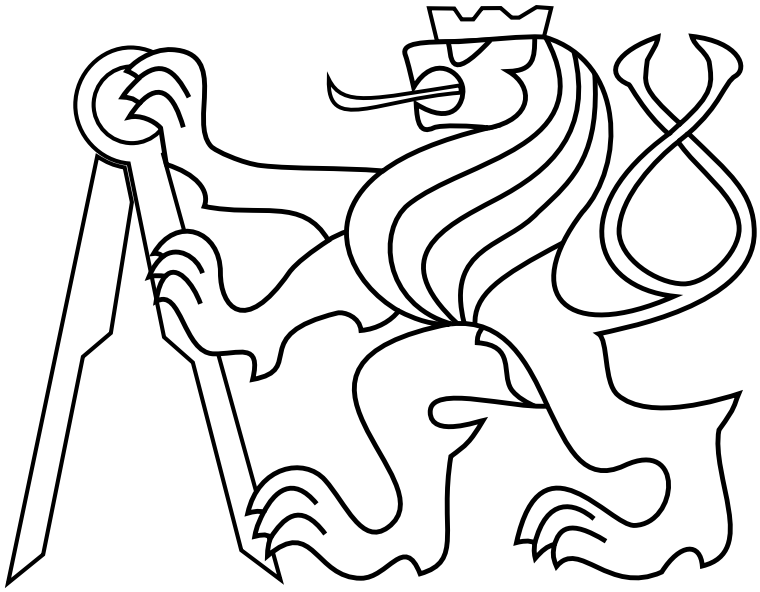
CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

# MASTER'S THESIS



Jiří Horyna

## Automatic Control for Collaborative Payload Carrying by a Pair of Unmanned Helicopters

**Department of Control Engineering**

Thesis supervisor: **Ing. Tomáš Báča**

## 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 thesis.

Prague, date..... .....

signature

# MASTER'S THESIS ASSIGNMENT

## I. Personal and study details

Student's name: **Horyna Jiří** Personal ID number: **457177**

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

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

Study program: **Cybernetics and Robotics**

Branch of study: **Cybernetics and Robotics**

## II. Master's thesis details

Master's thesis title in English:

**Automatic control for collaborative payload carrying by a pair of unmanned helicopters**

Master's thesis title in Czech:

**Automatické řízení pro kolaborativní nošení nákladu dvěma bezpilotními helikoptérami**

Guidelines:

The thesis aims to design, implement, and verify a system of automatic control of unmanned aerial vehicles (UAVs) for the collaborative carrying of an object. The motivation arises from the MBZIRC 2020 robotic challenge, where a group of UAVs is tasked with building a brick wall. The central part of this work is to design a specialized gripper for carrying a ferrous object. The thesis will be divided into the following tasks:
1) To design a gripper that will allow carrying of a large object by a pair of UAVs as well as carrying of smaller objects by a single UAV.
2) The gripper will be assembled as a hardware prototype as well as a model in the realistic Gazebo simulator. Both the HW prototype and simulator model will be integrated into the current UAV system via the Robot Operating System (ROS).
3) To design a collaborative grasping guidance system. This system should secure synchronized attaching and detaching of an object by both of the UAVs.
4) The student will design an automatic control method for a task of collaborative load-carrying by pair of UAV. The designed method will be verified in the realistic Gazebo simulator.
5) The student will prepare the system for real experiment verification that will be realized in future work.
The complete task of the automatic wall assembly is a complex challenge. This thesis should not solve other tasks, such as visual object detection, UAV localization, placement spot detection, and others. The student should collaborate with other members of the MRS team and either receive working modules or emulate them if necessary.

Bibliography / sources:

[1] Quigley, Morgan, et al. "ROS: an open-source Robot Operating System." ICRA workshop on open source software. Vol. 3. No. 3.2. 2009.
[2] Loianno, Giuseppe, et al. "Localization, grasping, and transportation of magnetic objects by a team of mavs in challenging desert-like environments." IEEE Robotics and Automation Letters 3.3 (2018): 1576-1583.
[3] Rossi, Enrica, et al. "Cooperative Aerial Load Transportation via Sampled Communication." IEEE Control Systems Letters 4.2 (2019): 277-282.

Name and workplace of master's thesis supervisor:

**Ing. Tomáš Báča, Multi-robot Systems, FEE**

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

**Ing. Martin Saska, Dr. rer. nat., Multi-robot Systems, FEE**

Date of master's thesis assignment: **10.01.2020** Deadline for master's thesis submission: **22.05.2020**

Assignment valid until:
**by the end of summer semester 2020/2021**

| Ing. Tomáš Báča | prof. Ing. Michael Šebek, DrSc. | prof. Mgr. Petr Páta, Ph.D. |
|---|---|---|
| Supervisor's signature | Head of department's signature | Dean's signature |

## III. Assignment receipt

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

. _____
Date of assignment receipt

_____
Student's signature

## Acknowledgements

Firstly I would like to thank my supervisor Ing. Tomáš Báča for his great support during this thesis. I wish to thank Dr. Martin Saska for the opportunity to work on interesting projects. Furthermore, I thank other members of Multi-robot Systems group for their advice. My thanks also belong to my family for providing me the opportunity to study.

*Abstract*

This thesis deals with design of a system for cooperative manipulation tasks with two Unmanned Aerial Vehicles (UAV). We propose a gripping system in a form of simulator model and hardware prototype, both integrated into the current UAV system at the Multi-robot System laboratory. The main focus of this thesis deals with design of an automatic control method for collaborative motion of the UAVs. A cascade PID controller is implemented together with a guidance system (in a form of state automata) that ensures synchronized task performance. Simulated experiments verified the capability of the system to achieve collaborative manipulation tasks. The proposed system was prepared for oncoming real experiments.

**Keywords:** cooperative aerial manipulation, unmanned aerial vehicles, aerial object manipulation, cascade control, PID controller

*Abstrakt*

Tato práce se zabývá návrhem systému pro úlohy kooperativní letecké manipulace se dvěma bezpilotními letouny. Předkládáme systém pro uchopení předmětu ve formě modelu pro simulátor a HW prototypu. Obě části byly začleněny do systému bezpilotních letounů, který je nyní používán v laboratoři Multirobotických systémů. Tato práce se zabývá především návrhem automatického řízení pro kooperativní pohyb bezpilotních letounů. Byl implementován kaskádní PID regulátor společně s naváděcím systémem (ve formě stavového automatu), který zajišťuje synchronizované plnění úkolu. Simulované experimenty ověřily schopnost systému splnit úlohy kooperativního nesení předmětu. Navržený systém byl připraven na nadcházející reálné experimenty.

**Klíčová slova:** kooperativní letecká manipulace, bezpilotní letouny, manipulace s objekty za letu, kaskádní řízení, PID regulátor

# Contents

# List of Figures

# 1 Introduction

Unmanned aerial vehicle (UAV) is an aircraft, where a pilot is not present onboard. Such a device can be controlled remotely by an operator or its motion can be autonomous. The first group of UAVs takes the form of an airplane — an aircraft with conventional take-off and landing (CTOL). They are the most commonly used by the military during surveys or combat missions. However, they also take place in the civilian sector in police monitoring, terrain survey, or hobby applications (remotely-controlled model airplanes). Vehicles with vertical take-off and landing (VTOL) belong to another group of UAVs. This specific type of aircraft has expanded into many fields of application. They found their place in video making and photography, hobby sector and autonomous delivery. Another significant sector of use is research, where VTOL UAVs find wide applications for example in swarm systems [32], aerial manipulation [11], or in air safety [10]. This vehicle is understood as a multicopter (see figure 1) in the sense of this thesis. Multicopter is a device with typically 4 or more rotors attached to a rigid symmetrical body. Brushless DC electromotors are commonly used as rotors with ESC (Electronic Speed Control) drivers. We can control the tilt and thrust of such UAV by controlling the speed of individual propellers, that are attached to motors. The multicopters in this thesis have the task of collaborative aerial manipulation.

When we talk about aerial manipulators we mean UAVs are equipped with a grasping device or robotic arm. Such a system is able to interact with its environment. However, many challenges in the control of such a system arise. Challenges arise from disturbances,

Figure 1: An example of multirotor UAV used in the MRS laboratory.

for example, gusts during the approach to an object that needs to be grasped. Gusts have origin in the rotation of propellers and they can influence position or orientation of a lying object. Another challenge in control may occur when undesirable forces and moments caused due to heavy payload are acting on the UAV. This puts high requirements for precise and robust control strategies.

A single UAV may not be sufficient for aerial manipulation for several reasons. Large and heavy objects can cause forces and moments that make take-off or control of single UAV nearly impossible while using conventional multicopter. It brings us to use cooperative aerial manipulation in these scenarios, where several multicopters are carrying some payload. However, this approach puts high requirements for HW and SW of the collaborative system. Requirements primarily consist of precise and robust control of UAVs. The need for precise localization is the biggest challenge of this approach since RTK (real-time kinematics) GNSS system achieves accuracy in the order of centimeter even when a mobile RTK station is used. Such accuracy may not be sufficient for collaborative manipulation tasks, where rigid-type grippers are used. In that case, forces between UAVs and grasped object arise. To solve this issue, force sensors need to be applied. Further, relative localization [36] can be used as an alternative localization approach. Another requirement can be found in gripper systems design. It has to secure proper and synchronous grasping of a payload by all of the used UAVs.

This thesis deals with the design, implementation, and verification of control methods for a pair of UAVs carrying a solid payload. A specialized gripper that is suitable for such a task need to be designed and assembled as well. Our goal was to find such a solution that can solve or bypass the challenges described above to be able to use a designed system outside the laboratory environment. Thus, the solution does not rely on accurate localization method.

## 1.1 Problem statement and motivation

The task of this thesis is to design and implement a controller that will control a pair of helicopters manipulating with a solid object. A cooperative gripper needs to be designed and assembled as a hardware prototype. The grasping system will be then integrated within the current UAV system. Further, a simulated environment will be customized for designed controller verification. This approach was chosen with regards to the need test and debug the new system before it will be tested in a real experiment. The designed control system is in this thesis completed with a grasping and fail-safe system. The first system secures synchronized attaching and detaching of payload whereas the fail-safe system detects collaborative motion failures.

The aim is to design a system that could be used outside a laboratory environment. Specifically, the task of this thesis was motivated by the second challenge of the MBZIRC


Figure 2: Pair of UAVs during wall assembly challenge in MBZIRC 2020. The system designed in this thesis is intended for the largest orange bricks.

2020 competition, where a brick wall has to be assembled by several UAVs. We were looking for a solution that will be reliable enough with standard sensor equipment. The mentioned wall assembly challenge is shown in figure 3. Use of data from camera for vision-based brick localization is mentioned in this thesis. However, this thesis deals with control tasks. Image processing for real experiment was already solved for MBZIRC 2020 competition.

### 1.1.1 MBZIRC

The Mohamed Bin Zayed International Robotics Challenge (MBZIRC) [19] is an international robotic competition held by Khalifa University in Abu Dhabi. The best research teams in the world compete in several challenges during the MBZIRC competition. Each challenge is motivated by real-world problems that may be solved by autonomous robots. The main purpose of MBZIRC is to push technological limits in robotics forward. The first competition took place in 2017 when teams from around the world faced 3 challenges. In the first one, a UAV land on moving vehicle [31]. In the second one, a ground robot had to locate and operate a valve stem [33]. The last one tasked several UAVs to localize, grasp, and place a set of ferrous objects [31] (see figure 3). Multi-robot Systems group competed in the first and third challenge at MBZIRC 2017. The team from the Czech Technical University in Prague cooperated with the University of Pennsylvania and the University of Lincoln. Together they managed to win Challenge 3, they took the second place in Challenge 1 and the third place in the Grand Challenge, where all three challenges were held together.

The last MBZIRC competition took place during 23–25 February 2020, where 32 teams from around the world competed in 3 challenges. Aerial and ground autonomous robots had to be used to perform tasks in the pre-defined arena. Challenges of MBZIRC 2020 has motivations based on real problems with the following assignments:

- Challenge 1 is motivated by UAV safety, where a team of UAVs is supposed to interact autonomously with objects (intruders) in the arena. This challenge is divided into two subtasks, that are performed at the same time slot. UAVs had to recognize and pop balloons randomly placed in the arena in the first part. A ball carried by foreign UAV is required to be caught in the rest of the task.
- Challenge 2 is motivated by autonomous building construction using robots. A team of UAVs with one UGV (unmanned ground vehicle) is supposed to localize, grasp, and transfer a set of bricks to construct a pre-defined wall structure. Different lengths of bricks are used in this task. This thesis was motivated by the problem of carrying the largest brick available.
- Challenge 3 is motivated by autonomous urban firefighting. A team of robots is required to localize and extinguish simulated fires inside a building and in its surroundings. This task is made more difficult by smoke that is present in some rooms in the building.
- Any team can be qualified for the Grand Challenge by the results of previous challenges. The Grand Challenge combines all challenges simultaneously.

The team led by the MRS group competed in all disciplines of MBZIRC 2020 in an attempt to defend success at MBZIRC 2017. We managed to win the second challenge and we ended up second in the first challenge. Thanks to these results, we qualified into the Grand Challenge, which we managed to win as well.

## 1.2 State of the art

Different approaches can be found in the current state-of-the-art. Most research teams take their route in the sense of control strategy, grasping mechanism, and number of UAVs. In this section, we introduce several solutions to the related task.

Two multicopters with a robotic arm are used in the paper [13] to cooperatively manipulate with a stick-shaped object. Control of a coupled system is decentralized, thus both UAVs are controlled individually. A robust, multilayer motion controller is used for this purpose. It takes position references from a motion planner. This planner primarily produces such velocity commands that regulate internal forces between object and aerial

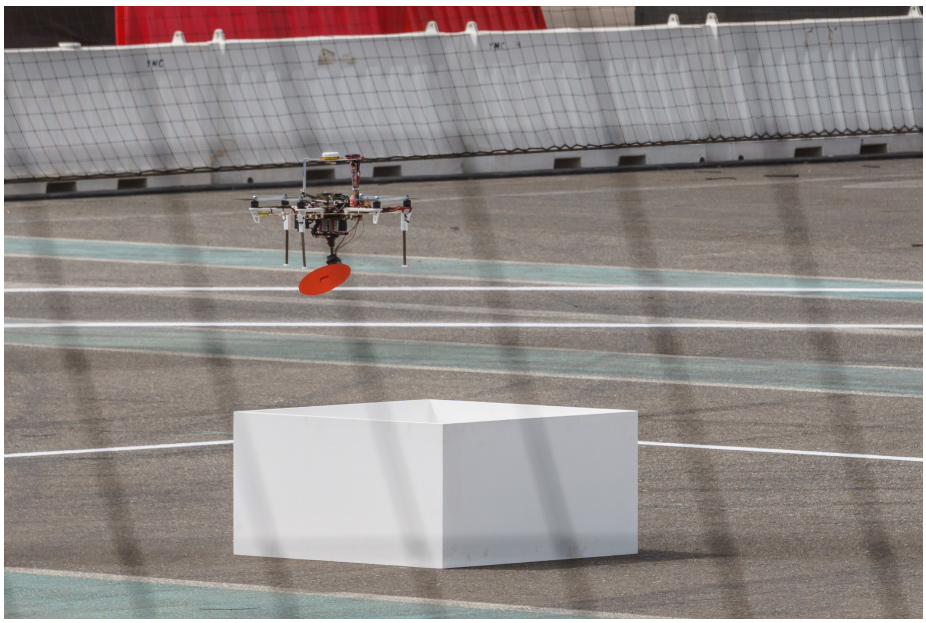


Figure 3: A UAV carrying a metal plate during MBZIRC 2017 Challenge 3. The grasped object had to be placed in the area defined by white boundaries.

manipulator. Further, the planner keeps a multirotor and carried objects in a sufficient distance to avoid collision between them. Last but not least, the planner produces commands to manipulate the carried object to its reference position. The results of the experiment show success in the design of the control and planner system. However, the paper shows verification of the proposed method only in one motion (tilt of the carried rod). Thus we can not assume the behavior of the system during more complex motions. Further, experiments are dependent on the motion capture system, which provides information about the position of the cooperative manipulator in space. Thus, this solution is suitable primarily for indoor experimental environments.

A similar approach was introduced in [15]. A pair of UAVs with 2-DOF (degrees of freedom) robotic arms is manipulating with a payload. An adaptive sliding mode controller is designed based on the closed chain dynamics of the coupled system. The trajectory of the individual helicopter is obtained from RRT* (rapidly-exploring random tree star) planning algorithm.

In paper [34], a set of three aerial vehicles are manipulation a payload attached via towed cables. Control laws and motion plans that allow controlling of payload along the desired trajectory are presented. The proposed strategy provides a feasible trajectory for each multi-copter that brings payload to its goal configuration. However, with this approach, the payload cannot be directly regulated while using cables, since there is no fixed connection between aerial manipulators and payload. Papers [21][7] deals with similar cooperative manipulator configuration.

A completely different approach to the task of cooperative aerial manipulation is introduced in [8]. In this parallel concept, a swarm of UAV is simulating the human hand, where each UAV represents individual fingers. Force sensing and feedback is used to control aerial manipulators.

Proceeding work of the MRS group dealt with cooperative manipulation during preparations for MBZIRC 2017 competition. Used approach [14] combined principles presented above. A pair of helicopters was carrying payload via cables. Each helicopter was controlled by onboard MPC (model predictive controller) [4] while moving along a trajectory generated by the RRT algorithm. However, the trajectory was preplanned for the familiar environment. Further, UAVs were controlled individually with the need of RTK GNSS and long towed cables.

