UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Real-Time Optimal Trajectory Generation for Fixed-Wing UAVs in Firefighting Missions via Numerical Integration and Constrained Optimization

Document:

Project Charter

Author:

Abimael Campillo Simón

Director/Co-director:

Prof. Dr. Alex Ferrer Ferre / Miguel Pareja Muñoz

Degree:

Bachelor in Aerospace Technology Engineering

Examination session:

Autumn 2025

BACHELOR FINAL THESIS

INTENTIONALLY LEFT BLANK.

Contents

List of Figures

Acronyms

DEM Data Elevation Model.

GPS Global Positioning System.

NFZ No-Fly Zones.

QHIL Quasi-Hardware-in-the-Loop.

RPA Remotely Piloted Aircraft.

UAS Unmanned Aerial Systems.

UAV Unmanned Aerial Vehicle.

WP Work-Package.

Chapter 1

Object

The main objective of this thesis is to develop an algorithm capable of computing the optimal flight trajectory for a Unmanned Aerial Vehicle (UAV) operating in firefighting missions. The algorithm is designed to generate restricted trajectories within a vertical plane while considering multiple environmental and operational constraints, such as the Data Elevation Model (DEM), pre-defined No-Fly Zones (NFZ), detected obstacles, and wind conditions.

Specifically, the algorithm aims to determine the descent path during the water-drop manoeuvre and the subsequent ascent trajectory, ensuring obstacle avoidance and compliance with mission constraints. Furthermore, it is intended to achieve a computational efficiency that enables real-time execution, making it suitable for on-board integration within the aircraft's guidance system.

This thesis is developed in collaboration of Singular Aircraft S.L. as part of a extracurricular internship agreement along with other tasks.

Chapter 2

Scope

This section includes a brief description of the work-packages this thesis addressed along with the high-level deliverables of each Work-Package (WP). This description follows a chronological order. Eventually, the not included points on this thesis are also remarked.

- **WP 0. Project Management.**
  - Set-up of the workspace environment.
  - Wording of the different deliverables of the project.
  - Scheduling of the tasks and meetings.
  - Wording of the minutes of meetings.
  - Version control of the code and all documents.
  - Environmental analysis and budget of the project.

  High-level deliverables: project charter, thesis' report (with appendices, environmental analysis and budget), developed code.

- **WP 1. Literature review and theoretical background.**
  - State-of-the-art of trajectory optimisation concepts, actual commercial, open-source and academic software used in optimal path planning computation.
  - Identification of the actual research gaps.
  - Theoretical background about flight mechanics applied to bidimensional vertical plane restricted manoeuvres.

- Theoretical background on open-source back-end solvers (p.e CasADi or ACADOS) for optimisation problems.

High-level deliverables: state-of-the-art of actual situation in trajectory optimisation problems and path planning algorithms, a theoretical background on the flight mechanics equations and optimisation solvers.

- **WP 2. Implementation of benchmark problems.**
  - Formulation of the flight mechanics associated to a vertical plane restricted flight trajectory.
  - Implementation and validation of a code that integrates the flight mechanics equations and obtains trim conditions for different flight phases.
  - Implementation of the different benchmark problems considering a descent-cruise-ascent manoeuvre: manoeuvre with constant mass; manoeuvre with mass variation; manoeuvre with mass variation and DEM; manoeuvre with mass variation and NFZ; manoeuvre with mass variation and no calm wind conditions.
  - Implementation of a function that transforms a continuos trajectory in a discretised set of points and converts those relative points in absolute Global Positioning System (GPS) coordinates, given an initial absolute GPS point (i.e. flight plan generator function).
  - Verification of the discretised solution with a real autopilot in a Quasi-Hardware-in-the-Loop (QHIL) environment.

High-level deliverables: the code associated to the defined benchmark problems and the flight plan generator function and, the conclusions about the validation and verification processes.

