Department of Mathematics and Computer Science

University of Heidelberg

Master thesis

in Scientific Computing

submitted by

Athanasios Raptakis

born in Athens, Greece

2020

Optimization of Flight Trajectories of an Aircraft with respect to Fuel Consumption

This master thesis has been carried out by Athanasios Raptakis

at the

Institute for Applied Mathematics

under the supervision of

Prof. Dr. Hans Georg Bock

Prof. Dr. Ekaterina Kostina

Acknowledgements:

I would like to express my appreciation to Prof. Hans Georg Bock and Prof. Ekaterina Kostina for guiding me throughout the whole period of work on this thesis, and for introducing me to the problem of Optimization of Aircraft Trajectories with respect to fuel consumption.
I would also like to thank Dr. Leonard Wirsching and Dr. Andreas Sommer for helping me to start off with MUSCOD and for many helpful suggestions on the problem formulation.
And of course, there are no words that would be enough to express my gratitude to my parents who invested every possible effort to help me to go through my studies.

Abstract:

This master thesis project tackled the problem of trajectory optimization of aircraft with respect to fuel consumption from the optimal control perspective. The problem was formulated as a multi stage optimal control problem in order to represent the climb, cruise and descent flight phases. The standard atmosphere model was used to determine the temperature, pressure and air density. Jet engine thrust and thrust specific fuel consumption were modeled according to BADA 4 models. Minimum fuel and minimum time trajectories were compared and extensive research was done for different flight parameters, like maximum cruising altitude and maximum flight duration. It was demonstrated that a fuel optimal trajectory results in significant savings by adjusting the thrust and angle of attack control policy. An airplane consumes less fuel, when it flies at higher cruising altitudes. Furthermore, we can achieve the biggest fuel savings, when the flight duration is optimized as a free parameter together with fuel. The numerical experiments where carried out with MUSCOD II which implements a direct optimal control method and all the used simulation and optimization algorithms.

Optimierung der Flugbahnen eines Flugzeugs hinsichtlich des Treibstoffverbrauchs:

Dieses Masterarbeitsprojekt befasste sich mit dem Problem der Flugbahnoptimierung von Flugzeugen hinsichtlich des Treibstoffverbrauchs aus der Perspektive der mathematischen Optimalsteuerung. Das Problem wurde als ein mehrstufiges Optimalsteuerungsproblem formuliert, um die Steig-, Reise- und Sinkflugphasen darzustellen. Zur Bestimmung der Temperatur, des Drucks und der Luftdichte wurde das Standardatmosphärenmodell verwendet. Der Triebwerksschub und der schubspezifische Treibstoffverbrauch wurden nach BADA 4-Modellen modelliert. Zeitoptimale Trajektorien und Trajektorien, die den Treibstoffverbrauch minimieren, wurden verglichen und es wurden umfangreiche Untersuchungen für verschiedene Flugparameter, wie die maximale Reiseflughöhe und die maximale Flugdauer, durchgeführt. Es konnte gezeigt werden, dass eine durch Anpassung der Schub- und Anstellwinkelregelungspolitik bezüglich des Treibstoffverbrauchs optimierte Flugbahn zu erheblichen Einsparungen führt. Ein Flugzeug verbraucht weniger Treibstoff, wenn es in höheren Reiseflughöhen fliegt. Darüber hinaus können wir die größten Treibstoffeinsparungen erzielen, wenn die Flugdauer als freier Parameter zusammen mit dem Treibstoffverbrauch optimiert wird. Die numerischen Experimente wurden mit MUSCOD II durchgeführt, welches eine direkte Optimalsteuerungsmethode und alle verwendeten Simulations- und Optimierungsalgorithmen implementiert.

Contents

1 Introduction

Fuel consumption reduction problem

This master thesis project tackles the problem of trajectory optimization of aircraft with respect to fuel consumption from the optimal control perspective. In order to combat climate change and global warming caused by CO_2 emissions, the European Union has adopted sustainability policies for a climate-neutral and emission-free mobility by 2050 [1],[2]. According to "European Aviation Environmental report 2019" [3] aviation currently accounts for 3% of global carbon emissions and long-term forecasts indicate that air traffic is expected to continue increasing.

Moreover, the recent Covid-19 pandemic has created a financial crisis for the aviation sector worldwide. According to "COVID19 impact on European air traffic EUROCONTROL comprehensive assessment" by EUROCONTROL [4], traffic at EU during April and May was reduced more than 80 % and during September was reduced 54% below 2019 levels. Hence, reduction of fuel consumption can contribute decisively in order to reduce the operating costs of aviation operators and in order to ensure the quick recovery and financial viability of the airline companies. Fuel typically comprises 25% or more of airline costs and accounts for over 97% of airline CO_2 emissions. Typically, the price for fossil-based aviation fuel would be €600 per tonne and as of September 2018 EU allowances for CO_2 were traded over €20 per tonne at EU Emissions Trading System (ETS) [3],[6]. From the above, it is clear that the minimization of fuel consumption in aircraft makes sense both for environmental and financial reasons.

Until now the fuel consumption reduction methods include the renewal of the existing aircraft fleet with newer lighter more fuel efficient aircraft, the development of new fuel efficient jet engines, carbon-neutral bio-fuels, hydrogen or electric propulsion or the reduction of aerodynamic drag [3],[5]. Other approaches include better air traffic management systems and attempts to estimate fuel consumption accurately with big data and machine learning techniques[7]. Furthermore, research about optimization of climb, cruise and descent trajectories has been attempted through optimal control techniques [8],[9],[10]. So far, researchers try to optimize each phase of the flight separately. For example, optimal cruise trajectories or optimal climb and descent fuel trajectories which could be used for Continuous Climb Operations (CCO) or Continuous Descent Operations (CDO) [11].

Our approach was to use multistage optimal control in order to model each of the flight stages and determine fuel optimal flight trajectories with numerical simulation and nonlinear optimization methods programmed in MUSCOD II [12]. To our knowledge this thesis is the only work that combines airplane dynamics with a

fuel flow model, a jet-engine model and a realistic atmosphere model. The second novelty of our work is that we treat and optimize the three flight phases together. The motivation behind our approach is that the end of the climb phase is the start of the cruise phase and the end of the cruise phase is the start of the descent phase. In other words, the optimal trajectory and the optimal controls for the next flight phase depend on initial conditions (initial velocity, climb angle and altitude, etc.) generated by the state and controls applied during the previous flight phase. As a result, the best approach is to optimize all the flight phases together, in order to minimize the consumed fuel not in the end of a single phase but in the end of the entire flight. A third advantage of our approach is that multistage direct optimal control provides a strong mathematical framework for the treatment of discontinuities on the model, which can be easily implemented with MUSCOD II in C++ code.

During this thesis we performed many different simulations and comparisons with minimum time and minimum fuel trajectories in order to determine the effects of important flight parameters, like cruise altitude and total flight duration. By thorough analysis of numerical experiments, we provide useful optimal control policy for the thrust lever position and angle of attack. Furthermore, we compare a simple thrust model with the realistic jet engine thrust and we provide useful insight for the physical phenomena dictating the behavior of fuel consumption during flight.

In the following chapter, we present thoroughly the direct optimal control algorithms implemented by MUSCOD II, which we use for our numerical experiments. At the third chapter, we present a state of the art aircraft model, the velocity dependent thrust specific fuel flow model, the international standard atmosphere model, and the jet engine thrust model described at BADA 4 [13]. Then, we develop minimum fuel and minimum time objective functions and we formulate the equality and inequality constraints for each flight phase. In the final chapter, we discuss the results of our simulations, we compare minimum time and minimum fuel trajectories and we provide valuable insight for the effects of cruising altitude and flight duration. We also discuss the difference between a simple thrust model and the realistic jet engine thrust model.

2 Multistage optimal control problem

Optimal control is a branch of mathematical optimization which tries to optimize the behavior of a dynamical system [15],[16]. For many real life problems we need to determine the optimal control strategy and optimal system parameters which are going to result in an optimal state trajectory. Such a solution should also respect physical limitations or design specifications called constraints. The solution of the optimal control problem is optimal in the sense that it optimizes a quantitative measure of the system's performance, called the objective function.

Modeling the dynamical system is called the choice of an adequate mathematical model (in the form of ordinary differential equations (ODE) or differential algebraic equations(DAE)), and the choice of suitable constraints. The determination of a proper objective function is also a design choice made by the engineer in order to accomplish a specific goal.

In many realistic problems it is not possible to describe all the existing dynamics, interactions and specifications of our problem with the same set of dynamical equations and constraints, for example due to state and right hand side discontinuities. In order to overcome this problem the full horizon of interest $[t_0, t_f]$ is divided to i phase intervals $[t_i,t_{i+1}]$ also called stages.

Each phase can vary with respect to:

- objective function (Mayer and Lagrange terms)
- dynamics and type of differential equations (ODE/DAE)
- different constraints
- different duration (fixed or free)
- number of states, number of controls, number of free parameters,. . . ,etc

In the next section we present the general form of such a multistage problem and present a numerical method to solve it.

2.1 General form of multistage optimal control problem

For this section we followed the notes for "Optimization with Differential Equations" by professor Hans Georg Bock[15], "The Theory of MUSCOD in a Nutshell" by Daniel B. Leineweber [20] and the MUSCOD-II manual [18].
The general form of multi stage optimal control problem can be expressed as:

$$\min_{x,u,p} \sum_{i=0}^{M-1} \left(\int_{t_i}^{t_{i+1}} L_i(t, x_i(t), u_i(t), p)dt + \Phi_i(t_{i+1}, x(t_{i+1}), p) \right), \tag{2.1}$$

$$t \in [t_i, t_{i+1}], \ i = 0, 1, \ldots, M-1$$

$$\textit{such that :} \qquad \frac{dx_i}{dt} = f_i(t, x_i(t), u_i(t), p) \tag{2.2}$$

$$x_{i+1}(t_{i+1}) = x_i(t_{i+1}) \tag{2.3}$$

$$g(t, x(t), u(t), p) \left\{ \begin{array}{c} = \\ \geq \end{array} \right\} 0 \in R_i^{n_g} \tag{2.4}$$

$$r(x(0), \ldots, x(t_f), p) \left\{ \begin{array}{c} = \\ \geq \end{array} \right\} 0 \in R_i^{n_r} \tag{2.5}$$

We assume that the time horizon of interest $[t_0, t_f]$ is divided into M model stages corresponding to the subintervals $[t_i, t_{i+1}]$, $i = 0, 1, \ldots, M-1$ and the duration $h_i := t_{i+1} - t_i$ of model stage i may be variable. On each model stage interval, the corresponding system state is described by the differential state vector $x_i(t) \in \mathbb{R}^{n_x^i}$. The system behavior is described by an ordinary differential equation of the form (2.2) and it is controlled by the control vectors $u_i(t) \in \mathbb{R}^{n_u^i}$ and the global design parameter vector $p \in \mathbb{R}^{n_p^i}$. The decoupled constraints are included at $g(\cdot)$ and the coupled constraints are included at $r(\cdot)$. In order to impose continuity during stage transition a "states continuity condition" (2.3) is included.

Usually, we assume that the state dimensions are constant between stages: $(n_x^{i+1} = n_x^i)$ and no "jumps" occur in the states. Though, special care has to be taken in case of "jumps" in the differential states or changes in the number of the differential states $(n_x^{i+1} \neq n_x^i)$. To solve this problem MUSCOD II implements a special type of model stage, called transition stage. During a transition stage the state $x_i(t_{i+1})$ is evaluated by a transition function $c_i(\cdot)$ which implements the "jump" or changes the state dimensions $x_{i+1}(t_{i+1}) = c_i(t_i, x_i(t_i), u_i(t_i), p)$. Usually, the "transition stage" duration is set to zero, i.e. $t_{j+1} = t_j$. Then the next stage after the "transition stage" is integrated normally following the continuity condition.

Objective function

The Objective function is a quantitative measure of the performance of the system under study[17],[15]. This objective could be time, energy, cost or any quantity or combination of quantities that can be represented by a single number. The equation (2.1) defines a Bolza-type objective function and it is a combination of:

- Lagrange objective function

$$J_L := \int_{t_0}^{t_f} L_i(t, x(t), u(t), p)dt$$

which is defined over all the time horizon from t_0 to t_f

- Mayer objective function

$$J_M := \Phi_i(t_f, x(t_f), p)$$

which is the terminal cost at final time t_f.
Those two types are equivalent by a mathematical point of view and can be transformed into each other.

Equality and inequality constraints

Special mention has to be given to the definition of different kinds of constraints which can be imposed for all variables, i.e. states, controls, parameters, and stage durations.

Equation (2.4) represents the decoupled constraints $g(\cdot)$ which can be specified at single points in time ($t = t_k$) or on complete model stages (i.e. $\forall t \in [t_k, t_{k+1}]$). These constraints can be path and interior point constraints in the form of equalities or inequalities. Examples of inequality constraints can be:

- Initial conditions

$$x(t_0) = x_0 \tag{2.6}$$

- Terminal conditions

$$r(t_f, x(t_f), p) = 0 \; or \; \geqslant 0$$

Inequality constraints often contain simple bounds on state and control profiles as well as parameters:

- Bounding box conditions

$$\underline{x} \leqslant x(t) \leqslant \overline{x} \tag{2.7}$$
$$\underline{u} \leqslant u(t) \leqslant \overline{u} \tag{2.8}$$
$$\underline{p} \leqslant p(t) \leqslant \overline{p} \tag{2.9}$$

• Interior point constraints can also be used for single times $t_k \in [t_i, t_{i+1}]$ with $k = 0, \ldots, l$

$$\tilde{g}((t_k), x(t_k), p) = 0 \; or \; \geq 0 \tag{2.10}$$

Coupled constraints $r(\cdot)$ (2.5) allows to couple different time points, for example:

• Periodicity

$$x(t_f) - x(t_0) = 0 \tag{2.11}$$

Assumptions

We assume that:

- the controls u are piecewise continuous with finitely many discontinuities
- the differential equations f are piecewise differentiable on subintervals and piecewise continuous
- the states x are continuous and piecewise differentiable.

Free stage duration and final time formulation

As we said earlier in this chapter the stage duration and the final time can be set free and be subject to optimization. In order to do so we can treat the start and end times of stages t_i and t_{i+1} with $i = 0, 1, \ldots, M-1$ as free parameters.

The original [OCP] Bolza-type objective function (2.1) and Ode of the form (2.2) can be transformed by introducing the normalized fixed time τ:

$$t =: t_i + (\tau - i)(t_{i+1} - t_i) = \varphi(\tau), \quad \tau \in [i, \; i+1] \qquad \left(\tau = i + \frac{t - t_i}{(t_{i+1} - t_i)}\right) \tag{2.12}$$

where the initial and final time $0 = t_0 < t_1 < \ldots < t_{M-2} < t_{M-1} = t_{final}$ of each stage is transformed to $0 = \tau_0 < \tau_1 < \ldots < \tau_{M-1} = M - 1$. For example, for a simple system with only one stage, if $\tau = 0$ then $t = t_s$ and if $\tau = 1$ then $t = t_f$

$$\frac{d}{d\tau}\varphi(\tau) \Rightarrow \frac{dt}{d\tau} = 0 + (t_{i+1} - t_i) \Rightarrow dt = (t_{i+1} - t_i)d\tau \tag{2.13}$$

Now the objective function and the state dynamics are transformed as follows:

$$J := \sum_{i=0}^{M-1} \left(\int_{\tau_i}^{\tau_{i+1}} L_i(x(\tau), u(\tau), p)(t_{i+1} - t_i)d\tau + \Phi(\tau_{i+1}, x(\tau_{i+1})) \right), \quad \tau \in [i,\ i+1] \tag{2.14}$$

$$\frac{dx}{d\tau} = f_i(\tau, x(\tau), u(\tau), p)(t_{i+1} - t_i), \quad \tau \in [i,\ i+1] \tag{2.15}$$

where the initial and final times t_i and t_{i+1} for each stage i are now transformed to additional parameters.

2.2 The direct method for OCP

In order to solve the optimal contol problem there are three different approaches:

- Dynamic Programming and Hamilton-Jacobi-Bellman equation
- Indirect Methods based on calculus of variations and Pontryagin Maximum Principle
- Direct Methods based on discretization of states and controls

The key problem we have to overcome in order to solve the Optimal Control Problem described at (2.1) to (2.5) is the infinite dimensionality of states $x(t)$ and controls $u(t)$. H.G.Bock and K.J.Plitt introduced a multiple shooting algorithm for direct solution of optimal control problems[19], which solves the optimal control problem directly in terms of control and state variables. Multiple shooting parameterization of the state differential equations is coupled with a simultaneous control parameterization. As a result the original continuous optimal control problem is reformulated as a finite non linear programming problem [NLP], which then can be solved iteratively with sequential quadratic programming [SQP]. According to Bock and Plitt [19] the algorithm is globally convergent, its local convergence is super-linear with an asymptotic convergence rate that is essentially independent of the meshsize used in the parameterization. Furthermore, one of the most important properties of the algorithm is that it is completely derivative-free, because the gradients are generated by an internal numerical differentiation scheme [IND]. In short, the basic idea of IND is to approximate the finite differences for the needed derivatives and the IVP with the same sequence of stepsizes. More about IND can be found at "The Theory of MUSCOD in a Nutshell" written by Daniel B. Leineweber [20].