## 1.3 Outline

This thesis is structured as follows. Preliminaries, where necessary terms and tools of this thesis are introduced, are in chapter 2. A description of a custom-built gripper suited for collaborative manipulation as well as for single UAV manipulation is presented in chapter 3. Chapter 4 describes configurations of the gripper that can be realized with aerial manipulation. Discussion about the reasons for the choice of the configuration used in this thesis can be found there. The creation of the model of the gripper is introduced in chapter 5. The model is used in the robotic simulator environment. Customization of this environment for purposes of cooperative manipulation task is described in the same chapter. A proposed control strategy for a coupled system consisting of two aerial manipulators and a payload can be found in chapter 6. Control laws for individual motions are derived there. Verification of the proposed control method is an objective of the chapter 7. This verification is done in the simulated environment. A system that secure proper attach and detach of a payload follows in chapter 8. The description of its subsystem that plays the role of a fail-safe system can be found in the second part of chapter 8. This thesis is concluded in chapter 9, where discussion of the achieved objectives and future work is located.

## 1.4 Contribution

We present a system that can be used in outdoor experiment scenarios, where a group of UAVs is collaboratively manipulating with a solid object. The new part of the system consists of a gripper system, a cooperative control system, and support systems that secure the safe operations of the collaborative manipulator. Our solution was tested in simulations. Experiments have shown that the designed cascade PID control strategy is sufficient for a coupled system control. Further, a real experiment has shown that the gripper system was successfully integrated into the current UAV system. Our contribution

is summarized in the following points:

- We present the specialized gripper suitable for cooperative tasks as well as for single UAV manipulation.
- We designed a coupled system that can be controlled without a knowledge of precise localization information of all presented UAVs.
- We formulated the cascade PID controller for a collaborative motion of two UAVs attached to a solid payload.

## 1.5 Mathematical notation

The mathematical notation used in this thesis is shown in the following table.

| Symbol | Description |
|---|---|
| lowercase letter, e.g., $x$ | a scalar |
| bold lowercase letter, e.g., $\mathbf{x}$ | a column vector |
| $\mathbf{x}^T$ | vector transpose |
| $\dot{\mathbf{x}}, \ddot{\mathbf{x}}$ | first/second derivative by time |
| $\lvert\mathbf{x}\rvert$ | absolute value of each element in a vector |
| $\mathbf{x}^{(W)}$ | $x$ in coordinate system $W$ |
| $\Delta x$ | x small change in variable $x$ |

Table 1: Used mathematical notation

# 2 Preliminaries

In this chapter, we explain several terms and describe the tools used. First, this section describes Robot Operating System [29][23] and Gazebo simulator [26]. The used platform with its sensors is the subject of the second part of this section.

## 2.1 ROS and Gazebo

Software architecture used in connection with UAV platform development in the MRS laboratory is currently implemented in the Robot Operating System (ROS). The software implemented in this thesis follows this approach. ROS is a flexible framework that allows us to design and develop a robot system. It is a middleware system that provides services above the host operating system. ROS is most supported by Unix-based platforms, thus the development of this thesis took place in Ubuntu 18.04. Figure 4 shows a diagram of a communication architecture provided by ROS to running programs called ROS nodes. ROS nodes can communicate through services or topics with each other. Service is based on one-to-one communication between two nodes, while more than one node can publish or subscribe a message to/from a topic. Detailed information about ROS can be found in [29].

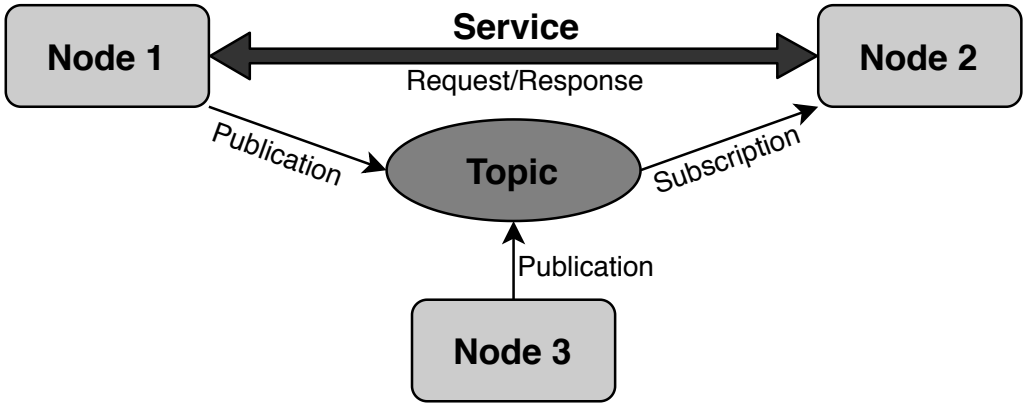



Figure 4: Structure of communication between nodes in ROS. Service, as well as a topic communication channel, is shown.

A realistic Gazebo simulator is another important tool that is commonly used in robotic system development. It allows us to experimentally verify developed systems without the need for a hardware device. Chapter 5 is dedicated to the Gazebo simulator and its customization.

## 2.2 UAV platform

The UAV platform that is proposed for this thesis is a custom-build system. It consists of several modules that serve as sensors, actuators, or computing unit. Figure 5 shows a system diagram of all the onboard components.

Tarot 650 Sport was selected as a frame for the UAV. It is a quadcopter frame with arms made of carbon fiber. The frame is fitted with brushless electromotors, propellers, and ESC (electronic speed control) drivers. Commands to these drivers are sent from the Pixhawk 4 flight controller that stands here in the lowest control layer. A GPS module with a built-in compass is connected to the flight controller. It provides coordinates of a UAV in 3D space and its orientation. An onboard computer (Intel NUC) with Intel processor is

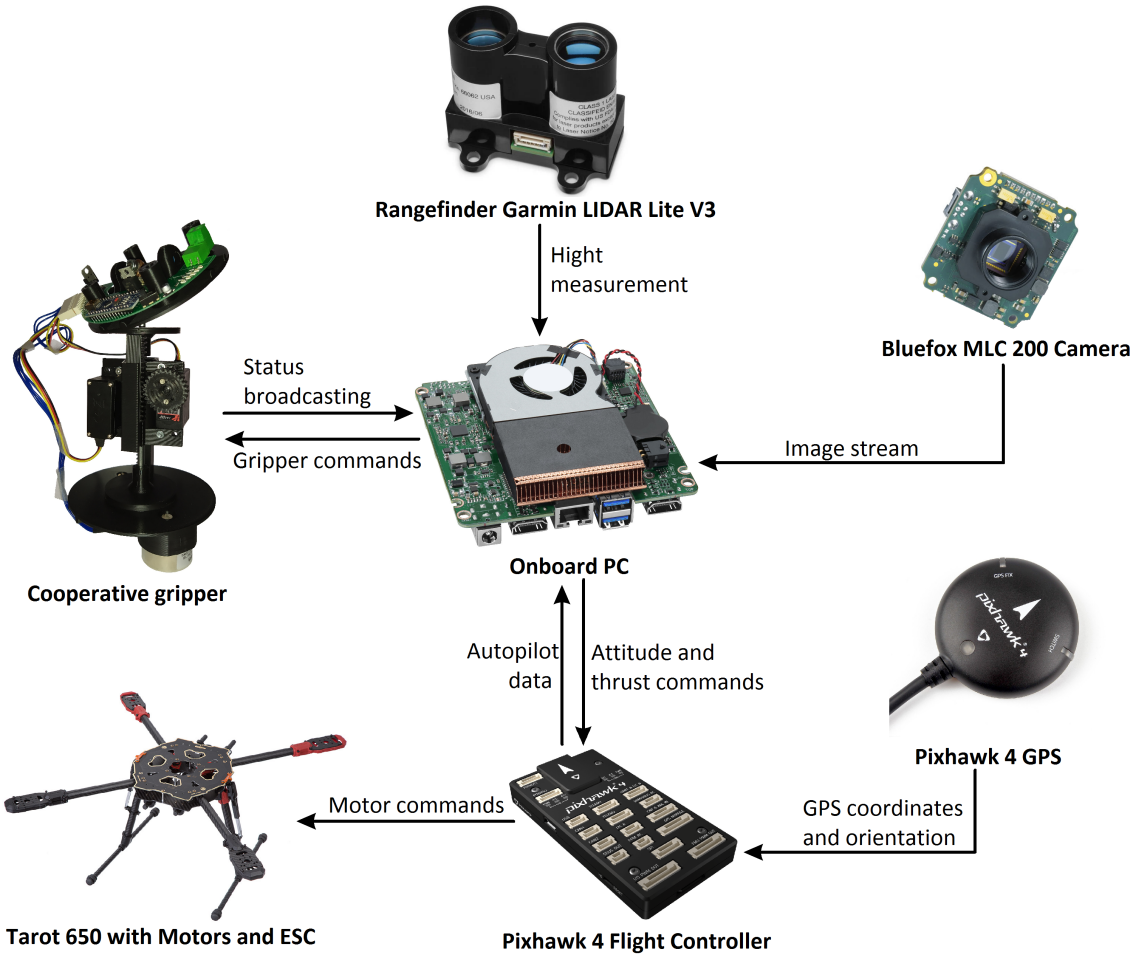



Figure 5: UAV platform used for collaborative manipulation with all necessary sensors. The designed gripper is connected through a serial communication channel to the rest of the system.

the main computing unit. Ubuntu operating system with ROS is running on this PC. It provides enough computational power to solve all necessary processes (feedback control, motion planning, image processing) onboard. Further, onboard PC provides attitude and thrust commands for the Pixhawk controller. Images from Bluefox camera are used with image processing for payload detection. The last equipped sensor is the Garmin rangefinder. It measures the height of the UAV above ground. The designed gripper is connected through a serial communication channel to the rest of the system. Commands and gripper states are sent through the communication channel.

# 3 Gripper design

The following chapter presents results of research of the gripper, that would meet all the requirements listed below as much as possible. Three types of grippers (or grasping/manipulating devices) are commonly used in aerial manipulation tasks. The first of them is a rigid gripper [17][20]. Such a device may have several passive joints or a spring mechanism. Grippers of this type are used primarily for grasping objects by a single UAV.

An active manipulator [12][9] is another type of grasping device. Their design is similar to industrial robots. They consist of several links connected with actuated joints. Such a manipulator operates in a work envelope around the UAV. On the other hand, manipulators are more complex devices than rigid grippers. Further, they create challenges in the control of the aerial manipulator. These challenges are caused by the moving center of gravity of the whole system and due to centrifugal and Euler forces [11] acting on the UAV.

Towed cables (or rods) [35][18] are the third possibility of attaching an object to an aerial vehicle. The simplicity of this approach is one of the few advantages. Positioning of a manipulated object and control of the UAV are challenges of this approach. Problems are related to a pendulum motion of the towed object. The negative effect is caused by the length of the cable, mass of the towed object, the material of cable and shape of the towed object in general. However, towed cables are commonly used with cooperative manipulation, where the pendulum effect is reduced.

The gripper that was developed for this thesis is a hybrid type gripper. The goal of the design was to combine features of grasping devices described above to meet the following objectives:

- It can be used for cooperative aerial manipulation (cooperative mode) as well as for single UAV (single mode).
- Switching between those modes can be done automatically using onboard electronics and mechatronics.
- Possibility of horizontal stabilization of the towed object.
- It can be equipped with an electromagnetic end-effector for ferrous objects grasping.
- It can be equipped with feedback for correct attachment detection.
- It should allow disabling of yaw rotation between UAV and end-effector (typical for spherical joints).
- It will have as few actuators as possible.
- Low weight.

## 3.1 Gripper v1

Several versions of a gripper were made. However, all of them are based on the same principle. Differences between the first and last version will be discussed in the section 3.2. The structure of the gripper is shown in figure 7. The top plate can be attached directly under the battery of the UAV platforms used in the MRS laboratory. At the same time, it serves as housing for a couple of radial bearings. A steel rod is inserted through these bearings on which housing for another a couple of radial bearings is attached. These two couples of bearings have the perpendicular axis of rotation. This arrangement replaces the spherical joint with disabled yaw rotation (the principle of motion is the same as for so-called universal [5] or Cardan joint). Another steel rod is attached to the second bearing couple. This rod represents a towel cable. The same mechanism as described above is located on the other end of the rod. The bottom plate provides mounting places for electromagnets. A plastic body with two linear servomechanisms is situated in the middle of the towed rod. The servomechanism consists of an electrical servomotor, a piston plate sliding on a towed rod, and rack and pinion gear. The gear converts the radial motion of the servomotor's rotor to the linear motion of the piston plate. The group consisting of the upper or bottom plate, the servomotor, gear, and piston plate will be further referred to as the *upper* or *bottom* lock.

Figure 6: Gripper v1 during an experimental flight in May 2019. Single UAV mode, electronics, and communication were tested.

The upper universal joint allows the end-effector of the gripper to move in two degrees of freedom. Its workspace is part of the envelope of a sphere under the UAV. We can limit the range of the motion of the towed rod by the positioning of the *top lock*. The towed rod is not allowed to move when the *top lock* is pressed against the upper plate (the *top lock* is locked in other words). This means that the rod is perpendicular to the top plate and the

upper part of the gripper became rigid. The lower universal joint has the same freedom of movement as the upper one and the *lower lock* has the same features as the *upper lock*. Thus, while it is locked, the bottom plate is perpendicular to the towed rod.

When the *upper lock* is locked and the *lower lock* is unlocked, the gripper acts as a rigid gripper. Thus, it is suitable for grasping some object by a single aerial vehicle. Before placing the brick, the *lower lock* can be locked to stabilize the brick horizontally. When the *upper lock* is unlocked, the gripper can be used for cooperative manipulation. More information about possible configurations of those mechanisms will be described further in chapter 4.

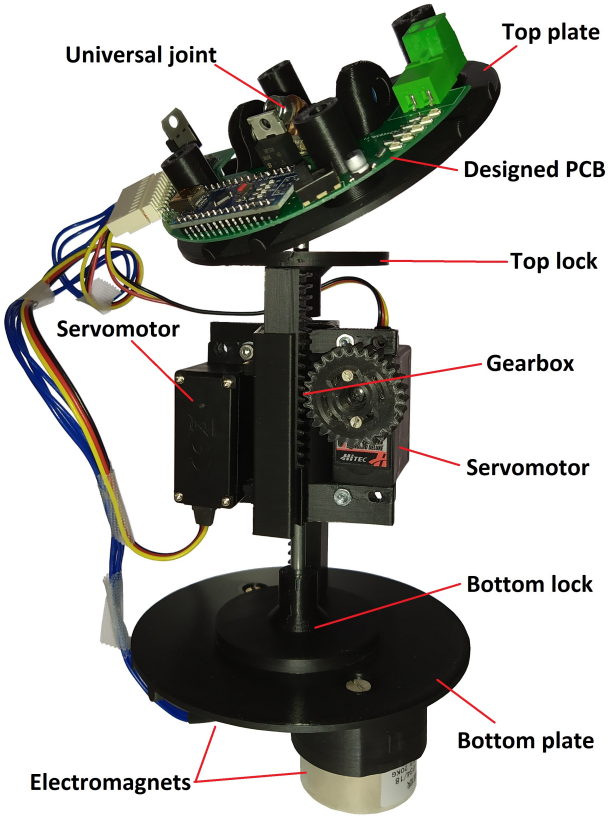



Figure 7: The latest version (v2) of the designed gripper. Improvement in physical properties, electronics and communication have been done.

An experiment was done with the first prototype of the gripper to observe the behavior of the aerial vehicle during flight with the attached prototype (figure 6). The multicopter was remotely operated by a pilot during the flight. A command to lock the upper lock was sent to the gripper after the multicopter took off. The aerial vehicle was then guided above a brick, which was grasped. Different configurations of lock mechanisms were tested after that. During the flight with both locks unlocked, it was unable to control the aerial vehicle properly because of the pendulum effect. This behavior was expected. However, it was hard to control multicopter smoothly even in some other modes. Redesign of the

prototype, where the focus on weight loss and shortening of the gripper, was set as the solution to this problem. On the other hand, we successfully tested the functionality of mechanisms, horizontal stabilization of the brick, and communication between the gripper and the multicopter.

## 3.2 Gripper v2

Differences between the first and second generation of the gripper are minor but crucial. All the steel parts including bearings and towed rod were replaced with smaller ones. Further, all plastic parts were remodeled concerning dimension and weight loss. Weight and length reduction of about 40% was achieved. Further, electronics and communication with multicopter's computer were improved and debugged. This version of the gripper is a pattern for the simulator model, which is described in section 5.

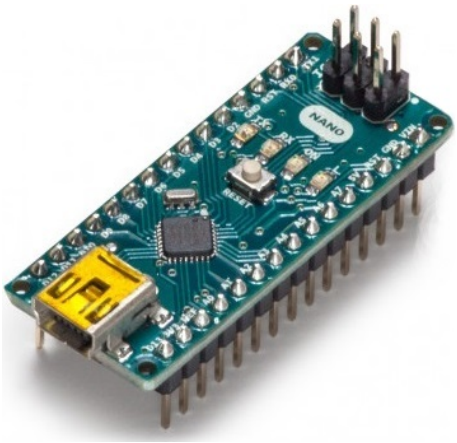
## 3.3 Electronics and communication with UAV computer

Electronics of the gripper is dependent on the electronics of the UAV. It shares the power from the accumulator with other onboard devices. Further, the gripper is controlled with commands sent from the UAVs computer. The main electronic parts are the microcontroller, custom PCB for other electronic part connections, two electromagnets, two servomotors, and a feedback circuit. Electromagnets are placed in the position of the end effector and allow to grasp metal ferromagnetic objects. Detailed information about electronic parts and communication protocol is presented below.

### 3.3.1 Microcontroller and servomotor

The clone of the Arduino nano [2] module was chosen as the control board for the gripper. Arduino is an open-source platform providing a single-board microcontroller with supporting software. IDE (integrated development environment) of the same name allows us to develop code and compile and upload it directly to the microcontroller. Wiring is the programing language used here (based on C++). Arduino Nano is a small size, lightweight board, equipped with an ATmega328 microcontroller. It is a single-chip microcontroller from the megaAVR family designed by Atmel. ATmega328 has 32kB of flash memory. That is enough for the gripper control. Nano board is equipped with a bootloader, so no other device is needed for uploading do code into the microcontroller.

