This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 861111

Drones4Safety

Research & Innovation Action (RIA)

Inspection Drones for Ensuring Safety in Transport Infrastructures

Collaborative drone swarm system hardware and software

D5.3

Due date of deliverable: 31.05.2022

Start date of project: June 1st, 2020

Type: Deliverable
WP number: WP5

Responsible institution: Aarhus University (AU)

Editor and editor's address: Rune Hylsberg Jacobsen (rhj@ece.au.dk)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 861111

Version 1.0
Release Date: May 31, 2022

Project funded by the European Commission within the Horizon 2020 Programme		
Dissemination Level		
PU	Public	X
CO	Confidential, only for members of the consortium (including the Commission Services)	

Authors/contributors

The following members of the project have contributed to the content and writing of this deliverable:

- Rune Hylsberg Jacobsen (AU, editor)
- Kaspar David Hageman (AU)
- Frederik Falk Nyboe (SDU UAS)
- Nicolaj Haarhøj Malle (SDU UAS)
- Liping Shi (AU)
- Lea Matlekovic (SDU IMADA)
- Emad Ebeid (SDU UAS)

Change Log

Rev.	Date	Who	Site	Change
0.1	01/06/2020	Annika Lindberg	SDU	Created initial version
0.2	16/05/2022	Rune H Jacobsen	AU	First version
0.3	24/05/2022	Rune H Jacobsen	AU	Updated after review comments. Executive summary added.
0.4	27/05/2022	Rune H Jacobsen	AU	Video links updated
1,0	30/05/2022	Rune H Jacobsen	AU	Final version. Updated with comments from the review.

1 Executive Summary

Deliverable D5.3 provides a demonstration of the collaborative swarm system to support autonomous inspection missions. The demonstrations integrate core parts of the research of the innovation undertaken in WP5. Accompanying this report, 5 video presentations demonstrate these parts in action.

First, we demonstrate the inspection drone hardware including its sensors and computational electronics. The second video demonstrates autonomous formation flying and inspection with two drones use a leader follower mechanism. A mockup bridge is built in the indoor drone test facility at Aarhus University. The third video demonstrates the autonomous inspection of this bridge structure. The fourth video demonstrates the collaboration with cloud services of WP6. It shows the interplay with mission planning functions and the function of detecting and uploading images classified with faults in the inspected infrastructure. Fifth, and final video demonstration shows the collaboration with an autonomous drone and a human operator. The scenario accommodate situation where an inspection drone needs to be tethered due to flight restrictions and regulations.

This deliverable can be read in combination with deliverable D5.4 “Test and validation report”, that documents in details the results from test and validation activities in WP5.

Contents

Acronyms

Acronym	Description
AC	Alternating Current
AI	Artificial Intelligence
CPU	Central Processor unit
DC	Direct Current
FPGA	Field-programmable gate array
FPS	Frames per second
IMU	Initial Measurement Unit
OBC	On-Board Computer
OS	Operating System
ROS	Robot Operating System
ROS2	ROS version 2
WP	Work Package

2 Introduction

We are witnessing an increasing uptake of autonomous drones for carrying out complex and potentially unsafe tasks in society. While most demonstrations of the application of drones today have involved only a single drone, the industry is rapidly advancing the transition towards the deployment of groups of collaborating autonomous drones, also known as "swarms", driven by a demand for cost reductions. Swarming involves the coordinated operation of multiple drones to accomplish a large-scale or complex mission. Swarms may be composed of multiple drones or groups of homogeneous drones controlled by a centralized or decentralized algorithm. The benefits of swarming include improved performance on tasks that can run in parallel, the ability to perform multiple actions simultaneously in different locations, as well as increased fault tolerance. Deliverable D5.3 of the project aims to demonstrate critically important parts of the research and innovation on the collaborative multi-drone system of WP5 also referred to as the drone swarm.

2.1 Purpose and scope

The aim of the deliverable is to demonstrate the prototype hardware and software of the drone system. Demonstrations are provided through a series of video demonstrations of selected parts of the system. This approach is common in the robotics community and to publish such demonstration videos alongside research and innovation output.

The videos are the key contributions of D5.3 with this document being the "wrapping" of the demonstrations outlining the intended content of the video recordings to be made during the last part of WP5. The demonstrations are output of Task T5.6 bringing together elements from Task T5.1 (drone hardware), T5.2 (communication infrastructure), T5.3 (algorithms for autonomous collaboration) and T5.4 (cloud services interactions).

