MASTER THESIS

WAPS: A smart decision tool for automated airborne measurements of aerosols in extreme environments

Supervised by: Roman Pohorsky & Prof. Julia Schmale
Extreme Environments Research Laboratory

Arthur Tabary

August 4, 2025

Abstract

Atmospheric aerosols play a crucial role in the Earth's climate system by interacting with solar radiation and cloud formation. However, their vertical distribution — a key to accurate climate modeling — remains poorly characterized, especially in remote regions like the Arctic. To address this gap, the Modular Multi-platform Compatible Air System (MoMuCAMS) was developed by the Extreme Environments Research Laboratory (EERL). It is a tethered-balloon-based instrumental platform designed for vertical in situ measurement of aerosol properties and trace gases of the lower troposphere in harsh environments. However, MoMuCAMS currently relies on manual winch operation and continuous oversight by trained personnel, limiting its scalability for long-term, distributed deployments.

This Master's thesis overcomes these operational constraints by designing, implementing, and validating a Winch AutoPilot Software (WAPS) for altitude regulation and system safety. WAPS supports three predefined flight programs—Direct Ascent/Descent, Stepped Descent, and Safe Descent—and incorporates real-time safety monitoring for high wind speeds, abnormal altitude loss, low battery voltage, and communication failure. It is implemented in Python with a modular architecture and features a Qt-based graphical interface, allowing safe and intuitive operation by a single non-expert user.

Due to airspace restrictions and logistical constraints, direct field validation was not feasible. To address this, a three-phase methodology was adopted: unit tests first verified flight logic under deterministic conditions; next, the autopilot was evaluated using live, realistic data generated by a flight simulator; finally, integration tests confirmed the reliability of communication and coordination across all system modules and under near-realistic conditions.

This work lays the foundations for autonomous deployment of MoMuCAMS in upcomming campaigns, starting with Greenfjord in August and the Tara Polar Station (TPS) expedition in late 2025. Future extensions aim to enable real-time detection of clouds and temperature inversion layers, allowing dynamic adjustment of descent step altitudes and durations and further reducing the need for operator intervention.

Abstract (French)

Les aérosols atmosphériques jouent un rôle crucial dans le système climatique terrestre en influençant le rayonnement solaire et la formation des nuages. Cependant, leur distribution verticale, essentielle pour une modélisation climatique précise, reste mal caractérisée, en particulier dans les régions éloignées comme l'Arctique. Pour combler ce retard, le Modular Multi-platform Compatible Air System (MoMuCAMS) a été développé par le Extreme Environments Research Laboratory (EERL). Il s'agit d'une plate-forme instrumentale accrochée à un balloon captif, conçue pour mesurer verticalement et in situ les propriétés des aérosols et les gaz à l'état de traces dans la troposphère même des environnements difficiles. Cependant, le MoMuCAMS dépend actuellement de l'utilisation manuelle d'un treuil et d'une surveillance continue par un personnel formé, ce qui limite son déploiement à plus grande échelle.

Ce projet de master répond à ces contraintes opérationnelles en concevant, en implémentant et en validant un pilote automatique intelligent, nommé WAPS, qui régule automatiquement l'altitude tout en assurant la sécurité du système et des opérateurs à terre. WASP prend actuellement en charge trois programmes de vol: ascension/descente directe, descente par paliers et descente d'urgence. Il surveille en temps réel plusieurs paramètres de sécurité comme la vitesse du vents, les pertes d'altitude anormales, la tension de la batterie et les problèmes de communication.

En raison des restrictions de l'espace aérien et des contraintes logistiques, il n'a pas été possible de valider le dispositif en conditions réelles. Pour pallier cela, une méthodologie en trois phases a été mise en place : d'abord, des tests unitaires ont permis de vérifier la logique de vol dans des conditions déterministes ; ensuite, WAPS a été testé en temps réel avec des données réalistes générées par un simulateur de vol ; enfin, des essais d'intégration ont confirmé la fiabilité de la communication et la coordination entre tous les modules du système.

Ce travail établit les bases d'un déploiement autonome du MoMuCAMS lors des prochaines campagnes, à commencer par Greenfjord en août et l'expédition Tara Polar Station (TPS) à la fin de 2025. De futurs développements viseront à intégrer la détection en temps réel des nuages et des couches d'inversion de température, afin de permettre une adaptation automatique de l'altitude et de la durée des paliers de descente sans l'intervention d'un opérateur.

Contents

List of Abbreviations

ABL Atmospheric Boundary Layer

CCN Cloud Condensation Nuclei

EERL Extreme Environments Research Laboratory

GUI Graphical User Interface

SIL Software-in-the-Loop

MoMuCAMS Modular Multi-platform Compatible Air System

PBL Planetary Boundary Layer

PCA Principal Component Analysis

RF radio-frequency

RH relative humidity

SBI Surface-Based inversion

TPS Tara Polar Station

WAPS Winch AutoPilot Software

Acknowledgements

I am deeply grateful to Professor Julia Schmale, who first supported me during my time with the Sailowtech association in developing an ambitious expedition to Greenland. Although that project was ultimately cancelled, her trust did not waver. On the contrary, she turned that setback into an opportunity by offering me this exciting and challenging master's thesis. More recently, she has continued to believe in me offering me the incredible opportunity to join an expedition next December to field-test the control tool developed in this master thesis.

I am also very grateful to Roman Pohorsky for his valuable time and his expertise throughout the project. Your availability, responsiveness, and invaluable guidance—right up to the very last minute-made a real difference.
Thanks to Lionel Favre as well for his technical expertise, help with setting up physical tests, and above all, his great energy and the good moments we have shared (and those still to come aboard the Tara Polar Station!).
More broadly, I want to thank the entire team at the EERL lab for their warm welcome, thoughtful advice, and the many enriching discussions along the way.

And last but certainly not least, heartfelt thanks to my family and friends (especially Tania, Matthieu, Dédé and Pierrot) for their unwavering support, the shared laughs, and for bearing with me through the ups and downs.

1 Introduction

Atmospheric aerosols play a critical role in the Earth's climate system due to their interactions with solar radiation and clouds. These particles vary in size, composition, and origin. They influence climate both directly—by scattering or absorbing solar radiation—and indirectly—by serving as cloud condensation or ice-nucleating particles. This affects cloud formation and properties (Boucher et al., 2013; Haywood and Boucher, 2000; Seinfeld and Pandis, 2016). The vertical distribution of aerosols within the Atmospheric Boundary Layer (ABL) and lower free troposphere is particularly valuable, as it determines their impact on radiative forcing and cloud processes (Carslaw, 2022). For instance, the altitude of absorbing aerosols can lead to localized heating, changing atmospheric stability and potentially causing cloud dissipation (Samset et al., 2013). In polar and alpine regions, where stable boundary layers are common, aerosols often exhibit pronounced vertical stratification, making ground-based measurements unrepresentative of conditions aloft (Brock et al., 2011; Creamean et al., 2021; Jacob et al., 2010). This vertical heterogeneity complicates the accurate representation of aerosol processes in numerical models, contributing to significant uncertainties in anthropogenic radiative forcing estimates (IPCC, 2023). Understanding the vertical distribution of aerosols is thus essential for improving climate models and constraining their climatic effects, particularly in extreme environments like the Arctic, where observational data are sparse (Koffi et al., 2016; Sand et al., 2017). To close these knowledge gaps—especially in remote, vertically stratified environments—new observational tools are essential.

Tethered balloon systems enable the acquisition of high-resolution vertical profiles over extended durations, making them particularly well suited for investigating aerosol transport, transformation, and aerosol–cloud interactions in complex atmospheric environments (Pohorsky et al., 2024). In recent years, several tethered-balloon platforms have been developed for vertical atmospheric profiling, targeting aerosols and meteorological parameters in various contexts (Ferrero et al., 2016; Mazzola et al., 2016; Canut et al., 2016; Pilz et al., 2022). However, these systems were generally designed around fixed measurement goals and offer limited flexibility in modifying the instrument payload. In this context, the Modular Multi-platform Compatible Air System (MoMuCAMS)—developed by the Extreme Environments Research Laboratory (EERL)—introduces a modular approach tailored for in situ measurements of aerosol properties and trace gases in the lower troposphere and extreme environments (Pohorsky et al., 2024). The EERL's mission is to advance the understanding of climate-related processes in extreme environments, such as the Arctic, by studying atmospheric composition, particularly aerosols, and their interactions with the cryosphere, ocean, land, and human activities. Designed for operations in windy and cold conditions, MoMuCAMS is mounted on a Helikite (a helium-filled tethered balloon) and integrates a suite of instruments within a lightweight, insulated enclosure. The system enables various configurations for measuring variables such as the aerosols' size distribution, their optical properties, chemical composition in combination with trace gas concentrations, and meteorological variables.

While the MoMuCAMS system has demonstrated its effectiveness in addressing key questions related to the vertical distribution of aerosols, current measurements are limited to short-term campaigns. As part of its long-term vision, the EERL aims to enable broader and more systematic deployments of tethered-balloon systems in polar regions to collect standardized, high-resolution atmospheric data on a regular basis. Integrating such data into climate models would significantly improve constraints on the role of aerosols in these environments.

To fully realize the scientific potential of MoMuCAMS and expand its usability, current operational limitations must be addressed. At present, the system requires multiple trained operators for deployment and flight management, which limits its scalability. A critical component is the winch, which controls the tether and hence the balloon's altitude. Currently, this winch is manually operated, requiring constant human supervision to manage ascent and descent.

The primary objective of this Master's thesis is to simplify and partially automate the operation of MoMuCAMS by developing a Winch AutoPilot Software (WAPS) capable of executing predefined flight plans autonomously, while ensuring the safety of both the system and ground personnel. Ultimately, the system should operate without human intervention between takeoff and landing. Furthermore, the autopilot should be able to detect scientifically relevant features in the atmospheric profile—such as cloud layers, pollution plumes, or the top of the atmospheric boundary layer (ABL)—and adapt the flight plan in real time. This automation represents a key step toward the wider deployment and simplification of the system.

A first test deployment of the automated system is planned for August 2025 aboard the Forel boat during Greenfjord expedition. This will be followed by a long-term deployment on Tara Polar Station (TPS) in December 2025. The TPS deployment is expected to operate continuously over several years without any on-site personnel from the EERL, further underscoring the need for an intuitive, robust, and autonomous flight autopilot for MoMuCAMS.

2 System Overview

The MoMuCAMS system architecture consists of three core modules: the **Flight Computer**, the **Winch Controller**, and the **Ground Computer**. Each module functions independently yet communicates continuously to enable coordinated operation of the tethered balloon system. The overall configuration and data flow between these components are illustrated in Figure 1.

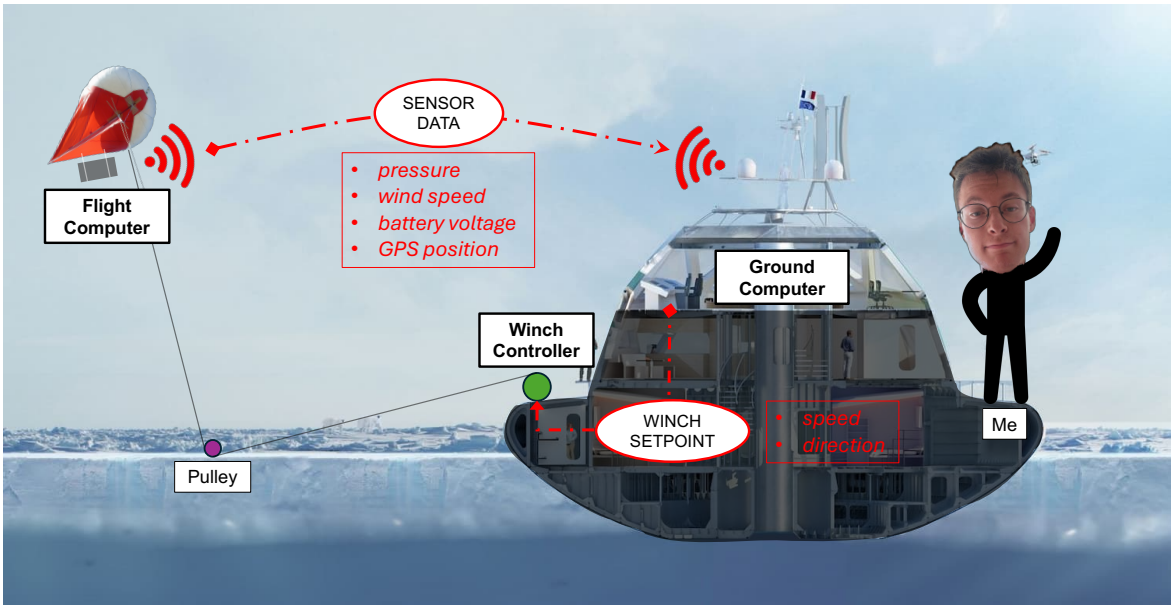


Figure 1: System configuration for the TPS. The flight computer transmits telemetry via radio-frequency (RF) to the ground computer, which processes the data and sends speed commands via Ethernet to WAPS.

The **Flight Computer**, implemented on a Raspberry Pi single-board computer, is mounted onboard the Helikite payload. It manages an array of onboard sensors to collect telemetry data such as wind speed, atmospheric pressure, battery voltage, temperature, and GPS location. This data is logged locally and transmitted in real time to the ground station via a wireless radio frequency (RF) link. Serving as the system's data acquisition unit, the flight computer provides essential atmospheric and system status information.

