EFFICIENT AND ROBUST AIRCRAFT LANDING TRAJECTORY OPTIMIZATION

A Thesis
Presented to
The Academic Faculty

by

Yiming Zhao

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Daniel Guggenheim School of Aerospace Engineering

Georgia Institute of Technology
May 2012

EFFICIENT AND ROBUST AIRCRAFT LANDING TRAJECTORY OPTIMIZATION

Approved by:

Professor Panagiotis Tsiotras, Advisor
Daniel Guggenheim School of Aerospace Engineering
Georgia Institute of Technology

Professor Eric Feron
Daniel Guggenheim School of Aerospace Engineering
Georgia Institute of Technology

Professor John-Paul Clarke
Daniel Guggenheim School of Aerospace Engineering & H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Magnus B. Egerstedt
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Ryan P. Russell
Daniel Guggenheim School of Aerospace Engineering
Georgia Institute of Technology

Professor Eric N. Johnson
Daniel Guggenheim School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: January. 2012

To my family

ACKNOWLEDGEMENTS

It is my pleasure to thank so many people who made this thesis possible. First and foremost, I would like to thank my supervisor, Dr. Tsiotras, for his valuable advice and constant encouragement throughout the years of my Ph.D. study. His scholarly independent thinking and the pursuit of theoretical rigor have greatly influenced my research. It has always been an enjoyable experience to listen to his lectures and make discussions because of his excellent style of explaining complicated things in a surprisingly clear and simple way. In particular, I would like to thank him for his great support throughout my whole thesis-writing period, which are very much appreciated. My gratitude to him could never be overstated.

I would like to thank the members of my committee Dr. Eric Johnson, Dr. Eric Feron, Dr. Ryan Russell, Dr. John-Paul Clarke, and Dr. Magnus Egerstedt for their insightful comments and suggestions. They provided valuable opinions on promising research directions, and have inspired me to explore deeper in some subjects of my research. I would also like to thank Dr. Amy R. Pritchett for her guidance in cognitive engineering and air traffic control while I worked in the Cognitive Engineering Center at Georgia Tech in the early years of my graduate study.

I am indebt to those great teachers at Georgia Tech from whom I was fortunate enough to take some of the finest classes, including Dr. Olivier Bauchau (Advanced Dynamics), Dr. J.V.R. Prasad (Flight Dynamics), Dr. Wassim Haddad (Nonlinear System, and Robust Control), Dr. Panagiotis Tsiotras (Robust Control, and Optimal Guidance and Control), Dr. Magnus Egerstedt (Network Control), Dr. John-Paul Clarke (Air Traffic Control), and Yorai Wardi (Linear System), just to name a few.

I wish to thank my fellow student colleagues who built up a lively and fun environment during the lengthy process towards accomplishing this thesis. I have enjoyed my conversation and time with Dongwong Jung, Sachin Jain, Atri Dutta, Raghvendra Cowlagi, Stathis Bakolas, Oktay Arslan, Dae-Min Cho, Yunpeng Pan, Daniel Kuheme, Imon Chakraborty,

Nuno Filipe, Liang Du, Hancao Li, Tim Wang, Claus Christmann, Mehrdad Pakmehr, Jenna Stahl, So Young Kim, Jean-Francois Castet, Keumjin Lee, etc. I am also grateful to Qing Hui, Seung Man Lee, and Karen Feigh for their inspiration in different stages of my Ph.D. study.

Last, but the most, thank my family, especially my wife, for always encouraging and supporting me to pursue my interest. This thesis is devoted to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis addresses the challenges in the *efficient* and *robust* generation and optimization of three-dimensional landing trajectories for fixed-wing aircraft subject to prescribed boundary conditions and constraints on maneuverability and collision avoidance. In particular, this thesis focuses on the airliner emergency landing scenario and the minimization of landing time.

The main contribution of the thesis is two-fold. First, it provides a hierarchical scheme for integrating the complementary strength of a variety of methods in path planning and trajectory optimization for the improvement in efficiency and robustness of the overall landing trajectory optimization algorithm. The second contribution is the development of new techniques and results in mesh refinement for numerical optimal control, optimal path tracking, and smooth path generation, which are integrated in the hierarchical scheme and applied to the landing trajectory optimization problem.

A density function based grid generation method is developed for the mesh refinement process during numerical optimal control. A numerical algorithm is developed based on this technique for solving general optimal control problems, and is used for optimizing aircraft landing trajectories. A path smoothing technique is proposed for recovering feasibility of the path and improving the tracking performance by modifying the path geometry. The optimal aircraft path tracking problem is studied and analytical results are presented for both the minimum-time, and minimum-energy tracking with fixed time of arrival. The path smoothing and optimal path tracking methods work together with the geometric path planner to provide a set of feasible initial guess to the numerical optimal control algorithm.

The trajectory optimization algorithm in this thesis was tested by simulation experiments using flight data from two previous airliner accidents under emergency landing scenarios. The real-time application of the landing trajectory optimization algorithm as part of the aircraft on-board automation avionics system has the potential to provide effective guidelines to the pilots for improving the fuel consumption during normal landing process, and help enhancing flight safety under emergency landing scenarios. The proposed algorithms can also help design optimal take-off and landing trajectories and procedures for airports.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

1.1 Motivation

According to statistical data of civil aviation (including commercial airline and general aviation) in the United States in the year 2008, an average of 92 touch-downs happened in each minute nationwide, which summed up to more than 48 million touch-downs throughout the whole year [2]. Considering the large number of landing operations and the low fuel efficiency in the current "stair case" descend phase, the implementation of optimal landing trajectories is expected to substantially improve the operational efficiency and the greenhouse gas footprint of current aviation systems.

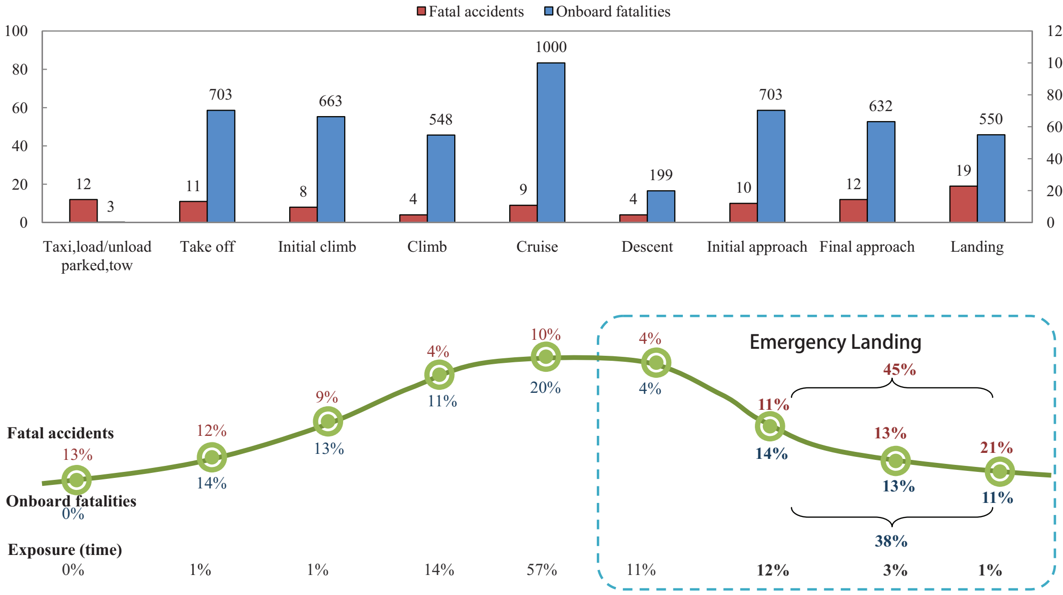

Figure 1: Statistical summary of commercial jet airplane accidents.

In addition to the obvious economic and environmental benefits, another strong motivation for flying optimized aircraft trajectories is the potential for safety record improvement in case of emergency landing. Figure 1 shows the distribution of fatal accidents and onboard fatalities worldwide according to the phase of flight from 2000 to 2009 [7] (the exposure as the percentage of flight time for different phases is estimated for an 1.5 hour flight). As shown in the figure, about 38% of onboard fatalities happen during the approach and landing phases, which is a considerable portion, especially considering the relatively short time span spent in those phases. Furthermore, when an accident happens during other flight phases such as during climb or cruise, an emergency landing procedure should also be initiated shortly as long as the aircraft does not completely lose its maneuverability. Therefore, effective automation aids for emergency landing process optimization can provide valuable help such that the pilot can provide fast and proper response to accidents in all flight phases.

Reference [7] records a total of 89 fatal accidents between 2000 and 2009, with 58% of those accidents categorized as loss-of-control in flight (including the engine thrust), controlled flight into terrain, runway excursion, undershoot and overshoot, etc, which are related to inadequate or inappropriate control inputs to the aircraft. Some of these accidents may not have been fatal should the pilots had been able to quickly plan and execute a safe landing trajectory by implementing appropriate control inputs.

One of the fatal accidents that could possibly have been remedied by pursuing a timely generated optimal landing trajectory is the case of Swissair flight 111, which was on a scheduled flight from JFK, New York City to Geneva, Switzerland on September 2, 1998, and crashed en route near Halifax after an infight fire accident. The pilots were not able to plan and execute a proper landing trajectory during the very short time window in which the initiation of an emergency descent could possibly have saved the aircraft.

A recent inspiring success story of an emergency landing is the US Airways 1549's crash-landing into Hudson River, on January 15, 2009. The Airbus A320 aircraft lost thrust in both engines during its climbing phase after encountering and striking a flock of birds. The captain, who happened to be an experienced glider pilot, successfully landed the aircraft on the Hudson river without a single casualty.

As demonstrated by these incidents, the emergency landing scenario requires (but is not limited to) the evaluation of the aircraft performance, the selection of a landing site, the fast construction of a feasible (at least close to) optimal landing trajectory, and the execution of such a trajectory. These tasks require intensive computations, comparison, and evaluation of various alternative plans, and must be accomplished within a very limited time. Such tasks can be processed effectively by automation tools with efficient and robust trajectory optimization algorithms.

This thesis focuses on the problem of efficient and robust aircraft landing trajectory planning having as the motivation of future onboard avionics implementation for more efficient flight and safer landing (especially during emergencies). This work fits into NASA's resilient aircraft emergency planning architecture with integrated trajectory planning, as shown in Fig. 2, and aims to function as a core component in the Intelligent Flight Planning and Guidance module in the Flight Management System (FMS).

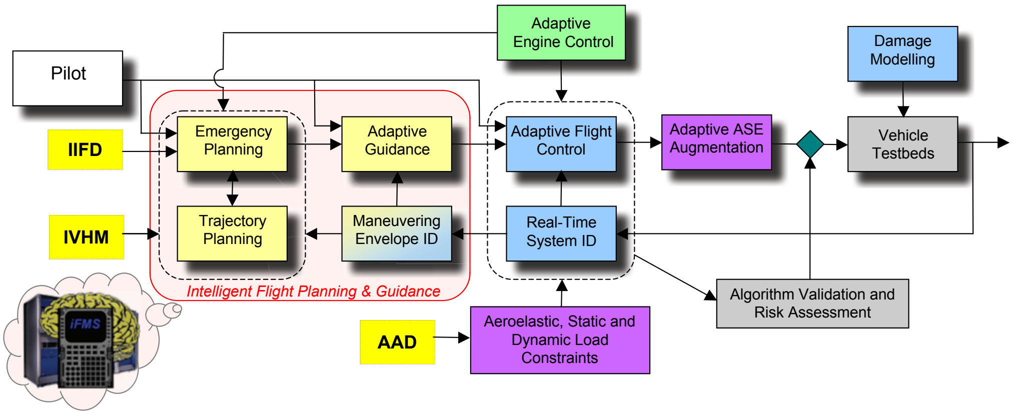

Figure 2: NASA Resilient Aircraft Emergency Planning Architecture.

1.2 Problem Statement

In this section we will state the problems addressed in this dissertation. Before proceeding, it is convenient to distinguish between the words *curve/path* and *trajectory*, which are used throughout this thesis. A *curve/path* is a purely geometrical concept and consists of a one-dimensional continuum of points. A *trajectory* is a curve/path along which the coordinates are given as functions of the time [126].

In this thesis we consider the movement of an aircraft in a three dimensional *geometric space* $\mathcal{G} \subseteq \mathbb{R}^3$. The set $\mathcal{O} \subset \mathcal{G}$ represents the collection of obstacles. The full state x of the aircraft, which completely describe the configuration and instantaneous movement of the aircraft, belongs to a *state space*, which is denoted by $\mathcal{X}$. Let x_G denote the components of

x which belong to the geometric space $\mathcal{G}$.

The time evolution of the state variable $x(t)$ depends on the initial condition $x(t_0) = x_0 \in \mathcal{X}$ at the start time $t_0 \in \mathbb{R}$, and is affected by the control input $u \in \mathcal{U}$, as described by a set of ordinary differential equations as follows

$$\dot{x}(t) = f(x(t), u(t)), \tag{1}$$

where $t \in [t_0, t_f] \subseteq \mathbb{R}$, $\mathcal{U}$ is the set of admissible controls, and f is a sufficiently smooth vector-valued function, such that there is a unique solution to the previous set of differential equations.

In is often required that the state x and control u must satisfy certain constraints representing the flight envelop of the aircraft, such as load factor constraint, speed constraint, etc. These constraints are typically enforced as algebraic, and pointwise-in-time constraints, in the form

$$C(x(t), u(t)) \leq 0, \quad t \in [t_0, t_f]$$

where C is a real vector-valued function and the inequality is enforced component-wise.

Problem 1.2.1 (Feasible Trajectory Generation)**.** Given the initial and final conditions x_0, $x_f \in \mathcal{X}$, the initial time $t_0 \in \mathbb{R}$, determine the final time t_f, the control input $u(t) \in \mathcal{U}$ and the corresponding state history $x(t)$ for $t \in [t_0, t_f]$ such that

1. $x(t_0) = x_0$ and $x(t_f) = x_f$, and
2. for all $t \in [t_0, t_f]$,

$$\dot{x}(t) = f(x(t), u(t)), \tag{2}$$

$$C(x(t), u(t)) \leq 0, \tag{3}$$

$$x_{\mathcal{G}}(t) \in \mathcal{G} \setminus \mathcal{O}. \tag{4}$$

Sometimes it is desirable that the generated trajectory is not only feasible, but also has good performance, which can be measured by a certain functional of the state and control variables, denoted by

$$J(x, u) = \int_{t_0}^{t_f} \mathcal{L}(x(t), u(t)) \, \mathrm{d}t. \tag{5}$$

Hence, we have the following trajectory optimization problem:

Problem 1.2.2 (Trajectory Optimization)**.** Given the initial and final conditions x_0, $x_f \in \mathcal{X}$, initial time $t_0 \in \mathbb{R}$, determine the final time t_f, the control input $u(t) \in \mathcal{U}$ and the corresponding state history $x(t)$ for $t \in [t_0, t_f]$ which minimize the cost function $J(x(\cdot), u(\cdot))$ and satisfy

1. $x(t_0) = x_0$ and $x(t_f) = x_f$, and
2. For any $t \in [t_0, t_f]$,

$$\begin{aligned}
&\dot{x}(t) = f(x(t), u(t)),\\
&C(x(t), u(t)) \leq 0,\\
&x_{\mathcal{G}}(t) \in \mathcal{G} \setminus \mathcal{O}.
\end{aligned}$$

Solving Problem 1.2.2 is not an easy task. More often than not, the required amout of computations prohibit any attempt to solve Problem 1.2.2 in real-time. Thus, for real-time applications it may be more practical to accept a reasonably suboptimal trajectory. For many physical systems, the task of trajectory generation and optimization can be decomposed into two layers: the geometric layer, and the dynamics layer. It is possible to find a suboptimal solution to Problem 1.2.2 by solving Problem 1.2.1 on the geometric layer and the dynamics layer separately with certain (possibly heuristic) consideration of optimality on each layer. The planning result on each individual layer can help improve the performance of the final trajectory. Such an approach can be classified as *hierarchical motion planning*, which will be briefly discussed in the next section. Here we define two optimization problems which can be applied to suboptimal trajectory generation using a hierarchical approach.

For many non-holonomic systems such as car and aircraft, the path geometry has critical influence on the feasibility and performance of path tracking. For example, a discontinuity in the derivative of the path may correspond to a sudden change of the speed vector, which would render the path infeasible (no admissible control inputs exist for following such a path exactly). Besides, for two paths with the same length, and the same initial and final positions, better tracking performance can usually be achieved with the smoother path [41, 139]. One way of improving the smoothness of a path is to solve the following variational problem, which regulates the curvature of the path:

Problem 1.2.3 (Path Smoothing)**.** Let $\mathbf{r}(s) \in \mathcal{G} \setminus \mathcal{O}$ be a collision-free path parameterized by its path length $s \in [s_0, s_f] \subset \mathbb{R}$. Consider a variation $\delta\mathbf{r}$ of $\mathbf{r}$, and denote the new perturbed path by $\tilde{\mathbf{r}}$. Let $\tilde{\kappa}(s)$ be the curvature of the perturbed path $\tilde{\mathbf{r}}$ at s. Let $w : [s_0, s_f] \mapsto \mathbb{R}_+$ be a weight function. Find the variation $\delta\mathbf{r}$ such that

1. (Collision Avoidance) The perturbed path $\tilde{\mathbf{r}}(s) = \mathbf{r}(s) + \delta\mathbf{r}(s) \subset \mathcal{G} \setminus \mathcal{O}$, for any $s \in [s_0, s_f]$,

2. (Boundary Conditions) $\tilde{\mathbf{r}}(s_0) = \mathbf{r}(s_0)$, $\tilde{\mathbf{r}}(s_f) = \mathbf{r}(s_f)$,

3. (Local Curvature Constraint) $\kappa_{\min}(s) \leq \tilde{\kappa} \leq \kappa_{\max}(s)$, where $\kappa_{\min}(s)$ and $\kappa_{\max}(s)$ are specified bounds on curvature at s, and

4. The following integral is minimized

$$\int_{s_0}^{s_f} w(s)\tilde{\kappa}^2(s)\mathrm{d}s \tag{6}$$

Problem 1.2.3 is a purely geometric problem. The dynamics of the system is not addressed in the process of solving Problem 1.2.3, although it is expected that the minimization of (6) will result in a reasonably good tracking performance, measured by (5).

The following optimization problem is on the dynamics layer with fixed path geometry:

Problem 1.2.4 (Optimal Time Parameterization/Tracking of a Geometric Path)**.** Given the initial and final conditions x_0, $x_f \in \mathcal{X}$, initial time $t_0 \in \mathbb{R}$, let $\mathbf{r}(s) \in \mathcal{G}\setminus\mathcal{O}$ be a geometric path parameterized by its path length $s \in [s_0, s_f] \subset \mathbb{R}$. Determine the final time t_f, the control input $u(t)$ and the corresponding state history $x(t)$ for all $t \in [t_0, t_f]$ that minimize the cost function $J(x, u)$ and satisfy

1. (Boundary Conditions) $x(t_0) = x_0$ and $x(t_f) = x_f$, and

2. (Dynamics and Path Constraints) for any $t \in [t_0, t_f]$,

$$\dot{x}(t) = f(x(t), u(t)),$$
$$C(x(t), u(t)) \leq 0,$$

3. (Path Tracking) There exists a strictly monotone mapping $\nu : [s_0, s_f] \mapsto [t_0, t_f]$ with $\nu(s_0) = t_0$ and $\nu(s_f) = t_f$ such that $x_{\mathcal{G}}(\nu(s)) = \mathbf{r}(s)$, $s \in [s_0, s_f]$.

1.3 Review of Trajectory Optimization Techniques

Problem 1.2.2 can be addressed either as an *optimal control* problem, or a *motion planning* problem. Roughly speaking, motion planning methods can easily deal with complex geometric constraints, such as obstacles, but their capability for dealing with vehicle dynamics is less developed, as compared to the optimal control approach. Other techniques such as differential flatness and hybrid/hieararchical methods can also be applied to trajectory generation and optimization. Next, we go through a brief review of these trajectory optimization techniques.

1.3.1 Motion Planning

Motion planning is a term commonly used in robotics and artificial intelligence, referring frequently to the planning of the motion of a robot in a two or three dimensional space containing obstacles [79]. Specifically, motion planning means the generation of a plan for moving a system from one location to another location in the configuration space to accomplish a task, while avoiding collisions with obstacles or other undesirable behaviors. It may also be required that the plan makes efficient use of the available resources to achieve the goal by optimizing a certain cost [50].

Early motion planners did not take dynamics into account. Instead, they only considered the geometry of the robot and the obstacles. A considerable amount of effort in motion planning had thus been devoted to facilitating the representation of geometry such that the enforcement of geometric constraints becomes relatively easy. The most well-known motion planning algorithms include cell decomposition methods, roadmap methods (visibility graph [82], Voronoi diagrams [11]), and artificial potential field methods [16, 39]. It is also noted that graph search is a common tool used in many motion planning methods developed in robotics and artificial intelligence [79, 93].

Because the vehicle dynamics are not taken into account in these path planning methods which consider only the geometric constraints, it is sometimes difficult, or even impossible, to force the vehicle to follow the generated path. For example, visibility graph search produces the shortest distance path connecting the starting and end points in an environment. Such a path usually contains corners, which makes it impossible to be followed by, say, a fixed-wing aircraft because the differential constraint stemming from the vehicle dynamics requires however the path to be continuously differentiable at least to the second order. Even if some smoothness technique is used to eliminate the corners, while preserving the collision avoidance of the path, there is still no guarantee for the successful execution of the path because the stall limit of the aircraft also imposes constraints on the curvature of the path. A "forced" implementation of the path may lead to deviation from the reference path and may even cause a collision with the obstacles or result in the stall of the aircraft.

Even if the path is feasible in terms of system dynamics, the performance could severely deteriorate by following such a path. For example, a helicopter can follow any piecewise linear path, but the frequent acceleration, deceleration and hovering associated with a zigzag path would result in considerable waste of fuel and prolonged flight time, which may render such a path practically undesirable.

Vehicle dynamics usually impose complicated constraints on the problem and may result in extensive computations. To circumvent the computational complexity of deterministic algorithms, randomized algorithms have been developed [79], including Probabilistic Road Map (PRM) [72] and Rapidly-exploring Random Trees [80], etc. Because of the substantial reduction of workload in randomized motion planning algorithms, it is possible to extract dynamically feasible trajectory segments from the configuration space, while maintaining collision avoidance. Those randomized algorithms, if converge, could provide dynamically feasible trajectories.

Very few motion planning techniques have been directly applied to the trajectory optimization of aircraft because of the complicated nonlinear dynamics constraints required.

One work in this category is the motion planning for small autonomous helicopters in [50, 51], where an RRT algorithm is used. The optimality of sampling algorithms is studied in [69], which shows that the cost of the returned path converges to the best value almost surely.

1.3.2 Differential Flatness and Inverse Dynamics

The concept of differential flatness was introduced in [47, 104]. Roughly speaking, a controlled system with equations as in (1), $t \in \mathbb{R}_+$, $\mathcal{X} = \mathbb{R}^n$, $\mathcal{U} = \mathbb{R}^m$, is said to be differentially flat if the following conditions are satisfied:

1. There exists a finite set $y = (y_1, \dots, y_m) \in \mathbb{R}^m$ of variables which are differentially independent, i.e., are not related by any differential equations.

2. The y_i $(i = 1, \dots, m)$, are differential functions of the system variables, i.e., are functions of system variables (state x, control u) and a finite number of their derivatives.

3. Any system variable is a differential function of y_i and a finite number of their derivatives.

The output $y = (y_1, \dots, y_m)$ is called a *flat* or *linearizing* output.

