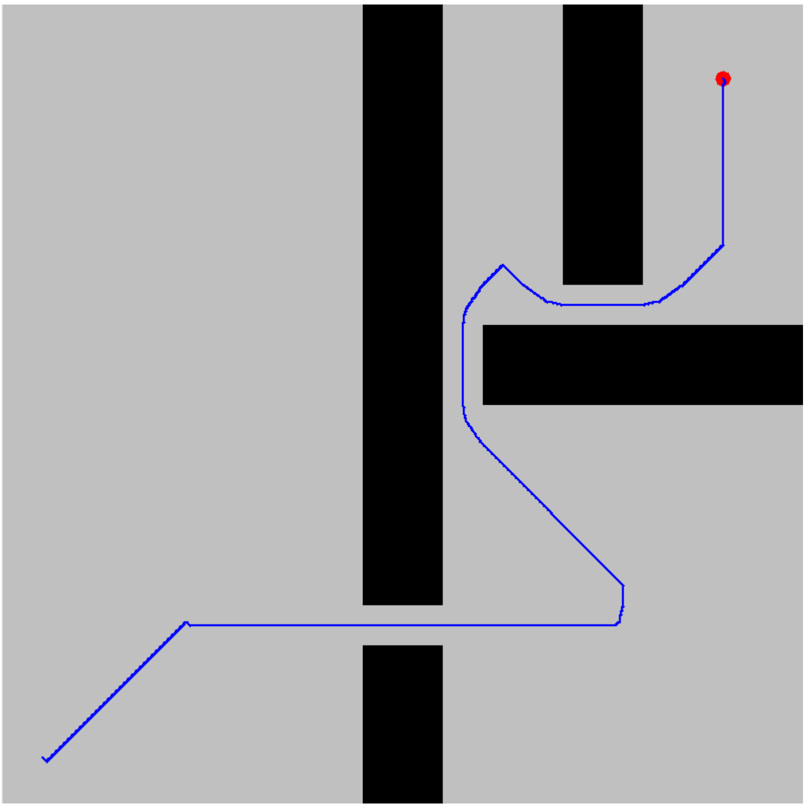


Figure 8.4: Path found by Dijkstra algorithm

The cheapest node is also expanded in each iteration. A neighborhood of expanded nodes is added to open list. If there already is the same node the distance from starting node is compared and if smaller then update it. When the node is picked from the open list, it is added to closed list. A closed list is used for prevent deadlock.

The distance between nodes can be computed by different ways. Euclidean distance can be used as for the A* algorithm, but for our purposes, will be better to use distance map, created in Chapter 8.1. This approach does not provide a minimum path length, but the maximizing distance from obstacles. An example of path found by Dijkstra algorithm and distance map is shown on Figure 8.4.

Chapter 9

Conclusion

In this thesis, the system for autonomous grasping and carrying of objects by a pair of helicopters was developed. The system was designed like state machine which coordinates each phase of object carrying. We designed grasping system, capable of attaching an object with ferromagnetic surface and with a sensor that indicates attachment state. The gripper was integrated into the system, prepared for the MBZIRC competition and tested. The synchronization mechanism of two model predictive controllers was proposed and tested in the simulator and also in real experiments. A planning technique based on a rapid exploration of random trees has been developed to optimize the cost function, and to ensure feasible solution in an environment with narrow passages. Developed planning method was used for planning

The following tasks were completed according to the assignment:

- The system for grasping of objects with metal surface based on permanent magnets and servo mechanism was developed and a sensor that indicates a proper attachment was integrated.

- The gripper was integrated into the system designed for the MBZIRC competition.

- The synchronization of two model predictive controllers being run in parallel on two UAVs was proposed and verified in Gazebo simulator.

- The motion planning technique based on unique combination of TRRT and RRT-Path algorithms was designed and verified in Gazebo simulator.

- The system was experimentally verified on the real platform of the Multi-Robot Systems group.

The whole system was tested in Gazebo simulator. The planning technique was tested on real UAVs, and the results were compared with results from the simulator and expected values. In experiments, it was found that GPS is not accurate and reliable enough for precise positioning. To ensure expected distance between UAVs some technique for relative localization should be implemented.

References

[1] *Mohamed Bin Zayed International Robotic Challenge (MBZIRC)*. URL: http://www.mbzirc.com/.

[2] Martin 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: *The International Journal of Robotics Research* 33.10 (2014), pp. 1393–1412. DOI: 10.1177/0278364914530482. eprint: http://dx.doi.org/10.1177/0278364914530482. URL: http://dx.doi.org/10.1177/0278364914530482.

[3] Martin Saska et al. “Fault-Tolerant Formation Driving Mechanism Designed for Heterogeneous MAVs-UGVs Groups”. In: *Journal of Intelligent & Robotic Systems* 73.1 (2014), pp. 603–622. ISSN: 1573-0409. DOI: 10.1007/s10846-013-9976-6. URL: http://dx.doi.org/10.1007/s10846-013-9976-6.

