University of Tartu

Faculty of Science and Technology

Institute of Technology

Cathy Toomast

ESTCube-2 Mission Control System: Preparation for In-orbit Operation

Master Thesis (30 EAP)

Robotics and Computer Engineering

Supervisors:

M.Sc.Eng. Kristo Allaje

PhD. Indrek Sünter

Tartu 2023

ESTCube-2 Mission Control System: Preparation for In-orbit Operation

Abstract

Space missions rely on the mission control system (MCS) for spacecraft health monitoring, commissioning, routine operations, and emergency procedures. However, often the MCS stays in the shadow of the satellite itself, creating a situation where the satellite is launch-ready, but the system necessary to operate the satellite from the ground is not. This is also the case with ESTCube-2, a satellite that started development in 2016 and is soon ready for launch. ESTCube-2 is a spacecraft mainly developed by student volunteers, and its main mission is to demonstrate deorbiting with plasma brake technology [1]. To operate the spacecraft, there is a need for a functional mission control system. The MCS has been developed through the years, but when it was initially built, the satellite was not ready to be tested with the system. For this reason, the system is not yet widely used and has unresolved issues.

The author of the thesis investigates and lists the actions that need to be taken to have an operational MCS by the start of the mission. Furthermore, to understand the needs of mission operating systems, the author used a qualitative research method, interviewing four people with experience with operating satellites.

For the final system to be useful for the mission, the author made changes in the mission control system during the thesis and implemented suggestions from interviews with spacecraft operators. Initial tests on the ground were performed with the ESTCube-2 engineering model. Additionally, the author will list ideas and notes regarding what could be done better in the future at the end of the thesis.

CERCS: T120 Systems engineering, computer technology; T320 Space technology; P160 Operation research, programming

Keywords: Space technology, ESTCube-2, Mission Control System, thematic analysis, spacecraft operations, web software development

ESTCube-2 missioonijuhtimissüsteem: ettevalmistus operatsiooniks orbiidil

Lühikokkuvõte

Kosmosemissioonid tuginevad missioonijuhtimissüsteemile kosmoselaevade töö jälgimiseks, seadistamiseks, rutiinseks toimimiseks ja hädaabiprotseduuride jaoks. Kuid sageli jääb missioonijuhtimissüsteemi arendus satelliidi enda arenduse varju, luues olukorra, kus satelliit on stardivalmis, kuid maapinnalt satelliidi juhtimise süsteem veel mitte. Nii on ka 2016. aastal alustatud ESTCube-2 satelliidi puhul, mis on peagi stardiks valmis. ESTCube-2 on satelliidiprojekt, mida arendavad peamiselt vabatahtlikud tudengid ja mille põhiülesandeks on demonstreerida plasmapiduritehnoloogia abil satelliidi orbiidi langetamist [1]. Kosmoselaeva juhtimiseks on oluline funktsionaalne missioonijuhtimissüsteem. Missiooni juhtimissüsteemi on aastate jooksul arendatud, kuid kui see algselt ehitati, polnud satelliit ise testimiseks piisavalt valmis. Sel põhjusel pole süsteemi veel laialdaselt kasutatud ja see sisaldab vigu.

Töö autor uurib ja loetleb toimingud mida tuleb teha, et missiooni juhtimissüsteem oleks missiooni alguseks töökorras. Lisaks kasutas autor missioonijuhtimistarkvara vajaduste mõistmiseks kvalitatiivset uurimismeetodit, intervjueerides nelja inimest, kellel on satelliitide opereerimise kogemus.

Selleks, et lõplik süsteem oleks missiooni jaoks kasulik, tegi autor lõputöö käigus muudatusi missiooni juhtimissüsteemis ning rakendas kosmoselaevade operaatoritega tehtud intervjuudest saadud soovitusi. Esialgsed katsetused maa peal viidi läbi ESTCube-2 insenerimudeli peal. Lisaks loetleb autor lõputöö lõpus ideid ja märkmeid selle kohta, mida saaks tulevikus paremini teha.

CERCS: T120 Süsteemitehnoloogia, arvutitehnoloogia; T320 Kosmosetehnoloogia; P160 Operatsioonanalüüs, programmeerimine

Märksõnad: Kosmosetehnoloogia, ESTCube-2, missioonijuhtimissüsteem, temaatiline analüüs, kosmoselaeva opereerimine, veebitarkvara arendamine

Contents

List of Figures

List of Tables

Acronyms

AOS Acquisition of Signal (or Satellite). 27

AX.25 Amateur X.25. 19

CRC Cyclic Redundancy Check. 20

FITS Flexible Image Transport System. 30

LEO Low Earth Orbit. 27

LEOP Launch and Early Orbit Phase. 48

LOS Loss of Signal (or Satellite). 27

MCS Mission Control System. 2, 6, 10, 11, 15–17, 19, 21, 28, 30, 31, 33, 35–37, 39, 41–50

NORAD North American Aerospace Defense Command. 17

SEO Search Engine Optimization. 39, 43

UI User Interface. 19, 39

UUID Universally Unique IDentifier. 19, 20, 35

UX User Experience. 42

1 Introduction

There is no doubt that space is an exciting topic for many people. For thousands of years, people have looked up to the sky, trying to make sense of what lies beyond. They have grouped stars into constellations and created myths about their origin. People have spent countless hours looking for guidance from the heavens by trying to decipher the movements and positions of celestial bodies in space and used that knowledge in various fields like seafaring and even farming.

Due to technological advancements, humans could finally start exploring space beyond our Pale Blue Dot. Even today, space is difficult to explore for humans. The five main hazards for space explorations are: radiation, isolation and confinement, distance from Earth, gravity, and hostile environments [2]. That is why we send unmanned spacecraft and robots to do most of the exploring for us.

Space technology is a relatively new field, even though space has always existed - reaching it has not been easy. The first spacecraft was sent to space in 1957. It was a Soviet Union-developed spacecraft called Sputnik 1, which began the space age. After that achievement, the space technology field started rapidly improving. [3]

Shortly after the first satellite, the first human was sent to space in 1961. The Soviet Union achieved it; the first man in space was Yuri Gagarin. He did his 108-minute orbital flight in the Vostok-1 spacecraft. Only eight years later, USA's specialists managed to get humans even further. In 1969, the Apollo 11 crew succeeded in landing on the Moon. [4, 5]

As one can notice from these big missions, historically, exploring space has been the realm of the most wealthy countries. Mainly because of the cost of the projects, which could go into the millions, if not billions. Since then, space technology has become more affordable for others as well. According to 2022 data, 75 countries have at least one spacecraft in orbit [6].

Space systems can be divided into three segments: the space segment, the transfer segment, and the ground segment. The space segment contains three main elements - payload, orbit,

and spacecraft. The transfer segment involves the launcher, which takes the spacecraft with its payload to orbit. The ground segment consists of mission operations and a ground station network. [7]

Mission operations from the ground segment are a crucial part of the mission. Mission operations have many duties, including managing flight tasks, remote maintenance of the spacecraft, analyzing the status and trends in the mission, working through procedures, generating command sequences, and conducting flight maneuvers. Operations activities vary depending on the phase of the mission. The most challenging phase for the operations team is the launch and early orbit phase (LEOP), when the spacecraft has been activated after launch, and initial tasks need to be done. The following phases, in-orbit testing, and routine operations are less demanding. [7]

Estonia launched its first satellite, ESTCube-1, in 2013 [8] and launched two more CubeSats called Koit and Hämarik in 2019, and 2020 [9]. The fourth satellite, ESTCube-2, is in development and should launch in the summer of 2023.

ESTCube-2 main mission is to test E-sail and plasma brake technologies. Other onboard experiments include a deep-space nano-spacecraft platform, two cameras for Earth observation, and corrosion testing in space. The satellite is mainly developed by volunteer students. For the mission operations in ESTCube-2, the team has created an in-house Mission Control System (MCS).

MCS is an essential subsystem in ESTCube-2, but sadly it has not kept up with the time and needs a lot of work to become ready for operational use. Testing and improving the existing MCS would help the satellite team greatly. Before the launch, the system would be helpful for the team with testing the satellite, and after the launch, it is needed to operate the satellite remotely.

1.1 Thesis Structure

This master thesis consists of 8 main chapters. In the first chapter, the author lists prior work, explains the motivation for the research, and poses a few research questions for analysis. The second chapter describes spacecraft communications and provides an overview of ESTCube-1 and ESTCube-2 mission operations. It should give the reader an understanding of how spacecraft operations work and the current state of the ESTCube-2 mission control system. The third chapter discusses the research methodology for interviews with space mission operators and

the analysis of the results. In the fourth chapter, the author creates themes to help answer the research questions and explains them thoroughly by using the results from the analysis of the interviews. The fifth paragraph provides an overview of the practical work done as part of the thesis. In the sixth chapter, the author explains the testing done with the system. The seventh chapter is about the potential future work that the author recommends, the limitations of the current design, and potential future work and improvements. The last paragraph concludes the work done and answers the research questions created at the beginning of this thesis.

1.2 Objectives

In this thesis, the experience of several space mission operators will be gathered and analyzed to gain insight into what the MCS needs and how the mission operations for ESTCube-2 might look. Additionally, the current state of the ESTCube-2 MCS will be analyzed and MCS will be tested to see what needs to be improved to make the system ready for operational use.

There will also be some software changes to prepare better the MCS for in-orbit operations and, later, some simple integration testing with the engineering model.

Finally, the author will point out recommendations for the system from the analysis and give ideas on the following steps to improve the system further.

1.3 Research Questions

The main question researched in this thesis is the following:

- **How to improve ESTCube-2 Mission Control System based on real-life experiences in satellite operations?**

ESTCube-2 satellite is close to being launched, but the mission control system is yet to be finalized. This question is meant to map out the requirements needed to operate the spacecraft successfully. Also, this fundamental question helps to prepare the MCS for the launch.

Communicating with a satellite can be quite complicated and full of surprises. By incorporating the lessons learned from other missions and spacecraft operations, some surprises and pitfalls could be avoided for ESTCube-2. Based on interviews with operators from other space missions, we could obtain ideas to improve the current MCS and how it would be used for mission operations.

The following questions were devised to guide the interviews:

- **Q1 What does a usual orbital pass look like for the team involved?**

From the perspective of the ground station, the spacecraft flies across the sky as it orbits the Earth. While the spacecraft could be visible during different passes for a different amount of time, there could be a standard set of preparations and operations performed for each pass. This question aims to get an overview of the preparations needed before and after each pass and the typical operations performed during each pass.

The scope of this question extends from the mission's commissioning phase to automating the operations when the satellite is more stable.

- **Q2 What kind of data management would a mission control system need to do?**

Data management can cover various topics, for example, how much data is collected, analyzed, and stored, how data would be analyzed and stored, how the data volume differs across telemetry types (beacon, housekeeping data, payload measurements), how it varies over time, and so on. All these may be very different from mission to mission.

- **Q3 Which issues have happened?**

This question will cover the issues that have happened during space missions. The question will help to prepare the ESTCube-2 team for common problems that may arise during spacecraft operations. Also, to get some idea on which actions to take when facing obstacles.

- **Q4 What are some good things to keep in mind while developing the system?**

The last question is broader and allows the interviewee to add further ideas or recall some interesting details that previous questions might not have covered. This question can provide valuable insight for making the system ready by the start of operations.

2 Related Work

The chapter describes typical spacecraft operations and provides examples for ESTCube-1 and ESTCube-2.

2.1 Spacecraft communication and operations

Communication is the exchange of messages and information. In space missions, it can be elaborated on and said that it is the exchange of telecommands and telemetry between the spacecraft and ground controllers and the processing and distribution of payload data to users. [10]

Communications architecture for space missions has four elements: the spacecraft, ground stations, control center, and if needed, relay satellites. Spacecraft is the element in space. Ground stations are the antennas, receivers, and transmitters on Earth. The operators control the spacecraft from the control center. The relay satellites link the primary spacecraft with ground stations using relay satellites. In the ESTCube-2 mission, the communications architecture does not use relay satellites. Information is exchanged through radio down- and uplink. Downlink refers to data being sent from the spacecraft to the ground station, and uplink refers to the ground station sending data to the spacecraft. [10]

The downlinked data is called telemetry. The telemetry from a spacecraft usually contains information about its health, the overall status and operational mode of the spacecraft and its main components, as well as any measurements or data from the payloads. The downlinked telemetry is typically passed to the control center through a communication network and contains spacecraft health information, its status, and payloads. This data flow is illustrated in Figure **2.1**. [10]

Starting from the launch phase, the operators send commands to the spacecraft so that the spacecraft takes some specific action or updates the software. There are two types of commands: real-time commands that are run immediately and stored commands that will be run in the