The **Winch Controller** is an industrial-grade programmable logic controller (Siemens LOGO!8), integrated within the winch hardware. The MoMuCAMS platform is connected to the winch by a high-strength Dyneema cable, which physically links the Helikite to the ground system. By controlling the uncoiling and reeling in of this cable, the winch regulates the balloon's altitude. The logic controller manages the winch motor's operation, continuously monitors cable tension (not implemented yet) and drum rotation speed, and supports both manual and autonomous operation modes. While it can execute real-time commands from the ground computer during autonomous flights, it can also be immediately switched to manual mode at any time for safety reasons. This manual override allows operators to bypass the autopilot in the event of system faults or unexpected behavior, ensuring full control and safe operation under all circumstances.

The **Ground Computer** is a terrestrial PC hosting the control software developed for this thesis. It receives telemetry streams from the Flight Computer via the RF link, processes the data and sends real-time speed and direction commands to the winch controller over a wired Ethernet (RJ45) connection. This module bridges the data acquisition onboard the Helikite and the winch actuation on the ground, enabling execution of flight plans without any human intervention.

Prior to this thesis work, winch control was entirely manual: operators adjusted the tether length based on telemetry visualized on the ground computer. The winch controller and ground computer operated independently, without communication. The automation introduced here enables the ground computer to command the winch controller directly, reducing operator workload and enhancing safety.

3 Smart Winch AutoPilot:

Building upon the established hardware architecture and communication framework, the automation of tethered balloon flight operations relies critically on the intelligent winch autopilot. This software component

embodies the system's decision-making capabilities, executing predefined flight plans autonomously while continuously monitoring safety parameters to protect both the instrument payload and surrounding environment. The following section presents a comprehensive description of the autopilot's design, including supported flight programs, safety mechanisms, control logic, and user interface features.

3.1 Flight Programs

The flight autopilot currently supports three mission programs, each designed to address specific atmospheric profiling objectives. Two of these programs—Program A and Program B—can be selected by the operator through the autopilot interface. Each program includes configurable parameters that determine its flight behavior. Once a program is selected and its parameters are defined, the mission can be launched. Importantly, the parameters may also be modified during the mission in real time to accommodate changing observational needs or environmental conditions.

Figure 2 provides an overview of the trajectory associated with each program and illustrates their configurable parameters.

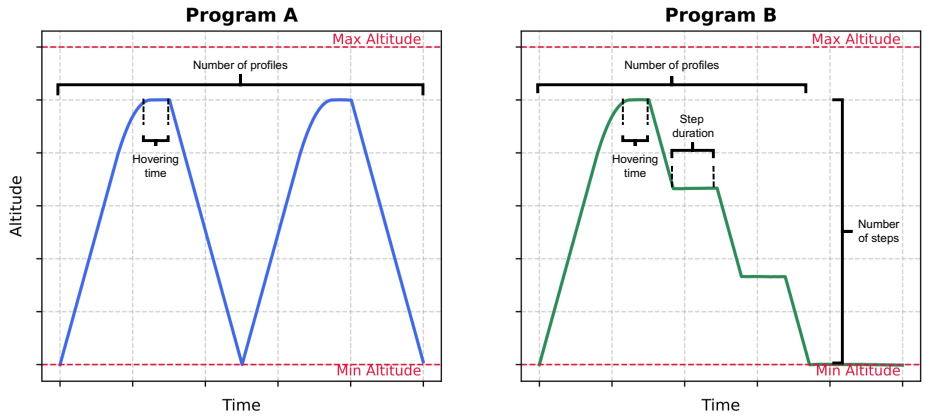


Figure 2: Overview of the flight trajectories and configurable parameters associated with Program A (Direct Ascent/Descent) and Program B (Ascent with Stepped Descent).

Program A: Direct Ascent/Descent. This program performs a continuous vertical profiling of the atmosphere, offering a snapshot of its thermodynamic structure. It is particularly well suited for detecting aerosol layers and analyzing their alignment with temperature and pressure stratification. When repeated over time, it further enables the observation of temporal changes in atmospheric conditions throughout the duration of the flight.
Operators can configure the hovering time at top of the profiles, as well as the number of profiles to be completed during the mission.

Program B: Ascent with Stepped Descent. This program begins with a direct ascent to obtain an initial vertical profile and detect features of scientific interest, such as cloud layers, pollution plumes, or temperature inversions. Following the ascent, the system performs a stepped descent, during which the Helikite hovers at several discrete altitudes for defined durations. This allows enhanced vertical resolution and data quality by enabling extended sampling at specific altitudes. The approach is particularly valuable for improving the statistical robustness of particle count measurements and other low-signal observations. It is also well suited for missions involving aerosol collection on physical substrates for offline chemical analysis, where longer sampling durations are necessary to accumulate sufficient particle mass. In such cases, the number of steps can be reduced in favor of extended hovering periods.
Configurable parameters for this program include the hovering time at top of the profiles, the number of profiles, the number of descent steps, and the duration of each step.

Emergency Program: Safe Descent. A third program, *Safe Descent*, serves as a critical safety mechanism. When triggered, it overrides any ongoing flight program and initiates an immediate descent to the original altitude. To ensure a rapid return, the winch operates at maximum descent speed. This program can be triggered automatically by WAPS in response to safety violations, as detailed in Section 3.2, or manually by the operator

at any point during the flight.

When a flight program is launched, the current altitude of the Helikite is recorded and used as the minimum altitude for the mission. This value defines the bottom of the vertical profiles (see Figure 2) for the remainder of the flight. As a result, it is essential that operators manually perform the initial ascent to establish a safe starting altitude. The autopilot is not designed to handle takeoff or landing procedures

A maximum altitude can also be specified by the operator before or during the mission. This limit is useful when the altitude of a layer of interest is known—such as an expected inversion or pollution plume—or when airspace regulations impose a ceiling on flight altitude. However, the Helikite may stabilize and stop ascending before reaching this maximum (see Figure 2), depending on the equilibrium between its buoyant lift and the cumulative gravitational force of the system (including the balloon, the MoMuCAMS payload, and the deployed Dyneema cable).

3.2 Safety Monitoring and Emergency Conditions

While the objective of the winch autopilot is to optimize data collection through intelligent decision-making based on live measurements from MoMuCAMS, operational safety remains the foremost priority—especially in the absence of direct human supervision. To guarantee safety throughout a flight program (see Section 3.1), the autopilot includes a set of safety checks that are continuously executed during the mission. If the system detects conditions that pose a threat to the integrity of the Helikite, the payload, or third parties, the safety protocol is triggered. This protocol leads to immediate abort of the mission and launch of the *Safe Descent* program (see Section 3.1), if no human intervention is detected. The safety considerations and their implementation are detailed below. The corresponding safety parameters along with their default values are presented in Table 1.

High Wind Speeds. Excessive wind speeds can affect the operational stability and safety of the Helikite system. High winds can generate significant tension on the tether, increasing the mechanical stresses likely to cause structural failure. In addition, they can cause significant horizontal drift of the Helikite, potentially taking it out of its safe operating area. For example, when the system is deployed from the TPS, the tether could become entangled in the station's antennas, damaging them. Strong winds can also compromise the stability of the balloon, causing vibrations that degrade the performance and accuracy of the payload's on-board instruments.
To mitigate these risks, wind is continuously monitored and evaluated. Specifically, the autopilot calculates the average wind speed over a predefined time window, T_{wind} (default: 15 s), to capture sustained wind behavior rather than transient gusts. This averaging is performed using a moving average filter, which smooths out short-term fluctuations in the wind data and reduces the likelihood of false alarms triggered by momentary spikes. If the computed average wind speed exceeds a critical threshold W_{max} (default: 15 m/s), the wind conditions are flagged as dangerous, triggering the safety protocol.

Unexpected Altitude Loss. A rapid and unplanned altitude loss of the platform during periods when the tether length is constant or increasing, presents a significant operational risk. Such altitude drop may be indicative of underlying issues such as helium leakage or balloon icing, which can compromise the buoyancy and lift capacity of the Helikite. Ultimately, this could lead to a loss of control or even a crash, damaging both the Helikite and the on-board instruments.
To mitigate these risks, the rate of altitude change is continuously monitored and evaluated over a time window T_{loss} (default: 30 s). If its moving average exceeds a predefined maximum threshold, ν_{max} (default: 0.2 m/s), the safety protocol is triggered.

Low Battery Voltage. If the battery voltage falls below a critical threshold, the Helikite onboard sensors are no longer adequately powered, rendering subsequent scientific measurements ineffective and the remainder of the mission unproductive. Moreover, the loss of live telemetric data effectively blinds the autopilot system, which relies on continuous sensor input to supervise the flight program and ensure safe operation. As a result, both the program supervision and the safety mechanisms become unreliable, severely compromising the autonomy and integrity of the system.
To prevent this scenario, the system continuously monitors the battery voltage. If the voltage remains below a critical threshold V_{min} (default: 19.5 V) for a period longer than T_{bat} (default: 60 s), the battery is considered to be in a low state and thus the safety protocol is triggered.

Radio Signal Loss. As described in Section 2, communication between the ground computer—running the autopilot software—and the onboard flight computer is established via a RF link using directional antennas mounted on the MoMuCAMS. However, the link is susceptible to disruptions due to imperfect antenna alignment and environmental factors such as atmospheric attenuation. For example, cloud layers, which contain high concentrations of water droplets, can degrade signal quality. These conditions may lead to packet loss or corruption. Since the current version of the ground software discards corrupted packets, any loss or corruption directly results in missing telemetry data. When too many packets are lost, the autopilot system becomes effectively blind, unable to monitor the platform or respond to critical events, thereby compromising mission supervision and safety.
To detect such situations, the system continuously evaluates the quality of the radio link by monitoring the packet loss ratio. If the proportion of lost packets exceeds a specified threshold R_{max} (default: 1) for a sustained period T_{signal} (default: 60 s), the communication is deemed unreliable. In response, the system initiates the safety protocol.

Safety Protocol. When any of the above safety conditions are detected, the safety protocol is triggered. The system immediately alerts the operator via a dialog box on the ground computer (see Figure 3) and an audible alarm on both the winch and the ground computer. The operator then has T_{popup} seconds (default: 30 s) to either acknowledge and dismiss the alert or initiate the *Safe Descent* program. If the operator selects the latter—or if no action is taken within the allotted time—the autopilot automatically engages the *Safe Descent* procedure to return the Helikite to its predefined safety altitude (see Section 3.1). If the operator chooses to ignore an alert, that specific safety condition is temporarily disabled for a configurable duration T_{disable} (default: 120 s): it will not trigger any further popups or safety procedures during this period.
At any point during the execution of the *Safe Descent* program, the operator retains the ability to override the system. This can be done either by cancelling the emergency procedure through the control software or by switching to manual mode directly via the winch controls.

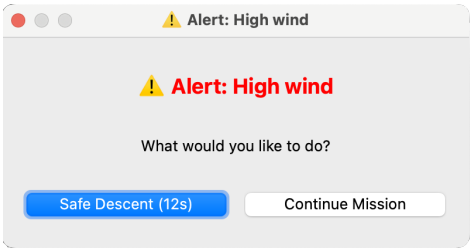


Figure 3: Example of an autopilot software dialog box that appears when a safety condition is met (in this case, high wind).

All safety-related variables and their default values are listed in Table 1. These parameters can be adjusted by the operator via the configuration file `GroundComputer/controllerParameters.yaml`.

Table 1: Safety thresholds and monitoring parameters.

Symbol	Condition	Default Value
W_{max}	Max average wind speed	15 m/s
T_{wind}	Wind averaging duration	15 s
v_{max}	Max descent rate	0.2 m/s
T_{loss}	Altitude monitoring window	30 s
V_{min}	Minimum battery voltage	19.5 V
T_{bat}	Battery monitoring duration	60 s
T_{signal}	Max signal outage duration	60 s
R_{max}	Ratio of loss packets (1 = all lost)	1
T_{popup}	Popup duration	30 s
T_{disable}	Popup disable time	120 s

3.3 AutoPilot Architecture and Implementation

The winch autopilot is implemented in Python 3.13 using an object-oriented architecture. It is fully contained within the script `GroundComputer/autopilot.py`, independent from the rest of the ground computer software. This self-contained design promotes ease of testing, maintainability, and portability.

A key architectural principle is the clear separation between **safety supervision** and **mission logic**. This modularity enhances fault tolerance, simplifies debugging, and allows new scientific routines or safety policies to be integrated with minimal disruption. While mission logic is encapsulated in the abstract `Program` class and its subclasses, all safety-related operations are managed within the `Autopilot` class (see Figure 4).