Control discretization

In this section the discretization of controls is presented for a single stage in order to keep notation simple, but it can be very easily generalized for multiple stages. Discritization of Control is done by choosing a suitable mesh.

$$t_0 = \tau_0 < \tau_1 < \ldots < \tau_m = t_f$$

Each model stage i is divided into m multiple shooting intervals $I_j := [\tau_j, \tau_{j+1}]$, $j \in \{0, 1, \ldots, m-1\}$. The position and number of the mesh points can be explicitly specified according to the goals and features of a particular control problem. For example in this thesis we used an equidistant grid.

Then a piecewise approximation $\hat{u}$ of the control function u on this grid is defined by the basis functions $\varphi_j(t, q_j)$ with only a finite set of "local" control parameters $q_j \in \mathbb{R}^{k_u}$.

$$\hat{u}(t) := \varphi_j(t, q_j), \ t \in I_j, \ j = 0, 1, \ldots, m-1$$

Usually $\varphi_j(q_j)$ is piecewise constant ($k_u = 1$):

$$\varphi_j(t, q_j) = q_j$$

or linear ($k_u = 2$):

$$\varphi_j(t, q_j) = q_j^1 + \frac{t - t_j}{t_{j+1} - t_j}(q_j^2 - q_j^1), q_j = \begin{pmatrix} q_j^1 \\ q_j^2 \end{pmatrix}$$

If we use a piecewise constant or piecewise linear approximation of the control u, then controls are no longer continuous at intermediate points. We can restore it by adding continuity constraints:

$$\varphi_j(\tau_{j+1}, q_j) - \varphi_{j+1}(\tau_{j+1}, q_{j+1}) = 0, \ j = 0, \ldots, m-2.$$

Though, it should be noted that using discontinuous approximations is advantageous because it results in a sparse optimal control problem structure with respect to the controls, which can be exploited.

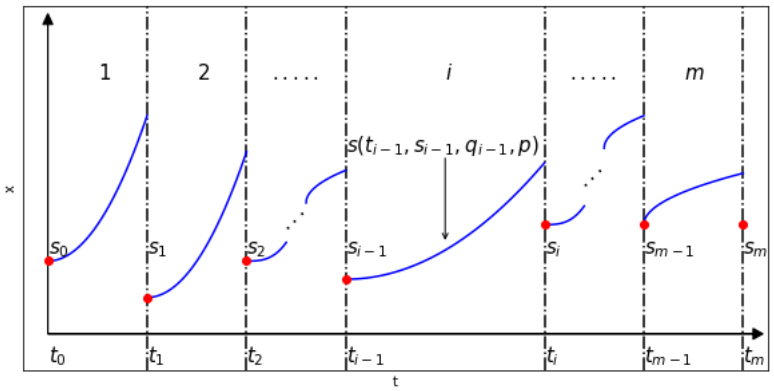

Figure 2.1: Multiple shooting state discretization

State discretization

The state $x(t)$ on every multiple shooting interval I_j, is simply replaced by the solution $x(t, s_j, q_j)$ of the initial value problem (IVP) as a function of the node value s_j and the control parameters q_j:

$$\dot{x} = f(t, x, \varphi(t, q_j), p) \tag{2.16}$$
$$x(\tau_j) = s_j, \quad t \in I_j \tag{2.17}$$

In order to ensure continuity of the state trajectory at the multiple shooting grid points we need to introduce m additional matching conditions c_j:

$$c_j(s_j, s_{j+1}, q_j) := x(\tau_{j+1}, s_j, q_j, p) - s_{j+1} = 0 \tag{2.18}$$
$$j = [0, 1, \ldots, m-1] \tag{2.19}$$

Discretization of bounds, interior point and path constraints

By using the discretization for controls and the state we can easily discretize the constraints. For example:

- the decoupled initial, end point and interior equality constraints:

$$s_0 = x_0, \; r(s_m, q_m) = 0 \tag{2.20}$$
$$r_j(s_j, q_j) = 0, \; j = 1, \ldots, m-2 \tag{2.21}$$

- the inequality constraints:

$$g_j(s_j, q_j) \geq 0, \; j = 0, \ldots, m-1 \tag{2.22}$$

- box constraints:

$$\begin{pmatrix} p - \underline{p} \\ \overline{p} - p \end{pmatrix} \geq 0 \tag{2.23}$$

Now we can rewrite all the above equality and inequality constraints as follows:

$$g(y) = \begin{pmatrix} g_0(\tau_0, s_0, q_0, p) \\ \vdots \\ g_j(\tau_j, s_j, q_j, p) \\ \vdots \\ g_m(\tau_m, s_m, q_m, p) \\ p - \underline{p} \\ \overline{p} - p \end{pmatrix} \geq 0 \in \mathbb{R}^{n_{ineq} + n_p} \tag{2.24}$$

$$h(y) = \begin{pmatrix} r(\tau_0, s_0, p) \\ \vdots \\ r(\tau_{j+1}, s_j, q_j, p) \\ \vdots \\ r(\tau_m, s_m, q_m, p) \\ x(\tau_0, s_0, q_0, p) - s_0 \\ \vdots \\ x(\tau_{j+1}, s_j, q_j, p) - s_{j+1} \\ \vdots \\ x(\tau_{m-1}, s_{m-1}, q_{m-1}, p) - s_m \end{pmatrix} = 0 \in \mathbb{R}^{m+n_{eq}} \tag{2.25}$$

All the new parameters are concatenated in a new parameter vector y for all intervals I_j. The new vector includes the control parameters q_j, the multiple shooting node values s_j, the phase times t_0, t_f and the original global parameters p.

$$s = \begin{pmatrix} s_0 \\ \vdots \\ s_m \end{pmatrix} \in \mathbb{R}^{(m+1)n_x}, q = \begin{pmatrix} q_0 \\ \vdots \\ q_{m-1} \end{pmatrix} \in \mathbb{R}^{n_u \cdot m \cdot k_u}, p = \begin{pmatrix} p_0 \\ \vdots \\ p_{n_p} \end{pmatrix} \in \mathbb{R}^{n_p} \tag{2.26}$$

$$y = \begin{pmatrix} s_0 \\ \vdots \\ s_m \\ q_0 \\ \vdots \\ q_{m-1} \\ t_0 \\ t_f \\ p_0 \\ \vdots \\ p_{n_p} \end{pmatrix} \in \mathbb{R}^{(m+1)n_x + n_u \cdot m \cdot k_u + n_p + n_{Ph}} \tag{2.27}$$

Now our discretized problem has $m \cdot n_x$ additional "matching conditions" c_j and $(m+1) \cdot n_x + n_u \cdot m \cdot k_u$ additional parameters for s_j and q_J, and n_{Ph} for the parameterized phase times.

The continuous optimal control problem is transformed to a constrained finite optimization problem with respect to the new parameter vector y. The discretized problem is a nonlinear programming problem [NLP] of large dimension:

$$\min_y \sum_{j=0}^{m-1} \int_{\tau_j}^{\tau_{j+1}} L_j(t, x(t, s_j, q_j), \varphi_j(t, q_j), p)dt + \Phi(\tau_m, s_m, p) =: \min_y F(y) \qquad (2.28)$$

such that :

$$g(y) \geq 0$$
$$h(y) = 0$$

2.3 Non linear programming theory

For the non linear programming theory we followed the classic book "Numerical Optimization" by Jorge Nocedal and Stephen J. Wright [17] and the notes for "Optimization with Differential Equations" by professor Hans Georg Bock [15].
The resulting problem (2.28) represents a large constrained optimization problem of the form:

$$\begin{aligned} & \min_x f(x) \qquad (2.29) \\ s.t. \quad & g(x) = 0,\ g : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}^{m_1} \\ & h(x) \geq 0,\ h : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}^{m_2} \end{aligned}$$

we assume $f,\ g,\ h \in \mathcal{C}^2(D)$ smooth and real-valued functions.

Definition 2.1 (Feasible set)
We define the feasible set S to be the set of points x that satisfy the constraints:

$$S := \{x | g(x) = 0,\ h(x) \geq 0\}$$

Definition 2.2 (Active set)
The active set $I(x)$ at any feasible x^* consists of the equality constraints and the inequality constraints for which $h_i(x^*) = 0$

$$I(x^*) = \{i | h_i(x^*) = 0\}$$

At a feasible point x^*, the inequality constraint i is said to be active if $h_i(x^*) = 0$ and inactive if the strict inequality $h_i(x^*) > 0$ is satisfied. Complementary inactive set:

$$I^{\perp}(x^*) = \{i | h_i(x^*) > 0\}$$

Theorem 2.1 (LICQ - linear independent constraint qualification)
Let x^* be feasible and the active set $I(x^*)$, the linear independence constraint qualification [LICQ] holds if the set of active constraint gradients $\{\nabla \tilde{g}_i(x), i \in I(x)\}$ is

linearly independent.

$$\tilde{h}(x) = \begin{pmatrix} h_1(x) \\ \vdots \\ h_s(x) \end{pmatrix}, \tilde{g} = \begin{pmatrix} g(x) \\ \vdots \\ \tilde{h}(x) \end{pmatrix}, rank\frac{\partial \tilde{g}}{\partial x} = m_1 + s \ (full\ rank) \tag{2.30}$$

Definition 2.3 (Tangent space)
Given a feasible point x^* and the active constraint set $I(x^*)$, the set of linearized feasible directions $T(x^*)$ is:

$$T(x^*) = \{p | \nabla_x^T g(x^*)p = 0, \nabla_x^T h_i(x^*)p = 0, i \in I(x^*)\} \quad = \{p | \nabla_x \tilde{g}(x^*)p = 0\} \tag{2.31}$$

Definition 2.4 (Lagrangian function)
We define the Lagrangian function:

$$\mathcal{L}(x, \lambda, \mu) = f(x) - \lambda^T g(x) - \mu^T h(x), \ \lambda \in \mathbb{R}^{m_1}, \ \mu \in \mathbb{R}^{m_2}$$

Theorem 2.2 (NOC1 - First order necessary conditions)
Suppose that x^* is a local minimum and regular (LICQ holds), the functions f, g, h are continuously differentiable. Then Lagrange multiplier vectors λ^*, μ^* exist, such that the following conditions are satisfied at KKT point(x^*, λ^*, μ^*)

- *Stationarity*
$$\nabla \mathcal{L}(x^*, \lambda^*, \mu^*) = 0 \tag{2.32}$$
- *Feasibility conditions:*
$$\begin{aligned} g(x^*) &= 0 \\ h(x^*) &\geq 0 \end{aligned} \tag{2.33}$$
- *Complementarity conditions:*
$$\mu^* \geq 0 \ and \ \mu^{*T} h(x^*) = 0 \tag{2.34}$$
or equivalently $\mu^* = 0$ if $h_i(x^*) > 0$ i.e. $\mu_i^* \cdot h_i(x^*) = 0 \ \forall \ i$, either constraint i is active $h_i = 0$ and $\mu_i = 0$ or $\mu_i > 0$ and $h_i(x^*)$ or possibly both.

NOC1 conditions are also known as Karush-Kuhn-Tucker Conditions [KKT].

Theorem 2.3 (Strict complementarity)
Given a local solution x^* of (2.29) and a vector μ^* satisfying the KKT conditions, the strict complementarity condition holds if exactly one of μ_i^* and $h_i(x^*)$ is zero $\forall i \in I$. In other words $h_i(x^*) = 0 \Leftrightarrow \mu_i^* > 0$.

Satisfaction of the strict complementarity property usually makes it easier for algorithms to determine the active set $I(x^*)$ and converge rapidly to the solution x^*. For a given problem (2.29) and solution point x^*, there may be many vectors $\lambda,\ \mu$ for which the KKT conditions are satisfied. When the LICQ holds, however, the optimal $\lambda^*,\ \mu^*$ is unique.

Definition 2.5 (Strictly active constraints)
The set of strictly active constraints is defined as:

$$I^+(x^*) := \{i | h_i(x) = 0 \text{ and } \mu_i^* = 0\} \subset I(x^*) \tag{2.35}$$

Definition 2.6 (Tangent space of strictly active constraints)
The tangent space of strictly active constraints is defined as:

$$T^+(x^*) := \{p | \nabla^T g(x^*)p = 0,\ \nabla h_i^T(x^*)p = 0,\ i \in I^+(x^*)\} \tag{2.36}$$
$$Note: I^+(x^*) \subset I(x^*) \Rightarrow T^+(x^*) \supset T(x^*)$$

Theorem 2.4 (NOC2 - Second-order necessary conditions)
Suppose that x^* is a local solution of (2.29) and that the LICQ condition is satisfied. Let $\lambda^*,\ \mu^*$ be the Lagrange multipliers for which the KKT conditions hold. Then:

$$p^T \nabla^2 \mathcal{L}_{xx}(x^*, \lambda^*, \mu^*)p \geq 0,\ \forall p \neq 0,\ p \in T(x^*) \tag{2.37}$$

i.e. The Hessian of the Lagrangian $\nabla^2\mathcal{L}$ is Positive Semi-Definite on $T(x^*)$

Theorem 2.5 (SOC - Second-Order Sufficient Conditions)
Suppose that x^* is a local solution of (2.29) and that the LICQ condition holds. Let $\lambda^*,\ \mu^*$ be the Lagrange multipliers for which the KKT conditions are satisfied and additionally the Hessian of $\mathcal{L}$ is positive definite:

$$p^T \nabla^2 \mathcal{L}(x^*, \lambda^*, \mu^*)p > 0,\ \forall p \neq 0,\ p \in T^+(x^*) \tag{2.38}$$

Then x^* is a strict local minimum.
i.e. The Hessian of the Lagrangian $\nabla^2\mathcal{L}$ is positive definite on $T^+(x^*)$

2.4 Newton's method

Now that we transformed our OCP to a general NLP we can use constrained optimization techniques to solve it. By the first order necessary optimality conditions [KKT conditions] if x^* is a local solution of (2.29) and λ^*, μ^* are the corresponding Lagrange multipliers for $\mathcal{L}(x, \lambda, \mu) = f(x) - \lambda^T g(x) - \mu^T h(x)$. By assuming active constraints $\tilde{h}(x)$ are known we can use Newton's method to iteratively solve the KKT conditions:

$$F(x^*, \lambda^*, \mu^*) = \begin{pmatrix} \nabla \mathcal{L}(x^*, \lambda^*, \mu^*) \\ g(x^*) \\ \tilde{h}(x^*) \end{pmatrix} = 0 \tag{2.39}$$

Now by a simple Taylor expansion of $F(\cdot)$ we derive the Newton iteration:

$$\begin{aligned} x_{k+1} &= x_k - \nabla F^{-1}(x_k) \cdot F(x_k) \\ \lambda_{k+1} &= \lambda_k + \Delta\lambda_k \\ \mu_{k+1} &= \mu_k + \Delta\mu_k \end{aligned} \tag{2.40}$$

At each iteration for k we solve the system:

$$\begin{bmatrix} \nabla^2 \mathcal{L}(x_k, \lambda_k, \mu_k) & -\nabla g(x_k) & -\nabla \tilde{h}(x_k) \\ \nabla^T g(x_k) & 0 & 0 \\ \nabla^T \tilde{h}(x_k) & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta\lambda_k \\ \Delta\mu_k \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) - \nabla g(x_k)\lambda_k - \nabla \tilde{h}(x_k)\mu_k \\ g(x_k) \\ \tilde{h}(x_k) \end{bmatrix}$$

which can be rewritten as:

$$\underbrace{\begin{bmatrix} \nabla^2 \mathcal{L}(x_k, \lambda_k, \mu_k) & -\nabla g(x_k) & -\nabla \tilde{h}(x_k) \\ \nabla^T g(x_k) & 0 & 0 \\ \nabla^T \tilde{h}(x_k) & 0 & 0 \end{bmatrix}}_{\textbf{KKT Matrix}} \begin{bmatrix} \Delta x_k \\ \lambda_k + \Delta\lambda_k \\ \mu_k + \Delta\mu_k \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) \\ g(x_k) \\ \tilde{h}(x_k) \end{bmatrix} \tag{2.41}$$

Which is equivalent to solving a Quadratic Problem:

$$\begin{aligned} \textbf{[QP]} \qquad \min_{\Delta x} \; & \frac{1}{2} \Delta x_k^T B_k \Delta x_k + \nabla f(x_k) \Delta x_k^T \\ s.t. \quad & g(x_k) + \nabla g(x_k) \Delta x_k^T = 0 \\ & \tilde{h}(x_k) + \nabla \tilde{h}(x_k) \Delta x_k^T = 0 \end{aligned} \tag{2.42}$$

Line search and Wolfe conditions

Line search methods compute at each iteration a search direction p_k and then decides how far to move along that direction:

$$x_{k+1} = x_k + \alpha_k \cdot p_k$$

Usually p_k has to be a descent direction such that: $p_k^T \nabla f_k < 0$ because the objective function should be reduced at each iterate.

$$p_k = -B_k^{-1} \nabla f_k$$

with B_k symmetric and positive definite matrix. Now we can see that if B_k is the exact Hessian $\nabla^2 f(x_k)$ then we have the Newton method and Quasi-Newton method if B_k is updated by an approximation.