Similar concepts may be traced back to the invertibility of nonlinear systems [58], in which a specific type of nonlinear system of the form (affine in control)

$$\begin{aligned}
\dot{x}(t) &= A(x(t)) + \sum_{i=1}^{m} u_i(t) B_i(x(t)); \quad x(0) = x_0 \\
y(t) &= c\left(x(t), u(t)\right),
\end{aligned} \tag{7}$$

is considered, where $t \in \mathbb{R}_+$, $x(t) \in \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$, $A : \mathbb{R}^n \to \mathbb{R}^n$, $B : \mathbb{R}^n \to \mathbb{R}^n$, $u(t) = (u_1(t), \dots, u_l(t))^T \in \mathbb{R}^l$. Let $y(t; u, x_0)$ denote the output of system (7) with initial condition $x(0) = x_0$ and control input u. This nonlinear system is *invertible* at x_0 if whenever u and $\hat{u}$ are distinct controls, the corresponding outputs $y(\cdot; u, x_0) \neq y(\cdot; \hat{u}, x_0)$. Thus, given the output $y(\cdot)$ for a system which is invertible at x_0, the corresponding control can be uniquely determined. In the context of differential flatness, the output y of system (7) is essentially the flat output of the system, from which the state x and control u can be recovered.

If a system is differentially flat, then the problem of trajectory planning can be solved by finding the history of the outputs satisfying the initial and final conditions and the state and control constrains which are mapped into this flat output space. Once the desired output history is found, the history of the state and control variables can be directly recovered without solving the equations of motion. Therefore, the main difficulty with trajectory planning for a differentially flat system is the identification of the output history which satisfies the state and control constraints.

In practice, the flat output is usually chosen from a finite-dimensional functional space spanned by a finite number of pre-determined basis functions Φ_j, as follows:

$$y\left(t\right)=\sum_{j=1}^{k}\alpha_j\Phi_j\left(t\right).$$

Usually polynomial functions are selected as the basis functions Φ_j for the ease of function evaluation and the computation of derivatives. Once the function basis for the desired output is determined, the next step is to choose the coefficients α_j such that all state and control constraints are satisfied. Note that the state and control constraints need to be satisfied for each $t \in [t_0, t_f]$, which indicates that the trajectory planning problem for a differentially flat system can be interpreted as a *semi-infinite programming* (SIP) problem, i.e., one such that there exists a finite number of decision variables and an infinite number of constraints. Some work on trajectory generation and optimization of differentially flat system under semi-infinite programming formulation can be found in [92, 46, 57].

The work in [46] is especially interesting because of its idea to divide the task of trajectory generation into two parts: an off-line part and an online part. The time-consuming SIP problem is performed off-line by finding the maximum volume polytope which satisfies all the constraints. Then the online part generates a feasible trajectory very fast using some point inside this polytope. Although not studied in [46], it is naturally expected that this pre-computation approach could speed up the online optimization to yield a suboptimal solution. It needs to be noted that [46] bypassed the difficult problem of solving the SIP involving the time as the parameter for the infinite-dimensional constraints by introducing collocation, and solved only the SIP for the polytopic approximation of the nonlinear constraints, which involve only the algebraic constraints on the flat output and its derivatives.

How to identify a differentially flat dynamical system is still an open question. Currently, a dynamical system is considered differentially flat only if a set of flat outputs are found explicitly. Furthermore, even if a system is differentially flat, the trajectory planning of such a system, as described before, also depends on the selection of the basis functions. This could be difficult when the system dynamics and the constraints are complicated. Despite these limitations, differential flatness is still an attractive approach for computationally efficient trajectory generation, where a set of flat outputs is available.

The inverse dynamics technique is also frequently used for trajectory planning [133, 42, 43]. The dynamic systems considered in these references are essentially differential flat systems with the position variables as the flat output space. As a result, dynamics, state, and control constraints can be converted to nonlinear algebraic constraints on the discretized position variables using the flatness property, and the trajectory generation problem becomes a Nonlinear Programming problem with discretized position as decision variables.

In this thesis the inverse dynamics technique is used to recover optimal control inputs after the optimal speed of the aircraft dynamics is solved semi-analytically.

1.3.3 Optimal Control and Mesh Refinement

As mentioned previously, Problem 1.2.2 can be formulated as an optimal control problem. As perhaps the most rigorous and general method for solving trajectory optimization problems, optimal control is historically rooted in the *Calculus of Variations*, which started in the 17$^{\text{th}}$ century with two famous problems: Newton's problem of Minimum Drag Revolution Surface in a Resisting Medium, and Bernoulli's Brachistochrone problem [77]. Detailed introduction on this subject can be found in [9, 75, 29]. Briefly speaking, the optimal control input is obtained by minimizing the Hamiltonian, according to the Pontryagin's Maximum Principle (PMP).

Numerous trajectory planning problems have been solved using the optimal control approach since the 60's, such as [73, 35, 32, 37, 56, 113, 112, 71, 70, 83, 30, 31], just to name a few. In an optimal control problem, the dynamics of the vehicle are explicitly formulated as part of the problem, hence the optimal control solution is naturally feasible in terms of dynamics, and optimality is also guaranteed, at least locally.

Solving an optimal control problem is often difficult. The optimal control formulation of a trajectory optimization problem using PMP leads to a Two-point Boundary Value Problem (TBVP), or a Multi-point Boundary Value Problem (MBVP) when the optimal trajectory is composed of multiple phases. For general constrained and nonlinear systems, TBVP and MBVP are very difficult to solve analytically, and numerical methods such as shooting method and multiple shooting methods are applied for solving TBVP and MBVP.

Another approach equivalent to the PMP is Bellman's Principal of Optimality, which is also the fundamental theorem behind *Dynamic Programming* [19]. When applied to solving optimal control problems, Bellman's Principal results in the Hamilton-Jacobi-Bellman (HJB) equation, which is a partial differential equation. Analytic solutions to (HJB) are also very difficult to obtain.

The difficulty associated with obtaining an analytic solution to general optimal control problems, either via PMP or HJB, necessitates the development of numerical methods. Numerical optimal control methods fall into two categories: indirect methods and direct methods (numerical methods for solving the HJB partial differential equation are out of the scope of this thesis). The major difference between direct and indirect methods is that the former do not involve the costate variables and necessary conditions for optimality during the optimization process. The shooting methods belong to the indirect methods. Although shooting methods tend to be more accurate, their convergence is more sensitive to the initial guess compared to direct methods [25]. One example of a numerical optimal control software using an indirect method is BNDSCO, which implements the multiple shooting method [94].

The main idea of direct methods is to discretize the original continuous-time optimal control problem into a finite-dimensional nonlinear programming problem (NLP), and minimize directly a discrete version of the objective function of the original continuous-time optimal control problem. The solution of this NLP, which consists of discrete variables, is used to approximate the continuous control and state time histories. A recent survey of numerical optimal control techniques for trajectory optimization can be found in [22]. Although the convergence of direct methods are not guaranteed because the problem itself is not convex in general, in practice it has been found that many problems can be solved using direct methods [86, 135].

Many numerical optimal control software packages have developed based on the direct methods, with SOCS [23], RIOTS [110], DIDO [103], PSOPT [17], GPOPS [101] as a few examples. The density function based mesh refinement algorithm (DENMRA) proposed in this thesis, which is used for solving numerical optimal control problem, is also taking a direct approach.

Differential Dynamic Programming (DDP) [64] is another approach that can be applied to optimal motion planning. It is based on Bellman's Principle of Optimality, and uses successive backward quadratic expansions of the objective function. DDP has been applied successfully to spacecraft trajectory planning for orbit transfer [78].

Direct methods rely on a certain grid for the discretization of the original control problem. It is well known that increasing the number of points in the grid can help improve the accuracy of the solution. However, it has also been observed that the solution accuracy obtained using a uniform grid is often not acceptable even with a very large mesh size. Such an observation has motivated recent research in mesh refinement algorithms for solving optimal control problems [110, 26, 66, 55].

There are two general methods for mesh refinement: static and dynamic [99]. In static refinements, after a solution of the discretized problem is computed, the same solution is used to refine the current mesh, by adding and/or moving points around using a certain strategy, so that the accuracy of the solution is improved in the next optimization step. In dynamic mesh refinements, some (or all) of the grid points are included as decision variables and allowed to move during optimization. Although dynamic mesh refinements may capture control discontinuities early on, they may also hinder convergence. They can also be less efficient than static mesh refinement strategies [110]. Several static mesh refinement strategies are proposed in Ref. [110]. Reference [26] also introduced a static mesh refinement method in which integer programming is used to minimize the maximum integration error during the mesh refinement iterations. Reference [67] proposed a multi-resolution trajectory optimization algorithm (MTOA) that refines a nonuniform mesh using local diadic partitioning after each iteration based on the interpolation error.

The mesh refinement methods in Refs. [110, 26, 66] use the integration/interpolation error to distribute/add the grid points, and are not the best choice for some problems. In particular, they contain no mechanism for directly placing mesh points at or near locations where control and/or trajectory constraints switch from being active to being inactive or vice versa. These are precisely the locations where the solutions are likely to be least accurate [110]. In these mesh refinement methods, extra mesh points near the location where such a constraint switch is likely to occur as suggested by local integration error, but this approach reduces the efficiency of optimization by introducing too many unnecessary points into the mesh. Hence, for better accuracy, it is necessary to somehow estimate the location of continuity irregularities and incorporate this information into the mesh refinement process, rather than using just the integration/interpolation error alone. A mesh refinement method following this philosophy has been proposed in [55]. It divides the time interval at the points with maximum absolute value of the first derivative of the control, but it does not capture higher order irregularities in the control.

Mesh generation and adaptation is a common problem in engineering and applied mathematics. As a result, similar concepts and methods have been developed in many areas. For instance, monitor functions control mesh concentration based on the equidistribution of their integral [18], and are used for the adaptive mesh generation for the numerical solution of PDEs. The notion of mesh density function was first introduced in the Finite Element Analysis field in Ref. [12], and further explored by Hagger [62]. The key idea is to represent a mesh such that the total amount of density in each element of the mesh is the same. Hence, in terms of mesh point allocation, mesh density functions and monitor functions are inherently the same. The idea of monitor functions has actually been used for the initialization of direct methods for solving optimal control problems based on some reasonable initial guesses [24]. As shown in the same reference, although different monitor functions can be used for mesh generation, an appropriate choice of monitor function can generate a mesh with better quality, thus improving the accuracy and speed of the algorithm. Hence, the problem of mesh generation is converted to a problem of finding an appropriate monitor (or density) function. The monitor function based mesh generation approach in [24] requires the selection of a numerical integration scheme and a reasonable initial guess, both of which are problem-dependent and require human interaction, hence, it is not suitable for fully automatic mesh refinement.

1.3.4 The Hierarchical Approach: Path Smoothing and Optimal Path Tracking

Because solving Problem 1.2.2 directly is usually too difficult or computationally intractable, especially for real-time applications, a hierarchical approach is sometimes adopted to find a feasible solution to Problem 1.2.1, which is close to the optimal solution of Problem 1.2.2 [115, 20, 27]. Hierarchical motion planning methods are sometimes called *hybrid methods* in the literature [50, 51, 98].

A typical hierarchical motion planning process decomposes the task of motion planning into subproblems on several levels [20]. In the first level, the environment is analyzed and represented usually in the form of a graph. The requirement of collision avoidance can be accomplished by properly determining the connectedness of different vertices in the graph that represents the environment. In the second level, a path is chosen from the graph which connects the desired start and end points. Usually, certain criteria are used for choosing such a path, such as shortest distance. In the last level, a trajectory is generated based on the path in the previous level, and is used as a reference trajectory for the vehicle/robot's tracking controller such that the trajectory can be actually executed. More abstract layers such as the *strategic* layer or the *tactical* layer may also be used on top of these three levels to introduce certain degree of "intelligence" into the motion planner and facilitate the planning process [50].

Reference [98] contains an example of a hierarchical motion planning algorithm. This work presents a synergistic combination of layers of planning (SyCLoP) such that a discrete searching process is performed on a high level while a sampling-based motion planning routine runs on the lower level. The higher level discrete search provides important strategic guidelines for the sampling-based motion planning algorithm, which also provides feedback to the discrete search in return.

The path smoothing method and optimal time parameterization method proposed in this thesis may work together in a hierarchical manner to produce feasible trajectories efficiently with acceptable optimality. Specifically, the path smoothing method works on the geometric layer, and the optimal time parameterization method, or, equivalently, the optimal path tracking method, ensures feasibility and exploits the optimality on the dynamics layer.

Path Smoothing

A discontinuity in the curvature profile of the path to be followed implies an instantaneous change of the steering wheel angle for a car-like vehicle or the bank angle/angle of attack for a fixed-wing aircraft, both of which require (theoretically) infinite control force. Therefore, the curvature of the path should be at least continuous for most practical applications. For this reason clothoid arcs have been used for continuous-curvature path planning based on the Dubins' path prototype [108, 48, 15]. Reference [97] used analytical splines and heuristics for smooth path generation. Reference [134] proposed a path planning algorithm which generates a smooth path by smoothing out the corners of a linear path prototype using Bézier curves based on analytic expressions. Although all these methods can generate paths with continuous curvature, obstacle avoidance is not guaranteed by these methods per se, and can only be done in an ad hoc manner.

One approach for smooth path planning in the presence of obstacles is to use a "channel" or "corridor," which is selected a priori, such that it does not intrude any of the obstacles. A smooth path is then found within the channel such that it is collision-free. For instance, Ref. [14] introduced a method for generating curvature-bounded paths in rectangular channels; reference [21] proposed a method for constructing bounded curvature paths traversing a constant width region in the plane, called corridors, and Ref. [68] introduced a method for generating smooth two-dimensional paths within two-dimensional bounding envelops using B-spline curves. A nonlinear optimization scheme is used to design collision-free and curvature-continuous paths in [85]. Because the channels are fixed, the results of these algorithms are limited by the collision-free channels which are chosen conservatively before the planning.

In this thesis, the path smoothing problem is formulated as Problem 1.2.3. The smoothness of the path is improved by minimizing the weighted L_2 norm of the path's curvature. It will be shown later in the thesis that a smoothed path may provide better tracking performance, such as tracking time, compared the original path.

Optimal Path Tracking

As one of the major problems considered in this thesis, Problem 1.2.4 seeks an optimal solution for tracking a prescribed geometric path subject to dynamics, state, and control constraints. Problem 1.2.4 shares the same cost function as the trajectory optimization problem, hence provides a tracking scheme with the best tracking performance for the given path geometry.

The optimal path tracking problem has been studied extensively in the literature. The minimum-time robotic manipulator and car path tracking problems are studied in [28, 119, 96, 118, 116, 129] for shortest travel time along the path subject to control and state constraints. The optimal solutions to these problems can help improve plant productivity [28, 119, 96, 118, 116], racing car performance [129], or faster aircraft landing as will be shown later in this thesis. These solutions correspond to the point-wise maximization of the speed along the path without any singular arcs[1].

When the tracking time is not of prime concern, it is often desirable to minimize the energy/fuel consumption of the system. Along this direction, the minimum work train operation problem has been studied [8, 61, 74, 59]. Unlike minimum-time problems, the minimum-work solutions usually contain singular arcs. When the travel time is free, the singular arc can be determined analytically. In the more practical case of fixed travel time

[1] The "singular arcs" in [116] actually refer to segments of speed profile with active speed constraints, which are different from the term's traditional meaning used in optimal control.

for scheduled operations [8, 61, 74, 59], the singular arc cannot be determined directly, and an iterative numerical procedure must be used to choose the appropriate singular arc with which the desired travel time and boundary conditions can be satisfied. Because the cost function for the minimum-work problem is not strictly convex, the optimal control approach as in [8, 61, 74, 59] can provide more reliable and accurate information about the singular arcs in the optimal solution than the numerical optimization approach as in [36, 25, 52, 130].

It is noted that, although originated from different physical systems, the path tracking methods as in Refs [8, 61, 74, 59, 28, 119, 96, 118, 116, 117, 129] involve the same key steps by which a scalar functional optimization problem is solved. Specifically, the point-mass train model has only one degree of freedom along the rail, hence the corresponding path tracking problem is naturally a speed optimization problem [8, 61, 74, 59]. Although the robot arm and car dynamics involve more than one state variables, the time parameterization problems for these systems can also be simplified to scalar functional optimization problems with state bounds [28, 119, 96, 118, 116, 117, 129].

In this thesis, we will solve Problem 1.2.4 with the aircraft dynamics with two different performance criteria: minimum-time, and minimum-energy with fixed Time Of Arrival (TOA).

1.4 Previous Research on Aircraft Trajectory Optimization

Next, we briefly review previous research on the optimization of aircraft trajectory. Most of those problems are formulated as optimal control problems and solved numerically.

1.4.1 Fuel and Range Optimization

Minimum-fuel optimization with fixed arrival time is studied in Ref. [32], where the author characterized the conflicts between optimal fuel consumption and required arrival time. Boeing aircraft including B737, B747 and B767 were considered. In Ref. [32], the aircraft model is simplified by introducing the energy state. The same problem was also studied in [56] for an F-4 type aircraft. Since the speed for maximum engine efficiency is usually different from the speed for minimum drag, the optimal speed of the aircraft has an oscillatory profile such that more engine power can be applied when the speed is beneficial for fuel efficiency. The problem is formulated as an optimal control problem using an aircraft model in the vertical plane, and appropriate boundary conditions are applied to enforce the periodicity of the solution. The corresponding boundary value problem is solved under the assumption that there are two throttle switches in each period. The effect of decreasing aircraft weight due to fuel consumption is also considered, and it was shown that such an effect is practically negligible.

The scenario of maximum-range trajectories for fixed flight time plays an important role in modern air combat scenarios. In [113, 112], Pontryagin's maximum principle is applied to determine range optimal trajectories for aircraft flying in the vertical plane. The considered aircraft model treats the energy, altitude and flight path angle as state variables with the load factor and throttle setting as the control variables. In addition, control limits and a dynamic pressure limit are imposed along the trajectory.

1.4.2 Performance Optimization

Although a high fidelity model captures more accurately the performance of the aircraft, it is difficult to use such a model directly for trajectory optimization. Noting that if an aircraft trajectory is given, it is relatively easy to propagate the "motion" inversely to the "control" using a high fidelity aircraft model even for complicated unconventional flight maneuvers [71], high fidelity aircraft models are useful for checking the feasibility of the trajectory during post-processing.

Reference [70] combined the inverse dynamics technique with optimal trajectory planning for a more robust near-optimal aircraft maneuver planning software. The trajectory planning was first solved using direct multiple shooting for a 3-DOF aircraft model, which is computationally tractable, yet sufficiently accurate for describing the translational dynamics of the vehicle. After the initial trajectory was obtained, an inverse simulation using a higher-fidelity 5-DOF aircraft model was employed to check the feasibility of the open-loop optimal control path obtained using the 3-DOF model. The comparison between the optimal and the inverse-simulated trajectories was performed visually, providing the information for the adjustment of a set of parameters affecting the computation of the trajectory. If the difference between the two trajectories remains within some specified tolerance, then the inverse-simulated trajectory is considered to be near optimal. Otherwise, the parameters affecting the optimization and inverse simulation are altered, and the same computations are repeated to obtain a modified trajectory. The authors also developed a software package in which the automatic solution of the near-optimal aircraft trajectory generation method was implemented.

1.4.3 Emergency Landing Trajectory Planning

Despite its importance, as discussed at the beginning of the introduction, not too much research has been done on the optimal landing problem. The abort landing problem in the presence of windshear has been studied in [30, 31]. The same problem is also studied in [90]. Note that in the physical space, the trajectory is occasionally represented as a four-dimensional flight path, following the tradition of air traffic control [37], with time as the fourth dimension in addition to the normally used three-dimensional representation of a path.

Reference [124] considers the generation of feasible trajectories using segments of trajectories corresponding to selected trim condition maneuvers (an equilibrium condition for the aircraft with constant speed, angle of attach, side slip angle, and angular velocity). A heuristic method is used to select a limited number of trim points covering a wide spectrum of flight conditions. The final landing trajectory is generated by searching and connecting the trim state trajectory segments such that the final position of the aircraft is close enough to the desired landing site. Note that the final trajectory as given by Ref. [124] may not be feasible at the junction points between different trajectory segments. A similar approach is used in Ref. [127] to study the emergency flight path planning problem for aircraft with left wing damage. LQR control has been used to generate the trajectory transiting the aircraft from one trim state to the other, hence, the generated trajectory is indeed feasible as long as the control constraints are not violated. The major problem with the approach in Ref. [124] is that the search results are limited to those that can be generated by connecting trim state trajectory segments with stable transitions. Because the unstable flight conditions are not considered in the search, the algorithm cannot identify any feasible trajectory containing unstable flight modes. Furthermore, the path length is used as the search criterion, which is less appropriate when compared to flight time for emergency landing, or fuel consumption for normal flight.

One of the common scenarios for emergency landing is the loss of thrust. Such a malfunction fundamentally changes the dynamics of the aircraft by turning it into a glider. The pilot not only needs to identify a reachable runway or landing site which meets the basic landing requirements for the specific type of aircraft, but he/she also needs to accurately steer and land the gliding aircraft to that runway or landing site. In this case, an onboard automation tool that optimizes and display the landing trajectory with a glider's dynamics would provide immediate assistance to the pilots' decision-making process [132].

Reference [10] studied the problem of emergency landing due to the loss-of-thrust using a hybrid approach. A two-step landing-site selection/trajectory generation process was adopted to generate safe emergency plans in real time under situations that require landing at an alternate airport. In the trajectory generation routine, a heuristic path planner was used to generate a three-dimensional trajectory connecting the current position of the aircraft to the runway, which consists of straight lines and circular arcs. This method is fast and simple. However, it has to stick to conservative aircraft maneuvers in order to reduce the chance of obtaining an infeasible trajectory. As a result, the optimality of the generated trajectory could be unacceptable for emergency landing, and further research is necessary to reduce such a conservatism.

1.4.4 Other Aircraft Trajectory Optimization Problems

The minimum-time, three-dimensional aircraft trajectory optimization prolem was considered in [109] by approximating the aircraft dynamics using an energy state to reduce the dimension of the problem for better convergence. This type of model reduction technique is commonly used for aircraft trajectory optimization [5]. Not surprisingly, trajectory planning problems have also been studied in the context of air traffic management (ATM) and automation. Reference [63] performed a sensitivity analysis of trajectory prediction for

ATM. The aircraft trajectory synthesis problem is studied in [120] to provide some basic tools for air traffic automation.

The aircraft terrain-following (TF) problem is analyzed in [83]. The TF problem is formulated as an optimal control problem that combines short flight time and path-following objectives. The analysis in [83] revealed that the optimal thrust profile is bang-bang in most cases. Inverse dynamics was employed to solve the problem numerically. Reference [84] studied the effect of nonlinear engine dynamics on the existence of singular arcs for a terrain-following aircraft. The result suggests that the usual practice of ignoring engine dynamics in aircraft trajectory optimization work does not lead to incorrect conclusions.

Reference [121] considered the generation of wind-optimal trajectory for cruising aircraft while avoiding the regions of airspace that facilitate persistent contrails formation. The shooting method is employed for solving the associated optimal control problem minimizing a weighted summation of flight time, fuel consumption, and a term penalizing the contrails formation. The aircraft dynamics considered in this reference is a simple kinematic model in the horizontal plane. The avoidance of the penalized region is achieved by tuning the corresponding weight factor in the cost function. The airspace avoidance problem is also considered in Ref. [65]. In this reference, the avoidance of restricted airspace is formulated as non-convex constraints in the optimization problem, and it is claimed that with a *feasible starting guess*, the efficiency of the optimization algorithm is not too degraded by the non-convex airspace constraints.

1.5 Thesis Outline and Statement of Contributions

Since both the feasibility and the optimality of the trajectory are critical for the emergency landing scenario, this thesis mainly takes the optimal control approach for landing trajectory generation. Due to the complexity of the aircraft dynamics, the optimal control formulation of Problem 1.2.2 cannot be solved analytically, therefore a numerical method is pursued to compute the optimal aircraft landing trajectory. Because time is the most critical factor in an emergency landing scenario [132], this thesis mainly focuses on the minimum-time landing problem, although a fuel-efficiency related problem is also discussed later in the thesis.

In this thesis, we also introduce several techniques which enable us to take a hierarchical trajectory generation approach to provide a set of high quality initial guess landing trajectory to facilitate the convergence of the numerical optimal control algorithm. On the other hand, because the trajectory generated using the hierarchical approach is mostly feasible, and the performance is usually acceptable, such a trajectory can be used as a back-up trajectory in case of the numerical optimal control algorithm failure, thus improving the robustness of the overall landing trajectory optimization algorithm.

