ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Variable Pitch System for UAV proprotors

Carlos André Pinto Ramos Gonçalves Rijo

Thesis Research Plan of the Mastersś degree in Critical Computing Systems Engineering

Supervisor: Ricardo Augusto Rodrigues Da Silva Severino
Co-Supervisor: José Renato Santos Machado

Porto, January 9, 2024

Abstract

TODO - abstract up to 200 words

Keywords: Keyword1, ..., Keyword6

Resumo

TODO - abstract up to 1000 words ???

Contents

List of Figures

List of Tables

List of Algorithms

List of Source Code

List of Abbreviations

MDU	**M**ain **D**evice **U**nit
SDU	**S**econdary **D**evice **U**nit
VPP	**V**ariable **P**itch **P**roprotor
FPP	**F**ixed **P**itch **P**roprotor

List of Acronyms

COTS	Commercial Off-The Shelf.
CPU	Central Processing Units.
FPGA	Field-Programmable Gate Arrays.
GNSS	Global Navigation Satellite System.
OBC	On Board Computer.
PCB	Pinted Circuit Board.
PWM	Pulse Width Modulation.
RPM	Revolutions Per Minute.
RTC	Real-Time Clock.
RTOS	Real-Time Operating System.
SoC	State of Charge.
UAV	Unmanned Aerial Vehicle.
UVP	Under Voltage Protection.
VTOL	Vertical Take-Off and Landing.

Chapter 1

Introduction

Unmanned Aerial Vehicles (UAVs) have witnessed a surge in popularity and research attention. They have become indispensable in various applications, ranging from surveillance to reconnaissance, due to their versatility and efficiency in various applications **uav1**.

This growing interest is evident in the numerous review papers exploring different aspects of UAV development, ranging from open-source hardware and software utilization **uav2**, **uav3**,**uav4**, frame design and optimization **uav5**, **uav6**, control systems, including both conventional and modern communication modalities such as 5G networks **uav7**, **uav8**, **uav10**, **uav11**, to efficient power management strategies, and alternative energy sources to extend UAV battery life **uav15**.

Nowadays, most Vertical Take-Off and Landing (VTOL) UAVs, rely on proprotors with fixed pitch systems because of their simplicity and lack of better Commercial Off-The Shelf (COTS) reliable and efficient solutions but they impose limitations on the achievable flight performance Cutler 2012. Thrust generation is confined to a single direction, hindering the UAV's ability to produce upward thrust relative to the vehicle body. Additionally, the control bandwidth is restricted by the inertia of the motors and proprotors, constraining the UAV's agility and maneuverability Cutler 2012.
As described in recent studies Cutler 2012, these limitations become more pronounced as UAV size increases, impacting stability and control. Larger UAVs face challenges as the need for larger motors with higher inertia compromises rapid control through Revolutions Per Minute (RPM) adjustments alone.

The development of Variable Pitch Proprotor (VPP) systems plays a crucial role in overcoming the limitations of traditional UAV designs, such as those with fixed pitch proprotors Chang et al. 2020, Cutler 2012. Several detailed descriptions of quadrotors, for instance, modeling and dynamics have been published **FPP2**, **FPP3**, **FPP4**, **FPP5**, emphasizing the need for dynamic control mechanisms.
The use of VPP systems, especially in VTOL UAVs, addresses challenges related to control instabilities and energy efficiency Chang et al. 2020.
The incorporation of variable-pitch proprotors provides the necessary flexibility to enhance stability and enable larger UAVs to perform sophisticated maneuvers, overcoming the constraints inherent in fixed-pitch designs Chang et al. 2020, Cutler 2012.

TODO: MORE INFO

VPP Video

1.1 Problem Analyses

The utilization of fixed-pitch propellers in UAVs presents a set of limitations that significantly impact aircraft performance and efficiency. These issues are evident in various phases of flight, including hover and forward flight, and have repercussions for the UAV.

TODO: MORE INFO

1.1.1 Maneuverability and Response

Fixed-pitch propellers inherently constrain UAVs from adjusting the pitch angle during flight Cutler 2012. This limitation results in compromised maneuverability and response, restricting the range of aerobatic maneuvers that a UAV can execute Cutler 2012. Additionally, the inability to change the pitch angle impedes the optimization of lift, landing, and thrust during flight, leading to sub-optimal performance in various operational scenarios Cutler 2012.

TODO: MORE INFO

1.1.2 Power Consumption

With fixed-pitch propellers, the power consumption will be higher. Without the ability to adjust the propeller angle, UAVs may be forced to operate at higher RPMs to compensate for this lack of adjustment Cutler 2012. This higher power consumption not only affects the efficiency of the UAV but also has implications for its endurance, limiting the time the vehicle can remain airborne.

TODO: MORE INFO

1.2 Motivation

TODO: MORE INFO

1.3 Objectives

TODO: MORE INFO

Chapter 2

State of the Art

2.1 UAV

TODO: MORE INFO TODO: ADD FIGURES

2.1.1 VTOL

TODO: MORE INFO TODO: ADD FIGURES

2.1.2 Fixed Pitch Proprotors

TODO: MORE INFO TODO: ADD FIGURES

2.1.3 Variable Pitch Proprotors

Historically, early aviation pioneers experimented with propellers that could only be adjusted on the ground. The first automatic variable pitch air screw was patented by L. E. Baines in 1919. The Gloster Hele-Shaw Beacham variable pitch propeller, developed in 1928, demonstrated practical controllable pitch capabilities. Over time, various designs and mechanisms, including hydraulic and pneumatic systems, were explored and refined. The development of constant-speed propellers marked a significant advancement in aviation technology, offering improved efficiency and performance Accelerator n.d.

A significant advantage of variable-pitch propellers is their ability to adapt to varying airspeeds. When an aircraft is stationary or moving slowly, the propeller blades can be set to a low angle of attack to reduce drag. As the aircraft gains speed, the pitch is increased to maintain optimal performance. This adaptability ensures efficient operation across a range of flight conditions.

The primary purpose of variable pitch propellers is to maintain the optimal angle of attack relative to the changing wind vector as the aircraft accelerates. Traditional fixed-pitch propellers face efficiency challenges in various flight conditions. Adjustable blade angles address this issue, allowing for improved efficiency during takeoff, climb, and cruise.Chipade et al. 2018.

Variable-pitch systems can adjust blade pitch to maintain a selected RPM enhancing overall performance, especially at high altitudes, by allowing the rotor to operate in its most economical speed rangeAccelerator n.d.,Chipade et al. 2018.