There are two servomotors attached to the gripper. Each of them controls one of two lock mechanisms. Forces acting against the force with origin in the servomotor (and its gearbox) depend on the weight and length of the gripper and the weight of the grasped




(a) Arduino Nano equipped with ATmega328 microcontroller [2].


(b) Hitec HS-430BH servomotor [27].

Figure 8: The Control board and actuator are one of the main electronic parts of the designed gripper.

brick. Thus we were looking for servomotor that would have a relatively good ratio between weight and its rotor torque. High voltage actuator Hitec HS-430BH was chosen. This servomotor has a plastic gearbox, output torque around 5 kg.cm$^{-1}$ (at voltage 7.4 V) and weight 45.5 g. The chosen actuator is a standard analog type without any extern feedback signal. Thus we can command the actuator through an analog data link, however, there is no possibility to monitor the status of the actuator without adding a sensor.

### 3.3.2 PCB design

A PCB (printed circuit board) was designed for gripper's electronics. It was built in a shape, that allows us to mount it directly on the top plate of the gripper. It contains presented microcontroller, 8 V and 5 V voltage regulators, MOSFET transistor with N-channel, and states signalizing LEDs. Several connectors and switches were designed into the board for better prototyping.

### 3.3.3 Feedback

A pair of helicopters manipulate with a solid brick with a ferrous plate on the top of it in this thesis. It is necessary to know if the brick is properly attached to both grippers. The Hall effect sensor Allegro A1324 was used for this purpose. A Hall effect sensor is a sensor that measures the magnitude of a magnetic field. Allegro A1324 is a linear sensor, thus

its output voltage is linearly dependent on the magnetic field strength. The used sensor embedded into the gripper's electromagnet is shown in figure 10.

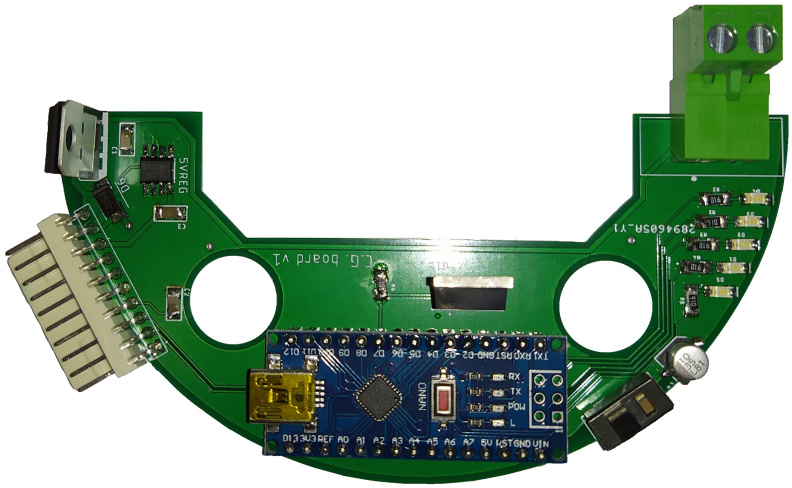



Figure 9: Designed PCB with a control board and other necessary electronic equipment.

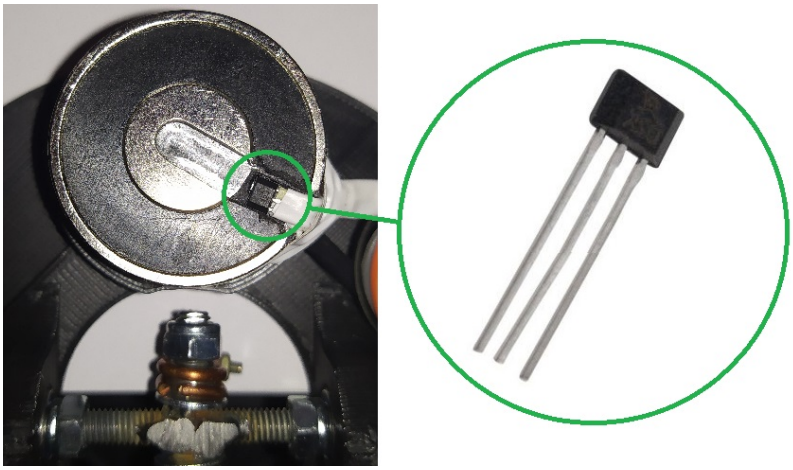


Figure 10: Hall effect sensor Allegro A1324 embedded into electromagnet on the bottom gripper's plate [1].

Each gripper has two Hall effect sensors (one for each electromagnet). Their outputs are brought to analog input pins of the control board to a 10-bits analog-to-digital converter (ADC). We can obtain raw data in the range between 0 and 1023. Raw data are filtered by a moving average. Further, a threshold that determines the boundary between states (grasped or released) is set. Threshold is unique and depends on electromagnet strength and thickness of the ferrous plate on the brick. The sensor's output and the magnetic field strength is temperature-dependent. This is caused by the heating up of an electromagnet, that affects its electrical impedance. The threshold has to be calibrated before each flight as

well as modified during the flight. A true/false information is obtained from the processed data which indicates if the electromagnet is attached properly or not. Logical conjunction is used with values from both sensors to determine if the whole gripper has the right grip. The described process is shown in figure 11.

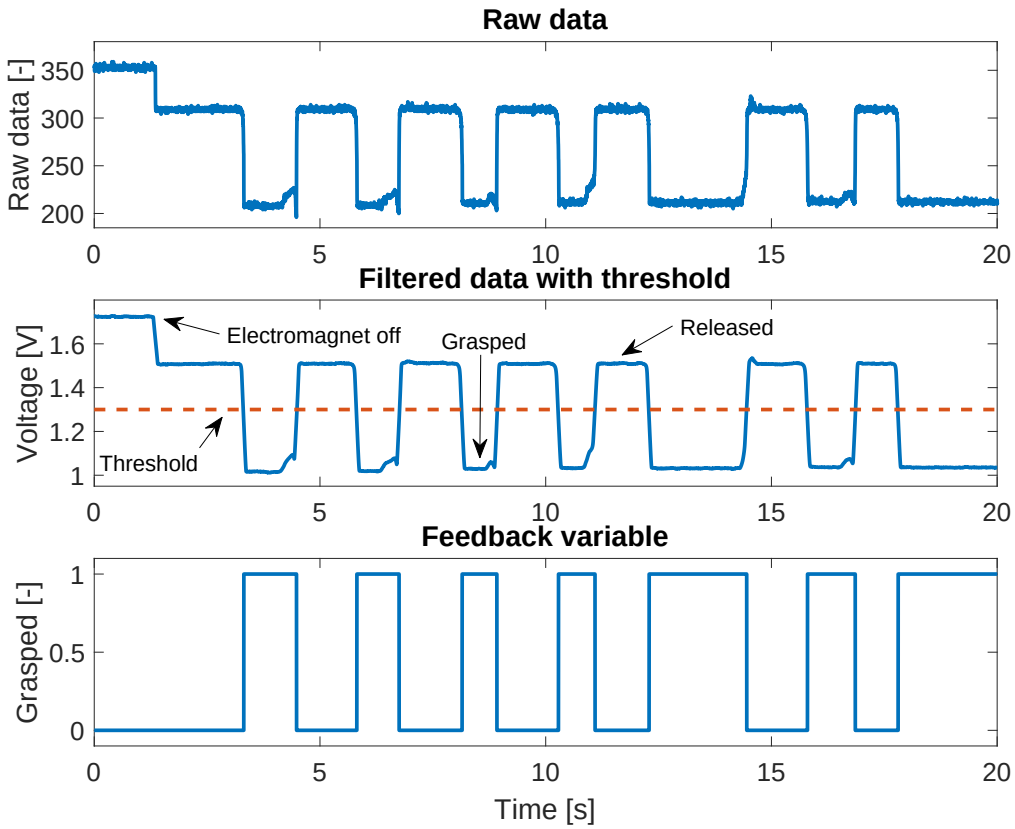



Figure 11: Recorded and processed data from Allegro A1324 sensor.

The resulting information about the grasping feedback is sent to the onboard computer of the helicopter. Data are then used in a cooperative guidance system to avoid situations where the brick could be badly grasped by one of the helicopters or even by both of them. It is also used to abort a collaborative mission and switch to a failsafe system in the case of loss of grip during flight.

### 3.3.4 Communication protocol

The main nodes of this thesis are running on the computer onboard on the multicopter. The gripper is commanded from one of the running nodes. The communication between the Nano board and the onboard computer is implemented through a serial link (via USB). Three message types can be sent through this channel:

| Msg number | Msg type | States |
|---|---|---|
| 1 | Top lock setting | Interval [0,1] |
| 2 | Bottom lock setting | Interval [0,1] |
| 3 | End-effector setting | Grasp/release |

Table 2: Message types

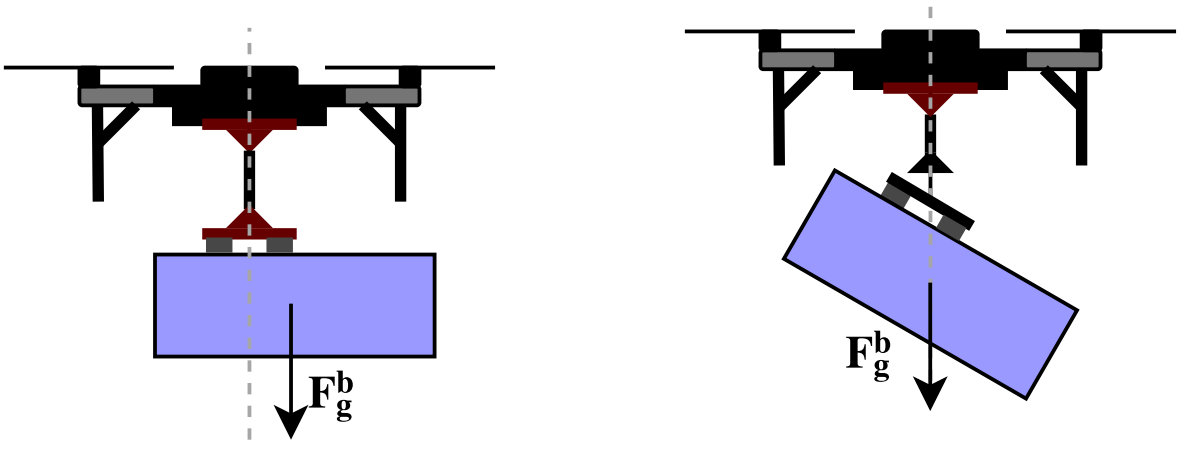
The first two message types set the lock rate in the range of interval [0,1] where 0 means unlocked lock and 1 means locked lock. The last message type is used to control the end-effector (electromagnets in this case) with two states (grasp/release). The communication channel is bi-directional. However, commands are sent just in the direction from the computer to the Nano board. The other direction is used for broadcasting states of the gripper. Data are sent in network packets that correspond to the UART communication protocol rules.

# 4 Flight configurations with the gripper

The design and mechanical properties of the gripper allows us to use this device in different configurations. The reason for this requirement is the need for switching between single UAV manipulation and cooperative manipulation, which is motivated by the MBZIRC 2020 challenge. Further, the cooperative manipulation is still at the beginning of research in the MRS laboratory. Thus, the gripper was designed so, that it will be an as universal device as possible. It will allow us to test the gripper in different configurations with different control approaches. The results of those tests can be then compared and a solution for further projects can be chosen and debugged. There are four configurations, that can be set for cooperative manipulation. These configurations depend on the lock setting which will be chosen. This section presents all possible configurations with a discussion about their possibility of use. It is important to note, that the solution for control of a cooperative manipulator will be implemented only for one of the presented configurations. We will choose the configuration, which will have the largest potential to succeed in real operations concerning technological and physical possibilities in the MRS laboratory.

## 4.1 Single UAV mode

The designed gripper attached to the single aerial vehicle can be used instead of a rigid-type gripper if the top lock is locked. Then the towed rod became rigid concerning the body of the UAV. The bottom part stays unlocked during the flight and the grasping of brick. This configuration suppresses the pendulum effect, that would be presented with the use of towed cables, that are commonly used for collaborative manipulation tasks. Further,




(a) The brick is horizontally stabilized.

(b) The bottom is unlocked to suppress undesirable momentum.

Figure 12: Designed gripper used for single UAV manipulation.

free movement of the bottom universal joint suppresses external momentum acting on the UAV. The further the gripper is from the center of gravity of brick, the greater the momentum is. Thus it makes the control of the vehicle more dependent on the weight of the grasped brick.

The bottom part of the gripper become rigid, before the final guidance to the brick placement position. It horizontally levels the brick for more precise placement. This solution could be used for grasping of smaller bricks in the wall challenge in the MBZIRC 2020 competition. The presented principle of use is shown in figure 12. The configuration with the bottom lock unlocked for free manipulating with a brick is in the right figure. Horizontally stabilized brick is in the left figure.

## 4.2 Cooperative mode

The collaborative manipulation will be tested with the cooperation of two UAVs. Each of them will be equipped with the designed gripper. Different approaches to control such a pair of UAVs can be found. One possible approach is to set one of the UAVs as a master vehicle and the other one as a slave. Then the slave vehicle is moved relatively with the master vehicle. Some visual-based algorithms to follow the master are commonly used for this purpose [36][37]. Further, we can control both helicopters separately, with the presence of robust controllers and precise position data. Modeling of the coupled system is another approach. A controller designed for the whole system is working with a model of the cooperative manipulator. Based on the presented principles, this section provides a comparison and discussion of possible configurations. There are following considered configurations of the coupled system considering the setting of locks:

- All two pairs of locks are locked.
- Upper locks of both UAVs are locked, the rest is unlocked.
- All two pairs of locks are unlocked.
- Lower locks of both UAVs are locked, the rest is unlocked.

The coupled system would have similar behavior as an aerial vehicle with just two propellers, in the first case, when the body is rigid (figure 13). This can be a typical adept for model-based control since there are a lot of similarities with control of another UAV. A pattern of such a model would be the same as the model of some bicopter. Then we could use the same control approaches as are used with control of a single multicopter. No degree of freedom of single UAV concerning the brick can be found in this setup. However, the manipulation with this system would be quite clumsy, especially during roll

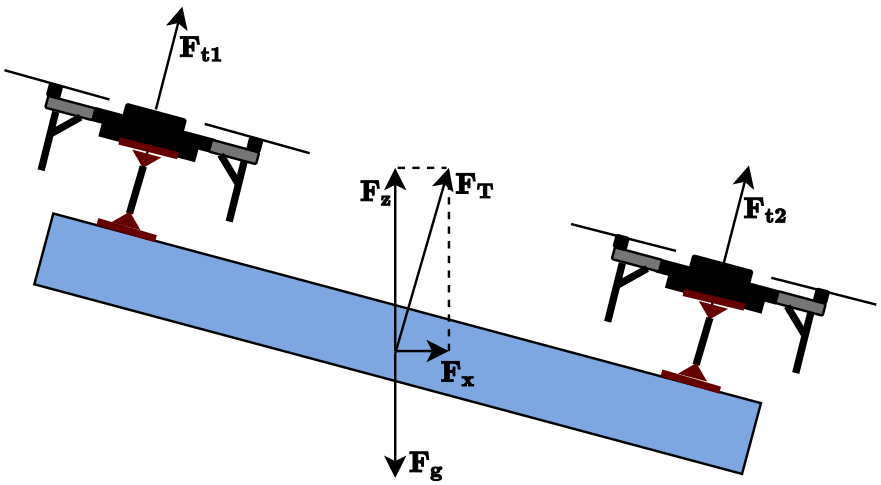



Figure 13: Longitudinal motion for rigid configuration.

tilting. Further, heading if the system is not directly controllable since there is no origin of momentum acting in the vertical axis.

The second configuration is different in that way that both bottom locks are unlocked (figure 14). The pair of aerial vehicles would move on a surface of the spheres with centers in the bottom universal joints and with radius given by gripper length. Both aerial vehicles have then 2 degrees of freedom concerning the grasped object. The positive thing about this setup is that the pendulum effect of the brick is fully disabled as well as in the previous case. On the other hand, it would be very challenging to find out such a control strategy allowing to move the UAV on such limited space (surface of the sphere) fluently. No possibility of the direct control of typical quadcopters in the horizontal plane is the reason for this challenge. A quadcopter needs to be tilted around its center of gravity to do such motion.

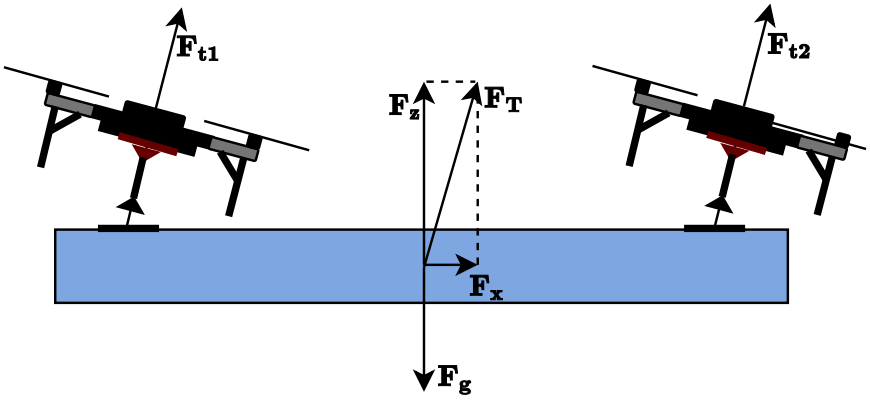



Figure 14: Longitudinal cooperative motion with both top locks locked.