Chapter 2 introduces a new mesh refinement method that utilizes a mesh density function for discretizing optimal control problems. The proposed method avoids the numerical integration step and the use of ODE solvers as in [24], and generates the mesh by the equidistribution of the integral of the selected density function, which is computed using the result of the previous iteration.

In Chapter 3 we introduce a path smoothing method that solves Problem 1.2.3 with a specific type of discretization scheme which ensures the convexity of the corresponding optimization problem.

In Chapter 4 we address the time-optimal path tracking problem for a fixed-wing aircraft. In this section we provide a semi-analytic method for solving the minimum-time landing problem along a prescribed geometric path. Numerical algorithms are also presented for solving these problems efficiently.

In Chapter 5, the energy-optimal path tracking problem for fixed-wing aircraft is considered. It is shown that the minimum-energy solution provides an approximation of fuel-optimal control during the landing process. The switching structure of the energy-optimal control is analyzed, and a numerical algorithm is designed for computing the energy-optimal solution.

In Chapter 6 we present a hierarchical scheme for integrating numerical optimal control, path smoothing, optimal path tracking and a geometric path planning method for the efficient and robust optimization of aircraft landing trajectory.

In Chapter 7, we analyze two emergency landing cases using the proposed landing trajectory optimization algorithm, and finally, in Chapter 8, we conclude the thesis and provide some ideas about future research directions in the area of aircraft emergency landing trajectory generation.

The main contributions of the thesis are:

1. A density function based mesh generation method for the discretization of optimal control problems. With such a method the problem of mesh refinement is converted into a problem of applying the appropriate density function. The density function provides a simple, yet effective, way for implementing and testing different mesh refinement schemes by choosing different density functions. It is shown that some of the previous mesh refinement schemes in the literature correspond to the choice of some particular density functions. With the density function technique, it is possible to refine the solution of the numerical optimal control problem iteratively without increasing the mesh size. A curvature density function is also proposed for mesh generation. Some good properties of such a density function are proved theoretically, and also verified by numerical simulations.

2. A reliable and computationally efficient new algorithm for the smoothing of a three-dimensional geometric path subject to a variety of constraints including collision avoidance, local curvature constraint, path length constraint, etc.

3. Original results on the time-optimal path tracking of fixed-wing aircraft. Theorems regarding the optimal switching structure between different extremals are provided. Two efficient numerical algorithms are developed for solving the time-optimal path tracking problem.

4. New results on the energy-optimal path tracking problem for fixed-wing aircraft. It is proved that the energy-optimal solution provides an approximation of the fuel-optimal solution during landing. New theoretical results regarding the optimal switching structure of the energy-optimal path tracking solution are established. A partial relaxation technique is introduced for identifying the state constrained arcs in the energy-optimal kinetic energy solution. A novel method is proposed which computes the energy-optimal

solution based on the minimum-time and the maximum-time solutions. A numerically efficient algorithm for solving the energy-optimal path tracking problems is proposed.

5. An efficient Dubins-like landing path planner, which produces near-optimal three-dimensional smooth landing path with continuous heading and path angles, and bounded path derivatives is proposed.

6. A robust and computationally efficient aircraft landing trajectory optimization approach, which integrates a hierarchical trajectory generation scheme and a numerical optimal control algorithm is developed. This hierarchical scheme incorporates a variety of path/trajectory planning tools including path smoothing, optimal path tracking, and various geometric path planning techniques, and is used to generate high quality initial guesses for the numerical optimal control algorithm to facilitate its convergence.

CHAPTER II

A MESH REFINEMENT METHOD USING DENSITY FUNCTIONS FOR SOLVING NUMERICAL OPTIMAL CONTROL PROBLEMS

This chapter introduces a simple, yet efficient, mesh generation method for solving optimal control problems. The method is based on density (or monitor) functions, which have been used extensively for the numerical solution of partial differential equations and in finite element methods [12, 18, 62]. Subsequently, the problem of mesh refinement is converted to a problem of finding an appropriate density function. We show that an appropriate choice of density function may help increase the accuracy of the solution and improve the numerical robustness.

2.1 Introduction

The accuracy and efficiency of mesh refinement algorithms that are used for solving numerical optimal control problems, have motivated a recent research activity in this area. Several mesh refinement methods are proposed in Ref. [110] demonstrating the advantage of such algorithms. Reference [26] introduced a mesh refinement method in which integer programming is used to minimize the maximum integration error during mesh refinement iterations. Reference [67] proposed a multi-resolution trajectory optimization algorithm (MTOA) that refines a nonuniform mesh using local dyadic subdivisions after each iteration. A common strategy behind these mesh refinement methods is the redistribution of the mesh points based on the local integration/interpolation error.

When the solution of the optimal control problem exhibits discontinuities in the control or its higher order derivatives, a locally dense mesh is typically necessary to achieve better resolution, and obtain more accurate estimation of the location of the discontinuity. Mesh generation based on the local integration/interpolation error does not incorporate any special treatment of the discontinuities, especially those appearing in higher order derivatives of the control or the state variables.

For better accuracy, it is necessary to estimate the location of such irregularities (namely, discontinuities in the control history and/or its higher order derivatives) and subsequently incorporate this information into the mesh refinement process.

A mesh refinement method following this philosophy has been proposed in Ref. [55]. This method divides the time interval at the points with maximum absolute value of the first derivative of the control, but it does not capture higher order discontinuities in the control time history.

Mesh generation and adaptation is a common topic in many areas of engineering and applied mathematics. The notion of mesh density function for mesh generation and refinement has been used in the FEM field [12, 62]. The concept of density functions is similar to monitor functions used for the numerical solution of PDEs [18]. However, despite their popularity in other fields, mesh density/monitor functions have rarely been used for discretizing optimal control problems. The only exception appears to be Ref. [24]. Additional studies are needed to understand how the density/monitor functions can be used in numerical optimal control and how they can influence the accuracy and robustness of numerical optimal control algorithms. Furthermore, the choice of "good" density/monitor functions for mesh discretization of optimal control problems seems to be open.

In this chapter we attempt to provide a partial answer to the previous questions. We introduce a method to distribute the mesh points efficiently using density/monitor functions. Although different monitor functions can be used for mesh generation, an appropriate choice of a monitor function can generate a better quality mesh, and can improve the accuracy of the solution, along with the speed of convergence. Hence, the problem of mesh generation can be treated as a problem of finding an appropriate density/monitor function. We propose two density functions which are computed based on the discrete control/state histories from the previous iteration during the mesh refinement process. The proposed method avoids the numerical integration step and the use of ODE solvers for the system dynamics as was done in [24]. Yet, it generates a mesh with a suitable level of adaptive discretization that provides sharp resolution around the places where the control switches or the trajectory meets/leaves state constraints, thus resulting in better accuracy of the overall final solution. Numerical examples are presented to demonstrate the advantage of the proposed method, and comparisons are provided against the industry standard Sparse Optimal Control Software (SOCS).

2.2 Problem Statement and Nonlinear Programming Formulation

We consider an optimal control problem minimizing the following Bolza cost functional

$$\mathcal{J} = \Phi(x(t_0), t_0, x(t_f), \mathbf{p}, t_f) + \int_{t_0}^{t_f} \mathcal{L}(x(t), u(t), \mathbf{p}, t)\mathrm{d}t, \tag{8}$$

where $t \in [t_0, t_f] \subseteq \mathbb{R}$ is the time, $x : [t_0, t_f] \to \mathbb{R}^n$ is the vector of state variables, $u : [t_0, t_f] \to \mathbb{R}^m$ is the vector of control variables, and $\mathbf{p} = [p_1, p_2, \ldots, p_l] \in \mathbb{R}^l$ the vector of additional optimization parameters. The Mayer term $\Phi : \mathbb{R}^n \times [t_0, t_f] \times \mathbb{R}^n \times \mathbb{R}^l \times [t_0, t_f] \to \mathbb{R}$, and the Lagrangian term $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \times [t_0, t_f] \to \mathbb{R}$ are given functions of suitable smoothness properties. Our objective is to minimize the cost (8) subject to the dynamic constraints

$$\dot{x}(t) = f\big(x(t), u(t), \mathbf{p}, t\big), \quad t_0 \leq t \leq t_f, \tag{9}$$

the boundary conditions

$$\Psi\big(x(t_0), t_0, x(t_f), t_f, \mathbf{p}\big) = 0, \tag{10}$$

and the path constraints

$$C\big(x(t), u(t), \mathbf{p}, t\big) \leq 0, \quad t_0 \leq t \leq t_f, \tag{11}$$

where $\Psi : \mathbb{R}^n \times [t_0, t_f] \times \mathbb{R}^n \times [t_0, t_f] \times \mathbb{R}^l \to \mathbb{R}^{N_\Psi}$ and $C : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \times [t_0, t_f] \to \mathbb{R}^{N_C}$.

To solve this problem through nonlinear programming, the states and controls are discretized on a mesh $\{t_i\}_{i=0}^N$ for some positive integer N, with $t_N = t_f$ and $t_i < t_{i+1}$ for $0 \leq i \leq N-1$. Let $\mathbf{X}$ be the vector of all decision variables, the corresponding discretization of the continuous time optimal control problem (8)–(11) can be written as

$$\min_{\mathbf{X}} J(\mathbf{X}) \tag{12}$$

subject to

$$|F(\mathbf{X})| \leq \zeta_d, \tag{13}$$

$$|\tilde{\Psi}(\mathbf{X})| \leq \zeta_b, \tag{14}$$

and

$$\tilde{C}(\mathbf{X}) \leq \zeta_C, \tag{15}$$

where the absolute value $|\cdot|$ and the inequalities are enforced element-wise, J, F, $\tilde{\Psi}$ and $\tilde{C}$ are appropriate discretizations of the cost function, dynamics constraint and path constraint of the original problem, respectively and $\zeta_d \in \mathbb{R}^{Nn}$, $\zeta_b \in \mathbb{R}^{N_\Psi}$ and $\zeta_C \in \mathbb{R}^{(N+1)\cdot N_C}$ represent defect vectors, whose elements are small positive real numbers. In particular, for the discretization of the differential constraint (9), the function J in (12) and F in (13) are obtained using a class of R-K methods ensuring consistency, such that the solution of the discrete problem converges to that of the continuous time problem[110]. For more details the reader may refer to Refs. [110, 135, 26, 66].

2.2.1 Density Function and Mesh Generation

A *mesh density function*, or simply *a density function*, is a non-negative function $\bar{f} : [a,\, b] \to \mathbb{R}_+$, $a, b \in \mathbb{R}$ that satisfies $\int_a^b \bar{f}(t)\,\mathrm{d}t = 1$, and is zero (at most) at countably many points. Since any non-negative function $f : [a,\, b] \to \mathbb{R}_+$ that has only countably many zeros can be normalized as

$$\bar{f}\,(t) = \frac{f\,(t)}{\int_a^b f\,(\tau)\,\mathrm{d}\tau}, \tag{16}$$

to obtain a mesh density function, from now on we may assume, without loss of generality, that any function f applied to mesh refinement has been already normalized.

The corresponding *cumulative distribution function* $F : [a,\, b] \to [0,\, 1]$ is defined by

$$F(t) \triangleq \int_a^t \bar{f}(\tau)\,\mathrm{d}\tau. \tag{17}$$

The value of $F(t)$ corresponds to the area below the graph of $\bar{f}$ between a and t. Clearly, $F(a) = 0$ and $F(b) = 1$. In the sequel, and without loss of generality, we will assume that $[a, b]$ is the unit interval. Consider a mesh $\{t_i\}_{i=0}^N$ containing a total of $N + 1$ points with $t_0 = 0$ and $t_N = 1$. Given a density function f, let F be the cumulative distribution function determined by f as in (17). For $i = 0, 1, \ldots, N - 1$, with the i^{th} point at t_i, the position of the $(i+1)^{\text{th}}$ point can be decided by

$$F(t_{i+1}) - F(t_i) = \frac{1}{N}. \tag{18}$$

A mesh can then be generated based on the density function f, such that the distribution of grid points conforms to an equidistribution of F. Alternatively, the mesh is dense where the value of $f(t)$ is large.

The previous mesh point allocation strategy usually requires solving a nonlinear algebraic equation repeatedly $N-1$ times, which can be a quite time-consuming task when N is large. An alternative technique for achieving equidistribution requires the integration of a system of ODEs, including the transformed dynamics and the inverse of the density function [24]. The integration of dynamics requires intensive computations, especially when the dimension of the problem is large. Besides, integration is also sensitive to the accuracy of the boundary conditions (if not fixed) and the accuracy of the control history obtained from the previous iteration.

To avoid the process of repeatedly solving nonlinear equations or integrating the system dynamics, an interpolation method is used in this work to compute the points $\{t_i\}_{i=1}^{N-1}$, by taking advantage of the monotonicity of F. Specifically, given any density function f, select a grid $\{t_j\}_{j=0}^{N_j} \in [0,\, 1]$, which contains N_j points. During the mesh refinement iterations, $\{t_j\}_{j=0}^{N_j}$ could be chosen as the mesh used in the previous iteration. Now $y_j = F(t_j)$ can be easily calculated by $y_j = \int_0^{t_j} f(\tau)\ \mathrm{d}\tau$. For any $y \in [0,1]$, define the inverse mapping $F^{-1}(y) = \{t | \int_0^t f(\tau)\ \mathrm{d}\tau = y\}$. From the properties of f, and hence F, the inverse F^{-1} is well defined and also continuous, with $t_j = F^{-1}(y_j)$. The set of pairs $\{(y_j, t_j)\}_{j=0}^{N_j}$ is then a discrete representation of the function F^{-1}. Note that the first and the last grid points are at $t_0 = 0$ and $t_N = 1$, respectively. For the allocation of the other grid points, the location t_i of the i^{th} mesh point can be obtained by interpolating $\{(y_j, t_j)\}_{j=1}^{N_j}$ using a spline function at the position $y_i = (i-1)/(N-1)$ for $2 \le i \le N-1$. Using this method, as long as the selected partition is dense enough, the location of all mesh points can be calculated very fast and with high accuracy. Note that the mesh point distribution is unique once the density function is given, but the converse is not true.

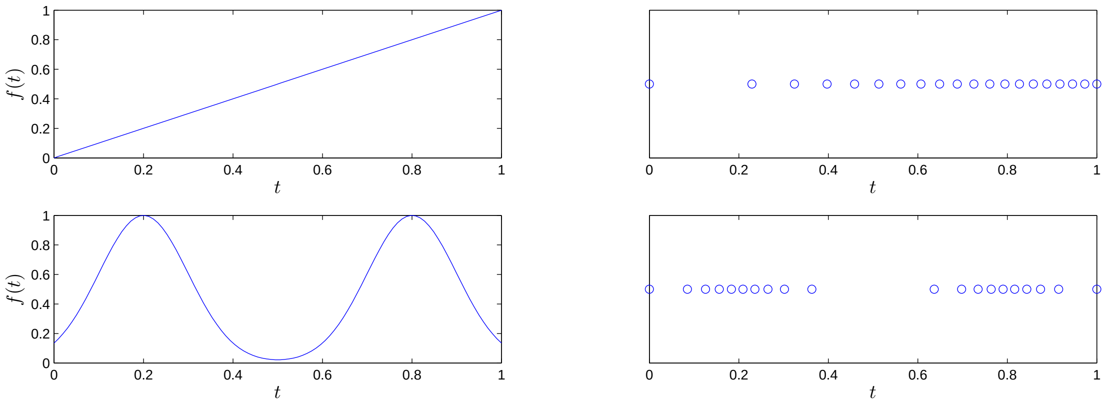

Figure 3: Density functions and corresponding distribution of grid points.

Figure 3 shows the mesh point distribution obtained by two specific density functions over the unit interval. The density function in the upper left of the figure is the linear function $f(t) = t$. The resulting mesh is shown in the upper right of the figure. The lower left plot shows the density function $f(t) = e^{-50t^2+20t-2} + e^{-50t^2+80t-32}$, with its mesh shown in the lower right of the figure. In both cases, the mesh contains a total of 20 grid points.

2.2.2 Selection of Density Function

By definition, a mesh density function needs only to be non-negative and integrable. This generality provides a great deal of flexibility for achieving desired mesh point distributions and for designing different mesh refinement schemes. The particular choice of the density function can have a major impact on the numerical performance of the overall algorithm.

Certain density functions can be used to regulate the integration error. For example, if the density function is chosen as a piecewise constant function whose value on each subinterval equals the corresponding *principal local truncation error function* (PLTE) as in Ref. [110], then the mesh point distribution process will be the static mesh refinement Strategy 1 introduced in the same reference. This strategy tries to approximately equidistribute the PLTE, and as a result, the mesh points would be denser where the PLTE was large in the previous iteration.

Another strategy for designing a good density function is to provide better approximation to the state and/or control histories to improve the accuracy of the solution. This approach places more emphasis on the geometric properties of the *graph of the function* to be approximated. The arc length monitor function in Ref. [24], for example, equidistributes the grid points along the graph of the state. As another example, the curvature-based density function proposed later provides the best piecewise linear interpolative approximation of the function of interest in the L_1 space. As it will be shown later, this density function is capable of capturing higher order discontinuities of the function to be approximated.

For more general mesh refinement schemes, it may be desirable to add new points only within certain specific time spans of the control and state histories, namely at those places where the control or state histories exhibit discontinuities or smoothness irregularities (e.g., very fast rate of change and/or discontinuities in higher order derivatives), while keeping other points fixed. This objective can also be easily achieved by defining multiple density functions on disjoint intervals; then the number of points assigned to each interval is proportional to the integral of the corresponding density function. The points are then distributed using the method introduced above. More details about this procedure are given in Ref. [136].

Although the density function uniquely determines the mesh once the total number of grid points is given, it does not provide any information what size of the mesh should be. In the density function-based mesh refinement algorithm proposed later, the discretization error estimation method in Ref. [26] is used to determine the size of the mesh in order to ensure that the new mesh provides a better discretization compared with the the one from the previous iteration.

2.3 A Density Function with the Best Piecewise Linear Interpolative Approximation of Piece-wise Smooth Planar Curves

We propose a density function that achieves best (in terms of the L^1-norm) piecewise linear approximation of C^3-smooth (at least piecewise C^3) curve. The main benefit of using the L^1 metric for measuring the approximation error is that the measurement corresponds to the area bounded by the curve and its approximation, which is invariant with respect to rotation. Hence, such a measure avoids the influence of the choice of coordinate orientation, and depends on the actual shape of the curve for its approximation.

Given an interval $\mathcal{I} = [t_a, t_b] \subset \mathbb{R}$, recall that a function $\Gamma : \mathcal{I} \to \mathbb{R}$ having piecewise second derivative implies that its intrinsic curvature is piecewise continuous and hence bounded. Recall that the curvature κ of Γ is given by:

$$\kappa(t) = \frac{|\Gamma''(t)|}{(1 + \Gamma'^2(t))^{3/2}},$$

where $\Gamma'' = \frac{\mathrm{d}^2\Gamma}{\mathrm{d}t^2}$ and $\Gamma' = \frac{\mathrm{d}\Gamma}{\mathrm{d}t}$. The natural coordinate s, defined by

$$s(t) = \int_{t_a}^{t} \left[1 + \Gamma'^2(t)\right]^{\frac{1}{2}} \mathrm{d}\, t,$$

is a measure of the length of the curve defined by the graph of Γ. Let $T_{\mathcal{I},N} = \{t_i\}_{1 \le i \le N}$ be a partition of the interval $\mathcal{I}$ using N points, where $t_a = t_1 < t_2 < \ldots < t_N = t_b$.

The function $\bar{\Gamma} : \mathcal{I} \to \mathbb{R}$ defined by

$$\bar{\Gamma}\,(t) = \Gamma(t_i) + \frac{t - t_i}{t_{i+1} - t_i}\left(\Gamma(t_{i+1}) - \Gamma(t_i)\right), \quad t \in [t_i, t_{i+1}),$$

where $t_i, t_{i+1} \in T_{\mathcal{I},N}$, $(1 \le i \le N-1)$, is a piecewise linear approximation of Γ on the interval $\mathcal{I}$ over the partition $T_{\mathcal{I},N}$.

With the density function ρ_Γ defined on $\mathcal{I}$, for any $N \ge 2$, the grid points denoted by $\{(t_i, \Gamma\,(t_i))\}_{i=1}^{N}$ are allocated on Γ such that $t_1 = t_a$, and

$$\int_{t_a}^{t_i} \rho_\Gamma\,(\tau)\,\mathrm{d}\tau = \frac{i-1}{N-1}. \tag{19}$$

Proposition 2.3.1. The best piecewise linear approximation of a function Γ with nonzero constant curvature κ using three points is obtained when the points are evenly distributed along the arc Γ.

Proof. The graph of a function with constant nonzero curvature is a circular arc, as shown in Fig. 4, with o denoting the center of the corresponding circle. The error in terms of the L_1-norm is exactly the shaded area shown in Fig. 4. The shaded area ξ_1 is given by

$$\xi_1(\theta_1) = \frac{\theta_1}{2}\kappa^{-2} - \kappa^{-1}\sin\left(\frac{\theta_1}{2}\right)\kappa^{-1}\cos\left(\frac{\theta_1}{2}\right) = \frac{\theta_1}{2}\kappa^{-2} - \frac{1}{2}\kappa^{-2}\sin\theta_1 = \frac{1}{2}\kappa^{-2}\left(\theta_1 - \sin\theta_1\right).$$

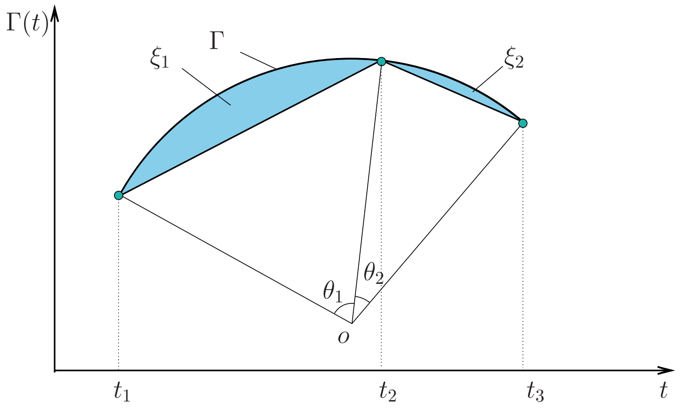

Figure 4: Approximation error in terms of the L^1-norm, for a curve Γ of constant curvature.

Similarly, the shaded area ξ_2 is given by:

$$\xi_2(\theta_2) = \frac{1}{2}\kappa^{-2}\left(\theta_2 - \sin\theta_2\right).$$

Let $s(t)$ be the path length of the graph of the function Γ between $t_1 = 0$ and t. The approximation error ξ of the piecewise linear approximation of Γ in terms of the $\mathcal{L}^1$-norm is given by the sum of ξ_1 and ξ_2:

$$\xi = \xi_1 + \xi_2 = \frac{1}{2}\kappa^{-2}(\theta_1 + \theta_2 - \sin\theta_1 - \sin\theta_2) = \frac{1}{2}\kappa^{-2}\big(\theta - \sin\theta_1 - \sin(\theta - \theta_1)\big),$$

where $\theta = \theta_1 + \theta_2 = s(t_3)\kappa$, which is constant for the given Γ. The first order derivative of ξ with respect to θ_1 is given by:

$$\frac{\mathrm{d}\xi}{\mathrm{d}\theta_1} = -\frac{1}{2}\kappa^{-2}\big(\cos\theta_1 - \cos(\theta - \theta_1)\big).$$

We assume that N is large enough such that the inequalities $\theta_1 < \pi$ and $\theta_2 < \pi$ hold. The first order necessary condition for the minimization of ξ, $\frac{\mathrm{d}\xi}{\mathrm{d}\theta_1} = 0$, yields that $\theta_1 = \theta_2$. This is indeed a minimum since

$$\frac{\mathrm{d}^2\xi}{\mathrm{d}\theta_1^2} = \frac{1}{2}\kappa^{-2}(\sin\theta_1 + \sin\theta_2) > 0.$$

Hence the proposition is proved. □

Lemma 2.3.1. The best piecewise linear interpolative approximation of a function Γ with constant curvature κ on a bounded interval $\mathcal{I}$ yields a constant density ρ_κ along the curve.