Three methods change the pitch: Hydraulic, Centrifugal, and Electromechanical control Accelerator n.d.
TODO: ADD FIGURES

Hydraulic Method

This system involves the use of engine oil pressure to control the pitch-changing mechanism and consists of a pump, control valves, and cylinders that actuate the movement of the propeller blades. In an aircraft without a variable-pitch proprotor system, the pilot uses hydraulics to manually control the pitch of the propeller blades Accelerator n.d.

Hydraulic systems provide a precise means of adjusting the propeller pitch, allowing efficient performance under different flight conditions, and contributing to the overall safety and reliability of the system.

But Hydraulic systems add complexity and weight to the overall aircraft system. More components means more elements could potentially fail or require maintenance. There is also the risk of fluid leakage or fluid contamination that may lead to a reduction in hydraulic pressure, potentially affecting the pitch control mechanism. Hydraulic systems may have a slow response time due to the time it takes for hydraulic pressure changes to propagate through the system which might be a concern in situations where rapid adjustments are required.Accelerator n.d. TODO: ADD FIGURES

Centrifugal Method

In the centrifugal systems, centrifugal weights can be attached directly to the propellers. An eccentric weight is placed near or in the spinner and secured with a spring and, when the propeller reaches a certain RPM, centrifugal force swings the weights outward, driving a mechanism that twists the propeller to a steeper pitch. As the propeller slows down, the RPM drops and the spring pushes the weight back, readjusting the propeller pitch to a shallower pitch.

As advantages, centrifugal systems are simpler compared to hydraulic systems since they involve fewer components. The reliance on mechanical components driven by centrifugal force can enhance reliability because there are fewer points of failure. There is no need to use external power sources, such as an engine-driven pump. Also, centrifugal systems can operate automatically without direct pilot intervention. The system responds to changes in rotational speed without the need for continuous manual control.

However, centrifugal systems may provide less precise pitch control than more advanced hydraulic or electronic systems. This limitation can affect the ability to finely tune the propeller for optimal performance. The response time of centrifugal systems may be slower compared to more sophisticated systems. This limitation could be a factor in situations where rapid adjustments to the propeller pitch are necessary.Accelerator n.d.

TODO: ADD FIGURES

Electromechanical Method

These systems involve electric motors and mechanical linkages to control the pitch of the propeller blades.

Electromechanical methods provide precise control over the pitch of the propeller blades, can offer rapid response times to changes in flight conditions, are often versatile, and can be adapted for various aircraft configurations. Compared to certain hydraulic systems, electromechanical systems might require less maintenance. They often have fewer components prone to wear and can be more straightforward to service.

As disadvantages, electromechanical systems, including motors and associated components, can add weight to the aircraft, require electrical power to operate, and are more complex than purely mechanical systems, increasing the chance of failures.Accelerator n.d.

TODO: ADD FIGURES

2.2 Control System

An On Board Computer (OBC) is a device capable of managing and/or controlling various functions such as:

- It can manage overall system operation.
- Implement safety mechanisms and respond to abnormal conditions.
- Execute algorithms and computations required for the system's functionality.
- Interface with external devices, sensors, actuators, or other embedded systems.
- Implement communication protocols for data exchange.
- Manage data storage and retrieval.
- Implement power-saving modes when appropriate.
- Manage and control peripherals such as communication interfaces, timers, and interrupt controllers.

There are, mainly, three types of control units: Microcontrollers, Microprocessors, and Field-Programmable Gate Arrayss (FPGAs).

Microcontrollers are integrated circuits that contain a processor core, memory, and programmable input/output peripherals. They are compact and cost-effective, have low power consumption, are designed for specific tasks, making them suitable for embedded systems, and often include integrated peripherals like timers, communication interfaces, and ADC.
However, microcontrollers have more limited processing power compared to microprocessors and are less flexible for general-purpose computing.

Microprocessors or Central Processing Unitss (CPUs) focus on processing tasks and rely on external components for additional functionalities. As an advantage, they have high processing power (suitable for general-purpose computing), can run complex operating systems, and have greater flexibility in application design. However, microprocessors have higher power consumption, may require additional components for specific applications, and have a larger form factor compared to microcontrollers.

FPGAs are integrated circuits that can be configured after manufacturing, allowing for custom digital logic circuits. They are customizable for specific applications, have parallel processing capabilities, and can be reprogrammed for different tasks.
But, they have a higher cost (compared to microcontrollers and microprocessors), have higher power consumption (compared to microcontrollers), and have a steeper learning curve for programming and design.

TODO: MORE INFO

2.3 Power Management System

TODO: MORE INFO

2.3.1 Power Distribution

TODO: MORE INFO

2.3.2 Battery Module

TODO: MORE INFO

2.3.3 Battery Protection

TODO: MORE INFO

2.4 External Memory and Storage Units

TODO: MORE INFO

2.5 Servo Motors

TODO: MORE INFO TODO: ADD FIGURES

2.6 Firmware

In embedded systems, the choice of programming languages and the use of a Real-Time Operating System (RTOS) in firmware development are critical decisions that can impact the performance, efficiency, and complexity of the embedded system.

As for Programming Languages, there is *C* widely used in embedded systems due to its low-level features and close to the hardware, efficient use of system resources, and strong support from the embedded development community. However manual memory management can lead to potential bugs if not handled carefully.

Assembly Language provides direct control over hardware and is highly efficient, and useful for writing low-level code, such as interrupt service routines but has a steeper learning curve and it is less portable across different microcontroller architectures.
C++ Language is an object-oriented feature that can enhance code organization and reusability and can provide abstraction without sacrificing performance. However, the code size can be larger and more complex.

Real-Time Operating System facilitates multitasking, allowing concurrent execution of multiple tasks, can provide task scheduling, priority management, and inter-process communication and it is suitable for systems with real-time requirements. But this can add overhead, especially in terms of memory footprint, and the learning curve is steeper Reddy 2006.
The following RTOS examples are open-source, well-documented, compact, and designed for resource-constrained systems, and they support various microcontroller architectures

- FreeRTOS
- ChibiOS
- Zephyr

TODO: MORE INFO

2.7 Communication

TODO: MORE INFO

2.7.1 Wired

TODO: MORE INFO

2.7.2 Wireless

While researching wireless communication, multiple protocols can be studied. They can, mainly, be separated into two categories: short-range and long-range. In the context of UAVs, the focus will be on short-range wireless communication protocols Zeng, Zhang, and Lim 2016, Montori et al. 2018, Ferro and Potorti 2005.

Short-range protocols offer advantages such as lower power consumption, reduced interference, and efficient data transfer within confined spaces. Within this category, options like Bluetooth, Wi-Fi, Zigbee, Z-Wave, and LoRa for short distances emerge as noteworthy candidates. Each of these protocols addresses specific requirements, making them suitable for various aspects of UAV operations, from intra-component communication to data transfer between the UAV and ground control Montori et al. 2018, Ferro and Potorti 2005.