During the line search iterations we have to make sure that f deceases sufficiently after each step. Unfortunately this is not adequate because if the step is chosen to be too small the algorithm will not do any meaningful progress. These two issues are tackled by the following conditions introduced by Wolfe:

• **Wolfe conditions**

1. **Sufficient decrease - Armijo condition**
   The step size α_k should be chosen to give sufficient decrease in the objective function:
   $$f(x_k + \alpha_k p_k) \leqslant f(x_k) + c_1 \cdot \alpha_k \cdot \nabla f_k^T \cdot p_k, \tag{2.43}$$

2. **Curvature condition**
   To rule out unacceptably short steps α_k should satisfy:
   $$\nabla f^T (x_k + \alpha_k p_k) \cdot p_k \geq c_2 \cdot \nabla f_k^k \cdot p_k \tag{2.44}$$
   with $0 < c_1 < c_2 < 1$

Now if we modify the curvature condition to force α_k to lie in at least a broad neighborhood of a local minimizer or stationary point by excluding points that are far from stationary points we get the strong Wolfe conditions:

• **Strong Wolfe conditions**
The strong Wolfe conditions require α_k to satisfy:

$$\begin{aligned} f(x_k, \alpha_k p_k) \leq & f(x_k) + c_1 \alpha_k \nabla f_k^T p_k, \\ | \nabla f^T (x_k + \alpha_k p_k) p_k | \leq & | \nabla f_k^T p_k | \\ with\ 0 < c_1 < & c_2 < 1 \end{aligned} \tag{2.45}$$

The BFGS method

Computing the exact Hessian matrix at each iterate has a big computational cost and requires lots of memory to store. For this reason, the idea of quasi Newton methods was developed. The fundamental idea of quasi-Newton updating is to compute an inexact Hessian update B_k by combining the most recently observed information about the objective function with the existing knowledge embedded in the current Hessian approximation. The BFGS method is presently considered to be the most effective of all quasi-Newton updating formulae. BFGS instead of calculating a Hessian approximation it approximates the inverse H_k. The updated approximation of the inverse H_{k+1} must be symmetric and positive definite, and must satisfy the secant equation $H_{k+1}y_k = s_k$.

• Secant equation
Suppose for a new iterate x_{k+1} and a new quadratic model $m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T \cdot p + \frac{1}{2}p^T B_{k+1}p$. The gradient of m_{k+1} should match gradient of f at x_k, x_{k+1}

$$\begin{aligned} \nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - \alpha_k B_{k+1}p_k = \nabla f_k \Rightarrow \\ B_{k+1}\alpha_k p_k = \nabla f_{k+1} - \nabla f_k \Rightarrow \\ s_k = x_{k+1} - x_k = \alpha_k p_k,\ y_k = \nabla f_{k+1} - \nabla f_k \\ B_{k+1}s_k = y_k \end{aligned} \tag{2.46}$$

• Curvature condition
Given the displacement s_k and the change of gradients y_k the symmetric positive definite matrix B_{k+1} maps s_k into y_k if curvature condition holds:

$$s_k^T y_k > 0$$

If f is convex, curvature conditions holds $\forall\ x_k, x_{k+1}$. For non convex functions we need to enforce curvature condition by line search choosing step length α_k if we impose Wolfe or strong Wolfe conditions. When curvature condition holds then Secant equation always has a solution.

• BFGS algorithm
Specify starting point x_0, inverse Hessian approximation H_0 and convergence criterion.
Until convergence repeat:

1. Compute search direction $p_k = -H_k \nabla f_k$
2. Determine α_k by line search to satisfy Wolfe conditions
3. Compute $s_k = x_{k+1} - x_k$ and $y_k = \nabla f_{k+1} - \nabla f_k$
4. Update H_{k+1} by the BFGS formula for the inverse Hesian:

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$$

with $\rho_k = \dfrac{1}{y^T \cdot s_k}$

5. $k = k + 1$

The BFGS update for the Hessian can be written as:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \tag{2.47}$$

• Damped BFGS update - Powell modification
In order to ensure that the BFGS update is always well defined we can use the Damped BFGS update:
Given a symmetric and positive definite matrix B_k we compute:

$$r_k = \theta_k y_k + (1 - \theta_k) B_k s_k$$

where θ_k is defined as:

$$\theta_k = \begin{cases} 1, & \text{if} s_k^T y_k \geq 0.2 s_k^T B_k s_k \\ (0.8 s_k^T B_k s_k)/(s_k^T B_k s_k - s_k^T y_k), & \text{if} s_k^T y_k < 0.2 s_k^T B_k s_k \end{cases} \tag{2.48}$$

The Powell modification practically interpolates between the current approximation B_k and the one produced by unmodified BFGS formula. By modifying the definition of y_k we guarantee that B_{k+1} is positive definite.

With BFGS each iteration can be performed at a cost of $O(n^2)$ (plus the cost of function and gradient evaluations) instead of $O(n^3)$ operations such as linear system solvers or matrix–matrix operations. Therefore, the use of quasi-Newton updates not only eliminates the need to calculate second derivatives, but also considerably reduces the amount of work necessary to calculate the correction s_k. Also, the algorithm is locally superlinearly convergent. This is in general not the case for BFGS with Powell modification. Moreover, the convergence can be globalized by including a line search or trust region method. Finally, Keeping the matrices B_k positive definite, even when the true Hessian matrix is not, has the great practical advantage that the resulting QP subproblems are convex and hence easier to solve. More information can be found in "Numerical Optimization" by Jorge Nocedal and Stephen J. Wright [17].

2.5 Condensing algorithm

As we can see at (2.28) the Mayer and Lagrange terms of the objective function are separable because each F_j depends only at a single shooting interval.

$$F(s,q,p) = \sum_{j=0}^{m-1} \tilde{F}_j(s_j, q_j, p) \tag{2.49}$$

Also the discretized variables s_j, q_j affect the objective function locally, only at each I_j. The same happens for the solution solution $x(t, s_j, q_j)$ and the constraints $g_j(\tau_j, s_j, q_j, p)$, $r(\tau_{j+1}, s_j, q_j, p)$ with the exception of the matching conditions $c_j = x(\tau_{j+1}, s_j, q_j, p) - s_{j+1} = 0$ which are linearly coupled.
Also the Hessian of $\mathcal{L}$ for problem (2.28) has a sparse block diagonal structure which can be exploited:

$$\nabla^2_{yy}\mathcal{L} = \begin{bmatrix} C_0 & & \\ & \ddots & \\ & & C_m \end{bmatrix}, C_m = \frac{\partial^2 \mathcal{L}(y)}{\partial s_m^2}, C_j = \frac{\partial^2 \mathcal{L}(y)}{\partial (s_j, q_j)^2},\ j = 0, \ldots, m-1 \tag{2.50}$$

The basic idea of condensing algorithm [19] is that we at first condense the size of the problem and then we recover the full solution. Instead of calculating the inverse of the KKT matrix by (2.32) at each step, we are going to transform it in a condensed QP and solve the a problem with greatly reduced size instead. The first step is to linearize the matching conditions:

$$\Delta s_{j+1} = G_j^s \Delta s_j + G_j^q \Delta q_j + h_j \tag{2.51}$$
$$G_j^s = \frac{\partial x(\tau_{j+1}, s_j, q_j)}{\partial s_j},\ G_j^q = \frac{\partial x(\tau_{j+1}, s_j, q_j)}{\partial q_j}$$

The linearized matching conditions can be writen in explicit form which enables us to compute the whole state direction.

1. Eliminate Δs_{k+1} from the original QP by applying Gauss elimination recursively for Δs_j, $j = 1, \ldots, m-1$ to get condensed QP of the form:

$$\frac{1}{2} \begin{pmatrix} \Delta s_0^k \\ \Delta q^k \end{pmatrix}^T \hat{B}^k \begin{pmatrix} \Delta s_0^k \\ \Delta q^k \end{pmatrix} + \hat{b} \begin{pmatrix} \Delta s_0^k \\ \Delta q^k \end{pmatrix} \tag{2.52}$$
$$v_1 + E_1^s \Delta s_0 + E_1^q \Delta q = 0 \tag{2.53}$$
$$v_2 + E_2^s \Delta s_0 + E_2^q \Delta q \geq 0 \tag{2.54}$$

   The condensed problem becomes the same size as of single shooting.

2. Solve the condensed QP with an efficient and stable QP solver for (Δ_0^k,Δq^k)
3. Do a forward recursion to get Δs^k and update s^{k+1} and q^{k+1}
4. Do a backward recursion to compute the Lagrange multipliers for original QP by the condensed QP Lagrange modifiers.

2.6 Active set strategy

An active set strategy [ASS] is used in order to identify the active set $I(x^*)$ during the SQP iterations. At each SQP iteration k the algorithm checks systematically a candidate set $I_k = \{i_1, \ldots, i_k\}$ if $\mu_{ij} \geq 0$.
Our original QP with equality and inequality constraints:

$$\begin{aligned} min & \frac{1}{2} d^T B_k d + g^T d \\ s.t. & \ Ad + a = 0, \ A \in \mathbb{R}^{l \times n} \\ & Cd + c \geq 0, \ C \in \mathbb{R}^{m \times n} \end{aligned}$$

Active set method

1. Start with a feasible point d_0, and first candidate set I_0
2. Solve the equality constraint QP for candidate set I_k and get $(\hat{d}, \hat{\lambda}, \mu)$

$$\begin{aligned} min & \frac{1}{2} d^T B_k d + g^T d \\ s.t. & \ Ad + a = 0 \\ & C_j d + c_j = 0, \ i \in I_k \end{aligned}$$

3. if $\hat{d}$ is a feasible solution then:
   - if all $\mu_{ij} \geq 0$ then the solution and the active set is found!
   - if at least one $\mu_k < 0$ then drop the k constraint by the working set $I_{k+1} = I_k \ \{k\}$ and solve the problem again.

In practice when we are far from the solution, the SQP approach is usually able to improve the estimate of the active set and guides the iterates towards a solution. It is important to make sure (with appropriate updates) that the hessian B_k is positive definite. Then the QP subproblem is Convex and it can be solved even with a wrong active set. It can be shown that ASS algorithm for strictly convex QPs, converges in a finite number of iterations (at most $n - l$ steps)[17].

2.7 Sequential quadratic programming

The sequential quadratic programming [17] is an iterative method in which the NLP is approximated by quadratic optimization subproblems on each step[17],[15]. The solution of the QP subproblem at each SQP iterate determines the new direction of search for the optimum. As we have already discussed in the previous paragraph in the direct multiple shooting the QP has a sparse structure which can be exploited by a condensing algorithm.

As we already mentioned in a previous chapter, If we subsume all multiple shooting variables (s_0, s_j, q_j, p) to a single vector y of large dimension n, we can write the objective function as $f(y) : \mathbb{R}^n \to \mathbb{R}$. Similarly, we can subsume all equality constraints to a vector $g(y)$ and the inequality constraints in a vector $h(y)$. Then, the parameterized optimal control problem can be written as a finite dimensional nonlinear program.

$$\begin{aligned} \textbf{[NLP]} \quad & \min_y f(y) \\ s.t. \ & g(y) = 0 \\ & h(y) \geq 0 \end{aligned} \tag{2.55}$$

Simplified SQP algorithm

1. Evaluate functions and gradients at initial point:

$$f(y_0), h(y_0), g(y_0), \Delta f(y_0), \Delta h^T(y_0), \Delta g^T(y_0)$$

2. Choose initial Hessian approximation B_0 by a suitable symmetric positive definite matrix as initial estimate for Hessian

3. Solve the current QP subproblem (apply Condensing modification). Employ active set strategy to update a candidate set of active constraints I_k.

$$\begin{aligned} & min \frac{1}{2} p^T B_k p + \nabla f(x_k)^T p \\ s.t. \ & \nabla g^T(x_k) p + g(x_k) = 0 \\ & \nabla h_i^T(x_k) p + h_i(x_k) = 0, \ i \in I_k \end{aligned}$$

   Calculate a KKT point $(p_k, \tilde{\lambda}_k, \tilde{\mu}_k)$

4. Check termination criterion. If convergence is achieved then y_k is the solution and I_k is the Active Set.

5. Update the estimate of the solution:

$$y_{k+1} = y_k + \alpha_k \cdot p_k$$

   Perform line search to determine step length $0 < \alpha_k < 1$ and set:

$$\lambda_{k+1} = \tilde{\lambda}_k$$

$$\mu_{k+1} = \tilde{\mu}_k$$

6. Calculate the gradient of the Lagrangian at old point. $\nabla_y \mathcal{L}(y_k, \lambda_{k+1}, \mu_{k+1})$ from gradients $\nabla f(y_k)$, $\nabla h^T(y_k)$, $\nabla g^T(y_k)$ by the currently available multiplier estimates λ_{k+1}, μ_{k+1}.

7. Evaluate functions and gradients at new point

$$f(y_{k+1}), h(y_{k+1}), g(y_{k+1}), \Delta f(y_{k+1}), \Delta h(y_{k+1}), \Delta g(y_{k+1})$$

8. Calculate the new gradient of Lagrangian function:$\nabla_y \mathcal{L}(y_{k+1}, \lambda_{k+1}, \mu_{k+1})$ from gradients $\nabla f(y_{k+1})$, $\nabla h^T(y_{k+1})$, $\nabla g^T(y_{k+1})$

9. Revise Hessian approximation. Calculate a symmetric positive definite B_{k+1} by a suitable update procedure (like BFGS).

$$B_{k+1} = B_k + U(B_k, \delta_k, \gamma_k)$$

   using $\delta_k = y_{k+1} - y_k$, $\gamma_k = \nabla_y \mathcal{L}(y_{k+1}, \lambda_{k+1}, \mu_{k+1}) - \nabla_y \mathcal{L}(y_k, \lambda_{k+1}, \mu_{k+1})$

10. Start a new iteration $k = k + 1$ and go to step 3.

More about the exact implementation of SQP methods at MUSCOD II can be found at "The Theory of MUSCOD in a Nutshell" by Daniel B. Leineweber [20].

3 Modelling and optimal control problem formulation

Modeling of airplane flight in order to study fuel consumption, is a very important step because we should describe adequately the aircraft dynamics and the atmospheric properties, as well as the the dynamics of jet engine thrust and fuel consumption. In the same time, we have to take special care in order to model each of the climb, cruise and descent flight phases, which differ on both the differential equations and the constraints. On the other hand, we have to make simplifying assumptions in order to make our problem solvable by the computer in a reasonable time:

3.1 Assumptions

1. **Point-mass aircraft model** In our experiments we used a simple, point-mass aircraft model based on the translational dynamics. The model ignores the rotational dynamics of the aircraft which occur due to moment of inertia and torques relative to an axis of rotation. To keep our model simple and with less state variables we can neglect rotational dynamics because in practice they are much faster than the translational dynamics.

2. **Flat, non-rotating Earth** In our model we neglect Earth's curvature and rotation to simplify the mathematical calculations. We assume that the airplane moves in a 3-D Cartesian space where x, y axis define the surface of the earth and z axis defines the altitude.

3. **No uncertainties and external disturbances** For the purpose of trajectory optimization of airplanes with respect to fuel consumption we do not need to take into account modeling error's or disturbances such as wind or turbulence.

4. **International Standard Atmosphere model** In order to study the aircraft fuel consumption we have to take into account the atmospheric properties for pressure, temperature and air density as a function of altitude. We used the international standard atmosphere model (ISA) [13] to model troposphere and stratosphere, as described at the base of aircraft data family 4 documents.

3.2 3D aircraft model

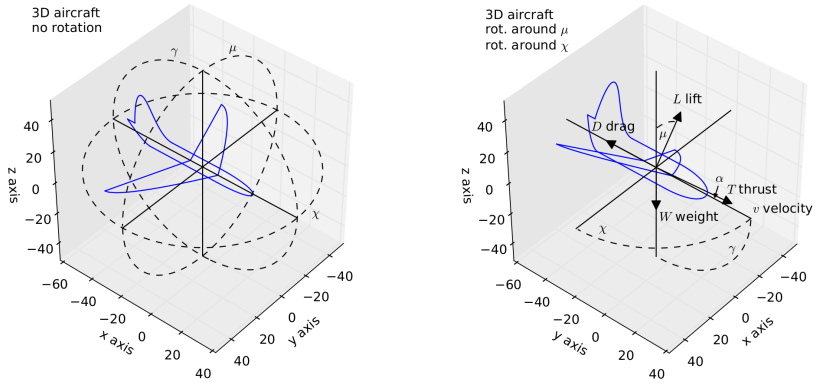

Figure 3.1: Visualization of states and forces acting on aircraft [Schubert, 2017][21]

In order to minimize the fuel consumption of an aircraft we first need to derive a dynamical model for the aircraft movement. The three-degrees-of-freedom point-mass aircraft model corresponds to the standards of the Base of Aircraft Data Family 4 (BADA 4) [13]. The derivation of the model is described at the master thesis of Markus Schubert [21] and Valeriy Semenov [22]. The translational equations of motion for a single point-mass aircraft are defined as follows:

$$\dot{x}(t) = v(t)cos\chi(t)cos\gamma(t), \tag{3.1}$$
$$\dot{y}(t) = v(t)sin\chi(t)cos\gamma(t), \tag{3.2}$$
$$\dot{z}(t) = v(t)sin\gamma(t), \tag{3.3}$$
$$\dot{v}(t) = \frac{1}{m(t)}(Thrust - D(v(t), \alpha(t))) - gsin\gamma(t), \tag{3.4}$$
$$\dot{\chi}(t) = \frac{1}{m(t)v(t)cos\gamma(t)}L(v(t), \alpha(t))sin\mu(t), \tag{3.5}$$
$$\dot{\gamma}(t) = \frac{1}{m(t)v(t)}(L(v(t), \alpha(t))cos\mu(t) - m(t)gcos\gamma(t)), \tag{3.6}$$
$$\dot{m}(t) = -f_r \cdot Thrust. \tag{3.7}$$

The 3D point mass aircraft is described by seven states and three controls and represents the aircraft's translational dynamics. The seven states demonstrated at figure(3.1) are the x, y position, the altitude z, the aircraft velocity v, the course angle χ, the climb angle γ and the aircraft mass m. The three controls are the angle of attack, the bank angle μ and the thrust lever position δ. The angle of attack α is the angle between the wings and the oncoming air. The angle of attack is controlled by the flap on the wings and its role is to balance the aerodynamic forces and moments around the aircraft's center of mass. The bank angle μ is controlled by the

ailerons and its role is to modulate the direction of lift vector with respect to the wind velocity vector at a fixed angle of attack. In our model the thrust is controlled by the thrust lever position δ. Thrust lever position has a value between 0 and 1, which represents zero and maximum thrust produced by the jet engines.