Deliverable D5.3 is a demonstration. Details concerning functions, verification and validation of the involved parts are described in Deliverable D5.4 "Test and validation report".

3 Demo video scenarios

3.1 Presentation of drone hardware

In the drone hardware demonstration, the hardware (sensors and computational electronics) is outlined along with a description of the firmware implementation, running on the onboard computer. As the sensors, computational electronics, and firmware/operating system, hence forward denoted the *drone hardware stack*, forms the basis on top of which more advanced behaviors are implemented, the video presentation is made to document and showcase exactly these elements. Contrarily, considerations about the drone airframe and mechanical design are not included in the presentation, because 1) the primary factor dictating the mechanical layout of the drone is the charging system, which is considered in WP3, and 2) the drone hardware stack is agnostic to the mechanical layout of the drone.

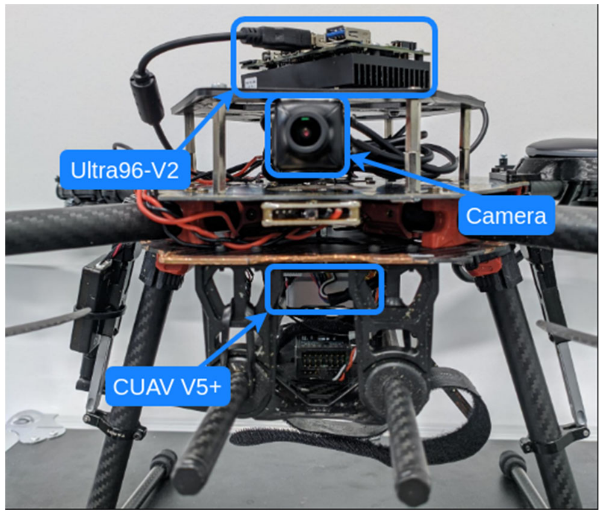

Figure 1: Drone hardware platform with a camera sensor.

The drone hardware stack comprises of the following elements and depicted in Figure 1:

1. **The sensing system** allows the drone to safely navigate in proximity to infrastructural objects with full awareness of the surrounding environment. The sensors equipped on the drone are therefore not considered as part of the drone payload but instead as part of the drone hardware stack, enabling low-level safe behaviors. The choice of sensors for the drone is additionally documented in the 2021 publication [1] and shown in Figure 2.

   The used sensors are

   a. **A camera** for deriving directional information about powerline cables in proximity to the drone;
   b. **A mmWave radar sensor** for obtaining three-dimensional information about cables and other objects in proximity to the drone; and
   c. **Magnetic sensors** for additional awareness of the environment in between cables carrying high currents.

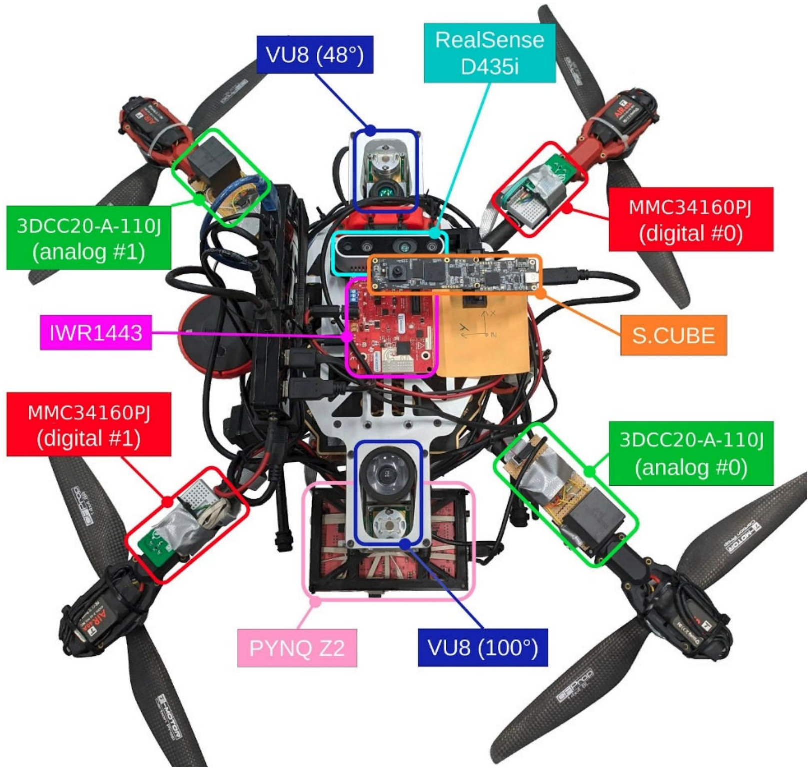

Figure 2: UAV sensors for cable detection.