The third configuration, when all locks are unlocked (figure 15), clearly simulates the use of towed cables. A couple of UAVs have then the greatest freedom of movement concerning the towed brick. The cooperative manipulation with towed cables is one of the most researched areas of this type of manipulation since it is close to swarming UAV systems. Experiments were done with the pair of UAV carrying a single object in the MRS group in 2016 [14]. Both helicopters were following a predefined trajectory. There were still quite high requirements on the localization of used helicopters, despite discussed freedom of movement. Further, the pendulum effect is included in the system movement since only two helicopters are used. Despite these requirements, we did not evaluate this configuration as a suitable candidate for control design in this thesis.

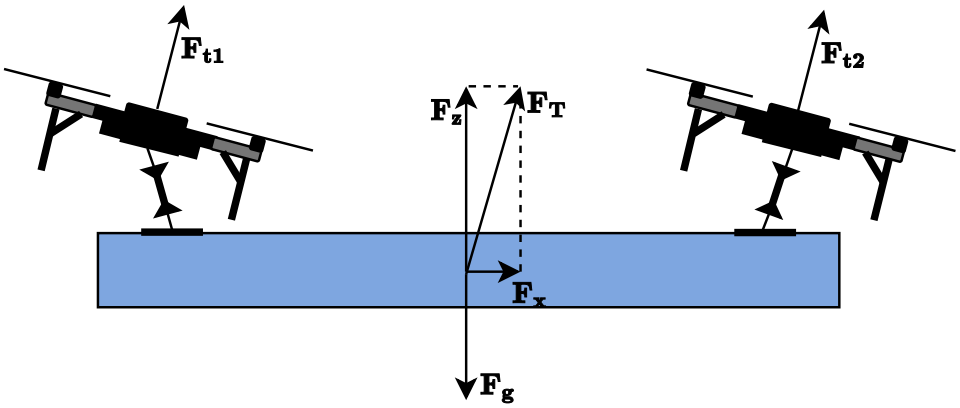



Figure 15: Collaborative manipulation with all free locks. Such configuration behaves similarly to towed cables.

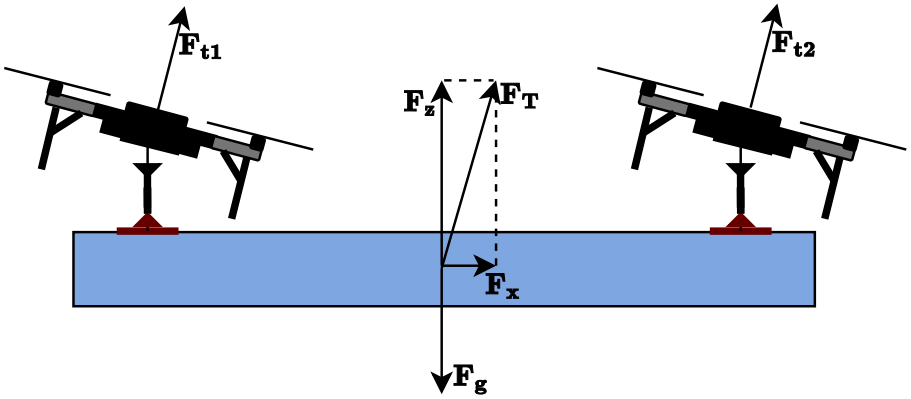



Figure 16: Longitudinal motion with a rigid bottom.

Only the bottom locks are locked in the case of the last approach (figure 16). The bottom part of the coupled system is rigid, thus the pendulum effect is eliminated in the longitudinal direction. The movement of individual UAVs is limited more than in the

previous case. Specifically, 2 degrees of freedom of single UAV are left (pitch, roll). This configuration can be modelled as a bicopter with tilting propellers. Each UAV is then taken as an actuator in this parallel concept.

The last configuration was chosen for control design in this thesis. The reason follows from the discussion above. The problem of precise position knowledge can be solved by position estimation thanks to the rigid bottom of this system. More details will be presented in the chapter 6.

# 5 Simulation

A robotic simulator is a virtual environment where robotic devices can be developed. It saves time during prototyping of new devices and algorithms. Further, it allows us to test our prototypes without the risk of damage. We use the Gazebo simulator in this thesis. It is a 3D open-source simulator that is fully integrated into ROS (Robot Operating System). The main programming language of the simulator is C++ in which all plugins are provided, however, other software written in Python can interact with it through ROS.

Several steps have to be done to create our simulation with a multicopter attached to the designed gripper. It is necessary to create a URDF (Universal Robot Description Format) [25][24] file, where the kinematics and dynamics of the gripper are defined. Then, a connection between Gazebo and ROS has to be implemented. This is a crucial part of making this simulation to be able to control gripper's virtual actuators. Several supporting files need to be created for this purpose. More information about the file structure follows in the subsections below. Further, the gripper will be connected to the Gazebo model of the quadcopter. A simulation environment, containing a pair of aerial manipulators and other necessary objects, is the subject of the conclusion of this chapter. All these steps are important to verify the designed control method before it could be considered for real experiments.

## 5.1 URDF

The URDF (Universal Robotic Description Format) file of the designed gripper was created, as the first step in the simulation design. URDF is a standardized XML file format supported by Gazebo. The kinematics and dynamics of the robot are specified in this file. Specifications of the robot are described through tag elements with several parameters. Link tags and joint tags are two inherent tag types in the URDF file structure. The first of them, the link tag, determines inertial, collision and visual properties of the object's rigid link. Further, the inertial description is done by setting the mass of the link and its matrix of inertia. Both, the collision and visual description can be determined either by basic predefined geometric structures (cylinders, spheres, boxes) through URDF geometry syntax or by referring to MESH file (3D file format), that contains modeled link. The collision element in the link tag describes a shape that will collide with other objects in the simulated environment. The visual element describes a shape that will be shown in the graphical user interface of the simulator. A relationship between two neighboring links in the kinematic chain of the designed object is defined by a joint tag. Type of the joint, parent and child link, an axis of rotation, origin, and limit are the main parameters of this tag. A URDF file with the structure described above can be inserted directly into the simulator. Several other steps need to be done in our URDF file to make the model as realistic as possible.

```
<joint name="top_lock_joint" type="prismatic">
    <parent link="base_rod"/>
    <child link="top_lock_link"/>
    <origin xyz="0.0 0.0 0.875" rpy="0 0 0"/>
    <axis xyz="0 0 1"/>
    <limit lower="-0.02" upper="0.02" effort="1.0" velocity="0.2"/>
</joint>
```

Listing 1: Implemented prismatic joint with meaning of top lock. Its limits correspond to real values.

URDF model of the gripper consists of link tags, which are mostly connected through a fixed type joint. Two pairs of revolute joints are used on the top and on the bottom of the gripper, where they simulate the function of bearings in the HW gripper. Two prismatic joints are used to demonstrate the function of servomotors with gearboxes presented with HW prototypes. An example of such a joint is in the listing 1. Further, we need to define collision properties between those locks and upper and lower plates for the proper function of the designed mechanism. Collision description through link tags describes only the space that can collide with other collidable objects in the simulated environment, as it was mentioned above. Thus, we need to define self-collision between specific links of the same object. This can be done by so-called *gazebo reference tags* (an example is presented in the listing 3). Transmission tags are other necessary elements in the gripper model. They are assigned to joints which are controlled by a virtual actuator. In general, it determines the relation between a joint and a virtual actuator. They are assigned to both defined prismatic joints in our case.

```
<gazebo>
    <plugin name="gazebo_ros_control" filename="
   libgazebo_ros_control.so">
        <robotNamespace>/2dof_model</robotNamespace>
    </plugin>
</gazebo>
```

Listing 2: Implementation of the Gazebo ROS control plugin. It is necessary for the control of designed prismatic joints from the ROS interface.

Whereas we have to command the gripper from ROS, we need to define the interaction between the ROS and Gazebo interface. For that reason, the last important tag in the URDF file of the gripper is the *gazebo* tag, where the Gazebo plugin for control joints from ROS is defined. It parses the transmission tags and it loads controller manager. The controller manager manages assigns controller to individual joints. Implementation of such tag is shown in the listing 2. Other practical examples of the URDF structure described in this section can be found in [11].

```
<gazebo reference="top_lock">
  <self_collide>1</self_collide>
</gazebo>
```

Listing 3: Description of self-collision for the upper lock. It allows the physical contact between the top plate and top lock.

## 5.2 Supporting files

Another two files need to be created to command the gripper model by publishing ROS messages. The first of them is a configuration file, which is located in the same package as the URDF file. In this file, it is possible to configure the controller used with the gripper model. The general type of the controller and its publish rate is set for the whole model here (the joint state controller is used here). We can define the specific type of the controller for each joint individually (however the joint position controller is set here for both prismatic joints). Finally, we can configure PID gains used by the joint position controller.

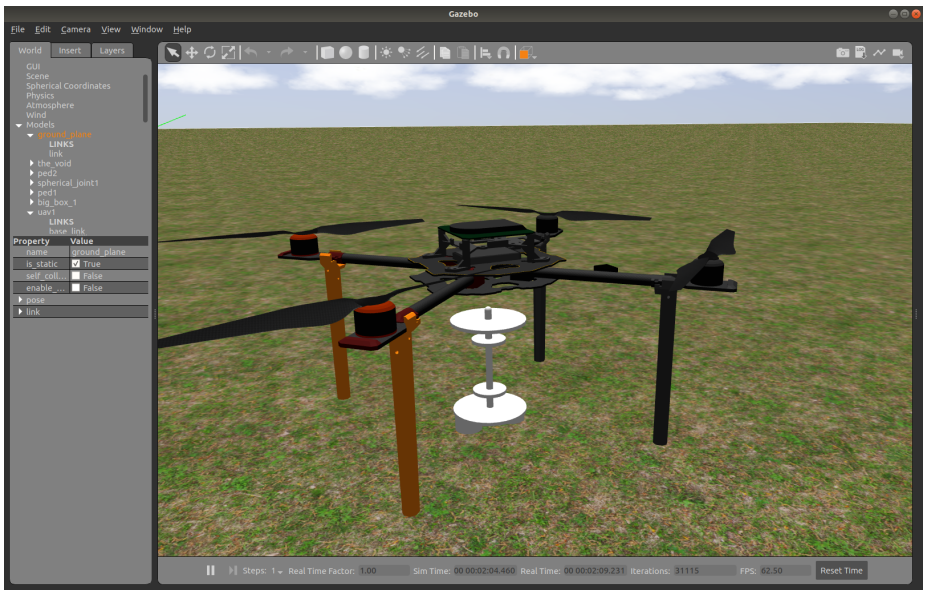



Figure 17: Designed gripper attached to spawned Tarot 650 model in the realistic Gazebo simulator.

The second necessary file is the Launch file where the reference to the URDF file is created. Further, it loads the configuration file which was described above. In the next step, the Launch file loads controller manager and spawns all defined controllers. It is possible to command the model from ROS at this point. To have a better overview of the states, the robot state publisher is loaded through the Launch file. We can observe states of prismatic joints from ROS then.

## 5.3 Attaching the gripper to the UAV

The structure of the quadcopter Gazebo model which is currently used during UAV simulations (Tarot 650 model) is located in several Xacro (XML maco) files. Both Xacro and URDF files use the same syntax for model description. Additional features of the Xacro format are macros and support of constants and other math operations. For that reason, it is possible to insert the content of the created URDF file into the newly created macro in the Xacro file, where the main body of the multicopter is defined. The connection between the gripper and the multicopter is done by an extra joint of fixed type. The final configuration is shown in figure 17, where a single aerial manipulator is captured during flight.

## 5.4 Colaborative manipulation in simulation

Two aerial manipulators (grippers attached to helicopters) need to be created for the collaborative manipulation task. Two different namespaces have to be used for gripper controllers to be able to control them separately. Another Launch file was created, where two different namespaces for robot descriptions are set (see listing 4). Two different controllers with two different configuration files can be launched then to control both aerial manipulators independently.

```
<launch>
    <param ns="gripper1" name="robot_description" command="cat '$(
   find heli)/urdf/gripper1.urdf'" />
    <param ns="gripper2" name="robot_description" command="cat '$(
   find heli)/urdf/gripper2.urdf'" />
</launch>
```

Listing 4: Description of two namespaces used for separate control of two aerial manipulators.

Another URDF was created to define a brick, that would be manipulated with. This brick is defined according to the largest brick used in the MBZIRC challenge. There is no built-in feature that makes a connection between two objects in the Gazebo simulator. Another Gazebo plugin is used to dynamically make a connection between the gripper and the object. This plugin dynamically (during a running simulation) creates a virtual fixed joint between two chosen links. This feature is allowed by service through the ROS interface with feedback. The feedback is used in grasping a guidance system for experiments in a simulated environment. We suppose that brick position and orientation are known since brick detection is not a task of this thesis. Thus, both quadcopters are moved to a position above the brick, where the automatic grasping guidance system is turned on. If the brick is

grasped properly by both of the UAVs, the system will take off the brick from the ground. After that, the current control system is switched to collaborative control for the coupled system. The hovering coupled system controlled by the designed control method is in figure 18.

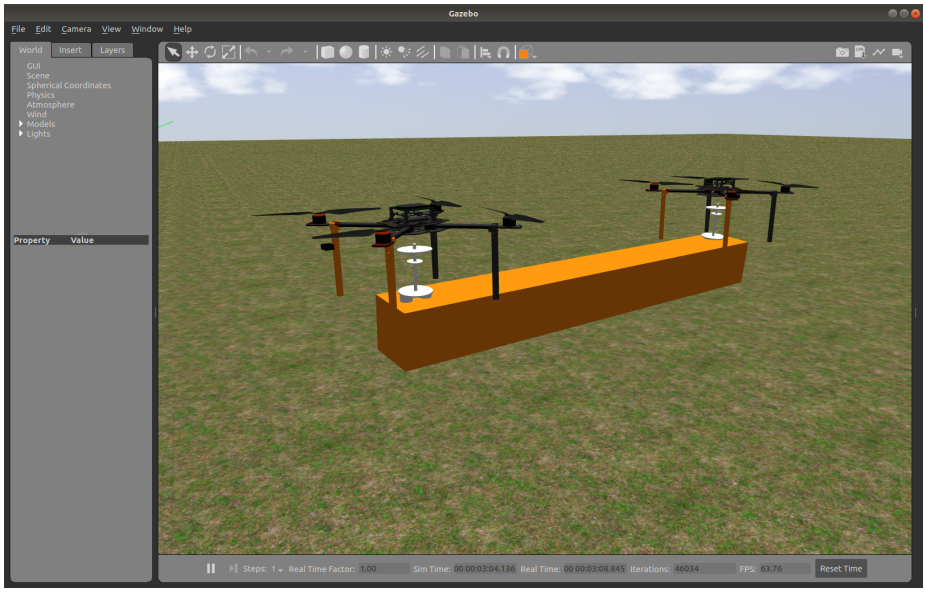



Figure 18: The pair of helicopters during cooperative manipulation in the simulated environment.

# 6 Control of the coupled system

Design and implementation of the control strategy is a main part of the cooperative manipulation task. Our goal is to make a controller that will bring the system from its current position to a reference position. We were looking for control approaches for collaborative manipulation, where a pair of helicopters were used, in the state of the art. Articles, where the coupled system consists of rigid bodies were taken into account since the configuration with both bottom locks locked was chosen in our solution. A velocity command planning is proposed in [13]. They found the desired velocity for individual multicopters by solving the inverse kinematics control problem. The velocity reference is taken as a setpoint for a robust two-layer model-based controller of multicopter. The outer control loop is designed to drive pitch, roll, yaw-rate and vertical velocity to their reference values. This loop is based on the multirotor model described by dynamic equations. The inner loop stays here as a compensator of model inaccuracies.

The rapidly exploring random tree star algorithm (RRT*) as a generator of a path for each UAV was introduced in [16]. The controllers are model-based as in the previous case. Another similar approach was chosen in [6]. A receding horizon nonlinear MPC (model predictive controller) was used there for leading aerial manipulators along the desired path.

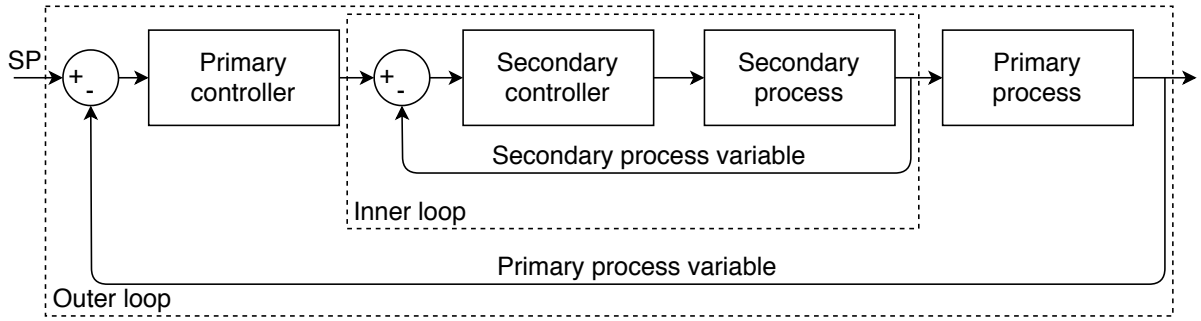



Figure 19: Cascade control structure.

All the approaches, which were introduced, focus on control strategies, where each aerial manipulator is controlled separately. This requires a complex and robust controller, that can control position (or velocity) of individual multicopters precisely. However, our goal is to design a controller without the need for precise position control of individual UAV. Such a controller could be used in a non-laboratory environment as MBZIRC 2020 competition. Thus we focused on the design of a controller for the whole coupled system. We chose a multilayer PID (proportional-integral-derivative) [30][38] control strategy as a suitable candidate for this purpose. We can obtain better dynamic performance thanks to the cascade structure. A typical use of cascade control is shown in figure 19. The primary process variable is controlled in the outer loop. Its output is brought to the input of the second controller as a setpoint. It controls the secondary process variable in the inner loop. In general, the inner loop controls a process with faster dynamics than the outer loop.