- WiFi (802.11x): Can be used for high-speed data transfer over short ranges. It's suitable when you need to transmit large amounts of data between the UAV and a ground station.
  - Advantages
    - High Data Rates
    - Widespread Standard

    * Bi-Directional Communication
  - Disadvantages
    * High Power Consumption
    * Interference in 2.4 GHz and 5 GHz bands

- Bluetooth: Common short-range wireless technology with low power consumption. It's suitable for communication between components on a UAV.
- Advantages
  - Low Power Consumption
  - Ubiquity
- Disadvantages
  - Limited Range
  - Data Transfer Rates
- Zigbee: Low-power, low-data-rate wireless communication technology that is suitable for short-range communication in embedded systems.
- Advantages
  - Low Power Consumption
  - Mesh Networking
  - Low Latency
- Disadvantages
  - Limited Data Rate
  - Limited Range
- Z-Wave: Low-power wireless communication protocol often used in home automation. It's suitable for control and monitoring applications in UAVs.
- Advantages
  - Low Power Consumption
  - Interference Avoidance
- Disadvantages
  - Limited Data Rate
  - Less Common in Non-Home Automation Devices
- LoRa (Long Range): While designed for long-range communication, LoRa can also be used in short-range applications. It provides low-power, long-range communication suitable for certain UAV scenarios.
- Advantages
  - Long Range
  - Low Power Consumption

- Disadvantages
  - Low Data Rates
  - Unidirectional Communication

Chapter 3

Proposed Approach

3.1 Concept

Fixed-pitch proprotor (FPP) system limitations can be addressed by developing variable-pitch proprotors. Adjusting the pitch in both flight phases may be more complex and expensive but it offers more adaptability and a great positive impact on the overall propulsion system efficiency, increasing the endurance and range. The UAV will consume less in hover and the cruise speed will be much higher.

This way, the proposed solution for this problem is to develop a stand-alone variable-pitch proprotor system that can, in real-time, change the propeller pitch according to each flight phase.

3.2 Requirements

It is important to define requirements, for this system, to better understand the fixed-pitch propeller's limitations and to develop the necessary functionalities to achieve a variable-pitch proprotor system.
This way, the requirements should align with mechanical, control, communication, integration, and validation specifications.

- **Mechanical Requirements**

  (REQ_01): The system shall be designed to retrofit existing UAVs or integrate seamlessly into new UAV designs.

  (REQ_02): The variable-pitch mechanism shall be lightweight to minimize the impact on overall UAV weight and balance.

  (REQ_03): The system shall be able to withstand the operational stresses and environmental conditions encountered during UAV flights.

- **Control System Requirements**

  (REQ_04): The control system shall enable real-time adjustment of the proprotor pitch during different flight phases.

  (REQ_05): It shall incorporate failsafe mechanisms to respond to unexpected malfunctions or loss of communication and revert to a fixed-pitch state in case of critical failures.

(REQ_06): The system shall provide precise control over the pitch angle, allowing for fine adjustments to optimize performance.

- **Wireless Communication Requirements**

(REQ_07): The wireless communication system shall be reliable, with minimal latency to ensure quick response times.

(REQ_08): It shall operate within designated frequency bands and comply with relevant aviation communication standards.

(REQ_09): Security measures shall be implemented to prevent unauthorized access or interference with the control signals.

- **Integration Requirements**

(REQ_10): The system shall be designed for easy integration with common UAV autopilot systems.

(REQ_11): It shall have compatibility with existing UAV avionics and navigation systems.

(REQ_12): The variable-pitch system shall not interfere with other onboard sensors or communication systems.

- **Testing and Validation Requirements**

(REQ_13): The system shall undergo rigorous testing under various operational scenarios, including weather conditions and flight profiles.

(REQ_14): Validation shall include simulated and real-world flights to assess performance and reliability.

(REQ_15): The system shall comply with relevant aviation regulations and standards.

3.3 System Architecture

As it is possible to see, in the proposed implementation of the System Architecture diagram (figure 3.1), the system will be composed of two subsystems: the Main Device Unit (MDU) and (multiple) Secondary Device Unit (SDU).