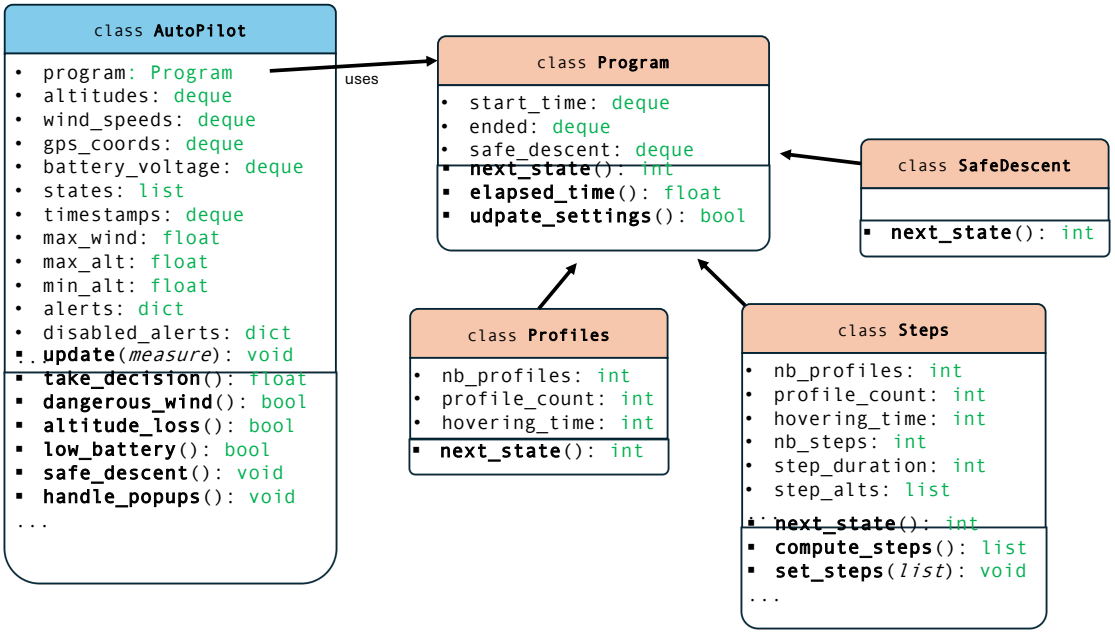


Figure 4: UML diagram illustrating the class structure and relationships of the autopilot software, highlighting the separation of safety supervision and mission logic.

At the core of the autopilot software is the `Autopilot` class, which determines winch actions—uncoil, hold, or coil—based on real-time sensor data from the flight computer. This process is handled entirely by the `take_decision()` method, which is called upon receipt of new sensor measurements (nominally once per second). Figure 5 summarizes the internal logic of `take_decision()`, which follows a fixed four-stage sequence:

1. **Update Sensor Buffers:** Incoming sensor data—altitude, wind speed, GPS position, battery voltage—is appended to rolling buffers implemented with fixed-length `deque` structures. Each buffer retains up to 600 samples, representing a 10-minute history at a 1 Hz sampling rate. This size balances memory usage and temporal resolution, ensuring sufficient data for robust sliding-window analysis to detect trends and anomalies. The buffer size can be extended for longer-term monitoring as needed.

2. **Safety Checks:** WAPS executes a series of safety checks to identify hazardous conditions such as strong wind (`dangerous_wind()`), unexpected altitude loss (`altitude_loss()`), low battery voltage (`low_battery()`), or signal loss (`signal_loss()`). When a safety condition is detected, the autopilot records the event by adding an entry to the `alerts` dictionary with the alert name and timestamp. If the operator chooses to ignore the alert, it is added to the `disabled_alerts` dictionary with a timestamp, preventing the safety procedure from being triggered by this condition (greater details in Section 3.2).

3. **Stability Check:** WAPS assesses whether the balloon has stabilized in altitude—typically near its equilibrium height—using the `is_altitude_stable()` method. This method computes the mean vertical velocity over a sliding window and compares it against a stability threshold. Both the size of this window and the stability threshold are configurable parameters defined in the `controllerParameters.yaml` file. When stability is detected, the system prompts the operator to choose between continuing ascent or holding position. If *ascent* is chosen, the autopilot transitions to the ascent state (state 1) commanding the winch to uncoil cable; if *hold* is chosen or no response is received within a timeout, it remains in the hold state (state 0), keeping the winch stationary.

The Helikite may reach a state of equilibrium before reaching the programmed maximum altitude limit. It is therefore essential to detect this stabilization promptly in order to prevent further uncoiling of the tether. Once the helium lift balances the combined weight of the Helikite, payload, and tether, any additional cable deployment is detrimental. Excess slack in the tether can lead to balloon drift, mechanical entanglement on the winch drum, and increased risk of structural damage to the system.

4. **Mission Logic Execution:** If no safety event occurs, the autopilot delegates decision-making to the active mission program. Mission logic is implemented in subclasses of the abstract `Program` class (`Profiles`, `Steps` or `SafeDescent`), each defining a `next_state()` method. This method computes the next winch action—ascend (+1), hold (0), or descend (−1)—based on current and previous sensor inputs. Notably, the `Steps` subclass includes additional methods such as `compute_steps()`, which calculates linearly spaced altitude steps between a predefined minimum altitude `min_alt` and the top altitude `max_alt` once the Helikite reaches the profile's summit. The operator may either accept these automatically generated step altitudes or modify them manually via the AutoPilot user interface, which uses the `set_steps()` to apply the selected configuration (see Figure 4).

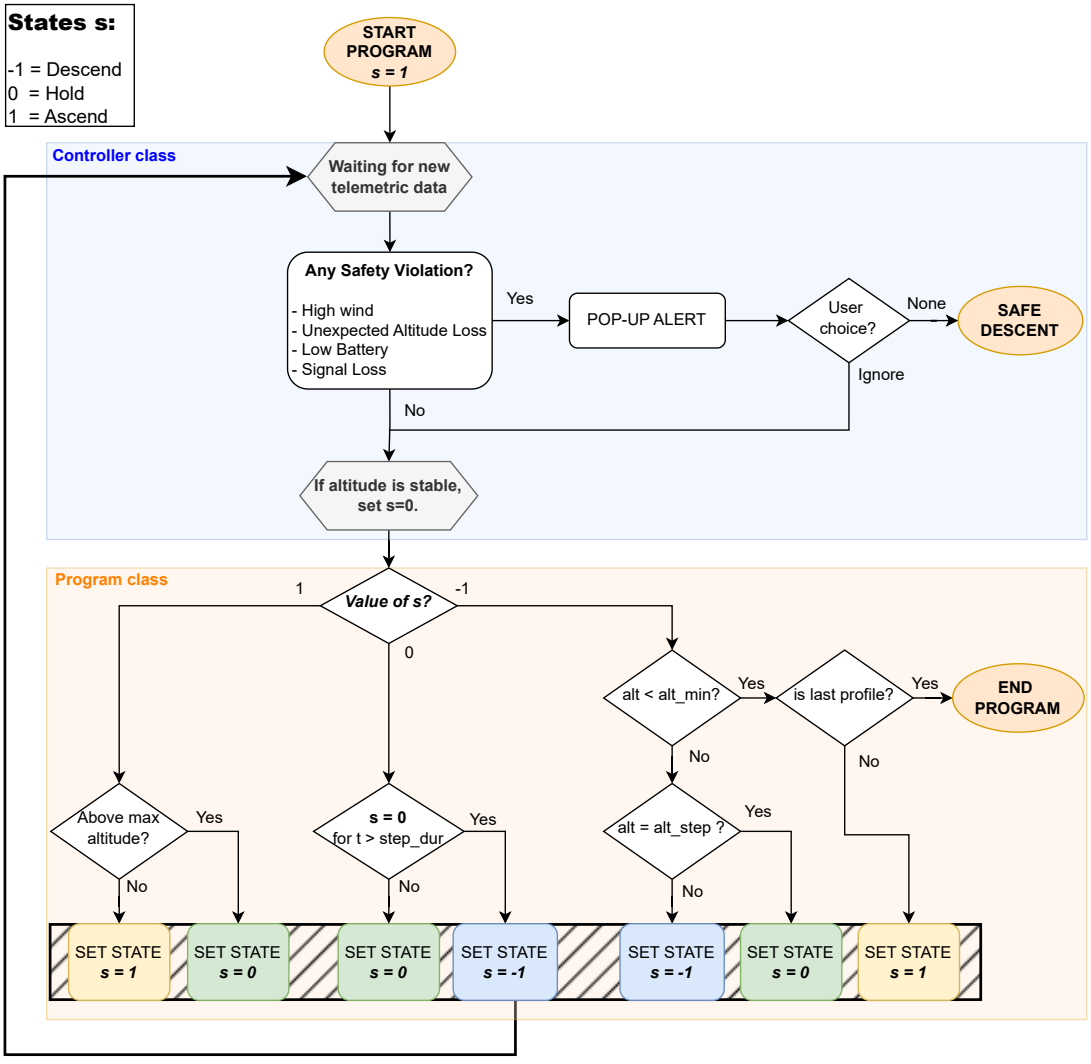
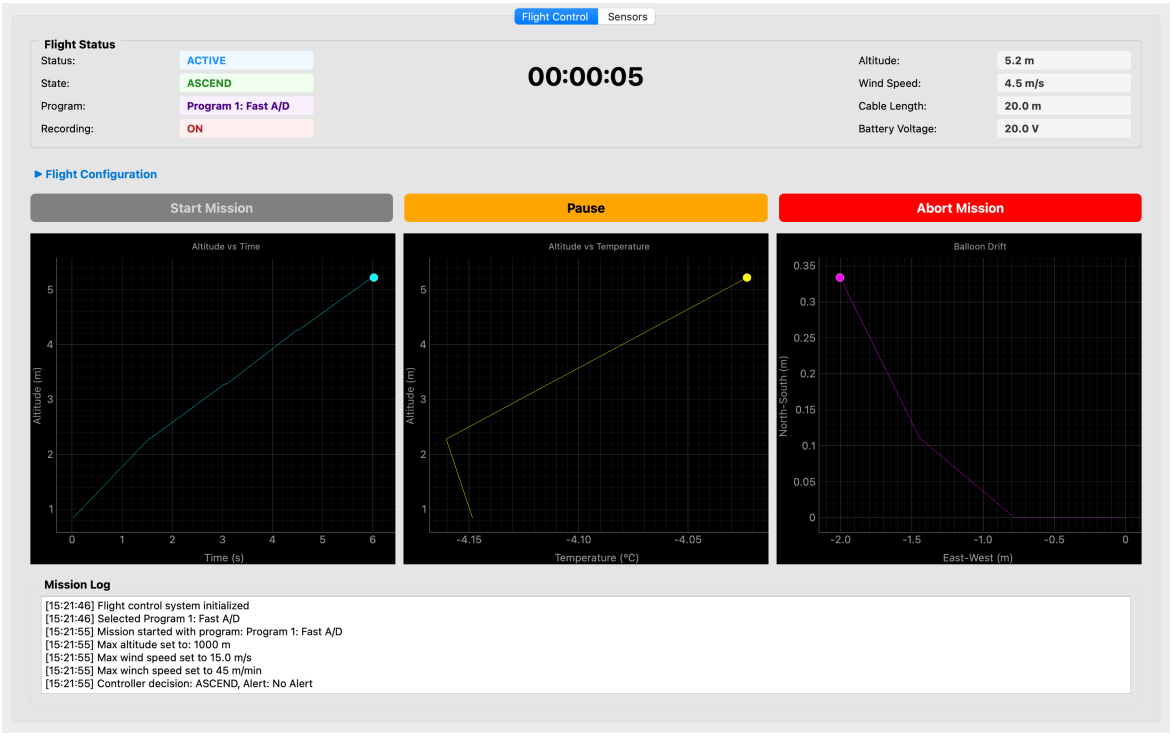


Figure 5: Flow diagram summarizing the internal logic of the `take_decision()` method in the `Autopilot` class.

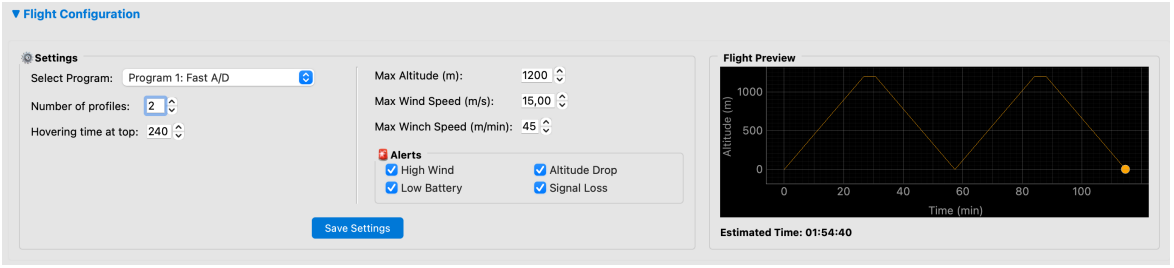
This modular structure enables straightforward extensibility: to add a new safety condition, developers implement a specific method within the `Autopilot` class and invoke it within the `safety_checks()` sequence. To add a new mission program, one creates a subclass of `Program` and defines the mission-specific `next_state()` polymorphic method. This approach allows WAPS to evolve without modifying core autopilot logic, maintaining robustness and clarity.

3.4 Graphical User Interface

The autopilot's Graphical User Interface (GUI) was developed in Python using the Qt framework and has been integrated into the existing control software of the ground computer. It has been designed with a strong emphasis on usability and clarity, enabling scientists with no prior software expertise to operate the system (see example screenshot in Figure 6).



(a) The main interface displaying mission status, real-time telemetry, and live plots for situational awareness.



(b) The Flight Configuration tab, allowing users to select programs and set operational parameters.

Figure 6: Screenshot of WAPS's GUI.