- **WP 3. Implementation of the water-drop manoeuvre.**
  - Formulation of the flight mechanics associated to a vertical plane water-drop manoeuvre, including discontinuous weight of the UAV during discharge and the aircraft operational limits (i.e problem definition).
  - Pre-processing, when needed, and formulation of the different constraints mentioned: DEM, NFZ and wind conditions [1] (i.e constraints applied to the problem).

[1]Notice that the detected obstacles will be treated as NFZ and, therefore, its definition is not necessary. When the UAV detects a new obstacle on its flight path, the algorithm will receive a set of coordinates to treat it as a new NFZ and then optimal trajectory will be re-computed.

- Implementation and validation of a code capable of computing the optimal flight trajectory to perform the water-drop manoeuvre in a given set of initial conditions (aircraft dynamics and pre-defined constraints) and the GPS discharge point.
- Implementation of changes, if needed, on the flight plan generator function and verification through QHIL.
- Analysis of code performance for different set of constraints.

High-level deliverables: the code associated to the water-drop manoeuvre, the flight plan generator function (if implemented changes). Conclusions about the validation and verification processes and code performance analysis.

- **WP 4. Implementation of a GO/NO-GO logic.**
  - Implementation of a GO/NO-GO logic based on obtaining all the vertical plane flight trajectories for a set of headings and the actual dynamics of the aircraft or, for a set of operations theatre dimensions and the actual dynamics of the aircraft.
  - Representation of all the plane associated trajectories in green (GO) or red (NO-GO) based on the aircraft actual dynamics.
  - Analysis of code performance.

  High-level deliverables: the code associated to GO/NO-GO logic, the flight plan generator function (if implemented changes). Conclusions about the verification process and code performance analysis.

- **WP 5. Code performance dedicated analysis.**
  - Recapitulation and summary about code performance.
  - Proposal of solutions, if needed, to improve code performance to get closer to real-time computation.
  - Comments about the feasibility for an embedded implementation on-board the UAV.

  High-level deliverables: overall analysis and description of the possible solutions to enhance the code performance towards a real-time computation or on-board implementation.

In the following figure, a work breakdown structure can be seen to clarify the work-packages addressed by this project and the tasks related in each case according to an specific codification.

