CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

BACHELOR'S THESIS

Daniel Maldonado

System for Aerial Docking of Autonomous Unmanned Aerial Vehicles

Department of Cybernetics and Robotics

Thesis supervisor: **Ing. Daniel Heřt**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Maldonado** Jméno: **Daniel** Osobní číslo: **499140**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra elektromagnetického pole**

Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Systém pro dokování autonomních bezpilotních letounů za letu

Název bakalářské práce anglicky:

System for Aerial Docking of Autonomous Unmanned Aerial Vehicles

Pokyny pro vypracování:

1) Design and build a system that would allow a small multirotor unmanned aerial vehicle (drone) to land and takeoff from a larger drone during flight.
2) Build a small drone equipped with the designed docking system and use one of the existing larger drones of the MRS group as the other member of the docking system.
3) Demonstrate the functionality of the mechanism in a static experiment where the large drone stays on the ground, while the smaller drone performs takeoffs and landings.
4) Integrate the docking system into the ROS [3] based MRS UAV System [2].
5) If the schedule of the MRS group allows, test the docking system in a real-world experiment with flying drones.

Seznam doporučené literatury:

[1] B. Caruso, M. Fatakdawala, A. Patil, G. Chen and M. Wilde, "Demonstration of In-Flight Docking Between Quadcopters and Fixed-Wing UAV," 2021 IEEE Aerospace Conference (50100), Big Sky, MT, USA, (2021)
[2] Baca, T., Petrlik, M., Vrba, M., Spurny, V., Penicka, R., Hert, D., and Saska, M., "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles", J Intell Robot Syst 102, 26 (2021).
[3] Quigley, Morgan, et al. "ROS: an open-source Robot Operating System." ICRA workshop on open source software. Vol. 3. No. 3.2. 2009.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Daniel Heřt Multirobotické systémy FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **02.02.2023** Termín odevzdání bakalářské práce: ______________

Platnost zadání bakalářské práce: **22.09.2024**

Ing. Daniel Heřt podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

.

Datum převzetí zadání	Podpis studenta

Author statement for undergraduate thesis

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

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

Acknowledgements

I would like to thank my supervisor Ing. Daniel Heřt and Ing. Pavel Stoudek for their guidance and patience. I would like to thank my parents for their support throughout my studies. I would also like to thank my brother for studying with me when my motivation was low. Furthermore, I would like to thank Bc. Murat Özden and Ing. Tomáš Báča, Ph.D. for their help with ROS and experiments. Finally, I would like to thank my friend František Knobloch for helping me on my journey of self-optimization during the entire study.

Abstract

This bachelor thesis deals with the development and implementation of a system for in-flight docking of two Unmanned Aerial Vehicles. It is a robust electromechanical system with feedback, based on spacecraft docking. The experiments performed have confirmed that the chosen approach is energy-efficient, and fully functional. This thesis also considers the issue of axial deviation of the drones with respect to each other, the interference of the downwash caused by the propellers of the drones and the electromagnetic interference of the system with other systems on the drones.

Keywords: Aerial Docking, Multi-robot Systems, Docking System

Abstrakt

Tato bakalářská práce se zabývá vývojem a realizací systému pro dokování dvou bezpilotních letounů za letu. Jde o robustní elektromechanický systém se zpětnou vazbou na bázi dokování kosmických lodí. Provedené experimenty potvrdily, že zvolený přístup je energeticky nenáročný, a plně funkční. Je zde též vzána v potaz problematika osové výchilky dronů vůči sobě, interference sestupného proudění vzduchu letounů i elektromagnetické interference systému s ostatními systémy na dronech.

Keywords: Dokování za letu, Multi-robotické systémy, Dokovací systém

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

Over the past few years, the popularity of using UAVs to perform various types of tasks has grown because of their accessibility and declining prices. As mass production of drone components increases, their quality increases and the price decreases. New software is also being developed to make these systems easier to use. UAVs are also very advantageous due to their agile movement through the air while also being able to carry additional cargo beyond their basic equipment. Therefore, scientists are exploring how drones, or more precisely multi-rotor drones, could help people in their daily lives. From transportation of objects or security systems to assistance to the integrated rescue service. Based on the specific requirements, various international competitions such as DARPA Subterranean Challenge (Defense Advanced Research Project Agency) or MBZIRC (Mohamed Bin Zayed International Robotics Challenge) are held, where UAVs perform demanding tasks autonomously. The MRS group, under which this bachelor thesis has been written, not only participates in these competitions repeatedly but also takes the podium.

1.1 State of the art

One of the focuses in research of the MRS group is swarm robotics. Using multiple drones working together increases their speed and efficiency in performing a given task. [3] The amount of time that cooperating drones can spend in the air is limited by the capacity of their batteries. In a situation where a swarm of drones would have to first reach a designated location and then begin work, their batteries would already be partially depleted by the time they arrive. Therefore, various systems are being developed to allow drones to dock in mobile docking stations. For more information reading Chapter 4 in [4] is recommended. Looking at things from a different perspective, individual swarm drones could be used to refuel the main drone to keep it in the air longer. This topic has been explored more in [5]. The main objective of this work is to design a docking system (software and hardware) for the MRS group that will allow two drones to perform aerial docking.

One way of docking in the air is to use a rod that can be lowered and raised. The drone can then grab onto this bar and thereby dock from underneath the transport drone. The paper [6] describes this method more in detail. Another refreshing approach to in-flight docking is described in [7]. The authors used a fairly complex system that uses a cage with a servo-operated hinge to lock the drone to a UAV wing. The way in which the problem is approached in this thesis is a little distinct. The principle of russian spacecraft docking is used here [8]. The docking system consists of two parts. The first is the Probe that is attached to the smaller drone and is essentially a rod with a spherical tip. The second part attached to the main drone is the Drogue, which can be described as a funnel with a locking mechanism underneath. These two parts slide into each other and the Probe is then locked inside the Drogue. This simple but effective solution will make the system robust and functional as well as relatively compact.

1.2 Outline

This work will be divided into four main parts. The next chapter will deal with the hardware. It will describe the drones used, the types of sensors and actuators utilized, and the circuit board design. It will also include information about the 3D models. Chapter 3 contains more detailed information about the operation of the Docking System itself. It will provide insight into the individual components as well as the operation of the mechanism as a whole. The fourth chapter will deal with the software part. Namely, a basic description of the code, of both the MCU (Microcontroller Unit) and the ROS (Robot Operating System) node, that allows the system to be integrated into the MRS (Multi Robot Systems) platform. The last chapter will be devoted to experiments with the system.