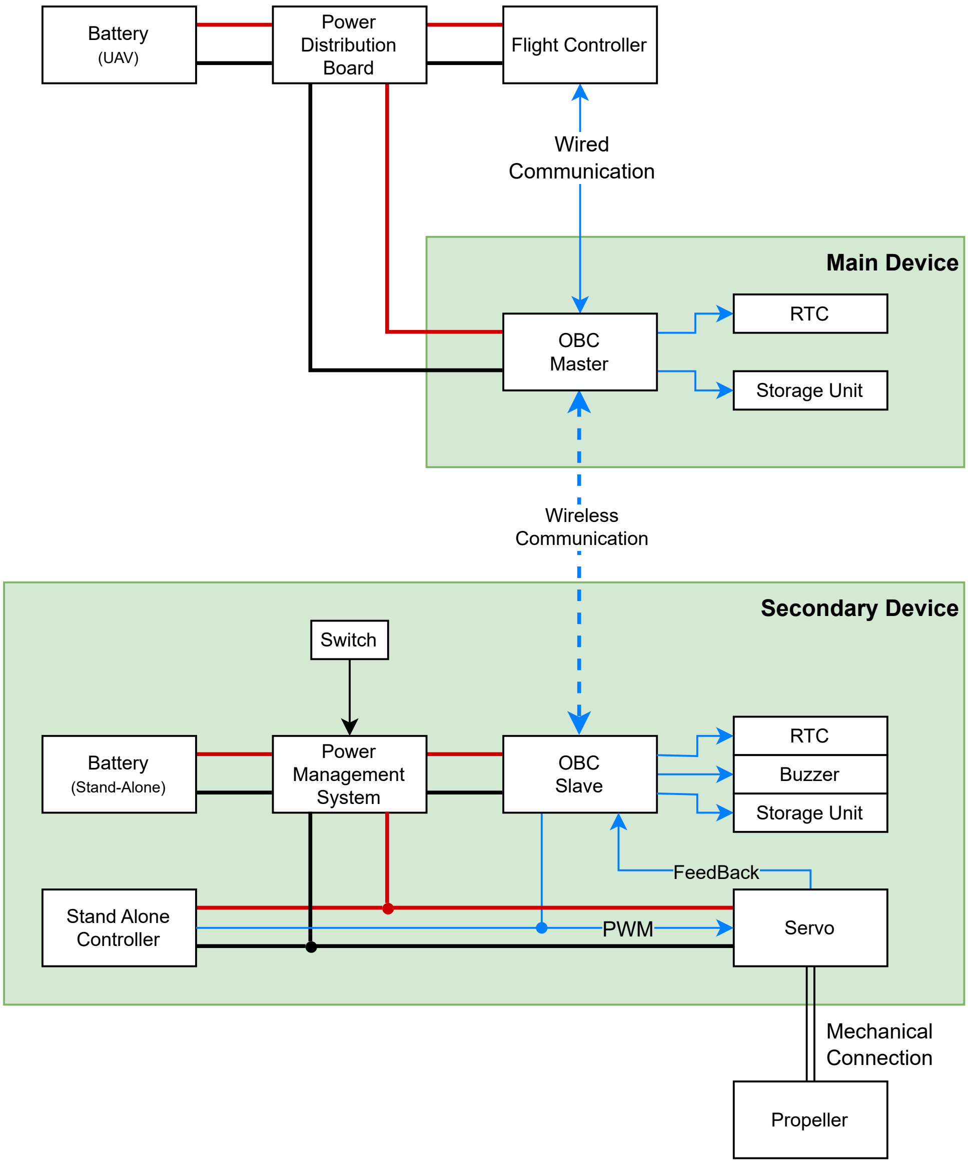

Figure 3.1: Proposed System Architecture High Level Diagram

The subsystems can be consider as stand-alone since the main UAV can work without the subsystems. These subsystems, described in the next subchapters, are responsible for monitoring the flight phase and control the propeller pitch angle.

3.3.1 Main Device

This subsystem will be, mainly, composed by an OBC, a Real-Time Clock (RTC), a Storage Unit, and a wireless communication module.
The MDU will communicate with the Flight Controller and the SDUs by receiving and transmitting information.

With the Flight Controller, through wired communication, the MDU will:

- Receive
  - Flight Phase message

- Maneuver control message
- Global Navigation Satellite System (GNSS) epoch time
- Heartbeat signal

- Transmit
  - Heartbeat signal
  - System Status message

The Flight Phase message and the Maneuver Control message will inform the MDU about the current flight phase and the need to make additional adjustments to the propeller pitch. After interpreting the message, the OBC will send a control command. This control command will be explained further in this chapter

The GNSS epoch time will update the OBC date and time (periodically or on startup). With the help of the RTC, the MDU system will be able to maintain the date and time even when the UAV system is powered off. The GNSS epoch time will be helpful when storing system logs (in the Storage Unit) and will help to calculate the latency of the communication between devices.

Lastly, the received heartbeat signal will work as a *keep alive* mechanism informing, this way, the OBC if the system is powered on. This will help save power since the OBC can shut down when the Flight Controller turns off.

The transmitted heartbeat, which also works as a *keep alive* mechanism, will inform the Flight Controller that the MDU is working correctly. This function will be crucial because if the MDU is not working (powered off or unresponsive) the UAV system will need to enter a failsafe mode and land, as soon as possible, since it can no longer control the pitch of the blades.

Since the MDU is responsible for managing all the SDUs, it must, periodically, inform the Flight Controller about the overall status of the system, so that, in case of any failure, the Flight Controller may enter in failsafe.

With the SDUs, through wireless communication, the MDU will:

- Receive
  - Heartbeat signal
  - SDU Status message
- Transmit
  - Heartbeat signal
  - Control Command
  - Epoch Time

The received and transmitted heartbeat signals will have the same functionality as explained previously. The MDU and SDUs will inform each other if they are working correctly.

The SDU Status message will help the MDU keep track of the status of all Secondary devices. In case of malfunction or if one or more SDUs can't change the propeller pitch, the MDU must be noticed so that it can communicate to the Flight Controller about the failure.

The MDU will send a Control Command, containing the desired pitch, to all the SDUs according to the phase of flight message received previously.

By sending the epoch time to all the SDUs, it is possible to keep the whole system updated and with the same date and time reference.

3.3.2 Secondary Device

This subsystem will be, composed of an OBC, a RTC, a Storage Unit, a battery (with a power management system), an on/off switch, a buzzer, a stand-alone Pulse Width Modulation (PWM) controller, a servo, and a wireless communication module.

The on/off switch and the buzzer will work as human-machine interfaces to help the user interact with the system.
Since the SDU will be designed to be stand-alone (with a dedicated power supply) the system needs to be powered on manually and the buzzer can notice the user that the system is powering on.

The servo, mechanically connected to the propeller, will be responsible for changing the propeller pitch according to the state of flight. It will be equipped with feedback functionality so that the system can control, more precisely, the pitch and know if the propeller has reached its goal.

There will also be implemented a stand-alone PWM controller, able to generate a fixed PWM signal, to control the servo in case of failure from the OBC. As a failsafe mechanism, if one or more SDUs fail, the PWM controller will generate a PWM signal fixing the pitch of the propellers to a designated angle.
This action will transform the UAV system into a fixed-pitch proprotor but will help avoid having a system with a single point of failure that can cause the UAV to crash and possibly hurt people.

The Power Management System will be responsible for monitoring the State of Charge (SoC) of the battery, for converting the voltage from the battery to the needed voltage levels and for distributing to all components.
An Under Voltage Protection (UVP) will be also implemented to ensure that the SDU subsystem shutdowns when the battery is at a critical level.

3.4 System Behavior

In order to properly design the system behavior it was developed a system flow chart. The flow chart, represented in figure 3.2, describes the expected high level behavior of both MDU and SDU.

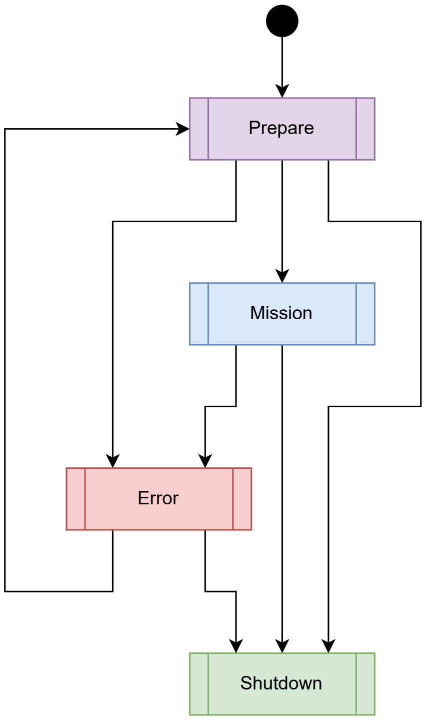

Figure 3.2: Proposed System Behavior - High Level Flow Chart

In this flow chart, there are four main tasks:

- Prepare
- Mission
- Error
- Shutdown

3.4.1 Main Device Unit Behavior

In this chapter it is described all the tasks referring to the Main Device Unit Behavior.

Prepare Task

The **Prepare** task is responsible for preparing the subsystem after activation anc checking if the system is ready for mission.