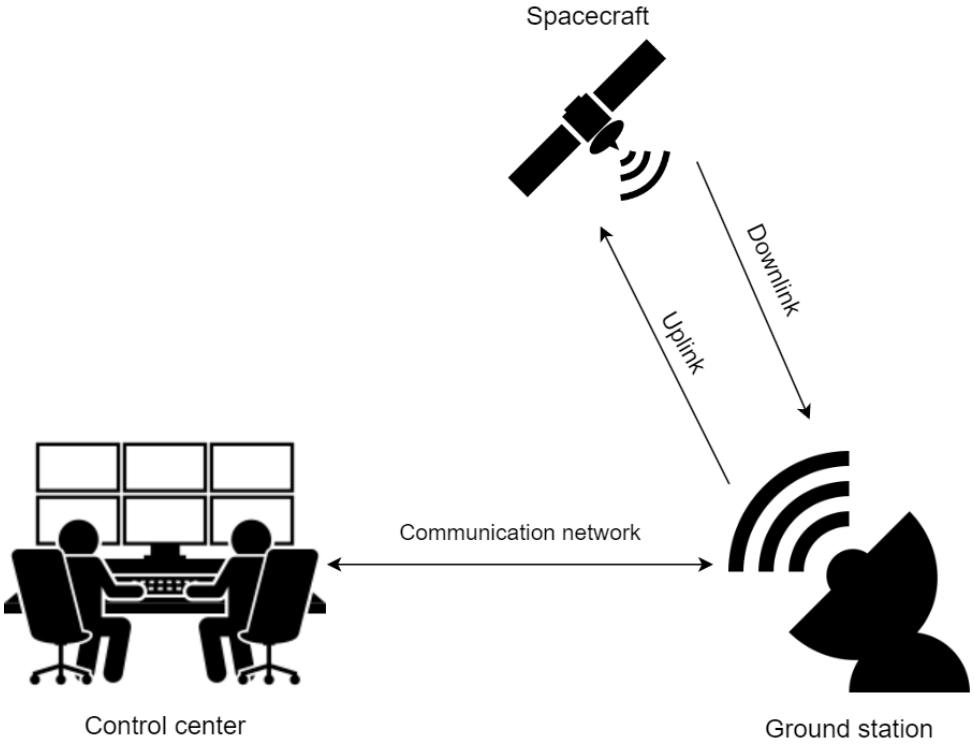

Figure 2.1: Spacecraft communication

future. Crucial commands are sent to the spacecraft in multiple steps - the command is sent, then it is verified that the command was received correctly, only then is the command executed, and eventually, the results of the command are verified. This ensures that the command is received and executed as intended. [10]

In the mission control center, telemetry is observed in real-time, meaning that the operators react to events as fast as possible. The ground station antennas have to point toward the spacecraft whenever it is visible so it is possible to exchange telecommands and telemetry. It is possible to monitor the location of a spacecraft in orbit by relying on worldwide tracking networks. [7, 10]

To minimize the effort of maintaining a large amount of data from spacecraft, it is vital to define and document the data structure for the database before the launch. Usually, there are multiple databases, each holding some of the information related to the mission. Separate databases can be for raw packets received, raw packets transmitted, telecommands, telemetry responses, and housekeeping parameters. The database structures are influenced by the mission, the design of the spacecraft, and also by the design of the control system itself. [11]

2.2 ESTCube-1 mission operations

To understand how operations were done in the ESTCube-1 mission, the author analyzed screen capture videos from operations done in 2013. For operations, a subsystem control software called ICPTerminal was used. For example, the view in this figure could be seen while performing firmware updates **2.2**.

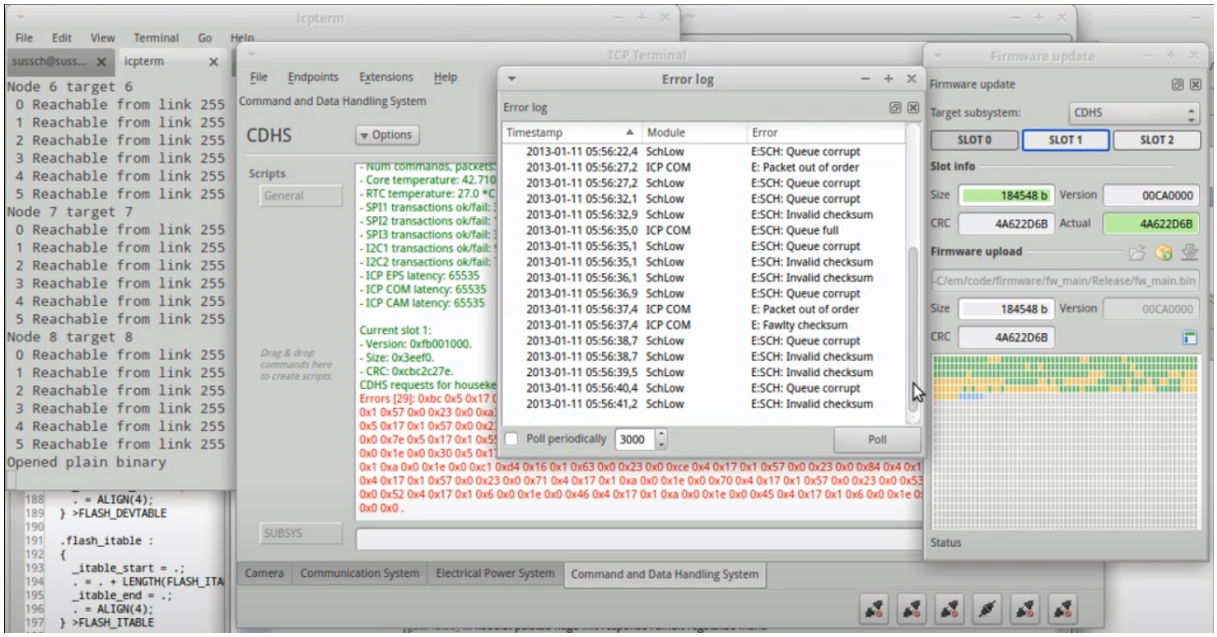

Figure 2.2: Remote firmware update for ESTCube-1 Command and Data Handling System, screen capture [12]

There was also a web-based mission control system developed for ESTCube-1, but it did not provide the main interface for mission operations. The ESTCube-1 MCS connected the ground stations, announced satellite passes, and offered other supportive features like displaying the telecommands transmitted and received telemetry. [12]

2.3 ESTCube-2 mission operations

In this section, the author briefly describes the current state of the ESTCube-2 Mission Control System. The information given here has been found from the ESTCube-2 MCS documentation, and from testing the system. Due to the ESTCube-2 mission being mostly developed by volunteer students, the people have changed and there have not been dedicated team leads to coordinate the developments of this system.

Since the start of the ESTCube-2 development, the flexibility of spacecraft operations has been kept in mind. More specifically, the MCS was designed to support multiple ground stations as well as to enable the participation of the radio amateur community through the MCS. A part of the MCS would be accessible to the general public, while the rest would be restricted to the team of spacecraft operators. The overview of ESTCube-2 communications can be seen in figure **2.3**.

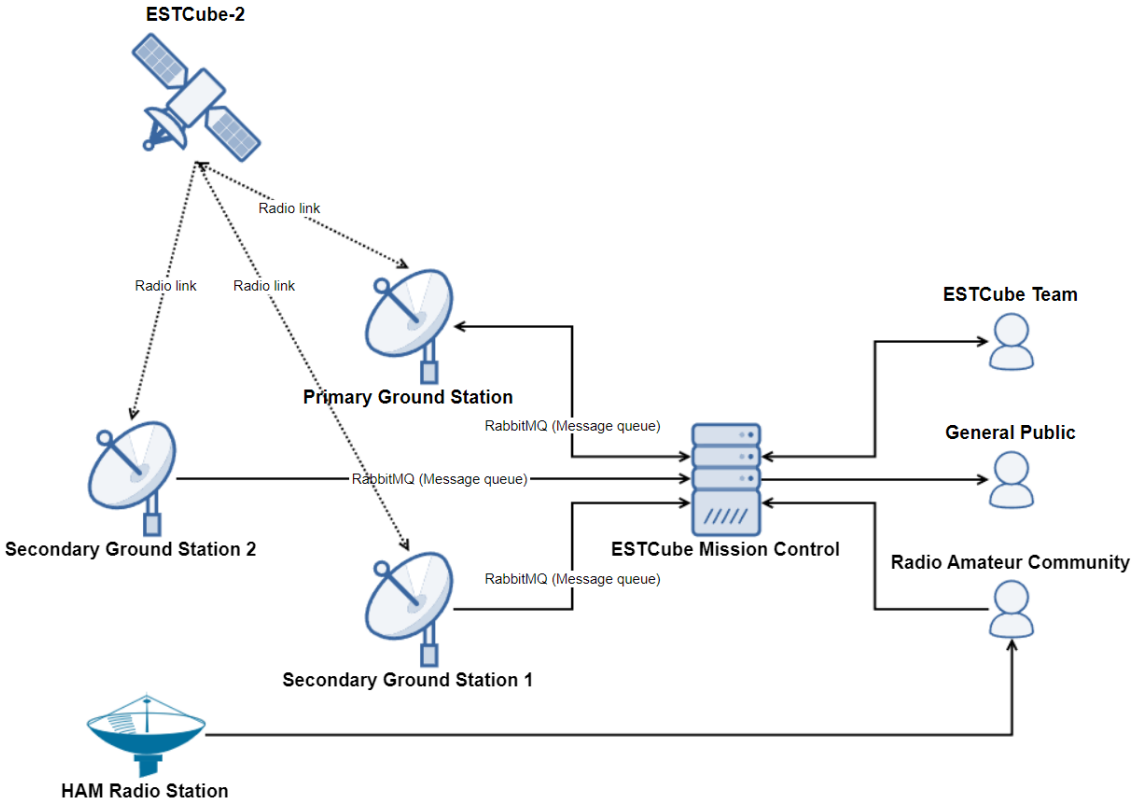

Figure 2.3: ESTCube-2 communications overview [13]

2.3.1 Mission Control System

MCS has been developed mostly by volunteer students since the decommissioning of the ESTCube-1 in 2015. Since that time, the system has undergone various reworks and changes.

Architecture

The ESTCube-2 MCS is modular and consists of at least 20 components, microservices. It was designed to work with big data while also being flexible.

The 2021 state of the components in the system is shown in Figure **2.4**. The author received explanations for the component states from the missions' team members, which are as follows. Components with yellow color (Groundstation Router, RabbitMQ, Adminer, Contact Automation, Satellite Location Predictor, Scripting Engine, Graylog) existed and some development was done but were not in development and use. Green color components (Telemetry Client, Authorisation, Authentication, MCS Codec, AX.25 Codec, MCS Front-end, MCS Back-end, Telemetry Reception, UUID Generator) were in development and use. A repository was made for the orange component (MCS Scripts), but no developments were made. The system has four databases (Terminal Table, AX.25 Table, ICP Table, Command Table) in blue color. The last is a red component (File Tracking) that had issues and was under construction.

The functional components currently in use are:

- **MCS Back-End** coordinates other components, used mainly as a relay. Ensures that packets with the correct parameters are forwarded while logging every step.
- **MCS Front-end** displays the user interface. Its state will be broken down in Section 2.3.1.
- **Command table** lists and describes the telecommands, their structure, and which spacecraft subsystems can handle these commands.
- **ICP table** stores, interprets, and prepares packets to and from the spacecraft.
- **ICP hash** is a service to verify the authenticity of the packets to and from the spacecraft, using an HMAC-SHA-224 hash function.
- **MCS codec** converts commands to ICP packets and vice-versa.
- **Terminal table** stores requests made from MCS Front-end and responses from MCS Back-end to MCS Front-end after being verified.
- **TLE fetcher** is a service that downloads the latest set of orbital elements from NORAD and propagates the spacecraft's orbit.
- **UUID generator** for other components.
- **RabbitMQ** messaging service exchanges data between MCS modules.

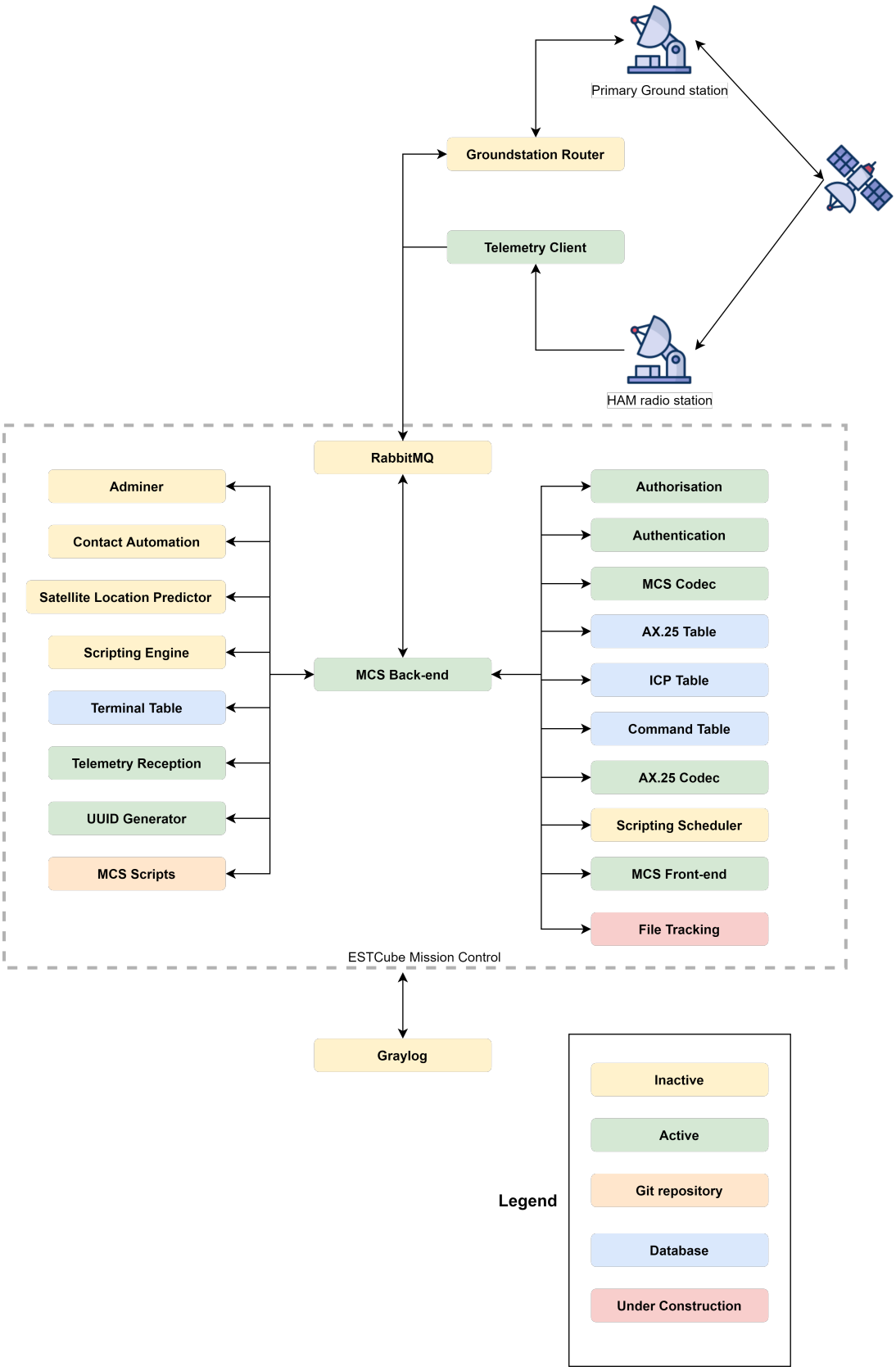

Figure 2.4: ESTCube-2 Mission Control System in 2021 [13]