Chapter 2

Hardware

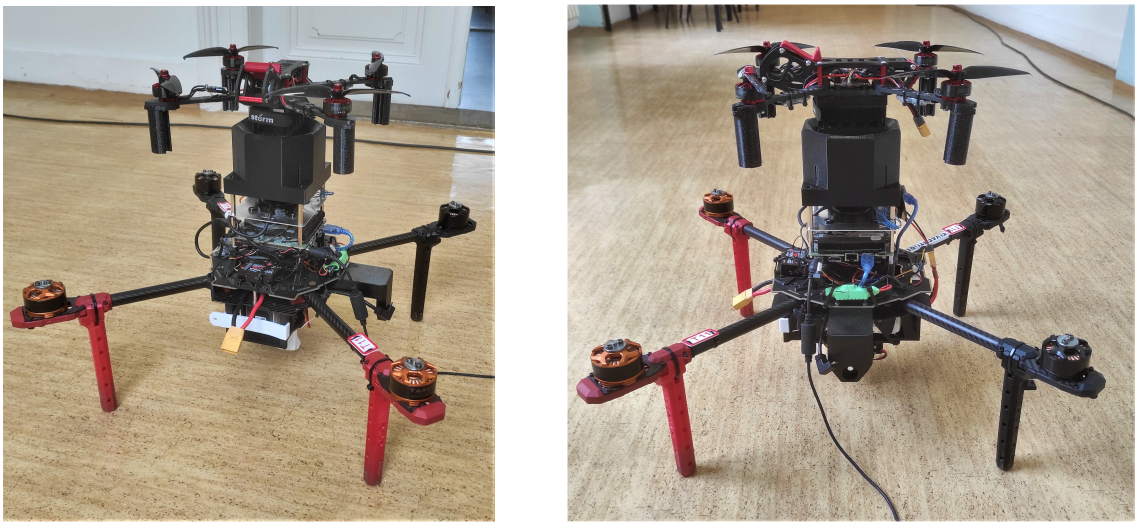
Contents

This chapter will mainly discuss the used hardware. That being, docking sensors and locking actuators. The two drones used for this project will be briefly showcased here. The transport drone will be referred to as "Mothership" and the drone that will dock as "Swarm Member". A custom PCB with an MCU has also been designed for this project. The two main parts (the Probe and the Drogue) of this system were printed on a 3D printer. The main concept is based on the Russian docking system, which was used, for example, in 1971 to dock to the Salyut 1 space station, see [8].

2.1 Used Drones

The drone used as the Mothership was a Tarot T650 provided by the MRS group. With a span of 650 mm between two diagonally mounted motors, it is the larger of the two drones as it must be able to carry the additional weight when the Swarm Member is docked. Another reason for its larger size is that it has to keep hovering steadily and withstand the downwash caused by the Swarm Member's propellers during its landing. The Mothership drone was equipped with 4114 320KV motors with Turnigy Multistar 51A ESCs (Electronic speed controllers). In addition, it was equipped with the NUC8i7BEH control computer, to which the Pixhawk 4 autopilot and a GPS module were connected. the drone and all its peripherals were powered by a LiPo 6S 8000mAh battery pack.

The used frame for the Swarm Member was the Rotorama Spectre with a diagonal motor-to-motor span of 250 mm. Attached to it were four HGLRC Aeolus 2306.5 2550Kv BLDC motors, which were driven by Skystars Talon 40A ESCs. The ESCs are connected to a PIXHAWK PX4 MINI PM06 power distribution board, that is controlled by SKYSTARS F405HD2 flight controller running the Betaflight software. Furthermore, the Swarm Member is equipped with a GPS module and a RF communication module. Figure 2.1 shows the Swarm Member drone docked on top of the Mothership drone.



(a) Front view (b) Side view

Figure 2.1: Swarm Member docked on top of the Mothership

2.2 3D Models

The main parts of the system, the probe and drogue module as well as the Swarm Member parts will be introduced here. It will also be discussed why the docking system is designed the way it is, and why it is attached in the specific locations where it is attached. A short mention will also be made of the eventually unused docking system featuring the solenoid locking mechanism.

2.2.1 Design and placement decisions

An important choice had to be made when selecting the location of the docking system on the drone. According to paper [5], Chapter IV. section A and B, the most convenient way to dock a smaller drone to a larger drone is from above. The main advantages of this approach are the minimal disruption to the airflow of both drones and the natural vertical alignment of the drones caused by the airflow.

Hence, the Probe was designed to be attached to the bottom of the Swarm Member and the Drogue to the top of the Mothership.

2.2.2 Swarm Member parts

The first few models that had to be designed, were key components for the Swarm Member drone. These were the battery compartment which is shown in figure 2.2 and the landing gear which can be seen in figure 2.3 that attach to the bottom of the Rotorama Spectre frame.

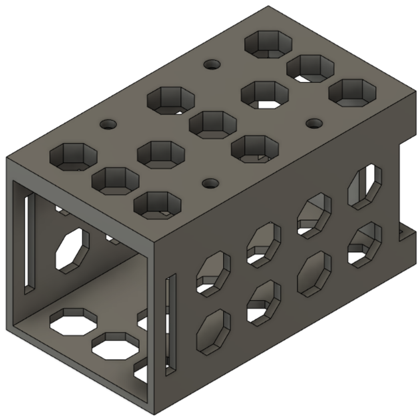


Figure 2.2: 3D model of Battery Container of the Swarm Member drone

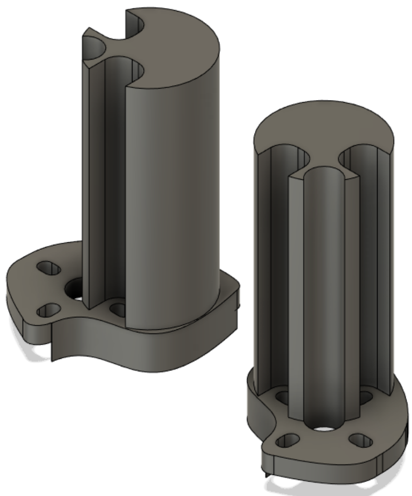


Figure 2.3: 3D model of Swarm member's landing gear

2.2.3 Probe

Originally the probe was going to be a separate part that would be mounted onto the Swarm Member, but it was determined quickly that incorporating it into the 3D model of the battery compartment would make the whole system more compact and rigid. The 3D model of the Probe can be seen in figure 2.4. The probe is essentially a cylinder that has a sphere at the end to allow it to effectively interact with the Drogue. It also features a centering cone that has the exact dimensions of the one on the Drogue, so that when the Probe is inside the Drogue, the two components are centered to one another. The last part I would like to mention is the 10mm cutout in the cylinder located just in front of the sphere. This cutout is caught by the locking mechanism inside the Drogue.

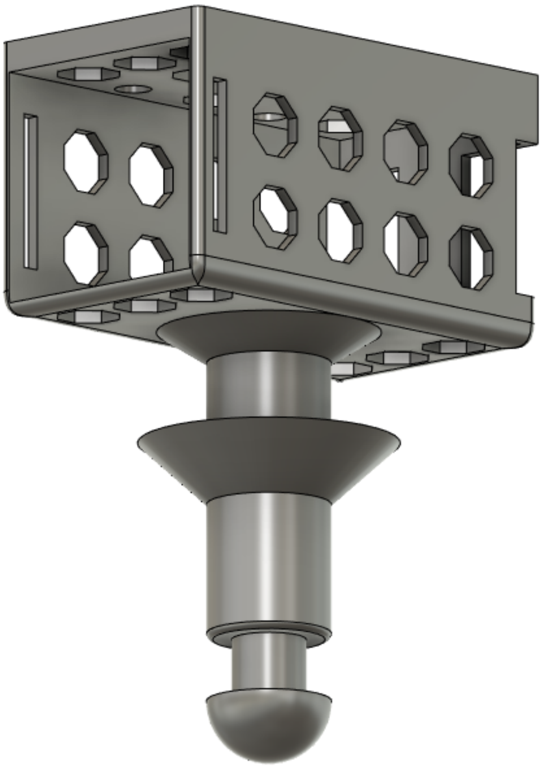
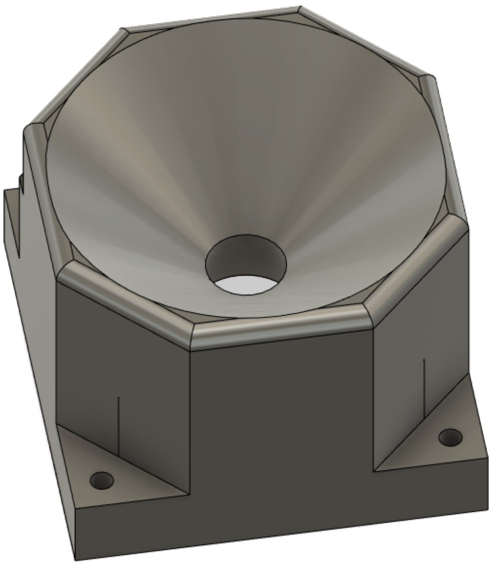