3.3 2D aircraft model

Most flights climb, cruise and descent in a more or less constant altitude without much turning. Thus, it is wise to simplify our dynamical model by removing the effect of Bank angle μ which controls the course angle χ and the y part of the aircraft position. The two-degrees-of-freedom point-mass aircraft model can be easily derived by setting $\mu = 0 \Rightarrow (\chi = 0,\ y = 0)$

$$\dot{x}(t) = v(t)cos\gamma(t), \tag{3.8}$$

$$\dot{z}(t) = v(t)sin\gamma(t), \tag{3.9}$$

$$\dot{v}(t) = \frac{1}{m(t)}(Thrust - D(v(t), \alpha(t))) - gsin\gamma(t), \tag{3.10}$$

$$\dot{\gamma}(t) = \frac{1}{m(t)v(t)}(L(v(t), \alpha(t)) - m(t)gcos\gamma(t)), \tag{3.11}$$

$$\dot{m}(t) = -f_r \cdot Thrust. \tag{3.12}$$

Now our new simplified model has only 5 states (x, z, v, γ, m) and only 2 controls: angle of attack α and thrust lever position δ, which makes our reduced problem numerically much simpler than the original.

Forces acting on the aircraft

In order to better understand the airplane trajectories described by the above dynamical equations we discuss the different forces acting on the aircraft shown at figure (3.1). Along the z axis the aerodynamic lift produced by the wings counteracts the weight W which points always to the ground. The lift depends on the aircraft velocity and on the angle of attack α. If we increase the angle of attack then more lift is produced and the airplane climbs with a positive climb angle γ. If we reduce the angle of attack then the climb angle becomes negative and our airplane descents. The same effect has the acceleration and deceleration by controlling the jet engine's thrust, while we keep the angle of attack constant because increase or decrease in speed results in an increase or decrease of Lift. Thus in order to change the lift we have to change the velocity or the angle of attack or both simultaneously.

Along the y axis the thrust produced by the jet engines has to counteract the aerodynamic drag which opposes the movement of the aircraft. In order to cruise

with a constant speed we need to keep the thrust equal to the Drag by controlling the thrust lever position δ. If the engines produce more thrust than the Drag then our airplane accelerates while if the thrust becomes smaller than the Drag then the airplane decelerates. In order to turn the aircraft we need to modulate the Lift through the bank angle μ. If the bank angle is not zero then the lift has a component along the x axis which forces the aircraft to turn. A positive bank angle results in a turn to the right and a negative bank angle results in a turn to the left.

The forces acting on the aircraft are modeled in accordance with Base of Aircraft Data Family 3 (BADA 3) [14].

$$L(v, \alpha) = \frac{\rho}{2} S(v)^2 C_L(\alpha), \tag{3.13}$$

$$D(v, \alpha) = \frac{\rho}{2} S(v)^2 C_D(\alpha) \tag{3.14}$$

where the lift and drag coefficients are, correspondingly:

$$C_L(\alpha) = C_{L,0} + C_{L,\alpha}\alpha, \tag{3.15}$$

$$C_D(\alpha) = C_{D,0} + C_{D,2} C_{L,\alpha}^2 \tag{3.16}$$

The weight is defined as:

$$W = m \cdot g \tag{3.17}$$

In simple models the Thrust is just a function of δ with a constant T_{max}

$$Thrust(\delta) = T_{max} \cdot \delta \tag{3.18}$$

on more realistic models the Thrust is a function of Altitude:

$$Thrust(H_p, \delta) = T_{max}(H_p) \cdot \delta \tag{3.19}$$

In addition we have to define the load factor $\xi(v, \alpha, m)$:

$$\xi(v, \alpha, m) := \frac{Lift(v, \alpha)}{m \cdot g} \tag{3.20}$$

The load factor should always be close to one in order to avoid sharp jerks and ensure a comfortable flight for the passengers.

$$0.8 \leq \xi(v, \alpha, m) \leq 1.2 \tag{3.21}$$

Parameters for the simplified model

gravitational acceleration:	g	=	$9.81\frac{m}{s^2}$			
air density:	ρ	=	$1.225\frac{m}{s^2}(Constant)$			
wing reference area:	S	=	$122.6m^2$			
maximum thrust:	T_{max}	=	$142310N(Constant)$			
fuel flow:	f_r	=	$0.000132444\frac{Kg}{N \cdot s}(Constant)$			
lift coefficients:	$C_{L,0}$	=	0.1 ,	$C_{L,a}$	=	0.2567
drag coefficients:	$C_{D,0}$	=	0.038,	$C_{D,2}$	=	0.0419

3.4 Atmosphere model (standard atmosphere)

In order to study the aircraft fuel consumption we have to take into account the atmospheric properties for pressure, temperature and air density as a function of altitude. For the purpose of numerical experimentation, we approximate the real atmosphere with an approximate model known as the international standard atmosphere (ISA)[13], created by ISO. The model assumes that:

- the air is devoid of dust, moisture or water vapor
- the air is without winds or turbulence
- the pressure and temperature change only with altitude

According to the standard model Earth's atmosphere is divided into two layers the Troposphere and the Stratosphere. The Tropopause is the separation between the two layers and its altitude is constant when expressed in terms of geopotential pressure altitude $H_{p,trop} = 11000\ [m]$. Below we provide expressions for the atmospheric properties at ISA as a function of altitude for each atmospheric layer according to BADA 4 [13]

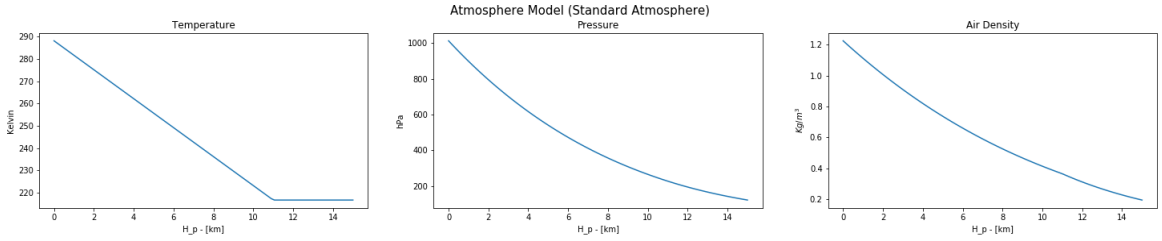

Figure 3.2: Atmosphere density ρ is not constant but a function of the altitude

1. **Troposphere** ($H_p =< 11km$)
   The Troposphere runs from the surface of the Earth to 11,000 meters. In the

Troposphere, the temperature decreases linearly and the pressure decreases exponentially. The rate of temperature decrease $\beta_{T,<}$ is called the lapse rate.

- Temperature:
$$T_<(H_p) = T_0 + \beta_{T,<} \cdot H_{p<} \tag{3.22}$$
- Pressure:
$$P_<(H_p) = P_0 \left(\frac{T_<}{T_0} \right)^{-\left(\frac{g_0}{\beta_{T,<} R}\right)} \tag{3.23}$$

2. **Stratosphere** ($H_p > 11km$)
The lower Stratosphere runs from 11,000 meters to 25,000 meters. In the lower Stratosphere the temperature is constant and the pressure decreases exponentially.

- Temperature:
$$T_>(H_p) = T_{trop} = 216,65 \qquad Kelvin \tag{3.24}$$
- Pressure:
$$P_>(H_p) = P_{trop} \cdot exp\left[-\frac{g_0}{(R \cdot T_{ISA,trop})} (H_{p,>} - H_{P,trop}) \right] \tag{3.25}$$

3. **Air density**
In each layer air density, ρ, in kg/m^3, is calculated from the pressure P and the temperature T at altitude H_p by using the perfect gas law:
$$\rho(H_p) = \frac{P(H_p)}{R \cdot T(H_p)} \tag{3.26}$$
The air density is decreasing as the altitude is increasing.

Note that the subindex $<$ denotes values below and at the Tropopause and subindex $>$ denotes values above the Tropopause.

Standard atmosphere model parameters (BADA 4)

The standard atmosphere model above uses the mean sea level (MSL) Standard atmosphere conditions. Those occur in the international standard atmosphere (ISA) at the point where the geopotential pressure altitude H_p is zero. They are denoted

as T_0, P_0, ρ_0 with the values listed below:

Standard atmospheric temperature at MSL:	$T_0 =$	$288.15 Kelvin$
Standard atmospheric pressure at MSL:	$P_0 =$	$101325 Pa$
ISA temperature gradient below Tropopause:	$\beta_{T,<} =$	$-0.0065 \frac{Kelvin}{m}$
Gravitational acceleration:	$g_0 =$	$9.8 \frac{m}{s^2}$
Real gas constant for air:	$R =$	$287.05287 \frac{m^2}{K \cdot s^2}$

3.5 Dynamics of fuel consumption

The dynamics of fuel consumption is determined by the thrust specific fuel consumption η and by the fuel flow f_{nom}. The thrust specific fuel consumption determines the mass of fuel consumed per second in order to produce one Newton of thrust. According to base of aircraft data the thrust specific fuel consumption of a Jet engine depends on aircraft speed in a more or less proportional way. The nominal fuel flow f_{nom} represents the mass of fuel consumed per second in nominal conditions. In our case the nominal fuel flow is equivalent to the rate of change of the aircraft mass $\dot{m} = -f_{nom}$

- **Thrust specific fuel consumption** $\eta(\upsilon)$
  For the Jet engines, the thrust specific fuel consumption, η $[Kg/(min \cdot KN)]$ is specified as a function of the speed υ:

$$\eta \;=\; C_{f1} \cdot \left(1 + \frac{\upsilon}{C_{f2}}\right) \tag{3.27}$$

- **Nominal fuel flow** f_{nom}
  The nominal fuel flow, f_{nom} $[Kg/min]$, can then be calculated using the thrust:

$$f_{nom} \;=\; \eta(\upsilon) \cdot Thrust(H_p) \tag{3.28}$$

- **Cruise fuel flow** f_{cr}
  An airplane cruises for the majority of a flight. For this reason, jet engines are designed to work at maximum efficiency while cruising. To represent this, base of aircraft data includes the cruise fuel flow correction factor C_{fcr}.

$$f_{cr} \;=\; C_{fcr} \cdot \eta(\upsilon) \cdot Thrust(H_p) \tag{3.29}$$

  Hence, the calculation of the fuel flow during climb and descent phases should be based at the nominal fuel flow f_{nom}, but the fuel flow during cruise is determined by f_{cr}.

If we take into account the definition for fuel flow the dynamical equation for $\dot{m}$ can be written as a function of altitude H_p, velocity v and thrust lever position δ

- during the Climb and Descent phase:

$$\begin{aligned} \dot{m} &= -f_{nom} && (3.30)\\ &= -\eta(v) \cdot Thrust(H_p, \delta) && (3.31)\\ &= -\eta(v) \cdot T_{max}(H_p) \cdot \delta && (3.32) \end{aligned}$$

- during the Cruise phase:

$$\dot{m} = -C_{fcr} \cdot \eta(v) \cdot T_{max}(H_p) \cdot \delta \quad (3.33)$$

3.6 Thrust model for Jet engines as a function of altitude

Simple dynamical models used for numerical simulations in aircraft often use a constant maximum thrust T_{max}. In this thesis we adopt a more realistic thrust model, which also takes into account the effect of altitude during flight. As we have already discussed fuel consumption depends on thrust through the nominal fuel flow ($f_{nom} = \eta(v) \cdot Thrust(H_p)$). The effect of altitude on Jet engine thrust can be understood if we recall that when the altitude increases, air density decreases. With lower air density, air molecules are further apart from each other and air flow through the engine decreases. As a result this leads to decrease in the thrust force produced by the Jet engines.

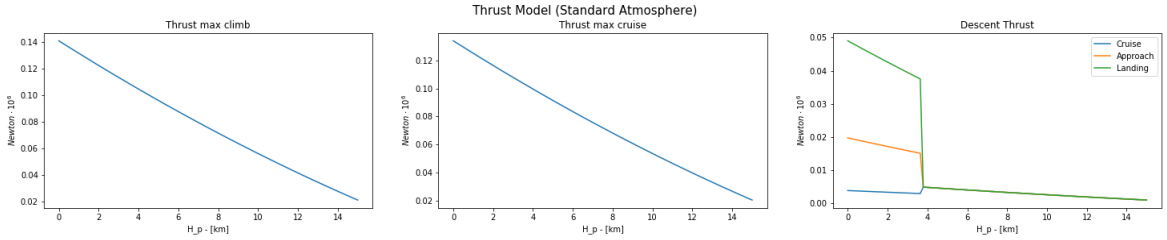

Figure 3.3: Maximum Jet engine thrust as function of altitude

The Jet engine thrust is calculated in Newtons and includes the contribution from all engines:

1. **Maximum Climb & Take-Off Thrust for Jet (BADA 4)**

$$Thr^{max}_{climb} = C_{TC,1} \cdot \left(1 - \frac{H_p}{C_{TC,2}} + C_{TC,3} \cdot H_p^2\right) \quad (3.34)$$

2. **Maximum cruise thrust for Jet (BADA 4)**

$$Thr_{cruise}^{max} = C_{TCR} \cdot Thr_{climb}^{max} = 0.95 \cdot Thr_{climb}^{max} \tag{3.35}$$

3. **Descent Thrust**

- if $H_p > H_{p,des}$:

$$Thr_{high}^{des} = C_{high}^{Tdes} \cdot Thr_{climb}^{max} \tag{3.36}$$

- if $H_p \leq H_{p,des}$:
  Configuration:

$$Cruise : Thr_{low}^{Tdes} = C_{low}^{Tdes} \cdot Thr_{climb}^{max} \tag{3.37}$$

$$Approach : Thr_{app}^{Tdes} = C_{app}^{Tdes} \cdot Thr_{climb}^{max} \tag{3.38}$$

$$Landing : Thr_{Ld}^{Tdes} = C_{Ld}^{Tdes} \cdot Thr_{climb}^{max} \tag{3.39}$$

```
CC====== Engine Thrust ================================================/
CC         Max climb thrust coefficients (SIM)                         /
CD     .14104E+06   .48917E+05   .65004E-10   .99797E+01   .80268E-02 /
CC     Desc(low)    Desc(high)   Desc level   Desc(app)     Desc(ld)  /
CD     .27207E-01   .45711E-01   .12398E+05   .13981E+00   .34750E+00 /
CC     Desc CAS     Desc Mach     unused       unused       unused    /
CD     .31000E+03   .78000E+00   .00000E+00   .00000E+00   .00000E+00 /
CC====== Fuel Consumption ============================================/
CC   Thrust Specific Fuel Consumption Coefficients                     /
CD     .63330E+00   .85903E+03                                        /
CC   Descent Fuel Flow Coefficients                                   /
CD     .91340E+01   .79668E+05                                        /
CC   Cruise Corr.     unused       unused       unused       unused    /
CD     .95423E+00   .00000E+00   .00000E+00   .00000E+00   .00000E+00 /
```

Figure 3.4: Engine thrust and fuel consumption coefficients for A320-212 aircraft [23]

Type	$C_{Tc,1}$ (N)	$C_{Tc,2}$ (ft)	$C_{Tc,3}$ (1/ft^2)	$C_{Tc,4}$ (°C)	$C_{Tc,5}$ (1/°C)	C_{f1} (kg/(min kN))	C_{f2} (knot)	C_{f3} (kg/min)	C_{f4} (ft)	C_{fcr} (dimensionless)
A320-212	141,040	48,917	6.5004E–11	9.9797	8.0268E–3	0.6333	859.03	9.134	79,668	0.95423
A321	158,520	45,206	1.1771E–10	9.8789	8.9517E–3	0.72987	1236.9	14.159	68,867	1
A330-301	363,130	57,820	–8.676E–12	9.6793	8.5160E–3	0.61503	919.03	21.033	112,230	0.93655
A340-313	381,180	56,784	1.4767E–11	9.8968	7.7097E–3	0.62965	1020.9	31.094	75,361	0.92082
B737-800	146,590	53,872	3.0453E–11	9.6177	8.5132E–3	0.70057	1068.1	14.190	65,932	0.92958
B757-200	193,530	52,967	4.9979E–11	9.9624	9.2638E–3	0.70334	914.35	12.967	100,900	1.0081
B767-300ER	322,410	56,718	1.3638E–11	9.5535	3.7698E–3	0.74220	2060.5	15.902	145,380	0.90050
B777-300	437,060	51,125	5.7969E–11	9.4595	4.5323E–3	0.49160	487.93	26.872	89,777	0.88658

Figure 3.5: Performance parameters for several aircraft (BADA 4)
Source: "Base of Aircraft Data-based operating cost prediction of arrival flight delay under short-term weather" Ming Zhang [23]

3.7 Formulation as Multistage Optimal Control Problem

According to BADA procedure model, the airplane flight consists of three separate flight phases: climb, cruise and descent.