The mechanical configuration of the system in this thesis consists of a rigid body and two multicopter attached with universal joints. Thus we can imagine our system as a bicopter with tilting actuators. The system is described by the following states:

- $x_c, y_c, z_c$ ... position in the 3D space
- $\theta_c$ ... pitch angle of the system
- $\psi_c$ ... heading angle of the system

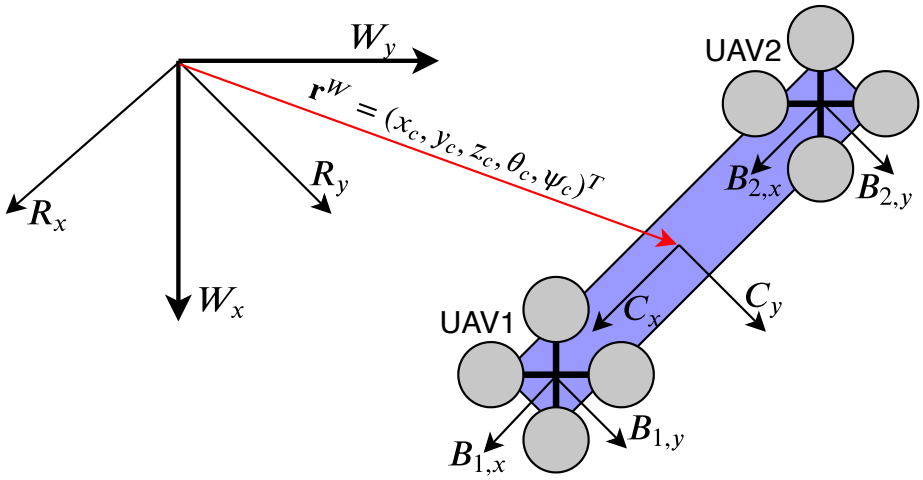



Figure 20: Upper view of presented coordinate systems for collaborative control. Z-axis of individual coordinate systems aims upward from the plane.

Another important variable is the distance between the pair of helicopters during collaborative motion. The constant depends on the length of the grasped brick and the precision of control the UAVs during grasping. Determining this constant is not the task of this thesis and it will be taken from image processing used during the guidance. However, the distance will be used in the further description of the control strategy in this section with the following notation:

- $d_c$ ... distance between geometric centers of the helicopters.

## 6.1 Coordinate systems

There are five coordinate systems used (figure 20). The first one is a global coordinate system, which is fixed to the world. Another coordinate system is a rotated coordinate system R. Relation between W and R is set by rotation around $W_z$ by $\psi_C$. The third one is

the coordinate system that corresponds to the coupled system. It is obtained by translation R into the geometric center of the collaborative system. Last two coordinate systems, $B_1$ and $B_2$, which have meaning of body frames of individual UAVs.

## 6.2 Control structure

The designed control structure for the pair of UAVs carrying a solid object (a coupled system) is shown in figure 21. This control structure is based on multilayer PID control strategy. Desired tilts and thrusts for individual multicopter are brought as the input for the coupled system. States and process variables of this system are the position, pitch angle, heading, velocity in the xy-plane and heading-rate. The velocity controller is in the first

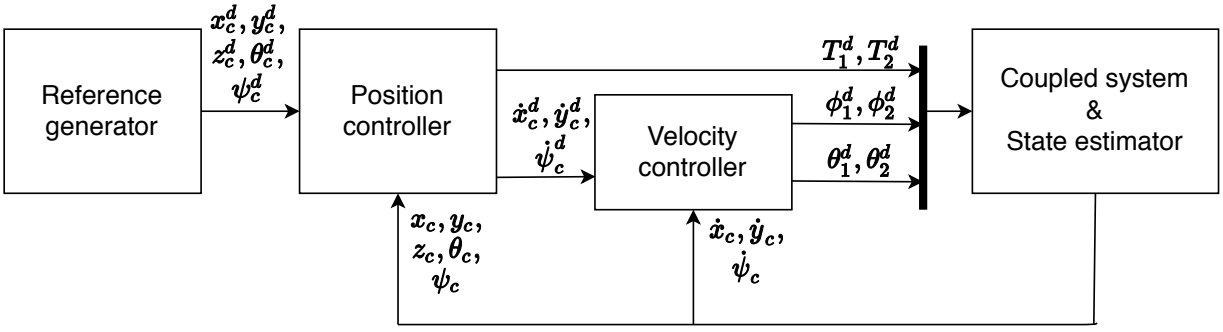



Figure 21: Used control structure with two layer control strategy.

control layer with desired xy-velocity and heading-rate on the input. The position controller is located above the inner layer. It takes the desired states of the coupled system as an input variable. The reference generator, that provides reference values, takes commands from an operator. Commands are supposed to be in the world coordinate system. The reference generator converts them into a rotated coordinate system using the following equation:

$$\begin{pmatrix} x^{(R)} \\ y^{(R)} \end{pmatrix} = \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} x^{(W)} \\ y^{(W)} \end{pmatrix}, \tag{1}$$

where $\begin{pmatrix} x^{(R)} \\ y^{(R)} \end{pmatrix}$ is a position in rotated coordinate system, $\begin{pmatrix} x^{(W)} \\ y^{(W)} \end{pmatrix}$ is a position in the global coordinate system and $\psi$ is heading angle of the system. This equation is also used further for coupled system states conversion. The control structure, which was described above, will be presented with more details in the following subsections.

### 6.2.1 Coupled system description

The coupled system consists of two UAV subsystems. Each of them contains an inner attitude controller and the UAV plant. It is necessary to know the position of individual

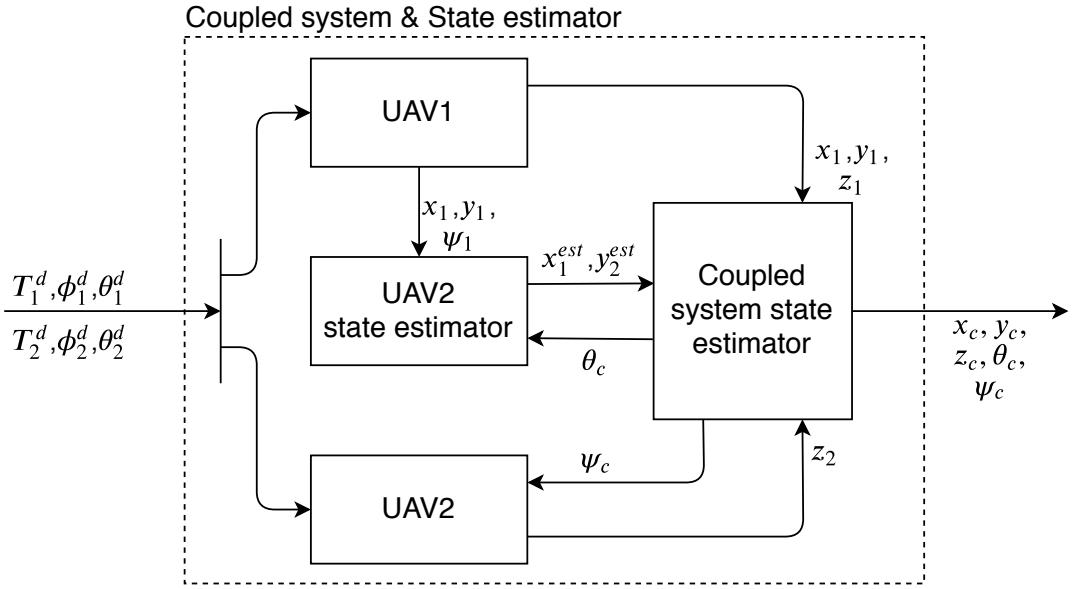



Figure 22: The coupled system structure.

UAV platforms to determine and control the states of the coupled system. However, RTK (Real-time kinematics) GNSS used on our platforms is not precise enough for this purpose. Thus, we utilize the mechanical properties of the designed gripper to estimate the position. RTK data are used only for obtaining the position of the first multicopter. Position of the second UAV is estimated through the equations:

$$x_2^{est} = x_1 - d\cos\psi_1\cos\theta_c, \tag{2}$$

$$y_2^{est} = y_1 - d\sin\psi_1\cos\theta_c. \tag{3}$$

Other values such as height and heading angle are taken from height rangefinder and compass. We define the position of the coupled system $(x_c, y_c, z_c)^T$ in the geometric center between the presented UAV subsystems. Other states are defined as follows:

$$\theta_c = \frac{\arcsin(z_2 - z_1)}{d}, \tag{4}$$

$$\psi_c = \operatorname{atan2}\left(\frac{x_2 - x_1}{y_2 - y_1}\right). \tag{5}$$

The presented system is shown in figure 22. The control input of this system is taken from the position and velocity controller. The desired heading angle for individual multicopter is set the same as the heading angle of the whole system.

### 6.2.2 Position control layer

A PID controller is used for position control of the coupled system (figure 23). It takes a desired position from a reference generator and it compares them with actual states of the system. Output variables, that correspond to the motion in the vertical plane (height

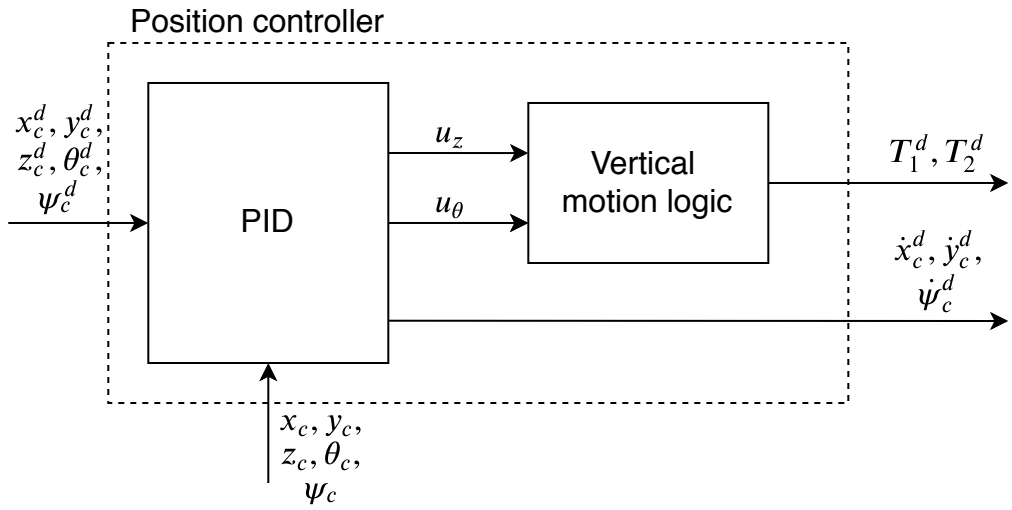



Figure 23: Structure of the position controller used in the highest control layer. The *vertical motion logic* block refers to the equation (14) and (15).

control and tilting) are processed and converted into thrust for individual multicopers. The remaining output variables stay here as a reference for the velocity controller in the lower control layer. They have the meaning of desired velocities in the horizontal plane (xy-velocity and heading rate). Clamping (conditional integration) is used here to avoid anti-windup. It stops integration if a velocity limit of the system is reached.

### 6.2.3 Velocity control layer

The velocity controller takes the desired velocity in the horizontal plane from the position controller as a reference. PD controller is used here. The derivative part makes the coupled system more stable during the motion. Lateral motion of the coupled system is related to the y-axis motion and heading rotation. Output variables, that correspond to these motions are converted to desired roll angles $\phi_1^d$ and $\phi_2^d$. The output of the x-axis velocity controller has the direct meaning of the desired pitch angles for both UAV plant. The structure of the velocity controller is shown in the figure 24.

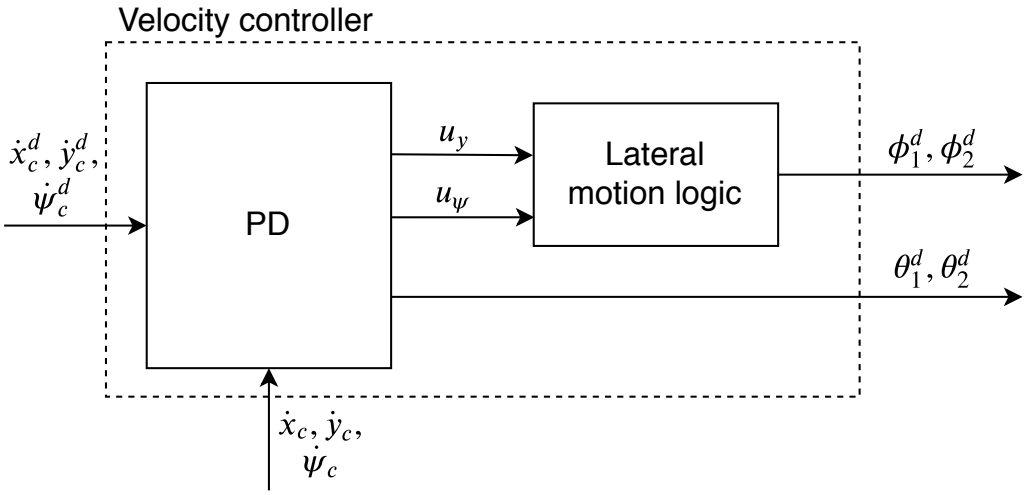



Figure 24: The velocity controller with the lateral motion conversion. The *lateral motion logic* block refers to the equation (8) and (9).

## 6.3 Control laws of individual motions

We have presented individual parts of our control strategy. Control laws for individual motions will be determined in the rest of this section. Our goal is to find relations for control variables $T_1^d, T_2^d, \phi_1^d, \phi_2^d, \theta_1^d, \theta_2^d$. We will divide this problem into several subtasks according to individual control variables. Thus we introduce a longitudinal, lateral and vertical motion of the coupled system. PID gains of individual control laws were determined empirically.

### 6.3.1 Longitudinal motion

The longitudinal motion is the motion across the x-axis of the system (axis $C_x$ in figure 20). This motion is caused by the tilting of the pair of helicopters around their y-axis ($B_1, y$ and $B_2, y$). We introduce the control law of this motion as follows:

$$\theta_1^d[n] = \theta_2^d[n] = K_{p,x}^v e_x^v[n] + K_{d,x}^v \frac{\Delta e_x^v[n]}{\Delta t}, \tag{6}$$

where

- $\theta_1^d, \theta_2^d$ are desired pitch angles of the pair of UAVs.
- $K_{p,x}^v, K_{d,x}^v$ are proportional and derivative gain of velocity PD controller.
- $e_x^v[n] = \dot{x}_c^d[n] - \dot{x}_c[n]$.
- $\Delta t$ is time difference.

- $n$ is the time sample index.

The desired velocity $\dot{x}_c^d$ is obtained from control law

$$\dot{x}_c^d[n] = K_{p,x}^p e_x^p[n] + K_{i,x}^p \sum_{i=0}^{n} e_x^p[i]\,\Delta t + K_{d,x}^p \frac{\Delta e_x^p[n]}{\Delta t}, \tag{7}$$

where

- $K_{p,x}^p, K_{i,x}^p, K_{d,x}^p$ are proportional, integral and derivative gain of position PID controller.
- $e_x^p[n] = x_c^d[n] - x_c[n]$.

### 6.3.2 Lateral motion

The lateral motion contains a motion across the y-axis (axis $C_y$) and heading rotation of the system (around $C_z$ axis). These motions are caused by tilting quadcopters around axis $B_1, x$ and $B_2, x$. The used control law according to the figure 24 is:

$$\phi_1^d[n] = u_y[n] + u_\psi[n], \tag{8}$$
$$\phi_2^d[n] = u_y[n] - u_\psi[n], \tag{9}$$

where $u_y$ and $u_\psi$ are control actions of lateral motion. They have the following meaning:

$$u_y[n] = K_{p,y}^v e_y^v[n] + K_{d,y}^v \frac{\Delta e_y^v[n]}{\Delta t}, \tag{10}$$
$$u_\psi[n] = K_{p,\psi}^v e_\psi^v[n] + K_{d,\psi}^v \frac{\Delta e_\psi^v[n]}{\Delta t}, \tag{11}$$

where

- $\phi_1^d, \phi_2^d$ are desired roll angles of the pair of UAVs.
- $K_{p,y}^v, K_{d,y}^v, K_{p,\psi}^v, K_{d,\psi}^v$ are proportional and derivative gain of velocity PD controller for lateral motion.
- $e_y^v[n] = \dot{y}_c^d[n] - \dot{y}_c[n]$.
- $e_\psi^v[n] = \dot{\psi}_c^d[n] - \dot{\psi}_c[n]$.

Desired velocities $\dot{y}_c^d$ and $\dot{\psi}_c^d$ are in the form:

$$\dot{y}_c^d[n] = K_{p,y}^p e_y^p[n] + K_{i,y}^p \sum_{i=0}^{n} e_y^p[i]\,\Delta t + K_{d,y}^p \frac{\Delta e_y^p[n]}{\Delta t}, \tag{12}$$

$$\dot{\psi}_c^d[n] = K_{p,\psi}^p e_\psi^p[n] + K_{i,\psi}^p \sum_{i=0}^{n} e_\psi^p[i]\,\Delta t + K_{d,\psi}^p \frac{\Delta e_\psi^p[n]}{\Delta t}, \tag{13}$$

where

- $K_{p,y}^p, K_{i,y}^p, K_{d,y}^p, K_{p,\psi}^p, K_{i,\psi}^p, K_{d,\psi}^p$ are proportional, integral and derivative gains of position PID controller for lateral motion
- $e_y^p[n] = y_c^d[n] - y_c[n]$.
- $e_\psi^p[n] = \psi_c^d[n] - \psi_c[n]$.