Figure 2.4: 3D Model of the docking system - Probe

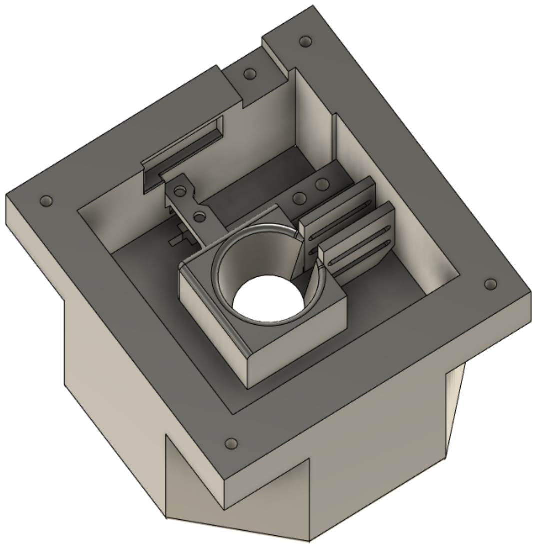
2.2.4 Drogue

In short, the Drogue can be described as a large cone that has a locking mechanism underneath. The Drogue can be seen in figure 2.5. The goal was to make the cone's platform as wide as possible relative to the cone's apex angle and the system's mounting holes to maximize the positional misalignment correction of the Swarm Member. Looking at the Drogue from below, one can see the sensor mounts where the limit switch is to be attached.

The rectangular slot located in one of the Drogue's walls is intended to house the servomotor which is then secured with two M3 bolts and a separately designed support, which is shown in figure 2.6. A custom latch is attached to the servo shaft that can be rotated 45 degrees to secure the Probe in place. It is accurately designed to integrate the original servo arm to ensure that the servo shaft does not spin within the latch freely. The mentioned latch can be seen in figure 2.7. The last part to be noticed is the rectangular section surrounding the Drogue entry hole. This block has two functions: Firstly, it supports the latch in the locked position so that any undesired movement of the Probe is minimised. Its second function is to ensure a smooth take-off. The smooth-edged cone, which surrounds the Drogue's inlet, ensures easy ejection of the Probe during Swarm Member takeoff.



(a) Drogue - Top view



(b) Drogue - Bottom view

Figure 2.5: 3D Model of the docking system - Drogue

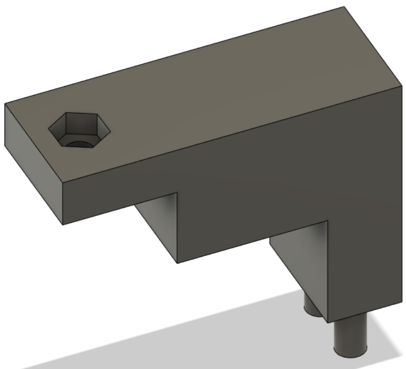


Figure 2.6: 3D Model of the servomotor support

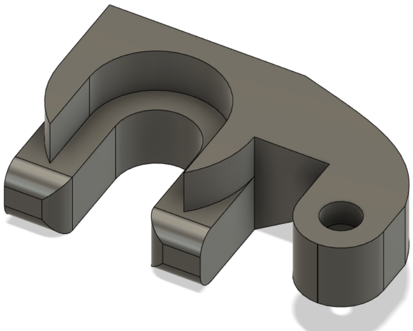


Figure 2.7: 3D Model of the locking latch

2.2.5 Unused 3D models

The system had a predecessor that used a linear solenoid to lock the probe in place. Unfortunately, it did not lock the Probe in position particularly well. The system had a simpler design than the current one, as placing a large servomotor in the ideal position to turn the latch is much more complicated than simply designing two mounts for the solenoid. The 3D model of the unused Drogue can be seen in figure 2.8.

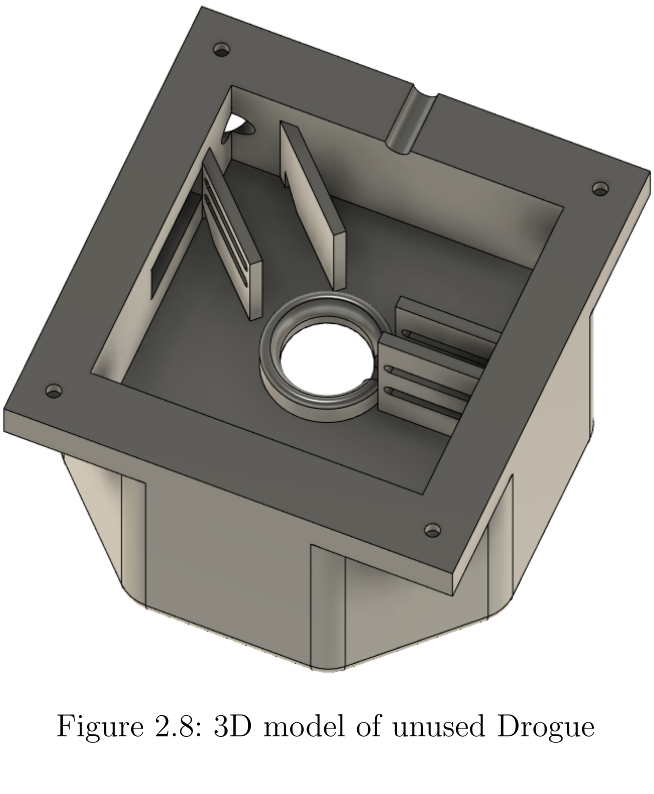


Figure 2.8: 3D model of unused Drogue

All mentioned parts were modelled in Fusion 360 by Autodesk and printed on a Prusa i3 MK3 3D printer in PET-G material.

2.3 Docking Sensors

In order for the Mothership to receive feedback on the docking status of the Swarm Member, the system must contain some type of docking sensor. A discrete docking sensor will suffice. There are many options when choosing a discrete docking sensor. One solution that comes to mind is the use of a magnet paired with a magnetic sensor such as a Hall effect sensor or a simple magnetic switch. This option was omitted due to its unnecessary complexity and to reduce the effect of the magnetic field on other parts of the drone, like the compass, which is critical for the GPS. The sensor could also be inadvertently affected by external magnetic fields, such as the magnetic field of motors. Another sensor that was considered, was an optical barrier. This design is applicable, but it is not possible to predict in which lighting conditions the system will be used. The optical sensor could be negatively affected by external lighting and return erroneous information about the Swarm Member's docking status.

2.4 Used Docking Sensor

A mechanical limit switch seems to be the optimal choice for this application. The advantages are simplicity of design, zero magnetic interference and zero standby power consumption. Specifically, the Jietong MSV-13 microswitch was used, which can be seen in figure 2.9, because it incorporates a wheel at the end of the switch lever. The idea was that this wheel would allow easy switching from both the top and bottom perpendicular to the axis of the wheel. However, when testing the fully assembled system, the switch either did not detect the presence of the Swarm Member reliably or the wheel at the end of the lever interfered too much with the Probe preventing the Swarm Member from taking off smoothly. For this reason, decisions were made to modify the switch lever by removing the wheel and bending the tip of the lever, which solved the issue. The modification can be seen in 2.10.