[4] V. Spurny, T. Baca, and M. Saska. “Complex manoeuvres of heterogeneous MAV-UGV formations using a model predictive control”. In: *2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*. Aug. 2016, pp. 998–1003. DOI: 10.1109/MMAR.2016.7575274.

[5] Martin Saska, Zdenek Kasl, and Libor Přeucil. “Motion planning and control of formations of micro aerial vehicles”. In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 1228–1233. ISSN: 1474-6670. DOI: http://dx.doi.org/10.3182/20140824-6-ZA-1003.02295. URL: http://www.sciencedirect.com/science/article/pii/S1474667016417817.

[6] M. Saska et al. “Navigation, localization and stabilization of formations of unmanned aerial and ground vehicles”. In: *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. May 2013, pp. 831–840. DOI: 10.1109/ICUAS.2013.6564767.

[7] M. Saska et al. “Coordination and navigation of heterogeneous UAVs-UGVs teams localized by a hawk-eye approach”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Oct. 2012, pp. 2166–2171. DOI: 10.1109/IROS.2012.6385517.

[8] M. Saska, T. Baca, and D. Hert. “Formations of unmanned micro aerial vehicles led by migrating virtual leader”. In: *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. Nov. 2016, pp. 1–6. DOI: 10.1109/ICARCV.2016.7838801.

[9] M. Saska. “MAV-swarms: Unmanned aerial vehicles stabilized along a given path using onboard relative localization”. In: *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. June 2015, pp. 894–903. DOI: 10.1109/ICUAS.2015.7152376.

[10] M. Saska, J. Vakula, and L. Přeućil. “Swarms of micro aerial vehicles stabilized under a visual relative localization”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. May 2014, pp. 3570–3575. DOI: 10.1109/ICRA.2014.6907374.

[11]V. Vonasek et al. “RRT-Path: a guided Rapidly exploring Random Tree”. In: *Robot Motion and Control 2009*. 2009.

[12]Vojtěch Vonásek et al. “Motion planning with adaptive motion primitives for modular robots”. In: *Applied Soft Computing* 34 (2015), pp. 678–692. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2015.05.002. URL: http://www.sciencedirect.com/science/article/pii/S1568494615003026.

[13]Vojtěch Vonásek et al. “High-level motion planning for CPG-driven modular robots”. In: *Robotics and Autonomous Systems* 68 (2015), pp. 116–128. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2015.01.006. URL: http://www.sciencedirect.com/science/article/pii/S0921889015000147.

[14]Martin 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. ISSN: 1573-0409. DOI: 10.1007/s10846-016-0338-z. URL: http://dx.doi.org/10.1007/s10846-016-0338-z.

[15]Martin 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* 41.4 (2017), pp. 919–944. ISSN: 1573-7527. DOI: 10.1007/s10514-016-9567-z. URL: http://dx.doi.org/10.1007/s10514-016-9567-z.

[16]M. Saska et al. “Autonomous deployment of swarms of micro-aerial vehicles in cooperative surveillance”. In: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. May 2014, pp. 584–595. DOI: 10.1109/ICUAS.2014.6842301.

[17]Korsak K. Meenen K. Meyers D. and Piasecki F. *MULTI-HELICOPTER HEAVY LIFT SYSTEM FEASIBILITY STUDY*. 1972. URL: http://www.dtic.mil/dtic/tr/fulltext/u2/743516.pdf.

[18]Mark D. Ardema. *Vehicle Concepts and Technology Requirements for Buoyant Heavy-Lift Systems*. 1981. URL: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19810022643.pdf.

[19]Alex Kushleyev et al. “Towards a swarm of agile micro quadrotors”. In: *Autonomous Robots* 35.4 (2013), pp. 287–300. ISSN: 1573-7527. DOI: 10.1007/s10514-013-9349-9. URL: http://dx.doi.org/10.1007/s10514-013-9349-9.

[20]Nathan Michael, Jonathan Fink, and Vijay Kumar. “Cooperative Manipulation and Transportation with Aerial Robots”. In: *Auton. Robots* 30.1 (Jan. 2011), pp. 73–86. ISSN: 0929-5593. DOI: 10.1007/s10514-010-9205-0. URL: http://dx.doi.org/10.1007/s10514-010-9205-0.

[21]Daniel Mellinger et al. “Cooperative Grasping and Transport Using Multiple Quadrotors”. In: *Distributed Autonomous Robotic Systems: The 10th International Symposium*. Ed. by Alcherio Martinoli et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 545–558. ISBN: 978-3-642-32723-0. DOI: 10.1007/978-3-642-32723-0_39. URL: http://dx.doi.org/10.1007/978-3-642-32723-0_39.

[22]Dr. Fedotov V.A. *BURAN Orbital Spaceship Airframe Creation, Air Transportation.* URL: http://www.buran-energia.com/documentation/documentation-akc-air-transportation.php.

[23]Iñaki Navarro and Fernando Matía. "An Introduction to Swarm Robotics". In: *ISRN Robotics* (2013). DOI: 10.5402/2013/608164. URL: http://dx.doi.org/10.5402/2013/608164.