- The climb phase starts after the take off and finishes when the aircraft reaches a predefined cruising altitude. For the airplane to climb with a positive climb angle γ, the lift created by the aerodynamic phenomenon on the wings has to overcome the weight of the aircraft. This happens by increasing speed by using more thrust and by increasing the angle of attack α.
- Cruise starts when the aircraft arrives at a predefined cruising altitude and travels in a straight line towards the destination ($\gamma \approx 0$). An aircraft cruises at a constant speed when the thrust produced by the jet engines is equal with the aerodynamic drag (Drag = Thrust).
- Descent phase is the portion of flight where the aircraft gradually decreases altitude with a negative climb angle ($\gamma < 0$). Usually the aircraft descents from the cruising altitude towards the airport for landing.

At each flight phase we have to use different models for thrust and fuel consumption, as well as different initial and final conditions. Thus, we have to model our problem as a multistage optimal control problem with ($i = 3$) three Stages: Climb, Cruise and Descent. During the climb phase the airplane starts climbing with an initial velocity $V_{TO} = 0.074\ km/sec$, and at Descent phase lands with a final velocity of $V_{LD} = 0.0698\ km/sec$. The initial time for climb is always fixed at $t_0 = 0$ while the initial and final time for cruise ($t_1,\ t_2$) are free parameters for optimization. The final time for descent phase t_f can be either set free or fixed for experimentation.

3.7.1 Objective function

During this thesis our objective is to optimize the airplane trajectory and the controls in order to minimize the fuel consumption during flight. The most logical objective function is to use a Mayer term that depends on the aircraft mass at final time t_f. For this reason our main objective is to maximize the aircraft's mass at final time t_f.

$$\max m(t_f) \Leftrightarrow \min -m(t_f) \tag{3.40}$$

Maximization of the final mass $m(t_f)$ is mathematically equivalent to minimizing the negative final mass $-m(t_f)$. In order to compare different control policies we also run minimum time experiments. For minimum time experiments we can just minimize the final time t_f.

- **The Bolza-type Objective function is of the form:**

$$J := \sum_{i=0}^{2} \int_{t_i}^{t_i+1} L_i(t, x(t), u(t), p)dt + \Phi(t_3, x(t_3)) \tag{3.41}$$

$$Stages: \ Climb, \ Cruise \ and \ Descent \ i \in \{0, 1, 2\}. \tag{3.42}$$

- **Mayer term for fuel consumption minimization**:

$$\Phi_3^m(t_3, x(t_3)) = -q_3^m \cdot m(t_f) \tag{3.43}$$

- **Mayer term for final time**:

$$\Phi_3^{t_f}(t_3, x(t_3)) = q_3^{t_f} \cdot t_f \tag{3.44}$$

- **Lagrange terms**:

$$L_i = q_i^{\delta} \cdot \delta^2 + q_i^h \cdot (z - H_p^{ref})^2 \tag{3.45}$$

The Lagrange functional L_i consists of two terms. The first term is δ^2 which makes the thrust smooth and close to zero. Also by adding δ quadratically we avoid singularities. Our problem depends linearly to δ through the dynamical equations, which can result in singularities because the second derivative equals to zero. The second term $(z - H_p^{ref})^2$ is the squared error between the altitude z and a reference cruising altitude H_p^{ref}. It is used only during cruise phase and only if we want to force the aircraft to cruise at a predefined altitude.

3.7.2 Constraints formulation

In order to complete the mathematical formulation of our problem we need to define equality and inequality constraints. The constraints are initial, final and interior point constraints for each stage, as well as box constraints for parameters, controls and state variables.

3.7.3 Box constraints for states and controls

In the table (3.1), we can see the box constraints for each phase for the state variables and the controls. The velocity v is quite free with a minimum that should not be smaller than the take off speed and a maximum of 0.25 km/sec which is realistic for an A320. The climb angle γ is positive during climb, negative during descent and it can be positive or negative at cruise phase. During cruise we do not need to constrain γ to be equal to zero very strictly at each node at cruise phase. In practice we get a good solution with $\gamma \approx 0$ just by restricting the start and the finish of the

cruise phase with initial and final equality constraints ($\gamma(t_1) = \gamma(t_2) = 0$). This also makes our problem more feasible and easier to solve numerically.

Climb	**Cruise**	**Descent**
$0.07 \leq v \leq 0.25$ km/sec	$0.07 \leq v \leq 0.25$ km/sec	$0 \leq v \leq 0.25$ km/sec
$0 \leq \gamma \leq 1.57$ rad	$-1.57 \leq \gamma \leq 1.57$ rad	$-1.57 \leq \gamma \leq 0$ rad
$0 \leq x \leq 1000$ km	$0 \leq x \leq 1000$ km	$0 \leq x \leq 1000$ km
$z_{min} \leq z \leq z_{max}$ km	$z_{min} \leq z \leq z_{max}$ km	$z_{min} \leq z \leq z_{max}$ km
$40 \leq m \leq 70$ tons	$40 \leq m \leq 70$ tons	$40 \leq m \leq 70$ tons
$0 \leq \alpha \leq 6$ deg	$0 \leq \alpha \leq 6$ deg	$0 \leq \alpha \leq 6$ deg
$0 \leq \delta \leq 1$	$0 \leq \delta \leq 1$	$0 \leq \delta \leq 1$

Table 3.1: Box Constraints during Climb, Cruise and Descent

The altitude z is constrained by z_{min} usually zero in our case and z_{max} which is the maximum cruising altitude which can vary between experiments. The state x is restricted between zero and the maximum distance 1000 km. The mass m is restricted between the starting mass $m(t_0) = 70$ tons and and 40 tons which is close to the minimum weight for an A320.

The controls are also constrained by minimum and maximum values. The angle of attack α is between zero and six degrees which is a region where the Drag and Lift are almost linear. The thrust lever position δ is a scalar between zero and one which represents values between zero and maximum thrust.

Finally, we need to constrain the parameters t_1, t_2, t_3, t_4 between some feasible minimum and maximum values:

$$
\begin{aligned}
t_1^{min} \leq t_1 \leq t_1^{max} \\
t_2^{min} \leq t_2 \leq t_2^{max} \\
t_3^{min} \leq t_3 \leq t_3^{max} \\
t_4^{min} \leq t_4 \leq t_4^{max}
\end{aligned}
$$

3.7.4 Initial, terminal and interior point constraints for each stage

During the experiments at chapter 4, we used both a simple thrust model, with a constant maximum available thrust ($T_{max} = const$) and the realistic jet engine thrust model presented at section (3.6). The realistic jet engine thrust model provided by BADA has a discontinuity at the descent phase when the airplane crosses from a high descent to a low descent at $H_p = H_{p,des}$. In order to treat this discontinuity we simply introduce one more model stage by splitting the descent phase to a high descent and a low descent phase.

Constraints for the simple thrust model (experiments [P1] and [P2])

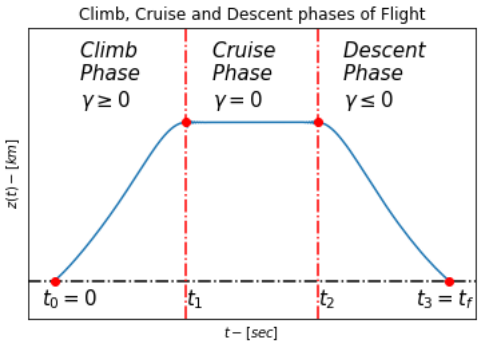

Figure 3.6: climb, cruise and descent stages for the simple thrust model with $t_3 = t_f$

For the simple thrust model (figure 3.6), use initial and terminal constraints for climb, cruise and descent stages at the initial and final times t_0, t_1, t_2, $t_3 = t_f$ and also interior point constraints:

$$r(t_0) = \begin{pmatrix} x(t_0) - X_{TO} = 0 \\ z(t_0) - H_{pTO} = 0 \\ v(t_0) - V_{TO} = 0 \\ m(t_0) - M_{TO} = 0 \\ \gamma(t_0) \geq 0 \\ \xi(t_0) - 0.8 \geq 0 \\ 1.2 - \xi(t_0) \geq 0 \end{pmatrix}, \; r(t_1) = \begin{pmatrix} (z(t_1) - H_p^{Cruise} = 0)* \\ \gamma(t_1) = 0 \\ \xi(t_1) - 0.8 \geq 0 \\ 1.2 - \xi(t_1) \geq 0 \end{pmatrix}, \tag{3.46}$$

$$r(t_2) = \begin{pmatrix} (z(t_2) - H_p^{Cruise} = 0)* \\ \gamma(t_2) = 0 \\ \xi(t_2) - 0.8 \geq 0 \\ 1.2 - \xi(t_2) \geq 0 \end{pmatrix}, \; r(t_3) = \begin{pmatrix} x(t_3) - X_{LD} = 0 \\ z(t_3) - H_{pLD} = 0 \\ v(t_3) - V_{LD} = 0 \\ -\gamma(t_3) \geq 0 \\ \xi(t_3) - 0.8 \geq 0 \\ 1.2 - \xi(t_3) \geq 0 \end{pmatrix}, \tag{3.47}$$

$$r_i(t) = \begin{pmatrix} \xi(t) - 0.8 \geq 0 \\ 1.2 - \xi(t) \geq 0 \\ t_2 - t_1 \geq 0 \\ t_3 - t_2 \geq 0 \end{pmatrix} \tag{3.48}$$

Constraints for the full Jet engine thrust model (experiments [P3] and [P4])

For the full jet engine thrust model (figure) we use Initial and Terminal constraints for climb, cruise, high descent and low descent stages at the initial and final times t_0, t_1, t_2, t_3, $t_4 = t_f$ and also interior point constraints:

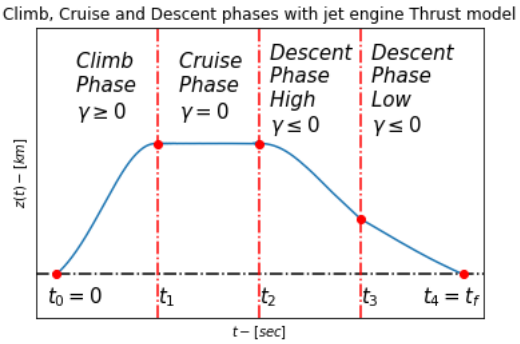

Figure 3.7: climb, cruise, high descent and low descent stages for the realistic jet engine thrust model (by BADA 4) with $t_4 = t_f$

$$r(t_0) = \begin{pmatrix} x(t_0) - X_{TO} = 0 \\ z(t_0) - H_{pTO} = 0 \\ v(t_0) - V_{TO} = 0 \\ m(t_0) - M_{TO} = 0 \\ \gamma(t_0) \geq 0 \\ \xi(t_0) - 0.8 \geq 0 \\ 1.2 - \xi(t_0) \geq 0 \end{pmatrix}, \; r(t_1) = \begin{pmatrix} (z(t_1) - H_p^{Cruise} = 0)* \\ \gamma(t_1) = 0 \\ \xi(t_1) - 0.8 \geq 0 \\ 1.2 - \xi(t_1) \geq 0 \end{pmatrix}, \tag{3.49}$$

$$r(t_2) = \begin{pmatrix} (z(t_2) - H_p^{Cruise} = 0)* \\ \gamma(t_2) = 0 \\ \xi(t_2) - 0.8 \geq 0 \\ 1.2 - \xi(t_2) \geq 0 \end{pmatrix}, \; r(t_3) = \begin{pmatrix} (z(t_3) - H_{p,des} = 0) \\ -\gamma(t_3) \geq 0 \\ \xi(t_3) - 0.8 \geq 0 \\ 1.2 - \xi(t_3) \geq 0 \end{pmatrix}, \tag{3.50}$$

$$r(t_4) = \begin{pmatrix} x(t_4) - X_{LD} = 0 \\ z(t_4) - H_{pLD} = 0 \\ v(t_4) - V_{LD} = 0 \\ -\gamma(t_4) \geq 0 \\ \xi(t_4) - 0.8 \geq 0 \\ 1.2 - \xi(t_4) \geq 0 \end{pmatrix}, \; r_i(t) = \begin{pmatrix} \xi(t) - 0.8 \geq 0 \\ 1.2 - \xi(t) \geq 0 \\ t_2 - t_1 \geq 0 \\ t_3 - t_2 \geq 0 \\ t_4 - t_3 \geq 0 \end{pmatrix} \tag{3.51}$$

The airplane always starts climbing from $(X_{TO},\ Z_{TO}) = (0, 0)$ km with an initial velocity $V_{TO} = 0.74$ km/sec and a positive climb angle $\gamma(t_0) > 0$ with an initial mass of $m(t_0) = 70$ tons. The airplane starts and finishes cruising in a horizontal position with a $\gamma(t_1) = \gamma(t_2) = 0$. The airplane finishes its descent after traveling a distance of 1000 km at $(X_{LD}, Z_{ld}) = (1000, 0)$ km with a final velocity $V_{LD} = 0.0698$ km/sec. The interior point constraints are evaluated at the nodes and they make sure that the load factor constraint $0.8 \leq \xi \leq 1.2$ is respected and that the parameters t_1, t_2, t_3, t_4 are ordered $t_1 \leq t_2 \leq t_3 \leq t_4$.

The equality constraints at the start and finish of the cruise phase marked with an asterisk $(*)$ $(z(t_1) = z(t_2) = H_p^{Cruise})$ is used only at the minimum time trajectories.

If we don't constraint our cruising altitude there are infinitely many minimum time trajectories. For example the algorithm will try to minimize t_f by also minimizing the distance traveled from the airplane by cruising in a lower altitude, even if we use the squared error $(z(t) - H_p^{Cruise})^2$ as a Lagrange term in the objective function. This additional equality constraints, ensure that the solution respects the reference cruising altitude at t_1, t_2, and in combination with the squared error $(z(t) - H_p^{Cruise})^2$ we cruise in a more or less constant altitude. Finally, we should also keep in mind that fuel consumption depends on the cruising altitude because the maximum thrust T_{max} and air density ρ are functions of the altitude H_p. Thus, in order to do fair comparisons between min time and min fuel experiments, we have to cruise in the same altitude.

The initial and final velocities are chosen to be realistic for an A320 type aircraft. In the scope of this thesis we didn't take into account the effect of take off and landing, it is realistic enough to asume that climb starts with a positive velocity already acquired during take off and that the airplane starts landing with a positive final speed. In practice the speed during landing cannot be zero because the airplane will stall. Moreover, landing speed cannot exceed a maximum threshold, otherwise the airplane may suffer structural damage during landing due to the big forces acting on the airplane. Hence, I chose a realistic landing speed for an A320 that has a mass around 70 tons. More importantly, I found in practice that by using fixed non zero initial and final velocities at t_0, t_f, it is easier to find feasible initial guesses for the multiple shooting algorithm initialization.

4 Numerical experiments

In order to determine an optimal strategy that minimizes the fuel consumption during flight, we have to study the effect of various flight parameters, mainly the effects of cruising altitude and the total time of flight t_f. For this reason we run two main experiments:

1. In the first type of experiment we study the effect of cruising altitude on the fuel consumption. The aircraft flies for 1000 km for a fixed final time t_f, and we study the fuel consumption for different cruising altitudes. For our first experiment, we expect that if we cruise higher, air density $\rho(H_p)$ and Drag(ρ,α,υ) are smaller because air is thinner. Thus, if less thrust will be used then less fuel will be consumed.

2. In the second type of experiment we study the effect of flight duration t_f on the fuel consumption. The aircraft flies for 1000 km for a fixed cruising altitude and we study the fuel consumption for different final times t_f. We also use a minimum time trajectory for comparison. For our second experiment, the main question is, if we could save fuel by prolonging the flight duration t_f. We want to investigate, if by cruising for a longer time with an overall lower speed, we could save fuel.

3. Another topic investigated during this thesis is the effect of Jet engine thrust on fuel consumption. As we already discussed at section 3.6, $T_{max}(H_p)$ is decreasing when altitude increases, as the air becomes less dense. The question is, how a decreasing maximum thrust is going to affect, (if at all) our optimal control policy, and how this more complicated thrust model compares with the simpler ($T_{max} = const$) model?