Commands

An overview of the ESTCube-2 MCS pipeline for sending commands from 2021 is shown in Figure **2.5**. The figure is not entirely correct anymore with the biggest difference being that AX.25 has moved from MCS to the ground station client side. The overview of the currently followed steps are:

- The operator sends the command from MCS Front-end.
- The command reaches the MCS Back-end and the packet processing starts with saving the request in the Terminal table.
- The command is validated against the Command Table, which checks that the number of arguments is correct, that the command is addressed to a subsystem supporting the command, etc.
- A UUID for the packet is automatically generated.
- The validated request is stored in the ICP Table.
- The MCS Codec converts the command to an ICP packet.
- The converted ICP packet is saved as a response in the ICP Table.
- As the last step of packet processing, the Hash Calculator is used to verify the integrity of the packet
- The packet is sent to RabbitMQ, where the ground station client picks it up and forwards it to the satellite.
- As a last step in the MCS Back-End, the packet is saved in the Terminal Table.

Current state for the operator

The author analyzed the overall status of the MCS and its sub-components and mapped all UI routes/pages along with their state before the start of the thesis improvements. The front-end had eight pages:

- **Dashboard** view, which provides a quick overview of the status of the satellite and its main subsystems. The dashboard consisted of a Grafana view which did not display anything at the time but had been configured to list telemetry reception.

- **Terminal** view for sending commands to the spacecraft and interpreting responses from the spacecraft. There are three options or modes for sending a command - ”Commands”, ”Raw Packet” and ”Python Script”. Raw packets were used for testing spacecraft communication with short and simple commands. The Terminal view contains a table that shows the requests sent to the spacecraft and the responses received. For each command and response, the table lists the destination subsystem ID, source subsystem ID, length, command ID, command version, UUID, mode, arguments, CRC, and the target ground station.

- **Commands** view for managing the set of commands which can be used in the terminal. The view lists available commands and allows adding new commands or editing or disabling existing ones.

- **Data Processing** view for managing data processing workflows had not been implemented.

- **Settings** view for operator preferences contained an option to change MCS into dark mode, but not all pages supported the dark mode.

- **Tracker** view shows ESTCube-1 satellite trajectory on a globe as a proof of concept, but actual functionality for orbit tracking had not been implemented.

- **Logs** view to show logs from the MCS back-end. Due to an unresolved issue in the implementation, the view only displayed the last log entries.

- **File management** view to exchange files with the spacecraft. The view had not been implemented.

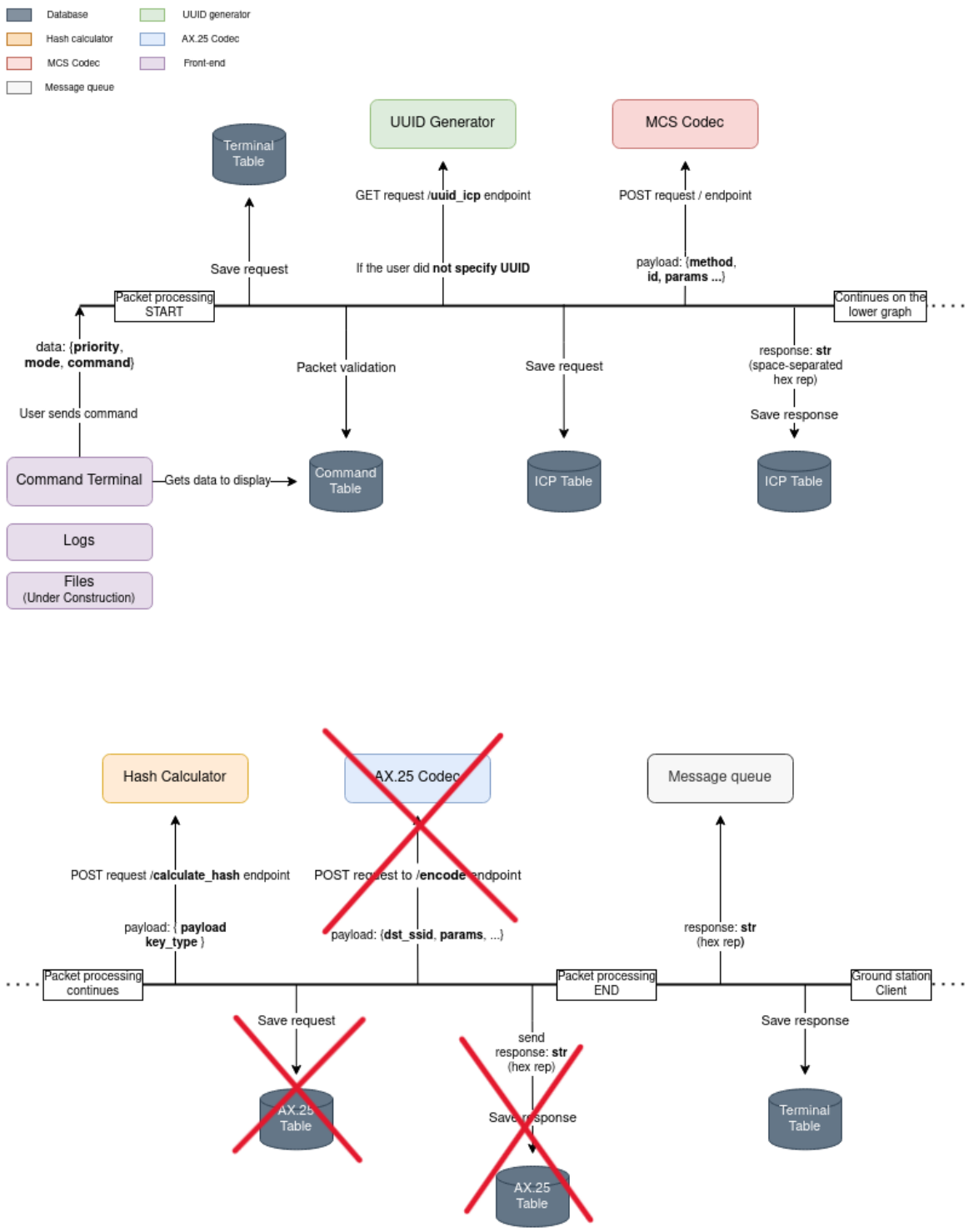

Figure 2.5: ESTCube-2 MCS pipeline in 2021 with red crosses on components that are currently not in use [13]

3 Research Methodology

For research, it is important to pick the most suitable methodology. There are many possible methodologies with different approaches and processes. Three typical research methods are quantitative, qualitative, and mixed methods. [14] Choosing the suitable method depends on the data the researcher needs to collect.

Qualitative methodology is used when the goal is to get complete data based on personal experience. Qualitative research usually involves more textual descriptions, explanations, or interpretations of collected data. This method does not have very strict boundaries considering that when using this method researchers can build up new theories on their collected data. This way of analysis is also meant to get to know something more thoroughly and in greater detail. One possible way of collecting data for qualitative research is to do interviews with open questions that allow interviewees to talk freely meaning there are no given answers from which to choose. [14, 15]

On the contrary, the quantitative methodology is used when the goal is to get the most objective data on a subject. In this case, the data could be collected using a questionnaire with a set of concrete answers to choose from. The results would be numerical. The main difference with the previous method is that with the quantitative method, the researchers aim to measure reality and get statistical information. [14, 15]

The third, mixed method, is used when researchers collect numerical data and have a descriptive way of analyzing the data. [14]

For the research in this thesis, the qualitative methodology was chosen because there is little information or materials on the operations of smaller satellites. Open-questions interviews were conducted to collect more descriptive information. The author wanted to know about the experiences and situations of other people related to satellites that would contribute to satellite operations. This kind of outcome is most convenient for the topic of this thesis considering that the data is needed to understand the problems in mission control systems and to improve the

current situation or to collect ideas for future developments.

3.1 Data Collection Method

Qualitative data collection methods are usually more focused on gaining insights and understanding underlying reasons by going deeper with gathering information. Collecting data for qualitative research includes individual interviews, qualitative surveys, focus group discussions, record keeping, case studies, and observations. [16] In this thesis, individual interviews are used.

Individual interviews are common because of their approach to getting more personal and detailed data from the participant. This type of interview can be done in structured, semi-structured or unstructured format. In a structured interview, there is a predefined list of questions that are asked. Whereas an unstructured interview needs little or even no preparations at all. Interviews give a bit of flexibility to both sides in an interview. The researcher can add questions as needed and the participant can talk more freely about the experience. [17]

The individual interview method was chosen for the research data collection, and the interviews were structured. The author preferred to send the questions to participants beforehand, so the interviewee could have the time to prepare for the interview considering it was an open-question interview. This approach makes the interview smoother and allows the researcher to ask additional or specifying question questions whenever she has the urge to do so.

The interviews were conducted via call and a recorder was used. For recording the audio, the author used OBS Studio software [18]. This way the researcher could focus on participating in the interview and did not need to take notes simultaneously. After starting the analysis, the author listened to these recordings, wrote down the answers, and prepared the data for further analysis.

The author chose four people to interview. Two of the interviews were in English and two were in Estonian. The interviewees were chosen because they all have some experience in satellite operations. Interview participants are shown in table **3.1**.

Table 3.1: Interview Participants

Name	Position in space missions	Missions participated
Subject A	Spacecraft engineer and operator	ESTCube-1, ESEO
Subject B	Spacecraft engineer and operator	Clyde Space, Kepler Communications, Cubehub, ESTCube-1
Subject C	Ground station operator and engineering supervisor	PicSat, ESEO, ESTCube-1, Koit, Hämarik, ...
Subject D	Spacecraft engineer and operator, ground station operator	Koit, Hämarik

3.2 Data Analysis

The author chose thematic analysis for analyzing the collected data. Thematic analysis is the most common approach to use along with qualitative analysis. The aim of using thematic analysis is to organize and construct the data in a way that helps the researcher extract most of the hidden information from the data gathered. Thematic analysis is used to find people's views, knowledge, opinions, experiences, or values from qualitative data. The thematic analysis identifies common themes - topics, ideas, and patterns of meaning that come up repeatedly. [19, 20] The author chose to use the following Braun and Clarke's recommended six-step process for thematic analysis. [21]

Step 1. Familiarization. Get to know the data. Read through the text, make notes, and get familiarized with the data. If the data is recorded as audio or video, then it might be necessary to transcribe the audio first. [22]

Step 2. Coding. In this step the data will be coded, which is done by highlighting some parts of the data and coming up with shorthand labels or a code for it. Each code should describe the idea that was given in the text. Every relevant or potentially interesting part should be coded. [22]

Step 3. Generating themes. Generally, themes are codes in a broader sense. Several codes

are combined into a single theme. In this step, some codes might be considered too vague or not relevant enough and discarded, whereas some codes might be important enough to become themes themselves. [22]

Step 4. Reviewing themes. Return to the dataset and match themes against it. Ensure that the themes are working well with the data; if not, then themes should be reworked. [22]

Step 5. Defining and naming themes. By the start of this step, there should be a final list of themes. Make sure that for each theme it is exactly clear what is meant under it and the name would be easily understandable. [22]

Step 6. Writing up. After the previous five steps are done and the data has been thoroughly analyzed it must be written down. The aim should be to pinpoint the findings from the analysis. Addressing each theme and describing their meaning should lead to structured data that will help to answer the research questions. [22]

For all these steps, a simple approach was used. Since there were only four interviews and they all played an important role in the result, the author listened to and transcribed all interviews by hand. This way the author became familiar with the responses and could already see patterns emerging. The texts were read and analyzed multiple times.

After gathering and preparing the data the author started with thematic analysis. Two tools, between which the author chose for thematic analysis were Microsoft Word and NVivo. Even though NVivo is the most used tool for thematic analysis, Microsoft Word was chosen because the author was already familiar with it and it was also suitable for this analysis, as there were four interviews to analyze.

After getting familiar with the data, the following next steps of the analysis were performed in Word:

1. Generating codes for interview texts.
2. Making a table for the codes.
3. Organizing the codes - adjusting the codes, so they cover a topic.
4. Grouping codes into themes.
5. Assigning themes to interview texts.