[24]Erol Şahin. "Swarm Robotics: From Sources of Inspiration to Domains of Application". In: *Swarm Robotics: SAB 2004 International Workshop, Santa Monica, CA, USA, July 17, 2004, Revised Selected Papers.* Ed. by Erol Şahin and William M. Spears. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 10–20. ISBN: 978-3-540-30552-1. DOI: 10.1007/978-3-540-30552-1_2. URL: http://dx.doi.org/10.1007/978-3-540-30552-1_2.

[25]M. Anthony Lewis and Kar-Han Tan. "High Precision Formation Control of Mobile Robots Using Virtual Structures". In: *Autonomous Robots* 4.4 (1997), pp. 387–403. ISSN: 1573-7527. DOI: 10.1023/A:1008814708459. URL: http://dx.doi.org/10.1023/A:1008814708459.

[26]František Duchoň et al. "Path Planning with Modified a Star Algorithm for a Mobile Robot". In: *Procedia Engineering* 96 (2014), pp. 59–69. ISSN: 1877-7058. DOI: http://dx.doi.org/10.1016/j.proeng.2014.12.098. URL: http://www.sciencedirect.com/science/article/pii/S187770581403149X.

[27]T. Paul, T. R. Krogstad, and J. T. Gravdahl. "UAV formation flight using 3D potential field". In: *2008 16th Mediterranean Conference on Control and Automation.* June 2008, pp. 1240–1245. DOI: 10.1109/MED.2008.4601984.

[28]Juan D. Contreras, Fernando Martínez S., and Fredy H. Martínez S. *Path planning for mobile robots based on visibility graphs and A* algorithm.* 2015. DOI: 10.1117/12.2197062. URL: http://dx.doi.org/10.1117/12.2197062.

[29]LaValle S. M. *Rapidly–exploring random trees: A new tools for path planning.* 1998.

[30]Y. Dong, C. Fu, and E. Kayacan. "RRT-based 3D path planning for formation landing of quadrotor UAVs". In: *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV).* Nov. 2016, pp. 1–6. DOI: 10.1109/ICARCV.2016.7838567.

[31]Sertac Karaman and Emilio Frazzoli. "Incremental Sampling-based Algorithms for Optimal Motion Planning". In: *CoRR* abs/1005.0416 (2010). URL: http://arxiv.org/abs/1005.0416.

[32]J. Cortés L. Jaillet and T. Siméon. "Sampling-Based Path Planning on Configuration-Space Costmaps". In: *IEEE Transactions on Robotics* 26.4 (Aug. 2010), pp. 635–646.

[33]L. Ros M. Manubens D. Devaurs and J. Cortés. "Motion Planning for 6-D Manipulation with Aerial Towed-cable Systems, Proceedings of Robotics". In: *Science and Systems* (2013).

[34]Marcos G. Berrios Mark B. Tischler Luigi S. Cicolani J. David Powell. *Stability, Control, and Simulation of a Dual Lift System Using Autonomous R-MAX Helicopters.* 2014. URL: `http://nams.usra.edu/NAMS/assets/AFDD/AHS_2014_Berrios.pdf`.

[35]ZuQun Li Joseph F. Horn Jack W. Langelaan. "Coordinated Transport of a Slung Load by a Team of Autonomous Rotorcraft". In: *AIAA Guidance Navigation and Control Conference* (2014).

[36]G Loianno T Baca 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).

[37]*Magnetic Field and Magnetic lines of Force.* URL: `http://electricalacademia.com/electromagnetism/magnetic-field-and-magnetic-lines-of-force/`.

[38]*Multi-robot Systems group - Mohamed Bin Zayed International Robotics Challenge.* URL: `http://mrs.felk.cvut.cz/projects/mbzirc`.

[39]GUNILLA BORGEFORS. "Distance Transformations in Digital Images". In: *COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING* (1985).

[40]George J. Grevera. "DISTANCE TRANSFORM ALGORITHMS AND THEIR IMPLEMENTATION AND EVALUATION". In: *Deformable Models, pp.33-60.* 1970.

Appendix A

CD Content

Directory	Description
trajectory-planner	Trajectory planner project in JAVA
ros_nodes	Sources of ROS nodes used in this thesis
gripper_model	3D models of gripper if stl format
gripper_firmware	Sources of gripper firmware
text	This thesis in pdf format
videos	Videos from the experiments

Table A.1: CD Content

Appendix B

List of abbreviations

Abbreviation	Meaning
3D	Three dimensional
CTU	Czech Technical University
DOF	Degree of freedom
EEPROM	Electrically Erasable Programmable Read-Only Memory
GPS	Global positioning system
MAV	Micro aerial vehicle
MBZIRC	Mohamed Bin Zayed International Robotics Challenge
MPC	Model predictive control
MRS	Multi-robot Systems
ROS	Robot operating system
RRT	Rapidly exploring random tree
RTK	Real time kinematic
T-RRT	Transition-based rapidly exploring random tree
UART	Universal asynchronous receiver transmitter
UAV	Unnamed aerial vehicle
US	United states
USB	Universal serial bus
VTOL	Vertical take-off and landing
WiFi	Wireless local area network

Table B.1: Lists of abbreviations