• A note on discontinuities and non-smoothness for atmosphere and thrust models

Atmosphere model and jet engine thrust models as described at paragraphs (3.4) and (3.6) suffer from non-smoothness and discontinuities. Though, we should not forget that our numerical algorithm is based on the calculation of derivatives for Newton's method and it uses runge-kutta methods to solve an initial value problem at each subinterval. At chapter 2 we assumed that our differential equations are piecewise differentiable on each model stage and piecewise continuous. In addition, we assumed that the states are continuous and piecewise differentiable.

1. The atmosphere model is non smooth at the tropopause $H_p = H_{P,trop} = 11.0km$. According to the ISA model the atmospheric temperature decreases

linearly until tropopause $H_p = H_{P,trop} = 11.0km$, and stays constant at the Stratosphere. As a result the temperature and air density $\rho(H_p)$ have a non-smooth "kink" at $H_{P,trop} = 11.0km$.

2. The descent thrust of a jet engine is discontinuous and has two phases. An A320 aircraft initially performs a "high altitude descent" from cruising altitude to $H_{P,des} = 3.7789104$ km with $Thr_{des}^{high} = C_{Tdes}^{high} \cdot Thr_{climb}^{max}$. In the second descent phase, called the "low altitude descent" the airplane descents further with $Thr_{des}^{low} = C_{Tdes}^{low} \cdot Thr_{climb}^{max}$ as it prepares for landing. For simplicity we assumed that the airplane performs a low descent at the "cruise" configuration.

The non-smoothness of the atmosphere model can be easily avoided, if we constrain our experiments strictly at Troposphere $z_{max} \leq H_{P,trop}$. This is effortless and our experimental results can still be generalized. The treatment of the discontinuity in the thrust during descent phase can be tackled if we split the descent phase in two consecutive stages "high altitude descent phase" and "low altitude descent phase".

Thus, the experiments are repeated two times. In the first set of problems (**P1** and **P2**) we use the simple thrust model ($T_{max} = const$) with maximum thrust constant and independent of altitude. For **P1** and **P2** we only need three flight phases: climb, cruise and descent ($n_{ph} = 3$).

In the second set of experiments (**P3** and **P4**), we test the full thrust model $T_{max}(H_p)$. To treat the discontinuity at $H_{P,des}$, **P3** and **P4** have four flight phases: climb, cruise, high descent and low descent ($n_{ph} = 4$).

Experiment	Type of Experiment	Thrust model	Number of stages
P1	Cruising Altitude	$T_{max} = const$	3
P2	Final Time	$T_{max} = const$	3
P3	Cruising Altitude	$T_{max}(H_p)$	4
P4	Final Time	$T_{max}(H_p)$	4

Table 4.1: Summary of experiments

4.1 Initial guesses for the multiple shooting variables

In this section we discuss the initialization of the multiple shooting variables for each model stage. MUSCOD-II solves optimal control problems for systems described by ordinary differential equations. With the multiple shooting method the optimization horizon is discretized by choosing a finite grid of subintervals. The differential equations are solved independently on each of these subintervals by a state-of-the-art ode solver. With this method we reformulate the original optimal control problem as a large-scale nonlinear program. Then we solve the NLP iteratively with Sequential

Quadratic Programming. During SQP iterations our algorithm starts with an initial guess and generates a sequence of iterates based on Newton's direction, in order to improve upon the approximate solutions. The algorithm iterates until convergence at some local minimum is achieved. The quality of the final solution and the total computational time, depend on the quality of the initial guess, as well as on the number and position of points of our grid. For example, if our initial guess is far away from the optimal solution the algorithm may converge very slowly, it may converge to a bad local minimum, or it may not converge at all. Additionally, we should keep in mind that the initial guess should be feasible, and we should make sure that solutions exist for the initial value subproblems for all the ode's in all subintervals.

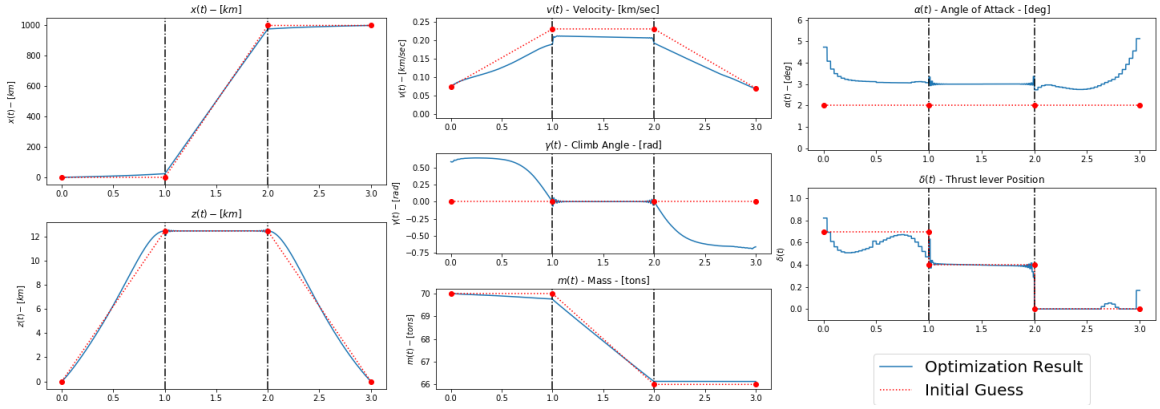

Figure 4.1: Initialization of Multiple Shooting with linear interpolation vs Optimization result

The first step is to descritize time by choosing a suitable grid for each phase. For experiments **P1** and **P2** Climb and descent phases are discretized with a dense grid of $n_{climb} = n_{descent} = 30$ multiple shooting nodes. Cruise phase was discretized with $n_{cruise} = 90$ shooting nodes, resulting in 151 discretization points in total. For experiments **P3** and **P4** Climb phase is discretized with $n_{climb} = 30$ multiple shooting nodes. Cruise phase was discretized with $n_{cruise} = 90$ shooting nodes, and descent low and descent high $n^{high}_{descent} = n^{low}_{descent} = 15$ resulting in 151 discretization points in total. Controls α and δ are approximated by piecewise constant basis functions, $\phi_{ij}(t, q_{i,j}) = q_{i,j}$ for the multiple shooting intervals $t \in I_{i,j}$. The initial values for multiple shooting variables are automatically generated by linear interpolation between the start and end values for each of the three stages. The start and end values for each of the three stages, are an educated guess by the user and it is important to choose them close to the final solution. As we can observe in the figure (4.1), even if the linear interpolation initialization is rather crude, it leads to a good local minimum. It is crucial that the initial values for the parameters t_1, t_2, t_3, t_4 are chosen to a feasible and close to the final solution. By trial and error I found that a good initialization for the climb time t_1 is between 200 sec and 300 sec. Also, I choose t_2, t_3 and t_4 such that the descent time is also between 200 and 300 sec. For

most experiments the final time is fixed in a predetermined feasible value, except when we want to optimize it as a parameter, together with a min fuel or a min time objective function. In general, I found out that the algorithm is sensitive to the choice of t_1, t_2, t_3, t_4, and if we do not choose good initial values it will fail to converge, or it may converge to bad and unrealistic local solutions (for example climb may be unrealistically long and cruise too short).

4.2 Results for simple thrust model

In this section we present the first two experiments **P1** and **P2** where we use constant maximum Thrust T_{max}=0.14104[KN]. Experiment **P1** studies the effect of different cruising altitudes on fuel consumption, while experiment **P2** studies the effect of different total time of flight t_f. Our model takes into account the effects of air density $\rho(H_p)$ in the aerodynamic forces, Lift and Drag but it ignores the fact that the maximum available thrust T_{max} decreases almost linearly with altitude. The three stages of the flight are modeled according to the figure (4.2):

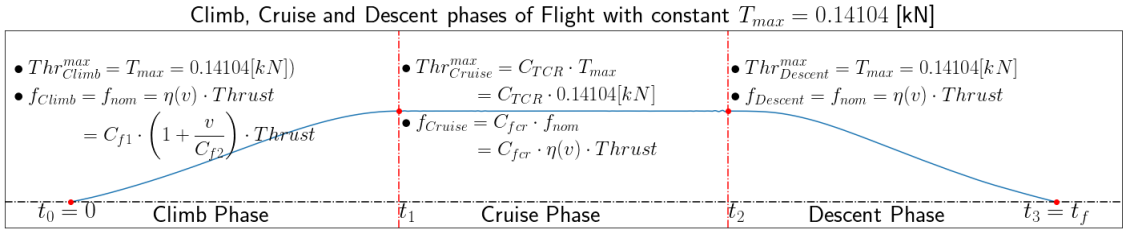

Figure 4.2: Differences between climb, cruise and descent stages

According to BADA the three stages of flight (figure 4.2) differ with respect to fuel consumption and with respect to maximum available Thrust. During climb and cruise we have available the total maximum thrust $T^{climb}_{max} = T^{descent}_{max} = T_{max}$, while on cruise phase we have only 95% of maximum thrust $T^{Cruise}_{max} = 0.95 \cdot T_{max}$. With respect to fuel consumption, the fuel flow during climb and during descent equals the nominal fuel flow $f_{climb} = f_{descent} = f_{nom}$, while during cruise phase the fuel flow is reduced to be $f_{cruise} = C_{fcr} \cdot f_{nom}$. The coefficient $C_{fcr} = 0.95423$ and is available for A320-212 aircraft by BADA database. Our model takes into account, that the airplane configuration during cruise is optimized for reduced fuel consumption, even if the maximum available thrust is slightly smaller.

4.2.1 [P1] Fuel minimization with different cruising altitude with simple thrust model

For our first experiment, during the climb phase the airplane starts climbing with an initial velocity $V_{TO} = 0.074\ km/sec$, and lands with a final velocity of $V_{LD} = 0.0698\ km/sec$ at Descent phase. The airplane flies for 1000 km without turning.

We keep constant $t_f = 5000sec$ and study how changing the cruising altitude affects the fuel consumption.

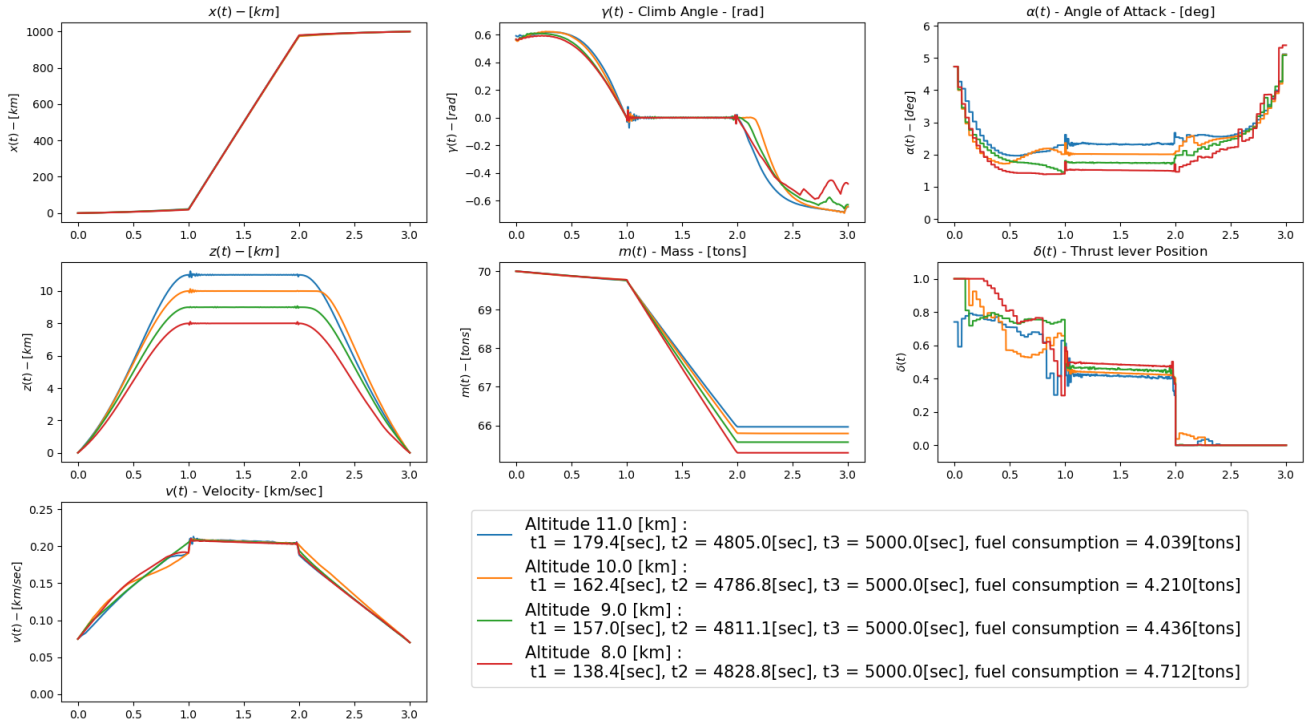

Figure 4.3: Comparison between different cruising altitudes for simple thrust model

As we can observe in Figure (4.3) and Table (4.2), for a fixed final time $t_f = 5000\ sec$ the higher you cruise the less fuel you consume. This can be explained by the fact that, as we cruise in higher altitude, air density ρ is smaller and thus the Drag becomes smaller. As a result, our airplane needs to use less thrust in order to counteract the drag in order to cruise in a constant speed (cruise with constant speed assumes: Thrust=Drag). Cruising in higher altitude, with smaller air density, also results in smaller lift. The optimal strategy compensates the loss of lift by increasing the angle of attack α (Lift(v,ρ,α)=$m(t) \cdot g$). Therefore, as the cruise altitude increases the optimal thrust lever position δ decreases and the optimal angle of attack α increases.

max Cruise Altitude [km]	**Consumed Fuel [tons]**
11.0	**4.039**
10.0	4.210
9.0	4.436
8.0	4.712

Table 4.2: Fuel consumption vs max cruise altitudes with $t_f = 5000$ sec (fixed)

For a min Fuel solution, Thrust lever position δ becomes very similar to a minimum time, bang-bang style solution. When the optimal control switches from one extreme to the other, then that control is referred to as a bang-bang solution. During Climb, the optimal control policy is to use the maximum amount of Thrust, until the airplane reaches the cruising altitude with the maximum speed. During cruise, we use a moderate amount of thrust, which is enough to counteract the aerodynamic Drag $Thrust = Drag$, while we cruise with a more or less constant speed. During Descent the thrust lever position is almost zero $\delta \approx 0$ and the angle of attack α increases to create drag, because our airplane needs to decelerate very quickly in order to reach a safe landing velocity.

In the table bellow we can see the weights for our Objective function. Where we give the most emphasis to our Mayer $m(t_f)$ objective with a big q_{mass} and a small weight q_d for the Lagrange objective δ^2

max Cruise Altitude [km]	q_d	q_{mass}
11.0	0.0001	1.0
10.0	0.0001	1.0
9.0	0.0001	1.0
8.0	0.0001	1.0

Table 4.3: Objective function weights for fuel consumption vs max cruise altitudes with $t_f = 5000$ sec (fixed)

4.2.2 [P2] Fuel minimization with different flight duration t_f with simple thrust model

In the second type of experiment, we keep the cruising altitude constant, and study how total flight duration t_f affects fuel consumption. During the climb phase the airplane starts climbing with an initial velocity $V_{TO} = 0.074\ km/sec$, and lands with a final velocity of $V_{LD} = 0.0698\ km/sec$ at Descent phase. The airplane flies for 1000 km without turning at a constant cruising altitude $H_{Cruise} = 11.0$[Km].