2. **The computational electronics** are the nervous system of the drone. Low- and high-level processing happens on the computational electronics. The computations necessary for the drone operations are generally divided into two categories: Flight critical processing and application processing. Following this distinction, the computational electronics covers two separate computational boards:
    a. **The flight controller** is responsible for flight critical processing. This includes redundant processing for the low-level flight controller, IMU interfacing, and safety features. The CUAVv5+ flight controller running the PX4 flight software stack is used in the project, as it is one of the most modern versions of the technology and offers a broad range of features.
    b. **The onboard computer** is responsible for all application processing for the drone. On this board, all non-flight critical systems are running, including detection processing, swarming applications, inspection applications, etc. The onboard computer fetches internal information from the flight controller, while providing the flight controller with high-level control inputs to achieve the mission. As an onboard computer, the Ultra96v2 board is used, featuring CPU as well as FPGA for adaptive computing, which is the newest branch of computational paradigm for mobile robotics with limited resources required to run advanced and heavy algorithms.
3. **The firmware/OS** is the low-level, enabling software running on the onboard computer. The firmware includes Linux drivers, hardware device tree specifications, and kernel modules built specifically for the Ultra96v2 board using a Yocto based Linux distribution designed with the Xilinx tools Petalinux and Vivado. Using the self-developed tool MPSoC4Drones, the low-level firmware is combined with an Ubuntu operating system for maximum compatibility with existing robotics software. On top of the Ubuntu operating system, ROS2 is running and communicating with the flight controller using

FastRTPS. Additionally, FPGA implementations can be appended to the firmware/OS stack for acceleration of heavy computational components.

The design-flow and the firmware/OS architecture, as well as the MPSoC4Drones toolset is thoroughly explained in the 2022 publication [2] and shown in Figure 3.

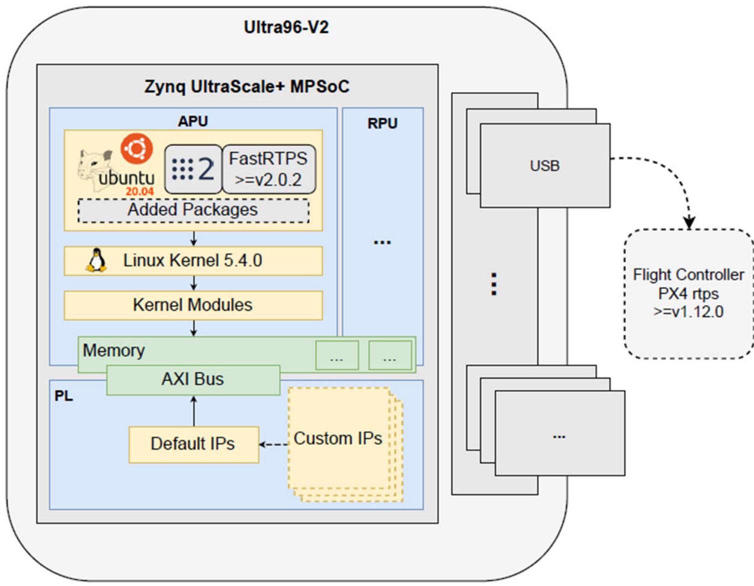

Figure 3: System diagram of MPSoC4Drones build results

A video presentation will be made to explain the drone hardware stack. The video will have the following segments to cover the general work:

1. First, the computational electronics are presented. The responsibilities of the onboard computer and the flight controller are outlined.
2. Then, the sensors are outlined. A visual demonstration of the sensor capabilities is given.
3. Then, the workflow of building the Ultra96v2 firmware/OS image with MPSoC4Drones is shown, and the resulting architecture is presented.
4. Finally, an example application is demonstrated, showcasing the interconnection between the elements of the drone hardware stack, as well as the MPSoC4Drones workflow. The drone will be shown to fly according to data obtained from processing a video stream from the camera using a heavy AI algorithm running on the FPGA.