As the last step in the analysis, the author described the results based on themes and codes in writing.

4 Results

The research was done by using thematic analysis which is described in chapter 3. The basis for the analysis was four interviews with experts. The analysis brought out four main themes which were the footing for describing the final results in this paragraph. The main reason for forming different themes is to answer the research questions, so the themes were derived from the questions.

Mind maps were created based on codes used in each theme: orbital pass actions in Figure **A.1**, mission data in Figure **A.2**, issues in Figure **A.3** and mission information in Figure **A.4**. In Sections 4.1 - 4.4, each theme is thoroughly described by using interviewees' answers.

The four main themes from the interviews are listed in Table **4**.

Table 4.1: Themes

Orbital pass actions
Mission data
Issues
Mission information

4.1 Orbital pass actions

For the mission operations not to cause significant problems and surprises, it is important to plan the actions to be taken during an orbital pass. By orbital pass, we refer to the time the satellite passes over the ground station while orbiting the Earth. This theme groups together the experiences of operating the satellite during a pass.

4.1.1 Orbital pass information and preparation

The short pass duration is mentioned in the interviews, so actions had to be planned beforehand. Usually, the pass for a polar LEO lasted from one to 13 minutes and there were around 6 to 10 passes per day. There was a pass every 90 minutes until a longer pause at night. It is common to prepare for the next pass and plan the actions, such as setting up the spacecraft for another experiment. This could be achieved by manually sending telecommands or uploading a script or predefined set of commands. When this activity is done, the command will be run in real-time or stored and run at a given time.

The arrival and loss of sight times for satellite were displayed and kept an eye on during a pass. If the satellite came into sight, it showed how much time is left until the end of the pass and when the next pass will start. When a satellite rises from behind the horizon and becomes visible, it is called AOS; when it descends behind the horizon, it is called LOS. The ground station calculates the time of arrival of sight, and a minute before the pass starts, the antennas are rotated to target the satellite. Throughout the orbital pass, antennas rotate and follow the satellite.

Considering the pass duration, the spacecraft operators had to plan beforehand, which actions were to be taken. Spacecraft operators had a document prepared for them with a list of actions they wanted to perform, for which they took into consideration pass time and trajectory. It is important to consider trajectory, because the higher the elevation angle of the spacecraft's flight trajectory over the ground station, the faster the antenna rotates to track the spacecraft. The lower the elevation angle during the spacecraft fly-over, the more likely it is that buildings or trees get in the way, or that there is additional noise from terrestrial sources. The actions taken during a pass were decided before the pass happened. Also, while the satellite was charging batteries from solar cells, then the operators could perform actions that needed more energy. So, not only is it essential to consider the trajectory of the spacecraft from the perspective of the ground station, but also from the perspective of the spacecraft - is it in sunlight, or is it entering an eclipse? From another perspective, it was mentioned that the orbital pass time is so short that the telemetry collection is done automatically with each pass. Although, pass time being short doesn't necessarily mean that telemetry collection is done automatically, having it automated makes it easier to use the short pass times efficiently.

4.1.2 Orbital pass manual and automated actions

From one interviewee's experiences with multiple missions, the spacecraft operations were manual at the beginning of the mission, but as the mission progressed and systems were developed further, more and more automated procedures were implemented. Another reason for automating the system is to reduce the cost of spacecraft operations because having more operators for spacecraft will also mean higher costs for the mission. There are also cases where the initiative to improve the mission operations comes from the passion of the team. On the other hand, for experimental missions where it is not known how things will work out after launch, there is no real point to assign a large team to operations before a successful launch and commissioning. When it seems that the mission will be up and running for longer, then there is a point to start automating different actions.

The two approaches for automating actions are either automating with time commands on the satellite or the ground with MCS. Both of these are used in missions. For example, scripts for regular connectivity and health checks could be run from MCS at the beginning of each pass. It is also possible to have a scheduled on-board operation - taking a picture on the other side of Earth at a given time. One of the interviewees added, that overall, there should be a balance between onboard autonomy and scripts on the ground. From the operators' point of view, it is good if actions that take multiple passes are automated. For example, it is good to automate passes where logs or pictures are received.

During critical operations or emergency situations, there is a need to take into consideration the reaction time of an operator. Since a script could notice anomalies and react faster than an operator, spacecraft safety is a good reason to automate some parts of the mission.

4.1.3 Orbital pass span

When satellites are launched into space, there is a commissioning phase at the beginning of operations, which differs from the usual passes. The commissioning phase can be challenging, especially when not all the systems were ready and properly tested beforehand, such that additional development and debugging are needed in this phase.

The most mentioned topic in the interviews under this theme is regular actions during satellite orbital passes. All operators have their own regular actions that are taken during a pass. One of the interviewees mentioned that these actions were: setup of all the prerequisites, preparing the commands, quickly testing the commands on an engineering model, sending the commands

to the spacecraft in orbit, and then starting a housekeeping script that would regularly monitor the spacecraft until the pass was over. Others also mentioned that satellite health data is being collected during a regular pass. At the start of the pass, the satellite subsystems are powered on, health information is gathered, and by the end of the pass the subsystems are powered off again. The operators updated the on-board orbital elements at the beginning of the pass, and also synchronized the time with a dedicated script.

As for a ground station during a regular pass, the data is forwarded to the mission operating center, the telecommands are sent to the satellite, and telemetry is received. Ground stations need to forward the telecommands with the correct modulation and on the correct frequency, also compensating with the Doppler effect. Also, a ground station would automatically track the satellite with its antenna.

4.1.4 Orbital pass types

In addition to the pass actions for getting satellite health data, there are passes where dedicated actions are performed. Firstly, there are passes where satellite firmware is updated. In the case of one mission it was mentioned that in the two years of mission span, there were 21 firmware updates to a subsystem on the satellite. This shows that a reliable and comfortable way to update satellite software from the mission control system is needed. In addition to updating firmware, there should also be an option to update on-board variables without the need to upload a new firmware each time.

There are also passes where pictures are downlinked. Depending on the capabilities of the communication system used on the satellite, receiving a picture can take multiple passes. For one mission it took around three days to get one photo. Getting telemetry logs from the satellite also took multiple passes.

4.1.5 Orbital pass analysis

From the telemetry received, the real-time analysis was done by an operator during an orbital pass. Just by looking at the numbers, the operators start to notice patterns, tendencies, and anomalies, and get an idea of what the time series of various parameters looks like. Operators keep an eye on error logs and subsystem statuses throughout the passes. Often there were multiple people involved while operating a single pass. Usually, one person was operating and others were checking logs in real-time to see if some issues would pop up, so multiple people

were assisting operations in real-time. They needed to have enough of a time buffer for the end of the pass when they could switch off everything and ensure that the spacecraft is in a safe state. The operator would then announce the end of the pass, reflect on what went great and what went poorly, and use this info for the next pass.

4.2 Mission data

Depending on the mission, there could be a large amount of data from the spacecraft and some from the mission operations which should be stored in the MCS. Under this theme, the different types of data and how they are used will be covered.

4.2.1 Telemetry

The interviewed people talked about telemetry, namely which type of data should be forwarded from the spacecraft to Earth. The most critical parameters to monitor are battery voltage, enabled subsystems, the current consumption of various components, switch statuses, angular rates, temperatures, generic modes, errors, and various indicators of what is nominal or anomalous.

An overview of satellite communication subsystem information is helpful for ground station operators. This is good for giving feedback to the operators if there is a need, for example, to change some parameters for the communication subsystem. Also, sometimes the spectrum of the radio signal is recorded to investigate distortions in the signal.

In the ESTCube-1 mission, the images arrived from the spacecraft as packets and dedicated software was used to store raw images with metadata in FITS format. The main reason for saving raw images was the lack of compression, which made it possible to post-process the images (remove atmosphere and enhance contrast, for example). Once calibrated and interpolated, the images could be exported into PNG files. Image metadata contained histograms, based on which it was possible to decide which images to downlink and how to configure the camera for acquiring the following images.

4.2.2 Data storage and analysis

There are multiple options for storing the mission data, each with its advantages and disadvantages. In one mission, keeping logs in SVN caused issues, the SVN repository became corrupt

due to the large amount of files and directories, causing data loss. In ESTCube-1 the data from each pass was stored in a folder named after the year, date, and time of the pass. The interviewees said having dedicated databases for storing telecommands and telemetry is better than using a code repository.

The interviews mentioned a tool for data analysis called Grafana, which is suitable for visualizing telemetry time series and eyeing parameters. Overall, graphs are very beneficial - the operator can quickly notice a downward trend in battery voltages, an upward trend in current consumption, or noise in measurements that might indicate system issues. Although not all missions had graphs for all of the data, scripts were used to go through the folders and generate graphs. More informative graphs make it easier and more comfortable to explore the data.

The main tracking place for one mission is a web page, with a table view showing the latest telemetry from the satellite and errors that have occurred. For example, it can be seen when a subsystem uses too much power and then the spacecraft engineer investigates and resolves the issue based on that.

ESTCube-1 operators had human-readable logs and within the logs they had different colors, formatting, and tables. It was possible to save logs as HTML and view them in the browser. In another mission, with more complicated cases the analysis was done by looking into received packets and reading raw binary packets.

It is very helpful for the operators if the MCS automatically keeps track of critical parameters and highlights cases where they exceed thresholds. It is good if, after getting telemetry, simple checks are performed to see if the parameters are within the allowed range. After a satellite pass, the parameters from health data are checked for more subtle trends or patterns. For the ground station, signal strength, spacecraft frequency and packet statistics on packet loss are analyzed. This will give an overview of the system's performance. A nice-to-have feature would be to have email notifications sent to spacecraft operators when something is wrong.

All in all, there is often a lot more telemetry collected from a mission than people are able to analyze during the mission.

4.3 Issues

A space mission is complicated and long, therefore it is quite common that during spacecraft operations unexpected issues may arise. The author gathered issues that have happened during the interviewees' missions, under this theme.

4.3.1 Communication issues

One of the bigger topics of issues that the experts have experienced is communication issues. One of the interviewees said that the scariest issue was the satellite's loss of communications, because, without functioning communication with the satellite, there is very little which could be done to investigate and resolve the issues. This sets a high priority on telemetry and information about communications. This type of telemetry must be easy to access and up to date, and potential issues indicated in the data must be discovered quickly.

Ground station issues

Communication issues with the spacecraft could either be caused by issues on the spacecraft or issues on the ground. For example, the interviewees mentioned issues with orbit propagation and satellite tracking software GPredict [23]. When the operators encountered issues while connecting with the satellite from the ground stations, it was helpful to have a radio spectrum view to see if the ground station was sending out telecommands and receiving telemetry. For example, the radio spectrum may be noisy, the spacecraft rotating weirdly, or the signal too weak to get the whole message. The issues with radio noise could be caused because the noise factor was not considered when choosing the location for the ground station. It is possible that during the course of a long mission, the surroundings and the noise environment of the ground stations change. Other interviewees brought out that there can be issues with ground stations caused by bad weather on the ground. To avoid these problems affecting the mission, it is always good to have a backup ground station, from which it is possible to operate the spacecraft.

In the interviews it was also emphasized how important it is to have a ground station ready for launch. There was a situation where the ground stations were not thoroughly planned and it caused a lot of issues for the mission in the beginning.

It was also mentioned that with high passes, there may be issues with the ground station antenna rotation. In one case, the ground station rotation was so slow that it caused to loss of communication with the satellite.

Other communication issues

With the communication subsystem, there have been problems with the Doppler effect being unaccounted for on the spacecraft.. As a workaround, the transmitter on the ground had to have the functionality to shift the frequency while transmitting a packet. In that case, the spacecraft

expected a precise uplink frequency already accounting for the Doppler shift for the communication to work.

The experts also mentioned issues where the radio dedicated to downlinking images from the spacecraft was broken, due to which this part of the mission, getting images, was unsuccessful. In such situations, it is good to reserve enough flexibility for workarounds to hardware issues.

4.3.2 Insufficient testing

There have also been issues with the whole communications flow from MCS to spacecraft. One reason for this is that the operations side has not gotten enough attention before the launch and the system has not been tested as thoroughly as it should have been. It can happen that the satellite itself is built with great passion from the team members, but it is not thought through how to operate it later. The costs of not testing the system correctly can be very high. Mistakes in operating the satellite in-orbit may result in the loss of spacecraft. Therefore it is a necessary practice to test all telecommands on an engineering model before applying the same procedure on the satellite in orbit. But even then, the engineering model may not be identical to the spacecraft in orbit. In the interviews, a mission was mentioned, for which all the testing before the launch went wrong. The satellite and the software were not ready for the time of testing and eventually, the mission did not end successfully.

Packet loss may be significantly worse from the orbit than it is on the ground. In one of the cases mentioned during an interview, this had not been considered before the launch. As it turned out, the packet loss was caused by satellite rotation. After realizing the issue, it was possible to mock it on the engineering model. For a solution, the team tried to use shorter commands when the satellite was spinning or using more transmission power so that the satellite would receive the signal.

4.3.3 Unready for launch

Closely related to the lack of proper testing, there are issues with software not getting ready for launch. At the beginning of one of the missions, the satellite was launched with bootloaders and therefore needed a lot of software updates. The satellite and the mission control system needed further development while the satellite was already launched. When the software is not ready for launch, the overall system testing will be poor and may cause communication issues later.