Firstly it will check the connection and the number of SDUs (this also represents the number of proprotors). In case of insufficient number of SDUs, the subsystem will enter in a state of Error and enter the **ERROR** task.

After this the subsystem checks the connection with Flight Controller (FC). If there is no response from the FC (meaning, for example, that the FC is turned off), the MDU subsystem will turn off by entering in the task **Shutdown**.

The next step is publish an heartbeat to the SDUs (using topic *mdu/heartbeat*) and to acquire Epoch time to update its own date and time (if not possible use stored date and time) and update all the connected SDUs by publishing in *mdu/date_time* topic.

Finally, in this task, the MDU will subscribe to *sdu/heartbeat* and *sdu/status*. If all the SDUs are ready, the subsystem is ready for the mission and, in this case, enter **Mission** task. Otherwise, if any or all the SDUs have any problem or can not be reached, the MDU will enter the **ERROR** task.

The MDU **Prepare** task flow is represented in figure A.1 in Appendix A.

Mission Task

In the **Mission** task the MDU will monitor the flight phase (given by the FC) and publish, accordingly, to *mdu/pitch_cmd* topic. Since each phase requires a different pitch angle, it was defined three flight phases:

- Take-Off
- Landing
- Forward Flight

And while Take-Off and Landing phase, the pitch angles are defined, fixed and equal between all proprotors, in Forward Flight phase each maneuver will require a different pitch angle and may require different pitches between each proprotor.

In parallel, the subsystem will also be publishing an heartbeat to the SDUs, check the heartbeat from th FC and monitoring the heartbeat and status of all SDUs. By constantly monitoring the heartbeat and status of all the SDUs it is possible to recognize errors in the system and try to find solutions (in **Error** task) to avoid mission failures.

The MDU **Mission** task flow is represented in figure B.1 in Appendix B

Error Task

Error task will be responsible for analyzing the error type and try to resolve the error before mission failure.

The MDU Error task flow is represented in figure C.1 in Appendix C

Shutdown Task

The MDU Shutdown task flow is represented in figure D.1 in Appendix D

3.4.2 Secondary Device Unit Behavior

In the same way, this chapter will describe all the tasks referring to the Secondary Device Unit Behavior.

Prepare Task

The SDU Prepare task flow is represented in figure E.1 in Appendix E

Mission Task

The SDU Mission task flow is represented in figure F.1 in Appendix F

Error Task

The SDU Error task flow is represented in figure G.1 in Appendix G

Shutdown Task

The SDU Shutdown task flow is represented in figure H.1 in Appendix H

TODO: CHANGE "TOPICS" TO "NODES" ?? TODO: MORE INFO

Chapter 4

Development Plan

4.1 Research Approach

To achieve the desired objectives and system requirements, the development approach will be composed of three phases: Dissertation Development, System Development, and Implementation.

During the first part of dissertation development it will be analyzed the problems with fixed-pitch propeller systems, described in the previous chapters, in UAVs to be able to find the best approach to solve the issue at hand, to define the new system requirements, determine the objectives of the proposed solution and to design the system architecture. And, to gain a better understanding of the present status of the subject, a study of the literature on related to the problem, will be conducted. In this phase, it will be also written all the steps and considerations took during the development and implementation of the solution and an analysis of the results obtained.

As we go on to the System Development phase, there will be a detailed procurement to find the most adequate components, according to the system architecture and requirements, since it is necessary to design and manufacture Pinted Circuit Boards (PCBs) for the final prototype. In this phase, the system flow charts, shown in appendix , will be modeled and validated using model checker tools like NuSMV. This step will increase the confidence in the designed firmware and validate the expected behavior of the system.

In the implementation phase, the Main and Secondary Devices wil be soldered and assembled, to, later on perform, bench and ground tests, and evaluate the developed system in comparison to predetermined goals and requirements. By documenting and analyzing the results it will be possible to make any necessary refinements to enhance performance.

4.2 Evaluation

In order to evaluate the system performance, in comparison to the requirements, the analyses will be divided in three categories.

In the Communication category, it will be analyzed the stability and the latency of the chosen communication technology.

Another category is the Pitch Angle Control in which the precision and stability of the control over the pitch angle will be evaluated. It will also be analyzed if the system has a quick response to change in flight phase and Maneuver, a quick response in error scenarios and if the fail-safe mechanism is able to set a fixed pitch angle.

The last category to be evaluated is the Firmware. In this category the model of the system flow will be validated with mathematical tools like explained before.

This way, the system will be evaluated in each subsystem and as a whole.

4.3 Timeline

In the Gantt chart (figure 4.1) all phases, described previously in the Research Approach section, were added together with multiple tasks and subtasks each with a given duration and dependencies.

The first task started is the Research Plan, part of Dissertation Development phase, and will be the starting point of future work. All the other tasks, in this phase, will be done in parallel until the end of the dissertation.

Next will be the System Development phase were hardware and firmware tasks will be made. These tasks will start mid January 2024 and end in early April 2024.

The last tasks will be from Implementation phase with tests, evaluations and refinement tasks. They will be carried out from mid March 2024 to mid July 2024.