Figure 2.9: Used electromechanical switch [1]

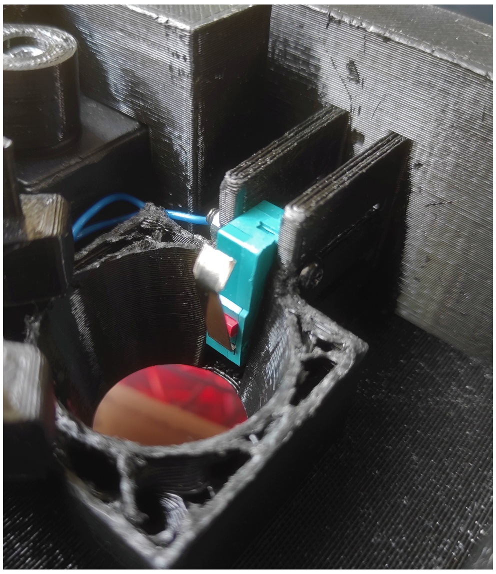


Figure 2.10: Modified electromechanical switch

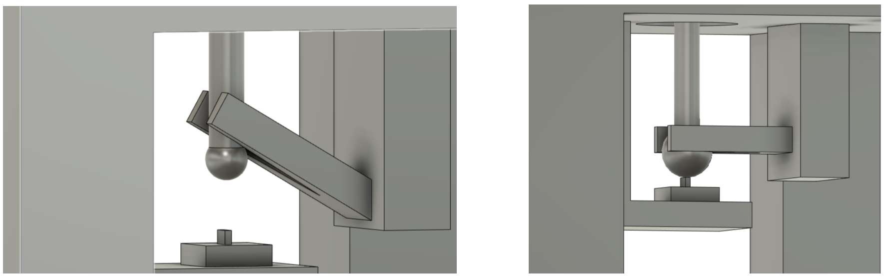
2.5 Locking mechanism

Next, a decision had to be made on how the drones would lock onto each other. Again, multiple solutions come to mind.

The first way can be to use a permanent magnet and an electromagnet. When the Swarm Member lands on the Mothership, the electromagnet is activated and the Swarm Member is pulled to the electromagnet. This solution brings with it a number of complications. One is that faster Mothership manoeuvres or worse wind conditions would risk unintentional disconnection of the Swarm Member. A second potential complication is that the switching magnetic field generated by the electromagnet could interfere with the Mothership's GPS and communications interface. Finally, the electromagnet would be in an active state for too long while carrying the Swarm Member, which would have a noticeable impact on battery consumption. If the system were to be used to carry robotic swarm technology (multiple Swarm Members at once), the consumption of all electromagnets would be significant. This problem could be solved by using electro-permanent magnets, however the negatives would still outweigh the positives and a more elegant solution can be implemented.

Another option is to use a servomotor or a stepper motor. The motor would have a specially designed hook attached to its shaft. When rotated, this hook would pull the Swarm Member's Probe down and lock it inside the Drogue section of the Mothership. This solution was more promising due to its low standby power consumption and even made it to the 3D modelling stage. The 3D model is shown in figure 2.11. However, this system was considered unnecessarily bulky, slow and complex.

The last pursued option was using a simple linear solenoid with a return spring. In this way, power consumption and EMI (Electromagnetic Interference) can be minimised as the solenoid only needs to be powered for a small period of time when the Swarm Member is docking or taking off, and can otherwise remain in a locked state. To prevent the Swarm Member from inadvertently leaving its anchored position, a special attachment had to be designed and attached to the solenoid as the return spring was not that strong. Even after this modification, the small coil did not hold the Swarm Member in the desired position, so reverting to a servomotor-actuated locking mechanism was the logical thing to do, only this time the system was designed with robustness in mind.



(a) Unlocked state (b) Locked state

Figure 2.11: 3D model of a motor-driven locking mechanism under development

2.6 Used Locking mechanism

As explained in 2.5, a servomotor was used for actuation. To be more precise, the ES 3001 servomotor was used, which can pull up to 3.2 kg with a 5 V power supply. The specific servomotor is shown in 2.12. The motor was used in combination with a specially designed locking arm which was described in the previous section 2.2.4.

Figure 2.12: Used servomotor

2.7 Control Board

In addition, it was decided that it would be useful to have a custom circuit board with an MCU (Microcontroller Unit) that drives the whole system. KiCad was used to create the circuit boards described in this section.

The PCB (Printed Circuit Board) implements an MCU together with a voltage regulator, the circuitry for UART to USB conversion, a crystal and of course several input and output pins. In particular, the ATmega328P MCU was chosen. This MCU is frequently used in the Arduino Nano. The voltage regulator is a AP64352Q buck converter, which converts 24 V from the drone's battery to 5 V. For programming purposes and serial communication with a PC, an FT23RL UART to USB converter was added.

A circuit board was made for an earlier version of the docking system with the solenoid locking mechanism, which had been left unused. The only difference between the two versions of the circuit boards is that the older version had an additional solenoid driver circuit, while the newer version only uses a digital output pin of the MCU (with the PWM option) to control the servomotor. The newer version was also designed to be more compact, as some PCB design experience was gained in the process of designing the earlier version. The schematic of the PCB is shown in figure 2.13 and the PCB itself is shown in figure 2.14.

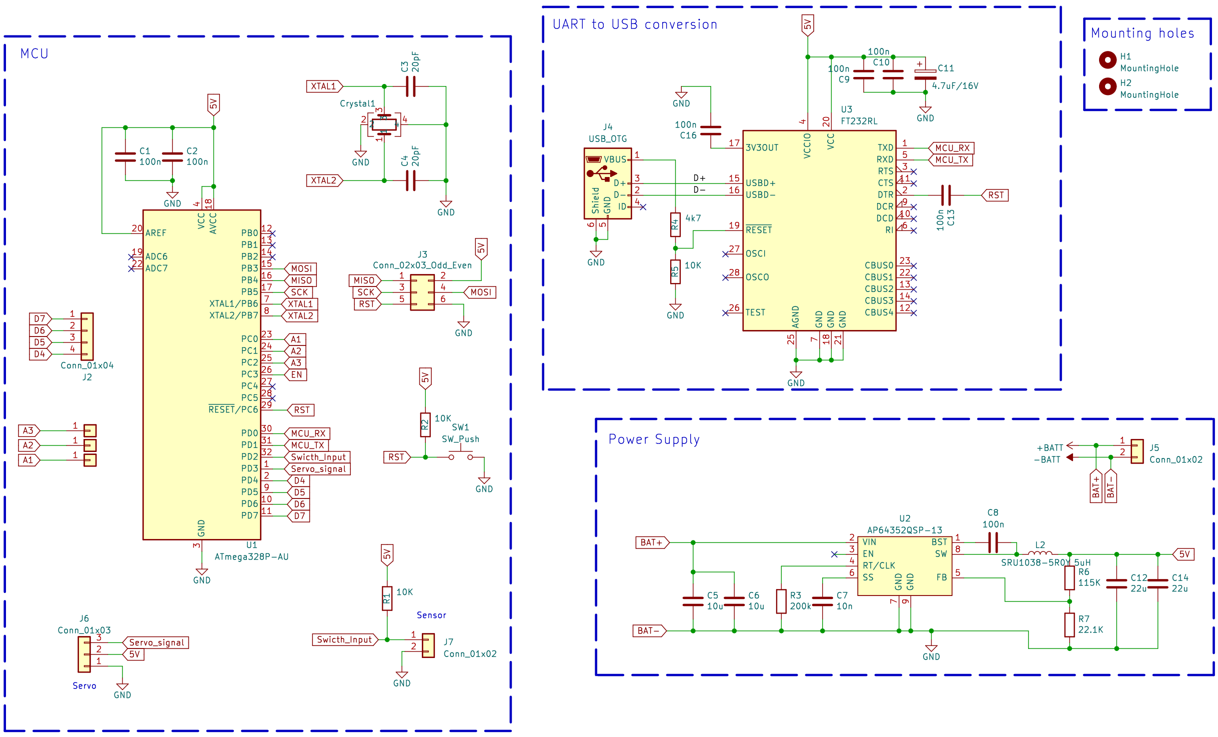
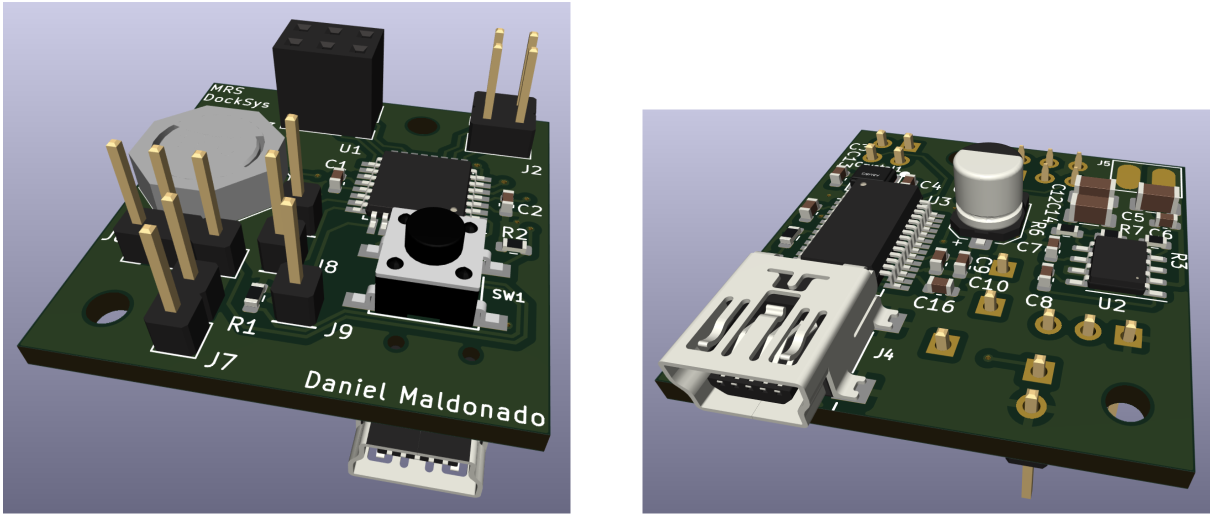