Proof. First, notice that if $\kappa = 0$, the result follows trivially. Thus, without loss of generality, assume that $\kappa \neq 0$. Assume now that the optimal piecewise linear approximation $\bar{\Gamma}$ of Γ corresponds to a distribution that is not equidistant. Owing to the one-one correspondence between the points of $\bar{\Gamma}$ (except the first one) and the angles $\theta_1, \theta_2, \ldots, \theta_{N-1}$ (see Fig. 4), the result is equivalent to the assertion that the best piecewise linear approximation corresponds to a distribution of angles $\theta^* = (\theta_1^*, \theta_2^*, \ldots, \theta_{N-1}^*)$, where $\theta_i^* \neq \theta_{i+1}^*$ for some $1 \leq i \leq N-1$. By virtue of Proposition 4.4.1, we can reduce the error over the arclength $\theta_i^* + \theta_{i+1}^*$ by moving the middle point such that the new angles are $\theta_i^{**} = \theta_{i+1}^{**} = (\theta_i^* + \theta_{i+1}^*)/2$, thus contradicting the minimality of the distribution θ^*. Hence θ^* must be equally distributed over the graph of Γ. □

Theorem 2.3.1. Consider a function Γ consisting of two segments Γ_1 and Γ_2 defined on contiguous, non-overlapping intervals, with constant curvature κ_1 and κ_2 of their respective graphs. Let N be the total number of points allocated to Γ. Then as $N \to \infty$, the error of the piecewise linear approximation of Γ is minimized by constant densities ρ_{κ_1} and ρ_{κ_2} on Γ_1 and Γ_2, respectively, satisfying

$$\frac{\rho_{\kappa_1}}{\rho_{\kappa_2}} = \left(\frac{\kappa_1}{\kappa_2}\right)^{\frac{1}{3}}. \tag{20}$$

Proof. Let N_1 be the number of points allocated to Γ_1 and let N_2 the number of points allocated to Γ_2 and let the corresponding angles over the arc lengths be $\theta_1 > 0$ and $\theta_2 > 0$, respectively. It follows that $\theta_1 = \kappa_1 S_1$, where S_1 is the length of Γ_1 and $\theta_2 = \kappa_2 S_2$, where S_2 is the length of Γ_2. With the best piecewise linear approximation of the function Γ_1 using N_1 points, the total approximation error $\xi_1(N_1)$ in the $\mathcal{L}^1$-norm is given by:

$$\xi_1(N_1) = \frac{1}{2}\kappa_1^{-2} \sum_{1}^{N_1-1} \left(\frac{\theta_1}{(N_1-1)} - \sin\frac{\theta_1}{N_1-1}\right) = \frac{1}{2}\kappa_1^{-2}\left(\theta_1 - (N_1-1)\sin\frac{\theta_1}{N_1-1}\right). \tag{21}$$

Similarly, with the best piecewise linear approximation of the function Γ_2 using $N_2 = N - N_1$ points, the total approximation error $\xi_2(N_2)$ in the $\mathcal{L}^1$-norm is given by:

$$\xi_2(N_2) = \frac{1}{2}\kappa_2^{-2}\left(\theta_2 - (N_2-1)\sin\frac{\theta_2}{N_2-1}\right). \tag{22}$$

Our objective is to minimize $\xi_1(N_1) + \xi_2(N_2)$ subject to $N_1 + N_2 = N$ as $N \to \infty$. Note that the last statement implies, in particular, that both $N_1, N_2 \to \infty$. (This is easy to see:

if both $N_1, N_2 \to \infty$ the approximation error goes to zero, whereas if either N_1 or $N_2 \not\to \infty$ as $N \to \infty$ the approximation error will not be zero and hence the point distribution is not optimal.)

To facilitate the proof, we consider the continuous version of this problem. To this end, let $x \in \mathbb{R}_+$ and $y \in \mathbb{R}_+$ and consider the problem of minimizing

$$\xi(x,y) = \xi_1(x) + \xi_2(y) = \frac{1}{2}\kappa_1^{-2}\left(\theta_1 - x\sin\frac{\theta_1}{x}\right) + \frac{1}{2}\kappa_2^{-2}\left(\theta_2 - y\sin\frac{\theta_2}{y}\right) \tag{23}$$

subject to $x + y = N$ and $N \to \infty$.

The first order necessary conditions for optimality for this problem lead to the expression

$$\kappa_2^{-2}\sin\left(\frac{\theta_2}{y}\right) - \kappa_2^{-2}\frac{\theta_2}{y}\cos\left(\frac{\theta_2}{y}\right) - \kappa_1^{-2}\sin\left(\frac{\theta_1}{x}\right) + \kappa_1^{-2}\frac{\theta_1}{x}\cos\left(\frac{\theta_1}{x}\right) = 0. \tag{24}$$

Using the power series expansion for the sine and cosine functions

$$\sin x = \sum_{n=0}^{\infty}\frac{(-1)^n}{(2n+1)!}x^{2n+1}, \qquad \cos x = \sum_{n=0}^{\infty}\frac{(-1)^n}{(2n)!}x^{2n},$$

equation (24) can be rewritten in terms of infinite series as

$$\kappa_2^{-2}\sum_{n=0}^{\infty}(-1)^{n+1}\frac{2n}{(2n+1)!}\left(\frac{\theta_2}{y}\right)^{2n+1} - \kappa_1^{-2}\sum_{n=0}^{\infty}(-1)^{n+1}\frac{2n}{(2n+1)!}\left(\frac{\theta_1}{x}\right)^{2n+1} = 0. \tag{25}$$

Since $N \to \infty$ we have that $x, y \to \infty$.

It follows that $\theta_1/x \ll 1$ and $\theta_2/y \ll 1$. As $x, y \to \infty$, the higher order terms in (25) vanishes, and one obtains

$$\kappa_2^{-2}\left(\frac{\theta_2}{y}\right)^3 - \kappa_1^{-2}\left(\frac{\theta_1}{x}\right)^3 = 0. \tag{26}$$

The solution of (26) yields,

$$\frac{x}{y} = \frac{S_1}{S_2}\left(\frac{\kappa_1}{\kappa_2}\right)^{1/3}, \tag{27}$$

from which we have

$$\frac{\rho_{\kappa_1}}{\rho_{\kappa_2}} = \lim_{N_1,N_2\to\infty}\frac{N_1/S_1}{N_2/S_2} = \lim_{x,y\to\infty}\frac{x/S_1}{y/S_2} = \left(\frac{\kappa_1}{\kappa_2}\right)^{1/3}. \tag{28}$$

The solution to (26) is indeed the optimal solution since the Hessian of $\xi(x,y)$ for $x, y \to \infty$, given by,

$$H(\xi) = \begin{bmatrix} \frac{\theta_1^2}{\kappa_1^2 x^3}\sin\left(\frac{\theta_1}{x}\right) & 0 \\ 0 & \frac{\theta_2^2}{\kappa_2^2 y^3}\sin\left(\frac{\theta_2}{y}\right) \end{bmatrix} \tag{29}$$

is positive definite for x, $y \neq 0$. □

Although Theorem (2.3.1) only gives an optimal density function for a 2D curve composed of two pieces of circular arcs, by induction, this conclusion holds also for curves with piecewise constant curvature profile, as described by the following Corollary:

Corolory 2.3.1. Let Γ be a planar curve with piecewise constant curvature κ. Let N be the total number of grid points allocated to Γ. Then as $N \to \infty$, the error of the piecewise linear approximation of Γ is minimized with the grid points distributed by the density function $\kappa^{1/3}$.

Before presenting the results regarding the best piecewise linear interpolative approximation of planar C^3-smooth curves,

One way for applying Corollary 2.3.1 to more general functions such as C^3-smooth functions would be first approximating the C^3-smooth function using a function with piecewise constant curvature profile, then generate the partition according to Corollary 2.3.1. As a result, it is necessary to estimate the approximation error of C^3-smooth functions using circular arc splines. *Circular arc spline*, or *arc spline*, is a curve comprising joined circular arcs. Circular arc splines has been studied in computational geometry and computer graphics, with Refs. [60, 88, 106] as a few examples. Note that a straight line is a circle with zero curvature, the piecewise linear spline used in this Chapter can be viewed as a special type of circular arc spline.

It is shown that C^3-smooth curves can be approximated to arbitrary precision using a specific type of circular arc splines which preserve the curve length of the original smooth curve, and an upper bound of the approximation error can be established [105]. To address the distribution of grid points, we consider a different type of arc splines which also preserve the curve length, but this type is different from the one in Ref. [105] in the sense that, on the interval between adjacent grid points, the curvature function of this type of arc splines is constant instead of being piecewise constant with two constant values; Besides, it is not required that the arc spline is tangent to the smooth curve at the grid points. Since the tangent condition is relaxed, it is easily shown that the path length is preserved by choosing appropriate curvature values for each arc.

Let a C^3-smooth curve Γ in the two dimensional plane be given by a curvature function $\kappa(s)$, $s \in [s_0, s_f]$. Consider the case that Γ does not contain any circular arcs. By partitioning Γ into a finite number of segments and assign a local coordinate to each segment, we may assume that κ is a strictly monotone function, and the angle α between the tangent of Γ and the x-axis is between $-\pi/6$ and $\pi/6$ ($\pi/6$ is an arbitrary choice for the convenience of proof). Let $x(s)$, $y(s)$ be the coordinate functions of Γ.

Given an arbitrary grid $\{s_i\}_{i=0}^{N-1}$ containing N points with $s_0 < s_1 < \cdots < s_{N-1} = s_f$, define an arc spline $\hat{\Gamma}$ for the approximation of Γ with the curvature function $\hat{\kappa}(s) = \kappa_i$, for $s \in [s_{i-1}, s_i)$, $i = 1, \ldots, N-2$ and $s \in [s_{N-2}, s_{N-1})$, where κ_i is chosen such that $\min\{\kappa(s_{i-1}), \kappa(s_i)\} < \kappa_i < \max\{\kappa(s_{i-1}), \kappa(s_i)\}$, and the length of Γ is preserved by $\hat{\Gamma}$. Besides, $\hat{\Gamma}(s_i) = \Gamma(s_i)$ for $i = 1, \ldots, N-1$. Let $\hat{x}$, $\hat{y}$ be the coordinate functions of the circular spline approximation, and $\hat{\alpha}$ be the angle between the tangent of $\hat{\Gamma}$ and the x-axis, with $-\pi/6\hat{\alpha} < \pi/6$. The grid $\{s_i\}_{i=0}^{N-1}$ also corresponds to a grid $\{x_i\}_{i=0}^{N-1}$ on the x-axis, which is well defined with $x_0 < x_1 < \cdots < x_{N-1}$.

The following theorem extend the result in Ref. [105] to the estimation of the L^1 norm of the approximation error. Because the arc spline considered here is different from that in Ref. [105], we also include a sketch of the proof regarding the error estimation on the deviation of the slope angel function $\alpha(s) - \hat{\alpha}(s)$. We also extends the result in Ref. [105] and provide an estimation of the L^1 norm of the approximation error.

Theorem 2.3.2 (Error estimation for circular arc spline approximation)**.** *There exist positive real numbers M_i, $i = 1, \ldots, N-1$, such that the deviation of the slope angle function of the approximation from the corresponding function of the original curve satisfies*

$$|\alpha(s) - \hat{\alpha}(s)| \leq M_i(s_i - s_{i-1})^2, \quad s \in [s_{i-1}, s_i], i = 1, \ldots, N,$$

Further more, the L^1 norm of the approximation error $y - \hat{y}$ on $[x_{i-1}, x_i]$, which is given by

$$\eta_i = \int_{x_{i-1}}^{x_i} |y(x) - \hat{y}(x)| \mathrm{d}x,$$

satisfies $\eta_i \leq M_i \max\{(s_i - s_{i-1})^4, (s_i - s_{i-1})^7\}$.

Proof. consider the interval $[s_{i-1}, s_i]$, and assume without loss of generality that $\kappa(s_{i-1}) < \kappa(s_i)$. Because $\kappa(s_{i-1}) < \kappa_i < \kappa(s_i)$, and κ is monotone, there exists $s_p \in [s_{i-1}, s_i]$ such that $\kappa(s_p) = \kappa_i$, i.e., $\kappa(s_p) - \hat{\kappa}(s_p) = 0$. Note that κ is Lipschitz since Γ is C^3 smooth. As a result, there exists a finite Lipschitz constant M_{a_i} such that $|\kappa(s) - \hat{\kappa}(s)| < M_{a_i}(s_i - s_{i-1})$, $s \in [s_{i-1}, s_i]$.

The angles α and $\hat{\alpha}$ are given by

$$\begin{aligned}\alpha(s) &= \alpha(s_{i-1}) + \int_{s_{i-1}}^{s} \kappa(s)\mathrm{d}s,\\ \hat{\alpha}(s) &= \hat{\alpha}(s_{i-1}) + \int_{s_{i-1}}^{s} \hat{\kappa}(s)\mathrm{d}s.\end{aligned}$$

Obviously, α and $\hat{\alpha}$ are continuous functions.

Because $\Gamma(s_{i-1}) = \hat{\Gamma}(s_{i-1})$ and $\Gamma(s_i) = \hat{\Gamma}(s_i)$, we must have

$$\left(\hat{\alpha}(s_{i-1}) - \alpha(s_{i-1})\right)\left(\hat{\alpha}(s_i) - \alpha(s_i)\right) < 0.$$

Therefore, by the continuity of α and $\hat{\alpha}$, there exists $s_q \in [s_{i-1}, s_i]$ such that $\hat{\alpha}(s_q) = \alpha(s_q)$.

Now rewritten the expressions of α and $\hat{\alpha}$ on $[s_{i-1}, s_i]$ as

$$\begin{aligned}\alpha(s) &= \alpha(s_q) + \int_{s_q}^{s} \kappa(s)\mathrm{d}s,\\ \hat{\alpha}(s) &= \hat{\alpha}(s_q) + \int_{s_q}^{s} \hat{\kappa}(s)\mathrm{d}s.\end{aligned}$$

Subtracting one of the above expression from the other, we have

$$|\alpha(s) - \hat{\alpha}(s)| = \left|\int_{s_q}^{s} \left(\kappa(s) - \hat{\kappa}(s)\right)\mathrm{d}s\right| \leq \int_{s_q}^{s} |\kappa(s) - \hat{\kappa}(s)|\,\mathrm{d}s \leq M_{a_i}(s_{i+1} - s_i)^2.$$

Because the L^1 norm of the approximation error equals to the area between two curves Γ and $\hat{\Gamma}$ on $[x_{i-1}, x_i]$, we will compute the value of η_i using the polar coordinate, which is illustrated in the Fig. 5.

$$\delta_i(s) = \int_{s_{i-1}}^{s} \tan(\alpha(\tau) - \hat{\alpha}(\tau))\mathrm{d}\tau.$$

Because $|\alpha| < \pi/6$ and $|\hat{\alpha}| < \pi/6$, we have $|\alpha - \hat{\alpha}| < \pi/3$. Since the tangent function is Lipschitz on $[-\pi/3, \pi/3]$, there exists a constant M_{t_i} such that

$$|\tan(\alpha(\tau) - \hat{\alpha}(\tau))| \leq \frac{M_{t_i}}{M_{a_i}}|\alpha(\tau) - \hat{\alpha}(\tau)|.$$

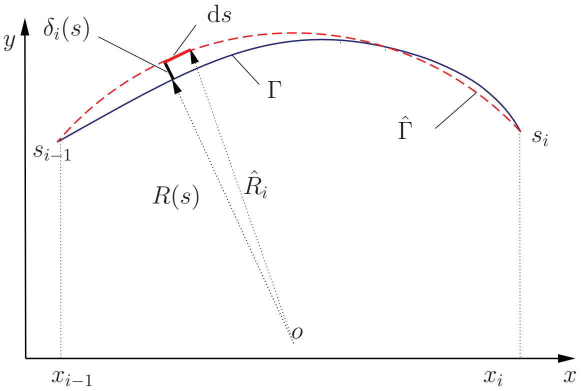

Figure 5: L^1-norm of the approximation error on $[s_{i-1}, s_i]$ for a C^3-smooth curve.

Then,

$$\begin{aligned}
|\delta_i(s)| &\leq \int_{s_{i-1}}^{s} |\tan(\alpha(\tau) - \hat{\alpha}(\tau))|\mathrm{d}\tau \\
&\leq \frac{M_{t_i}}{M_{a_i}} \int_{s_{i-1}}^{s} |\alpha(\tau) - \hat{\alpha}(\tau)|\mathrm{d}\tau \\
&\leq \frac{M_{t_i}}{M_{a_i}} M_{a_i} \int_{s_{i-1}}^{s} (s_{i+1} - s_i)^2 \mathrm{d}\tau \\
&= M_{t_i}(s_i - s_{i-1})^2 (s - s_{i-1}).
\end{aligned}$$

Hence,

$$\begin{aligned}
\eta_i &= \int_{s_{i-1}}^{s} \left|R_i^2 - R^2(\tau)\right| |\kappa_i| \,\mathrm{d}\tau \\
&= \int_{s_{i-1}}^{s} |\kappa_i| \left|(R_i + R(\tau))(R_i - R(\tau))\right| \mathrm{d}\tau \\
&= \int_{s_{i-1}}^{s} |\kappa_i| \left|(2R_i - \delta(\tau))\,\delta(\tau)\right| \mathrm{d}\tau \\
&\leq 2\int_{s_{i-1}}^{s} |\delta(\tau)|\,\mathrm{d}\tau + |\kappa_i| \int_{s_{i-1}}^{s} \left|\delta^2(\tau)\right| \mathrm{d}\tau \\
&\leq M_{t_i}(s_i - s_{i-1})^2 \int_{s_{i-1}}^{s_i} (s - s_{i-1})\mathrm{d}s \\
&+ |\kappa_i| M_{t_i}(s_i - s_{i-1})^4 \int_{s_{i-1}}^{s_i} (s - s_{i-1})^2 \mathrm{d}s \\
&= \frac{M_{t_i}}{2}(s_i - s_{i-1})^4 + \frac{M_{t_i}}{3}(s_i - s_{i-1})^7.
\end{aligned}$$

Let $M_i = \max\{M_{a_i}, 5M_{t_i}/6\}$, then the proof is complete. □

The following result from Ref. [81] extends Corollary 2.3.1 to more general functions and metrics.

Let $C^r_*[0,1]$ denote the set of the functions $\Gamma(t) \in C^{r+1}[0,1]$ such that $\Gamma^{(r)}(t) > 0$, $0 \leq t \leq 1$. Let $\Delta_n = \{t_0 < t_1 < \cdots < t_n\}$ be an arbitrary partition of the closed interval $[t_0, t_n]$, and $s_{n,r}(t)$ be the function which is an algebraic polynomial of degree at most $r-1$, $(r = 1, 2, \dots)$ on each of the closed intervals $[t_{i-1,n}, t_{i,n}]$ $(i = 1, 2, \dots, n)$. Define

$$\mathscr{E}(\Gamma; \Delta_n)_X = \inf_{s_{n,r}} \|\Gamma(t) - s_{n,r}(t)\|_{X_{[t_0,t_n]}},$$

and

$$\mathscr{E}_{n,r}(\Gamma)_X = \inf_{\Delta_n} \mathscr{E}(\Gamma; \Delta_n)_X,$$

where X is the space L^p.

Theorem 2.3.3 (The best choice of nodes for approximation using splines in L^p space[81]). Let $\Gamma \in C^r_*[0,1]$,

$$\mathscr{E}_{n,r}(\Gamma)_X = \frac{B_{r,p}}{r! n^r 2^{(rp+1)/p}} \left[\int_0^1 (\Gamma^{(r)}(t))^{p/(rp+1)} \mathrm{dt} \right]^{(rp+1)/p} + O(\frac{1}{n^{rp/(rp+1)+r}})$$

where

$$B_{r,p} = \min_{a_k} \|t^r - \sum_{k=0}^{r-1} a_k t^k\|_{L_p[-1,1]} \quad (1 \leq p < \infty)$$

as $n \to \infty$. Moreover, the asymptotically best location of the nodes is determined from the equations

$$\int_0^{x^*_{i,n}} \left[\Gamma^{(r)}(t)\right]^{p/(rp+1)} \mathrm{d}t = \frac{i}{n} \int_0^1 \left[\Gamma^{(r)}(t)\right]^{p/(rp+1)} \mathrm{d}t. \tag{30}$$

Although it is assumed for Theorem 2.3.3 that $\Gamma^2 > 0$, by partitioning Γ into segments based on the sign of the curvature, and assign proper local coordinate to the segments with negative curvature, this assumption can always be satisfied. Let

$$\rho^*(x) = \frac{\left[\Gamma^{(r)}(t)\right]^{p/(rp+1)}}{\int_0^1 \left[\Gamma^{(r)}(t)\right]^{p/(rp+1)} \mathrm{d}t},$$

The associated cumulative function is given by

$$F(x) = \int_0^x \rho^*(t) \mathrm{d}t,$$

Then equation (30) is equivalent to

$$F(x^*_{i,n}) - F(x^*_{i-1,n}) = \frac{1}{n},$$

Therefore ρ^* is the optimal density function for the grid point distribution scheme described by (30). When Γ is a C^3-smooth curves, $r = 2$. Since L^1 norm is considered in this thesis, we have $p = 1$. The optimal density function is

$$\rho^*(x) = \frac{\left[\Gamma^{(2)}(x)\right]^{1/3}}{\int_0^1 \left[\Gamma^{(2)}(t)\right]^{1/3} \mathrm{d}t} = c\left[\Gamma^{(2)}(x)\right]^{1/3},$$

where c is a constant. Let s be the path coordinate. Noticing that $\mathrm{d}s = \sqrt{1 + [\Gamma^{(1)}(s)]^2}\mathrm{d}t$, we have

$$\begin{aligned}\rho^*(s) &= c\left[\Gamma^{(2)}(s)\right]^{1/3}\left(1 + [\Gamma^{(1)}(s)]^2\right)^{-1/2},\\ &= c\left[\frac{\Gamma^{(2)}(s)}{\left(1 + [\Gamma^{(1)}(s)]^2\right)^{3/2}}\right]^{\frac{1}{3}}\\ &= c\kappa^{1/3}(s),\end{aligned}$$

Which is the same as the optimal density function for the case when Γ is piecewise circular.

2.4 Costate Estimation

In direct collocation methods, which are implemented in the previously mentioned DENsity function based mesh refinement algorithm (DENMRA), the decision variables include the states and controls only, while the costates are related to the Lagrangian multipliers associated with the NLP. The feasibility of the optimized solution can be checked easily by integrating the system dynamics using the optimized controls and compare the integration result with the optimized states. To check the optimality of the result, it is necessary to recover the costates from the Lagrangian multipliers and compute the Hamiltonian. In this section, we describe briefly the costate estimation technique from Ref. [114], which was implemented in DENMRA.

Consider the following optimal control problem stated in Mayer form.

$$\min_{u\in \mathrm{PWC}\left(\left[t_0,t_f\right]\right)^m, t_0,t_f\in\mathbb{R}} \varphi\left[x\left(t_f\right), t_f\right]$$

subject to the conditions

$$
\begin{aligned}
&\dot{x} = f\left(x\left(t\right), u\left(t\right), t\right)\\
&\psi_0\left(x\left(t_0\right), t_0\right) = 0\\
&\psi_f\left(x\left(t_f\right), t_f\right) = 0\\
&g_e\left(x\left(t\right), u\left(t\right), t\right) = 0\\
&g_i\left(x\left(t\right), u\left(t\right), t\right) \leq 0\\
&h_e\left(x\left(t\right), t\right) = 0\\
&h_i\left(x\left(t\right), t\right) \leq 0
\end{aligned}
$$

Here $t \in \mathbb{R}$, $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ are time, state vector and control vector, respectively. The functions

$$
\begin{array}{cc}
\varphi : \mathbb{R}^{n+1} \to \mathbb{R} & f : \mathbb{R}^{n+m+1} \to \mathbb{R}^n\\
\psi_0 : \mathbb{R}^{n+1} \to \mathbb{R}^{k_0} & \psi_f : \mathbb{R}^{n+1} \to \mathbb{R}^{k_f}\\
k_0 \leqslant n+1 & k_f \leqslant n\\
g_e : \mathbb{R}^{n+m+1} \to \mathbb{R}^{k_{ge}} & g_i : \mathbb{R}^{n+m+1} \to \mathbb{R}^{k_{gi}}\\
h_e : \mathbb{R}^{n+1} \to \mathbb{R}^{k_{he}} & h_i : \mathbb{R}^{n+1} \to \mathbb{R}^{k_{hi}}
\end{array}
$$

are sufficiently smooth with respect to their arguments. $\mathrm{PWC}\left([t_0, t_f]\right)^m$ denotes the set of piecewise continuous functions defined on interval $[t_0, t_f]$.

