

BACHELOR FINAL THESES



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:

Report

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

Abstract

Acknowledgments

,3Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Preface

,10Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Scope and Objectives

Motivation and Justification

Methodological Approach

Document Structure

Contents

List of Figures

List of Tables

Acronyms

DCM Direct Collocation Method.

KKT Karush-Kuhn-Tucker.

NLP Non-linear Programming.

OCP Optimal Control Problem.

OCT Optimal Control Theory.

ODE Ordinary Differential Equation.

PDE Partial Differential Equation.

PMP Pontryagin's Maximum Principle.

TPS Throttle Position Sensor.

Nomenclature

| Symbol | Description | Units |
|----------|---|-------|
| α | Angle of Attack, angle between the aircraft's x-body axis (x_b) and the projection of the velocity vector onto the aircraft's plane of symmetry ($x_b - z_b$) plane | [rad] |

1 | Fundamentals of Optimal Control and its Implementation

This chapter includes the mathematical formualtion of the Optimal Control Problem (OCP), which is the base of the Optimal Control Theory (OCT). It also includes a careful explanation about direct and indirect methods, which are two different approaches to solve optimal control problems, highlighting the Direct Collocation Method (DCM) and Pontryagin's Maximum Principle (PMP). Eventually, the Interior Point OPTimizer (IPOPT) is presented as the used approach, including the mathematical formulation of the method and its implementation from CasADi's Python library.

1.1 Introduction to Optimal Control Theory

The OCT is a control field related branch where an objective function has to be optimised -most of the times, it has to be minimised- in order to find the control trajectory for a determined dynamical system. This theory is historically realted with calculus of variations, where optimal points -either maxima or minima- are found using little variations in functions or functionals ([CITA 1](#)). This field of study, proposed by Isaac Newton, was developed as a solving approach for the brachistochrone problem, posed by Bernoulli in 1696. The brachistochrone curve is defined as the fastest descent path -most optimal path for minimising time- between two points A and B under a uniform gravitational field. Counterintuitively, it was found that the most optimal path was not but the cycloidal ramp, as can be seen in the image below, where the ball has to roll under gravity only at the different paths, and the same grade shaded figures correspond to same time instants.

In essence, the OCT responds the need to solve continuous time optimisation problems, as once presented. Thus, the OCP can be understood as a n-dimensional extension of the Non-linear Programming (NLP) problem. The NLP problem can be defined as the minimisation of a given function subject to a different set of equations or inequations that act as restrictions -also called constraints- and simple bounds ([CITA 3](#)).

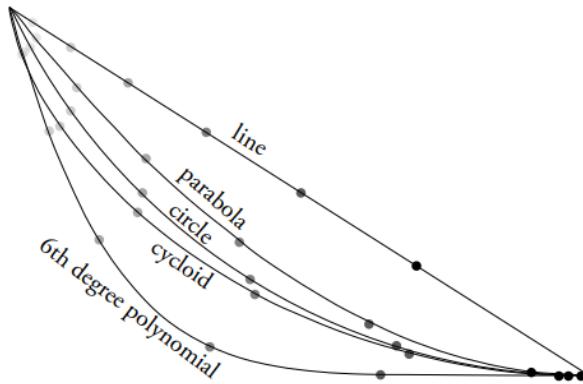


Figure 1.1: The brachistochrone problem. The ball rolls down the fastest on a cycloidal ramp. Points of the same shade correspond to the same moment in time. Extracted directly from [CITA 2](#)

$$\begin{aligned}
 & \max_{x \in \mathbb{R}^n} f(x) \\
 \text{s.t. } & g_j(x) \leq b_j \text{ for each } j = 1, \dots, m \\
 & x = (x_1, \dots, x_n) \\
 & g = (g_1, \dots, g_m).
 \end{aligned} \tag{1.1}$$

The NLP problem is limited to a finite number of state variables and restrictions, since it has a discretised treatment. At this point, it has to be mentioned that any NLP problem goes through three different phases, which are applicable to the OCP or any optimisation problem. The first phase is modelling, which is, in fact, a mathematical description of a dynamic system through simple equations that allow future state predictions of a system thanks to a given set of control variables. While state variables $y_i(t)$ are the variables that describe system conditions at a given time, control variables $u_i(t)$ are the variables that can be manipulated through time because they have an influence on the system. In other words, state variables are consequence of the control variables -p.e for a flight trajectory, the position and velocity of the aircraft would be the state variables for a set of different control variables as could be the α or the Throttle Position Sensor (TPS)-. Secondly, the restrictions have to be applied to the system in the form of equality or inequality constraints to describe its physical or operational limitations. In addition, the admissible trajectory and admissible control are described; that means that state and control variables satisfy the restrictions within all time domain, respectively -later will be referred to as simple bounds-. Third and last phase is optimisation, which means the maximisation or minimisation of the cost function or functional associated with the system while complying with all the constraints and bounds set for a determined optimisation criteria. For example, in the brachistochrone problem introduced before, there are several feasible paths between points A and B, but only the cycloidal path complies with the minimum time optimisation criteria ([CITA 4](#)).