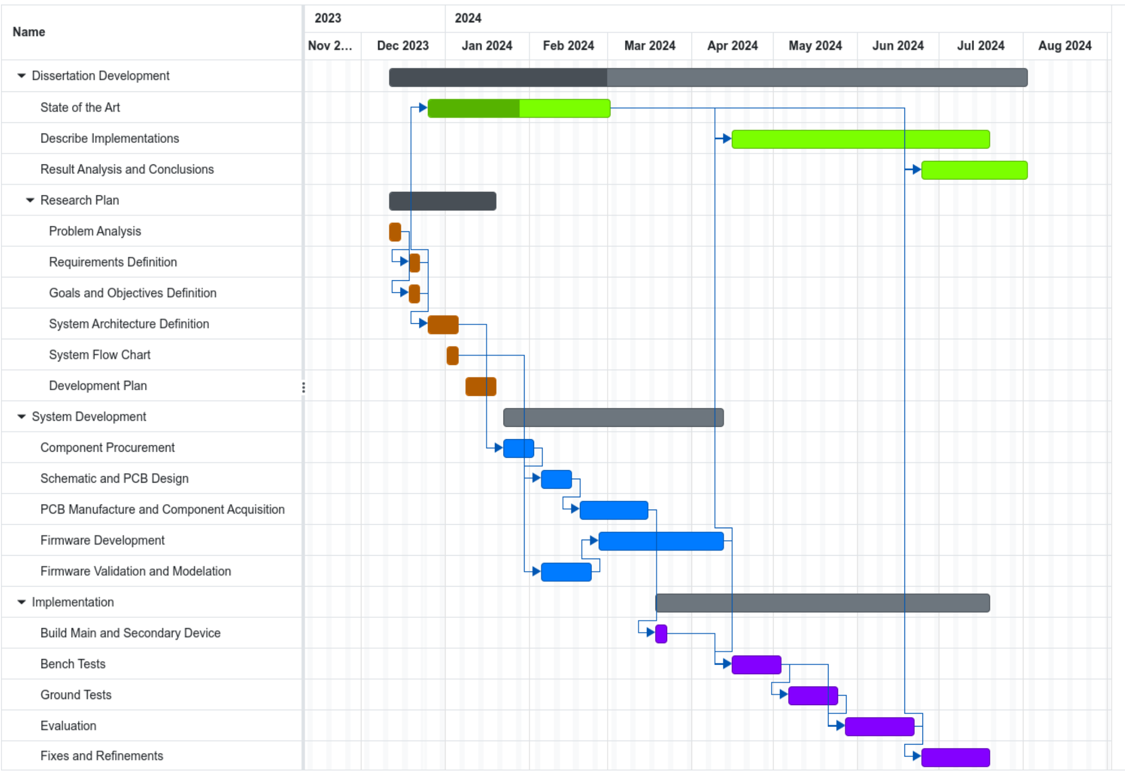

Figure 4.1: Project timeline Gantt chart

The work load will be the following:

- Dissertation Development - 170 days
- Research Plan - 30 days
- System Development - 60 days

- Hardware - 40 days
- Firmware - 50 days
- Implementation - 90 days
- Tests - 30 days

This timeline will help to ensure that all necessary activities are completed in the correct order and on time.

Bibliography

Accelerator, Academic (n.d.). *Variable Pitch Propeller Aeronautics: Most Up-to-Date Encyclopedia, News & Reviews*. en. url: `https://academic-accelerator.com/encyclopedia/variable-pitch-propeller-aeronautics`.

Chang, Ching-Wei et al. (Oct. 2020). "An Actuator Allocation Method for a Variable-Pitch Propeller System of Quadrotor-Based UAVs". In: *Sensors (Basel, Switzerland)* 20. doi: `10.3390/s20195651`.

Chipade, Vishnu S. et al. (Nov. 2018). "Systematic design methodology for development and flight testing of a variable pitch quadrotor biplane VTOL UAV for payload delivery". en. In: *Mechatronics* 55, pp. 94–114. issn: 09574158. doi: `10.1016/j.mechatronics.2018.08.008`.

Cutler, Mark (Aug. 2012). "Design and Control of an Autonomous Variable-Pitch Quadrotor Helicopter". PhD thesis.

Ferro, E. and F. Potorti (Feb. 2005). "Bluetooth and Wi-Fi wireless protocols: a survey and a comparison". In: *IEEE Wireless Communications* 12.1, pp. 12–26. issn: 1558-0687. doi: `10.1109/MWC.2005.1404569`.

Montori, Federico et al. (Oct. 2018). "Machine-to-machine wireless communication technologies for the Internet of Things: Taxonomy, comparison and open issues". In: *Pervasive and Mobile Computing* 50, pp. 56–81. issn: 1574-1192. doi: `10.1016/j.pmcj.2018.08.002`.

Reddy, S Ramanarayana (2006). "Selection of RTOS for an Efficient Design of Embedded Systems". en. In.

Zeng, Yong, Rui Zhang, and Teng Joon Lim (May 2016). "Wireless communications with unmanned aerial vehicles: opportunities and challenges". In: *IEEE Communications Magazine* 54.5, pp. 36–42. issn: 1558-1896. doi: `10.1109/MCOM.2016.7470933`.

Appendix A

Main Device Flow Chart - Prepare Task

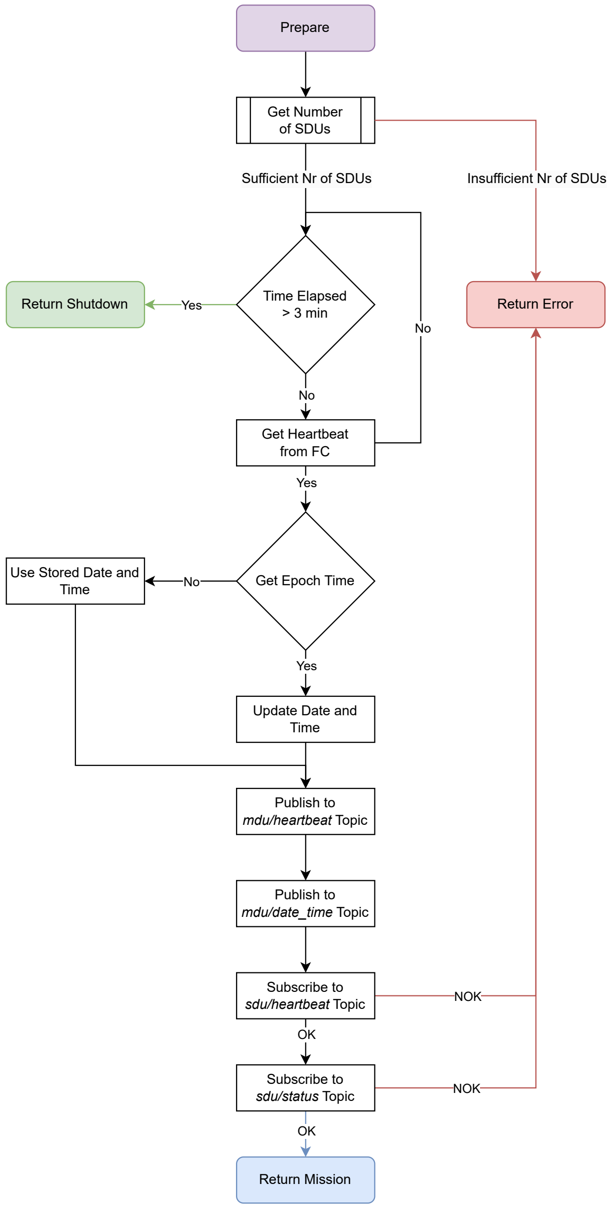

Figure A.1: Proposed System Behavior - Prepare Task Flow Chart (MDU)