The upper section **Flight Status** of the interface (Figure 6a) presents all critical mission information. On the left, it indicates the current status of WAPS (active or inactive), the operational mode of the winch (ascending, hovering, or descending), the selected flight program (A, B, or Safe Descent), and the recording status of MoMuCAMS sensor suite. Centered in this block is the elapsed mission time since launch. On the right, a summary of real-time telemetry includes altitude (calculated from barometric pressure), wind speed measured via the onboard anemometer, uncoiled tether length estimated from winch revolution counts, and the battery state of charge (i.e., voltage) measured by the flight computer.

The **Flight Configuration** tab (Figure 6b) provides control over mission planning. It is divided into three vertical sections: on the left, operators select the flight program and configure its parameters; in the center, they define general flight options such as maximum altitude or number of profiles; and on the right, a schematic preview displays the expected flight trajectory.

Finally, the **plots section** of the interface (Figure 6a) presents three real-time graphs designed to enhance situational awareness and support scientific objectives. The altitude-versus-time plot tracks the Helikite's vertical trajectory, providing oversight during profiling missions. The temperature-versus-altitude plot helps identify inversion layers quickly. Meanwhile, the drift plot displays horizontal motion, allowing operators to

detect dangerous drift—especially toward the control station—where the balloon's tether could entangle or damage nearby elevated structures such as antennas.

Complementing these visual tools, the interface also maintains a live **mission log** that records every decision made by the autopilot and all manual inputs from the operator. A more detailed version of this log is saved to disk at the end of each flight, capturing the full sequence of sensor inputs, system states, and operator interactions. This persistent record enables developers to reconstruct and analyze the mission timeline in detail, providing a powerful tool for debugging and verifying system behavior post-flight.

The user interface is explored in greater detail in the software user manual, available in the appendix A.

4 Testing Methodology

A major constraint during this project was the inability to conduct field testing. Airspace regulations and logistical limitations prevented Helikite flights within the thesis timeline. The first full-scale deployment is planned for the Greenfjord expedition in August 2025, where the Helikite will be launched from a sailboat in Southern Greenland.

To address this constraint, a multi-phase testing strategy was implemented to evaluate WAPS's performance and reliability under simulated mission conditions. Each phase focused on a different aspect of system functionality, progressively increasing in realism and integration:

4.1 Phase 1: AutoPilot Unit Testing

WAPS is first evaluated in isolation using deterministic input scenarios. A scenario refers to a scripted sequence of simulated sensor inputs—such as wind speed, battery voltage, GPS availability, and stabilization altitude—crafted to replicate specific environmental or failure conditions. These scenarios are designed to isolate and test particular autopilot behaviors, such as triggering (or deliberately avoiding) a safety response. The system's outputs are then compared against expected results to verify correctness and robustness.

For example, to test the wind safety logic, one scenario injects a constant wind level below the threshold, punctuated by brief spikes above the threshold to simulate gusts. This scenario verifies that brief gusts do not trigger the safety protocol unnecessarily. Another scenario may input a sustained wind level above the threshold to ensure that the safety check correctly initiates the emergency procedure. Similar crafted scenarios are used to evaluate all the other safety conditions and some of their responses are reported on Section 5.1. Figure 7 provides a schematic overview of the testing environment described above.

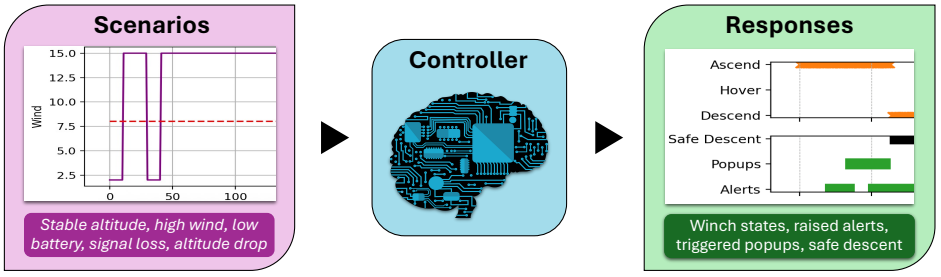


Figure 7: Unit testing environment. Scripted scenarios of wind speed, battery voltage, GPS dropout, and stabilization altitude are injected into the autopilot. Its deterministic responses are monitored to verify correct behavior.

4.2 Phase 2: Software-in-the-Loop Simulation

While unit testing confirmed WAPS responses in idealized environments, it failed to capture the variability of atmospheric conditions and the coupled dynamics of the full Helikite system. To address this limitation, a custom Software-in-the-Loop (SIL) simulation was developed to test the autopilot's behavior under realistic environmental and dynamic conditions.

In this SIL setup, the autopilot software interfaces with a custom-developed simulator to enable closed-loop, real-time control of a virtual Helikite system. "Closed-loop" here means that the autopilot receives simulated sensor data and sends control commands back to the simulation continuously during runtime. Although the flight computer, ground station, and winch controller all run on a single workstation, their communications are emulated as separate virtual serial links using the `socat` utility. This approach replicates the physical system's architecture, where sensors and actuators communicate over distinct interfaces.
By emulating these independent communication channels, the testing framework supports fault isolation and allows realistic modeling of network conditions such as latency, jitter, and packet loss. Controlled fault injection can be applied to evaluate the autopilot's robustness against communication disruptions commonly encountered in field deployments. Furthermore, encapsulating the simulator as an independent software module simplifies development and enables modular testing of each subsystem. Figure 8 presents an overview of the experimental configuration.

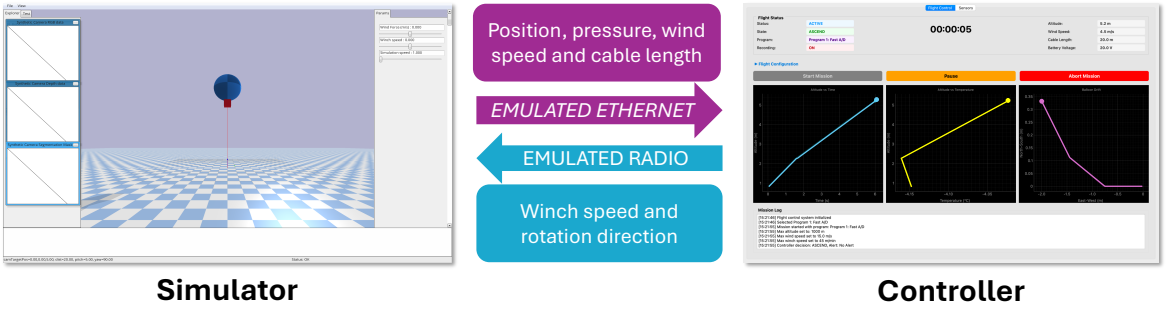


Figure 8: SIL simulation. WAPS interacts with a winch-Helikite simulation. Communication between units is emulated through virtual ports.

The simulator was developed using the `PyBullet` physics engine, chosen for its capability to perform real-time simulation of complex multibody dynamics. This makes it particularly suitable for modeling the coupled behavior of the Helikite system components: the balloon, kite, payload, and tether. The physical autopilot is integrated with the simulator in a closed-loop configuration, enabling realistic interaction between the control software and the virtual environment.
To provide a realistic atmospheric context, the simulator uses vertical profiles of temperature, pressure, wind speed, and particle concentration derived from measurement data collected during the 2023 ARTofMELT campaign in the Arctic. These profiles are used to initialize and update atmospheric conditions throughout the simulation.
Simulator development proceeded incrementally, with physical forces introduced and calibrated one at a time to closely reproduce flight trajectories observed in previous Helikite campaigns (see Figure 9). Calibration was performed by comparing simulated trajectories to reference flight data, ensuring model accuracy.

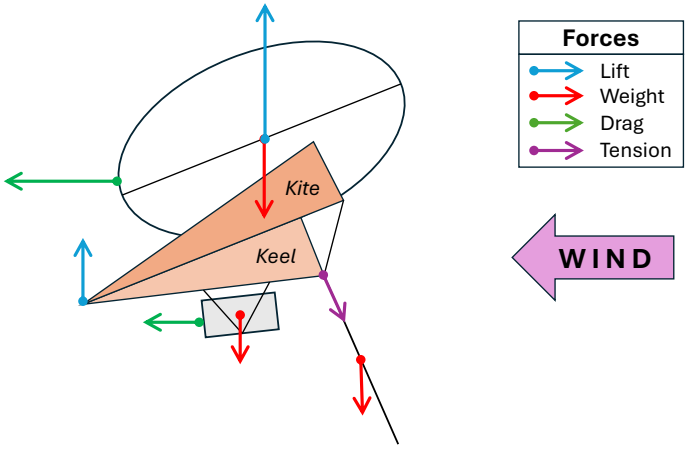


Figure 9: Schematic of the forces modeled in the Helikite simulator.

- **Gravitational force:** Acts on the balloon, payload, and tether based on their masses, providing the fundamental downward force that the system must counteract.
- **Buoyant lift (Archimedean thrust):** Generated by the helium-filled balloon, this upward force depends on the local atmospheric density and balloon volume. It is the primary source of lift that enables the Helikite to ascend and maintain altitude.
- **Aerodynamic drag forces:** Lateral forces caused by horizontal wind acting separately on the balloon and payload. Modeling these forces independently captures how wind affects each component's displacement, which is crucial for accurate simulation of system stability.
- **Tether dynamics:** The tether is modeled as a spring with variable mass distribution to reflect realistic mechanical behavior during reeling and unreeling.
- **Keel stabilization force:** Implemented as a damping and corrective force to reduce unwanted oscillations, this stabilizing effect helps maintain the kite's orientation toward the wind and enhances flight stability in turbulent conditions.
- **Stern lift:** A vertical aerodynamic force produced by wind acting on the inclined surface of the kite sail. This force is essential for achieving a realistic balance between vertical lift and horizontal drift, significantly impacting the Helikite's ability to maintain altitude and direction under high wind conditions.

The simulator played a key role in validating the autopilot's decision-making logic under realistic and repeatable conditions. By coupling the autopilot software with a virtual environment that replicates real-world dynamics and communication constraints, it enabled thorough testing of control strategies prior to field deployment.
Beyond this validation phase, the simulator also serves as a versatile tool for future system design. Its high configurability allows users to model alternative balloon volumes, payload masses and geometries, and tether materials—enabling pre-deployment testing of novel Helikite configurations. This capability supports early assessment of lift performance, aerodynamic stability, and the equilibrium altitude under realistic polar atmospheric profiles.

4.3 Phase 3: Full-System Integration

Following the SIL simulation phase, the final stage aimed to validate the complete hardware–software integration under near-realistic deployment conditions. Here, "near-realistic" refers to a test environment where the actual mission hardware and software are used—identical to the full-scale setup—but with the Helikite replaced by a mechanical pulley system. This setup offered two main benefits: it avoided the significant cost of helium—several thousand Swiss francs per fill—and bypassed the need for flight permit, which was not feasible due to time constraints.

The experiment was conducted on the Alpole campus, where the winch, ground computer, and MoMuCAMS payload were deployed exactly as in a real mission. Instead of being lifted by a helium-filled Helikite, the payload was suspended on a cable that ran through a pulley installed on an elevated structure of the Alpole building—effectively forming a crane-like system. The cable was connected to the ground-based winch, enabling vertical motion of the payload through controlled winding and unwinding (see Figure 10).

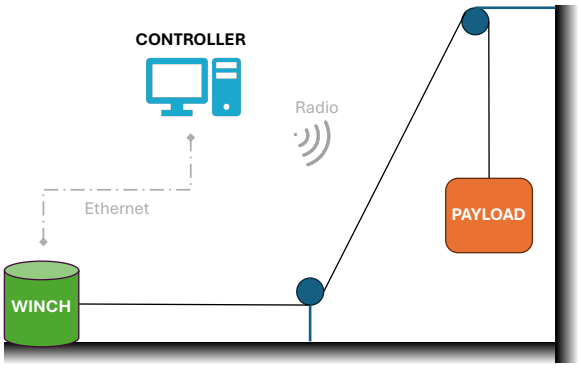


Figure 10: Full-system integration test. Real-time sensor data from the MoMuCAMS payload drives WAPS decisions, which in turn actuate the winch through a pulley-cable system.

One consideration was the inversion of motion logic: in the real Helikite setup, unwinding the winch allows the payload to ascend, while rewinding causes descent. This behavior was reversed in the pulley system: pulling the cable caused the payload to ascend. Minor adjustments were made to the autopilot code to reflect this inversion during testing.
Despite not using the Helikite, this setup preserved realistic vertical dynamics and enabled full use of the mission sensor suite—providing a highly effective proxy for end-to-end deployment testing.

5 Testing Results

Here we present the results of the multi-phase testing approach described in Section 4. The goal is to evaluate the smart autopilot's behavior across a range of simulated conditions, given that real-world flight testing was not possible.