Video link (2m 5s):

https://nextcloud.sdu.dk/index.php/s/98SScJ34DK3AfMi?dir=undefined&openfile=174302677

Password: Drones4Safety

File name: D4S-WP5 – Drone hardware.mp4

3.2 Formation flying

Formation flying requires communication between drones. The communication system is established based on a WiFi network with an access point. The formation flying function is designed using a leader-follower scheme. Specifically, the leader drone receives the mission description (a mission file) and mission control commands (service calls) from the ground control station. A mission file consists of a list of waypoints for both the leader drone and follower drones. These waypoints are pre-generated according to the mission. Then, the leader drone coordinates mission progress with the follower drone by controlling the distribution of the mission element, i.e., waypoints. At last of each iteration, the waypoint is executed by each drone using onboard position control.

We demonstrate the formation flight function with two drones in an indoor test environment (Figure 4). We made a mockup of a bridge segment. Two drones estimate their state based on onboard sensors using visual inertial odometry tracking software.

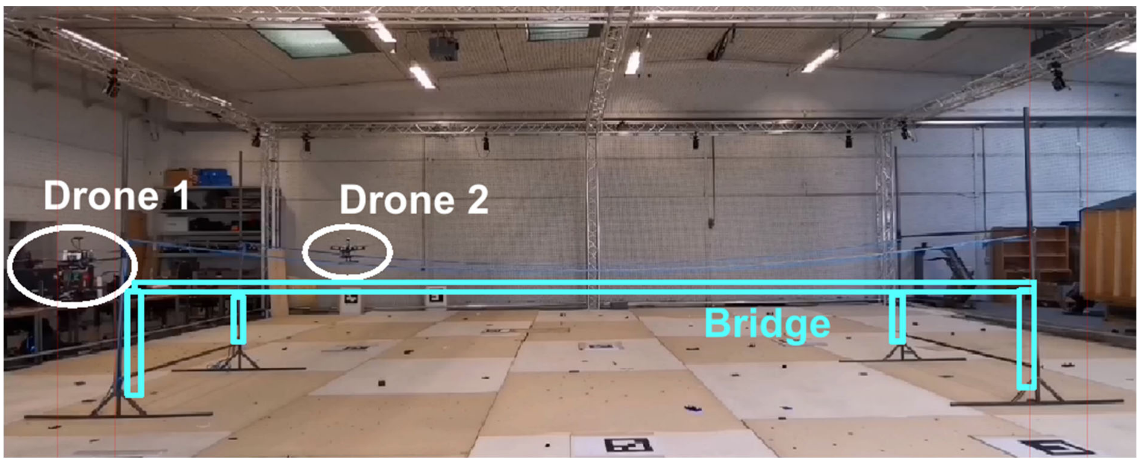

Figure 4: Demonstration of formation flying along a modeled bridge segment.

Two drones are controlled by their onboard computers while receiving mission command from the GCS. Communication between the two drones is established. Formation flying is first demonstrated with vertical movement along the bridge pillars (modeled with vertical metallic bars) followed by horizontal movement along the bridge deck (blue ropes). The second pylon structure is inspected with another vertical movement and drones are hereafter landed safely.

A final video demonstration was performed at the SDU test facility at HC Andersen Airport in Denmark. The test facility has a power line segment, and a pair of drones were autonomously inspecting the segment. A link to a video demonstration follows below.

Video link (1m 10s)

https://nextcloud.sdu.dk/index.php/s/98SScJ34DK3AfMi?dir=undefined&openfile=173955322

Password: Drones4Safety

File name: D4S-WP5 – Formation flying.mp4

3.3 Bridge mock-up inspection

A model mockup of a bridge has been built to demonstrate inspection capabilities of one or more drones (Figure 5). In this demonstration, a “fleet” of two drones is instructed to execute a mission, which consists of following a pre-computed path of waypoints. During the mission execution, both drones follow a charging schedule of when to return to a central charging station, which can only charge one drone simultaneously. The battery depletion and charging are simulated (e.g., the drones are not actually charging at a charging station but land at a designated charging location to simulate this instead).

Figure 5: Wooden bridge based on Leonardo da Vinci’s design. The bridge is ~10 m long and ~2 m on its highest point. Lower image series shows the build process.