Appendix B

Main Device Flow Chart - Mission Task

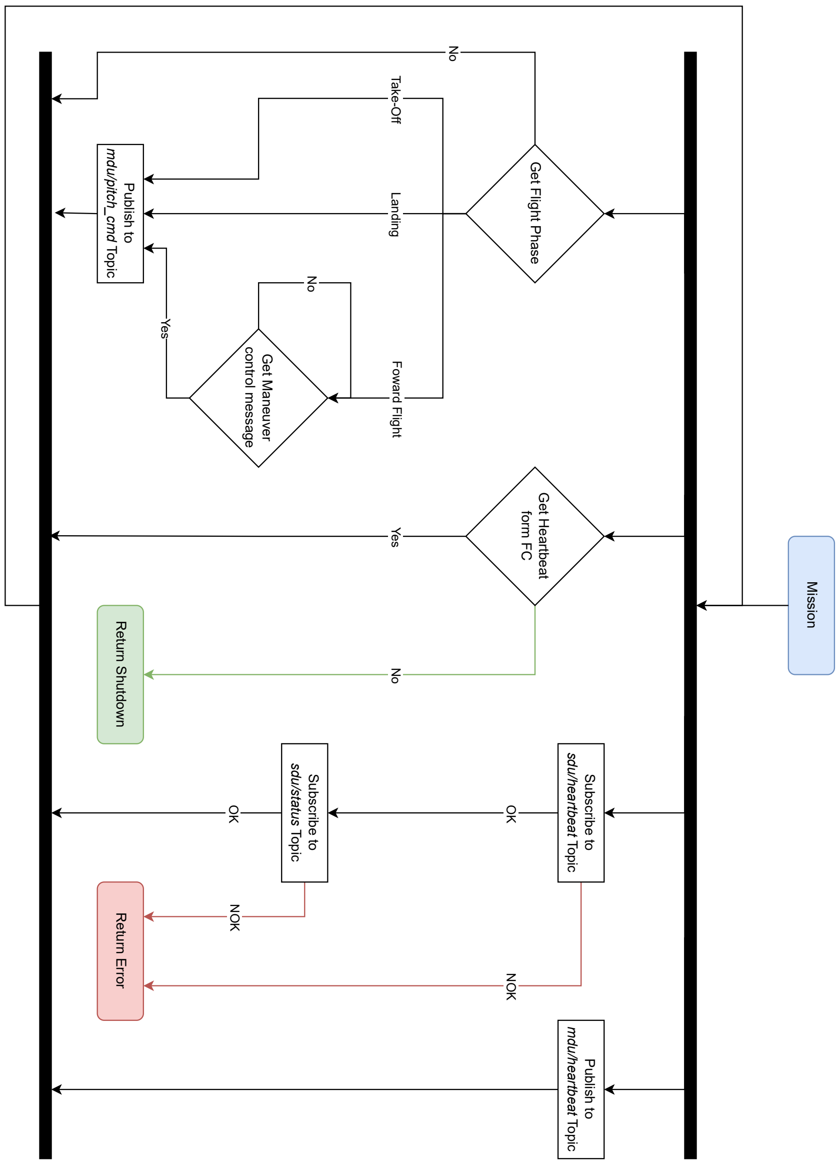

Figure B.1: Proposed System Behavior - Mission Task Flow Chart (MDU)

Appendix C

Main Device Flow Chart - Error Task

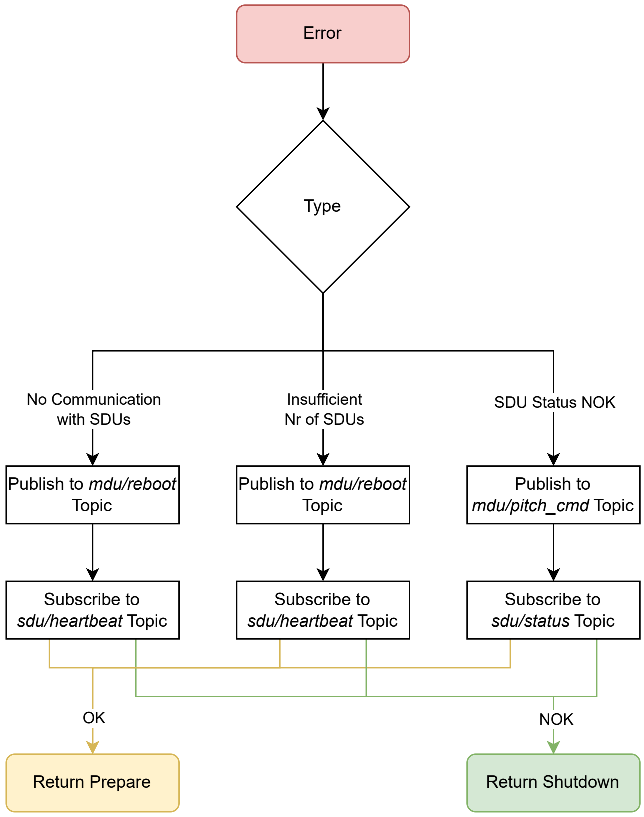

Figure C.1: Proposed System Behavior - Error Task Flow Chart (MDU)

Appendix D

Main Device Flow Chart - Shutdown Task

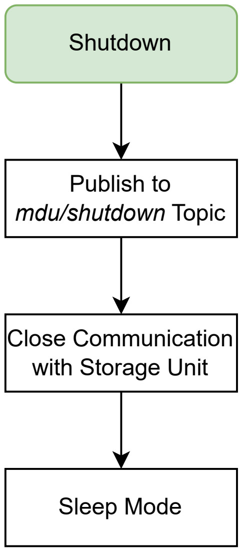

Figure D.1: Proposed System Behavior - Shutdown Task Flow Chart (MDU)

Appendix E

Secondary Devices Flow Chart - Prepare Task

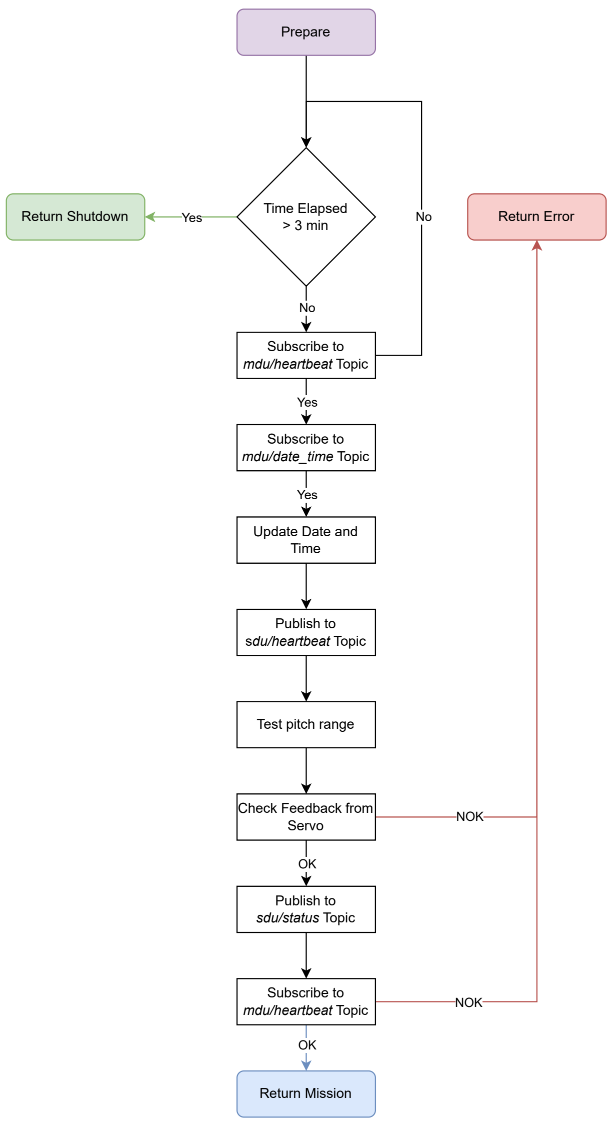

Figure E.1: Proposed System Behavior - Prepare Task Flow Chart (SDU)