5.1 Phase 1: Unit Testing

Unit testing results are presented to validate the system's autonomous behavior under nominal and hazardous scenarios. More than a dozen test cases were conducted to evaluate WAPS performance across a range of scenarios, including low battery levels, signal loss, strong winds and simultaneous alert events. From this larger campaign, we highlight three representative cases: (i) a nominal ascent/descent profile, (ii) the system's response to sustained high winds, and (iii) behavior following an unexpected loss of altitude. These scenarios were selected to showcase key aspects of the autopilot's logic, such as state transitions, safety program activation, and response to notification events. In all scenarios, user popups were intentionally ignored to isolate and assess WAPS's autonomous fallback mechanisms, particularly the triggering of the *Safe Descent* program.
Figure 11 presents the results for each scenario, showing the temporal evolution of altitude, autopilot states, wind conditions, and any triggered alerts or popups. For all tests, Program A is used as the active control program, with a wind threshold set at 12 m/s and a hover duration at the profile top fixed at 120 s. The safety-related thresholds governing event detection and automatic responses follow the default values listed in Table 1.

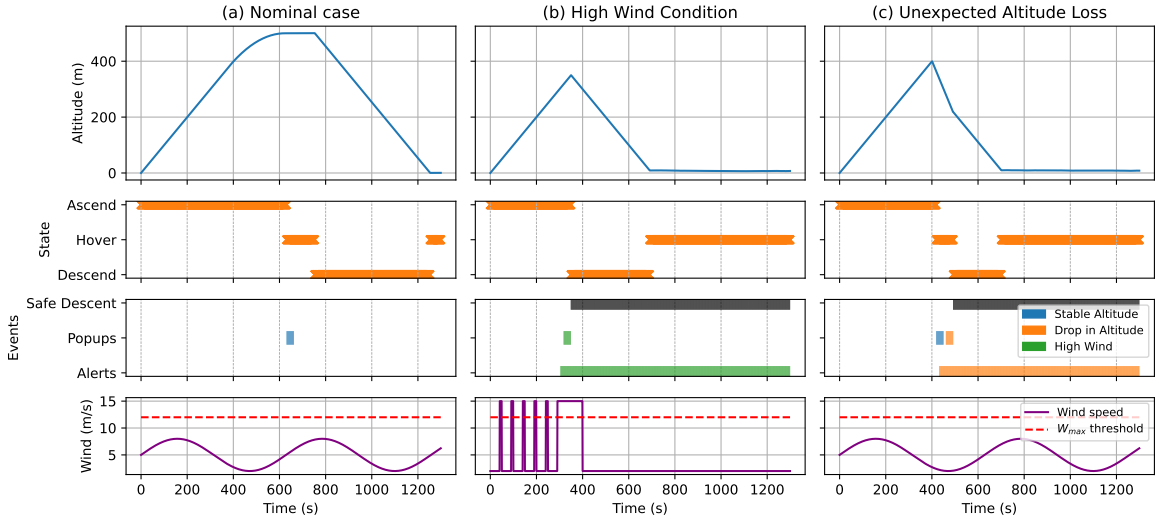


Figure 11: WAPS behavior under three unit test scenarios: (a) nominal ascent/descent profile, (b) response to sustained high wind, and (c) reaction to unexpected altitude loss. Each subplot shows the evolution of altitude, autopilot state, triggered popups or alerts and wind speed. Operator input is intentionally ignored to evaluate autonomous safety mechanisms.

5.1.1 Nominal Case

The first scenario (Figure 11a) demonstrates the standard behavior of *Program A* under nominal conditions. WAPS initiates the ascent by commanding the winch to unwind the cable. As altitude increases, the reduced atmospheric density results in a diminished buoyant force, while the unwinding tether increases the system's mass. This leads to a natural equilibrium altitude where buoyancy and gravitational forces are balanced.

The autopilot detects this steady state using a windowed averaging method over a duration $T_{\text{stable}} = 60$ s. A popup is then generated to request user confirmation (Figure 11a, third row). As this is ignored, the system waits for the specified timeout period $T_{\text{popup}} = 30$ s, assumes that the top of the profile has been reached, and—after an additional delay $T_{\text{top}} = 120$ s—begins descent. The Helikite returns smoothly to its original altitude, confirming proper execution of the default mission profile.

5.1.2 High Wind Conditions

The second scenario (Figure 11b) evaluates WAPS's robustness under wind-induced stress. Initially, transient gusts with durations less than $T_{\text{wind}} = 15$ s and speeds below $W_{\text{max}} = 12$ m/s are simulated. These disturbances are too short to trigger any alert. This behavior is nominal and has been implemented to avoid a mission interruption for short gusts, which can occur but are not as critical as high sustained wind.

At $t = 300$ seconds, a sustained wind of 15 m/s is applied for 100 seconds. This exceeds both the speed and duration thresholds, correctly triggering the safety procedure after 15 seconds. A popup is displayed to the operator, which, being ignored, leads the autopilot to initiate the *Safe Descent* program (Figure 11b, third row). This action overrides the current flight plan and commands the Helikite to return to its initial altitude. Once there, the system maintains this altitude in a holding pattern until manual control is resumed by the operator. This response confirms the autopilot's ability to autonomously detect and mitigate persistent hazardous wind conditions.

5.1.3 Unexpected Altitude Loss

In the third scenario (Figure 11c) , a $v = 2$ m/s drop in altitude is introduced at $t = 400$ seconds while the system is in *ascent* state. The autopilot first interprets the rate change as altitude stabilization and generates a corresponding popup. Shortly afterward, a second popup correctly identifies a dangerous descent rate. Since the second notification is ignored, the autopilot transitions into the *Safe Descent* program. This behavior aligns with safety expectations, though we note a minor issue in the altitude stabilization logic. The autopilot determines stability by computing the mean vertical speed over a $T_{\text{stable}} = 60$ seconds window. If the average rate of change remains below a fixed threshold $v_{\text{max}} = 0.2$ m/s, it assumes the altitude is stable. However, in scenarios where the Helikite initially ascends and then begins to descend, the averaged vertical speed may still fall below the threshold. This can incorrectly trigger the "stable altitude" popup just before detecting the actual loss of altitude. While this does not compromise safety, it suggests that the stabilization condition may benefit from additional constraints or directionality checks in future implementations.

5.2 Phase 2: Software-in-the-Loop simulation

Now, we evaluate WAPS under more realistic conditions using a SIL simulation. This configuration links actual autopilot software to a physics-based simulator that generates realistic sensor inputs from recorded atmospheric profiles and adds controlled disturbances to inter-module communication.

Figure 12 compares real-world data—collected during the ARTofMELT campaigns on (a) 18/05/2023 and (b) 23/05/2023 using manual winch control—with simulated flights executed autonomously by the autopilot under identical atmospheric conditions. To reproduce these flights in simulation, the autopilot executed (a) *Program B - Stepped Descent* (with one profile, 6-minute hover at the top, and two 1-hour steps) and (b) *Program A - Direct Descent* (with one profile and a 1-hour hover at the top). The close alignment of altitude profiles—particularly the equilibrium altitude reached during ascent—indicates that the simulator accurately models atmospheric density. Since buoyancy is directly affected by local air density, this agreement confirms that the simulator reliably captures the physical conditions governing vertical dynamics.

Additionally, SIL simulation served as a platform for evaluating the user interface under real-time conditions. To facilitate this, the simulation was slowed down such that one simulated second corresponded to one real second. This allowed for comprehensive testing of interface responsiveness, usability, and alert handling. Several iterative improvements were made to the interface based on these tests, particularly in popup design and alert clarity.

While the interface was primarily evaluated by the developer during this phase, it can also be used for training purposes by future operators to get familiar with the system before going to the field. It will also allow testing of new features in the future before any field deployment.

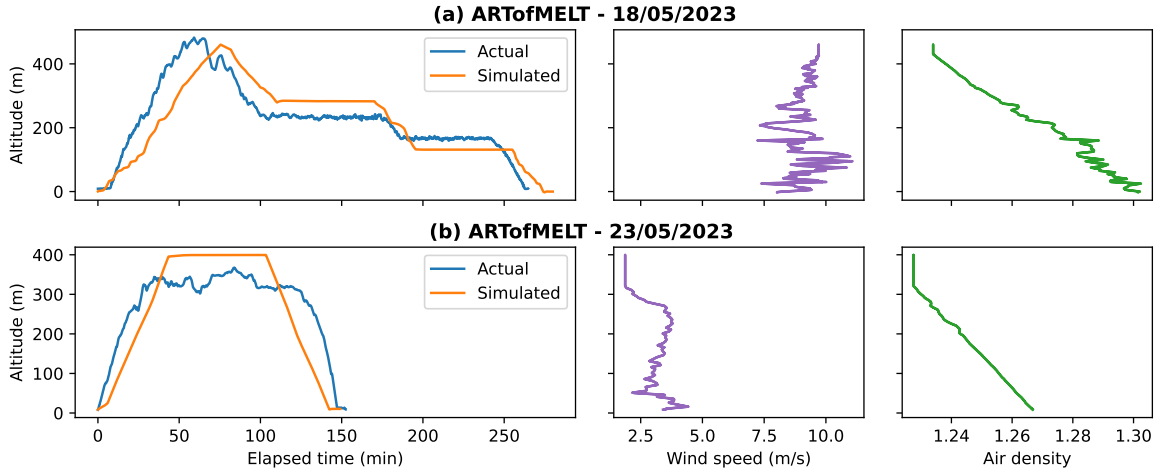


Figure 12: Comparison of actual and simulated Helikite trajectories during two ARTofMELT campaign flights: (a) under relatively strong wind conditions, and (b) under weaker wind.

5.3 Phase 3: Full-System Integration

The final integration of all system components was conducted in the Alpole courtyard (see Figure 13), following the assembly procedures outlined in Section 4. This phase served as a comprehensive end-to-end validation of the system in a near-operational environment.

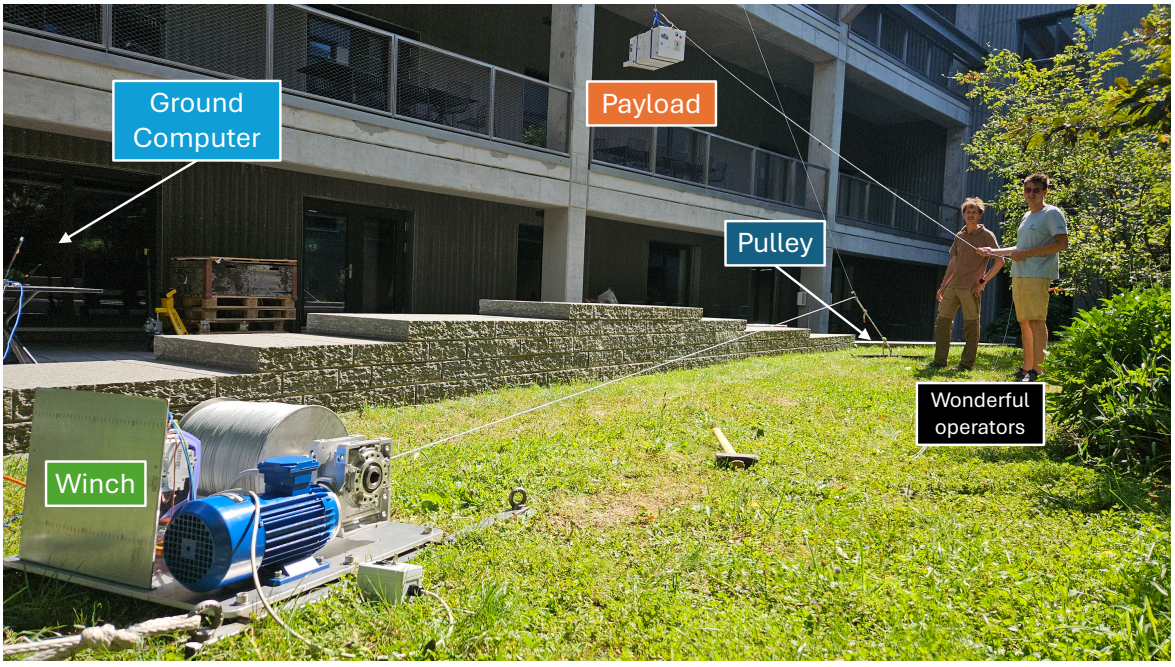


Figure 13: Photograph of the full-system integration test setup, conducted in the courtyard of the Alpole building.

The Ethernet-based communication between WAPS and the winch was tested under operational conditions and found to be both reliable and low-latency. Particular attention was given to transitions between manual (operator-driven) and automatic (software-controlled) winch modes during live operation. Manual control is provided through a dedicated control box directly connected to the winch controller, featuring up/down buttons, a mode toggle, and a joystick to adjust winch rotation speed.
Tests confirmed that the operator can safely switch from automatic to manual mode at any point during autonomous flight without disrupting system performance or control logic. In manual mode, the autopilot is immediately overridden—halting the current mission and blocking further commands. Switching back to automatic mode does not resume the interrupted program; instead, the system enters a neutral “hold” state,

requiring explicit user action to restart operations. This design reinforces operator authority and guarantees safe, unambiguous control handover at all times.

Second, the RF link between the ground station and the flight computer was assessed. WAPS successfully received all required telemetry from onboard sensors, validating the robustness of wireless communication and confirming that the control logic could operate with live data inputs.

The tests also enabled empirical tuning of the minimum safety altitude to account for communication delays and sensor accuracy—factors that cannot be fully captured in the SIL simulation. As described in Section 3.1, the operator is expected to manually position the Helikite at a safe altitude above ground before initiating any automatic program. This precaution is particularly important given that test results showed the autopilot consistently under or overshoots the target altitude by several meters due to latency and pressure sensor resolution. Since this initial altitude defines both the base of programmed profiles and the target altitude for emergency procedures like *Safe Descent*, it must not be set too close to the ground. Field tests led to the adoption of a fixed 10-meter safety margin, defining the safe altitude as the initial altitude plus 10 meters.