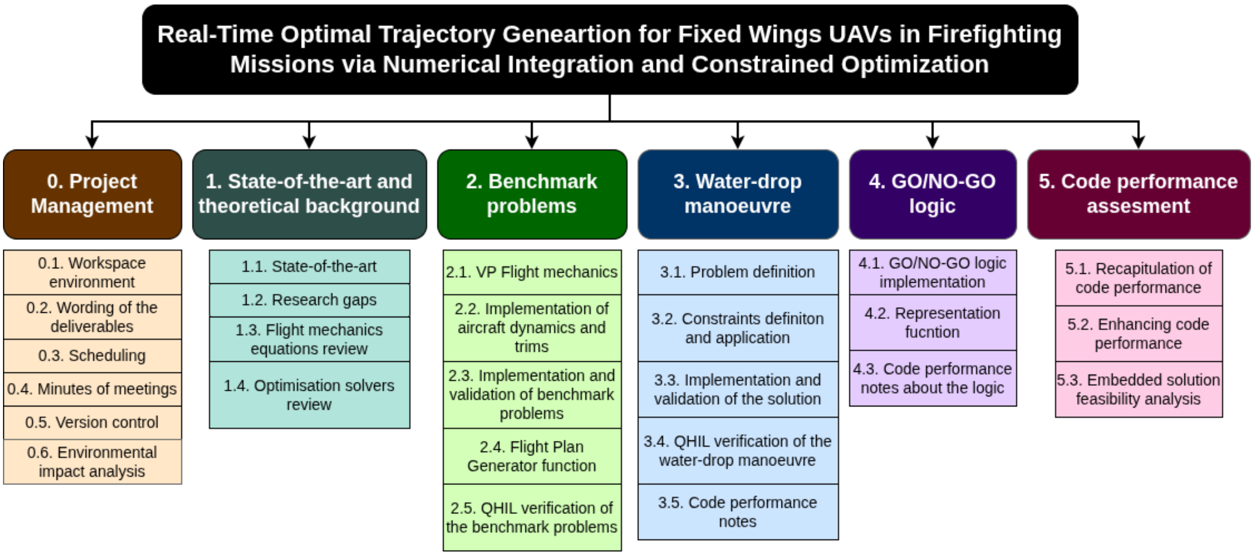

Figure 2.1: Work breakdown structure of the project, including all the work-packages and tasks. Own source.

In addition, the following points will **not** be included on the project development:

- The design of the UAV or its payloads, including any of the phases of the designing process.
- A fully description of the systems on-board the UAV or its characteristics, when non-related to the aim of the project.
- The execution of the computed trajectories in real flight missions or live hardware tests beyond the QHIL environment.
- The computation of flight trajectories for non-firefighting missions (p.e. search and rescue, surveillance, etc.).
- The assessment of environmental factors (e.g., turbulence, sensor noise, GPS drift) under real operational conditions.
- The integration of the algorithm within the payloads on-board the UAV or on ground segment systems, including the interfacing of the trajectory generation module with other on-board systems (e.g. telemetry, propulsion control, payload actuators) or a ground control station.
- Any low-level modification on the actual autopilot firmware, including inner-loop and outer-loop controllers reimplementation or modification.
- A full real-time implementation of the algorithm on resource-constrained embedded hardware, including any type of sensor data fusion.
- A high-fidelity modelling of the water-drop print, the fire plume or a shallow-water model of

the fire environment.

- A benchmarking against hard real-time constraints or performance guarantees under strict time budgets.
- The certification of the provided algorithm according to regulations affecting critical software embedded in Unmanned Aerial Systems (UAS) (i.e. RTCA DO-178C and RTCA DO-254) or cybersecurity related regulations (i.e. RTCA DO-326A, RTCA DO-365A and RTCA DO-355A).
- The migration of the code to the embedded system programming language (i.e C or C++).

Chapter 3

Requirements

This section includes the basic requirements and constraints that have been considered for this project development. Essentially, they have been divided into five different self-defined subsections or categories, as listed below. It has to be pinpointed that each requirement has an associated code in order to maintain traceability throughout all the project development.

- **Software and Tools requirements.**
  - REQ-ST-1: The trajectory optimization algorithm shall be implemented in an open-source programming language (e.g. Python $\geq$ 3.10).
  - REQ-ST-2: The numerical implementation shall use open-source scientific libraries, such asNumPy, SciPy, CasADi, Matplotlib, etc.
  - REQ-ST-3: The source code and documentation shall be maintained using Git version control.
  - REQ-ST-4: All high-level deliverables that are documents itself (report, project charter, appendices, budget and environmental impact) shall be written using LaTeX in Overleaf or in a local machine compiler. The source code must live on the Git repository for version control.
  - REQ-ST-5: No proprietary software (e.g. MATLAB, AMPL) shall be used in the development.
- **Numerics and Algorithmics requirements.**
  - REQ-NA-1: The flight dynamics shall be modelled using a point-mass approximation for fixed-wing UAVs in 2D vertical restricted motion, when needed.

- REQ-NA-2: The integration scheme used for the numerical solution of the equations of motion shall be chosen based on its numerical stability, accuracy, and computational efficiency, ensuring suitability for real-time or near real-time implementation.
- REQ-NA-3: The algorithm shall include environmental and operational constraints, such as DEM, NFZs, wind fields and the operational limits of the aircraft (e.g. maximum bank angle, maximum turn radius, load factors and flight envelope).
- REQ-NA-4: The output trajectories shall be discretized and converted into GPS waypoints expressed in latitude, longitude, and altitude coordinates, ensuring compatibility with any autopilot flight plan format.
- REQ-NA-5: The system shall include a GO/NO-GO logic to assess trajectory feasibility based on UAV dynamic limits.