Figure 2.13: Electric schematic of the PCB



(a) PCB - Top view

(b) PCB - Bottom view

Figure 2.14: 3D model of the PCB

Chapter 3

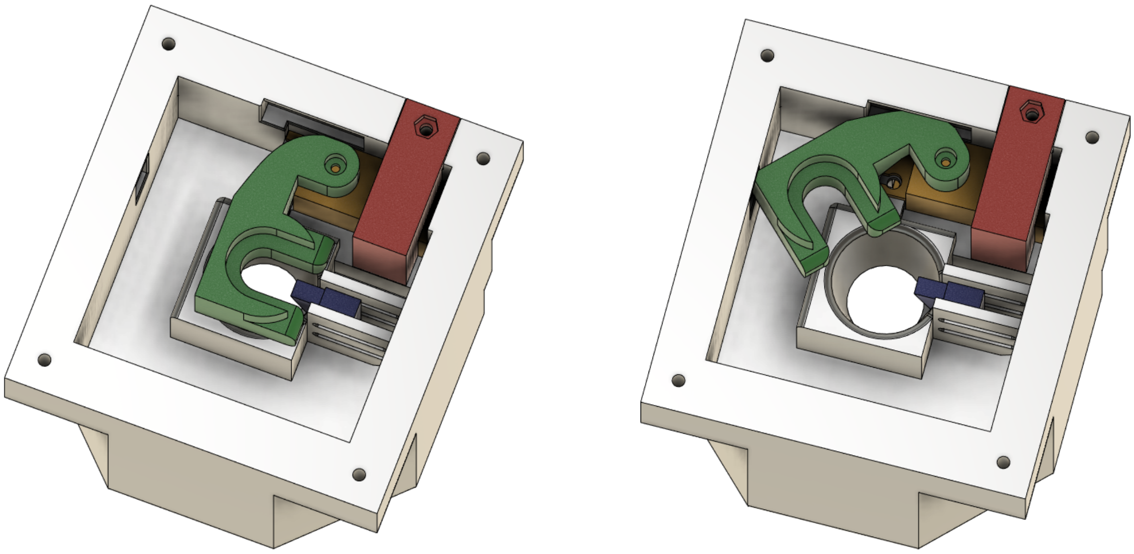
Docking System

Contents

This chapter discusses how the different parts of the docking system are connected and how they work. The sequential software logic that is used to control the peripherals of the system will also be demonstrated here.

3.1 System operation

In figure 3.1 or 3.2 one can observe the Drogue in the locked and unlocked state, which is changed by simply turning the locking latch by 45 degrees. It can also be seen how the parts mentioned in section 2.2.4 are assembled. The limit switch (in blue) is attached to its mounts by a pair of nuts and bolts of size M2. These mounts are purposely designed to allow the switch to be repositionable relative to the Drogue's inlet to ensure perfect placement. As stated in section 2.4, the switch had to be modified to avoid collisions between the switch lever and the Probe, while ensuring a triggered state when the Swarm Member is docked. The servo (in yellow) is inserted into its designated slot and secured to the Drogue using two M3 bolts and nuts. It is further held in place by the separate support (in red). This support is fixed in place by inserting its two pins into the predetermined holes on the Drogue. An M3 nut is then placed in the hexagonal cut-out and an M3 bolt is screwed into it from the opposite side. Finally, the locking latch (in green) is mounted onto the servo's shaft and fastened with an M3 bolt.



(a) Locked state (b) Unlocked state

Figure 3.1: 3D Model assembled docking system

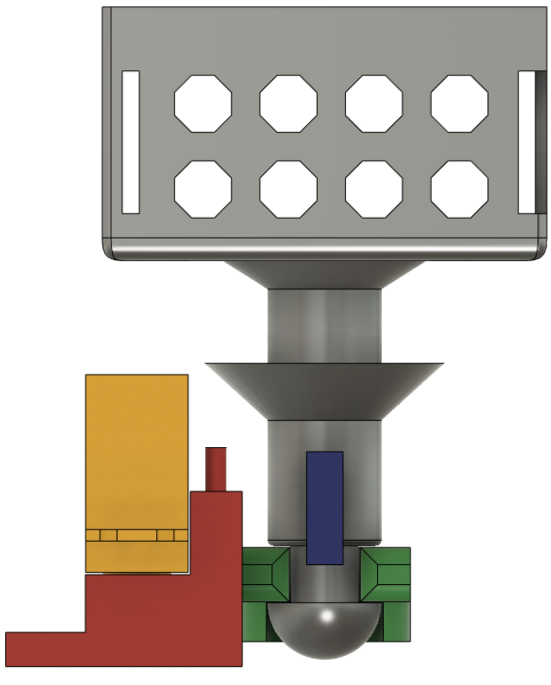


Figure 3.2: Cross section of the Drogue in the locked state

Position Feedback	Available ROS Command	Switch state	Lock state
	Expected procedure		
1. Midair	Land	Open	Unlocked
2. Landing - Stanby	-	Closing	Unlocked
3. Landing - Locking	-	Closed	Locking
4. Docked	Take off	Closed	Locked
5. Taking off - Unlocking	-	Closed	Unlocking
6. Taking off - Stanby	-	Opening	Unlocked
	Errors		
40. Switch Error - Midair	-	Closed	Unlocked
41. Switch Error - Docked	-	Open	Locked
42. Switch Error - Takeoff	-	Open	Unlocking

Table 3.1: Docking system sequential logic

3.2 Sequential software logic

The main control logic that runs concurrently with the communication protocols on the Docking System MCU is presented in table 3.1.