Core flight programs were executed and validated under quasi-realistic conditions during this final phase. However, retesting of alert detection and altitude stability was not performed in this phase due to time and resource constraints. Specifically, replicating conditions like wind-related alerts requires artificial wind data injection, while simulating equilibrium or sudden altitude loss demands controlled tether manipulation. Such tests are complex to conduct safely without specialized tools or extensive preparation. Nevertheless, these functions were thoroughly validated in prior phases (unit and SIL tests), making their omission acceptable here.

Finally, and importantly for future routine operations (as introduced in Section 1), this integration test demonstrated the potential for the Helikite system to be operated safely and efficiently by a single individual. While this was not a fully autonomous field deployment, the combination of clear operator authority and a user-friendly autopilot interface demonstrated promising potential for safe and efficient solo operation—addressing a key project objective.

6 Dynamic Flight Adaptation to Detected Atmospheric Features

The autopilot system developed in this work has proven robust and reliable for executing pre-programmed flight plans with minimal operator intervention. To further reduce operator workload and enhance the scientific value of each flight, an important goal is to enable dynamic adaptation of the altitude of descent pauses (program B) based on real-time atmospheric observations gathered during the ascent phase.

Extended measurement pauses within specific atmospheric layers—such as temperature inversions, clouds, or polluted layers—significantly improve data density and thus the statistical robustness of subsequent scientific analyses. Currently, such adaptation relies on manual operator input: the operator monitors live atmospheric profiles via the user interface and adjusts the default autopilot step altitudes accordingly. The objective is to automate this process by integrating an onboard algorithm that identifies the presence of clouds, pollution plumes or inversions when the balloon has reached the top of the profile and suggests an optimized descent flight pattern. This section presents preliminary steps toward this capability, supported by analysis of extensive datasets collected during prior EERL campaigns (ALPACA, ARTofMELT, Villum, GreenFjord, PaCE).

The datasets contain profiles of pressure (used to derive altitude), temperature, relative humidity (RH), and particle concentrations in two size ranges (8–280 nm and 186–3370 nm). Cloud presence is manually flagged for each measurement based on onboard video and field observations, using discrete flags: 0 (no cloud), 1 (below cloud), 2 (inside cloud), and 3 (above cloud).

6.1 Cloud detection

Cloud detection primarily relies on atmospheric variables that change in the presence of clouds, namely RH (RH) and aerosol particle concentrations. When RH increases and reaches supersaturation (i.e., RH exceeding 100%), water vapor condenses onto a subset of aerosol particles known as Cloud Condensation Nuclei (CCN). Once incorporated into water droplets, which are larger and heavier, these particles typically are not sampled by the MoMuCAMS inlet. This process results in a noticeable decrease in the total aerosol concentration—particularly for particles larger than 100 nm—measured by MoMuCAMS within the cloud.

To quantify relationships between cloud presence and atmospheric variables, Pearson correlation coefficients were computed between the cloud flag and several measurements (Table 2). Relative humidity exhibited the strongest correlation (0.68), confirming its key role in cloud formation detection. Particle concentrations in the 186–3370 nm size range correlated moderately (0.23), while temperature and pressure showed weaker correlations (0.12 each). Concentrations of smaller particles (8–280 nm) displayed a very weak correlation

(0.03). This weak association primarily reflects aerosol microphysics: smaller particles are generally less likely to activate as CCN at typical supersaturations, remaining suspended as free particles within clouds and thus continuing to be sampled by the instrument. Additionally, the small-particle measurement instrument records data less frequently (every few minutes) compared to other sensors with second-level resolution. This lower temporal resolution reduces spatial matching accuracy along vertical profiles, further weakening the correlation with cloud presence. Statistical significance tests should be considered to confirm the robustness of these correlations.

Table 2: Pearson correlation coefficients between the cloud flag and different measured variables.

Variable	**Correlation with Cloud Flag**
Relative Humidity	0.68
Particle Concentration (186–3370 nm)	0.23
Temperature	0.12
Pressure	0.12
Particle Concentration (8–280 nm)	0.03

Given the higher correlations for RH and particle concentration (186–3370 nm), their distributions across cloud flags were examined in detail. Boxplots (Figure 14) illustrate the distribution of RH and particle concentrations across cloud flags. Median RH reaches about 95% inside clouds (flag 2), consistent with saturation. However, measurements labeled "above cloud" (flag 3) also show relatively high humidity around 90%, and their interquartile ranges overlap notably with in-cloud values. This overlap arises from several factors. First, clouds only form when RH exceeds 100%, but RH sensors are not perfectly accurate, and their measurements can be slightly lower than actual saturation values within clouds. Second, cloud boundaries are inherently diffuse and variable in space and time, making precise identification challenging. The cloud flagging in this dataset is done manually using field observations and onboard camera footage, introducing subjectivity. As a result, some measurements near the edges of clouds or in partially cloudy conditions can be classified differently, contributing to the overlapping RH distributions between "in cloud" and "above cloud" categories.

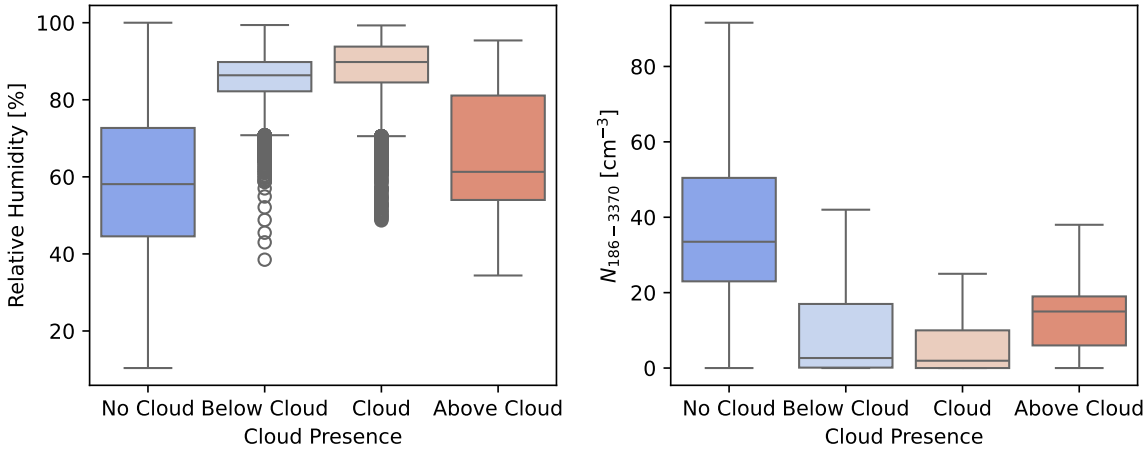


Figure 14: Boxplots showing the distribution of RH and particle concentrations (186–3370 nm size range) for different cloud flag categories across all campaigns.

Particle concentrations (186–3370 nm) are slightly lower inside clouds (median 18 cm^{-3}) but show wide variability with overlapping interquartile ranges across all cloud flags. This variability arises both from the subjectivity in manual cloud flagging and the complex physical behavior of aerosols. The smoothed median particle concentration profiles, normalized by cloud base height, from two subgroups of vertical profiles during the ARTofMELT campaign illustrate this (Figure15). The left panel shows the median and interquartile range from 5 profiles, while the right panel shows 11 profiles. Some profiles exhibit a distinct dip in particle concentration within clouds, whereas others display a more gradual variation with altitude, demonstrating the diversity of aerosol vertical structure in relation to cloud presence.

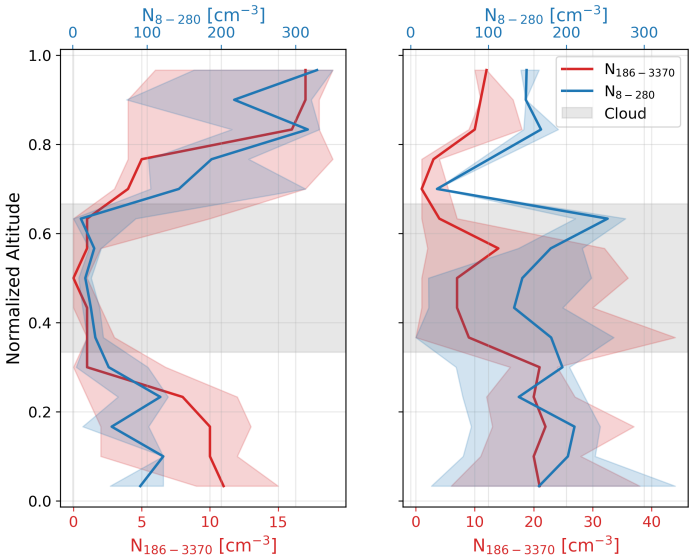


Figure 15: Smoothed median particle concentration profiles (186–3370 nm), normalized by cloud base height, for two subgroups of vertical profiles during the ARTofMELT campaign. Left: 5 profiles; Right: 11 profiles. Shaded areas represent interquartile ranges.

Overall, neither RH nor particle concentration alone reliably discriminates cloud presence. To explore whether combining multiple atmospheric variables improves detection, a PCA was performed (Figure 16). The first two principal components explain 39% and 18% of the variance, together capturing 57% of dataset variability—indicating moderate but incomplete representation of the underlying complexity. These components reflect linear combinations of variables such as humidity, aerosol concentration, temperature, and pressure (ie., PC1 may correspond to moist while PC2 might reflect vertical gradients in temperature or pressure). The dispersion of variance across many components suggests that atmospheric variability—and consequently cloud detection—results from a complex interaction of factors rather than any single dominant variable.

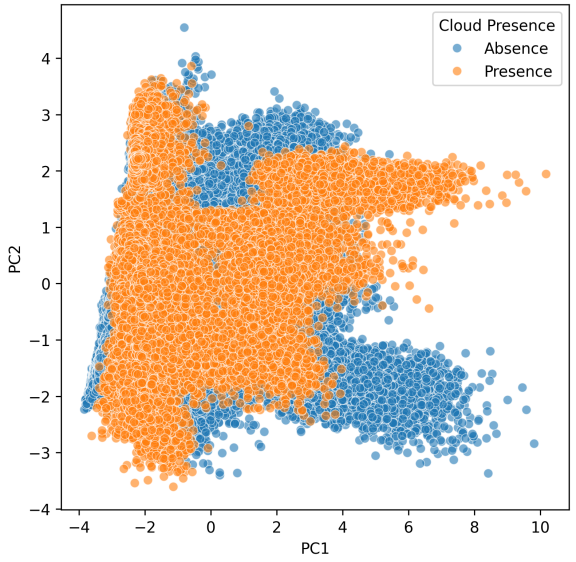


Figure 16: Principal Component Analysis (PCA) of all measured features colored by cloud flag categories. The first two principal components explain 39% and 18% of variance, respectively.

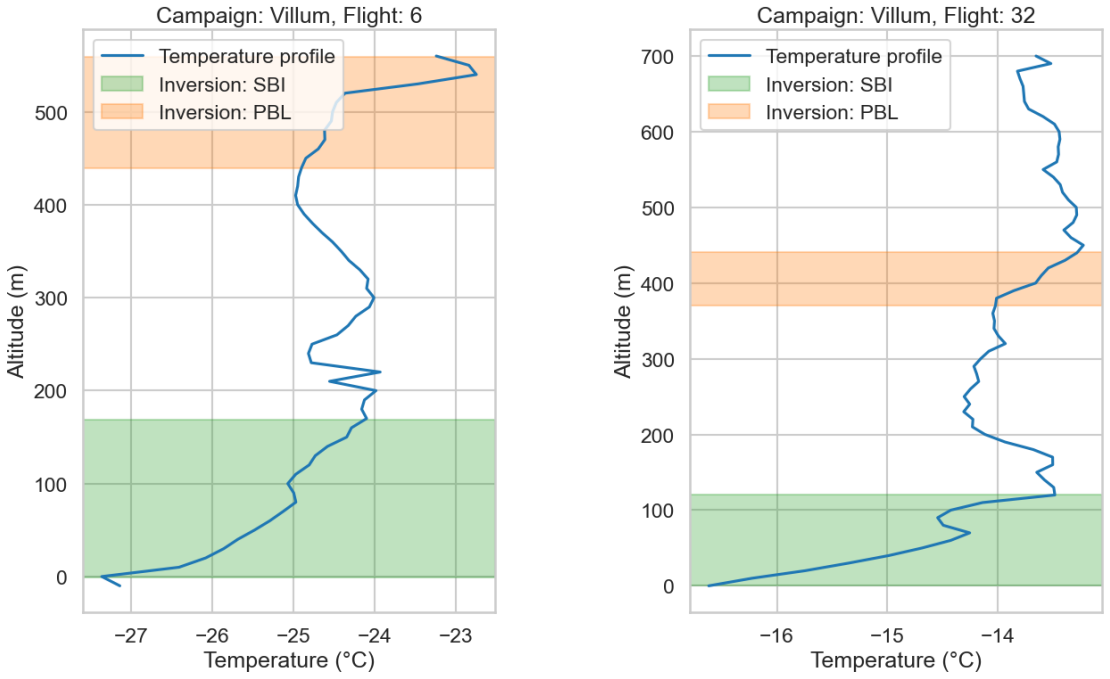
However, the PCA plot reveals no distinct clustering by cloud presence, suggesting that linear combinations of variables cannot effectively separate cloud states. This is partly due to nonlinear interactions between variables, measurement noise, instrument variability, and subjective manual cloud flagging. Moreover, limitations in sensor accuracy—especially the challenge of reliably measuring RH near saturation—contribute to uncertainty.