- **Validation and Verification requirements.**
  - REQ-VV-1: The algorithm shall be verified through numerical benchmark problems, including all the manoeuvres especified by WP2.
  - REQ-VV-2: The computed trajectories shall be validated in a QHIL setup using a real autopilot.
  - REQ-VV-3: The validation and verification processes shall not involve real flight testing or live hardware execution.
  - REQ-VV-4: The performance and correctness of the algorithm shall be assessed against known analytical or reference results when possible.

- **Computational constraints.**
  - REQ-CC-1: All computations shall be feasible on a standard personal computer without GPU acceleration.
  - REQ-CC-2: The algorithm shall balance numerical accuracy and computational efficiency, aiming for near real-time feasibility.
  - REQ-CC-3: The code shall be modular and maintainable, allowing future optimization for embedded implementation, although such implementation is out of scope.
  - REQ-CC-4: Execution times and convergence behaviour shall be analysed to assess code performance.

- **Study constraints.**
  - REQ-SC-1: The project scope is limited to firefighting missions using fixed-wing UAVs.
  - REQ-SC-2: The project shall exclude UAV design, propulsion, or payload development.
  - REQ-SC-3: No modification of autopilot firmware or embedded system programming shall be conducted.
  - REQ-SC-4: No real-time certified software compliance (e.g., DO-178C, DO-254) is required.
  - REQ-SC-5: The project shall include an environmental impact assessment and budget analysis as part of the final deliverables.
  - REQ-SC-6: The thesis shall be completed within the established academic timeline (i.e. January 12, 2026).

Chapter 4

Justification

In recent years, the increasing occurrence and intensity of wildfires have highlighted the urgent need for more effective and autonomous aerial firefighting solutions [1]. From a global perspective, enhancing the autonomy and operational efficiency of UAS for emergency response represents a major step toward safer and more sustainable fire suppression operations. In particular, fixed-wing UAS offer significant advantages over rotary-wing systems, including greater range, endurance, and payload capacity, making them suitable for long-duration firefighting missions in hazardous environments [2].

This project is developed within the framework of an extracurricular internship at Singular Aircraft S.L., an aerospace company founded by Luis Carrillo with over twelve years of experience in the design, manufacture, operation, and certification of Remotely Piloted Aircraft (RPA) and UAV. The company has designed and fully developed the Flyox I, currently the largest civilian fixed-wing UAV designed for firefighting, humanitarian aid, and surveillance missions [3]. Several previous projects and studies carried out within the company have focused on the improvement of the Flyox I's flight systems, payload integration, and mission control capabilities. Building upon these developments, this thesis contributes to the new generation of the Flyox I platform, which aims to introduce advanced levels of autonomy for critical mission phases.

Specifically, this new generation seeks to enable the aircraft to perform the water-drop manoeuvre autonomously, reducing operator workload and enabling missions under low-visibility or high-risk conditions. To achieve this, the aircraft must be equipped with a trajectory computation algorithm capable of planning and re-planning the discharge trajectory in real time, considering aircraft dynamics, environmental constraints, and safety margins, without human validation. This capability represents a fundamental step toward full operational autonomy in firefighting UAS.

The current work focuses on developing and validating such an algorithm through optimal control and numerical optimisation techniques. The approach aims to fill existing gaps in trajectory generation software for UAS by integrating non-standard operational constraints such as DEM, NFZs, atmospheric conditions, and aircraft operational limits. Although not all these constraints will be implemented in the present study, they define the broader framework in which this research is embedded.

The thesis will include the mathematical modelling of the water-drop manoeuvre and apply optimisation on a path planning algorithm. Furthermore, the algorithm will include a decision logic that evaluates the feasibility of each discharge mission based on operator-defined parameters (e.g., drop location, altitude, and heading) and the real-time state of the aircraft. This approach is relatively novel in the context of autonomous UAS firefighting operations.