There are many reasons why software may not get ready for launch. From experiences mentioned in the interviews, for example, it can be due to bad project management or the lack of a clear understanding of the mission and the actions and steps needed for the mission to succeed. With one mission, this caused the timetable to be packed and chaotic. Another reason was the launch being shifted to an earlier date. When the mission team was offered an early launch for a better price they accepted the earlier date, but the satellite's software development and testing were severely affected.

4.3.4 Team issues

An important part of space missions is the team that is doing the work, and having issues with the team can cost a lot in the quality of the mission. An interviewee mentioned that in their case there was poor teamwork and no set of rules that people followed while doing their work, heavily affecting the integration between the systems later. Due to this integration issue, some developments were discarded. Also, issues can come up due to missions being understaffed Finding passionate people with the necessary experience and know-how for the project may be difficult.

4.3.5 Poor hardware decisions

Some of the hardware issues mentioned during the interviews were closely related to the overall project management. Some decisions were made in the early stage of the mission when there was no clear idea of what the mission would be about, and some of these decisions were not re-evaluated in later stages. This created a situation with unused hardware on the satellite. It is not a terrible issue, when there is at least a chance to do firmware updates and with this continue with the mission while improving the satellite capabilities. However, it is better for the mission to have at least a minimum viable version working and ready, then to have a complicated satellite, which does not get enough testing and has issues later on. It is sometimes better to have simpler but more durable hardware. There was a case where the main radio did not work - it was built to have a lot of functionality but in the end, it was not reliable enough. Another example of poor hardware decisions is missing a real-time clock from the satellite, which caused problems with on-board tracking of mission time. It is needed for operators also to know and understand the state of the spacecraft hardware, as it is very easy to make mistakes without knowing how the hardware or firmware works underneath, and what kind of issues the systems might have.

4.3.6 Power issues

The need to optimize the power budget was also mentioned. This can be due to the very limited area for solar panels and the issue with the solar panels degrading quickly in the low Earth orbit environment. Another example from the interviews was a mission with lack of power for subsystems. This happened because there was no way to warm up the batteries, and batteries cannot be charged while the satellite is in the shadow of the Earth. Therefore, in the beginning all the energy went to warm up the satellite, and only then could it be used.

4.3.7 Mission control system issues

Most of the issues mentioned above were not strictly related to a mission control system, but more to the satellite operations or the satellite itself. With the MCS there were also issues, for example, cases where the MCS crashed. These were relatively easy to handle, a simple turn-off and -on helped. Issues with log storage were mentioned. There were two directories, the first for XML files describing the packets that came from mission operating software and were meant to be sent to the spacecraft, and the second for packets sent from the spacecraft. On the one hand, this functioned as a backup for restoring the MCS database. On the other hand, a lot of files in a single directory made the system slow and due to this some packets did not arrive at the correct time and sometimes it was difficult to find issues. The lack of UUIDs for commands and responses made it difficult to identify which response was for which command. There were also difficulties with estimating the time until the loss of sight, which made it seem as if the satellite had already passed behind the limb of Earth while the operators were still receiving packets. As a side effect, this caused another more significant issue with failing to power down subsystems, due to the inability to predict how much time was left before the end of the pass.

4.3.8 Anomalies

While it may be possible to prepare for previously mentioned issues with a lot of testing, there are some cases where it may be impossible to be fully prepared, and solutions would have to be figured out on the go. Issues mentioned in the interviews are brought out here as they may give ideas and examples of what may come up. Firstly, the issue mentioned may have been caused by radiation hitting a part of the spacecraft and flipping a bit. At times, the connection with the satellite was lost and there were times when the operators had to rely on a watchdog timer to

reset the satellite. One of the anomalies was a malfunctioning transceiver chip on a satellite, which needed to be fixed with software. Due to a firmware bug in a communication system, any packet which contained too many zeroes or ones in a row during the reception was dropped.

In other cases, it was important in which order systems are turned on because it might have caused ”major outgassing” and irreversible damage to the spacecraft otherwise. Sometimes camera images were lost, or logs with on-board measurements became corrupted. During one of the missions, sensor measurements got more and more noisy over time.

Not all of the issues mentioned here would make the mission fail, but for the spacecraft operations to go more fluently it is good to know these issues and be ready for when something may happen.

4.4 Mission information

Under this theme, various other points that are good to know for building mission control systems and operating spacecraft, are brought out.

4.4.1 Valuable utilities for Mission Control System

Interviewees said it is good when there is an option to access the mission control system from the web. Also, an option to see the ground station view from multiple computers is good for operating the spacecraft. It is helpful for the spacecraft operator to see the uplink and downlink from the radio waterfall view, as it helps to verify that everything is working from the ground station side.

Also, it is crucial for spacecraft operations to have a system with good usability. Having code completion while sending commands to the spacecraft is very comfortable for the operators and together with inline help messages for commands, it can significantly improve the efficiency of operations. The usability of a system improves when some of the actions are automated and the remaining manual actions are optimized for minimal clicks and key presses. Ideally, the operator would not have to retype the same set of commands and could just configure a button to do that for him. Overall, it is good to have automated actions wherever possible, with the possibility to override the automation if needed.

Another important quality of a good mission control system is flexibility. The experts mentioned that it is good to have a system where it is easy to implement changes and add func-

tionality. This was especially important with the missions that were not ready for launch. For example, adding a photo viewer to your mission control system later on. Another example of the updates to spacecraft operations is integrating to Grafana. Even if Grafana is added to the mission control at a later phase, it is good if all the data about the mission has been stored.

Additionally, MCS should have all the history of what has been done and the possibility to analyze it later. The interviews also mentioned that it is good to have a summary of the most important information from the last orbital pass. For example, packet loss statistics - how many packets were successfully transmitted or received, how many packets were re-transmitted and how many were lost. Also, it would be good if there is a possibility to export the data from the MCS.

On the other hand, an interviewee pointed out that not everything should be a part of the MCS. The subsystem or payload teams should have the flexibility to use their software of preference for data processing and analysis.

4.4.2 Miscellaneous

As mentioned before in Section 4.3.1, there may be unexpected issues with the ground station and therefore it is important that the MCS system is built to work with multiple ground station configurations. Another closely related point is to have the ability to get packets from radio amateurs as well, especially for missions that make use of radio amateur frequencies. In some of the missions mentioned during the interviews, there were cases when the mission team did not receive packets, but radio amateurs did.

One big factor in mission success is the team's passion. From the experience of one of the experts who has been involved in multiple missions, with people in the team who are passionate about improving the system, the mission progresses more quickly and ends up being more successful. There was also a view from the opposite side where the team was understaffed and the mission did not go so well, as most of the people only did the work for their thesis at the university and did not have the passion to push the limits for the mission itself.

In spacecraft operations, having different procedures for what to do when something goes wrong and documenting it is valuable. For example, in one of the missions, as the battery ran out of energy and the spacecraft entered survival mode to recharge it, there were pre-defined steps on how to work in these situations. Overall, experts emphasized the need for good documentation because of the time and effort it takes to train new operators, and since people may

forget things, having it all written down is helpful.

Additional points that were mentioned to keep in mind with the spacecraft operations include:

- Support for multiple firmware versions.
- To use the same commands on engineering and orbit models - so the operator could try the same commands on the ground first.
- To have low-level access to debug systems and work around issues in orbit.
- To keep in mind that actions might behave differently in orbit than on the engineering model.

5 Improvements to the system

The architecture and current state of the MCS and its components were analyzed and are brought out in subsection 2.3.1. A substantial amount of changes would be needed to reach an operational system comfortable for the operators. Following a discussion with other ESTCube-2 mission team members, further efforts were prioritized with the approaching launch in mind. The list of priorities is as follows:

- Update the commands view
- Update the terminal view
- Implement satellite telemetry storage
- Create a dashboard with the most important parameters

In the scope of this thesis, software improvements were made to the front-end, and a preliminary design for the dashboard was created. The improvements are described further in this chapter.

5.1 Technical state and front-end refactoring

To get started with updating the command view, it was needed to get an overview of the technical state of the UI.

Lighthouse [24] was used to analyze the MCS web interface technical state. Lighthouse is a tool for improving the quality of web pages. It audits many aspects of the site - such as performance, accessibility, progressive web apps, SEO, and more. Lighthouse can be run via Google Chrome DevTools, from the command line, or as a node module. [24]

Lighthouse was run in a Google Chrome web browser on the Commands page, but unfortunately, the page returned an error due to a web request taking too much time. A further look

into the issue showed that a web request stayed loading. The lighthouse tool was used again after fixes and the result can be seen in section 5.2.

The author set up the system locally to see the technical state by looking at the code base. The author ran ”npm outdated” and ”npm audit” commands to check if any of the front-end dependencies needed updates. The response to the ”npm outdated” command is shown in Figure **5.1**. Among other packages, it indicates that react, typescript, webpack, and next ought to be updated. These commands indicated that the versions of several dependencies had become

```
$ npm outdated
```

Package	Current	Wanted	Latest	Location	Depended by
@babel/cli	7.16.0	7.21.5	7.21.5	node_modules/@babel/cli	gui-mcs-front-end
@babel/core	7.16.0	7.21.8	7.21.8	node_modules/@babel/core	gui-mcs-front-end
@babel/eslint-parser	7.16.0	7.21.8	7.21.8	node_modules/@babel/eslint-parser	gui-mcs-front-end
@babel/plugin-proposal-class-properties	7.16.0	7.18.6	7.18.6	node_modules/@babel/plugin-proposal-class-properties	gui-mcs-front-end
@babel/preset-react	7.16.0	7.18.6	7.18.6	node_modules/@babel/preset-react	gui-mcs-front-end
@deck.gl/core	8.7.8	8.9.14	8.9.14	node_modules/@deck.gl/core	gui-mcs-front-end
@material-ui/core	4.12.3	4.12.4	4.12.4	node_modules/@material-ui/core	gui-mcs-front-end
@material-ui/icons	4.11.2	4.11.3	4.11.3	node_modules/@material-ui/icons	gui-mcs-front-end
@material-ui/lab	4.0.0-alpha.60	4.0.0-alpha.61	4.0.0-alpha.61	node_modules/@material-ui/lab	gui-mcs-front-end
@material-ui/styles	4.11.4	4.11.5	4.11.5	node_modules/@material-ui/styles	gui-mcs-front-end
@testing-library/jest-dom	5.15.0	5.16.5	5.16.5	node_modules/@testing-library/jest-dom	gui-mcs-front-end
@testing-library/react	12.1.2	12.1.5	14.0.0	node_modules/@testing-library/react	gui-mcs-front-end
@types/geojson	7946.0.8	7946.0.10	7946.0.10	node_modules/@types/geojson	gui-mcs-front-end
@types/react	17.0.34	17.0.59	18.2.6	node_modules/@types/react	gui-mcs-front-end
axios	0.21.4	0.21.4	1.4.0	node_modules/axios	gui-mcs-front-end
classnames	2.3.1	2.3.2	2.3.2	node_modules/classnames	gui-mcs-front-end
date-fns	2.25.0	2.30.0	2.30.0	node_modules/date-fns	gui-mcs-front-end
deck.gl	8.7.8	8.9.14	8.9.14	node_modules/deck.gl	gui-mcs-front-end
eslint	7.32.0	7.32.0	8.40.0	node_modules/eslint	gui-mcs-front-end
eslint-config-airbnb	18.2.1	18.2.1	19.0.4	node_modules/eslint-config-airbnb	gui-mcs-front-end
eslint-config-prettier	8.3.0	8.8.0	8.8.0	node_modules/eslint-config-prettier	gui-mcs-front-end
eslint-plugin-import	2.25.2	2.27.5	2.27.5	node_modules/eslint-plugin-import	gui-mcs-front-end
eslint-plugin-jsx-a11y	6.4.1	6.7.1	6.7.1	node_modules/eslint-plugin-jsx-a11y	gui-mcs-front-end
eslint-plugin-prettier	3.4.1	3.4.1	4.2.1	node_modules/eslint-plugin-prettier	gui-mcs-front-end
eslint-plugin-react	7.26.1	7.32.2	7.32.2	node_modules/eslint-plugin-react	gui-mcs-front-end
eslint-plugin-react-hooks	4.2.0	4.6.0	4.6.0	node_modules/eslint-plugin-react-hooks	gui-mcs-front-end
jest	27.3.1	27.5.1	29.5.0	node_modules/jest	gui-mcs-front-end
less	4.1.2	4.1.3	4.1.3	node_modules/less	gui-mcs-front-end
next	11.1.2	11.1.4	13.4.2	node_modules/next	gui-mcs-front-end
next-images	1.8.2	1.8.5	1.8.5	node_modules/next-images	gui-mcs-front-end
prettier	2.4.1	2.8.8	2.8.8	node_modules/prettier	gui-mcs-front-end
prettier-eslint-cli	5.0.1	5.0.1	7.1.0	node_modules/prettier-eslint-cli	gui-mcs-front-end
react	17.0.2	17.0.2	18.2.0	node_modules/react	gui-mcs-front-end
react-dom	17.0.2	17.0.2	18.2.0	node_modules/react-dom	gui-mcs-front-end
react-router	5.2.0	5.3.4	6.11.1	node_modules/react-router	gui-mcs-front-end
react-test-renderer	17.0.2	17.0.2	18.2.0	node_modules/react-test-renderer	gui-mcs-front-end
sass	1.43.4	1.62.1	1.62.1	node_modules/sass	gui-mcs-front-end
satellite.js	4.1.3	4.1.4	5.0.0	node_modules/satellite.js	gui-mcs-front-end
typescript	4.4.4	4.9.5	5.0.4	node_modules/typescript	gui-mcs-front-end
webpack	5.62.1	5.82.1	5.82.1	node_modules/webpack	gui-mcs-front-end

Figure 5.1: npm outdated command result

deprecated and a vulnerability had been discovered in one of them. The version of Next.js (v11.1.2) used in the front-end, is vulnerable to Denial of Service (DoS) attacks [25]. In addition to updating the version of Next.js, it was also necessary to update the versions of React.js and Material UI. In the source code of the commands view, there were some files in JavaScript and some in TypeScript. The source code was a mixture of React class and function-based components. This is understandable since the project was based on voluntary contributions by students of different backgrounds and experiences, so the code style varied from file to file.