It has to be pinpointed that the OCP is a n-dimensional interpretation of the NLP because state and control variables are time-continuous functions and, therefore, the problem's objective is to find the n_k -dimensional vector of state functions $y_k(t)$ and control functions $u_k(t)$ that optimise a determined functional $F(y, u)$.

1.1.1 Mathematical formulation

The OCP is defined as follows.

$$\begin{aligned}
 \min_{y,u} \quad & J(y, u) := \Phi[y(t_f), t_f] + \int_{t_0}^{t_f} L[y(t), u(t), t] \, dt \\
 \text{s.t.} \quad & \dot{y} = f[y(t), u(t), t] \\
 & g[y(t), u(t), t] \leq 0 \\
 & \varphi[y(t_0), u(t_0), t_0] = 0 \\
 & y_{lb} \leq y(t) \leq y_{ub} \\
 & u_{lb} \leq u(t) \leq u_{ub} \\
 & t \in [t_0, t_f]
 \end{aligned} \tag{1.2}$$

The previous description is the formal description for the OCP, since it contains a cost functional $J(y, u)$ that has to be minimised subject to a set of constraints and simple bounds **CITAS 5 y 6**. Specifically, the description above has the following terms:

- State variables $y(t)$: As explained before, state variables are time continuous functions that describe the system condition and allow to predict its future coindition. This state variables are limited by lower and upper bounds, y_{lb} and y_{ub} , respectively. Those bounds are the physical or operational limits that each state variable has; for example, on the brachistochrone problem, the horizontal position,of the ball -when movement is restricted to the vertical plane- cannot be lower than the point A horizontal coordinate and bigger than the point B horizontal coordinate -those are the simple bounds for horizontal position state variable in that case-.
- Control variables $u(t)$: Control variables are time continuous functions that describe the changes on the system to achieve a determined state or condition through time. Similarly, they are simple bounded, in this case by lower bound u_{lb} and upper bound u_{ub} ; for example, for the flight trajectory previously described, control variable α could be lower bounded by a null value and upper bounded by the stall angle, α_{stall} -those are the simple bounds for the α control variable in that case-.
- Time domain t : Time domain is the closed interval of time, defined on real numbers, between initial time t_0 -where initial state condition is applied- and the time horizon t_f -that can be fixed or free-. On OCP, time horizon can be set to a known value if its a problem's parameter or can be free if it is set a decision variable -also called control variable- of the problem through end time constraints or inside the cost functional.

- Dynamic constraints \dot{y} : Dynamic equations are a set of Ordinary Differential Equation (ODE)s equality constraints that describe the physics of the problem. This set of constraints are always active, which means that they apply for all time domain.
- Path constraints $g[y(t), u(t), t]$: Path constraints are another set of expressions that apply for a given time domain and that could refer to any type of extra restrictions to the problem. They could be equalities or inequalities that could both restrict state or controls in a determined way.
- Initial state conditions $\varphi[y(t_0), u(t_0), t_0]$: The initial state constraints are a set of equalities that restrict the initial value of states variables to a known value -the initial condition of the system is known-. They can also include restrictions for the initial value of control variables; nevertheless, it is not a common practice because the problem could be overrestricted and the solver may be unable to find a feasible solution.
- Cost functional $J(y, u)$: The cost functional is the cost objective function that has to be minimised to find the optimal solution. This cost is defined by two different terms that contribute differently to the whole objective: terminal and running costs. If both contributions are present, then the cost objective is defined as in its Bolza form. As clarified in [Bryson, A. E., & Ho, Y. C. \(1975\). Applied Optimal Control: Optimization, Estimation, and Control. Hemisphere Publishing Corp:](#)
 - Terminal cost is the contribution that only penalises the final value of state variables. Is represented by $\Phi[y(t_f), u(t_f), t_f]$ and, if cost objective $J(y, u) = \Phi[y(t_f), t_f]$, is it said that the whole cost functional is in its Mayer form.
 - Running cost is the contribution that penalises each state and control variable through time and, therefore, accumulates the whole penalisation. Is represented by $\int_{t_0}^{t_f} L[y(t), u(t), t] dt$ and, if cost objective $J(y, u) = \int_{t_0}^{t_f} L[y(t), u(t), t] dt$, is it said that the whole cost functional is in its Lagrange form.