1. The initial state (feedback) of the Swarm Member is considered to be ”Midair” until the system receives a ”Land” command from ROS.

2. Upon receipt of this command, the system feedback is set to ”Landing - Standby” and awaits a change in the state of the limit switch. Note that after this command is obtained, no further commands will be accepted until the UAV has successfully landed.

3. When the switch is activated, the status changes to ”Landing - Locking” and a timer is initiated that counts 2000 ms before sending the command to rotate the servo to ensure that the Probe and Drogue have sufficient time for alignment. If at any time during these 2000 ms the switch is deactivated, the timer resets and the feedback is set back to ”Landing - Standby”.

4. The feedback will then change to ”Docked” once the latch is in the locked position. At this stage the system is waiting for the ”Take off” command.

5. Once this command is received, all other commands are ignored and the position feedback changes to ”Taking off - Unlocking”. The servo will again rotate the locking latch to the unlocked position. To ensure that the servo has sufficient time to do this, the same 2000 ms timer is used to delay the takeoff. When these 2000 ms have elapsed, the feedback status shall change to ”Taking off - Standby”.

6. This last state (”Taking off - Standby”) waits for the release of the switch. The instant the limit switch is released, the feedback state will return to ”Midair” and the user will regain access to the ”Land” command.

In addition to the expected feedback, other feedback states are transmitted when the limit switch outputs a bad reading or an unexpected value.

40. As shown in the 3.1 table, this error condition is transmitted when the switch is enabled during the active "Midair" state. This would mean that the Swarm Member has docked without being commanded to do so. A more likely explanation would be that the sensor itself is not functioning properly.

41. The error called "Switch Error - Docked" is the exact opposite of the error described above ("Switch Error - Midair"). More precisely, the drone is in the docked state, but the switch shows an open state.

42. As explained earlier, when the "Take off" command is issued, the servomotor will start rotating and a timer is initiated to give the servo sufficient time. If the status of the limit switch changes during this time interval, this error will be displayed in the ROS terminal.

Note that not all of the above mentioned feedback states are passed back to ROS as the MCU is meant to handle low-level operations and some states are not that relevant to the user.

Chapter 4

Software

Contents

This chapter discusses the implementation of the Docking System in the Robot Operating System (ROS). It also shows how the Low Level Communication Protocol (LLCP) can be utilized to transfer data from the MCU to ROS and vice versa. The sequential logic by which the MCU of the Docking System executes its commands was introduced in section 3.2.

4.1 ROS

ROS is an open-source environment with a large community that facilitates the creation of software for robots. The main idea behind ROS is that a single project can be composed of many nodes that can be created by different people. Nodes are pieces of code in C++ or Python that perform specific tasks, such as sensor data collection, trajectory calculations, or motor control. To establish communication between these nodes, topics or services can be used. The process of communication via topics is performed by one node publishing data on a topic and another node or multiple nodes subscribing to that topic, making the published data available to them. For example, a node that collects data from a sensor will publish such data to a topic named ”sensorOUT”. Another node that drives a servomotor may then subscribe to the topic ”sensorOUT” and work with the collected data. While communication using topics is rather a continuous flow of data, communication using services is for most cases a one-time occurrence. When a node's service is called, a data message is sent to that node followed by an acknowledge message sent back by the node. A practical example of communication via a service would be sending a command to a node that controls an LED to toggle on. The node then sends back feedback that this has happened.

To make the Docking System compatible with the MRS group hardware, a new ROS node had to be created. Sharing this node will ensure further possible development or implementation of this system by any member of the MRS group in their future projects.

4.2 ROS node description

To help understand how the implemented ROS node, figure 4.1 is shown.

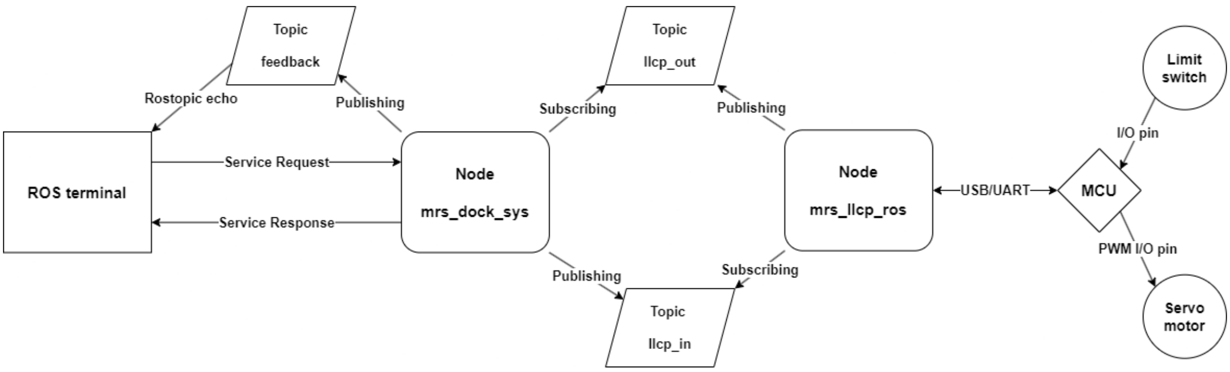


Figure 4.1: Diagram of ROS node implementation [2]

4.2.1 Issuing commands

A ROS service is used to issue commands to the docking system. The command that the MCU expects is an unsigned 8-bit integer. The format of the service message was defined in the ”DockSysCommand.srv” file. To be specific, the number two is represented as the landing command and the number four as the takeoff command.

When the command service is called by some node or from the ROS terminal, specifically using the ”rosservice call” instruction, and the correct parameter (2 or 4) is passed, the ”mrs_dock_sys” node sends this data to the ”mrs_llcp_ros” node where the data is transformed. This process will be described in more detail in the upcomming section 4.3. The transformed data is then sent via UART to the MCU, where it is processed according to section 3.2.

When the ”mrs_dock_sys” node receives a command, a boolean acknowledgement message is generated and sent back. The content of the mentioned .srv file is shown below.

```
int8 COMMAND
---
bool success
```

4.2.2 Receiving feedback

The feedback, which is periodically obtained from the Docking System, uses the publisher-subscriber communication method.

Based on the information received from the sensor and the previous feedback status, the MCU sends a message containing the current state of the Docking System via UART to the "mrs_llcp_ros" node. The structure of this message is defined in the file "dock_sys_msgs.h", which can be seen below.

```
#define DOCK_SYS_COMMAND_ID 50
#define DOCK_SYS_FEEDBACK_ID 51
#define HEARTBEAT_MSG_ID 52

struct __attribute__((__packed__)) dock_sys_command_msg
{
  uint8_t  id;  //has to have an ID
  uint8_t  command;
  /*
    2 = land
    4 = take off
  */
};
struct __attribute__((__packed__)) dock_sys_feedback_msg
{
  uint8_t  id;  //has to have an ID
  uint8_t  feedback;
  /*
    1 = airborne
    2 = landing
    3 = docked
    4 = takeoff - opening latch
    5 = takeoff - awaiting takeoff
    40 = sensor error - airborne
    41 = sensor error - docked
    42 = sensor error - takeoff
  */
};
```

```
struct __attribute__((__packed__)) heartbeat_msg
{
  uint8_t  id;
  bool     is_running;
};
```

Here it can be found that the message payload is another unsigned 8-bit integer containing one of the values from the range of 1-5 or 40-42. For more information see section 3.2. In the "mrs_llcp_ros" node the data is transformed into a ROS message format, see section 4.3. This transformed data is then published to the "feedback" topic by the "mrs_dock_sys" node. The user can then access the status of the docking system either by using the "rostopic echo" command in the ROS terminal or by subscribing their node to the "feedback" topic.