Appendix F

Secondary Devices Flow Chart - Mission Task

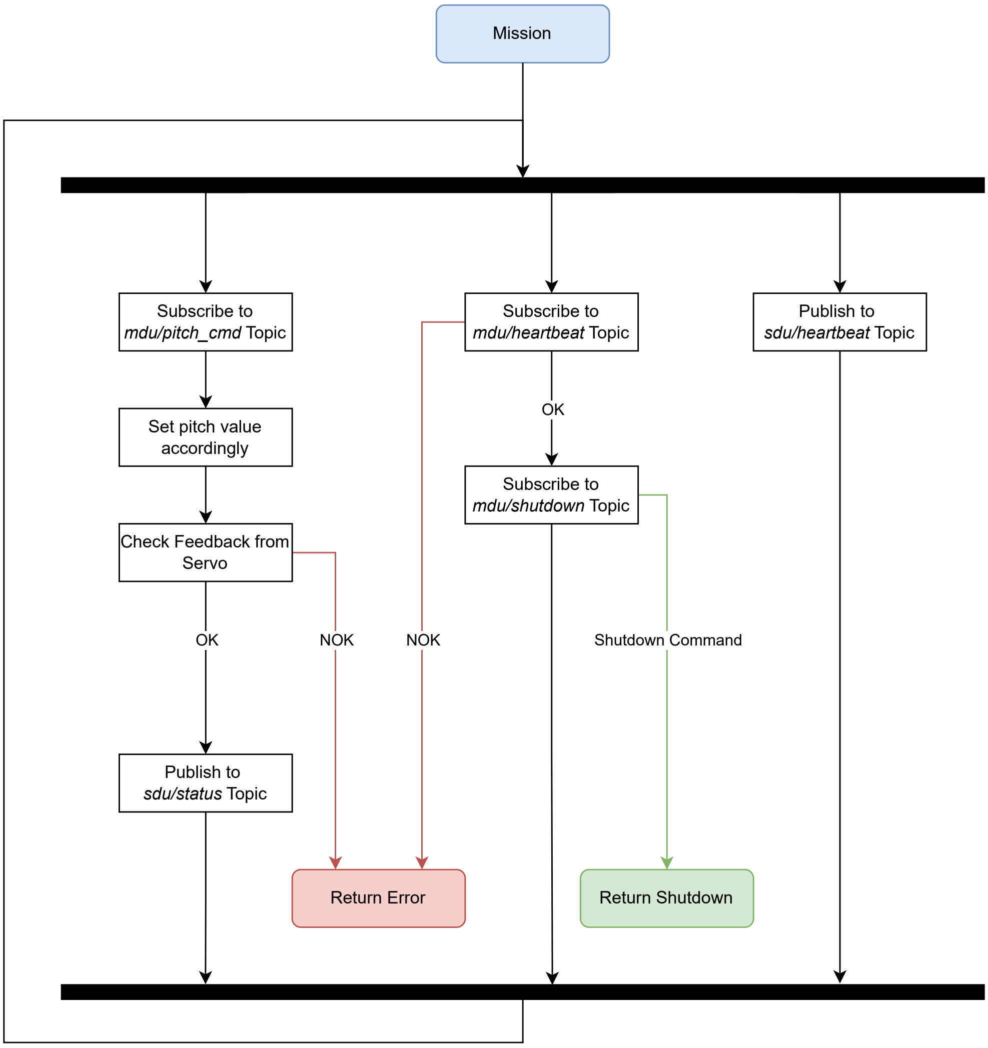

Figure F.1: Proposed System Behavior - Mission Task Flow Chart (SDU)

Appendix G

Secondary Devices Flow Chart - Error Task

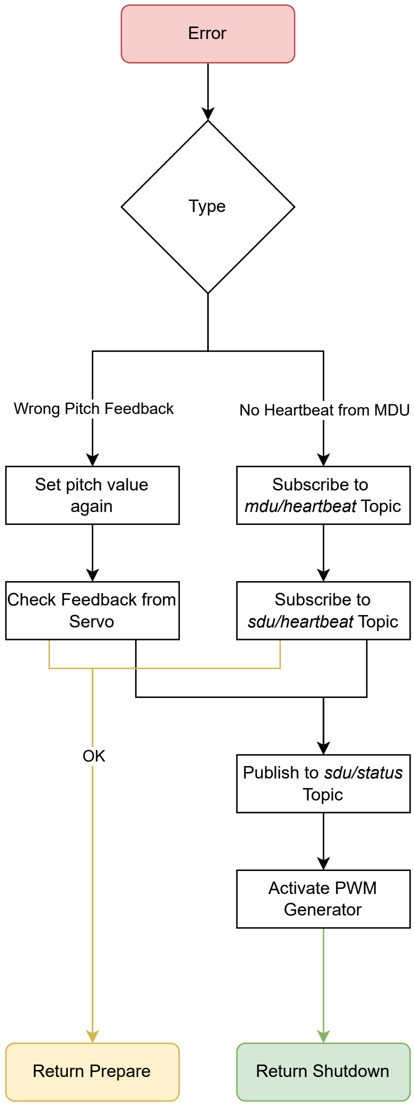

Figure G.1: Proposed System Behavior - Error Task Flow Chart (SDU)

Appendix H

Secondary Devices Flow Chart - Shutdown Task

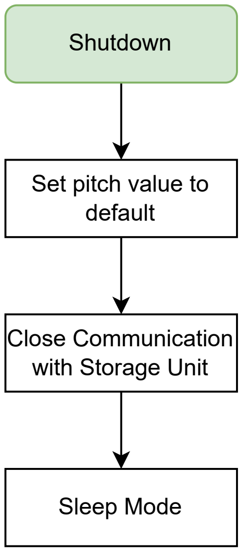

Figure H.1: Proposed System Behavior - Shutdown Task Flow Chart (SDU)