### 6.3.3 Vertical motion

Vertical motion is caused by the thrust of individual helicopters. We control height and pitch angle of the coupled system by the thrust. We define the following control law according to figure 23:

$$T_1^d[n] = u_z[n] + u_\theta[n], \tag{14}$$

$$T_2^d[n] = u_z[n] - u_\theta[n], \tag{15}$$

where $u_z$ and $u_\theta$ are control actions of vertical motion. They are defined as:

$$u_z[n] = u_0 + K_{p,z}^p e_z^p[n] + K_{i,z}^p \sum_{i=0}^{n} e_z^p[i]\,\Delta t + K_{d,z}^p \frac{\Delta e_z^p[n]}{\Delta t}, \tag{16}$$

$$u_\theta[n] = K_{p,\theta}^p e_\theta^p[n] + K_{i,\theta}^p \sum_{i=0}^{n} e_\theta^p[i]\,\Delta t + K_{d,\theta}^p \frac{\Delta e_\theta^p[n]}{\Delta t}, \tag{17}$$

where

- $u_0$ is empirically determined constant. It corresponds to the thrust of hovering UAV in a steady state.
- $T_1^d, T_2^d$ are desired thrusts of the pair of UAVs.
- $K_{p,z}^p, K_{i,z}^p, K_{d,z}^p, K_{p,\theta}^p, K_{i,\theta}^p, K_{d,\theta}^p$ are proportional, integral and derivative gain of position PID controller for vertical motion.
- $e_y^p, e_\theta^p$ are differences between desired states $z_c^d, \theta_c^d$ and measured states $z_c, \theta_c$.

# 7 Verification of designed control method

The designed control method has to be verified in the simulator before it will be applied to the hardware platform. This section presents the verification of the control section presented in chapter 6. We tested the system in diverse scenarios to show its functionality and find and mitigate its flaws. First, we will introduce a step response of the system for each type of motion (longitudinal, lateral, vertical), that will prove the fundamental behavior of the control system. Another test includes waypoint guidance and verification of the ability to reject external disturbance (simulated wind). Further, all constants of the cascade PID regulator have tuned experimentally thanks to this approach.

## 7.1 Longitudinal-axis control verification

Verification for longitudinal motion is primarily done by step response of the coupled system (see figure 25). We can see that setpoints with relatively small change (where the velocity limit is not reached) are achieved smoothly. Overshoot can be seen in the other case. However, experiments shown that this overshoot is not higher than 7 % of the setpoint change size and it is not higher than 0.35 m at the same time. We achieved these results thanks to the presence of the velocity controller. The right part of the figure 25 shows that maximal speed is limited successfully by the velocity controller. Noise behavior around the limit is caused by a rapidly changing setpoint of the inner control loop and by using the PD controller.

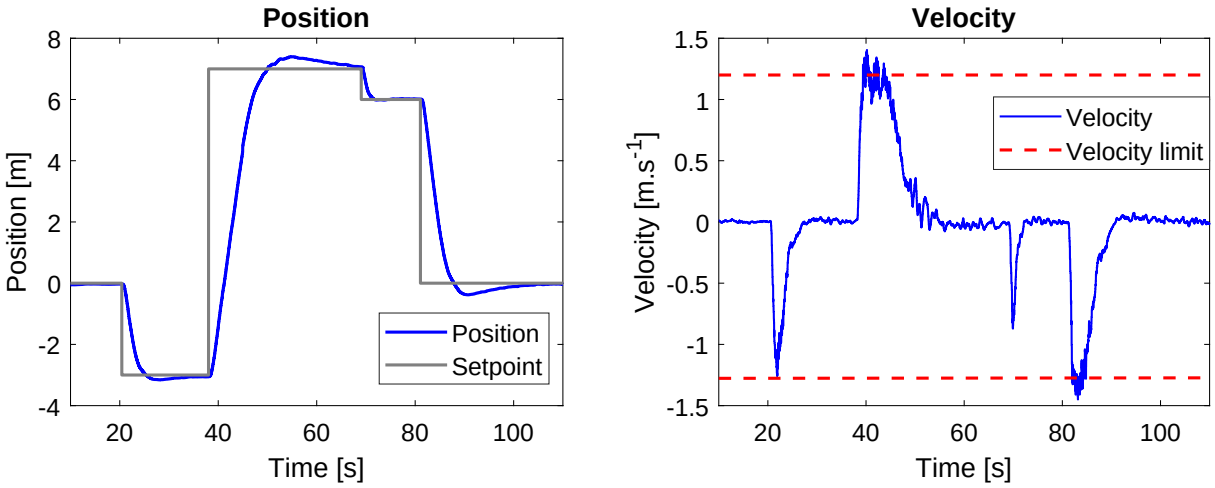



Figure 25: Position step response in the x-axis (longitudinal motion) complemented by velocity and its limit.

## 7.2 Lateral-axis control verification

We divided experiments verifying the control method for lateral motion into translation along the y-axis and rotation around the z-axis. Our goal was to show the mutual influence of these movements since they come from the same source (roll tilt of UAVs). Figure 26 shows, that we achieved similar results for lateral translation as in the longitudinal case. However, this motion causes disturbance in the heading of the system. The Source of this disturbance can be found in a slight difference in the tilt of individual multicopter that makes difference in UAV's speed. The result is, that one UAV starts to overtake the other one. Then the integration part of the PID controller mitigates this error. Nevertheless, error in the yaw angle of the system is in the range of tenths of a radian (normally it does not exceeds 15°) and it does not compromise other system functions.

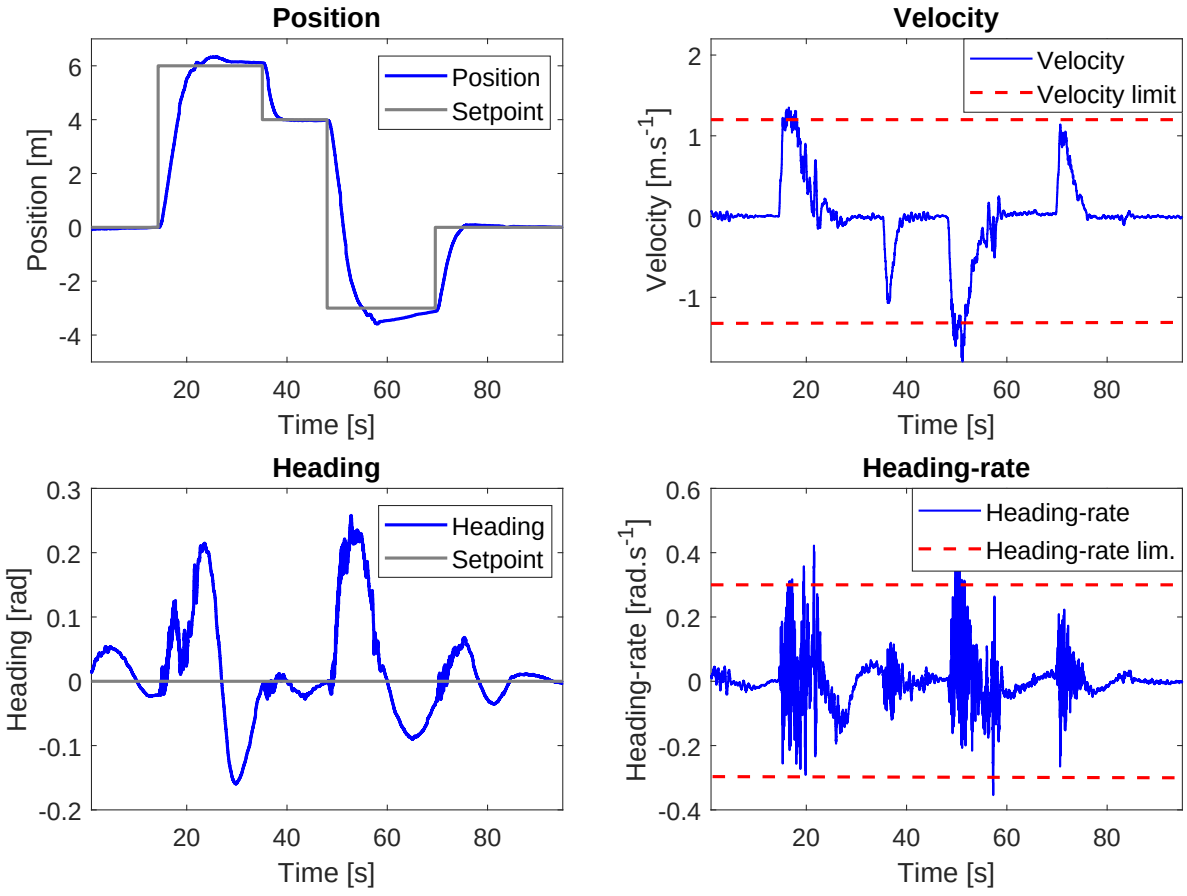



Figure 26: Lateral translation step response. An error in the heading of the system can be observed.

The same behavior can be seen during heading rotation, whose step response is in figure 27. Change in the heading of the system may have caused an error in the lateral

position which is significant for large rotation. However, unlike the error presented above, experiments show that this disturbance can be mitigated by greater limitation of yaw-rate in the inner controller loop.

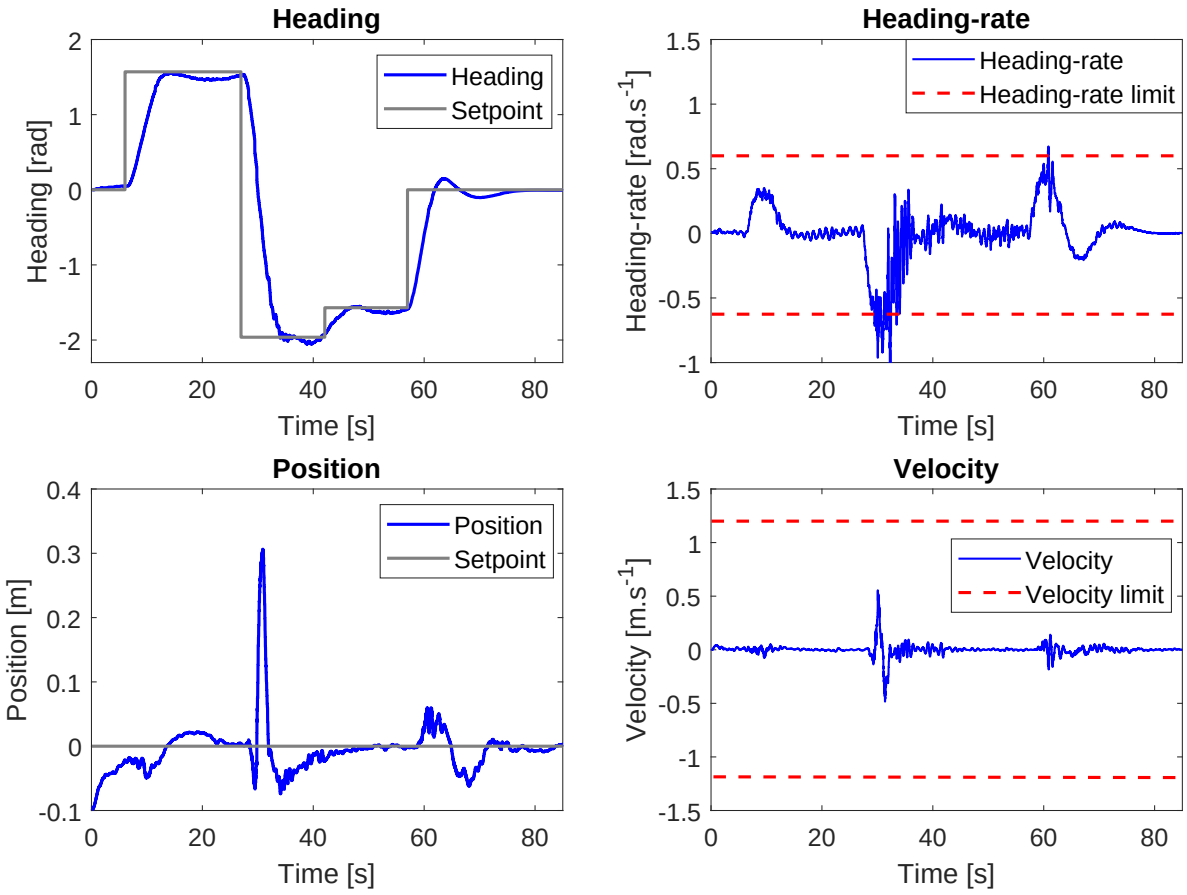



Figure 27: Heading rotation step response complemented by heading-rate and its influence on lateral position.

We have to state that the change of setpoint of heading is the riskiest part of automatic control of the coupled system in the simulator. It brings inconsistent step responses to the same setpoint changes (responses of varying speed, varying overshoot size, and lateral position error influence). Further, it also causes occasional physical crashes of the system and robot model collapsing in the Gazebo simulator. It was found out, that these problems have origin in the Gazebo physics engine that is not able to operate with inappropriate inertia definition. We introduce a description and possible solution to this problem below.

Universal joints of the designed gripper are implemented as two nested sphere links connected by revolute joints. These joints have the perpendicular axis of rotation and they allow tilt of UAV (rotation around x-axis and y-axis of the body frame of the spherical joint). Ideally, the final model disables yaw rotation of UAV considering a grasped payload,

since no joint could allow such motion. Sphere links are primarily defined by their radius, mass, and inertia. An imperfection of the Gazebo simulator is that the physical engine of Gazebo can not build reliable and accurate simulations for models with relatively small elements in defined inertia matrix [28] ($\approx 10^{-4}$ or less). In our case, the error manifests itself in a way that it allows limited yaw rotation of a UAV. The smaller the inertia is, the higher the range of undesirable yaw rotation is. From this, we can deduce the following two boundary states:

- Inertia parameters are corresponding to the real device, but they are so low, that the described glitch disables the possibility to estimate the position of the second UAV.
- Inertia parameters are multiplied so much that they significantly mitigate the Gazebo glitch. However, this approach brings inertial forces that act on the UAVs and grasped payload and they make control much more difficult.

A compromise was chosen between those states for simulation and verification of the designed control strategy. This has resulted in sufficient mitigation of the glitch. However, side effects are still partially present and they bring inconsistent behavior during yaw rotation of the coupled system. This problem will not be present in real experiments.

## 7.3 Vertical-axis control verification

The verification of motions in the vertical plane was done in the same manner as in the horizontal case. Figure 28 shows the step response in height control with its side

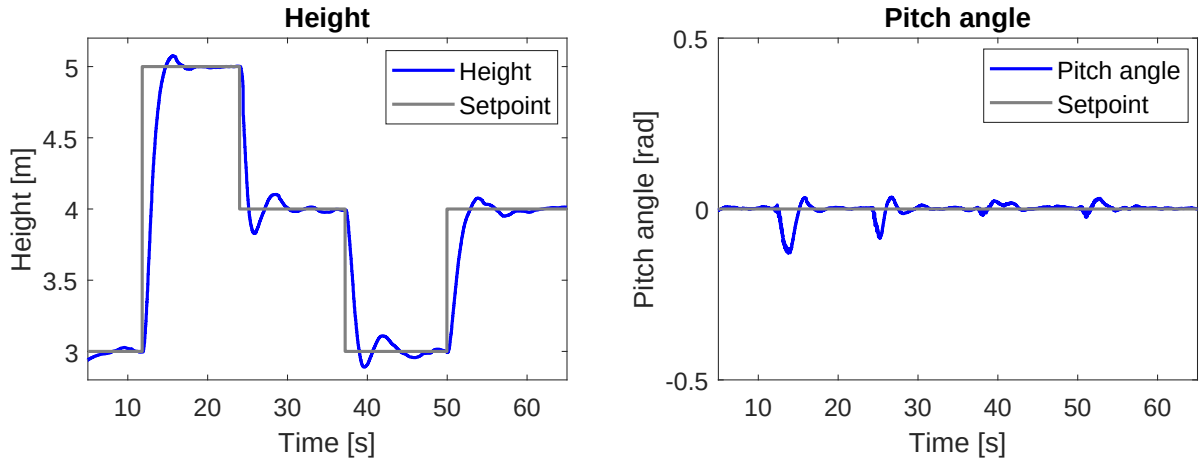



Figure 28: Height step response. Disturbances in pitch angle are observed.

effect on the pitch angle of the system. Observed disturbances have the same origin as

disturbances described above, thus one UAV starts slightly overtake the second one until the pitch controller settles pitch angle again.

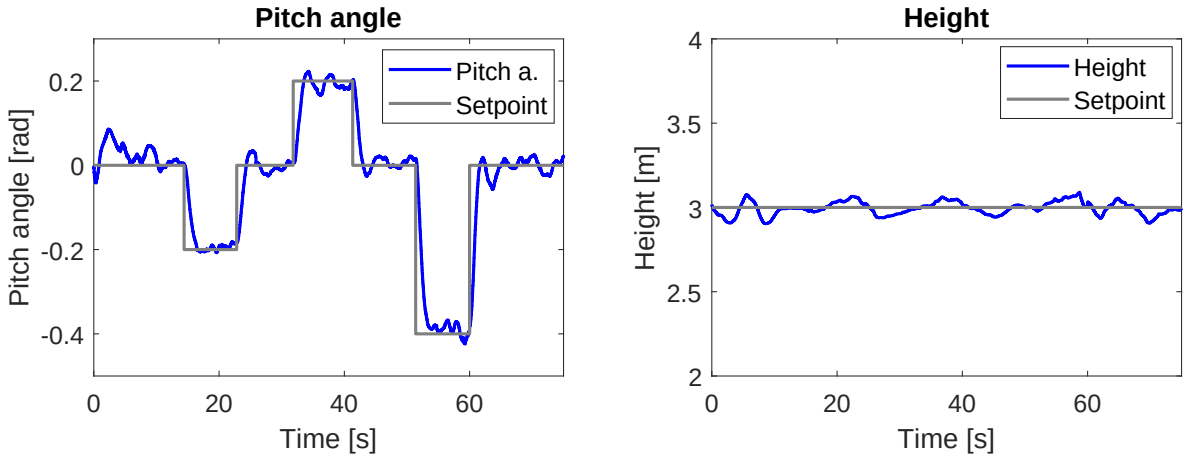



Figure 29: Pitch angle step response. No effect on height is observed.