A closer inspection of the source code highlighted the need to refactor the code before performing any other improvements. The author considered refactoring a high priority because it would reduce the effort and time in debugging and resolving issues in the system later on.

The front-end of MCS was updated to resolve the vulnerability and the issues with deprecated dependencies. After updating the dependencies, the versions of the main dependencies were as follows: React version 18, MUI version 5, and Next version 13. Together with the dependencies updates, the source code was refactored. It was quite a significant change from the Material UI (version 4) to MUI (version 5). The author rewrote the code from JavaScript to TypeScript for consistency as part of the refactoring. The most complicated change was to update the code around WebSocket, which is used for the terminal requests and responses, and logs from MCS Back-end. A general list of changes is as follows:

- All files were ported from Material UI to MUI. Toggling from light to dark mode stopped working because using multiple color themes is handled differently in the newer version. Currently, there exists only a light theme, as part of future work, color themes and toggling between them should be added using state management, for example, ContextAPI.
- Converting most of the source code from JavaScript to Typescript. Currently, API calls do not return typed results, this should be added as part of future work. Additionally, converting most of the source code from class to functional so that the components would support hooks usage in React. Furthermore, the React documentation encourages function usage over class. More detailed analysis is necessary to resolve the remaining issues with functionality and some of the pages which are not displayed.
- The old Logs view only displayed the latest log entries from the MCS Back-end component, as shown in Figure **5.2**. This issue was present already before the refactoring and updates to dependencies. It may be that the developer who initially implemented the logic did not check the actual WebSocket messages. The problem was caused by not keeping all the messages from the WebSocket, but only the latest one. ContextAPI was used to keep the WebSocket open and store the messages with React useState hook and local storage to resolve the issue. The new view is shown in Figure **5.3**.

During refactoring and updates in dependencies, the author made changes in over 60 files, with additional revisions from functional patches.

5.2 Commands page and the commands

After the refactoring was done on the front-end, it was possible to improve the commands view to fit the needs more and have better usability.

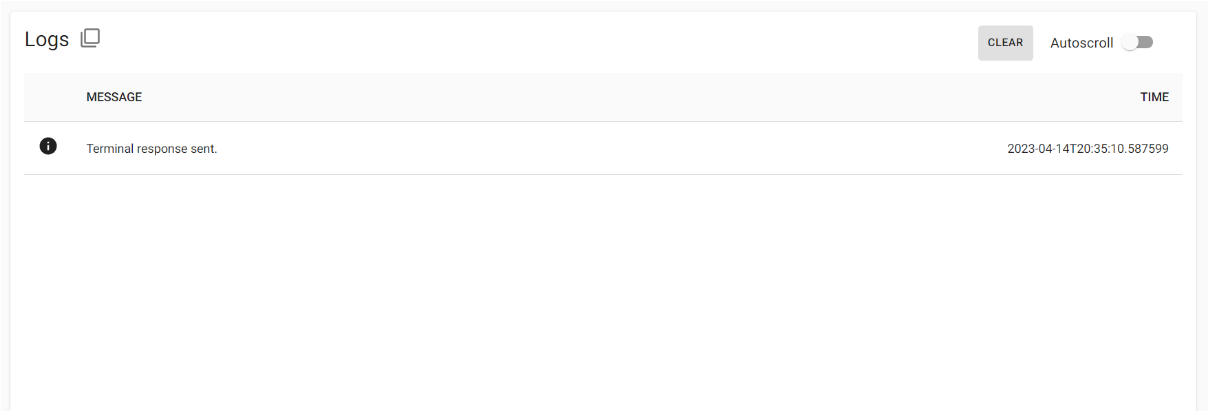

Figure 5.2: MCS logs old view

Figure 5.3: MCS logs fixed view

The previously existing commands view can be seen in Figure **5.4**. There were some issues with the UX; to see command parameters, the operator had to edit the command, and the commands were not ordered by ID. There were minor issues such as a mislocated button for adding commands, and the button to show hidden commands could go out of view when there were too many commands. The author also gathered ideas from other ESTCube-2 team members on how the table columns should look. Additionally, the option to add common commands was missing. On ESTCube-2, common commands have the same effect on all subsytems; a specific set of command IDs are reserved for this purpose. Common commands include functionality like file read and write, reboot and ping.

After resolving the aforementioned issues, the author added commands the spacecraft currently supports and can already be used for testing. To add commands to the system, it was necessary first to figure out which commands are currently usable on the engineering model. The

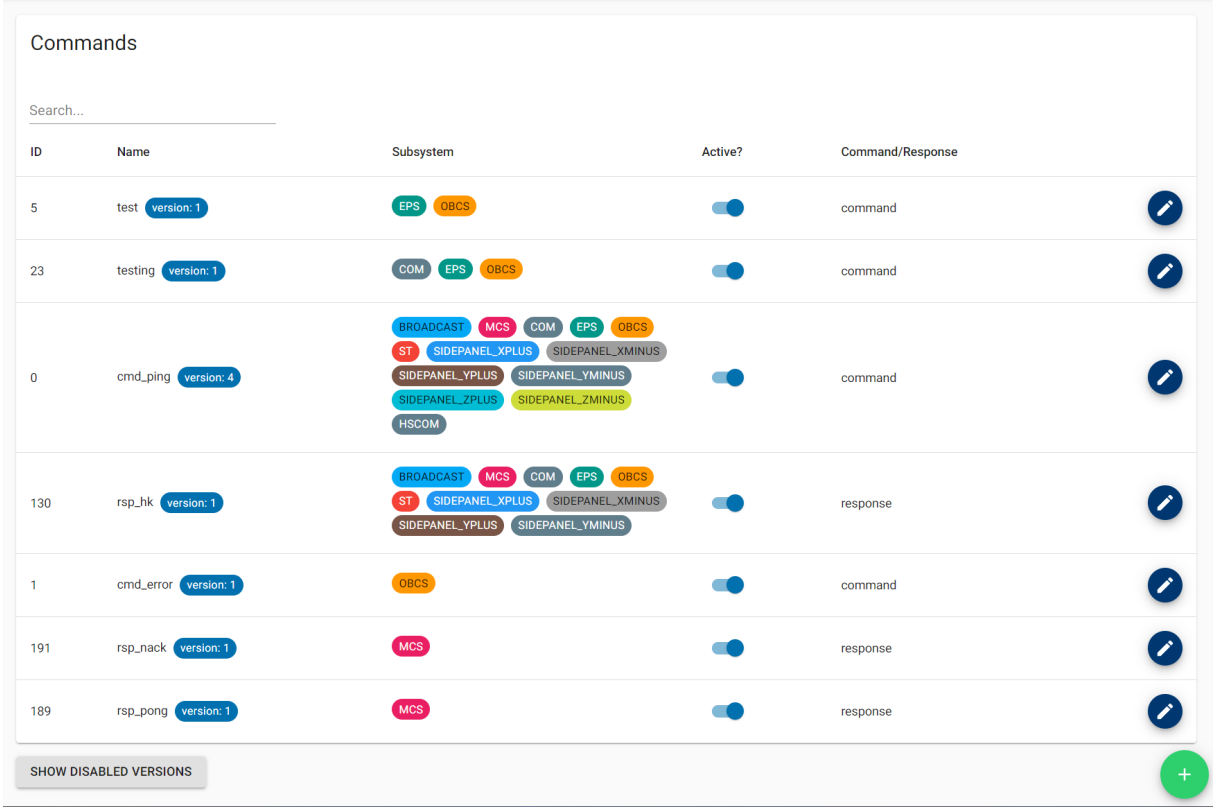

Figure 5.4: The previously existing MCS commands view

author checked the mission documentation, the source code of the satellite firmware, and asked team members for additional information. Initially, eight commands and four responses were added, all commands added are as common commands, and there are no subsystem-specific ones. The MCS commands view after changes implemented can be seen in Figure **5.5**.

After the changes were implemented, Lighthouse was used again on the commands view. The results were good, with some minor changes needed (e.g., a button not having an accessible name for people who use screen readers, and the site not using HTTPS); Performance - 97/100, Accessibility - 93/100, Best Practises - 92/100 and SEO 100/100.

5.3 Design idea for dashboard

After getting the feedback from the interview analysis it was certain that there needs to be a useful dashboard for the mission control system. The design for the dashboard was communicated with some of the ESTCube-2 team members, initial ideas were gathered and put into a design file in Figma [26].

The ideas given for an initial dashboard were:

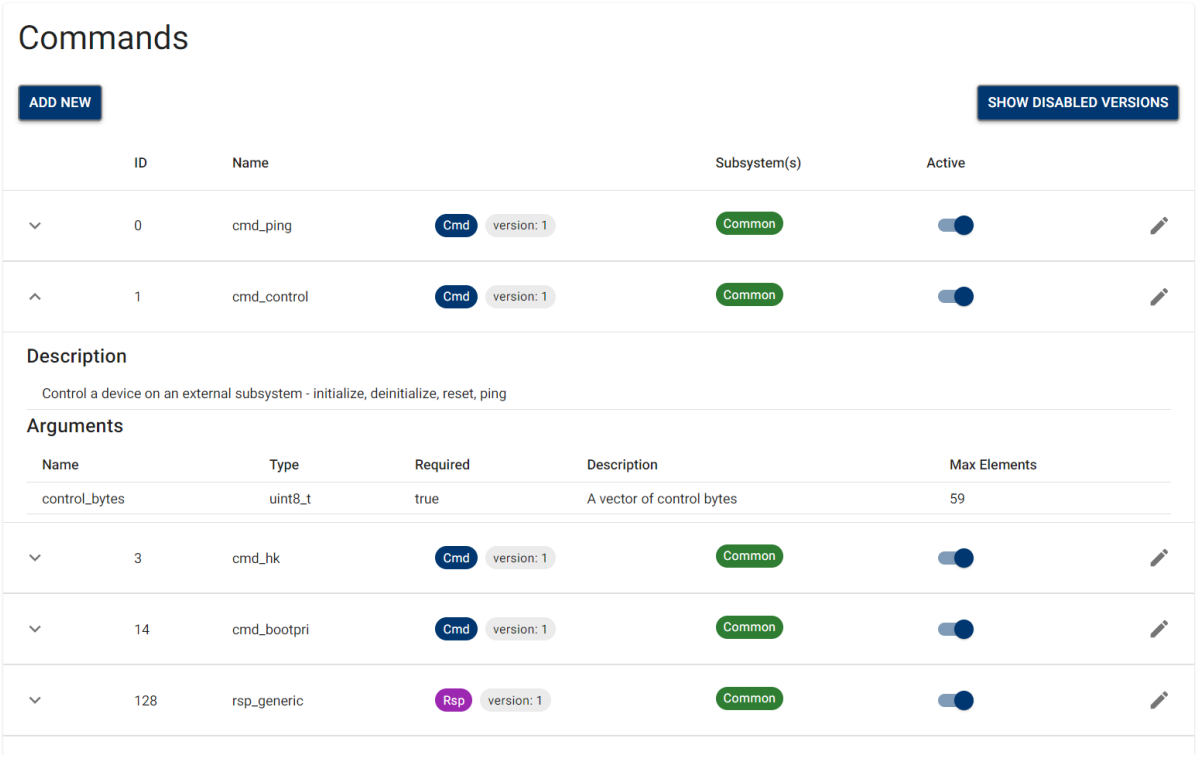

Figure 5.5: MCS commands view after changes

- Time until next pass/Time until the end of the pass
- Satellite health information: Battery voltage, MCU temperatures, Free heap and the number of resets together with the latest reset reason with the current uptime
- Commands handled by the MCS
- Last error and its description, number of errors graphed over time

The dashboard view can change based on operators' preferences and depending on how the mission progresses. In the current design, MCU temperatures are grouped into eight graphs, but there might also be four separate graphs, the same goes for displaying the free heap on the graph.

Before the view can be created the telemetry from the satellite needs to be stored.

The design created in Figma can be seen in Figure **5.6**.