Video link (1m 42s)

https://nextcloud.sdu.dk/index.php/s/98SScJ34DK3AfMi

Password: Drones4Safety

File name: D4S-WP5 – Mockup bridge inspection.mp4

3.4 Cloud-drone collaboration

Cloud services are essential to oversee the autonomous drone swarm operation and to provide information services for continuous inspection.

This video demonstrates the interplay between the multi-drone system and the cloud infrastructure. The cloud infrastructure is exposed through the web user interface enabling the user to globally plan the inspection mission. The mission plan entails flight waypoints for the drones to reach the infrastructure to be inspected. The web interface shows a 2D map and inspection targets. The user can select the number of drones participating in the mission and desired inspection targets. The system provides near-optimal determination of visiting order and calculation of appropriate paths. When the paths are obtained, the user sends path waypoints to the drones which use point-to-point flight controllers to reach waypoints. The data flow from the cloud system to the drones is shown in a diagram in Figure 6. When the drone begins executing the mission, it reports its location back to the cloud and the user can monitor the drone's movement on the web interface. Additionally, when the drone captures an image, it reports the image to the user by sending it to the cloud system and the image is visualized on the web interface. Data flows from drones to the cloud (and through the web interface to the user) are depicted in the diagram shown in Figure 7. Databases developed in the backend of the cloud system store mission data, telemetry data, and images received from the drones.

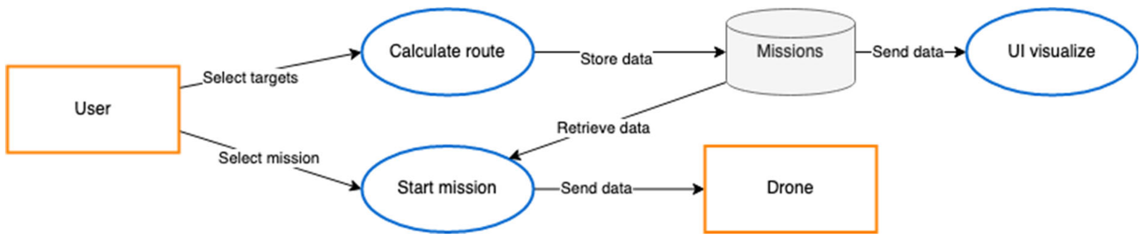

Figure 6: Data flow from cloud infrastructure to the drones.

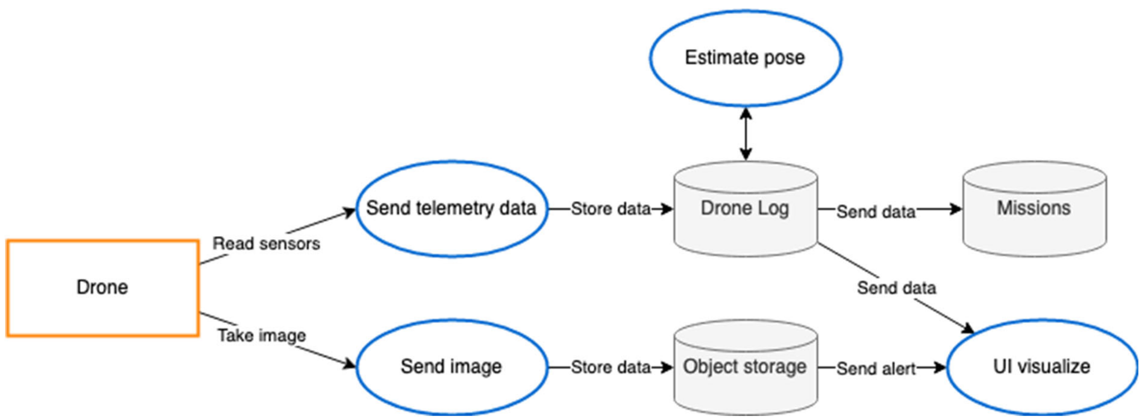

Figure 7: Reverse data flow from the drone to the cloud infrastructure.

The video demonstrates previously described data flows with a developed cloud system and a real drone in the airport scenario. Two power towers are located at the airport. Their locations are imported into the cloud system and visualized on the web user interface. In the video, we demonstrate simple route determination by selecting an airport tower and calculating the waypoints to reach it. The video shows the drone flying according to the received waypoints and reporting its location to the cloud. Location reporting is validated by monitoring the drone's movement on the web interface. The video as well demonstrates image capture and transfer to the cloud both using a simulated drone and a realistic experiment at the airport.