Step response in pitch angle control (figure 29) shows the possibility to tilt the coupled system. This feature can be used in operations, where a payload needs to be put on a sloping surface. However, this motion is not suitable for larger angles (more than 0.4 rad), where the system starts to be in the area of its stability limit.

## 7.4 Disturbance rejection

In this section, we are experimentally verifying the ability of the coupled system to reject or mitigate disturbances caused by external forces. We designed forces that simulate wind in the Gazebo simulator. We can define force and torque that can act in a defined reference point of some Gazebo model's link. Thus, we can implement any type of external force acting on our system. The magnitude of the applied force was computed by the expression

$$F_D = \frac{1}{2} C_D \rho A v^2, \tag{18}$$

where $F_D$ is the drag force, $C_D$ is drag coefficient, $\rho$ is the air density, $A$ is the cross-section area of the coupled system and $v$ is the speed of the object relative to the surrounding air (wind speed). The force corresponds to the influence of wind since some parameters in the calculation were estimated. However, we consider this approach accurate enough. Drag force is then proportionally divided into several parts that are applied in multiple points of the Gazebo model.

First, we did experiments with simulated wind acting in the longitudinal axis of the coupled system. We simulated force corresponding to wind speed up to 10 $ms^{-1}$ (Beaufort number 5 - fresh breeze). The results of this experiment are shown in figure 30.

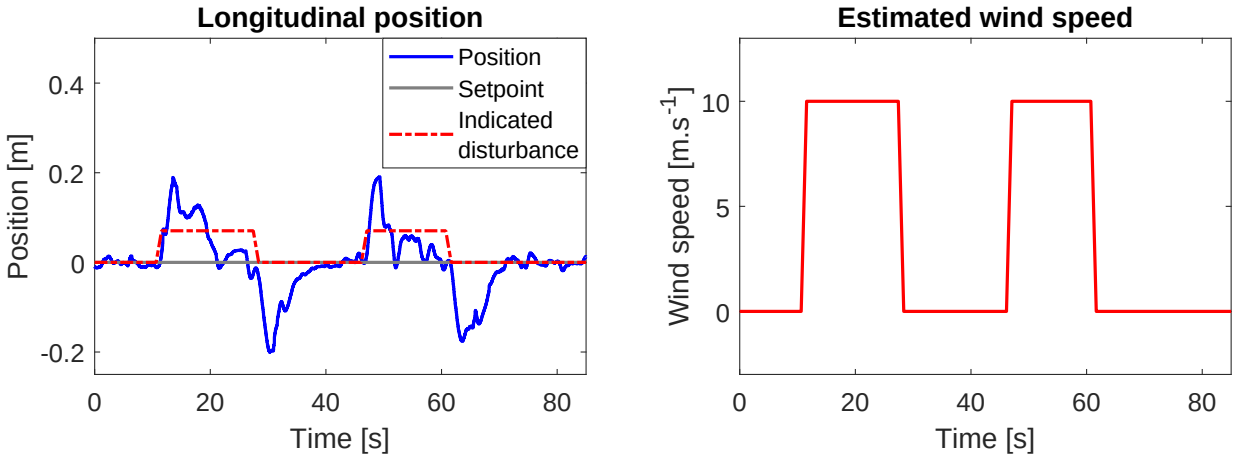



Figure 30: Disturbance rejection in the longitudinal direction. Simulated wind of speed up to 10 $ms^{-1}$ was used.

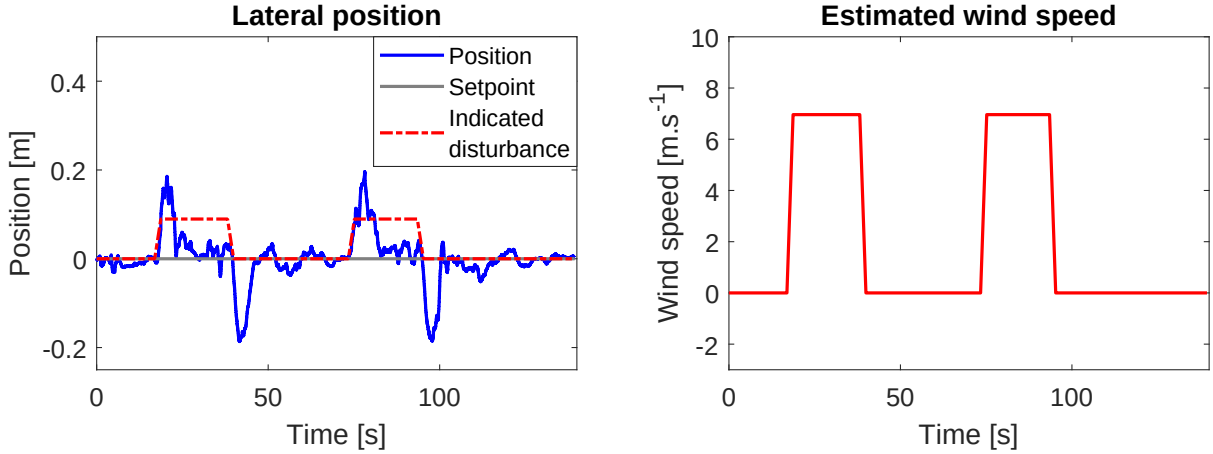



Figure 31: Disturbance rejection in the lateral direction with wind of approximate speed 7 $ms^{-1}$.

The same experiment had been done on the lateral axis of the system. The cross-section area of the system perpendicular to the lateral axis is larger than in the case of the lateral axis. Thus we simulated force corresponding to the wind just up to 7 $ms^{-1}$ (Beaufort number 4 - moderate breeze). The system was not able to stabilize itself safely for airspeed higher than 10 $ms^{-1}$. Figure 31 shown the results of the test. Considering results, we suppose that the coupled system could be used in mild weather conditions.

## 7.5 Waypoint motion

The last experiment showed the possibility to follow defined waypoints by the coupled system. A set of waypoints can be stored in a queue. The system determines heading between waypoints so that the movement is performed in the longitudinal axis.

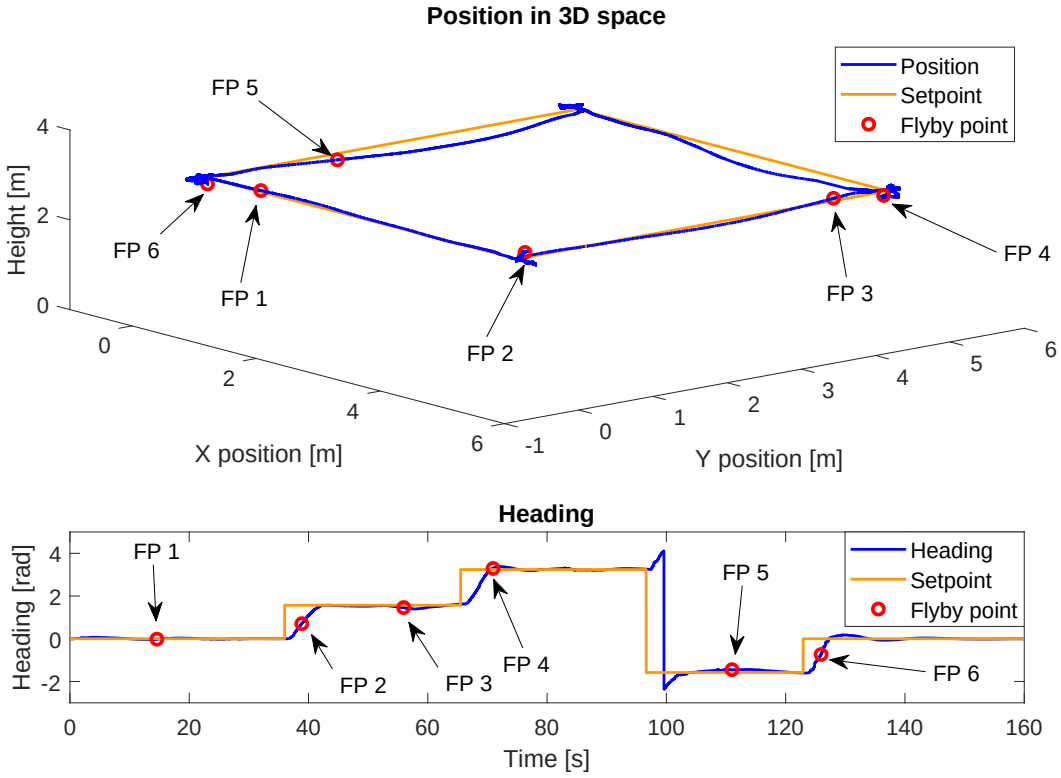


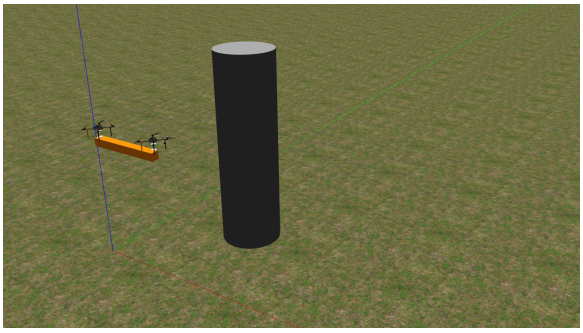

Figure 32: Flight of the coupled system through waypoints. Reference flyby points can be observed.

In the experiment, the coupled system is commanded to fly through 4 waypoints around a pillar model. The final state of the system after the flight is the same as its initial state. Figure 32 shows results of the test. We can observe reference flyby points, that can be compared to the Gazebo simulator flight record in figure 33.

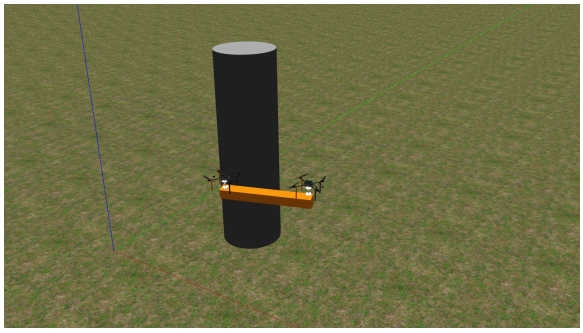
## 7.6 Summary

The verification of the designed control method was done in this section. First, step responses of partial motions were discussed to show the fundamental functionality of the system. Second, the ability to suppress disturbances was presented. This experiment was

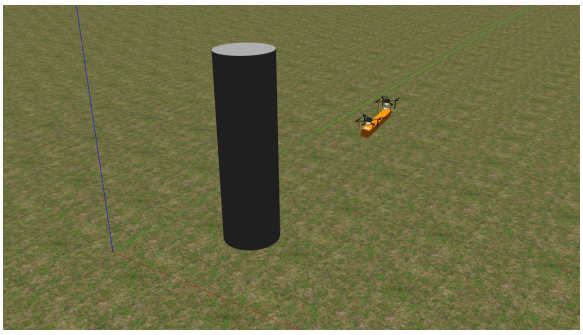
motivated by ordinary outdoor weather conditions. Third, flight through a trajectory set by waypoints showed the possibility of real application of the system in a task like MBZIRC 2020. The last simulated experiment is captured in the video `http://mrs.felk.cvut.cz/uav-cooperative-manipulation`.



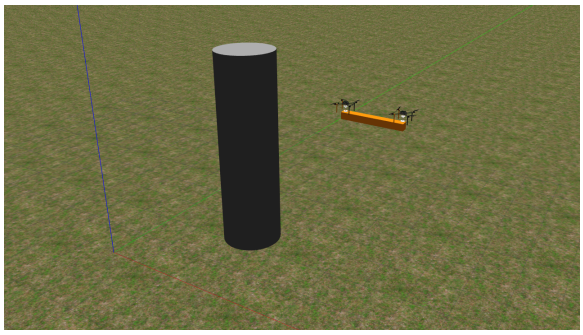
(a) Initial phase of the flight (FP 1).



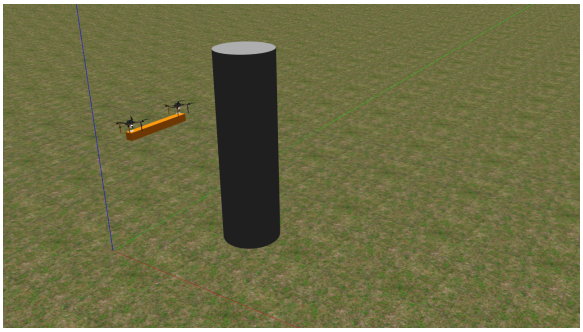
(b) Rotation in the first waypoint (FP 2).



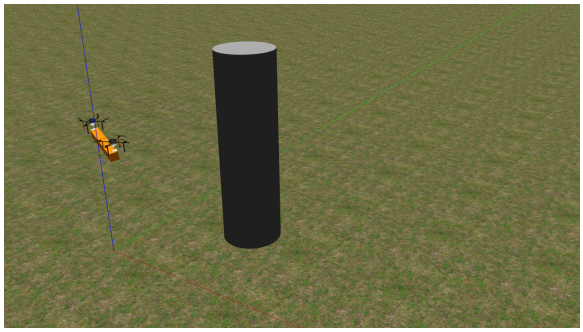
(c) Approach to the second waypoint (FP 3).



(d) Flight to the third waypoint (FP 4).



(e) Final phase of the flight (FP 5).



(f) Rotation around the init. position (FP6).

Figure 33: Representation of the simulated experiment in the Gazebo simulator. Individual figures relate to the flyby points in figure 32. The simulation is captured in the video `http://mrs.felk.cvut.cz/uav-cooperative-manipulation`.

# 8 Grasping and fail-safe systems

We already presented the designed mechanism for collaborative payload-carrying. We built all the necessary parts of the simulated environment to test the tasks of the thesis. Further, we focused on the cooperative control method and its verification in the past two chapters. There is no significant importance for applying safety systems for experiments made in simulation. Our goal is to prepare the whole system for real experiments which will be the subject of further work. Thus, simulation is a suitable place to test and debug supporting systems. The description of these implemented systems is the subject of this section. First, we will introduce a collaborative grasping guidance system. This system secure synchronized attaching and detaching of an object by both of the UAVs. The second presented system is a fail-safe system. It is able to recognize states of collaborative motion that are hazardous for continuing flight.

## 8.1 Collaborative grasping guidance system

The collaborative grasping guidance system (*CGGS*) is a system that secures safe contact between an object and a pair of helicopters. The system is primarily intended for landing and taking off of the coupled system. Landing and taking off are the two most dangerous maneuvers. We require information about contact between a grasping object (a brick with a metal plate on top of it) and an aerial manipulator to ensure synchronized attaching of both grippers. This information is taken from feedback from ROS service that allows us to make a connection between two models (see 5.4). Hall effect sensor is used in the case of a hardware device (see 3.3.3). The *CGGS* was implemented as a state machine which is shown in figure 34.

Presented state machine guarantees a virtual pairing of two aerial manipulators if an object, that is suitable for collaborative manipulation, is spotted. Both grippers are set to be rigid at this moment. Further, multicopters are moved above edges of the object, where aligning begins. They track their grasping areas and keep the same height and heading. Landing is synchronized in terms of vertical speed. Helicopters are returned to their default positions above the brick if their approach is significantly different. A timer is started when multicopters reach grasping height. Grasping is aborted if the timeout expires before both grippers are attached properly. The coupled system takes off only if both grippers are in contact with the brick by both electromagnets. Landing is repeated in the other case. Both gripper's locks are unlocked at the moment when it is attached to the brick. This makes a risk of a crash during take off much lower. Otherwise, undesirable forces may occur, since both helicopters are controlled still individually. The control method is switched and both lower locks are locked during hovering of the system after the take off.

The coupled system is moved to the dropping area after the collaborative task is

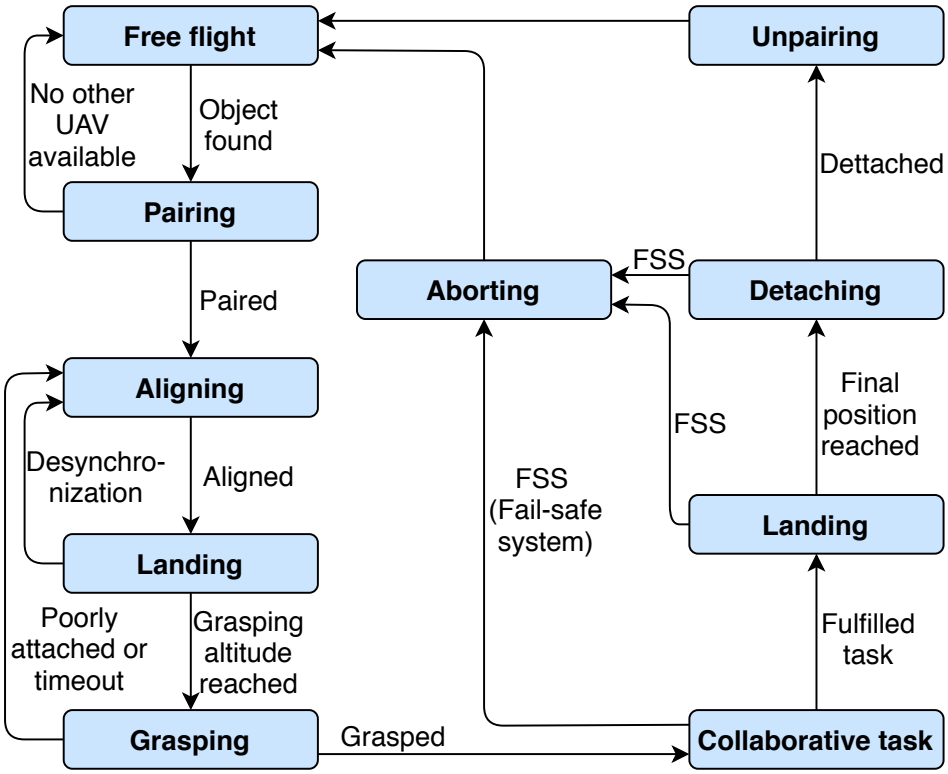



Figure 34: Collaborative grasping guidance system (*CGGS*) in the form of a state machine. The fail-safe system is a subsystem of *CGGS*.

fulfilled. A payload is detached by both grippers at the same time. Further, helicopters take off, they make their grippers rigid and they switch the control method back to the standard control used for single UAV. All actions during collaborative motions are protected with the fail-safe system.