2.4.1 Discretized Optimal Control Problem

By discretizing the above optimal control problem using collocation, both the states and controls are discretized, and the dynamic and state constraints are enforced only at isolated points. Using a trapezoidal rule to enforce the equations of motion at a single point between neighboring nodes, the scheme leads to the following NLP problem:

$$
\min_{x_0,\ldots,x_N,u_1,\ldots,u_N,t_0,t_N \in \mathbb{R}^{nN+1+mN+2}} \varphi\left(x_N, t_N\right)
$$

subject to the conditions

$$
\begin{aligned}
&\dot{\bar{x}}_j - f\left(\bar{x}_j, u_j, \bar{t}_j\right) = 0, j = 1, \ldots, N,\\
&\psi_0\left(x_0, t_0\right) = 0,\\
&\psi_f\left(x_f, t_f\right) = 0,\\
&\ g(\bar{x}_j, u_j, \bar{t}_j) \leq 0, \quad , j = 1, \ldots, N,\\
&\ h(\bar{x}_j, u_j, \bar{t}_j) \leq 0, \quad , j = 1, \ldots, N,
\end{aligned}
$$

where

$$
\left.
\begin{aligned}
\bar{t}_j &= \tfrac{t_j + t_{j-1}}{2}\\
\bar{x}_j &= \tfrac{x_j + x_{j-1}}{2}\\
\dot{\bar{x}}_j &= \tfrac{x_j - x_{j-1}}{t_j - t_{j-1}}
\end{aligned}
\right\} \quad j = 1, \ldots, N.
$$

The Lagrangian function associated with the discretized optimal control problem is given by

$$
\begin{aligned}
L = \varphi\left(x_N, t_N\right) + \pi_0^T \psi_0\left(x_0, t_0\right) + \pi_f^T \psi_f\left(x_N, t_N\right) + \sum_{j=1}^{N} \lambda_j^T \left[f\left(\bar{x}_j, u_j, \bar{t}_j\right) - \dot{\bar{x}}_j\right]\\
+ \sum_{j=1}^{N} \sigma_j^T g\left(\bar{x}_j, u_j, \bar{t}_j\right) + \sum_{j=0}^{N} \mu_j^T h\left(\bar{x}_j, \bar{t}_j\right).
\end{aligned}
$$

2.4.2 Costate Estimates

It is well-known that the Lagrangian multipliers λ_j correspond to the sensitivity of the optimal cost with respect to the perturbations in the state vector x_j at time t_j. However, in order to provide a valid estimation of the costates in the original optimal control problem, certain post processing of the Lagrangian multipliers is necessary.

When the state constraint is not active at initial time t_0, the costate is given by

$$
\lambda\left(t_0^-\right)^T = \frac{\lambda_1^T}{2} \frac{\partial f}{\partial x}\bigg|_{(\bar{x}_1, u_1, \bar{t}_1)} + \frac{\lambda_1^T}{t_1 - t_0} + \frac{\sigma_1^T}{2} \frac{\partial g}{\partial x}\bigg|_{(\bar{x}_1, u_1, \bar{t}_1)} + \frac{\mu_0^T}{2} \frac{\partial h}{\partial x}\bigg|_{(\bar{x}_1, \bar{t}_1)}.
$$

If the state constraint becomes active at t_0, then the above expression actually gives the value of the costate just before the state constraint is active, and the costate jump introduced later should be used to compute the costate at t_0.

At each individual node t_i, supposing that the state constraint is not active at t_i, the value of the costate can be obtained by deleting the i leading nodes ($i = 0, \ldots, i-1$) and consider t_i as the initial time. Then the costate estimate would be

$$\lambda\left(t_i^-\right)^T = \frac{\lambda_{i+1}^T}{2}\left.\frac{\partial f}{\partial x}\right|_{(\bar{x}_{i+1},u_{i+1},\bar{t}_{i+1})} + \frac{\lambda_{i+1}^T}{t_{i+1}-t_i} + \frac{\sigma_{i+1}^T}{2}\left.\frac{\partial g}{\partial x}\right|_{(\bar{x}_{i+1},u_{i+1},\bar{t}_{i+1})} + \frac{\mu_i^T}{2}\left.\frac{\partial h}{\partial x}\right|_{(\bar{x}_i,\bar{t}_i)}, \tag{31}$$

for $i = 0, \ldots, N-1$. Again, if no state constraints are active at t_i, then the costate function $\lambda(t)$ is continuous at t_i, so $\lambda\left(t_i^-\right)^T$ can be replaced by $\lambda\left(t_i\right)^T$, otherwise equation (31) is only an estimate of the costate value before the jump at t_i.

The costate value at the final node t_f is computed by the expression below:

$$\lambda\left(t_N\right)^T = \left.\frac{\partial\varphi}{\partial x}\right|_{(\bar{x}_N,\bar{t}_N)} + \nu_f^T\left.\frac{\partial\psi_f}{\partial x}\right|_{(\bar{x}_N,\bar{t}_N)},$$

where $\nu_f = -\pi_f$.

Suppose that the state constraints are active for a certain number of nodes, namely,

$$h\left(x_j, t_j\right) = \begin{cases} < 0, & j = 0, \ldots, i_{a-1}, \\ = 0, & j = i_a, \ldots, i_b, \\ < 0, & j = i_{b+1}, \ldots, N. \end{cases}$$

In the variational approach to the state-constrained optimal control problems, the active state constraint $h(x(t),t) = 0$ on $t \in [t_a, t_b]$ is transformed into an equivalent combination of interior point constraint and a control constraint:

$$M\left(x\left(t_a\right), t_a\right) = \begin{bmatrix} \left.h\left(x,t\right)\right|_{t_a} \\ \left.\dfrac{\mathrm{d}h\left(x,t\right)}{\mathrm{d}t}\right|_{t_a} \\ \vdots \\ \left.\dfrac{\mathrm{d}^{q-1}h\left(x,t\right)}{\mathrm{d}^{q-1}t}\right|_{t_a} \end{bmatrix} = 0,$$

and $\dfrac{\mathrm{d}^q h\left(x,t\right)}{\mathrm{d}^q t} = 0$ for $t \in [t_a, t_b]$, where q is the smallest integer for which the control appears explicitly in the corresponding derivative.

Then the jump in the costate due to the activation of state constraint is given by

$$\lambda\left(t_i\right)^T - \lambda\left(t_i^-\right)^T = l^T\frac{\partial M\left(x\left(t_i\right), t_i\right)}{\partial x},$$

where the components of l are given by $l_j = \sum\limits_{k=i}^{i_b}\dfrac{\mu_k}{j!}\left(t_k - t_i\right)^j$, $j = 0, \ldots, q-1$.

2.4.3 Numerical Example

Consider the Brachistochrone problem with state constraint as in Ref. [114]. In Meyer form, the problem can be stated as follows:

$$\min_{u \in \mathrm{PWC}[t_0, t_f]} \quad t_f$$

subject to the equations of motion

$$\dot{x}\left(t\right) = v\left(y\right)\cos\theta\left(t\right),$$

$$\dot{y}\left(t\right) = v\left(y\right)\sin\theta\left(t\right),$$

boundary contitions

$$x\left(0\right) = 0,$$

$$x\left(t_f\right) = 1,$$

$$y\left(0\right) = 0,$$

$$y\left(t_f\right)\,\text{free},$$

and the state constraint

$$\mathrm{y}\left(t\right) - x\left(t\right)\tan\gamma - h_0 \leqslant 0.$$

The quantities $v_0 = 1$, $g = 1$, $\gamma = 20\,\mathrm{deg}$, and $h_0 = 0.05$ are constants. The angle θ is the only control, v denotes the velocity, and can be computed by $v = \sqrt{v_0^2 + 2gy}$. The state inequality is of first order, and the optimal switching structure is free—constrained—free. The costates computed using the Lagrangian multipliers are shown in Fig. 6 and Fig. 7. The Hamiltonian is shown in Fig. 8. It can be seen that the Hamiltonian is constant at -1, which suggests the local optimality of the solution and the validity of the costate estimation.

It needs to be pointed out that this costate estimation method is tailored for the trapezoidal discretization scheme [114], and does not hold for other schemes. New formulas need to be derived if other discretization schemes are to be applied. Besides, as can be seen in this example, a successful implemenation of the costate estimation technique also requires the correct knowledge of the structure of the engagement of the state constraint. Furthermore, the differentiation of the state constraint $h(x(t), t)$ also needs to be derived and implemented before the computation of the jump of the costate, the complexity of this process could vary depending on the problem to be solved. For very complicated problems, the implementation of this costate estimation method may not be very easy.

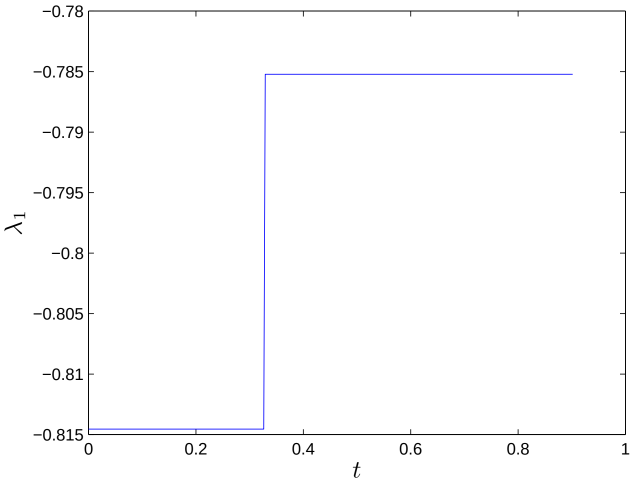

Figure 6: Costate history: λ_x.

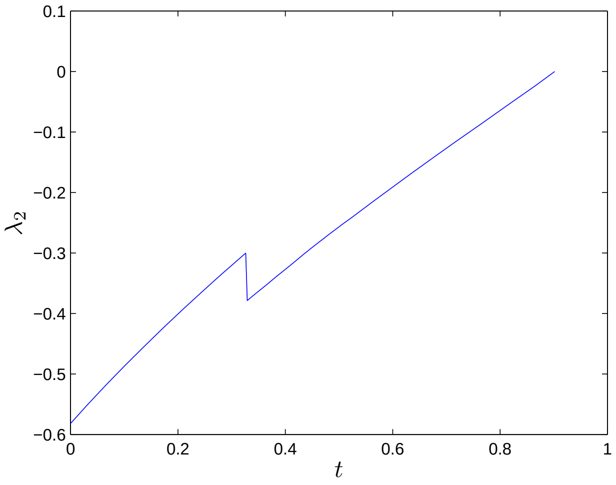

Figure 7: Costate history: λ_y.

2.5 Density Function-based Mesh Refinement Algorithm (DENMRA)

In this section we present the DENsity function-based Mesh Refinement Algorithm (DENMRA), which is an iterative algorithm for solving optimal control problems, utilizing the mesh generation method based on a mesh density function introduced previously.

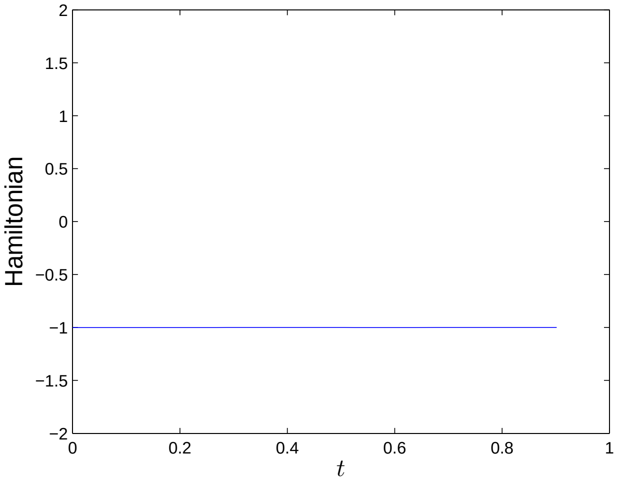

Figure 8: Hamiltonian history.

The use of a density function is one of the key components in DENMRA. General optimal control problems involve ordinary differential equations in terms of the state variables, which describe how the control changes the vector field of the states. For such problems, since the states are continuous, irregularities in the smoothness in the states usually correspond to fast (or discontinuous) changes in the control. Hence, typically, the control history is used in DENMRA for computing the density function to capture smoothness irregularities in both the state and the control histories, although this is not restrictive. The state histories can be used as well, if needed.

2.5.1 Major Steps of DENMRA

When solving a general optimal control problem that minimizes the cost function J using m control inputs, DENMRA goes through the following four major steps:

(1) Set $j = 1$. Choose a positive integer N_j and generate the initial uniform mesh $T_1 = \{t_i\}_{t_i=1}^{N_j}$, where $t_i = (i-1)/(N_j-1)$, Generate an initial guess for the state and control variables, and solve the discretized problem that minimizes J;

(2) Calculate the density function f using the discretized control $\{(t_i, \mathbf{u_i})\}_{i=1}^{N_j}$ of the previous solution, where $\mathbf{u_i} \in \mathbb{R}^m$;

(3) Determine the mesh size increment ΔN_j by discretization error estimation which is introduced in Ref. [26]. Let $N_{j+1} = N_j + \Delta N_j$, and generate the new mesh $T_{j+1} = \{t_i\}_{i=1}^{N_{j+1}}$ based on f. Set $j = j + 1$;

(4) Generate the initial guess based on the previous solution for mesh T_j, solve the problem, and go to Step (2), unless some stopping rule is met.

2.5.2 Technical Details

The details of these steps are given below.

Initial Guess

For simplicity, DENMRA may start from a constant initial guess for all control and state variables, but – as typical with nonlinear optimization problems – any good initial guess based on prior experience with the problem or good engineering judgment can improve convergence.

Optimization

After the cost function and the dynamic, state, control and path constraints have been discretized on the given grid, DENMRA calls a nonlinear programming (NLP) solver. In this implementation, we have used the optimization software SNOPT [54] for solving the corresponding nonlinear programming problem stemming from the discretized optimal control problem.

Density Function Computation

In DENMRA, when the density function based on the local curvature as described in [136] is used, the discrete control $\{(t_i, \mathbf{u_i})\}_{i=1}^{N_j}$ from the previous iteration is used to estimate the curvature of the graph of the control history. This curvature based density function provides the best piecewise linear interpolative approximation to the graph in an asymptotic sense as the size of grid increases. For more details about the proof please refer to the appendix. The calculation of the density function corresponding to the control $\mathbf{u}$ is therefore computed as follows:

(1) Let $u_{i,k}$ be the k^{th} component of the discrete control value $\mathbf{u_i}$ at t_i, $\dot{u}_{i,k}$ be the first order derivative of the k^{th} component of control at time $t'_i = (t_{i+1} + t_i)/2$, and $\ddot{u}_{i,k}$ be the

second order derivative at time $t_i'' = (t_{i+1}' + t_i')/2$. Then, for $k = 1, \ldots, m$, the values $\{\dot{u}_{i,k}\}_{i=1}^{N_j-1}$ and $\{\ddot{u}_{i,k}\}_{i=1}^{N_j-2}$ can be approximated by $\dot{u}_{i,k} \approx (u_{i+1,k} - u_{i,k})/(t_{i+1} - t_i)$ and $\ddot{u}_{i,k} \approx (\dot{u}_{i+1,k} - \dot{u}_{i,k})/(t_{i+1}' - t_i')$, respectively. Interpolate $\{(t_i',\, \dot{u}_{i,k})\}_{i=1}^{N_j-1}$ using a spline function at t_i'' and obtain $\{(t_i'',\, \dot{u}_{i,k}')\}_{i=1}^{N_j-2}$.

(2) Compute density function as $\rho_{i,k} = (\kappa_{i,k}^{1/3} + \epsilon)\sqrt{1 + \dot{u}_{i,k}^2} = |\ddot{u}_{i,k}|^{1/3} + \epsilon\sqrt{1 + \dot{u}_{i,k}^2}$, where $\epsilon > 0$. The actual curvature $\kappa(t)$ is chosen as a piecewise constant function with $\rho_k(t) = (\rho_{i,k} + \rho_{i+1,k})/2$ for $t \in [t_i, t_{i+1}]$. Note that a small positive number ϵ is added to the actual curvature density function. In practice, this means that a few grid points are kept even on the parts of the control history that are straight lines or segments with very small curvature. This is always a good idea since the control history on $\mathcal{I}_s$ may change in subsequent iterations, and it is thus advisable to keep some points in the interior of the interval $\mathcal{I}_s$ in order to capture possible changes of the control histories.

(3) The overall (non-normalized) density function f is obtained by merging the density functions for all controls. For instance,

$$f(t) = \left(\sum_{k=1}^{m} \rho_k^2(t) \right)^{\frac{1}{2}}, \tag{32}$$

and

$$f(t) = \max_k \rho_k(t) \tag{33}$$

are two possible methods to generate the overall density function.

2.5.2.1 Mesh Generation

DENMRA typically starts with a coarse uniform mesh in order to capture the basic structure of the control history. In subsequent iterations, the user can either let DENMRA decide the mesh size based on the integration error, or adjust the final mesh size and the number of iterations according to the desired or imposed speed and accuracy requirements depending on the problem at hand. In the former case, at each mesh refinement iteration, cubic splines are used to approximate the state and control histories, and the local discretization error of the previous mesh is estimated. After the density function is computed based on the result of the previous iteration, a temporary new mesh size $\tilde{N}_j$ is found by gradually increasing $\tilde{N}_j$ from N_j until the maximum local discretization error of the new mesh generated using the density function with $\tilde{N}_j$ points is smaller than that of the previous mesh. Let $N_{\max}$ be a limit on the final mesh size, then the actual mesh size increment after the j^{th} iteration is determined by $\Delta N_j = \min\{\tilde{N}_j - N_j, \Delta N_{\max}\}$, where $\Delta N_{\max} = N_{\max} - N_j$. if $\Delta N_j = \tilde{N}_j - N_j$, then the last temporary mesh would be used for the next iteration. Otherwise a new mesh would be generated with $N_j + \Delta N_{\max}$ points.

2.5.2.2 Stopping Rule

DENMRA stops either when the maximum number of mesh refinement iterations is reached, or when the optimality of the problem cannot be further improved and the local integration error is smaller than the specified tolerance.

2.6 Numerical Examples

In this section we report the results from two numerical examples, generated to illustrate the good properties of the proposed mesh generation method. The first example is the double integrator minimum energy problem [29]. Since this problem has an analytical solution, it can be used to check the accuracy and optimality of the proposed method. It also includes a state constraint, which is used to demonstrate that the proposed methods is able to handle higher order state irregularities stemming from such state constraints. The second example deals with a “hypersensitive” optimal control problem [102] and it is used to test the robustness of the method when dealing with problems requiring highly concentrated grid points at certain phases of the solution. For comparison, the same two problems are also solved using SOCS [23], which is a widely used software for solving trajectory optimization problems. Both algorithms start with trapezoidal integration, and switch to higher order Hermite-Simpson integration later on to meet the desired accuracy/optimality. A feasibility tolerance of 10^{-10} is used for both algorithms.

2.6.1 Minimum Energy for Double Integrator

The double integrator problem is given by:

$$\begin{aligned} \dot{v} &= u, \qquad & v(0) = -v(1) = 1, \\ \dot{x} &= v, \qquad & x(0) = x(1) = 0, \end{aligned}$$

and the goal is to find $u(t)$, where $0 \leq t \leq 1$, to minimize

$$J = \frac{1}{2}\int_0^1 u^2 \, \mathrm{d}t,$$

with the state constraint $x(t) \leq \ell$, where ℓ is a positive real number.

The solution of the optimal control $u^*(t)$ can be obtained as follows [29]:

$$u^*(t) = -2, \qquad 0 \leq t \leq 1, \qquad \text{for } \ell \geq \tfrac{1}{4};$$

$$u^*(t) = \begin{cases} -8(1-3\ell) + 24(1-4\ell)t, & 0 \leq t \leq \frac{1}{2}, \\ -8(1-3\ell) + 24(1-4\ell)(1-t), & \frac{1}{2} < t \leq 1, \end{cases} \qquad \text{for } \tfrac{1}{6} \leq \ell < \tfrac{1}{4};$$

$$u^*(t) = \begin{cases} -\frac{2}{3\ell}(1 - \frac{t}{3\ell}), & 0 \leq t \leq 3\ell, \\ 0, & 3\ell < t \leq 1 - 3\ell, \\ -\frac{2}{3\ell}(1 - \frac{1-t}{3\ell}), & 1 - 3\ell < t \leq 1, \end{cases} \qquad \text{for } \ell < \tfrac{1}{6}.$$

2.6.1.1 Comparison in Terms of Accuracy and Optimality

The curvature based-density function is used for mesh refinement in DENMRA for this problem. This density function is given by $\rho_\kappa(t) = \kappa(t)^{1/3}$, $t \in [0,1]$, where κ is the curvature of the graph of the control function. As mentioned previously in Section 2.3, this density function provides the best piecewise linear interpolative approximation of the control. The same problem was also solved using the commercial numerical optimal control code SOCS, which implements the mesh refinement strategy of [26]. Both algorithms were tested on the same computer, and cold-started using the same linear initial guess.

Table 1 summarizes the results from DENMRA and SOCS for the double integrator problem. In the table, N is the size of the final mesh, $|J - J^*|$ is the optimality error, and

$\|u_i - u^*(t_i)\|_\infty = \max_i |u_i - u^*(t_i)|$ is the norm of the error between the discretized control $\{u_i\}_{i=1}^N$ and the exact solution u^*. Our numerical experiments showed that SOCS could not achieve highly accurate solution for this problem even if the local integration error tolerance has been set to 10^{-14}. The optimality error of the SOCS solution was around $10^{-4} \sim 10^{-6}$ with a maximum control error around $10^{-2} \sim 10^{-3}$. DENMRA exhibited an optimality error at the order $10^{-7} \sim 10^{-13}$, and a maximum control error at the order of $10^{-5} \sim 10^{-6}$.

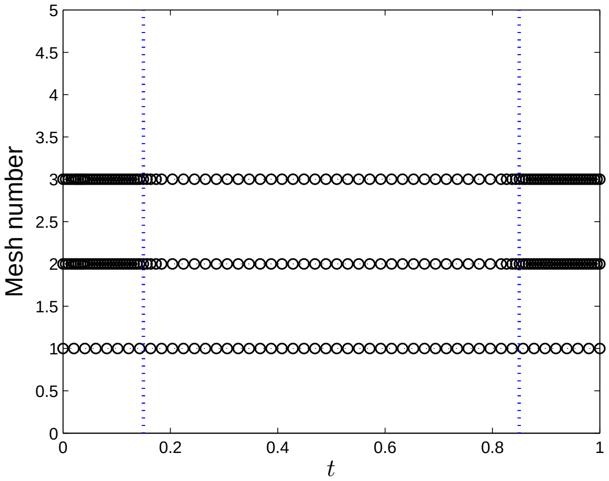

Figure 9: Mesh refinement, SOCS, $\ell = 0.05$.

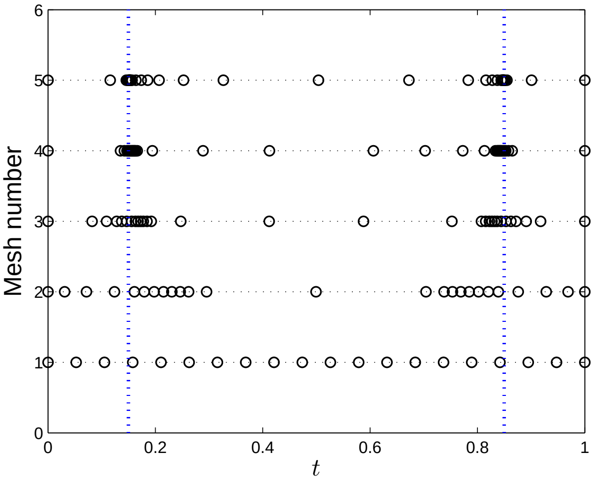

Figure 10: Mesh refinement, DENMRA, $\ell = 0.05$.