Video link (4m 0s)

https://nextcloud.sdu.dk/index.php/s/98SScJ34DK3AfMi?dir=undefined&openfile=173955337

Password: Drones4Safety

File name: D4S-WP5 - Cloud2drone interactions.mp4

3.5 Reactive Motion Planning

Aerial manipulation enables drones to undertake more complex tasks. For example, tasks may involve surface operation and omnidirectional physical contacts. One area of aerial manipulation focuses on tethered drones attached to fixed or moving points by a flexible link. This can be a cable to improve flight stability, a hose for spray painting and firefighting tasks, or a power cable for increased flying time.

This work investigates a motion planer framework for drones to deal with dynamic environments. It enhances the safety of the drone operation in an environment where there are human interactions.

The work in this part of the project was carried out in collaboration with Autonomous Systems Lab, ETH Zurich.

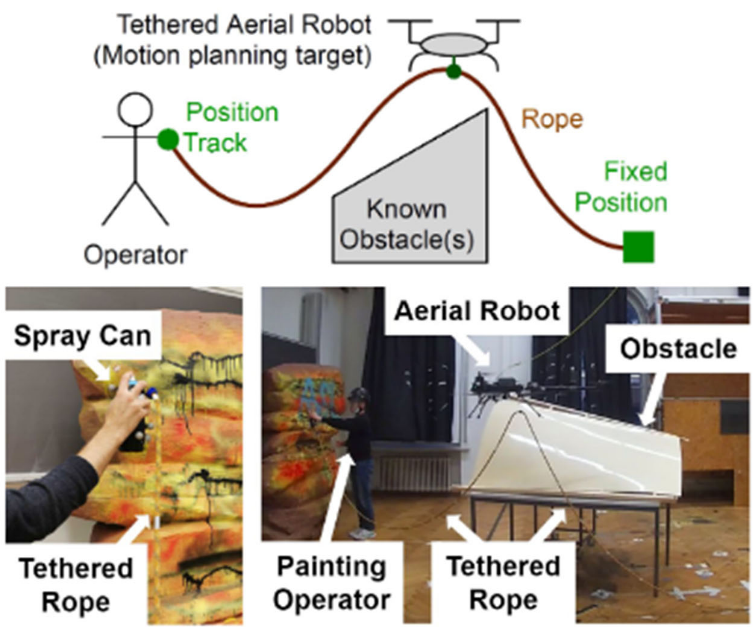

Figure 8: Top: System diagram. Bottom: An experiment where a human operator controls an end-effector (spray nozzle) that is connected to a rope (hose) while the proposed planner manipulates the rope, actively keeping the robot and the rope out of collision.

We propose to use a drone companion to aid with rope-type manipulation problems, e.g., the hose in a spray-painting task. The robot aims to keep a rope (e.g., a hose) out of collisions with cluttered environments while the end effector is controlled by a human or another robot. Figure 8 shows the system diagram, which consists of a drone, a human operator and a rope. The human operator manipulates the end-effector of the rope for a task, e.g., spray painting.

We estimate the state of the rope with force interactions from multiple touchpoints in real-time by adapting a computer graphics rope model and extending it to represent collisions with the surface mesh. We integrate the rope model with a fast motion planner based on Forced Geometric Fabrics.

We test the planner for a spray-painting task in a cluttered environment. The motion planner controls the aerial robot doing rope manipulation. The aerial robot moves the rope to avoid collisions with other objects.

Meanwhile, the aerial robot follows the human painter in a safe distance. We compare the robot's behavior with and without the rope collision avoidance policy.

The work described here is currently under review [3].