## 8.2 Failsafe system

The failsafe system is engaged when the automatic control for the coupled system is switched on. It protects the coupled system before crashes. These crashes can be caused by the following reasons:

- Loss of grip by at least one gripper.
- Bump of the coupled system into the environment.
- Unknown error in collaborative control. This error can cause unexpected overshoot or oscillation around a setpoint.

The goal of the failsafe system is to detect the presented causes before they destabilize the system. Information about the loss of the grip is taken from Hall effect sensors. A signal from these sensors is processed in control boards of grippers. Further, the information about the proper grip is broadcasting through serial communication to UAV's onboard computer. The failsafe system is executed, if the system evaluates that at least on gripper lost its grip. This case of error is verified in every control loop of the system (running on 100 Hz), due to the high risk of the crash of the coupled system.

We suppose that planning algorithms, that would be used with a designed system, choose trajectories that are not collidable. However, our goal is to minimalize the risk of a crash as much as possible. Thus we want to make bump protection. We remind the states of the system as

$$\mathbf{r}^{(W)} = \begin{pmatrix} x_c & y_c & z_c & \theta_c & \psi_c \end{pmatrix}^T, \tag{19}$$

where $\mathbf{r}^{(W)}$ is a position vector of the coupled system concerning the world coordinate frame. We evaluate the state of the system as a hit on some unknown object if the following inequality is satisfied:

$$| \ddot{\mathbf{r}}^{(W)} | > k\ddot{\mathbf{r}}^{(W)}_{max}, \tag{20}$$

where $\ddot{\mathbf{r}}^{(W)}_{max}$ is a maximal unaffected acceleration of the coupled system when the designed collaborative control method is used. This constant vector was set during experimental flights in the simulator environment as the highest possible acceleration of the system at any setpoint change. Further, $k$ is a safety coefficient defining the boundary between normal and failsafe state.

Automatic detection of abnormal system behavior is vital. Specifically, we detect an excessive overshoot, unwanted oscillations, or faulty reaction of the system on change of a setpoint. Following fault detection solution is valid for any reference signal type, however, we will demonstrate them on step signal reference. Lets define safety limits of a reference signal $\mathbf{s}_r(t)$ as

$$\mathbf{s}_u(t) = \mathbf{s}_r(t) + c, \tag{21}$$

$$\mathbf{s}_l(t) = \mathbf{s}_r(t) - c, \tag{22}$$

where $\mathbf{s}_u(t)$, $\mathbf{s}_l(t)$ are upper and lower safety limits and $c$ is a constant determining size of these limits. We suppose that system is free of faults if all following inequalities hold:

$$\mathbf{r}^{(W)}[T] \leq \mathbf{s}_u[T], \tag{23}$$

$$\mathbf{r}^{(W)}[T] \geq \mathbf{s}_l[T], \tag{24}$$

$$\left|\mathbf{s}_r[T] - \mathbf{r}^{(W)}[T]\right| \leq \left|\mathbf{s}_r[T] - \mathbf{r}^{(W)}[T - \Delta t]\right|, \tag{25}$$

where $T$ is a current control time sample and $\Delta t$ is a relatively small time difference between two samples. The used norm has the meaning of the absolute value of each element in a

vector. Faults detected by this approach are demonstrated in figure 35. The fail-safe system is triggered if the collaborative manipulator is located outside the safety limit and it is receding from the setpoint at the same time.

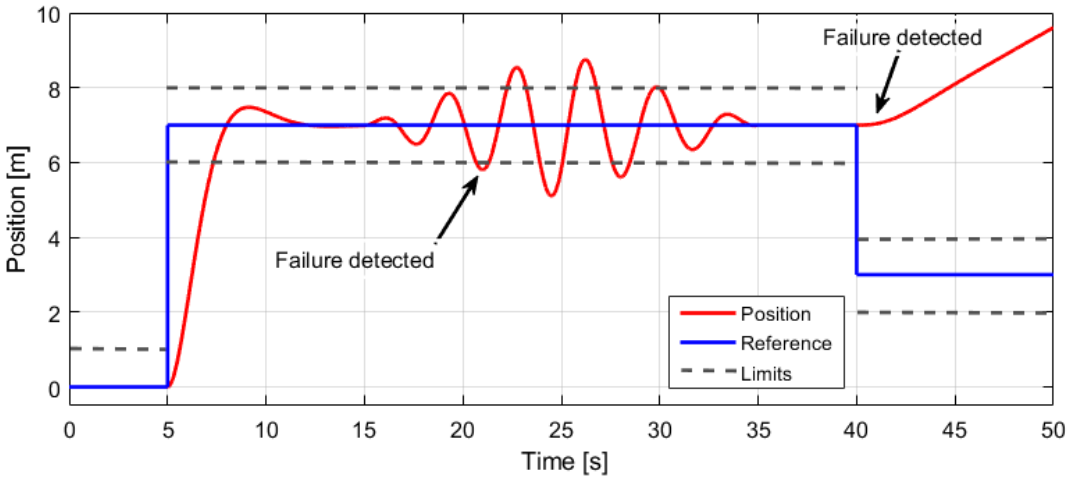



Figure 35: Simulated detection of fault behavior of the controlled system. Unexpected overshoots, oscillations, and other behavior can be detected.

The failsafe system is executed if one of the cases described above is detected. It will switch off all magnets and lock of upper locks in the first step. This action releases the payload immediately. Further, it makes each aerial manipulator rigid. This makes easier to control individual UAVs. The collaborative controller is then switched to a regular control pipeline [22] used in the MRS laboratory for single UAV control [3]. Then stabilization of UAVs and landing in the place outside the dropped payload area or another task is commenced.

# 9 Conclusion

In this thesis, a solution that allows collaborative manipulation motivated by MBZIRC 2020 competition was developed. The proposed system can grasp and transport brick by a pair of unmanned helicopters. A novel hybrid-type gripper was designed for collaborative transport. The use of this device is not limited only to presented competition and it allows rapid prototyping of systems related to the topic of cooperative tasks. Further, a completely new controller was implemented for this thesis. Its cascade structure provides satisfying dynamic properties. The controller was tested and verified in simulated experiments, that showed the success of our approach. Apart from the verification of step responses of partial motions, disturbance rejection ability, and possibility to move through waypoint were presented.

The solution, which was introduced, includes a system for synchronous collaborative grasping of payload. It was a necessary step for the planned commissioning of the system in real experiments. According to the assignment of this thesis, the following tasks have been completed:

- A specialized gripper with a locking mechanism was designed. It allows carrying of large brick by a pair of UAVs as well as carrying smaller brick by a single UAV.
- The gripper was assembled as an HW prototype (for future real experiments) as well as the simulator model.
- Both devices (HW prototype and SW model) was fully integrated into the current system implemented in ROS. The functionality of the HW device was tested during an experimental outdoor camp.
- A cascade PID controller was developed for the coupled system. It allows collaborative payload-carrying by a pair of UAVs.
- The designed system was tested in the realistic Gazebo simulator. The functionality of the cascade PID controller was verified.
- Supporting systems securing the safety of the coupled system during grasping of payload and collaborative manipulation was implemented.
- Both, HW and SW of the system were developed to such conditions, that the system is prepared for real experiments.

This work is one of the first attempts at collaborative manipulation task in Multi-Robot System group. It contributes to previous work primarily by the implementation of a coupled system controller without the need for accurate localization of both UAVs.

## 9.1 Future work

One of the main objectives of this thesis was the preparation of the system for real experiments. Therefore, the objective of future work follows this requirement. The designed controller will be experimentally tested on the hardware platform to verify results that have been achieved during simulated tests. It will primarily entail tunning of controller and verification of partial parts of the designed system in a real environment.

Another future work will follow up on real experiments. Regarding hardware, another iteration of the design of the gripper prototype can be done. We would like to achieve improvement in physical attributes of mechanism (weight, length, etc.) and improvement in reliability and temperature independence of grasping feedback. Regarding software, the controller will be reworked considering shortcomings that will be observed during real experiments. By this approach, we want to achieve a robust and reliable controller that would play the same role as the current control pipeline using with a single UAV.

# References

[1] Allegro MicroSystems, LLC. Allegro A1324. Allegro MicroSystems website, 2019 (accessed May 6, 2020). `https://www.allegromicro.com/en`.

[2] Arduino. Arduino nano documentation. Arduino website, 2020 (accessed May 14, 2020). `https://store.arduino.cc/arduino-nano`.

[3] T Baca, D Hert, G Loianno, M Saska, and V Kumar. Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8, 2018.

[4] Tomáš Báca. Model predictive control of micro aerial vehicle using onboard microcontroller, 2015. Master's thesis, Czech Technical University in Prague.

[5] Gilbert E Contreras and Robert W Graves. Universal joint, May 16 1972. US Patent 3,663,044.

[6] Jan Eric Dentler. *Real-time Model Predictive Control for Aerial Manipulation*. PhD thesis, University of Luxembourg, 2018.

[7] Jonathan Fink, Nathan Michael, Soonkyum Kim, and Vijay Kumar. Planning and control for cooperative manipulation and transportation with aerial robots. *The International Journal of Robotics Research*, 30(3):324–334, 2011.

[8] Guido Gioioso, Antonio Franchi, Gionata Salvietti, Stefano Scheggi, and Domenico Prattichizzo. The flying hand: A formation of uavs for cooperative aerial telemanipulation. In *2014 IEEE International conference on robotics and automation (ICRA)*, pages 4335–4341. IEEE, 2014.

[9] Guillermo Heredia, AE Jimenez-Cano, I Sanchez, Domingo Llorente, V Vega, J Braga, JA Acosta, and Aníbal Ollero. Control of a multirotor outdoor aerial manipulator. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3417–3422. IEEE, 2014.

[10] Daniel Hert. Autonomous predictive interception of a flying target by an unmanned aerial vehicle, 2018. Master's Thesis, Czech Technical University in Prague.

[11] Jiri Horyna. Robotic manipulation onboard an unmanned aerial vehicle, 2018. Bachelor's Thesis, Czech Technical University in Prague.

[12] Suseong Kim, Seungwon Choi, and H Jin Kim. Aerial manipulation using a quadrotor with a two dof robotic arm. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4990–4995. IEEE, 2013.

[13] Suseong Kim, Hoseong Seo, Jongho Shin, and H Jin Kim. Cooperative aerial manipulation using multirotors with multi-dof robotic arms. *IEEE/ASME Transactions on Mechatronics*, 23(2):702–713, 2018.

[14] Petr Kohout. *A system for Autonomous Grasping and Carrying of Objects by a Pair of Helicopters*. PhD thesis, Master's Thesis, Czech Technical University in Prague, Prague, Czech Republic, 2017.

[15] Hyeonbeom Lee, Hyoin Kim, and H Jin Kim. Path planning and control of multiple aerial manipulators for a cooperative transportation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2386–2391. IEEE, 2015.

[16] Hyeonbeom Lee, Hyoin Kim, and H Jin Kim. Planning and control for collision-free cooperative aerial transportation. *IEEE Transactions on Automation Science and Engineering*, 15(1):189–201, 2016.

[17] Giuseppe Loianno, Vojtech Spurny, Justin Thomas, Tomas Baca, Dinesh Thakur, Daniel Hert, Robert Penicka, Tomas Krajnik, Alex Zhou, Adam Cho, et al. Localization, grasping, and transportation of magnetic objects by a team of mavs in challenging desert like environments. *IEEE Robotics and Automation Letters*, 3(3):1576–1583, 2018.

[18] Montserrat Manubens, Didier Devaurs, Lluis Ros, and Juan Cortés. Motion planning for 6-d manipulation with aerial towed-cable systems. 2013.

[19] MBZIRC. MBZIRC competition information. MBZIRC official website, 2020 (accessed May 14, 2020). `http://gazebosim.org/`.

[20] Daniel Mellinger, Michael Shomin, Nathan Michael, and Vijay Kumar. Cooperative grasping and transport using multiple quadrotors. In *Distributed autonomous robotic systems*, pages 545–558. Springer, 2013.

[21] Nathan Michael, Jonathan Fink, and Vijay Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86, 2011.

[22] MRS group. Multi-robot Systems Group UAV system. GitHub website, 2020 (accessed May 14, 2020). `https://github.com/ctu-mrs/mrs_uav_system`.

[23] Jason M. O'Kane. *A Gentle Introduction to ROS*. CreateSpace Independent Publishing Platform, 10 2013. `http://www.cse.sc.edu/~jokane/agitr/`.

[24] Open Source Robotic Foundation. Gazebo and URDF tutorial. Gazebo website, 2014 (accessed April 3, 2020). `http://gazebosim.org/tutorials/?tut=ros_urdf`.

[25] Open Source Robotic Foundation. URDF tutorial. ROS wiki website, 2019 (accessed April 3, 2020). `http://wiki.ros.org/urdf`.

[26] Open Source Robotics Foundation. Gazebo simulator. Gazebo official website, 2014 (accessed April 10, 2020). `http://gazebosim.org/`.

[27] Pecka modelar s.r.o. Hitec HS-430BH. Peckamodel RC store website, 2013-2020 (accessed May 11, 2020). `https://www.peckamodel.cz/1hi3082-hs-430-bh`.

[28] J Peraire and S Widnall. Lecture l26-3d rigid body dynamics: The inertia tensor. *Dynamics*, pages 1–12, 2009.

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

[30] Jiao Ren, Du-Xin Liu, Kang Li, Jia Liu, Yachun Feng, and Xiaoxin Lin. Cascade pid controller for quadrotor. In *2016 IEEE International Conference on Information and Automation (ICIA)*, pages 120–124. IEEE, 2016.

[31] Martin Saska, Tomas Baca, Vojtěch Spurný, Giuseppe Loianno, Justin Thomas, Tomáš Krajník, Petr Stepan, and Vijay Kumar. Vision-based high-speed autonomous landing and cooperative objects grasping—towards the mbzirc competition. In *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems—Vision-based High Speed Autonomous Navigation of UAVs (Workshop), Daejeon, South Korea*, pages 9–14, 2016.

[32] Martin Saska, Vojtěch Vonásek, Jan Chudoba, Justin Thomas, Giuseppe Loianno, and Vijay Kumar. Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles. *Journal of Intelligent & Robotic Systems*, 84(1-4):469–492, 2016.

[33] Max Schwarz, David Droeschel, Christian Lenz, Arul Selvam Periyasamy, En Yen Puang, Jan Razlaw, Diego Rodriguez, Sebastian Schüller, Michael Schreiber, and Sven Behnke. Team nimbro at mbzirc 2017: Autonomous valve stem turning using a wrench. *Journal of Field Robotics*, 36(1):170–182, 2019.

[34] Koushil Sreenath and Vijay Kumar. Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots. *rn*, 1(r2):r3, 2013.

[35] Marco Tognon, Chiara Gabellieri, Lucia Pallottino, and Antonio Franchi. Aerial comanipulation with cables: The role of internal force for equilibria, stability, and passivity. *IEEE Robotics and Automation Letters*, 3(3):2577–2583, 2018.

[36] V. Walter and M. Saska and A. Franchi. Fast mutual relative localization of uavs using ultraviolet led markers. In *2018 International Conference of Unmanned Aircraft System (ICUAS)*, 2018.

[37] V. Walter and N.Staub and M. Saska and A. Franchi. Mutual Localization of UAVs based on Blinking Ultraviolet Markers and 3D Time-Position Hough Transform. In *14th IEEE International Conference on Automation Science and Engineering (CASE)*, 2018.

[38] VanDoren, Vance. Fundamentals of cascade control. Control Engineering website article, 2014 (accessed May 2, 2020). `https://www.controleng.com/articles/fundamentals-of-cascade-control/`.

# Appendix A CD Content

In Table 3 are listed names of all root directories on CD.

| Directory name | Description |
|---|---|
| thesis | Master's thesis in pdf format |
| thesis_sources | latex source codes |
| models | STL files for 3D printing |
| gripper | sources for gripper board |
| src | executable and supporting files from ROS |

Table 3: CD Content

# Appendix B List of abbreviations

In Table 4 are listed abbreviations used in this thesis.

| Abbreviation | Meaning |
|---|---|
| **DOF** | Degree of Freedom |
| **GNSS** | Global Navigation Satellite System |
| **IDE** | Integrated Development Environment |
| **MBZIRC** | Mohamed Bin Zayed International Robotics Challenge |
| **MPC** | Model Predictive Controller |
| **MRS** | Multi-Robot Systems group |
| **PID** | Proportional-Integral-Derivative controller |
| **ROS** | Robot Operating System |
| **RRT** | Rapidly-exploring Random Tree |
| **RTK** | Real-time Kinematic |
| **UART** | UUniversal Asynchronous Receiver-transmitter |
| **UAV** | Unmanned Aerial Vehicle |
| **URDF** | Universal Robot Description Format |
| **USB** | Universal Serial Bus |
| **Xacro** | XML macro |

Table 4: Lists of abbreviations