1.2 Direct and Indirect Methods

Once the basic problem has been set, it has to be noticed that although the variables are time-continuous, they have to be discretized in order to solve the problem. This discretization is necessary because the problem is not affordable analytically and it has to be treated numerically. This implicates to discretize the time domain, converting the infinite-dimensional treatment inherited from the running cost into a sum of different time-instants. There are two different types of approaches, as stated by John T. Betts on [CITA 5](#):

- Direct methods, which are based in the evaluation of the images from a function and the comparison between different values at different instants to find the derivative -p.e Direct collocation method-.

- Indirect methods, which are based on direct evaluation of the derivative and its proximity to zero value. In this case, there is need to explicit derivation of the equations and is a method subjected to initial conditions -p.e Pontryagin's maximum principle-.

1.2.1 Direct collocation method as direct method

Direct collocation methods are a direct method in which the functions are discretised in a set of predefined points called collocation points. These collocation points are part of a collocation grid that is set along the domain in both state and control variables. It has to be pinpointed that between collocation methods two different types can be distinguished: low-order collocation methods, applied to solve ODEs, and pseudospectral methods, applied to solve Partial Differential Equation (PDE)s. Since this bachelor thesis addresses the flight mechanics of an aircraft for low-flight manoeuvres, which dynamic equations consists on different ODEs, the wording will focus on low-order collocation methods (DCM from now on) [CITA 9](#).

Essentially, in the DCM the state and the control variables are both discretised in different collocation points. This collocation points are a set of N time-instants that form the collocation grid (in an equivalent way, the reader can think about the collocation grid as the element grid usually made for FEM or CFD). On this points, the dynamic of the problem will be evaluated, or forced, to solve the equations and, therefore, the whole problem. Thus, the DCM for the control variables is defined as follows, extracted from [CITA 7](#).

$$u(t) = \begin{cases} U_k^u(u_k, u_{k+1}, t) & \forall t \in [t_k, t_{k+1}] \\ U_{N-1}^u(u_{N-1}, u_N, t) & \forall t \in [t_{N-1}, t_N] \end{cases} \quad (1.3)$$

$$\bar{u} = [u_0, u_1, u_2, \dots, u_N]^T \in \mathbb{R}^{(N+1) \cdot n_u} \quad (1.4)$$

The previous expressions [\(??\)](#) and [\(??\)](#) are the time-domain discretisation in N subintervals with specific nodes t_i with $i \in [0, N]$, what allows to obtain a decision vector for control. In other words, it is an interpolation that avoids sudden discontinuities that can generate instabilities on the integrator or solver, allowing a more precise capture of the different variations. In an equivalent way, the discretisation for state variables can be obtained as:

$$\bar{y} = [y_0, y_1, y_2, \dots, y_N]^T \in \mathbb{R}^{(N+1) \cdot n_y} \quad (1.5)$$

For a better comprehension, the mathematical formulation behind discretisation can be simplified to the following scheme, which exemplifies the conversion of a time-continuous function to its discretised form.

$$\begin{aligned} u(t) &\longrightarrow u_i = u(t_i) \quad \text{for each } t_i \text{ with } i \in [0, \dots, N] \\ y(t) &\longrightarrow y_i = y(t_i) \quad \text{for each } t_i \text{ with } i \in [0, \dots, N] \end{aligned}$$

At this point, a trapezoidal scheme is proposed to be used on the numerical approach since it takes into account not only the present time-instant but the following time-instant for the function evaluation and its integration. This numerical integration scheme is preferred above others such Runge-Kutta or Hermite-Simpson because of its simplicity, its easy applicability and its precision, which is enough for the scope of this project. The trapezoidal integration scheme for the running cost, adapted from [CITAS 8 y 9](#), can be state as:

$$\int_{t_0}^{t_f} L(y, u, t) dt \approx \sum_{k=0}^{N-1} \frac{1}{2} h_k \cdot (L_k + L_{k+1})$$

where

$$h_k = t_{k+1} - t_k \tag{1.6}$$

$$L_k = L[y(t_k), u(t_k), t_k]$$

$$L_{k+1} = L[y(t_{k+1}), u(t_{k+1}), t_{k+1}]$$

$$t_k \text{ for each } k \in [0, N - 1]$$

Furthermore, the trapezoidal rule has to be applied to the constraints of the problem, now called collocation constraints. For the simple bounds, path and initial state constraints, the application of the discretisation scheme is easy to reach because the equations have to be evaluated at the given colocation points directly, without need of trapezoidal rule. About the dynamic equations, which are the only ones that have to be integrated by applying the fundamental theorem of calculus, the following development can be obtained, extracted from [CITA 8](#).