4.3 LLCP

As mentioned in the previous sections, the middleman between the "mrs_dock_sys" node and the MCU is another ROS node called "mrs_llcp_ros". This node was written by my supervisor.

4.3.1 Basic node functionality

When sending data from the "mrs_llcp_ros" node to the MCU, the user must first predefine the structure of the message. Each structure must contain an original ID along with the variables to be sent or received. Both sides of the communication must be familiar with the message structure. Then specific values are assigned to the user variables. After that, the structures are converted into a payload vector. Next, this vector is sent via UART to the MCU byte by byte, where an inverse conversion process is performed to provide the MCU with predefined structure ID and the user variables.

When the "mrs_llcp_ros" node is receiving data from the MCU, the same process takes place in the opposite direction.

4.3.2 Utilization in the Docking System

The "dock_sys_msgs.h" file contains three message structures:
The "dock_sys_command_msg" structure is a message that is published to the "llcp_in" topic and is used to transfer commands to the MCU. The "dock_sys_feedback_msg" structure is used to pass feedback data in the opposite direction, where it is then published to the "llcp_out" topic. The last structure is called "heartbeat_msg". This is again a message

that is sent from the MCU to ROS every 5 seconds to confirm the presence of the device connected to the UART bus.

When the "mrs_dock_sys" node receives a command (number), this number is assigned to the "command" variable in the "dock_sys_command_msg" structure. Since the node regularly publishes this data to the "llcp_in" topic, the "mrs_llcp_ros" node can access it (as it is subscribed to said topic) and send it to the MCU as specified in section 4.3.1.

As discussed earlier, the MCU periodically generates feedback. This feedback is assigned to the variable "feedback" in the structure "dock_sys_feedback_msg". It is then sent via UART (in the manner described above, see 4.3.1) to ROS, where it is deserialized and published to the "llcp_out" topic to which the "mrs_dock_sys" node is subscribed.

Chapter 5

Experiments

Contents

This chapter will describe the experiments performed on the Docking System to verify its functionality.

5.1 Basic functionality

The first conducted experiment consisted of placing the Mothership on the ground, issuing the land command and manually placing the Swarm Member on top of the Mothership as demonstrated in figure 5.1. Then, the takeoff command was given and the Swarm Member was removed. During this process, data published to the ROS topics "feedback" and "llcp_in" were recorded. The collected data were then processed in Matlab.

Figure 5.1: Demonstration of manual testing

5.1.1 Results

The system responded as expected. When the Probe was inserted into the Drogue, the latch closed after a delay of 2 seconds and the feedback status changed to "Docked - Standby". The Swarm Member was firmly secured in place. Pulling or twisting it could not cause it to break free. Upon sending the takeoff command, the latch opened. As soon as the Probe was removed from the Drogue, the feedback state changed to "Midair - Standby". Figure 5.2 shows the measured state of the feedback and commands over time as well as the moment of activation or deactivation of the docking sensor.

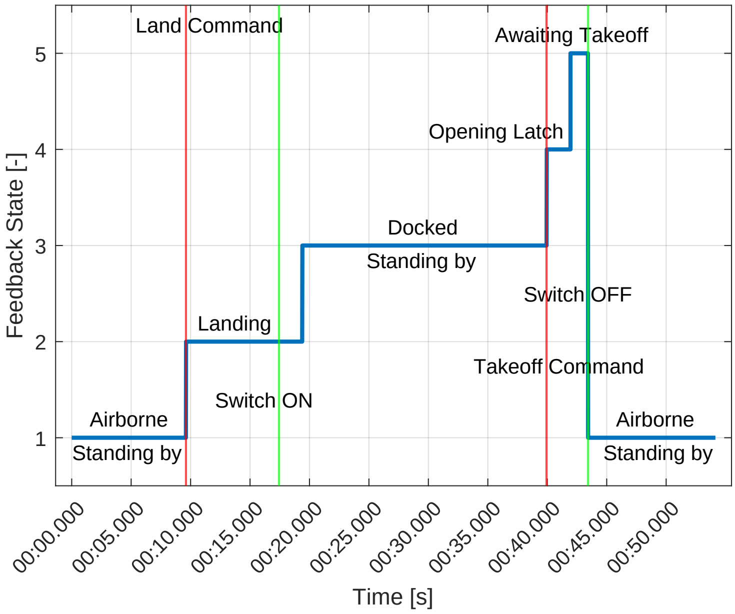
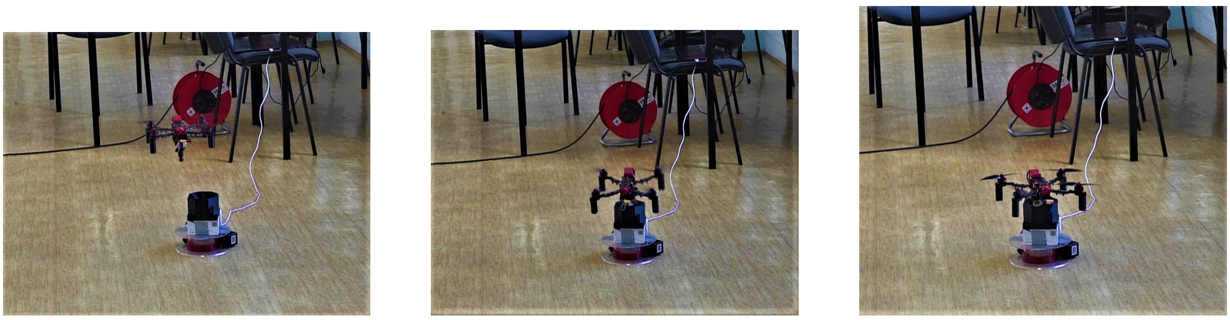


Figure 5.2: Measured time characteristics of the feedback and command status

5.2 Semi-aerial docking

In this experiment, the Drogue was placed on the ground. The land command was sent and the Swarm Member was remotely guided above the Drogue and landed on it. The takeoff process was then tested using the takeoff command and flying the Swarm Member off the Drogue and back to the ground. For a better understanding figure 5.3 is shown.



(a) Swarm Member midair (b) Swarm Member landing (c) Swarm Member docked

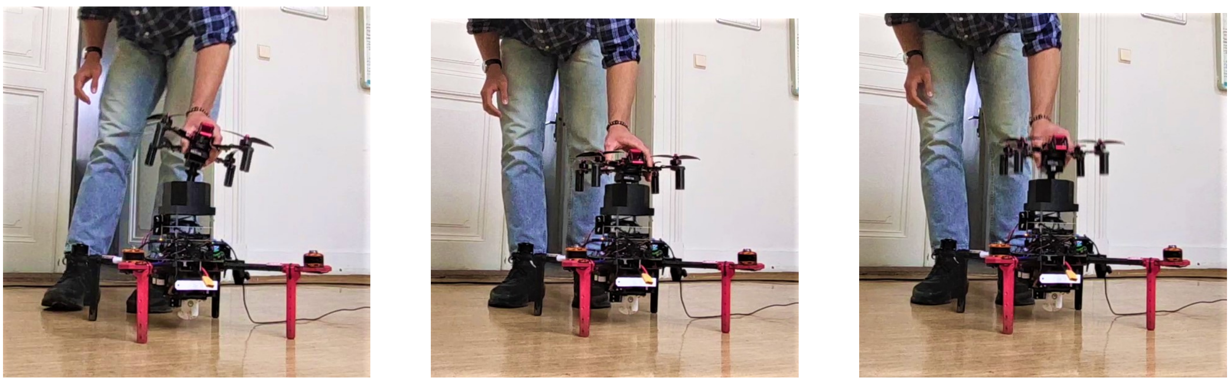
Figure 5.3: Demonstration of remote testing

5.2.1 Results

The system reacted as expected. 2 seconds after the Swarm Member landed on the Drogue, the latch closed, locking it in place in the same manner as in the previous experiment. When the takeoff command was issued, the latch opened and the Swarm Member was able to take off with no resistance. During landing and takeoff, there was no severe damage to either part of the Docking System. A video demonstration of the docking can be found in the appendix.