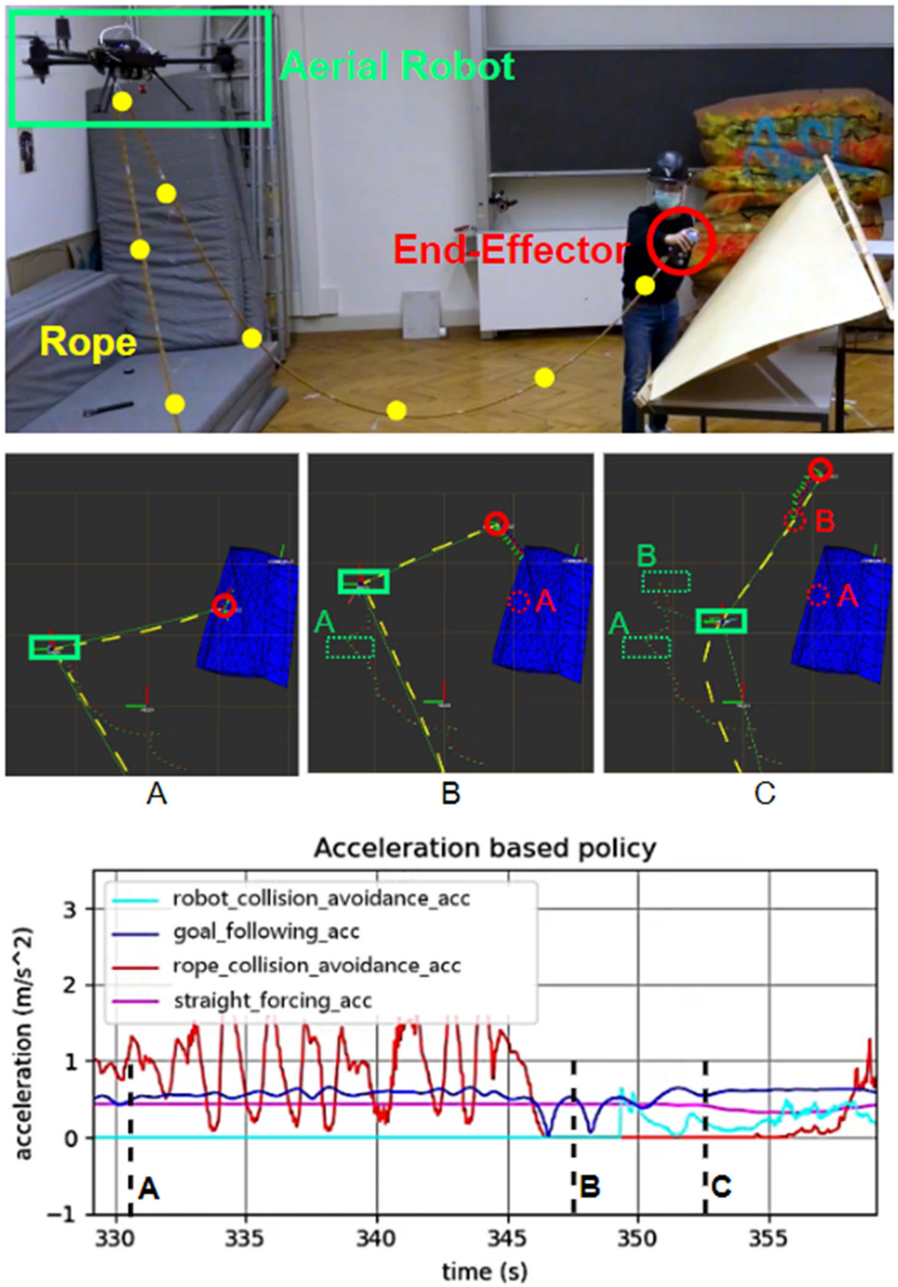

Figure 9: Experiment visualization and policy profile.

Video link (4m 4s)

https://nextcloud.sdu.dk/index.php/s/98SScJ34DK3AfMi?dir=undefined&openfile=173955325

Password: Drones4Safety

Filename: D4S-WP5 - Tethered drone.mp4

4 References

[1] Nicolaj Haarhøj Malle, Frederik Falk Nyboe, Emad Ebeid, Survey and Evaluation of Sensors for Overhead Cable Detection using UAVs, 2021 International Conference on Unmanned Aircraft Systems, ICUAS 2021, 15. Jun 2021, pp. 361-370. URL: https://portal.findresearcher.sdu.dk/en/publications/survey-and-evaluation-of-sensors-for-overhead-cable-detection-usi.

[2] Frederik Falk Nyboe*, Nicolaj Haarhøj Malle, Emad Ebeid, MPSoC4Drones: An Open Framework for ROS2, PX4, and FPGA Integration, accepted for publication. URL: https://portal.findresearcher.sdu.dk/da/publications/mpsoc4drones-an-open-framework-for-ros2-px4-and-fpga-integration.

[3] Liping Shi et al. "Reactive Motion Planning for Rope Manipulation and Collision Avoidance using Aerial Robots", under review.