The mesh refinement histories of the two algorithms for the case with $\ell = 0.05$ are shown in Figs. 9-10. In these figures, the vertical dotted lines indicate the points of discontinuities in the analytical solution (at $t = 0.15$ and $t = 0.85$). As can be seen from Fig. 10, when DENMRA is used to solve this problem, the grid points get denser around the two points with discontinuities in the control derivative after each iteration, thus providing a better resolution. The mesh refinement scheme in SOCS is based on the integration error, and allocates more points on the two intervals $[0, 0.15]$ and $[0.85, 1]$ where the absolute value of $\dot{u}^*$ is large, but beyond this, the discontinuities in control did not receive any additional special treatment. As a result of this mesh refinement procedure, SOCS always keeps the points from the previous mesh, and hence tends to generate a larger mesh size. By solving this problem with different values of ℓ, it was confirmed that, for this problem, the mesh generated by DENMRA always provides better resolution around the points of discontinuities.

Table 1: Comparison of precision and optimality.

ℓ	Algorithm	N	$\lvert J - J^* \rvert$	$\lVert u_i - u^*(t_i) \rVert_\infty$
0.04	SOCS	99	7.5e-5	4.2e-3
	DENMRA-ρ_κ	40	8.9e-7	4.4e-5
0.08	SOCS	99	6.9e-6	1.4e-3
	DENMRA-ρ_κ	40	1.9e-8	4.8e-5
0.12	SOCS	50	9.6e-5	3.9e-3
	DENMRA-ρ_κ	40	1.2e-9	1.0e-5
0.16	SOCS	50	7.2e-5	1.8e-2
	DENMRA-ρ_κ	40	2.7e-13	5.8e-6

Comparison in Terms of Resolution

By "resolution" here we mean not only the ability of an algorithm to capture the discontinuities in the control history or its higher order derivatives using a locally denser grid, but also the ability to distinguish adjacent points of discontinuity.

(1) When $\ell \geq 1/6$, the optimal control $u^*(t)$ is either constant or smooth, both DENMRA and SOCS converge to the theoretical solution.

(2) When $\ell < 1/6$, the optimal control $u^*(t)$ contains two corners. It is challenging to

distinguish these corners when ℓ tends to zero or 1/6: in the former case, the corners are very close to the endpoints of the mesh, and the fast change of control between the corner and the corresponding end point makes it more difficult to obtain an accurate solution; in the second case, the two points of discontinuity tend to merge, which makes them difficult to distinguish.

Table 2: Comparison of resolution.

Algorithm	ℓ	ID	Δt	$\lvert J - J^*\rvert$	$\lVert u_i - u^*(t_i)\rVert_\infty$
SOCS	0.025	D1	0.075	8.2×10^{-4}	8.5×10^{-3}
	0.153	D2	0.082	2.8×10^{-5}	8.5×10^{-3}
DENMRA-ρ_κ	0.014	D1	0.042	7.3×10^{-9}	1.7×10^{-4}
	0.1662	D2	0.0028	1.9×10^{-9}	9.0×10^{-4}
D1: the smallest ℓ keeping$\lVert u_i - u^*(t_i)\rVert_\infty \leq 10^{-2}$ without algorithm failure.					
D2: the largest ℓ keeping $\lVert u_i - u^*(t_i)\rVert_\infty \leq 10^{-2}$ while separating the discontinuities					

The resolution test results are listed in Table 2. Both algorithms were able to gradually decrease ℓ until $\lVert u_i - u^*(t_i)\rVert_\infty \leq 10^{-2}$ without inducing any algorithm failure. The resolution is denoted by Δt. When $\ell \to 0$, $\Delta t = 3\ell$, where Δt is the distance between the discontinuities and the nearby endpoints of the mesh. When $\ell \to 1/6$, $\Delta t = 1 - 6\ell$, which is the distance between the two points of discontinuity. In both cases, a smaller Δt means a better resolution. For all test cases, DENMRA terminates with 40 points, SOCS starts from 50 points, and the final mesh sizes have 83 points when $\ell = 0.025$, and 50 points when $\ell = 0.162$. As shown in Table 2, DENMRA provides sharper resolution than SOCS while preserving the accuracy of the solution.

2.6.2 Hypersensitive Problem

This problem minimizes the cost function

$$J = \int_0^{t_f} \left(x^2(t) + u^2(t)\right) \mathrm{d}t,$$

subject to the differential constraint

$$\dot{x} = -x^3 + u,$$

and endpoint state constraints $x(0) = 1$, $x(t_f) = 1.5$. For large values of t_f, the solution of this hypersensitive problem has a three-segment structure with two boundary layers [102], namely, a "take-off, cruise and landing" structure. The "cruise" phase is determined by the cost function and the system dynamics,while the "take-off" and "landing" phases are determined by the boundary conditions, cost function, system dynamics, and the requirement to reach the cruise phase.

As pointed out in Ref. [102], the key to solving hypersensitive problems using direct methods is to use a denser grid during the boundary layers—"take-off" and "landing" phases— in which the state changes fast; a nonuniform mesh is imperative for the solution of this problem with large values of t_f. The hypersensitive problem with large t_f is suitable for testing the robustness of mesh refinement algorithms, because the length of the "cruise" phase increases with respect to t_f, which makes it more difficult to allocate enough grid points to the two boundary layers. We solved this problem for various values of t_f using both SOCS and DENMRA. Observing that the boundary layer is characterized by a large absolute value of the derivative of control, we used the density function $f(t) = |\dot{u}(t)|^{\frac{1}{2}}$ to capture these boundary layers during mesh generation in DENMRA.

SOCS was started from a mesh containing 150 points, and the maximum number of mesh refinements was set at 15. DENMRA started from a uniform mesh containing 25 points, with a maximum number of 15 mesh refinement iterations and a maximum mesh size of $N_{\max} = 100$. The problem was solved on the same computer as in the previous example. The results are summarized below.

In our numerical experiments, when ρ_κ is used for mesh generation and refinement, DENMRA failed to allocate enough points at both ends of the mesh, and did not converge for large values of t_f. In contrast, the use of the density finction $f(t) = |\dot{u}(t)|^{\frac{1}{2}}$ captures a larger region of the two boundary layers. Figure 11 shows the result of DENMRA using the f density function for $t_f = 1 \times 10^5$. As can be seen from the figure, the majority of the grid points are successfully allocated inside the two boundary layers.

Both SOCS and DENMRA were challenged by solving this hypersensitive problem for t_f as large as possible. To estimate the maximum solvable value of t_f, each algorithm was

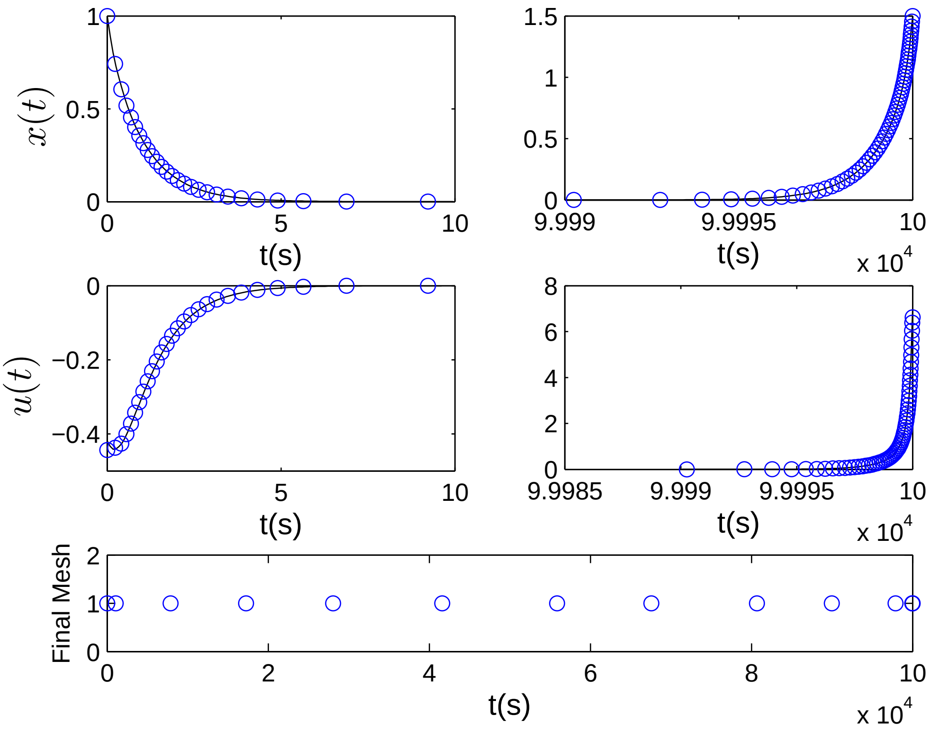

Figure 11: DENMRA solution, $t_f = 100,000$.

used to solve the hypersensitive problem for an increasing sequence of t_f values starting from $t_f = 100$. Numerical results showed that the optimal value $J^* \approx 6.724$. If the problem was successfully solved with the final objective value $J < 7$, then t_f was updated as $t_f = t_f + \Delta t_f$, where $\Delta t_f = 10^N$ if $10^N \leq t_f < 10^{N+1}$, for some positive integer N, and the problem was solved again with the new t_f. This process was repeated until $J \geq 7$. The results are shown in Table 3. As shown in the table, DENMRA exhibited good robustness by solving the hypersensitive problem for large values of t_f, which is attributed to its ability to redistribute the grid points to the boundary layers even with the presence of very long "cruise" phases. As a matter of fact, DENMRA was able to provide a solution up to a maximum value of $t_f = 2 \times 10^6$, whereas SOCS was limited to a maximum value of $t_f = 30,000$.

The optimality of SOCS and DENMRA is shown in Table 4. It was found that the optimality of the results obtained by DENMRA deteriorates when t_f is very large, while the optimality of the SOCS solution is consistent within the range of t_f values it can solve. The mesh refinement histories of two algorithms are similar, except for the fact that the mesh generated by SOCS contains many more grid points.

Table 3: Hypersensitive problem, robustness test.

Algorithm	t_f	N_{Iter}	N_f	J
SOCS	30,000	15	475	6.7241
DENMRA-f	2×10^6	15	100	6.8211

Table 4: Hypersensitive problem, optimality test.

Algorithm	t_f	N_{Iter}	N_f	J
SOCS	2×10^2	11	1020	6.7241
	2×10^3	14	1201	6.7241
	2×10^4	15	1014	6.7241
DENMRA-f	2×10^2	13	100	6.7240
	2×10^3	13	100	6.7240
	2×10^4	15	100	6.7239

In Ref. [24], a density (monitor) function of the form

$$\varphi(x, u) = \left(\alpha + \sum_{i=1}^{n} \beta_i g_i(x, u) \right)^{1/2},$$

where $g_i(x, u)$ is the i^{th} component of the system dynamics, and α and β_i are constants to be adjusted, was used to initialize SOCS for solving the hypersensitive problem. This "arc length" monitor function was also tested for mesh refinement. It was found that when DENMRA uses this arc length monitor function, the maximum solvable t_f value is 10,000. A density function providing an equidistribution along the arc length of the graph of the system state is not therefore the best choice for mesh refinement for this specific problem.

2.6.3 Optimal Aircraft Landing Trajectory with Limited Thrust

In this example DENMRA was used to investigate several optimal landing scenarios for a DC9-30 commercial aircraft. The equations of motion are as follows [45]:

$$\begin{aligned}
\dot{x} &= v\cos\gamma\cos\psi, && (34)\\
\dot{y} &= v\cos\gamma\sin\psi, && (35)\\
\dot{z} &= v\sin\gamma, && (36)\\
\dot{v} &= \frac{1}{m}\left(T\cos\alpha - D(\alpha, v, z)\right) - g\sin\gamma, && (37)\\
\dot{\gamma} &= \frac{1}{mv}(T\sin\alpha + L(\alpha, v, z)\cos\phi) - \frac{g}{v}\cos\gamma, && (38)\\
\dot{\psi} &= -\frac{1}{mv\cos\gamma}L(\alpha, v, z)\sin\phi, && (39)
\end{aligned}$$

where the variables are

m:	mass,	v:	airspeed,	ψ:	heading angle,	γ:	path angle,
x:	position(east),	y:	position(north),	z:	altitude,	T:	thrust,
ϕ:	bank angle,	L:	lift force,	D:	drag force.	α:	angle of attack.

The lift and drag forces are functions of α and v, as described in the following equations:

$$\begin{aligned}
D(\alpha, v, z) &= Q(v, z)SC_D(\alpha),\\
L(\alpha, v, z) &= Q(v, z)SC_L(\alpha),
\end{aligned}$$

where $Q(v, z)$ is the dynamic pressure given by $Q(v, z) = \frac{1}{2}\rho(z)v^2$, $\rho(z)$ is the air density at altitude z, and S is the wing surface area. The lift and drag coefficients $C_L(\alpha)$ and $C_D(\alpha)$ can be calculated, as usual, by the following equations,

$$\begin{aligned}
C_L(\alpha) &= C_{L_0} + C_{L_\alpha}\alpha,\\
C_D(\alpha) &= C_{D_0} + KC_L^2(\alpha),
\end{aligned}$$

where C_{L_0} is the lift coefficient at zero angle of attack, and C_{L_α} is the lift coefficient slope. The coefficient C_{D_0} accounts for the drag of the whole aircraft, and the second term in $C_D(\alpha)$ accounts for the induced drag, specifically, $K = 1/(0.95e\pi \text{Æ})$, where e is the efficiency factor, which is corrected by 0.95 for the assumed landing configuration. Æ is the aspect ratio of the aircraft defined by $\text{Æ} = b^2/S$, where b is the wing span. In the current model, it is assumed that the mass of the aircraft m is constant. Since large civil aircraft usually fly at a high altitude, a realistic atmospheric model is used for solving the optimal landing problem [91]. The values of the parameters in the former equations are given in Table 2.6.3, where $T_{\max}$ is the maximum thrust.

Table 5: Parameters for the DC9-30.

m	49.940 kg	g	9.8kgm/s^2	ρ_0	1.225kg/m^3
S	112 m^2	C_{L_α}	4.2	C_{L_0}	0.4225
$T_{\max}$	137.81kN	K	0.0459	C_{D_0}	0.0197

Finding a good initial guess turns out to be challenging for this problem. Large civil aircraft usually cruise at an altitude of around $10,000\,\mathrm{m}$, where the air density is about $0.4140\,\mathrm{kg/m^3}$, which is only 33.8% of the value at sea level. Constantly changing air density during the landing process makes it difficult for the NLP solver to converge, especially when the initial guess is not good. Our numerical experiments have shown that an arbitrary affine or constant initial guess of states and controls works satisfactorily for the constant air density scenario, but it is difficult to find a converging initial guess for the altitude-varying air density scenario. Experience may provide good intuition about the shape of the optimal path but, in general, this is not so for the velocity profile and histories of controls to fly along such a path. Besides, if the initial guess of the states and controls are not dynamically consistent, then this initial guess may also lead to the failure of the solver for a sensitive problem.

For the sensitive cases in which the landing problem with a realistic air density model and constant initial guess failed to converge, the same problem with constant air density was solved, and the result was subsequently used as an initial guess for solving the problem with the altitude-varying air density model again. For all sensitive cases which have been tested, this procedure led to convergent solutions.

When an aircraft looses thrust because of engine failure, fuel depletion, or any other unforeseen problem, a reasonable option to guarantee the safety of the passengers is to land the aircraft at a nearby airport as soon as possible. This can be treated as a minimum-time optimal control problem with fixed boundary conditions. In this section we consider two cases for the zero-thrust, minimum-time landing problem. For both cases the aircraft loses power at an altitude of $z = 10\,\text{km}$, cruise speed $v = 240\,\text{m/s}$, and flight path angle $\gamma = 0°$, and needs to land at a nearby airport using only the angle of attack α and the bank angle ϕ as control inputs. We considered four landing scenarios to demonstrate DENMRA's capability for solving the aircraft landing problem, with different run way position and orientation. The runway layouts are listed in Table 6.

Table 6: Runway layout table.

	x_f(km)	y_f(km)	z_f(km)	ψ_f(°)
Runway No.1	60	50	0	350
Runway No.2	60	-30	0	30
Runway No.3	-60	-45	0	150
Runway No.4	-70	45	0	310

The zero-thrust emergency landing scenarios in Table 6 were solved by minimizing the final arrival time t_f in DENMRA. The corresponding landing trajectories are shown in Fig. 12 and Fig. 13. The same landing problems were also formulated using an industrial-strength numerical optimal control software—Sparse Optimal Control Software (SOCS), however, no convergent solution was found.

2.7 Summary

A new mesh refinement method is proposed, which is based on a mesh density function that determines the mesh point distribution. By using an appropriate density function, the proposed DENsity function-based Mesh Refinement Algorithm (DENMRA) generates a non-uniform mesh by suitably allocating the grid points over the whole time interval, putting emphasis on the points of discontinuity of the control variables or on the non-smoothness of the state variables. The grid point allocation process is completely automatic. Two density functions are also introduced, one based on the local curvature of the graph of the intermediate solution and the other based on the first derivative of the control variable. The density function can also be chosen as the integration error, leading to the mesh refinement

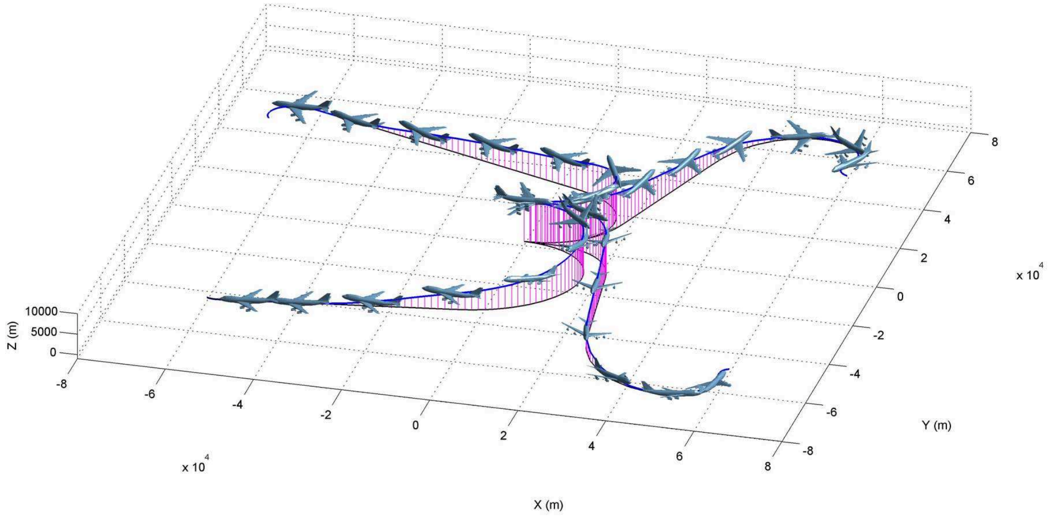

Figure 12: 3D glider landing trajectories generated in DENMRA.

scheme proposed in Ref. [110]. Numerical results have shown that DENMRA automatically maintains an appropriate local level of discretization over the whole control and state time histories for different problems. The grid generation is very simple and easy to implement, while still maintaining high numerical accuracy for the overall solution. The numerical examples also demonstrated the importance of choosing an appropriate density function that captures the smoothness irregularities in the intermediate solution for best accuracy, optimality and robustness, especially when solving challenging problems.

Another attractive advantage of DENMRA is that it can be used to distribute a fixed number of grid points so as to maximize the accuracy of the final solution. In terms of real-time (or close to real-time) applications, this may be of greater interest, since the number of decision variables and constraints of the resulting nonlinear optimization problem is related to the number of grid points used. If the computational resources impose limitations on the number of constraints that can be handled during each iteration, it makes sense to limit the size of the optimization problem by keeping the number of grid points fixed. This can be easily achieved using the proposed algorithm.

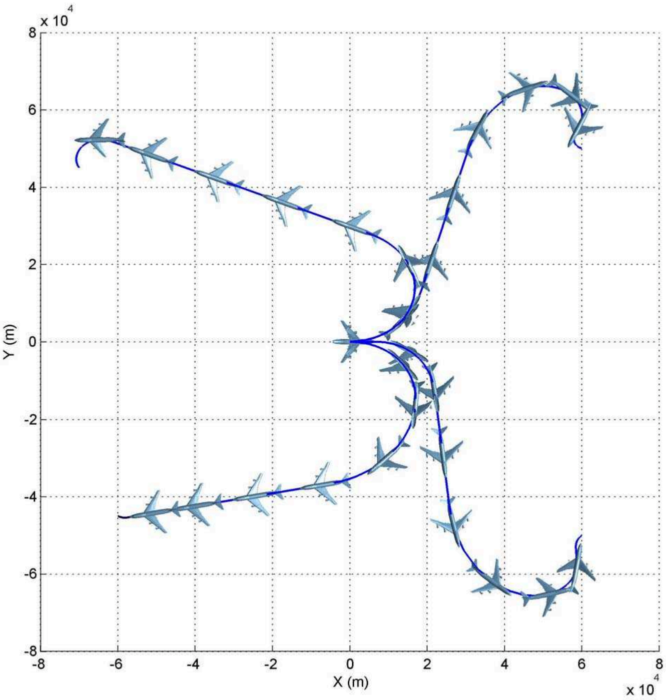

Figure 13: 2D projection of glider landing trajectories generated in DENMRA.

CHAPTER III

PATH SMOOTHING USING ITERATIVE QUADRATIC PROGRAMMING

Path smoothness is a desirable property for the precise tracking of such a path by mechanical systems. With a discontinuity in the first derivative, the path exhibits corner points, at which points the system must stop completely for precise tracking. Some other systems require even higher order smoothness of the path in order to be tracked exactly. However, the paths generated by most geometric path planning methods often do not have the desired smoothness characteristics, and need to be smoothed. In this chapter, we consider the problem of smoothing a three-dimensional geometric path, which is proposed as Problem 1.2.3 in Chapter 1. The method introduced in this chapter works equally well as a post-processing technique for various geometric path planning methods generating smooth and collision-free paths.

3.1 Background

Let $\mathbf{r}(s) = \{(x(s), y(s), z(s)) : 0 \leq s \leq s_f\} \in \mathbb{R}^3$ represent a parameterized path to be followed by a vehicle, where s is the arc length coordinate. While obstacles pose constraints on the image of $\mathbf{r}$, vehicle dynamics place constraints on its higher order derivatives. The challenge of smooth path planning lies in the coordination between these two different layers of constraints.

The most commonly used high order path constraint is the curvature constraint. Although Dubinsvehicle paths address curvature constraints, the result is optimal only for a vehicle having constant speed [44]. For more realistic vehicles with acceleration/deceleration capability, curvature has greater influence on both the optimality and feasibility of the path. For example, the traveling time along a longer path with small maximum curvature can be shorter than that along a shorter path with large maximum curvature [41]. Besides, a path may be infeasible due to a “minor” violation of the curvature constraint, such that the feasibility can be recovered by a small local variation of the path. Hence, smoothing a path via local curvature regulation may lead to improvement in terms of feasibility and optimality.

A discontinuity in the curvature profile implies an instantaneous change of the steering wheel angle for a car-like vehicle or the bank angle/angle of attack for a fixed-wing aircraft, both of which require (theoretically) infinite control force. Therefore, the curvature of the path should be at least continuous for practical applications. For this reason, clothoid arcs have been used for continuous-curvature path planning based on the Dubins' path prototype [108, 48, 15]. Reference [97] used analytical splines and heuristics for smooth path generation. Reference [134] proposed a path planning algorithm which generates a smooth path by smoothing out the corners of a linear path prototype using Bézier curves based on analytic expressions. Although all these methods can generate paths with continuous curvature, obstacle avoidance is not guaranteed by these methods per se, and can only be done in an ad hoc manner.