Validation will be performed in a QHIL environment using the same autopilot as the real Flyox aircraft, ensuring consistency and reliability between simulation and physical systems. This contributes to the practical applicability of the results within Singular Aircraft's future development roadmap.

From a critical standpoint, the proposed approach offers clear advantages: it enables a higher degree of autonomy, improved safety by reducing human involvement in dangerous environments, and adaptability to real-time mission changes. However, potential limitations include the computational cost of real-time optimisation, the need for accurate environmental data, and the complexity of integrating such systems into certified aircraft architectures. Despite these challenges, this research represents an essential step toward achieving the next level of autonomy in firefighting UAS operations.

Chapter 5

Schedule

The following section includes a brief description of the tasks that can be seen on the work breakdown structure and how each task contributes to the achievement of the object of this project. Each description also includes a time estimation (in hours, h) of time required for its completion. Then, the dependencies between tasks is identified to produce a Gantt diagram, that can be seen at the end of the section.

5.1 Tasks description and dependencies

For each task, the description includes the main object, the time estimation (Time in h) and its dependencies with other tasks [Associated Tasks].

WP 0. Project Management.

- **0.1. Workspace environment (5 h) [–]:** Configuration of the workspace environment in local folders but using a cloud repository in GitHub. It also includes the configuration of the LaTeX documents (packages, layouts, etc.).
- **0.2. Wording of the deliverables (60 h) [0.6/WP1/WP2/WP3/WP4/WP5]:** Wording of the thesis project charter, thesis report, appendices, budget, environmental impact and other related documents that could arise.
- **0.3. Scheduling (6 h) [–]:** Identification of time dedication for each task. Gantt diagram production and task allocation on a calendar. The possible changes due to unforeseen circumstances are also considered part of this task.
- **0.4. Minutes of meetings (2 h) [–]:** Wording of the minutes of the meetings done within

the development of the overall project.

- **0.5. Version control (2 h) [WP0/WP1/WP2/WP3/WP4/WP5]:** Git version control implementation on the repository. Also consists on commit modifications every time something (e.g. code, documents) change.
- **0.6. Environmental impact analysis (8 h) [WP1/WP2/WP3/WP4/WP5]:** Production of an environmental impact analysis of the project once it is almost finished (i.e. when all the other WPs are done).

WP 1. State-of-the-art and theoretical background.

- **1.1. State-of-the-art (15 h) [–]:** Investigation and reviewing of actual algorithms used in trajectory optimisation and analysis of the back-end used in existing tools, examining the advantages and disadvantages of each.
- **1.2. Research gaps (3 h) [1.1]:** From the state-of-the-art and actual tools, the research gaps have to be identified in order to understand which are the main features the algorithm that is going to be developed has to include.
- **1.3. Flight mechanics equations review (13.5 h) [–]:** Mathematical review and understand of the flight mechanics equations regarding vertical plane restricted trajectories for an aircraft.
- **1.4. Optimisation solvers review (13.5 h) [–]:** Mathematical review and understand of the different optimisation methods applied for solving optimisation problems in the existing open-source algorithms.

WP 2. Benchmark problems.

- **2.1. VP Flight mechanics (6 h) [–]:** Mathematical formulation of the equations of motion for the benchmark problems defined: .
- **2.2. Implementation of aircraft dynamics and trims (14 h) [2.1]:** Implementation of the equations in a code program, including the trim calculation for a given flight phase.
- **2.3. Implementation and validation of benchmark problems (30 h) [1.3 / 1.4 / 2.2]:** Implementation of the equations of motion for an aircraft point mass model with restricted motion in a code program for three different benchmark cases. All cases have to be validated.
- **2.4. Flight Plan Generator function (6 h) [2.3]:** Generation of a function that discretises

the continuous trajectory into a set of waypoints, which converts the relative position into an absolute GPS position for a given initial point. The output of the function has to be in flight plan file format.