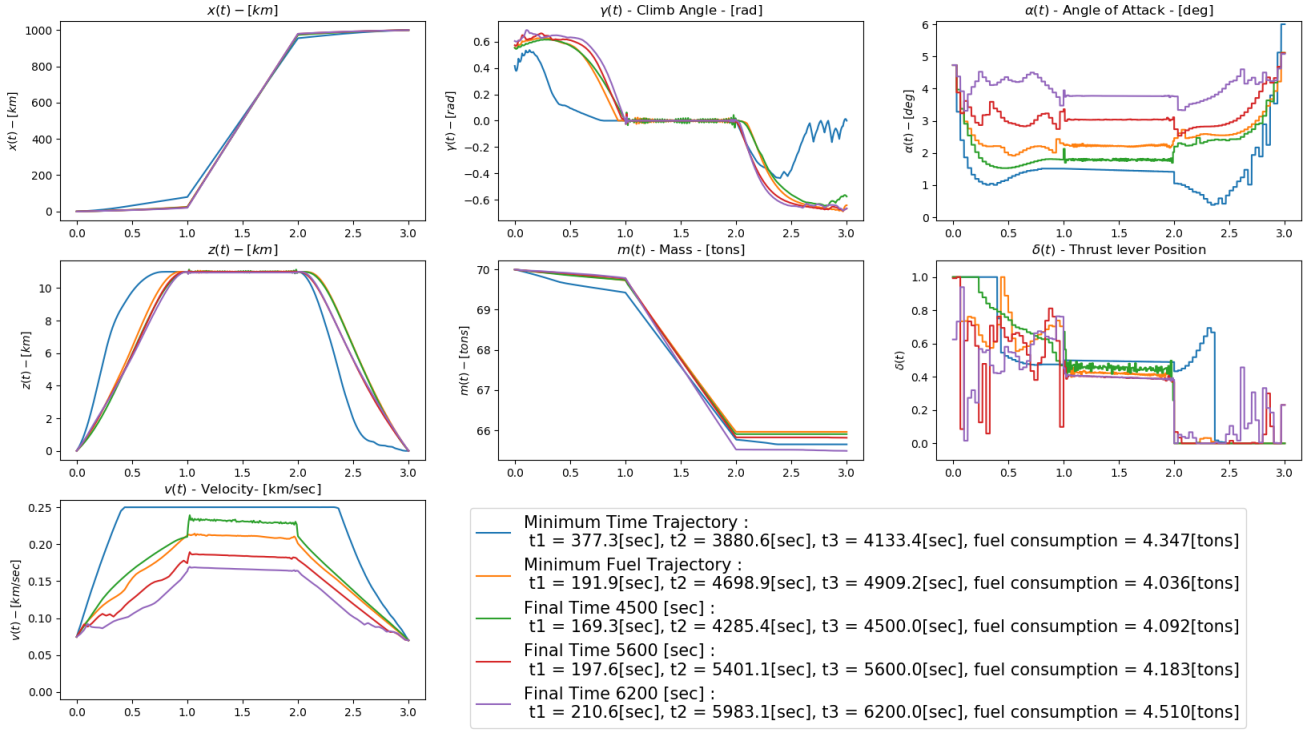

Figure 4.4: Comparison between different final time t_f for simple thrust model

At first, at Figure (4.4) and Table (4.4) we can observe that the optimal result t_f^*, is given by the experiment named **min fuel**, when the final time $t_3 = t_f$ is optimized as a free parameter together with fuel consumption. If we get far away from this optimal t_f^* by flying slower or faster, fuel consumption is increased. If we try to fly faster with $t_f < t_f^*$, our control policy resembles more the **min time** solution, and we consume more fuel. This happens because in min time trajectories we need to use thrust more aggressively to travel at maximum speed.

max Final Time t_f[sec]	**Consumed Fuel[tons]**
4133.4 (Min Time)	4.347
4500	4.092
4909.2 (Min Fuel)	**4.036**
5600.0	4.183
6200.0	4.510

Table 4.4: Fuel consumption vs final time t_f

If we try to fly slower with an increased final time $t_f > t_f^*$, we use less thrust δ in order to fly with less speed v. In order to compensate for $Lift(v, \alpha) = m \cdot g$ we have to travel with a higher angle of attack α. Unfortunately, this reduction on thrust has a negligible effect on fuel consumption because the cruising time $= t_2 - t_1$

becomes much bigger. If we assume a constant cruising speed in a constant cruising altitude with a constant Thrust then $f_{cr} = C_{fcr} \cdot \eta(v) \cdot Thrust$ is going to be a more of less constant number.

$$v \approx const,\ Thrust \approx const\ then: \tag{4.1}$$
$$f_{cr} = C_{fcr} \cdot \eta(v) \cdot Thrust \approx const \Rightarrow \tag{4.2}$$
$$\dot{m} = -f_{cr} \Rightarrow \tag{4.3}$$
$$m(t_2) = m(t_1) - f_{cr} \cdot (t_2 - t_1) \tag{4.4}$$

Hence, the change of mass during cruise depends linearly on the duration of cruise phase ($= t_2 - t_1$). Furthermore, if we take into account that usually the climb and descent duration are very small compared to cruise duration, and that the majority of the fuel is consumed during cruise phase, we can deduce that prolonging the flight with $t_f > t_f^*$ is also going to increase the overall fuel consumption of the flight.

t_f **[sec]**	q_d	q_h	q_{mass}	q_t	**constraints**
min fuel	0.0001		5.0		
min time	0.1			1.0	*
4500	0.0001		5.0		
5600	0.0001	0.01	100.0		
6200	0.0001	0.01	100.0		

Table 4.5: Objective function weights for fuel consumption vs final time t_f sec with simple thrust model

On the Table(4.5) above, we can see the weights used in our Objective function for each experiment. On the **min fuel**, $t_f = 4500$ sec experiments we use only $q_d = 0.0001$ for the Lagrange objective δ^2 and $q_{mass} = 5$ for the Mayer objective $m(t_f)$. For experiments with $t_f = 5600$ and 6200 sec the algorithm could not converge in a solution with a constant altitude during cruise phase. Fuel consumption depends indirectly on altitude, due to the effect of air density $\rho(H_p)$ on aerodynamic Lift and Drag. Therefore, to perform a fair comparison between different experiments all of them have to be in the same altitude. In order to enforce a local solution with a constant cruising altitude ($H_p^{cruise} = 11.0$ km), I added, only during cruising phase, the squared error between the altitude z and the reference cruising altitude $q_h \cdot (z - H_p^{cruise})^2$ as an additional Lagrange objective. The **min time** experiment has only a Lagrangian term $q_d \cdot \delta^2$ and the final time $q_t \cdot t_f$ as a Mayer term. Moreover, in order to force the **min time** solution to travel at a cruising altitude $H_p^{cruise} = 11.0$ km it was sufficient to fix $(*)z(t_1) = z(t_2) = H_p^{cruise}$ with two additional equality constraints at the start and finish of cruise phase (t_1,t_2).

4.3 Results with realistic Jet engine thrust

In this section we will study the generation of optimal flight trajectories with respect to fuel consumption, when the jet engine dynamics are taken into account. This would be a logical next step for our study because fuel flow $f_{nom} = \eta(v) \cdot Thrust(H_p) = \eta(v) \cdot \delta \cdot Thr_{max}(H_p)$ depends on thrust. The maximum available Thrust $Thr_{max}(H_p)$ depends both on altitude and the aircraft configuration on each stage of the flight. As we fly higher air density decreases and as a result less maximum thrust is available. Furthermore, at each stage of flight the airplane changes settings and configuration in order to optimize performance.

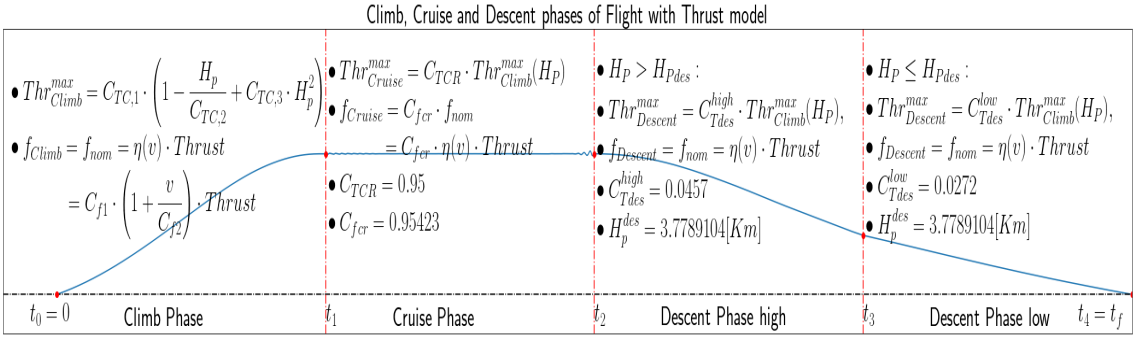

Figure 4.5: Summary of stages for climb, cruise, descent high and descent low with jet engine thrust as a function of altitude

As we can observe on Figure (4.5) the thrust has a separate nonlinear model at each stage of flight. During climb phase the maximum available thrust decreases almost linearly. Cruise phase has available only 95 % of the thrust at climb phase. During descent phase the thrust is small and discontinuous at $H_p^{des} = 3.778$ km. Additionally, the fuel flow during climb and descent phase equals the nominal fuel flow ($f_{climb} = f_{descent} = f_{nom}$), while during the cruise phase the fuel flow is reduced for efficiency ($f_{cruise} = C_{fcr} \cdot f_{nom}$).

4.3.1 [P3] Fuel minimization with different cruising altitude with realistic Jet engine thrust

The third experiment incorporates realistic jet engine thrust under ISA conditions, during the climb phase the airplane starts climbing with an initial velocity $V_{TO} = 0.074$ km/sec , and lands with a final velocity of $V_{LD} = 0.0698$ km/sec at Descent phase. The airplane flies for 1000 km without turning. We keep constant $t_f = 5200sec$ and study how changing the cruising altitude affects the fuel consumption.

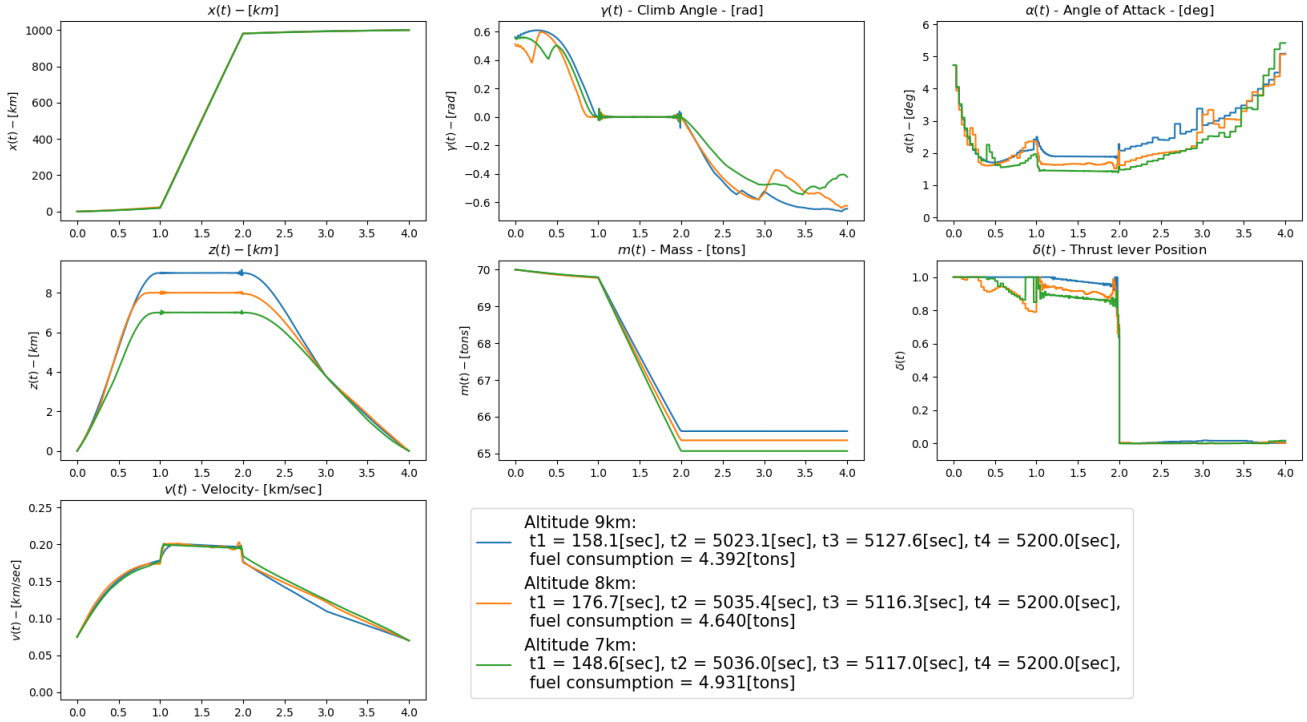

Figure 4.6: Comparison between different cruising altitudes with Jet engine thrust as a function of altitude

The results from experiment **P3** at Figure (4.6) confirm that flying on a higher cruising altitude is the best strategy to minimize fuel consumption. The optimal control strategy for δ keeps the bang-bang shape which becomes even more pronounced with even higher δ during climb and cruise. The difference in this experiment is that δ increases when the airplane cruises at higher altitude. We should keep in mind that the maximum thrust decreases with altitude. Therefore, as we fly higher, we need higher thrust lever position δ in order to compensate for the small maximum available Thrust at high altitudes. The optimal angle of attack α follows the same strategy as **P1**. As the cruising altitude increases the optimal angle of attack α also increases.

max Cruise Altitude [km]	**Consumed Fuel [tons]**
9.0	**4.392**
8.0	4.640
7.0	4.931

Table 4.6: Fuel consumption vs max cruise altitudes with Jet engine thrust and $t_f = 5200$ sec (fixed)

Bellow we can see the weights for the objective function optimized for this experiment.

max Cruise Altitude [km]	q_d	q_{mass}
9.0	0.0001	100.0
8.0	0.0001	100.0
7.0	0.0001	100.0

Table 4.7: Objective function weights for fuel consumption vs max cruise altitudes with Jet engine thrust and $t_f = 5200$ sec (fixed)

4.3.2 [P4] Fuel minimization with different flight duration t_f with realistic Jet engine thrust

Finally, with this last experiment we study how total flight duration t_f affects fuel consumption with the use of jet engine thrust model. During the climb phase the airplane starts climbing with an initial velocity $V_{TO} = 0.074\ km/sec$, and lands with a final velocity of $V_{LD} = 0.0698\ km/sec$ at Descent phase. The airplane flies for 1000 km without turning at a constant cruising altitude $H_{Cruise} = 9.0$[Km].

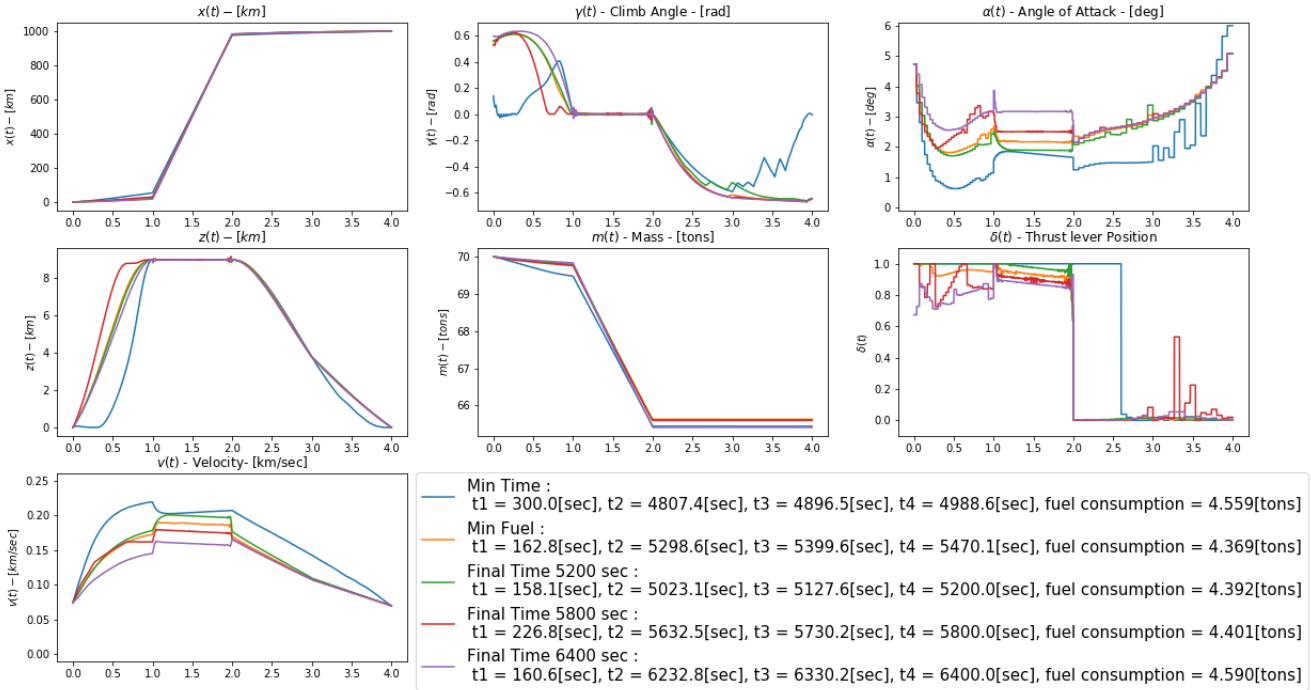

Figure 4.7: Comparison between different final time t_f with Jet engine thrust as a function of altitude

The results at Figure (4.7) with the realistic Thrust model, confirm our findings with the constant maximum thrust model. The only difference is that the airplane has less maximum thrust, thus it is not feasible to fly in the same maximum altitudes and with the same maximum speed. The best result t_f^* is given when we optimize

the final time $t_4 = t_f$ as a free parameter together with fuel consumption. If we try to arrive to our destination slower or faster then the fuel consumption is increased. As we have explained already in **P2** at section (4.2.2) when the airplane tries to arrive faster than the optimal $t_f < t_f^*$ then the optimal control policy resembles that of the bang-bang min time solution. The airplane accelerates as fast as possible by using maximum thrust $\delta \approx 1$ during climb cruise and high descent phases. Then, at low descent phase it decelerates quickly by using zero thrust $\delta \approx 0$. When we fly slower with $(t_f > t_f^*)$ the fuel consumption again increases. As we have proven with equations (4.1)-(4.4) the fuel consumption increases almost linearly with the duration of the cruise phase. Thus, prolonging the flight also increases fuel consumption. Furthermore, this experiment confirms that when we fly slower with less thrust the algorithm compensates by increasing the angle of attack α. Thus, when the final time increases the optimal angle of attack also increases.

max Final Time t_f[sec]	**Consumed Fuel[tons]**
4988.6 (Min Time)	4.559
5200.0	4.392
5470.1 (Min Fuel)	**4.369**
5800.0	4.401
6400.0	4.590

Table 4.8: Fuel consumption vs final time t_f with Jet engine thrust as a function of altitude

At Table (4.9) we can see the weights for the objective function used for each experiment.

t_f **[sec]**	q_d	q_h	q_{mass}	q_t	**constraints**
min fuel	0.0001		100.0		
min time	0.0001			1.0	*
5200	0.0001		100.0		
5800	0.0001		100.0		
6400	0.0001	0.01	100.0		

Table 4.9: Objective function weights for fuel consumption vs final time t_f sec with Jet engine thrust as a function of altitude

During the minimum fuel experiments we used $q_{mass} = 100$ for Mayer objective $m(t_f)$ and $q_d = 0.0001$ for the Lagrange objective δ^2. Only at min fuel experiment for $t_f = 6400$ sec we also included at cruise phase $q_h \cdot (z - H_p^{Cruise})$ with $q_h = 0.01$ in order to find a local solution with constant altitude. The min time experiment has only a Lagrangian term $q_d \cdot \delta^2$ and the final time Mayer objective $q_t \cdot t_f$ as a Mayer term.