These findings indicate that more advanced, nonlinear machine learning methods, combined with rigorous preprocessing (e.g., noise filtering, outlier removal), will be necessary for robust cloud detection. The planned addition of a cloud droplet analyzer, providing direct microphysical data, is expected to enhance detection capability by complementing RH and aerosol measurements. Together, these improvements are critical for developing reliable autopilot features that can detect and respond to cloud layers effectively during flight.

6.2 Temperature inversion detection

Temperature inversion is a key atmospheric feature associated with both Surface-Based inversion (SBI) and Planetary Boundary Layer (PBL) inversions. SBIs typically form close to the ground during nighttime or cold conditions, when the surface cools and traps a layer of cooler air beneath warmer air above. In contrast, PBL inversions occur near the top of the boundary layer and are shaped by larger-scale meteorological dynamics such as subsidence or radiative cooling. Understanding and detecting these inversions is important for studying boundary layer processes and atmospheric stability.

A preliminary inversion detection algorithm was developed as follows: temperature profiles are binned by altitude, smoothed with a Savitzky–Golay filter (Savitzky and Golay, 1964) to reduce noise while preserving gradients, then vertical temperature gradients between bins are computed. Consecutive altitude segments exceeding a positive gradient threshold—indicating temperature increases with altitude—are flagged as potential inversions. Nearby segments separated by small gaps are merged, and segments below a minimum thickness threshold are discarded to avoid false positives. Detected inversions are classified as SBI if below 50 m or PBL otherwise. The algorithm outputs a summary table with altitude range, temperature change, thickness, and classification for each inversion. Figure 17 shows two example detections illustrating the algorithm's ability to identify distinct inversion layers.



(a) Example 1

(b) Example 2

Figure 17: Two examples of temperature profiles showing detected temperature inversion layers identified by the algorithm. Inversions are highlighted where the vertical temperature gradient exceeds the defined threshold.

Although promising, the algorithm requires further tuning of parameters such as gradient thresholds, smoothing window sizes, and minimum thickness to improve detection rates. Currently, only 92 inversions (SBI or PBL) were detected across more than 40,000 profiles, reflecting both the rarity of inversions and potential detection limitations. Critically, the lack of labeled ground truth for temperature inversions hampers performance evaluation, motivating future efforts to generate reference data and validate the algorithm.

6.3 Future Work and Timeline

To meet the TPS milestone in December 2025, the first step is to clearly define the scientific objectives by specifying which atmospheric layers—such as clouds, temperature inversions, or polluted layers—are of primary interest, along with the precise characteristics and detection accuracy required for each. Following this, efforts will focus on enhancing the dataset by developing automated methods to segment flight data into individual profiles and accurately tagging temperature inversion layers, thereby establishing a ground truth dataset for rigorous evaluation of detection algorithms. Once the algorithms achieve satisfactory performance, they will be integrated into the autopilot software with careful consideration of safety, ensuring that operator override authority is always maintained. Finally, the new adaptive descent functionality will undergo extensive testing within the developed simulator to verify that it meets performance goals without interfering with existing autopilot features. Completing these steps is essential to deliver a robust system capable of dynamically adapting flight plans based on real-time atmospheric observations.

7 Conclusions and outline

This thesis presented the design, implementation, and validation of an autonomous autopilot for Helikite flights with the MoMuCAMS platform, enhancing its capability for atmospheric aerosol monitoring in remote and challenging environments. WAPS automates flight operations via predefined programs (Direct Ascent/Descent, Stepped Descent, and Safe Descent) and incorporates safety mechanisms for high winds, altitude anomalies, battery loss, and signal interruption. Built in modular Python, the system is designed for extensibility and robustness. Its graphical interface, developed with the Qt framework, allows scientists with minimal technical background to operate the system effectively.

Due to regulations and logistical restrictions, direct field testing was not possible to fully evaluate the system. Instead, a structured testing approach was adopted to overcome this limitation. The first phase consisted of unit tests, where WAPS was exposed to scripted sensor input scenarios—such as wind gusts, battery voltage drops, and GPS loss—to verify its logic under deterministic conditions. These tests confirmed that the autopilot responded as expected, correctly activating safety protocols or maintaining normal operation. In the second phase, software-in-the-loop (SIL) simulations evaluated the system's performance using realistic sensor inputs and communication disturbances generated by a custom Helikite simulator. This validated the autopilot's behavior under near-real conditions and enabled refinement of the user interface for effective mission operation. Finally, a full-system integration test used the actual mission hardware in a ground-based pulley setup to replicate deployment conditions without flight. This test verified communication between subsystems, validated real-time response, and demonstrated full system coordination. Collectively, these tests confirmed the system's reliability, safety, and accuracy—even under simulated disturbances—providing strong confidence for the first field deployment in Greenland in August 2025.

A key outcome is the autopilot's capacity for single-user operation, greatly enhancing its scalability for deployments such as the TPS expedition in late 2025. While the system met all primary objectives, further refinement of the altitude stabilization logic—particularly in accounting for directional trends—may improve responsiveness and alert accuracy.

Future work will prioritize implementing real-time detection algorithms for key atmospheric features—including temperature inversion layers, clouds and pollution plumes—to enable adaptive step-based descents within Program B. This emerging capability aims to empower the autopilot system to autonomously modify step altitudes and pause durations based on scientifically relevant atmospheric conditions, thereby reducing operator workload. Field validation during the Greenfjord (August, 2025) and the TPS (December, 2025) expediton is also planned. These enhancements will consolidate MoMuCAMS as a scalable, field-ready tool for atmospheric research, contributing to increased capabilities for vertical profiling in extreme environments in the perspective of the future TPS drifts and the 5th International Polar Year.

Code Availability

The autopilot software developed in this work is publicly available on GitHub at `https://github.com/EERL-EPFL/HFC2`. The Helikite system simulator code can be accessed at `https://github.com/tuturta/helikite_flight_simulator`.

References

O. Boucher, D. Randall, P. Artaxo, C. Bretherton, G. Feingold, P.M. Forster, V.-M. Kerminen, Y. Kondo, H. Liao, U. Lohmann, P. Rasch, S.K. Satheesh, S. Sherwood, B. Stevens, X.Y. Zhang, and I.P.C.C. Climate Change 2013: The Physical Science Basis. In Stocker by, T. F., D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, and P. Midgley, editors, *Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge Universiy Press, 2013.

C.A. Brock, J. Cozic, R. Bahreini, K.D. Froyd, A.M. Middlebrook, A. McComiskey, J. Brioude, O.R. Cooper, A. Stohl, K.C. Aikin, J.A. Gouw, D.W. Fahey, R.A. Ferrare, R.-S. Gao, W. Gore, J.S. Holloway, G. Hübler, A. Jefferson, D.A. Lack, S. Lance, R.H. Moore, D.M. Murphy, A. Nenes, P.C. Novelli, J.B. Nowak, J.A. Ogren, J. Peischl, R.B. Pierce, P. Pilewskie, P.K. Quinn, T.B. Ryerson, K.S. Schmidt, J.P. Schwarz, H. Sodemann, and Spackman. Characteristics, sources, and transport of aerosols measured in spring 2008 during the aerosol, radiation, and cloud processes affecting Arctic Climate (ARCPAC) Project. *J*, 11:2423–2453,, 2011. doi: 10.5194/acp-11-2423-2011. URL https://doi.org/10.5194/acp-11-2423-2011,.

G. Canut, F. Couvreux, M. Lothon, D. Legain, B. Piguet, A. Lampert, W. Maurel, and E. Moulin. Turbulence fluxes and variances measured with a sonic anemometer mounted on a tethered balloon, Atmos. *Meas. Tech*, 9:4375–4386,, 2016. doi: 10.5194/amt-9-4375-2016. URL https://doi.org/10.5194/amt-9-4375-2016,.

Ken S. Carslaw. Chapter 2 - Aerosol in the climate system. In *Aerosols and Climate*, pages 9–52. Elsevier, 2022. ISBN 978-0-12-819766-0.

J.M. Creamean, G. Boer, H. Telg, F. Mei, D. Dexheimer, M.D. Shupe, A. Solomon, and A. McComiskey. Assessing the vertical structure of Arctic aerosols using balloon-borne measurements, Atmos. *Chem. Phys*, 21:1737–1757,, 2021. doi: 10.5194/acp-21-1737-2021. URL https://doi.org/10.5194/acp-21-1737-2021,.

L. Ferrero, D. Cappelletti, M. Busetto, M. Mazzola, A. Lupi, C. Lanconelli, S. Becagli, R. Traversi, L. Caiazzo, F. Giardi, B. Moroni, S. Crocchianti, M. Fierz, G. Močnik, G. Sangiorgi, M.G. Perrone, M. Maturilli, V. Vitale, R. Udisti, and E. Bolzacchini. Vertical profiles of aerosol and black carbon in the Arctic: a seasonal phenomenology along 2 years (2011–2012) of field campaigns, Atmos. *Chem. Phys*, 16:12601–12629,, 2016. doi: 10.5194/acp-16-12601-2016. URL https://doi.org/10.5194/acp-16-12601-2016,.

J. Haywood and O. Boucher. Estimates of the direct and indirect radiative forcing due to tropospheric aerosols: A review. *Rev. Geophys*, 38:513–543,, 2000. doi: 10.1029/1999RG000078. URL https://doi.org/10.1029/1999RG000078,.

IPCC. *Climate Change 2021 - The Physical Science Basis : Working Group I Contribution to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge Universiy Press, Cambridge, June 2023.

D.J. Jacob, J.H. Crawford, H. Maring, A.D. Clarke, J.E. Dibb, L.K. Emmons, R.A. Ferrare, C.A. Hostetler, P.B. Russell, H.B. Singh, A.M. Thompson, G.E. Shaw, E. McCauley, J.R. Pederson, and J.A. Fisher. The Arctic Research of the Composition of the Troposphere from Aircraft and Satellites (ARCTAS) mission: design, execution, and first results, Atmos. *Chem. Phys*, 10:5191–5212,, 2010. doi: 10.5194/acp-10-5191-2010. URL https://doi.org/10.5194/acp-10-5191-2010,.

B. Koffi, M. Schulz, F.-M. Bréon, F. Dentener, B.M. Steensen, J. Griesfeller, D. Winker, Y. Balkanski, S.E. Bauer, N. Bellouin, T. Berntsen, H. Bian, M. Chin, T. Diehl, R. Easter, S. Ghan, D.A. Hauglustaine, T. Iversen, A. Kirkevåg, X. Liu, U. Lohmann, G. Myhre, P. Rasch, Ø. Seland, R.B. Skeie, S.D. Steenrod, P. Stier, J. Tackett, T. Takemura, K. Tsigaridis, M.R. Vuolo, J. Yoon, and K. Zhang. Evaluation of the aerosol vertical distribution in global aerosol models through comparison against CALIOP measurements: AeroCom phase II results. *J. Geophys. Res.-Atmos*, 121:7254–7283,, 2016. doi: 10.1002/2015JD024639. URL https://doi.org/10.1002/2015JD024639,.

M. Mazzola, M. Busetto, L. Ferrero, A.P. Viola, and D. Cappelletti. AGAP: an atmospheric gondola for aerosol profiling, Rend. *Lincei-Sci. Fis*, 27:105–113,, 2016. doi: 10.1007/s12210-016-0514-x. URL https://doi.org/10.1007/s12210-016-0514-x,.

C. Pilz, S. Düsing, B. Wehner, T. Müller, H. Siebert, J. Voigtländer, and M. Lonardi. CAMP: an instrumented platform for balloon-borne aerosol particle studies in the lower atmosphere, Atmos. *Meas. Tech*, 15:6889–6905,, 2022. doi: 10.5194/amt-15-6889-2022. URL https://doi.org/10.5194/amt-15-6889-2022,.

Roman Pohorsky, Andrea Baccarini, Julie Tolu, Lenny H. E. Winkel, and Julia Schmale. Modular Multiplatform Compatible Air Measurement System (MoMuCAMS): a new modular platform for boundary layer aerosol and trace gas vertical measurements in extreme environments. *Atmospheric Measurement Techniques*, 17(2):731–754, January 2024. ISSN 1867-1381. doi: 10.5194/amt-17-731-2024. URL https://amt.copernicus.org/articles/17/731/2024/. Publisher: Copernicus GmbH.

B.H. Samset, G. Myhre, M. Schulz, Y. Balkanski, S. Bauer, T.K. Berntsen, H. Bian, N. Bellouin, T. Diehl, R.C. Easter, S.J. Ghan, T. Iversen, S. Kinne, A. Kirkevåg, J.-F. Lamarque, G. Lin, X. Liu, J.E. Penner, Ø. Seland, R.B. Skeie, P. Stier, T. Takemura, K. Tsigaridis, and K. Zhang. Black carbon vertical profiles strongly affect its radiative forcing uncertainty, Atmos. *Chem. Phys*, 13:2423–2434,, 2013. doi: 10.5194/acp-13-2423-2013. URL https://doi.org/10.5194/acp-13-2423-2013,.