- **2.5. QHIL verification for benchmark problems (12 h) [2.4]:** Planning and execution of verification tests under QHIL environment. This task may include planning, preparation, acceptance criteria definition and testing.

WP 3. Water-drop manoeuvre.

- **3.1. Problem definition (10 h) [–]:** Definition of the water-drop manoeuvre including all the relevant phases of the manoeuvre, its trim conditions for each phase and the operational limits affecting each phase.
- **3.2. Constraints definition and application (16 h) [3.1]:** Definition of the constraints the code will handle DEM, NFZs, wind conditions.
- **3.3. Implementation and validation of the solution (60 h) [3.1 / 3.2]:** Implementation of the equations of motion for an aircraft point mass model with restricted vertical plane motion in a code program for the water drop manoeuvre. Validation of the code using the water-drop manoeuvre itself or a benchmark problem from the previous WP.
- **3.4. QHIL verification of the water-drop manoeuvre (12 h) [WP2 / 3.4]:** Planning and execution of verification tests under QHIL environment regarding the water-drop manoeuvre. This task may include planning, preparation, acceptance criteria definition and testing.
- **3.5. Code performance notes (8 h) [3.5]:** Analysis of the code performance analysis. This task may include planning,definition of code performance criteria that has to be analysed and testing.

WP 4. GO / NO-GO logic.

- **4.1. Go / No-Go logic implementation (20 h) [3.4]:** Implementation of a function that logically accepts or deny a solution regarding constraints compliance.
- **4.2. Representation function (4 h) [4.1]:** Implementation of a function that represents in a plot the vertical planes and its trajectories in different colours when the trajectory its feasible (Go) and not feasible (No-Go) for a given set of headings and and initial altitude or point.

- **4.3. Code performance notes about the logic (16 h) [4.2]:** Analysis of the code performance analysis for different criteria (e.g. different initial conditions, different optimality conditions, etc.). This task may include planning,definition of code performance criteria that has to be analysed and testing.

WP 5. Code performance assessment.

- **5.1. Recapitulation of code performance (6 h) [3.6 / 4.4]:** Summary of the conclusions about code performance as result of previous tasks 3.6 and 4.4.
- **5.2. Enhancing performance (6 h) [5.1]:** Conduct a research about what measures could be implemented for enhancing the critical points in code performance from previous task.
- **5.3. Embedded solution feasibility analysis (6 h) [5.2]:** Description of the implementations that could be made to enhance code performance towards a real time embedded solution in a microcontroller.

5.2 Gantt diagram

The following page includes a landscape Gantt diagram, including all the tasks listed above and with its dependencies mark on the Gantt. As can be seen, the timeline is in days and the dedication per day is, 2h per day on WP0, WP1, WP2 and WP4. For WP3 and WP5 the dedication is represented by 3h per day. In addition, it has to be pinpointed that the wording of the deliverables includes this project charter writing and is the only task, along WP4, that is carried out alongside other tasks.