5.3 Error states

During this experiment, error conditions that could occur in the real world were tested. A situation could arise where the Swarm Member lands on the Drogue but leaves again before the latch is closed. If the assumptions are correct, the Drogue should remain in "Landing - Standby" mode and leave the latch open. This theory was again verified by manually placing the Swarm Member on the Drogue and then quickly removing it. This process is shown in figure 5.4.



(a) Swarm Member landing (b) Swarm Member docked (c) Swarm Member taking off

Figure 5.4: Demonstration of manual error testing

5.3.1 Results

The system reacted as expected. After inserting the Probe into the Drogue, a 2 second timer is activated, after which the latch would close and the feedback status would change to "Docked - Stand by". However, if the Probe is removed from the Drogue before this timer expires, as in this case, the timer will reset and the feedback state will remain in "Landing - Stand by".

5.4 Aerial docking

In the last experiment, the full potential of the Docking System should have been tested. The Mothership would be remotely brought to a hover state. The "land command" would be issued and then the Swarm Member would be hovered over the Mothership and docked. With the Swarm Member docked, some basic maneuvers would be performed with the Mothership. The system would then be tested for take-off in the same manner as in the previous experiments. Finally, both drones would be safely landed on the ground.

5.4.1 Results

Unfortunately, this experiment was not performed because the Mothership must be larger than the provided Tarot T650 drone due to the downwash of the Swarm Member's propellers. A larger drone was not available at the time. Based on the results of the "Semi-Aerial Docking" experiment, it can be estimated that aerial docking would be possible with this system.

Chapter 6

Conclusion

Contents

The aim of this thesis was to design and manufacture a system for aerial docking of drones for the MRS group of the Czech Technical University in Prague. The system is based on the Russian spacecraft (probe-drogue) docking system. In the course of this work, several options were considered, some were pursued and finally the best one was selected. These decisions were made during the design of the 3D printed parts and the PCB, the selection of the docking sensor and the actuator, all of which are described in detail. For the purpose of this project, a small drone (Swarm Member) was built from the ground up and an existing larger drone of the MRS (Tarot T650) was used as the Mothership. The hardware was integrated into the MRS UAV platform by creating a new ROS package. The package contains the ROS node, its firmware, a launch file and other dependencies. The mentioned ROS node utalizes the MRS LLCP package for communication with the MCU. All mentioned code as well as the MCU firmware was written in C++. Experiments have shown that this system allows a smaller drone to land on a larger, stationary drone with ease. This was verified by recording data published to active ROS topics. Figure 5.2 shows this data on a timeline.

6.1 Future work

The developed docking system is reliable, however, further adjustments can be done to increase its reliability and extend its range of applications.

The first adjustment that could be made is to establish real-time communication between the Swarm Member and the Mothership. This would require adding a microcomputer, such as a Raspberry Pi, to the Swarm Member. This modification would make the system more robust as the only current form of communication between the two drones is the docking sensor.

Establishing communication allows the Mothership to transmit its current position, velocity and heading to the Swarm Member. The Swarm Member could use this information for feed-foward control and autonomously land on the Mothership.

Another way to make the docking work autonomously is by using AprilTags. Paper [9] contains more information on relative positioning of UAVs using AprilTags. The AprilTag would be fixed to the top of the Mothership next to the Drogue. The Swarm Member, augmented with a camera and a microcomputer, could then land on the Mothership using the visual servoing method.

Another modification that comes to mind is making the system more compact, as more drones are to be attached to the Mothership at once. While experimenting with smaller actuators or other locking mechanisms in general could reduce the overall size, the pitfall of miniaturization is that the Drogue's cone must still remain similarly sized if it is to compensate for axial misalignment of the two drones.

An alternative to the previous idea would be to modify the Docking System so that the Mothership could remove the Swarm Member from the Drogue after docking and move it to another location on the drone. This would mean that multiple drones would only require a single Drogue to dock on the Mothership. This could be achieved by creating a special sealable cutout in the Drogue's cone that would seal during landing and takeoff. When a Swarm Member lands, the cutout would open the Swarm Member's Probe would be captured from below the Docking System by a specially designed belt that would rotate around the Mothership, allowing more Swarm Members to be housed.

Another thought that comes to mind is to incorporate a battery charging system that connects the Swarm Member's battery to the Mothership's battery once it docks. Depending on the specific situation, it would be possible to decide which battery is to be charged and which is to be discharged. For this task the above mentioned paper [5] could be used as inspiration.

Bibliography

[1] "Micro switch msw-13," www.jietongswitch.com. [Online]. Available: https://www.jietongswitch.com/micro-switch-msw-13-products/

[2] D. Maldonado, "Docking system software block diagram," https://app.diagrams.net/. [Online]. Available: https://app.diagrams.net/

[3] Y. Zhou, B. Rao, and W. Wang, "Uav swarm intelligence: Recent advances and future trends," *IEEE Access*, vol. 8, pp. 183 856–183 878, 2020.

[4] C. G. Grlj, N. Krznar, and M. Pranjić, "A decade of uav docking stations: A brief overview of mobile and fixed landing platforms," *Drones*, vol. 6, no. 1, 2022. [Online]. Available: https://www.mdpi.com/2504-446X/6/1/17

[5] K. P. Jain and M. W. Mueller, "Flying batteries: In-flight battery switching to increase multirotor flight time," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3510–3516.

[6] R. Miyazaki, R. Jiang, H. Paul, K. Ono, and K. Shimonomura, "Airborne docking for multi-rotor aerial manipulations," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4708–4714.

[7] B. Caruso, M. Fatakdawala, A. Patil, G. Chen, and M. Wilde, "Demonstration of in-flight docking between quadcopters and fixed-wing uav," in *2021 IEEE Aerospace Conference (50100)*, 2021, pp. 1–9.

[8] M. Cislaghi and C. Santini, "The russian docking system and the automated transfer vehicle: A safe integrated concept," in *Proc. 3rd International Association for the Advancement of Space Safety*, 2008, pp. 1–42. [Online]. Available: https://web.archive.org/web/20130203014131/http://www.congrex.nl/08a11/presentations/day1_S02/S02_03_Cislaghi.pdf

[9] B. Zhao, Z. Li, J. Jiang, and X. Zhao, "Relative localization for uavs based on apriltags," in *2020 Chinese Control And Decision Conference (CCDC)*, 2020, pp. 444–449.

Appendices

CD Content

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

Directory name	Description
thesis	the thesis in pdf format
thesis_sources	latex source codes
ROS_code	ROS package
MCU_code	latex source codes
videos	video demonstrations
fotos	pictures

Table 1: CD Content

List of abbreviations

In Table 2 are listed abbreviations used in this thesis.

Abbreviation	Meaning
EMI	electromagnetic interference
ESC	electronic speed controllers
GPS	global positioning system
LED	light emitting diode
LLCP	low level communication protocol
MCU	microcontroller unit
MRS	multirobot systems
PCB	printed circuit board
RF	radio frequency
ROS	robot operating system
UART	universal asynchronous receiver/transmitter
UAV	unmanned aerial vehicle

Table 2: Lists of abbreviations