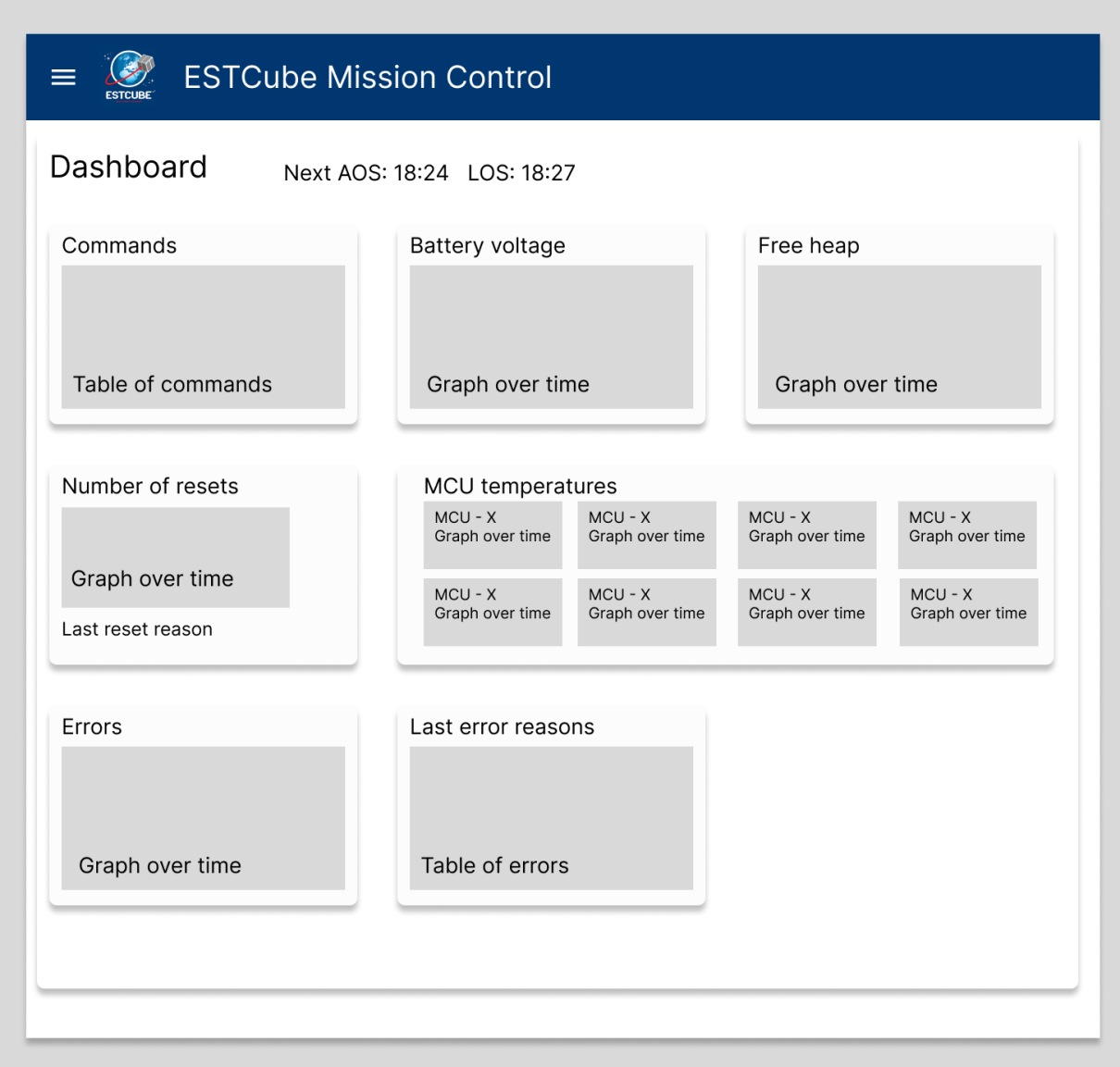

Figure 5.6: MCS dashboard design idea

6 Testing

As the interviews showed, testing is a crucial part of the mission. Therefore the author hoped the development would reach a state where it is possible to do integration testing with the satellite and the MCS. Sadly the initial plans for integration testing could not be achieved due to the MCS development not reaching a state where telemetry could be properly stored in the appropriate databases, also, the satellite was not ready enough to be tested with the MCS, either. After the MCS improvements, some basic integration tests were still done on the engineering model.

The first tests were done with only the communication subsystem and running commands to ping and ask for housekeeping data. Further tests on the system were done by sending the housekeeping data from multiple subsystems to the MCS every minute. The telemetry view result can be seen in Figure **6.1**.

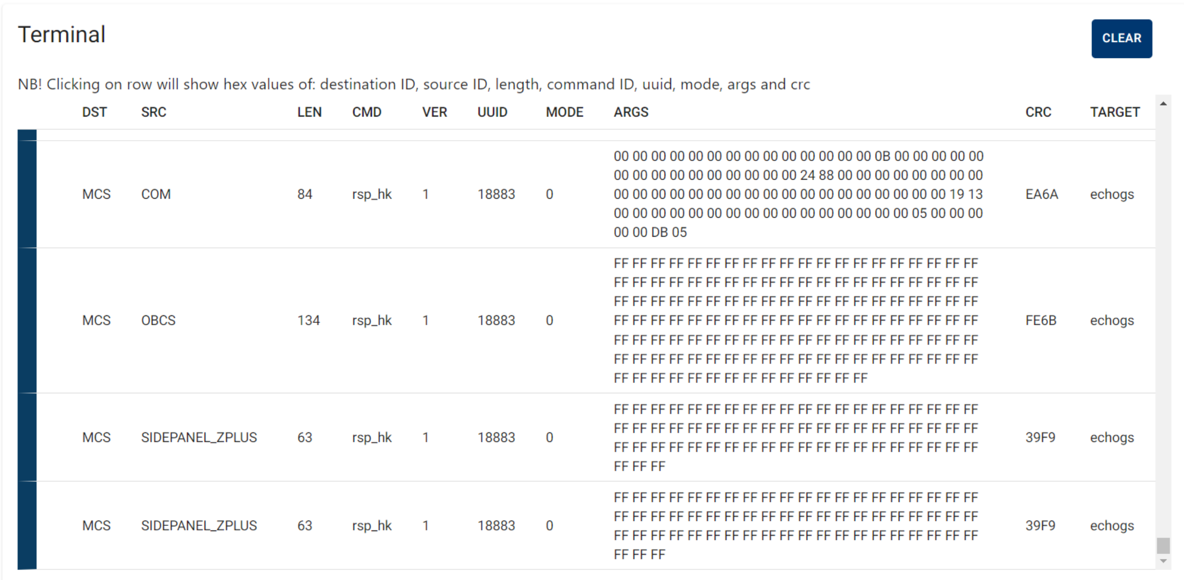

Figure 6.1: MCS telemetry testing

During the testing, there were issues with one of the MCS components, and since the logs view was fixed, it was easier to figure out that the problem lay with the MCS Codec component. A simple restart on the component fixed the issue, and sending commands was working again, the issue was probably in a faulty deployment of a component. The logs view can be seen in Figure **6.2**.

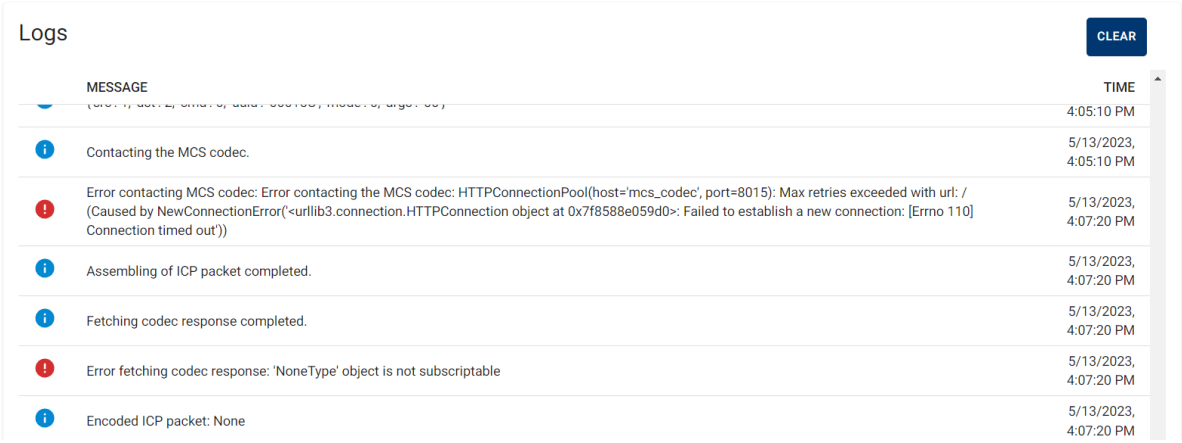

Figure 6.2: MCS logs view

The commands that were used, ping and housekeeping, worked, but there is an issue from the ground station client where the connection cuts off at some point. The issue needs further investigation on why it happens and how to solve it.

7 Recommendations

Based on the interview responses and the analysis done on mission operations, the author will give recommendations to follow while finalizing the ESTCube-2 MCS.

Firstly, for the orbital pass, there are actions for the mission spacecraft operations which are beneficial to keep in mind. There is a more complicated commissioning phase at the beginning of the mission. Initially, missions are more about manual work; later, when the mission is more stable, the work can be more and more automated. All commands should be tested on the engineering model before sending them to the spacecraft. Overall, there should be a clear idea of actions taken during a pass and have it documented.

Secondly, space missions generate a lot of data - from the spacecraft and operations itself. It is essential to think through how the data is stored before the launch so that there would not be any data loss during the mission. There should be various graphs showing trends and other information from the telemetry. This can give an insight into potential issues.

Thirdly, during a space mission, there can be a lot of issues that vary in significance; some can be critical, and some just an annoyance for the mission. The most severe problems are usually related to the loss of communications, and there may be many causes for these issues. Some causes might be due to issues with the ground station, such as the Doppler effect, lack of proper testing beforehand, and not being ready for launch.

Lastly, there is additional miscellaneous general information that is good to know for spacecraft operations and does not fall into any of the categories above. The mission control systems should be built to be easy to use, flexible to extend, and conveniently accessible from the web. Before launch, the team needs to create documented actions to follow after launch, in LEOP, and during an orbital pass.

7.1 Future Work

There is still work to be done in the software development and testing before the system is ready for launch. It is necessary to add databases for telemetry and implement the dashboard, once the telemetry exists. After this part is implemented, the integration testing with the satellite can begin. In the logs view, the logs should be grouped based on commands because, currently, it is difficult to say when a new command starts. There is an issue with connection loss on the ground station client, this needs to be investigated and resolved before the launch.

Furthermore, a proper way to deploy code to different environments is missing and the system currently does not have any security implemented. Since another team member is currently working on this topic, it was not covered in the scope of this thesis, but it still needs to be mentioned that it is an essential topic for the system.

Finally, as the analysis on the state of ESTCube-2 MCS showed, it is crucial to keep MCS up to date to ensure that there are no dependency issues.

8 Summary

During this thesis the author worked on improving the ESTCube-2 mission control system to ready it for the upcoming launch. The objective was to analyze spacecraft operations, study the ESTCube-1 and the ESTCube-2 mission operations in more detail, and find out the current state of the ESTCube-2 MCS.

The author conducted four interviews using qualitative research methodology to analyze the topic. This gave an understanding of the needs for spacecraft operations. Based on the interview results, four themes were created.

Space missions can have a lot of unknown issues for which it is difficult to prepare. It is essential to have a motivated team, clear goals for the mission, and good testing before the launch; then, many issues can be avoided. In the seventh chapter, the author discusses recommendations, limitations, and future work based on the analysis.

Work was done regarding ESTCube-2 MCS software development and analysis. At the end of the thesis a state was reached where sending and receiving commands in the MCS system was possible. Saving the results in the database and having a good dashboard for a mission overview were analyzed.

The software changes done in the MCS were based on the analysis of the current state of ESTCube-2 and the acute need for integration tests between the satellite and mission control. Some plans and analyses were made for the next possible steps, based on the points that came out from the interview results.

The work done during this master thesis is useful for the ESTCube-2 mission and may also be useful for other missions where the spacecraft operations need attention. The result of the analysis of operators' experiences can be useful for the next space missions in general.

Acknowledgements

I would like to express my gratitude to the thesis supervisors, Kristo Allaje and Indrek Sünter for their help and guidance throughout the thesis. I am thankful to the people who were willing to participate in the interviews and share their experiences and knowledge. I am grateful to my family and friends, for supporting me.

Cathy Toomast

References

[1] Hans Teras, ”The ESTCube-2 mission”
from https://www.estcube.eu/blog/The-ESTCube–2-mission (accessed: 14.04.2023)

[2] Mars, Kelli, ”5 Hazards of Human Spaceflight”, *NASA. Retrieved 6 October 2019*, 2019.

[3] Cracknell, Arthur P and Varotsos, Costas A, ”Editorial and cover: Fifty years after the first artificial satellite: from Sputnik 1 to ENVISAT”, *International Journal of Remote Sensing*, **28** 2007, 2071-2072

[4] Doran, Jamie, and Piers Bizony, *Starman: The truth behind the legend of Yuri Gagarin*, Bloomsbury Publishing USA, 2011.

[5] Loff, Sarah, ”Apollo 11 Mission Overview”
from https://www.nasa.gov/mission_pages/apollo/missions/apollo11.html
(accessed: 19.04.2023)

[6] Primož Rome, ”Every Satellite Orbiting Earth and Who Owns Them”,
DEWESoft, January 18, 2022, from
https://dewesoft.com/blog/every-satellite-orbiting-earth-and-who-owns-them
(accessed: 19.04.2023)

[7] Wilfried Ley, Klaus Wittmann, Willi Hallmann *Handbook of Space Technology*, John Wiley & Sons, UK, 2009.

[8] ESTCube-1 mission description from Estonian Student Satellite Foundation
from https://www.estcube.eu/projekt/ESTCube-1 (accessed: 21.04.2023)

[9] Homepage for the Koit & Hämarik student satellites, Technical University of Tallinn
from https://satelliit.taltech.ee/#/ (accessed: 21.04.2023)

[10] Jerry Jon Sellers, Robert B. Giffen, William J. Astore, Wiley J. Larson *Understanding Space: An Introduction to Astronautics*, McGraw-Hill Custom Pub., 2003

[11] J. Houser & M Pecchioli, 'Database Administration for Spacecraft Operations – The Integral Experience', *ESA Publications*, **bulletin 103**, August, 2000

[12] Indrek Sünter. Private collection of videos about ESTCube-1 mission operations.