In order to force the min time solution to travel at a cruising altitude $H_p^{cruise} = 9.0$ km it was sufficient to set $(*)z(t_1) = z(t_2) = H_p^{cruise}$ with two additional equality constraints at the start and finish of cruise phase (t_1, t_2).

4.4 Comparison between the two thrust models

In this paragraph we will try to compare the two thrust models. The question is if one of the models is better than the other and which one should be used for numerical experiments. As we already described, problems **P1** and **P2** use a simple thrust model where $T_{max} = const$ while **P3** and **P4** use the altitude dependent thrust model provided by BADA 4.

At first by comparing the experiments both result in the same optimal control strategy for δ and α. This holds true for both the varying cruising altitude experiments and for the varying final time experiments. Though, if we compare the results between the two models we see a significant difference in the results. For example, the minimum fuel trajectory at **[P2]** estimates a minimum fuel consumption $m^*(t_f^*) = 4.036$ tonnes with an optimal final time $t_f^* = 4909.2$ sec while the same experiment at **[P4]** estimates a minimum fuel consumption $m^*(t_f^*) = 4.369$ tonnes with an optimal final time $t_f^* = 5470.1$ sec. We can see that for the a minimum fuel trajectory the simple thrust model underestimates the minimum fuel consumption by 333 kg and the final time t_f is 560.9 sec faster than the realistic jet engine thrust model. The minimum time experiment at **[P2]** estimates a minimum time of $t_f^* = 4133.4$ sec with $m^*(t_f^*) = 4.347$ tonnes of consumed fuel. The minimum time experiment at **[P4]** estimates a minimum time of $t_f^* = 4988.6$ sec with $m^*(t_f^*) = 4.559$ tonnes of consumed fuel. For a minimum time experiment the simple thrust model predicts t_f to be 855.2 sec faster and with 212 kg less fuel consumed than the realistic jet engine model.

If we take for granted that the jet engine model provided by BADA 4 is very accurate and very close to reality, the above examples show that the simple thrust model will produce inaccurate results as it underestimates the consumed fuel and it is very optimistic about the final time. Finally, as the simple thrust model does not depend on altitude it also overestimates the maximum cruise altitude and the aircraft's maximum speed. For the purpose of simple numerical experiments such as ours these discrepancies are not important, but if the purpose is the accurate prediction of fuel consumption, then the BADA 4 jet engine thrust model should be used.

Experiment	iterations	mean Wall Time (sec)	mean over experiments(sec)
[P1] 8km	20	4.882	6.0885
[P1] 9km	25	5.767	
[P1] 10km	34	8.106	
[P1] 11km	21	5.599	
[P2] 4500sec	23	5.366	6.0865
[P2] 5600sec	26	7.070	
[P2] 6200sec	36	5.747	
[P2] min fuel	30	6.163	
[P3] 7km	29	6.155	6.302
[P3] 8km	19	5.094	
[P3] 9km	30	7.657	
[P4] 5200sec	30	7.624	7.104
[P4] 5800sec	25	8.177	
[P4] 6400sec	26	6.456	
[P4] min fuel	19	6.159	
[P2] min Time	25	10.789	
[P4] min Time	23	35.977	

Table 4.10: Wall time comparison between experiments

The simple thrust model is less numerically complex. The first advantage of this is that our numerical optimization algorithm is less sensitive to the initial guesses than the full jet engine model, which means that we can spent less effort during initialization to achieve convergence.

Another interesting topic is how quickly the two models converge. We executed each experiment 100 times and we present the average wall times at table (4.10). The experiments were performed with a laptop equipped with an Intel Core i7-8750H CPU at 2.20GHz and 16 GB of RAM. As we can observe, **[P1]** is 3.4 % faster than **[P3]** and **[P2]** is 14.3 % faster than **[P4]**. On the minimum time experiments the simple thrust model is 70 % faster than the BADA 4 model. Thus, the simple thrust model could be exploited when we want to emphasize speed over accuracy. From all the above, we conclude that one model is not strictly better than the other, but that we should choose the thrust model according to the problem at hand.

5 Conclusions

5.1 Discussion

This thesis studied the optimization of aircraft trajectories with respect to fuel consumption from the optimal control point of view. The main question is if we can decrease the cost related to fuel consumption and CO_2 emissions without upgrading or replacing the existing aircraft fleet. In this thesis we demonstrate how to minimize fuel consumption by controlling optimally the angle of attack and the thrust lever position of an aircraft.

In order to perform our numerical experiments we modeled the three main stages of flight, namely climb, cruise and descent phases. A multistage optimal control approach was adopted, because each flight phase is described by different state differential equations and different constraints. The original 3D point mass aircraft model was reduced to a simpler 2D model with only five states and two controls, which greatly reduced the computational complexity of the problem. As a result, the convergence properties of the numerical algorithm improved and the computational time was decreased. Our mathematical formulation took into account the atmospheric properties for pressure, temperature and air density as a function of altitude by incorporating the international standard atmosphere model (ISA), as described at the base of aircraft data family 4 (BADA 4) documents. For more precise results, we incorporated the velocity dependent thrust specific fuel consumption $\eta(\upsilon)$ instead of a constant f_r. Finally, our models included the weakening of jet engine thrust when the flight altitude increases.

Extensive numerical experiments were conducted with MUSCOD II in order to to study and compare the effect of important flight parameters. Two types of experiments where conducted. In the first experiment, we studied the effect of the cruising altitude when the flight duration remained constant. In the second experiment, we kept the cruising altitude constant and compared the aircraft performance for different flight duration. We repeated the aforementioned experiments two times with different thrust models. The first time we assumed the maximum available thrust T_{max} to be constant and independent of altitude. The second time we took into account the weakening effect of increasing altitude in jet engine thrust. Minimum fuel and minimum time objective functions were developed and optimal control strategies for angle of attack and thrust lever position were generated, compared and analyzed. Our experiments lead to the following conclusions:

As we cruise in higher altitude with fixed flight duration:

- For a min Fuel solution, thrust lever position δ is similar to a bang-bang

solution. The maximum amount of thrust is used during Climb. During cruise phase, enough thrust is used in order to counteract the aerodynamic drag ($Thrust = Drag$). And during descent phase the airplane uses almost no thrust ($\delta \approx 0$) in order to decelerate as fast as possible.

- During cruise phase, for the simple thrust model when the altitude increases, the thrust lever position δ decreases because the aerodynamic drag is smaller in higher altitude ($Thrust = Drag$). This is not true though when we use the full jet engine thrust model, because the thrust weakens when we fly higher and thus we use δ more aggressively to produce the same amount of thrust. Hence, for the full jet engine model when the altitude increases the thrust lever position δ also increases.

- Angle of attack α increases when the altitude increases in order to create adequate lift to counteract the airplane's weight (Lift(v,α)=$m(t) \cdot g$).

As we cruise at a constant altitude with varying flight duration:

- The best result comes when t_f is optimized as a free parameter together with fuel consumption (marked as min fuel). Getting far away from this optimal t_f^* results in increased fuel consumption.

- If we try to fly faster $t_f < t_f^*$, our control policy resembles more the min time solution, and we consume more fuel. This happens because in min time trajectories we need to use thrust more aggressively to travel at maximum speed.

- If we try to fly slower $t_f > t_f^*$, we use less thrust δ in order to fly with less speed v. In order to compensate for $Lift(v, \alpha) = m \cdot g$ we have to travel with a higher angle of attack α. This reduction on thrust has a negligible effect on fuel consumption because the cruising time $= t_2 - t_1$ becomes much bigger.

- Prolonging the flight duration is going to increase the overall fuel consumption of the flight. This happens because the fuel consumption depends linearly to the duration of the cruise phase ($t_2 - t_1$).

In all cases the optimal control approach gives optimal climb and descent flight profiles which are respecting the Continuous Climb Operations (CCO) and the Continuous Descent Operations (CDO) [11]. The algorithm converges to optimal solutions which do not include any segments of level flight during climb or descent phases.

Another contribution of this thesis is the comparison between the simple thrust model (T_{max} = const) at experiments **P1** and **P2** and the realistic jet engine thrust model at **P3** and **P4**. The comparison between the two thrust models, showed that both result in the same optimal control strategy for δ and α, which we already summarized in the previous paragraph. The big difference between the two models

is the realism and accuracy of the calculations. The disadvantage of the simple thrust model is that it provides less accurate results, which have a large difference from the realistic BADA4 jet engine model. If we compare the optimization results for parameters t_1, t_2, t_3, t_4 and fuel consumption we can see that there is a large discrepancy between the optimal values of the two models. This means that the BADA 4 jet engine thrust model is a better option if we need to estimate accurately the fuel consumption.

The simplified thrust model is less complex numerically, which results in faster convergence and smaller sensitivity to the initial guesses than the full model. Furthermore, the airplane with the realistic jet engine thrust model has less maximum thrust, thus it is not feasible to fly in the same maximum altitudes or with the same maximum speed.

5.2 Difficulties

The first issue we had to resolve was the existence of non-smoothness and discontinuities on our model. This can be a big problem for a numerical algorithm which assumes that the right hand side is many times continuously differentiable in order to calculate gradients and in order to use runge kutta integration methods. As we described on chapter 3, the temperature of the standard atmosphere model has a non-smooth "kink" when the geopotential pressure altitude is equal to $H_{p,trop} = 11$ km. We decided to take care of this non-smoothness by just doing experiments with maximum cruising altitudes from $H_{p,trop}$ and bellow. With this simple decision we avoid the addition of even more model stages to our model without loss of generality for our findings. The discontinuity is introduced on our model because we have to use a different fuel flow and different maximum thrust at each flight stage. A characteristic example is the maximum jet engine thrust explained at (3.6) which has a jump at $H_{p,des}$. In order to treat that jump for **P3** and **P4** we formulated one additional model stage by splitting the descent phase at high descent phase and low descent phase. In this way we ensure that our differential equations are piecewise differentiable and piecewise continuous on each model stage.

The second difficulty during this thesis was making the algorithm to converge. Initial implementations of the fuel minimization problem with poorly or not scaled values did not result in convergence. In order to make our problem well-conditioned we had to use both internal and external scaling. At first we scaled the values of states and all parameters to be close to one. For example, we use kilometers instead of meters, tonnes instead of kilograms and $[tons \cdot km/sec^2]$ instead of Newtons $(=[kg \cdot m/sec^2])$, etc. Furthermore, we scaled the states and parameters internally at MUSCOD II by choosing appropriate scaling factors at the .dat files. Moreover, our algorithm is sensitive to the choice of initial conditions for states, controls and for the parameters t_1, t_2, t_3, t_4. In order to determine good initial values we had to use trial and error and physical insight for the problem. We performed numerous different trials and then evaluated the results in order to choose good initial values

that lead to quick convergence on good local minima.

5.3 Future work

During this thesis we tackled the problem of airplane trajectory optimization with respect to fuel consumption. Our approach took into account the effects of thrust specific fuel consumption and the atmosphere by including the models for air density and jet engine thrust. Our experiments compared only the quantity of consumed fuel measured in tonnes. Future work could also take into account the cost of flight. The cost of flight could easily include the cost of fuel and the cost of CO_2 emissions or other flight related costs like flight delays or early arrivals. Taking into account the cost would be a straightforward task, because it could be easily included in the objective function. Another important problem for future investigation is the minimization of aircraft noise during flight near residential areas. According to "European Aviation Environmental Report 2019" from EASA, noise exposure could be added to the objective function with the L_{den} noise indicator [3]. Furthermore, future work could study the effects of turning on fuel consumption by reintroducing into the model the effects of course angle χ and the bank angle μ. Commercial flights may need to change direction in order to avoid bad weather, forbidden areas like military facilities or densely populated cities. Finally, this thesis ignored the uncertainties in the model and the effects of disturbances such as wind shear or bad weather. In the future it would be interesting to formulate the problem as a real time control problem for real time parameter estimation and disturbance mitigation.

Part I

Appendix

A Lists

A.1 List of Figures

A.2 List of Tables

B Bibliography

[1] "Paris Agreement", United Nations, 2015

[2] "The European Green Deal", European Commission, Brussels, 2019.

[3] "European Aviation Environmental Report 2019", European Environment Agency, EASA, EUROCONTROL.

[4] "COVID19 Impact on European Air Traffic", EUROCONTROL Comprehensive Assessment, Tuesday, 15 September 2020.

[5] "Cost Assessment of Near and Mid-Term Technologies to Improve new Aircraft Fuel Efficiency", The International Council on Clean Transportation, Anastasia Kharina, Daniel Rutherfor, Mazyar Zeinali.

[6] "Method for estimating aviation fuel burnt and emissions", EMEP/EEA air pollutant,emission inventory guidebook 2016, EUROCONTROL

[7] "Fuel consumption optimization in air transport: a review, classification, critique, simple meta-analysis", and future research implications Vedant Singh, Somesh Kumar Sharma, Springer, 2015

[8] "Reducing fuel consumption of subsonic aircraft by optimal cyclic cruise". Sachs G, Christodoulou T (1987) J Aircr 24(9):616–622

[9] "Optimized profile descent arrivals at Los Angeles international airport". Clarke JP, Brooks J, Nagle G, Scacchioli A, White W, Liu SR (2013) J Aircr 50(2):360–369

[10] "Non-optimality of the steady-state cruise for aircraft". Speyer JL (1976) AIAA J 14(11):1604–1610

[11] "The Benefits Of CCO/CDO Operations-European Task Force Findings, Climate Change Mitigation:Technology and Operations", The European CCO/CDO Task Force, David Brain and Marylin Bastin.

[12] "MUSCOD-II: Multiple Shooting Code for Optimal Control", (https://neos-server.org/neos/solvers/miocp:MUSCOD-II/AMPL.html)

[13] "User Manual for the Base of Aircraft Data (BADA) Family 4", revision 3.11, 2013.

[14] "User Manual for the Base of Aircraft Data (BADA) Family 3",Report No. 2009-009 ,2009.

[15] "Lecture Notes Optimization with Differential Equations", Version 0.7, Prof. Dr. Dr. h. c. mult. Hans Georg Bock, Winter Semester 2015/16, Heidelberg University

[16] "Optimal Control of Ordinary Differential Equations and Differential-Algebraic Equations", Matthias Gerdts,Universität der Bundeswehr München, 2006

[17] "Numerical Optimization". Jorge Nocedal and Stephen J. Wright. Springer, 2006.

[18] "MUSCOD II User Manual", release 6.0, Christian Hoffmann, Christian Kirches, Andreas Potschka, Sebastian Sager, Leonard Wirsching, Simulation and Optimization Group Hans Georg Bock and Johannes P. Schlöder, Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg, Germany 2014

[19] "A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems", Hans Georg Bock and K.J Plitt, In 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, July 1984.

[20] "Analyse und Restrukturierung eines Verfahrens zur direkten Lösung von Optimal-Steuerungsproblemen" (The Theory of MUSCOD in a Nutshell) (Diploma thesis), Daniel B. Leineweber, Interdisziplinares Zentrum fur Wissenschaftliches Rechnen (IWR), Heidelberg University, 1995

[21] "Optimal Control of 3D Separation Management" (master thesis), Markus Schubert, Ruprecht-Karls-University Heidelberg Faculty of Mathematics and Computer Science, Heidelberg, 2017.

[22] "Multi-Stage Strategy to Numerical Trajectory Optiization for Multiple Aircraft Approaching an Airport" (master thesis), Valeriy Semenov,Ruprecht-Karls-University Heidelberg Faculty of Mathematics and Computer Science, Heidelberg,2019.

[23] "Base of Aircraft Data–based operating cost prediction of arrival flight delay under short-term weather", Ming Zhang, Kai Liu, Xianglu Kong, Ming Zhang, Advances in Mechanical Engineering, SAGE, 2018

Erklärung:

Ich versichere, dass ich diese Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den (Datum) .....

Declaration:

I hereby confirm that I wrote this work independently and did not use any sources other than those indicated.

Heidelberg, (Date) .....