$$\begin{aligned} \dot{y} &= f(y, u, t) \\ \int_{t_k}^{t_{k+1}} \dot{y} dt &= \int_{t_k}^{t_{k+1}} f(y, u, t) dt \end{aligned}$$

Resulting in:

$$y_{k+1} - y_k = \frac{1}{2} h_k \cdot (f_k + f_{k+1}) \tag{1.7}$$

where

$$f_k = f(y_k, u_k, t_k) \text{ for each } k \in [0, N]$$

Eventually, the terminal cost can be added taken into account its contribution on final time-instant only. Thus, the optimal control problem can be written in its discretized form in a way such that could be solved applying a DCM.

$$\begin{aligned}
 \min_{y,u} \quad & J(y, u) := \Phi[y(t_N), u(t_N), t_N] + \sum_{k=0}^{N-1} \frac{1}{2} h_k \cdot (L_k + L_{k+1}) \\
 \text{s.t.} \quad & y_{k+1} = y_k + \frac{1}{2} h_k \cdot (f_k + f_{k+1}) \\
 & g[y(t_k), u(t_k), t_k] \leq 0 \text{ for } \forall k \\
 & \varphi[y(t_0), u(t_0), t_0] = 0 \\
 & y_{lb} \leq y(t_k) \leq y_{ub} \\
 & u_{lb} \leq u(t_k) \leq u_{ub} \\
 & t_k \text{ for each } k \in [0, N]
 \end{aligned} \tag{1.8}$$

1.2.2 Pontryagin's maximum principle as indirect method

As said before, the indirect methods search a solution for the optimality conditions directly. In this context, they doesn't need to discretise the control variables but they will need an approximation; thus, they are subject to the initial conditions. Moreover, since the indirect methods are based on the calculus of variations, they also need to comply a set of necessary conditions called transversality conditions. These remarks about conditions are important and are stated before presenting the PMP.

The optimality conditions are consequence of the application of the Euler-Lagrange equation, which can be found on [CITA 14](#). They mean to be the necessary but not sufficient conditions for the resolution of the optimisation problem; in other words, they are three proof that the solution found is actually the optimal one. They are careless of meaning until they are directly found for the problem. They are also called Karush-Kuhn-Tucker (KKT) conditions. Essentially, they are a set of equations related to the gradient or derivative of a functional respect other parameters and variables that have to be null in order to verify optimality. A general optimisation problem and its Lagrangian can be defined as follows.

$$\begin{aligned}
 \min_{x \in \mathbb{R}^n} \quad & f(x) \\
 \text{s.t.} \quad & g_i(x) \leq 0 \text{ for each } i = 1, \dots, m \\
 & h_j(x) = 0 \text{ for each } j = 1, \dots, p
 \end{aligned} \tag{1.9}$$

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{i=0}^m \lambda_i g_i(x) + \sum_{j=0}^p \mu_j h_j(x) \tag{1.10}$$

The KKT conditions are stationarity, complementary slackness, primal feasibility and dual feasibility. Its meaning and mathematical formulation can be seen below as an adaptation from [CITA 15](#).

- Stationarity. Referred to the balanced effect of the gradient of the active constraints and the gradient of the objective functions. In other words, it describes how each

constraint influences the cost function itself, ensuring that at the optimum no further improvement is possible without violating constraints.

$$\nabla_x \mathcal{L} (x, \lambda, \mu) = \nabla f(x) + \sum_{i=0}^m \lambda_i \nabla g_i(x) + \sum_{j=0}^p \mu_j \nabla h_j(x) = 0 \quad (1.11)$$

- Complementary slackness. The multiplier reflects how much the constraint influences the optimal solution. For each inequality constraint, either the constraint is active and its Lagrange multiplier can be positive, or the constraint is inactive and its multiplier is zero.