[13] ESTCube-2. Unpublished internal company document

[14] Carrie Williams, "Research Methods", *Journal of Business & Economic Research*, **5**, March 2007

[15] Meri-Liis Laherand *Kvalitatiivne uurimisviis*, Infotrükk, Tallinn, 2008.

[16] Adi Bhat, "Qualitative Data Collection: What it is + Methods to do it" from https://www.questionpro.com/blog/qualitative-data-collection-methods/ (accessed: 23.04.2023)

[17] Harpal Singh & Shabeen Shareef, "Qualitative Interview: What it is & How to conduct one" *Question Pro*, from https://www.questionpro.com/blog/qualitative-interview/ (accessed: 23.04.2023)

[18] Lain Bailey, "The homepage of OBS Studio", from https://obsproject.com/ (accessed: 14.05.2023)

[19] Lacey A. and Luff D *Qualitative Data Analysis*, The NIHR RDS for the East Midlands / Yorkshire & the Humber, 2007.

[20] Bhandari, P., "What Is Qualitative Research? | Methods & Examples" *Scribbr.*, January 30, 2023, from https://www.scribbr.com/methodology/qualitative-research/ (accessed: 23.04.2023)

[21] B. Virginia and C. Victoria, 'Using thematic analysis in psychology', *Qualitative Research in Psychology*, **3 (2)**, 2006, 77-101

[22] Caulfield, J., "How to Do Thematic Analysis | Step-by-Step Guide & Examples" *Scribbr.*, November 25, 2022, from https://www.scribbr.com/methodology/thematic-analysis/ (accessed: 23.04.2023)

[23] Alexandru Csete, ”The homepage of Gpredict”,
from http://gpredict.oz9aec.net/ (accessed: 14.05.2023)

[24] Chrome Developers, ”The documentation of the Lighthouse overview”,
from https://developer.chrome.com/docs/lighthouse/overview/ (accessed: 09.05.2023)

[25] Snyk Limited, ”next@11.1.2 vulnerabilities”,
from https://security.snyk.io/package/npm/next/11.1.2 (accessed: 13.05.2023)

[26] Figma, Inc., ”The homepage of Figma, a collaborative interface design tool”,
from https://www.figma.com/ (accessed: 15.05.2023)

Appendixes

1 Interview questions

1. Describe yourself a bit.
2. Describe the satellite project you participated in.
   (a) What was your position in the satellite team?
3. Describe a usual satellite orbital pass.
   (a) Which actions were performed?
4. How did you analyze the data that came from the spacecraft?
   (a) What were the most typical parameters to check?
   (b) Did you have tables and/or graphs to keep an eye on potential issues?
5. What went wrong?
   (a) Were there any unexpected issues? If there were, then what were these?
6. How many changes did you need to make to your system during the time when the spacecraft was in space?
7. What functionality would you like to have in your satellite control system, but did not have?
8. (Additional) What did you like the most about your mission control system?

2 Thematic maps

Appendix 2.1 - Theme - Orbital pass actions

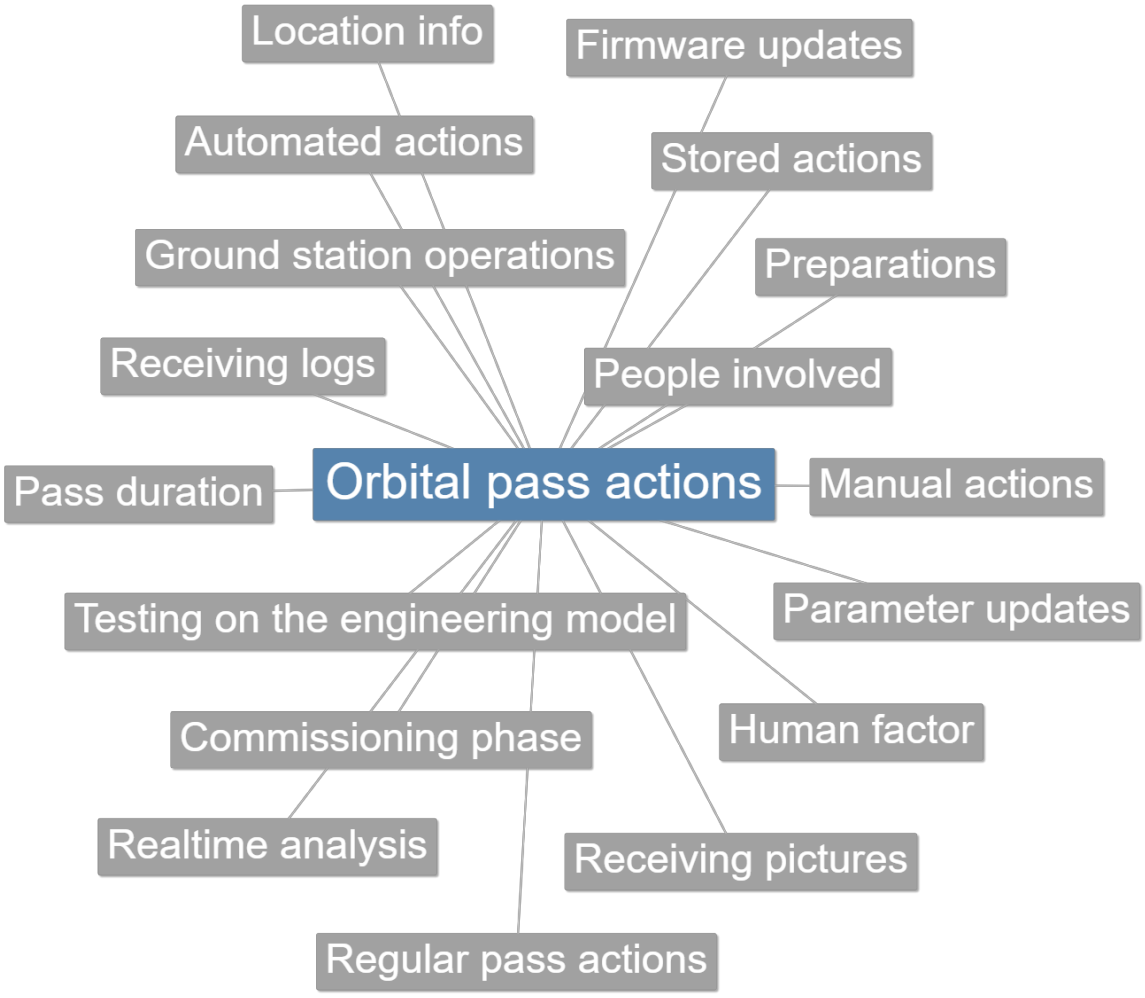

Figure A.1: Theme - Orbital pass actions

Appendix 2.2 - Theme - Mission data

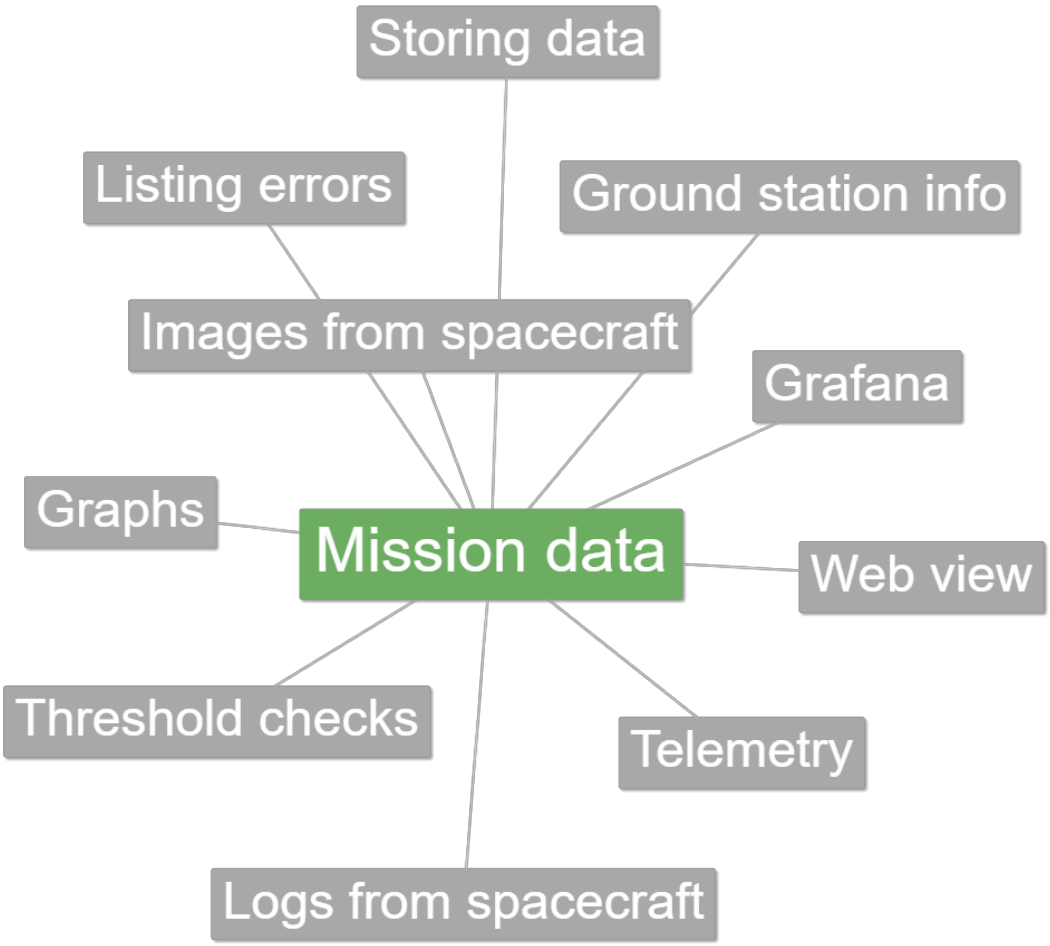

Figure A.2: Theme - Mission data

Appendix 2.3 - Theme - Issues

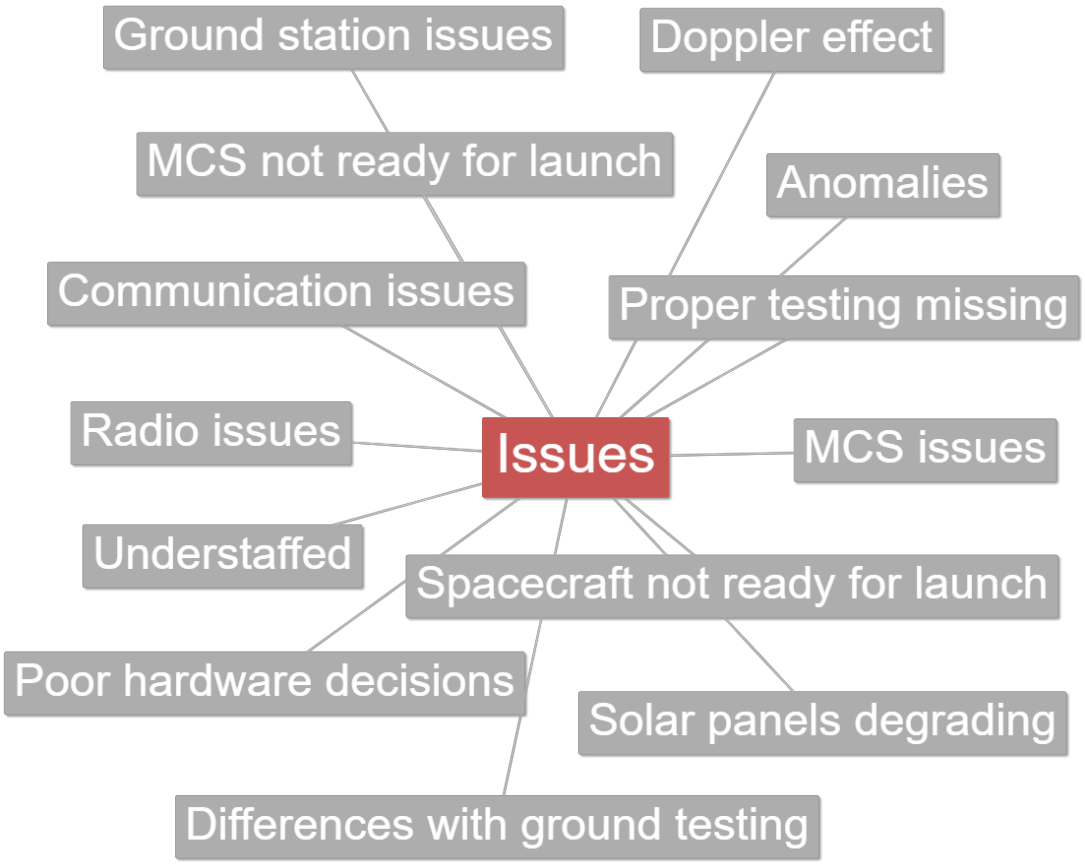

Figure A.3: Theme - Issues

Appendix 2.4 - Theme - Mission information

Figure A.4: Theme - Mission information

Non-exclusive licence to reproduce thesis and make thesis public

I, Cathy Toomast

1. grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis

   "ESTCube-2 Mission Control System: Preparation for In-orbit Operation"

   subervised by Kristo Allaje and Indrek Sünter

2. I grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in points 1 and 2.

4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Cathy Toomast

20.05.2023