One approach for smooth path planning in the presence of obstacles is to use a "channel" or "corridor," which is selected a priori, such that it does not intrude any of the obstacles. A smooth path is then found within the channel such that it is collision-free. For instance, [14] introduced a method for generating curvature-bounded paths in rectangular channels; reference [21] proposed a method for constructing bounded curvature paths traversing a constant width region in the plane, called corridors, and reference [68] introduced a method for generating smooth two-dimensional paths within two-dimensional bounding envelops using B-spline curves. A nonlinear optimization scheme is used to design collision-free and curvature-continuous paths in [85].

Next, we will present an iterative method for smoothing a three-dimensional path subject to curvature and obstacle clearance constraints. The proposed method minimizes the weighted L_2 norm of the curvature along the path, which is analogous to the strain energy stored in a deflected elastic beam. During the optimization process, a sequence of obstacle-free perturbations are generated along the normal direction of the path. This idea is similar to the perturbation technique in [53] for eliminating noise in GPS measurement data. When combined with other geometric path planning algorithms that provide the initial collision-free path prototype, the proposed method generates collision-free paths under length and localized curvature constraints.

3.2 Curve Representation

Instead of dealing with a curve (path) in the infinite dimensional space, we reduce the dimensionality of the problem by considering a finite number of characteristic nodes on the curve, and represent the path using a cubic spline passing through those nodes.

To this end, suppose that the path is defined in parametric form as $\mathbf{r}(s) = [x(s), y(s), z(s)]^{\mathrm{T}}$, parameterized by its arc length s. The curve passes through N characteristic nodes $\mathbf{r}_1$, $\mathbf{r}_2, \ldots,$ $\mathbf{r}_N \in \mathbb{R}^3$ at $s_1, s_2, \ldots, s_N$, respectively, i.e., $\mathbf{r}(s_i) = \mathbf{r}_i = (x_i, y_i, z_i)$, $i = 1, 2, \ldots, N$, where $s_1 = 0$ and $s_N = s_f$. These characteristic nodes are chosen such that they are equally spanned along the path length with $s_2 - s_1 = s_3 - s_2 = \cdots = s_N - s_{N-1} = \Delta s$. We introduce the notation $\mathbf{r}\{k\}$ to denote the k^{th} component of $\mathbf{r}$, i.e., $\mathbf{r}_i\{1\} = x_i$, $\mathbf{r}_i\{2\} = y_i$, and $\mathbf{r}_i\{3\} = z_i$ for $i = 1, \ldots, N$.

In the smoothing process, the first and the last nodes are fixed, and the smoothing of the path is equivalent to the deployment of the other $N - 2$ characteristic nodes subject to certain smoothness criteria.

Because we require that the path has continuous second derivative, cubic splines are used for the interpolation between the nodes. Specifically, for every $i = 2, \ldots, N - 2$, a group of four adjacent nodes (the $i - 1^{\mathrm{th}}$, i^{th}, $i + 1^{\mathrm{th}}$, and $i + 2^{\mathrm{th}}$ nodes) is used to construct a local cubic curve for the interpolation between the i^{th} and the $i + 1^{\mathrm{th}}$ node, as shown in Fig. 14. A local path length coordinate τ is assigned to each group of nodes such that $\tau = 0$ for the $i - 1^{\mathrm{th}}$ node, $\tau = 1/3$ for the i^{th} node, $\tau = 2/3$ for the $i + 1^{\mathrm{th}}$ node, and $\tau = 1$ for the $i + 2^{\mathrm{th}}$ node.

With a slight abuse of notation, the coordinate x of the path between $\tau = 1/3$ and $\tau = 2/3$ is given by a cubic interpolative spline passing through the x components of $\mathbf{r}_{i-1}$, $\mathbf{r}_i$, $\mathbf{r}_{i+1}$, $\mathbf{r}_{i+2}$ as:

$$x(\tau; i) = a_{x_i}\tau^3 + b_{x_i}\tau^2 + c_{x_i}\tau + d_{x_i}$$

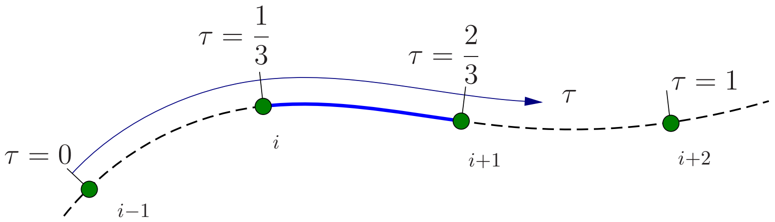

Figure 14: Cubic spline interpolation.

where a_{x_i}, b_{x_i}, c_{x_i} and d_{x_i} are constants for which the following constraints must be satisfied:

$$x_{i-1} = x(0;i) = d_{x_i}, \tag{40}$$

$$x_i = x(\tfrac{1}{3};i) = \frac{a_{x_i}}{27} + \frac{b_{x_i}}{9} + \frac{c_{x_i}}{3} + d_{x_i}, \tag{41}$$

$$x_{i+1} = x(\tfrac{2}{3};i) = \frac{8a_{x_i}}{27} + \frac{4b_{x_i}}{9} + \frac{2c_{x_i}}{3} + d_{x_i}, \tag{42}$$

$$x_{i+2} = x(1;i) = a_{x_i} + b_{x_i} + c_{x_i} + d_{x_i}, \tag{43}$$

from which we have

$$\begin{bmatrix} a_{x_i} \\ b_{x_i} \\ c_{x_i} \\ d_{x_i} \end{bmatrix} = \mathbf{G} \begin{bmatrix} x_{i-1} \\ x_i \\ x_{i+1} \\ x_{i+2} \end{bmatrix},$$

where

$$\mathbf{G} = \begin{bmatrix} -\frac{9}{2} & \frac{27}{2} & -\frac{27}{2} & \frac{9}{2} \\ 9 & -\frac{45}{2} & 18 & -\frac{9}{2} \\ -\frac{11}{2} & 9 & -\frac{9}{2} & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Therefore, we have the following expression

$$x(\tau;i) = \begin{bmatrix} \tau^3 & \tau^2 & \tau & 1 \end{bmatrix} \begin{bmatrix} a_{x_i} \\ b_{x_i} \\ c_{x_i} \\ d_{x_i} \end{bmatrix} = \begin{bmatrix} \tau^3 & \tau^2 & \tau & 1 \end{bmatrix} \mathbf{G} \begin{bmatrix} x_{i-1} \\ x_i \\ x_{i+1} \\ x_{i+2} \end{bmatrix}. \tag{44}$$

Similarly, the expressions for $y(\tau;i)$ and $z(\tau;i)$ can also be derived, allowing $\mathbf{r}(\tau;i)$ to be given by the expression

$$\mathbf{r}(\tau; i) = \begin{bmatrix} \tau^3 & \tau^2 & \tau & 1 \end{bmatrix} \mathbf{G} \begin{bmatrix} \mathbf{r}_{i-1} \\ \mathbf{r}_i \\ \mathbf{r}_{i+1} \\ \mathbf{r}_{i+2} \end{bmatrix}. \tag{45}$$

By taking derivatives of (45), we have

$$\frac{\mathrm{d}\mathbf{r}(\tau; i)}{\mathrm{d}\tau} = \begin{bmatrix} 3\tau^2 & 2\tau & 1 & 0 \end{bmatrix} \mathbf{G} \begin{bmatrix} \mathbf{r}_{i-1} \\ \mathbf{r}_i \\ \mathbf{r}_{i+1} \\ \mathbf{r}_{i+2} \end{bmatrix}, \tag{46}$$

and

$$\frac{\mathrm{d}^2\mathbf{r}(\tau; i)}{\mathrm{d}\tau^2} = \begin{bmatrix} 6\tau & 2 & 0 & 0 \end{bmatrix} \mathbf{G} \begin{bmatrix} \mathbf{r}_{i-1} \\ \mathbf{r}_i \\ \mathbf{r}_{i+1} \\ \mathbf{r}_{i+2} \end{bmatrix}. \tag{47}$$

These derivatives are proportional to the derivatives of the path with respect to the path coordinate s. Let $\mathbf{r}'$ and $\mathbf{r}''$ denote, respectively, the first and second derivatives of the path $\mathbf{r}$ with respect to s. Then it can be easily shown that for $s \in [s_i, s_{i+1}]$, $i = 2, \ldots, N-2$,

$$\mathbf{r}'(s) = \frac{1}{3\Delta s} \frac{\mathrm{d}\mathbf{r}(\tau; i)}{\mathrm{d}\tau}, \quad \text{and} \quad \mathbf{r}''(s) = \frac{1}{9\Delta s^2} \frac{\mathrm{d}^2\mathbf{r}(\tau; i)}{\mathrm{d}\tau^2},$$

where $\tau = (s - s_i)/3\Delta s$.

At any point $s \in [s_0, s_f]$, the tangent vector $\mathbf{t}(s)$ is given by $\mathbf{r}''(s)$. The normal vector $\mathbf{n}(s)$ is given by $\mathbf{n}(s) = \mathbf{r}''(s)/\|\mathbf{r}''(s)\|$, and the binormal vector $\mathbf{b}(s)$ is given by $\mathbf{b}(s) = \mathbf{t} \times \mathbf{n}$, where '×' denotes the cross product.

We also define

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{r}_{i-1} \\ \mathbf{r}_i \\ \mathbf{r}_{i+1} \\ \mathbf{r}_{i+2} \end{bmatrix}, \quad \mathbf{R}_i\{k\} = \begin{bmatrix} \mathbf{r}_{i-1}\{k\} \\ \mathbf{r}_i\{k\} \\ \mathbf{r}_{i+1}\{k\} \\ \mathbf{r}_{i+2}\{k\} \end{bmatrix}, \quad k = 1, 2, 3; \quad i = 2, \ldots, N-2.$$

3.3 Path Variation

Consider a specific variation of the path $\mathbf{r}(s)$ by perturbing the path at the characteristic nodes along the associated "normal directions" $\mathbf{n}_i = \mathbf{n}(s_i)$ and "binormal directions" $\mathbf{b}_i = \mathbf{b}(s_i)$. Note that when $\mathbf{r}''_i = 0$, then $\mathbf{n}_i$ is not well-defined. In this case, an arbitrary unit vector perpendicular to $\mathbf{r}'_i$ is used as the normal vector.

Let δ_i denote the magnitude of variation along the direction of $\mathbf{n}_i$ at the i^{th} node $\mathbf{r}_i$, and let λ_i denote the magnitude of variation along the $\mathbf{b}_i$ direction. The nodes of the perturbed path are given by

$$\tilde{\mathbf{r}}_i = \mathbf{r}_i + \mathbf{n}_i\delta_i + \mathbf{b}_i\lambda_i = \mathbf{r}_i + (\mathbf{n}_i\{1\}, \mathbf{n}_i\{2\}, \mathbf{n}_i\{3\})\,\delta_i + (\mathbf{b}_i\{1\}, \mathbf{b}_i\{2\}, \mathbf{b}_i\{3\})\,\lambda_i. \tag{48}$$

Let $X = [\delta_1, \ldots, \delta_N, \lambda_1, \ldots, \lambda_N]^{\mathrm{T}}$, which is the collection of decision variables, and define $\mathbf{X}_i = [\delta_{i-1}, \delta_i, \delta_{i+1}, \delta_{i+2}]^{\mathrm{T}}$, and $\mathbf{Y}_i = [\lambda_{i-1}, \lambda_i, \lambda_{i+1}, \lambda_{i+2}]^{\mathrm{T}}$. Also define

$$\mathbf{N}_{i,\{k\}} = \begin{bmatrix} \mathbf{n}_{i-1}\{k\} & 0 & 0 & 0 \\ 0 & \mathbf{n}_i\{k\} & 0 & 0 \\ 0 & 0 & \mathbf{n}_{i+1}\{k\} & 0 \\ 0 & 0 & 0 & \mathbf{n}_{i+2}\{k\} \end{bmatrix}, \quad k = 1, 2, 3.$$

$$\mathbf{B}_{i,\{k\}} = \begin{bmatrix} \mathbf{b}_{i-1}\{k\} & 0 & 0 & 0 \\ 0 & \mathbf{b}_i\{k\} & 0 & 0 \\ 0 & 0 & \mathbf{b}_{i+1}\{k\} & 0 \\ 0 & 0 & 0 & \mathbf{b}_{i+2}\{k\} \end{bmatrix}, \quad k = 1, 2, 3.$$

Then we have

$$\tilde{\mathbf{R}}_i\{k\} = \mathbf{R}_i\{k\} + \mathbf{N}_{i,\{k\}}\mathbf{X}_i + \mathbf{B}_{i,\{k\}}\mathbf{Y}_i, \quad k = 1, 2, 3; \quad i = 2, \ldots, N-2.$$

The perturbed path is obtained using a cubic curve interpolation at the perturbed characteristic points $\tilde{\mathbf{r}}_i$, $i = 1, \ldots, N$.

3.4 Quadratic Programming Formulation for the Path Smoothing Problem

In this section we formulate the path smoothing problem as a quadratic program, which approximately minimizes the L_2 norm of the curvature profile, while maintaining the path length and local curvature constraints, boundary conditions and collision-avoidance.

Definition 3.4.1. The problem

$$\min J(x), \quad x \in \mathcal{D} \subseteq \mathbb{R}^n$$

is a *linear-quadratic mathematical programming problem* (or a *quadratic program* for short), if J is a linear-quadratic function, that is,

$$J(x) = \frac{1}{2} x^{\mathrm{T}} H x + F^{\mathrm{T}} x + c, \tag{49}$$

where $H = H^{\mathrm{T}} \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^n$, and $c \in \mathbb{R}$, and $\mathcal{P}$ is a convex polyhedron, namely $\mathcal{P} = \{x \in \mathbb{R}^n : Ax \leq b\}$, where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.

Note that $\mathcal{P}$ is a convex set. A linear quadratic programming problem is a special case of a convex optimization problem when H is a positive semi-definite matrix. Both can be solved very efficiently using numerical methods.

3.4.1 A Quadratic Cost Function

The L_2 norm of the signed curvature function of the perturbed path is defined by

$$\|\tilde{\kappa}\|_2 \triangleq \left(\int_{s_0}^{s_f} w(s) \tilde{\kappa}^2(s) \mathrm{d}s \right)^{\frac{1}{2}}, \tag{50}$$

where $w : [s_0, s_f] \to \mathbb{R}_+ \backslash \{0\}$ is a weight function. With the cubic spline curve representation of the path, the integral in (50) can be computed analytically. Specifically, note that $\mathbf{r}(s)$ as parameterized by its path length coordinate has a unit first derivative, and its curvature is the magnitude of the acceleration, i.e.,

$$|\kappa(s)| = \|\mathbf{r}''(s)\|.$$

To obtain an analytic expression of (50), we may assume that $w(s)$ is a piecewise constant function with $w(s) = w_i$ for $s \in [s_i, s_{i+1})$, $i = 1, \ldots, N-1$ and $w(s_f) = w_{N-1}$. Therefore,

$$\begin{aligned} \|\tilde{\kappa}\|_2^2 &= \int_{s_0}^{s_f} w(s) \tilde{\kappa}^2(s) \mathrm{d}s \\ &= \int_{s_0}^{s_f} w(s) \langle \tilde{\mathbf{r}}''(s), \tilde{\mathbf{r}}''(s) \rangle \mathrm{d}s \\ &= w_1 \int_{s_1}^{s_2} \langle \tilde{\mathbf{r}}''(s), \tilde{\mathbf{r}}''(s) \rangle \mathrm{d}s + \sum_{i=2}^{N-2} w_i \int_{s_i}^{s_{i+1}} \langle \tilde{\mathbf{r}}''(s), \tilde{\mathbf{r}}''(s) \rangle \mathrm{d}s + w_{N-1} \int_{s_{N-1}}^{s_N} \langle \tilde{\mathbf{r}}''(s), \tilde{\mathbf{r}}''(s) \rangle \mathrm{d}s \end{aligned}$$

For $i = 2, \ldots, N-2$,

$$
\begin{aligned}
&\int_{s_i}^{s_{i+1}} \big\langle \tilde{\mathbf{r}}''(s), \tilde{\mathbf{r}}''(s) \big\rangle \mathrm{d}s \\
&= \frac{1}{81\Delta s^4} \int_{\frac{1}{3}}^{\frac{2}{3}} \begin{bmatrix} 6\tau & 2 & 0 & 0 \end{bmatrix} \mathbf{G}\tilde{\mathbf{R}}_i\tilde{\mathbf{R}}_i^{\mathrm{T}}\mathbf{G}^{\mathrm{T}} \begin{bmatrix} 6\tau \\ 2 \\ 0 \\ 0 \end{bmatrix} \mathrm{d}\tau \\
&= \frac{1}{81\Delta s^4} \sum_{k=1}^{k=3} \int_{\frac{1}{3}}^{\frac{2}{3}} \begin{bmatrix} 6\tau & 2 & 0 & 0 \end{bmatrix} \mathbf{G}\tilde{\mathbf{R}}_i\{k\}\tilde{\mathbf{R}}_i^{\mathrm{T}}\{k\}\mathbf{G}^{\mathrm{T}} \begin{bmatrix} 6\tau \\ 2 \\ 0 \\ 0 \end{bmatrix} \mathrm{d}\tau \\
&= \frac{1}{81\Delta s^4} \sum_{k=1}^{k=3} \tilde{\mathbf{R}}_i^{\mathrm{T}}\{k\}\mathbf{G}^{\mathrm{T}} \int_{\frac{1}{3}}^{\frac{2}{3}} \begin{bmatrix} 6\tau \\ 2 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 6\tau & 2 & 0 & 0 \end{bmatrix} \mathrm{d}\tau \mathbf{G}\tilde{\mathbf{R}}_i\{k\} \\
&= \frac{1}{81\Delta s^4} \sum_{k=1}^{k=3} \tilde{\mathbf{R}}_i^{\mathrm{T}}\{k\}\mathbf{G}^{\mathrm{T}} \begin{bmatrix} \frac{28}{9} & 2 & 0 & 0 \\ 2 & \frac{4}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{G}\tilde{\mathbf{R}}_i\{k\} \\
&= \frac{1}{81\Delta s^4} \sum_{k=1}^{k=3} \tilde{\mathbf{R}}_i^{\mathrm{T}}\{k\}\mathbf{M}_i\tilde{\mathbf{R}}_i\{k\},
\end{aligned}
$$

where

$$
\mathbf{M}_i = \begin{bmatrix} 9 & -\frac{27}{2} & 0 & \frac{9}{2} \\ -\frac{27}{2} & 27 & -\frac{27}{2} & 0 \\ 0 & -\frac{27}{2} & 27 & -\frac{27}{2} \\ \frac{9}{2} & 0 & -\frac{27}{2} & 9 \end{bmatrix}.
$$

Similarly, we have

$$\int_{s_1}^{s_2} \langle \tilde{\mathbf{r}}''(s), \tilde{\mathbf{r}}''(s) \rangle \mathrm{d}s = \frac{1}{81\Delta s^4} \sum_{k=1}^{k=3} \tilde{\mathbf{R}}_2^{\mathrm{T}}\{k\}\mathbf{M}_1\tilde{\mathbf{R}}_2\{k\},$$

and

$$\int_{s_{N-1}}^{s_N} \langle \tilde{\mathbf{r}}''(s), \tilde{\mathbf{r}}''(s) \rangle \mathrm{d}s = \frac{1}{81\Delta s^4} \sum_{k=1}^{k=3} \tilde{\mathbf{R}}_{N-2}^{\mathrm{T}}\{k\}\mathbf{M}_{N-1}\tilde{\mathbf{R}}_{N-2}\{k\},$$

where

$$M_1 = \begin{bmatrix} 63 & -\frac{297}{2} & 108 & -\frac{45}{2} \\ \frac{297}{2} & 351 & -\frac{513}{2} & 54 \\ 108 & -\frac{513}{2} & 189 & -\frac{81}{2} \\ -\frac{45}{2} & 54 & -\frac{81}{2} & 9 \end{bmatrix},$$

and

$$M_{N-1} = \begin{bmatrix} 9 & -\frac{81}{2} & 54 & -\frac{45}{2} \\ -\frac{81}{2} & 189 & -\frac{513}{2} & 108 \\ 54 & -\frac{513}{2} & 351 & -\frac{297}{2} \\ -\frac{45}{2} & 108 & -\frac{297}{2} & 63 \end{bmatrix}$$

Hence, we have

$$\|\tilde{\kappa}\|_2^2 = \frac{1}{81\Delta s^4} \sum_{i=1}^{N-1} \sum_{k=1}^{k=3} \tilde{\mathbf{R}}_i^{\mathrm{T}}\{k\}\mathbf{M}_i\tilde{\mathbf{R}}_i\{k\}$$

According to equation (48), we have

$$\begin{aligned} \tilde{\mathbf{R}}_i^{\mathrm{T}}\{k\}\mathbf{M}_i\tilde{\mathbf{R}}_i\{k\} &= \left(\mathbf{R}_i^{\mathrm{T}}\{k\} + (\mathbf{N}_{i,\{k\}}\mathbf{X}_i + \mathbf{B}_{i,\{k\}}\mathbf{Y}_i)^{\mathrm{T}}\right)\mathbf{M}_i\left(\mathbf{R}_i\{k\} + \mathbf{N}_{i,\{k\}}\mathbf{X}_i + \mathbf{B}_{i,\{k\}}\mathbf{Y}_i\right) \\ &= \mathbf{R}_i^{\mathrm{T}}\{k\}\mathbf{M}_i\mathbf{R}_i\{k\} + 2\mathbf{R}_i^{\mathrm{T}}\{k\}\mathbf{M}_i\mathbf{N}_{i,\{k\}}\mathbf{X}_i + \mathbf{X}_i^{\mathrm{T}}\mathbf{N}_{i,\{k\}}\mathbf{M}_i\mathbf{N}_{i,\{k\}}\mathbf{X}_i \\ &\quad + 2\mathbf{R}_i^{\mathrm{T}}\{k\}\mathbf{M}_i\mathbf{B}_{i,\{k\}}\mathbf{Y}_i + \mathbf{Y}_i^{\mathrm{T}}\mathbf{B}_{i,\{k\}}\mathbf{M}_i\mathbf{B}_{i,\{k\}}\mathbf{Y}_i + 2\mathbf{Y}_i^{\mathrm{T}}\mathbf{B}_{i,\{k\}}\mathbf{M}_i\mathbf{N}_{i,\{k\}}\mathbf{X}_i. \end{aligned}$$

Because the term $\mathbf{R}_i^{\mathrm{T}}\{k\}\mathbf{M}_i\mathbf{R}_i\{k\}$ in the above expression is a constant independent of the variation $\mathbf{X}_i$, it suffices to consider the other terms only in the optimization. As a result, the minimization of (50) is equivalent to the minimization of the following cost function

$$\begin{aligned}
J(X) = &\sum_{i=2}^{N-2} w_i \sum_{k=1}^{k=3} \mathbf{X}_i^{\mathrm{T}} \mathbf{N}_{i,\{k\}} \mathbf{M}_i \mathbf{N}_{i,\{k\}} \mathbf{X}_i + 2\sum_{i=2}^{N-2} w_i \sum_{k=1}^{k=3} \mathbf{R}_i^{\mathrm{T}}\{k\}\mathbf{M}_i\mathbf{N}_{i,\{k\}}\mathbf{X}_i \\
&+ w_1 \sum_{k=1}^{k=3} \mathbf{X}_2^{\mathrm{T}} \mathbf{N}_{2,\{k\}}\mathbf{M}_1\mathbf{N}_{2,\{k\}}\mathbf{X}_2 + 2w_{N-1}\sum_{k=1}^{k=3} \mathbf{R}_{N-2}^{\mathrm{T}}\{k\}\mathbf{M}_{N-1}\mathbf{N}_{N-2,\{k\}}\mathbf{X}_{N-2} \\
&+ w_{N-1}\sum_{k=1}^{k=3} \mathbf{X}_{N-2}^{\mathrm{T}}\mathbf{N}_{N-2,\{k\}}\mathbf{M}_{N-1}\mathbf{N}_{N-2,\{k\}}\mathbf{X}_{N-2} + 2w_1\sum_{k=1}^{k=3} \mathbf{R}_2^{\mathrm{T}}\{k\}\mathbf{M}_1\mathbf{N}_{2,\{k\}}\mathbf{X}_2 \\
&+ \sum_{i=2}^{N-2} w_i \sum_{k=1}^{k=3} \mathbf{Y}_i^{\mathrm{T}}\mathbf{B}_{i,\{k\}}\mathbf{M}_i\mathbf{B}_{i,\{k\}}\mathbf{Y}_i + 2\sum_{i=2}^{N-2} w_i\sum_{k=1}^{k=3} \mathbf{R}_i^{\mathrm{T}}\{k\}\mathbf{M}_i\mathbf{B}_{i,\{k\}}\mathbf{Y}_i \\
&+ w_1\sum_{k=1}^{k=3} \mathbf{Y}_2^{\mathrm{T}}\mathbf{B}_{2,\{k\}}\mathbf{M}_1\mathbf{B}_{2,\{k\}}\mathbf{Y}_2 + 2w_{N-1}\sum_{k=1}^{k=3}\mathbf{R}_{N-2}^{\mathrm{T}}\{k\}\mathbf{M}_{N-1}\mathbf{B}_{N-2,\{k\}}\mathbf{Y}_{N-2} \\
&+ w_{N-1}\sum_{k=1}^{k=3} \mathbf{Y}_{N-2}^{\mathrm{T}}\mathbf{B}_{2,\{k\}}\mathbf{M}_{N-1}\mathbf{B}_{N-2,\{k\}}\mathbf{Y}_{N-1} + 2w_1\sum_{k=1}^{k=3}\mathbf{R}_2^{\mathrm{T}}\{k\}\mathbf{M}_1\mathbf{B}_{2,\{k\}}\mathbf{Y}_2 \\
&+ 2\sum_{i=2}^{N-2} w_i \sum_{k=1}^{k=3} \mathbf{X}_i^{\mathrm{T}}\mathbf{N}_{i,\{k\}}\mathbf{M}_i\mathbf{B}_{i,\{k\}}\mathbf{Y}_i, +2w_1\sum_{k=1}^{k=3}\mathbf{X}_2^{\mathrm{T}}\mathbf{N}_{2,\{k\}}\mathbf{M}_1\mathbf{B}_{2,\{k\}}\mathbf{Y}_2 \\
&+ 2w_{N-1}\sum_{k=1}^{k=3}\mathbf{X}_{N-2}^{\mathrm{T}}\mathbf{N}_{N-2,\{k\}}\mathbf{M}_{N-1}\mathbf{B}_{N-2,\{k\}}\mathbf{Y}_{N-2}.
\end{aligned}$$

which can be written in a more compact form as

$$J(X) = \frac{1}{2}X^{\mathrm{T}}\mathbf{H}X + \mathbf{F}X,$$

where $\mathbf{H} \in \mathbb{R}^{N\times N}$ and $\mathbf{F} \in \mathbb{R}^{1\times N}$, $X = [X_n^{\mathrm{T}}, X_b^{\mathrm{T}}]^{\mathrm{T}}$, $X_n, X_b \in \mathbb{R}^{N\times 1}$ are the vectors containing the magnitude of variations at each node along the normal and binormal directions, respectively. The details for the computation of matrices **H** and **F** are given in Appendix A.