M. Sand, B.H. Samset, Y. Balkanski, S. Bauer, N. Bellouin, T.K. Berntsen, H. Bian, M. Chin, T. Diehl, R. Easter, S.J. Ghan, T. Iversen, A. Kirkevåg, J.-F. Lamarque, G. Lin, X. Liu, G. Luo, G. Myhre, T.V. Noije, J.E. Penner, M. Schulz, Ø. Seland, R.B. Skeie, P. Stier, T. Takemura, K. Tsigaridis, F. Yu, K. Zhang, and H. Zhang. Aerosols at the poles: an AeroCom Phase II multi-model evaluation, Atmos. *Chem. Phys*, 17:12197–12218,, 2017. doi: 10.5194/acp-17-12197-2017. URL https://doi.org/10.5194/acp-17-12197-2017,.

Abraham. Savitzky and M. J. E. Golay. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, 36(8):1627–1639, July 1964. ISSN 0003-2700. doi: 10.1021/ac60214a047. URL https://doi.org/10.1021/ac60214a047. Publisher: American Chemical Society.

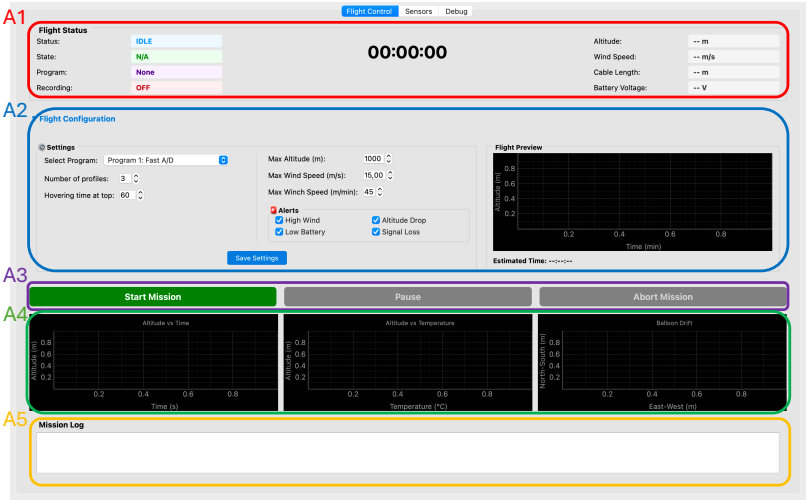
J.H. Seinfeld and S.N. Pandis. *Atmospheric chemistry and physics: from air pollution to climate change*. Wiley, April 2016. ISBN 978-1-118-94740-1. Pages: , 978–1–118–94740–1, Place: John Wiley&Sons, Hoboken, USA.

Appendix A User's Manual

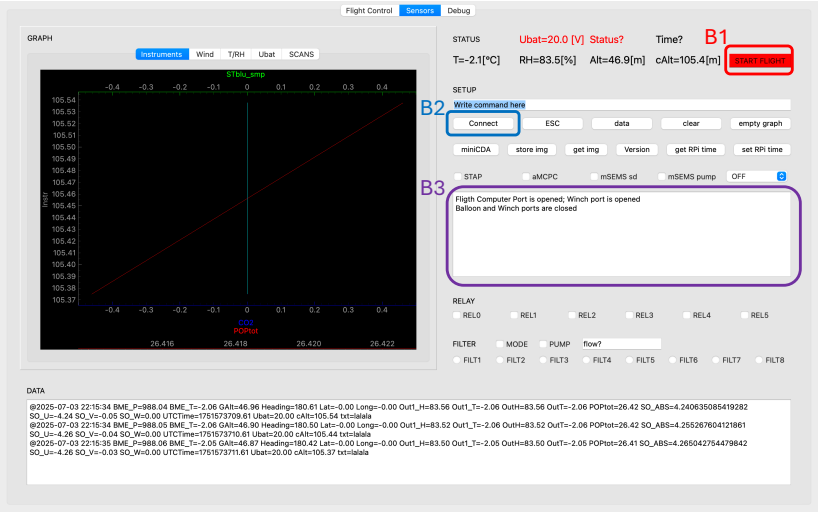
A.1 Overview

The Winch Autopilot System (WAPS) is a software-controlled platform designed to automate the vertical profiling of the atmosphere using a tethered balloon system. It interfaces with ground-based winch hardware and onboard sensors to carry out controlled ascent and descent maneuvers with minimal operator input. WAPS manages flight execution, monitors live telemetry, and provides an intuitive graphical interface for mission planning and real-time supervision. While the system is capable of autonomous operation, it is designed to ensure that the operator retains full authority to monitor, intervene, or override the mission at any time.
The user interface is divided into three tabs (see Figure 18).

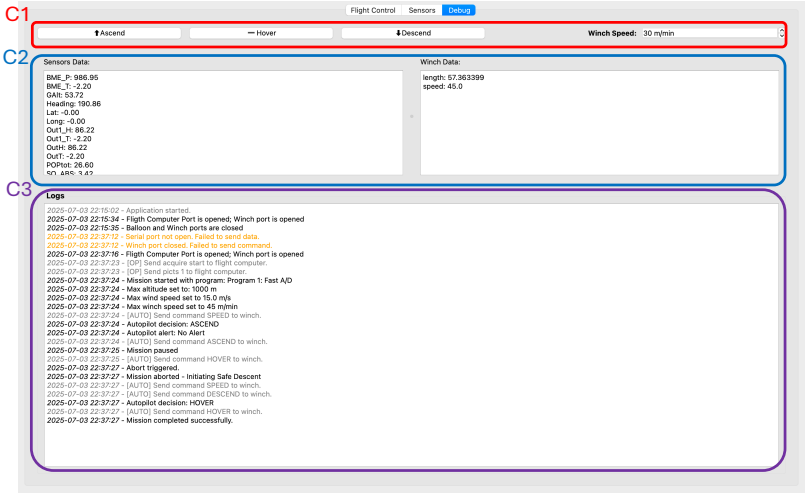
- **Flight Control (Fig. 18a):** This tab is dedicated to autopilot control and monitoring. It consists of five panels:
  - **A1: Autopilot Status**
    Displays the current status of the autopilot, including the program chosen, winch state, mission elapsed time, active alerts, and real-time telemetry data from various sensors.
  - **A2: Flight Configuration**
    Enables the user to design the flight mission by selecting waypoints and configuring flight parameters. It also allows previewing the planned mission path and setting which alerts to monitor, along with other general system settings.
  - **A3: Mission Control Buttons**
    Contains the main control buttons to *Start Mission*, *Pause*, or *Abort* the flight operation.
  - **A4: Flight Data Plots**
    Shows real-time graphs of key flight parameters such as altitude, temperature, and horizontal drift to help monitor flight conditions visually.
  - **A5: Mission Log**
    Provides a detailed log of autopilot actions and user commands throughout the mission, serving as a record for analysis and debugging.
- **Sensor (Fig. 18b):** This is the first Helikite measurement monitoring interface, created prior to the thesis. It includes two main functionalities useful for the autopilot:
  - **B1: Start Flight Button**
    Starts local sensor recording on the flight computer, enabling data capture during the flight.
  - **B2: Connect Button**
    Attempts to simultaneously connect to the MoMuCAMS and the winch controller. The success or failure of the connection attempt is indicated in panel B3.
- **Debug (Fig. 18c):** Intended for system debugging during initial flights; this tab will eventually be removed. It consists of three panels:
  - **C1: Winch Command Panel**
    Allows sending speed and rotation direction commands directly to the winch.
  - **C2: Sensor Monitoring Panel**
    Displays real-time sensor data from both the MoMuCAMS (left side) and the winch sensors (right side).
  - **C3: Log Panel**
    Records all actions taken by the autopilot and user, as well as software errors. Logs are saved as files on the ground computer for later review.



(a) Flight Control tab.



(b) Sensors tab.



(c) Debug tab.

Figure 18: Overview of the WAPS user interface and its different tabs (a) Flight control, (b) Sensors and (c) Debug.

A.2 Usage Procedure

1. **Establish the Ethernet connection with the winch system.**
   Plug the Ethernet cable into the Ground Computer, then ensure that the network settings are configured with the IP address `192.168.204.174`.

2. **Set up the radio-frequency link with the Flight Computer.**
   Turn on the Flight Computer, then connect the radio module system to the Ground Computer via the USB port (COM4).

3. **Launch the autopilot software.**
   Open the software named `HFC2` located on the Desktop of the Ground Computer (it has a Helikite logo).

4. **Initiate the connection to the winch and Flight Computer.**
   In the **Sensor** tab, press the "Connect" button (**B2**, Fig. 18b).

5. **Verify connection success.**
   Check the log messages in panel **B3** (Fig. 18b) to confirm whether the connections were established successfully.

6. **Pre-position the Helikite.**
   Using the winch's **manual control**, raise the Helikite to an altitude of approximately 10 meters. This altitude will be used as the base of the profiles and the safety threshold. **Do not start the autopilot flight from the ground!**

7. **Plan the flight mission:**

   (a) Navigate to the **Flight Control** tab.

   (b) Select the desired program and configure its corresponding options (**A2**, Fig. 18a).

   (c) Visualize the flight mission and optionally press "Save settings" to preview it (**A2**, Fig. 18a).

   (d) If necessary, adjust general settings such as maximum altitude, wind speed threshold, and winch speed setpoint (**A2**, Fig. 18a).

8. **Start the flight mission:**
   Press the "Start Flight" button (**A3**, Fig. 18a).

   (a) If sensors are not recording, a popup will prompt you to start data recording.

   (b) Once the flight begins, monitor the mission using the raw data display (**A1**) and the plots panel (**A4**).

   (c) Flight settings can be modified during the mission and saved using the "Save" button (**A2**).

   (d) If the Helikite stops ascending due to buoyancy limits, a popup will notify you. You can choose to hold position or continue trying to ascend.

   (e) For Program B, the "Edit steps" option will become available (**A2**) once the top altitude is reached. Use this to manually adjust the automatically generated descent step altitudes.

   (f) You may pause the mission at any time, which will hold the Helikite in place. Resuming will continue the mission from the paused point.

9. **End the flight:**

   (a) You can abort the mission at any time using the "Abort" button (**A3**, Fig. 18a). A popup will let you choose between a safe descent or holding position.

   (b) Alternatively, the mission will automatically end upon completion of the defined flight profiles.

A.3 Handling Alerts

When a safety condition is detected (e.g., excessive wind, unexpected drift), the autopilot activates a safety protocol to ensure the Helikite's protection and operator awareness.

A warning dialog box appears on the Ground Computer (Fig. 19), accompanied by an audible alarm on both the winch system and the Ground Computer.

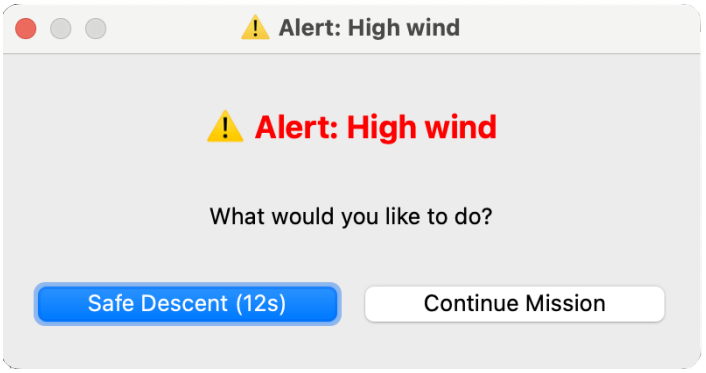


Figure 19: Example of a warning popup triggered by a safety condition (e.g., high wind speed).

The operator has T_{popup} seconds (default: 30 s) to respond by either:

- **Dismissing the alert:** Acknowledges the warning and temporarily disables the associated safety condition.
- **Starting Safe Descent:** Immediately triggers the *Safe Descent* program to bring the Helikite back to its safety altitude.

If no response is given within the allotted time, the system will automatically initiate the *Safe Descent* procedure. Once an alert is dismissed, the same condition will not trigger additional warnings for a duration of T_{disable} (default: 120 s).

A.4 Advanced Settings

The source code and configuration files are available directly on the Ground Computer at the following path: `C://Documents/HFC2_v2`.
This directory contains all necessary scripts, configuration files, and logs for modifying, testing, and debugging the autopilot software. The following files and tools can be used to adjust advanced settings:

- **Safety Parameters:** All threshold values for wind speed, drift, altitude, time averaging windows, and popup durations are configurable in:
  - `GroundComputer/autopilotParameters.yaml`
- **Communication Ports:** The connection details for both the winch (IP address) and the Flight Computer (COM port) are stored in:
  - `GroundComputer/config.txt`
- **Log Files:** All system logs are automatically saved in the `logs/` folder. Each log file is named using the format `app.log.YYYY-MM-DD` and is retained for 20 days. These logs contain detailed information that developers can use for debugging and post-flight analysis.

Debug Interface (Fig. 18c): This tab displays real-time sensor data from both the Helikite and the winch (C2). It also contains a comprehensive log of all commands issued by the user and the autopilot (C3), including execution errors. This tool is essential for identifying and diagnosing software bugs.
The operator can also send manual winch commands in panel C1 to test the connection between winch and autopilot software.