$$\lambda_i g_i(x) = 0 \quad (1.12)$$

- Primal feasibility. Ensures that the found solution satisfies the original constraints of the problem, which means that the inequality conditions must be held and the equality conditions must be satisfied totally.

$$\begin{aligned} g_i(x) &\leq 0 \\ h_j(x) &= 0 \end{aligned} \quad (1.13)$$

- Dual feasibility. Related to inequality constraints multipliers, they verify they are nonnegative values and describe how would the cost and the solution change if these constraints are relaxed or inactivated.

$$\lambda_i \geq 0 \quad (1.14)$$

On its part, the transversality conditions are the boundary conditions for the co-state variables λ . They are a necessary condition for optimisation in state free-end problems, which means that there's no end known for the trajectory. This transversality conditions can be defined as the existence of a n-vectorial function of the co-state variables $\lambda(t)$ and a m-vectorial function $\nu(t)$ such that the Hamiltonian function $H(\lambda, \nu, t)$, see equation (??), can be defined like in a boundary value problem in his canonical form with ODEs in all time domain $t_0 \leq t \leq t_f$ CITAS 11 y 12.

$$H (\lambda, \nu, t) = \lambda^T f + \nu^T g \quad (1.15)$$

Once the Hamiltonian is defined, both transversality conditions can be found as the partial derivative of the Hamiltonian respect to the state variables and, the evaluation of the co-state at the end of time domain, respectively. Formulation has been adapted from CITAS 11 y 13.

$$\dot{\lambda} = -\frac{\partial H}{\partial y} = -\lambda^T \frac{\partial f}{\partial y} - \nu^T \frac{\partial g}{\partial y} \quad (1.16)$$

$$\lambda(t_f) = \frac{\partial \Phi}{\partial y(t_f)} - \nu^T \frac{\partial \varphi}{\partial y(t_f)} \quad (1.17)$$

Eventually, the PMP is a fundamental result of optimal control theory, since it establishes the necessary conditions to find the optimal trajectory for control variables. In other words, this principle gives the strategy that has to be followed on the control variables, the ones that can be modified through time on the whole problem, to minimise the cost functional while the system dynamic and its constraints are satisfied. From [CITA 16](#), the PMP is defined as the partial derivative of the Hamiltonian function respect the control variables for a problem where $y(0) = y_0 \in \mathcal{S}$ and $u(t)$, a control trajectory, has a state trajectory $y(t)$ associated for a given dynamic system. Thus, a co-state trajectory $\lambda(t)$ can be defined as follows.

$$\begin{aligned} \dot{\lambda}(t) &= -\frac{\partial H}{\partial y} \\ \lambda(T) &= \frac{\partial \Phi}{\partial y(T)} \end{aligned} \quad (1.18)$$

where

$$\begin{aligned} H(y, u, \lambda, t) &= L(y, u, t) + \lambda^T f(y, u, t) \\ \dot{y}(t) &= f(y, u, t) \end{aligned}$$

Therefore, the PMP can be expressed as the minimisation of the Hamiltonian with respect to the control variables, as indicated in equation [\(??\)](#). This formulation arises because the original statement of PMP is given as a maximisation problem, but it can be equivalently rewritten in minimization form by changing the sign of the Hamiltonian [CITA 17](#) -also called Pontryagin's Minimum Principle-. In essence, the principle reformulates the original optimal control problem into an equivalent system involving the state and an adjoint -or costate- system. The optimal control trajectory is then obtained by ensuring that, at each instant of time, the Hamiltonian is minimised with respect to the admissible controls.

$$u(t) = \arg \min_{u \in U} H(y, u, \lambda, t) \quad (1.19)$$

PENSAR SI AÑADIR FORMULACIÓN DEL PROBLEMA CON ADJUNTO PARA CLARIFICACIÓN EN APÉNDICES (APÉNDICES A Y B DE LIBRETA)

1.3 Introduction to IPOPT

The IPOPT is a robust interior-point line-search algorithm designed to solve large-scale NLP problems. This method is particularly well-suited for high-dimensional trajectory optimisation, such as the low-altitude flight manoeuvres required during water-discharge phases in firefighting missions -the primary focus of this thesis-.

IPOPT identifies local optimal solutions by employing a primal-dual barrier approach to handle inequality constraints and a filter-based line-search method to ensure global convergence. The following subsections detail the mathematical formulation of the IPOPT algorithm and its practical implementation via the CasADi numerical optimisation framework.

1.3.1 Mathematical formulation

1.3.2 Implementation in CasADi