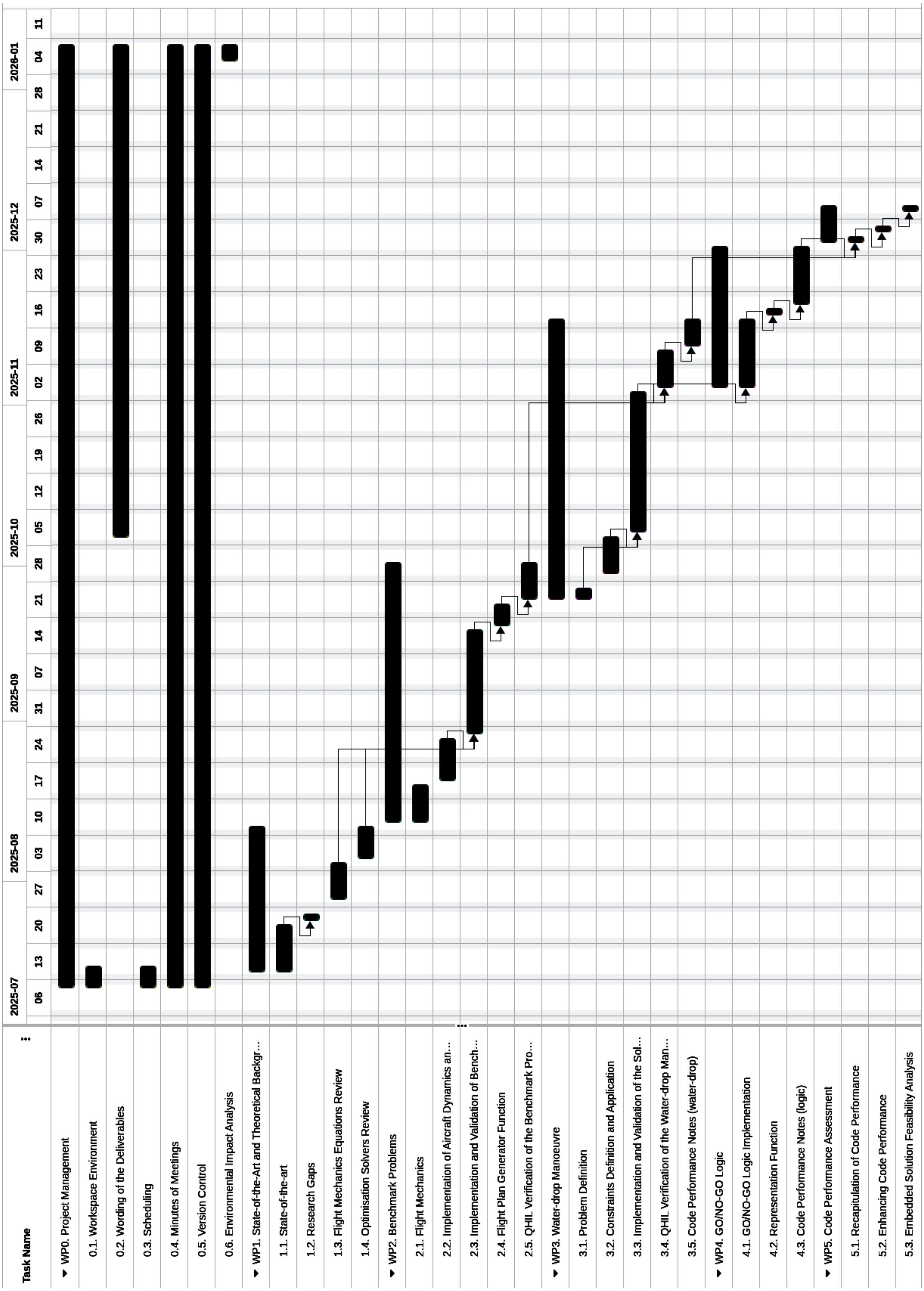

Figure 5.1: Gantt diagram of the project Real-Time Optimal Trajectory Generation for Fixed-Wing UAVs. Own source.

References

1. FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS. *Global Forest Fire Assessment 2023* [`https://www.fao.org/forest-resources-assessment/fire/en/`]. 2023. Accessed: 2025-10-08.
2. RESTAS, Agoston. Drone Applications for Wildfire Management: From Detection to Suppression. *Fire Technology*. 2022, vol. 58, no. 3, pp. 1011–1036. Available from DOI: `10.1007/s10694-021-01169-9`.
3. SINGULAR AIRCRAFT S.L. *Flyox I – The World's Largest Civilian UAV* [`https://singularaircraft.com/`]. [N.d.]. Accessed: 2025-10-08.