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

List of Tables

Acronyms

DEM Data Elevation Model.

GPS Global Positioning System.

NFZ No-Fly Zones.

QHIL Quasi-Hardware-in-the-Loop.

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 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 trajectory computation.
  - Identification of the actual research gaps.
  - Theoretical background about the optimal control problem and the direct and indirect approaches used to solve it.

   - Theoretical background on gradient descent techniques to solve optimisation problems by the simplification of the problem to an adjoint problem.

  High-level deliverables: state-of-the-art of actual situation in trajectory optimisation problems and, a theoretical background on the optimal control problem, gradient descent techniques and the adjoint problem.

- **WP 2. Implementation of benchmark problems.**
   - Formulation of the flight mechanics associated to a vertical plane restricted flight trajectory and to a coordinated turn restricted in the horizontal plane.
   - Implementation and validation of a code that integrates the flight mechanics equations using a trapezoidal scheme in three different scenarios that would constitute the benchmark problems: a flight path that consists in an descent, cruise level-flight and an ascent; a restricted trajectory between two points A and B; a coordinated turn in an horizontal plane with a wind-field (adapted version of Zermelo's problem).
   - 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 three 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.**
   - Requirements of the stakeholders of the project.
   - 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.
   - Pre-processing, when needed, and formulation of the different constraints mentioned: DEM, NFZ and wind conditions. [1]

[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 recomputed.

  - 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 for a minimum time flight trajectory.
  - 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.
  - Representation of all the plane associated trajectories in green (GO) or red (NO-GO) based on the aircraft actual dynamics.
  - Implementation of a function that choices the most suitable trajectory to descent, discharge and ascent safely within the aircraft operational limits.
  - 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.

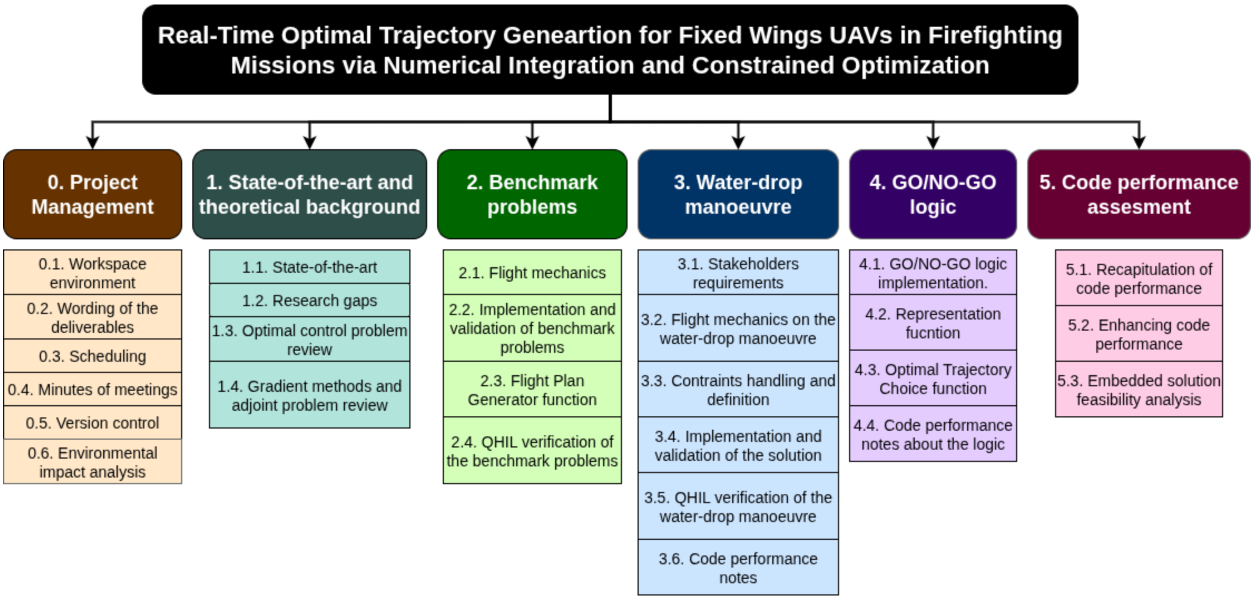

Figure 2.1: Work breakdown structure of the project, including all the work-packages and tasks. Own source.

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 ≥ 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) or planar (horizontal) 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 optimization shall rely on gradient-based methods, using either finite-difference or adjoint-based gradient computation.
  - REQ-NA-4: 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-5: 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-6: 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 descent–cruise–ascent, A to B trajectory, and horizontal Zermelo navigation.
  - 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

Justificar la necesidad de este proyecto mediante la inexistencia de calculadores de trayectorias optimas en maniobras de extinción de incendios para UAVs (campo inexplotado en esta concreción, pues algoritmos de cálculo de trayectorias ya existen). Añadir el contexto del proyecto de Singular Aircraft, su definición, las necesidades del Flyox en cuanto a autonomia y la maniobra de descarga. Destacar la visión global del proyecto pero profundizar en las partes específicas, detallando las ventajas y desventajas de la visión que se pretende implementar.

Approach and explanation of the need to carry out the project from a global vision, i.e. detailing why this project may be necessary.

Furthermore, in case the thesis is based on another project or is developed within the framework of a company or research group, the section must also include a brief explanation about the context of the work, that is, about possible related projects and thesis that have been done before, focusing and contextualizing the work.

Although this section begins with a global vision, it must approach a more specific vision, detailing the possible advantages and disadvantages that the student's approach may have.

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.

Tasks description including time estimation, dependencies table and explanation, Gantt diagram of WP and tasks and explanation

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 (8 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 (100 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 (8 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 (2.5 h) [–]:** Wording of the minutes of the meetings done within the development of the overall project.
- **0.5. Version control (2.5 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 (12 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 (X h) [–]:**
- **1.2. Research gaps (X h) [–]:**
- **1.3. Optimal control problem review (X h) [–]:**
- **1.4. Gradient methods and adjoint problem review (X h) [–]:**

WP 2. Benchmark problems.

- **2.1. Flight mechanics (X h) [–]:**
- **2.2. Implementation and validation of benchmark problems (X h) [–]:**
- **2.3. Flight Plan Generator function (X h) [–]:**
- **2.4. Gradient methods and adjoint problem review (X h) [–]:**

WP 3. Water-drop manoeuvre.

- **3.1. Stakeholders requirements (X h) [–]:**
- **3.2. Flight mechanics on the water-drop manoeuvre (X h) [–]:**
- **3.3. Constraints handling and definition (X h) [–]:**
- **3.4. Implementation and validation of the solution (X h) [–]:**
- **3.5. QHIL verification of the water-drop manoeuvre(X h) [–]:**
- **3.6. Code performance notes (X h) [–]:**

WP 4. GO / NO-GO logic.

- **4.1. Go / No-Go logic implementation (X h) [–]:**
- **4.2. Representation function (X h) [–]:**
- **4.3. Optimal Trajectory Choice function (X h) [–]:**
- **4.4. Code performance notes about the logic (X h) [–]:**

WP 5. Code performance assessment.

- **5.1. Recapitulation of code performance (X h) [–]:**
- **5.2. Enhancing performance (X h) [–]:**
- **5.3. Embedded solution feasibility analysis (X h) [–]:**

5.2 Gantt diagram