3.4.2 Path Length Constraint

Because the length of the path affects the traveling time, it is desirable to have a constraint on the total length of the path. When a path is perturbed at each node along the normal and binormal directions, the total length of the path is not necessarily preserved—it could either increase or decrease depending on the perturbation scenario. Therefore, it is necessary to characterize the relationship between the perturbation and the change of the total length of the curve, and implement certain bounds on the latter.

When the spacing between adjacent characteristic nodes is small enough, the total length of the curve can be approximated by the total length of the line segments connecting each pair of the adjacent nodes. Let D_i denote the change of the length of the line segment between nodes $\mathbf{r}_i$ and $\mathbf{r}_{i+1}$ induced by the perturbation δ. The new positions of the nodes after the perturbation are given by $\tilde{\mathbf{r}}_i = \mathbf{r}_i + \delta_i \mathbf{n}_i + \lambda_i \mathbf{d}_i$ and $\tilde{\mathbf{r}}_{i+1} = \mathbf{r}_{i+1} + \delta_{i+1}\mathbf{n}_{i+1} + \lambda_{i+1}\mathbf{b}_{i+1}$. For notational convenience, let $\mathbf{p}_i \mu_i = \delta_i \mathbf{n}_i + \lambda_i \mathbf{d}_i$, and $\mathbf{p}_{i+1}\mu_{i+1} = \delta_{i+1}\mathbf{n}_{i+1} + \lambda_{i+1}\mathbf{d}_{i+1}$.

Then $\|\tilde{\mathbf{r}}_{i+1} - \tilde{\mathbf{r}}_i\|$ is the length of the corresponding line segment of the perturbed path. We assume that the variations δ_i, δ_{i+1}, λ_i, and λ_{i+1} are small enough such that μ_i, $\mu_{i+1} \ll \|\mathbf{r}_{i+1} - \mathbf{r}_i\|$. The length of the line segment of the perturbed path between nodes s_i and s_{i+1} is

$$\begin{aligned}\|\tilde{\mathbf{r}}_{i+1} - \tilde{\mathbf{r}}_i\| &= \|\mathbf{r}_{i+1} + \mu_{i+1}\mathbf{p}_{i+1} - \mathbf{r}_i - \mu_i \mathbf{p}_i\| \\ &= \sqrt{\|(\mathbf{r}_{i+1} - \mathbf{r}_i) + (\mu_{i+1}\mathbf{p}_{i+1} - \mu_i \mathbf{p}_i)\|^2}.\end{aligned}$$

By the polarization identity for the Euclidean inner product,

$$\begin{aligned}\|\tilde{\mathbf{r}}_{i+1} - \tilde{\mathbf{r}}_i\| =& \big(\|\mathbf{r}_{i+1} - \mathbf{r}_i\|^2 + \|\mu_{i+1}\mathbf{p}_{i+1} - \mu_i \mathbf{p}_i\|^2 \\ &+ 2\left\langle \mu_{i+1}\mathbf{p}_{i+1} - \mu_i \mathbf{p}_i, \mathbf{r}_{i+1} - \mathbf{r}_i \right\rangle \big)^{\frac{1}{2}}.\end{aligned}$$

Then the segment length D_i can be written as in (51).

$$\begin{aligned}D_i =& \|\tilde{\mathbf{r}}_{i+1} - \tilde{\mathbf{r}}_i\| - \|\mathbf{r}_{\mathbf{i+1}} - \mathbf{r}_{\mathbf{i}}\| \\ =& -\|\mathbf{r}_{i+1} - \mathbf{r}_i\| + \sqrt{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|^2 + \|\mu_{i+1}\mathbf{p}_{i+1} - \mu_i \mathbf{p}_i\|^2 + 2\left\langle \delta_{i+1}\mathbf{p}_{i+1} - \mu_i \mathbf{p}_i, \mathbf{r}_{i+1} - \mathbf{r}_i \right\rangle} \\ =& \frac{1}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|} \frac{\|\mu_{i+1}\mathbf{p}_{i+1} - \mu_i \mathbf{p}_i\|^2 + 2\left\langle \mu_{i+1}\mathbf{p}_{i+1} - \mu_i \mathbf{p}_i, \mathbf{r}_{i+1} - \mathbf{r}_i \right\rangle}{1 + \sqrt{1 + \frac{\|\mu_{i+1}\mathbf{p}_{i+1} - \mu_i \mathbf{p}_i\|^2}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|^2} + 2\left\langle \frac{\mu_{i+1}\mathbf{p}_{i+1} - \mu_i \mathbf{p}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|}, \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|} \right\rangle}}.\end{aligned} \tag{51}$$

By the small variation assumption, and dropping the square terms, expression (51) yields the following approximation for D_i

$$\begin{aligned}D_i \approx& \left\langle \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|}, \mu_{i+1}\mathbf{p}_{i+1} \right\rangle - \left\langle \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|}, \mu_i \mathbf{p}_i \right\rangle \\ =& \left\langle \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|}, \delta_{i+1}\mathbf{n}_{i+1} \right\rangle - \left\langle \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|}, \delta_i \mathbf{n}_i \right\rangle \\ &+ \left\langle \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|}, \lambda_{i+1}\mathbf{b}_{i+1} \right\rangle - \left\langle \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|}, \lambda_i \mathbf{b}_i \right\rangle\end{aligned} \tag{52}$$

$$
\mathbf{E}_\delta = \begin{bmatrix}
-\langle \mathbf{r}_2 - \mathbf{r}_1, \mathbf{n}_1 \rangle & \langle \mathbf{r}_2 - \mathbf{r}_1, \mathbf{n}_2 \rangle & & & 0 \\
 & -\langle \mathbf{r}_3 - \mathbf{r}_2, \mathbf{n}_2 \rangle & \langle \mathbf{r}_3 - \mathbf{r}_2, \mathbf{n}_3 \rangle & & \\
 & & \ddots & \ddots & \\
0 & & & -\langle \mathbf{r}_N - \mathbf{r}_{N-1}, \mathbf{n}_{N-1} \rangle & \langle \mathbf{r}_N - \mathbf{r}_{N-1}, \mathbf{n}_N \rangle
\end{bmatrix} \tag{53}
$$

$$
\mathbf{E}_\lambda = \begin{bmatrix}
-\langle \mathbf{r}_2 - \mathbf{r}_1, \mathbf{b}_1 \rangle & \langle \mathbf{r}_2 - \mathbf{r}_1, \mathbf{b}_2 \rangle & & & 0 \\
 & -\langle \mathbf{r}_3 - \mathbf{r}_2, \mathbf{b}_2 \rangle & \langle \mathbf{r}_3 - \mathbf{r}_2, \mathbf{b}_3 \rangle & & \\
 & & \ddots & \ddots & \\
0 & & & -\langle \mathbf{r}_N - \mathbf{r}_{N-1}, \mathbf{b}_{N-1} \rangle & \langle \mathbf{r}_N - \mathbf{r}_{N-1}, \mathbf{b}_N \rangle
\end{bmatrix} \tag{54}
$$

In order to write equation (52) in a more compact form, let $\mathbf{B} = \mathrm{diag}([1/\|\mathbf{r}_2 - \mathbf{r}_1\|, \ldots, 1/\|\mathbf{r}_N - \mathbf{r}_{N-1}\|])$, and define matrices $\mathbf{E}_\delta$ and $\mathbf{E}_\lambda$ as in (53) and (54). Also, let $\mathbf{1}_{N-1}$ denote the $N-1$ dimensional column vector with all elements equal to one. Let $\Delta_L(X)$ denote the change of the total length of the path induced by the variation X. Then Δ_L can be approximated by $\Delta_L(X) \approx \mathbf{1}_{N-1}^{\mathrm{T}} \mathbf{B} [\mathbf{E}_\delta, \mathbf{E}_\lambda] X$, which is a linear function of X. The constraint on the total length of the path is given by the following linear inequality constraint on X:

$$
L_{\min} - L \leq \Delta_L(X) \leq L_{\max} - L, \tag{55}
$$

where L is the length of the path before perturbation, and $L_{\max}$ and $L_{\min}$ are the upper and lower bounds of the path length, respectively. These inequalities are enforced elementwise. Alternatively, if the length of the path is fixed, then the linear equality constraint $\Delta_L(X) = 0$ is applied ($L_{\min} = L = L_{\max}$):

3.4.3 Curvature Constraints

Localized curvature constraints are important for practical path planning. For example, a ground vehicle requires a larger turning radius when moving on a slippery surface compared with the same operation on normal ground. Let $K_{\max,i}$ and $K_{\min,i}$ be the maximum and minimum curvature constraints allowed in a neighborhood of $\mathbf{r_i}$ $(i = 1, 2, \ldots, N)$ which are determined by the vehicle dynamics and the local environment.

According to (47), for $i = 2, \ldots, N-1$, the second derivative of the path at the i^{th} node is

$$\begin{aligned}
\mathbf{r}''(s_i) &= \frac{1}{9\Delta_s^2}\frac{\mathrm{d}^2\mathbf{r}(\frac{1}{3})}{\mathrm{d}\tau^2} \\
&= \frac{1}{9\Delta_s^2}\begin{bmatrix} 2 & 2 & 0 & 0 \end{bmatrix} \mathbf{G} \begin{bmatrix} \mathbf{r}_{i-1} \\ \mathbf{r}_i \\ \mathbf{r}_{i+1} \\ \mathbf{r}_{i+2} \end{bmatrix} = \frac{1}{\Delta_s^2}\begin{bmatrix} 1 & -2 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{r}_{i-1} \\ \mathbf{r}_i \\ \mathbf{r}_{i+1} \\ \mathbf{r}_{i+2} \end{bmatrix} \\
&= \frac{1}{\Delta_s^2}\left(\mathbf{r}_{i-1} - 2\mathbf{r}_i + \mathbf{r}_{i+1}\right).
\end{aligned}$$

Therefore the second derivative of the perturbed path $\tilde{\mathbf{r}}$ at s_i is given by

$$\tilde{\mathbf{r}}''(s_i) = \frac{1}{\Delta_s^2}\left(\tilde{\mathbf{r}}_{i-1} - 2\tilde{\mathbf{r}}_i + \tilde{\mathbf{r}}_{i+1}\right).$$

Neglecting the change of the normal direction caused by the variation and assume that $\tilde{\mathbf{n}}_i = \mathbf{n}_i$, $\tilde{\mathbf{b}}_i = \mathbf{b}_i$, $i = 1, \ldots, N$, we have

$$\begin{aligned}
\tilde{\kappa}_i &= \langle \tilde{\mathbf{n}}_i, \tilde{\mathbf{r}}_i'' \rangle \\
&\approx \langle \mathbf{n}_i, \tilde{\mathbf{r}}_i'' \rangle \\
&= \frac{1}{\Delta_s^2}\left(\langle \mathbf{n}_i, \tilde{\mathbf{r}}_{i-1} \rangle - 2\langle \mathbf{n}_i, \tilde{\mathbf{r}}_i \rangle + \langle \mathbf{n}_i, \tilde{\mathbf{r}}_{i+1} \rangle\right) \\
&= \frac{1}{\Delta_s^2}(\langle \mathbf{n}_i, \mathbf{r}_{i-1} \rangle + \langle \mathbf{n}_i, \mathbf{n}_{i-1} \rangle \delta_{i-1} + \langle \mathbf{n}_i, \mathbf{b}_{i-1} \rangle \lambda_{i-1} \\
&- 2\langle \mathbf{n}_i, \mathbf{r}_i \rangle - 2\langle \mathbf{n}_i, \mathbf{n}_i \rangle \delta_i - 2\langle \mathbf{n}_i, \mathbf{b}_i \rangle \lambda_i + \langle \mathbf{n}_i, \mathbf{r}_{i+1} \rangle + \langle \mathbf{n}_i, \mathbf{n}_{i+1} \rangle \delta_{i+1} + \langle \mathbf{n}_i, \mathbf{b}_{i+1} \rangle \lambda_{i+1}) \\
&= \kappa_i + \frac{1}{\Delta_s^2}\left(\langle \mathbf{n}_i, \mathbf{n}_{i-1} \rangle \delta_{i-1} - 2\delta_i + \langle \mathbf{n}_i, \mathbf{n}_{i+1} \rangle \delta_{i+1}\right) + \frac{1}{\Delta_s^2}\left(\langle \mathbf{n}_i, \mathbf{b}_{i-1} \rangle \lambda_{i-1} + \langle \mathbf{n}_i, \mathbf{b}_{i+1} \rangle \lambda_{i+1}\right).
\end{aligned}$$

Similarly, the curvature of the perturbed path at the first node can be estimated by

$$\begin{aligned}
\tilde{\kappa}_1 \approx \langle \mathbf{n}_1, \tilde{\mathbf{r}}_1'' \rangle = \kappa_1 &+ \frac{1}{\Delta_s^2}(-5\langle \mathbf{n}_1, \mathbf{n}_2 \rangle \delta_2 + 4\langle \mathbf{n}_1, \mathbf{n}_3 \rangle \delta_3 - \langle \mathbf{n}_1, \mathbf{n}_4 \rangle \delta_4) \\
&+ \frac{1}{\Delta_s^2}(-5\langle \mathbf{n}_1, \mathbf{b}_2 \rangle \lambda_2 + 4\langle \mathbf{n}_1, \mathbf{b}_3 \rangle \lambda_3 - \langle \mathbf{n}_1, \mathbf{b}_4 \rangle \lambda_4),
\end{aligned}$$

and

$$\begin{aligned}\tilde{\kappa}_N \approx \langle \mathbf{n}_N, \tilde{\mathbf{r}}''_N \rangle = \kappa_N &+ \frac{1}{\Delta_s^2} \left(-\langle \mathbf{n}_N, \mathbf{n}_{N-3} \rangle \delta_{N-3} + 4\langle \mathbf{n}_N, \mathbf{n}_{N-2} \rangle \delta_{N-2} - 5\langle \mathbf{n}_N, \mathbf{n}_{N-1} \rangle \delta_{N-1} \right) \\ &+ \frac{1}{\Delta_s^2} \left(-\langle \mathbf{n}_N, \mathbf{b}_{N-3} \rangle \lambda_{N-3} + 4\langle \mathbf{n}_N, \mathbf{b}_{N-2} \rangle \lambda_{N-2} - 5\langle \mathbf{n}_N, \mathbf{b}_{N-1} \rangle \lambda_{N-1} \right).\end{aligned}$$

Let $\tilde{K} = [\tilde{\kappa}_1, \ldots, \tilde{\kappa}_N]^{\mathrm{T}}$ be the curvature of the perturbed path $\tilde{\mathbf{r}}$ at the characteristic nodes. Then the three expressions above can be written as $\tilde{K} = K + \mathbf{C}X$, where $\mathbf{C} \in \mathbb{R}^{N \times N}$. The details for the computation of matrix $\mathbf{C}$ is given in Appendix A. Let $K_{\max} = [K_{\max,1}, K_{\max,2}, \ldots, K_{\max,N}]^{\mathrm{T}}$ and $K_{\min} = [K_{\min,1}, K_{\min,2}, \ldots, K_{\min,N}]^{\mathrm{T}}$. The curvature of the perturbed path then need to satisfy the linear inequality constraint

$$K_{\min} - K \leq \mathbf{C}X \leq K_{\max} - K. \tag{56}$$

3.4.4 Bounds on the Variation and Collision Avoidance

In the computation of the L^2 norm of the curvature, it is assumed that the path lengths between adjacent nodes is preserved by the perturbation. Such an assumption is valid only if the perturbation is small enough. Some constraints, such as (56), also require small variation along the path. Hence, it is necessary to impose limits on the allowable magnitude of variation. The small variation is also required by the approximation used in the path length constraint. On the other hand, the magnitude of the variation should also be limited for collision-avoidance, since a large variation of the path in a neighborhood of an obstacle may lead to a collision.

By carefully choosing the bounds of variation, we can also ensure, at least approximately, the collision-avoidance of the perturbed path. The whole path contains $N-1$ segments. For the path segment between the $i-1^{\text{th}}$ and i^{th} nodes, consider the variations along the normal and binormal directions separately. Specifically, for $i = 2 : N-2$, Let $\bar{\delta}_{i,u} = \delta_{\max}$, where $\delta_{\max}$ is a predetermined small positive number, and keep other characteristic nodes unperturbed. Consider a variation given by $\delta \mathbf{r}_i = \delta \mathbf{r}_{i+1} = \mathbf{n}_i \bar{\delta}_{i,u}$, and $\delta_j = 0$ for $j = 2, \ldots, N-1$, $j \neq i$, $j \neq i+1$. If this segment is still collision-free after the variation, then $\bar{\delta}_{i,u} = \delta_{\max}$, otherwise decrease $\bar{\delta}_{i,u}$ while keeping $\delta_i = \delta_{i-1} = \bar{\delta}_{i,u}$ until the perturbed segment is collision-free. Collision is checked along the perturbed path at the $i-1^{\text{th}}$ and i^{th} nodes, as well as at a certain number of interpolating points between these two nodes. The locations of the these points after perturbation can be computed using equation (45).

Similarly, the variation lower bound $\bar{\delta}_{i,l}$ of the same segment is determined by initially choosing $\bar{\delta}_{i,l} = -\delta_{\max}$ and $\delta_{i-1} = \delta_i = \bar{\delta}_{i,l}$. If collision occurs, gradually increase $\bar{\delta}_{i,l}$ while keeping $\delta_{i-1} = \delta_i = \bar{\delta}_{i,l}$ until the perturbed path is collision-free, and let $l_{\delta_i} = \delta_i$. In the same way, the bounds $\bar{\lambda}_{il}$ and $\bar{\lambda}_{iu}$ on the variation λ_i, $i = 2, \ldots, N-1$ can also be determined.

Let $l_{\delta_1}, \ldots, l_{\delta_N}$ and $u_{\delta_1}, \ldots, u_{\delta_N}$ be the lower and upper bounds of the variations δ_i, and let $l_{\lambda_1}, \ldots, l_{\lambda_N}$ and $u_{\lambda_1}, \ldots, u_{\lambda_N}$ be the lower and upper bounds of the variations λ_i, respectively. Because the path is required to pass through the start and target positions, the variation must be zero at these two points, which can be achieved by setting the bounds as $l_{\delta_1} = u_{\delta,1} = 0$, $l_{\delta_N} = u_{\delta_N} = 0$, $l_{\lambda_1} = u_{\lambda,1} = 0$, $l_{\lambda_N} = u_{\lambda_N} = 0$. The bounds of the variations of the second and the $N-1^{\text{th}}$ nodes are given by $l_{\delta_2} = \bar{\delta}_{2,l}$, $u_{\delta_2} = \bar{\delta}_{2,u}$, $l_{\lambda_2} = \bar{\lambda}_{2,l}$, $u_{\lambda_2} = \bar{\lambda}_{2,u}$. For $i = 3, \ldots, N-2$, the bounds on the variation are given by $l_{\delta_i} = \max\{\bar{\delta}_{i-1,l}, \bar{\delta}_{i,l}\}$, $l_{\lambda_i} = \max\{\bar{\lambda}_{i-1,l}, \bar{\lambda}_{i,l}\}$, $u_{\delta_i} = \min\{\bar{\delta}_{i-1,u}, \bar{\delta}_{i,u}\}$ $u_{\lambda_i} = \min\{\bar{\lambda}_{i-1,u}, \bar{\lambda}_{i,u}\}$.

Let $X_{\min} = \{l_{\delta_1}, \ldots, l_{\delta_N}, l_{\lambda_1}, \ldots, l_{\lambda_N}\}$ and $X_{\max} = \{u_{\delta_1}, \ldots, u_{\delta_N}, u_{\lambda_1}, \ldots, u_{\lambda_N}\}$. With $X_{\min} \leq X \leq X_{\max}$ as a constraint in the optimization, the collision avoidance requirement is approximately satisfied. Simulation results show that the performance of such a simple treatment is acceptable. More rigorous treatment for collision avoidance is also possible. As an example, for the i^{th} segment of the path $\mathbf{r}$ between the i^{th} and the $i+1^{\text{th}}$ nodes, an obstacle-free convex polygon containing this segment can be find. By requiring that a certain number of interpolating points along this segment of the perturbed path stay within the polygon, collision avoidance is enforced along this segment. The collision avoidance of the whole path is then guaranteed by finding $N-1$ convex polygons for each segment and requiring that each segment stays within the corresponding polygon. Such a laborious treatment leads to a large number of linear inequality constraints on X.

3.4.5 Initial and Final Condition

Suppose that the perturbed path is required to satisfy tangent constraints at the first and last nodes with

$$\mathbf{r}'_1 = \left[t_{1_x}, t_{1_y}, t_{1_z}\right], \qquad \mathbf{r}'_N = \left[t_{N_x}, t_{N_y}, t_{N_z}\right].$$

Let $\mathbf{b}_1$ and $\mathbf{b}_N$ denote the binormal direction at the first and last nodes, which are given by $\mathbf{b}_1 = \mathbf{r}'_1 \times \mathbf{n}_1$, and $\mathbf{b}_N = \mathbf{r}'_N \times \mathbf{n}_N$.

The tangent, or the path derivative of of $\tilde{\mathbf{r}}$ at the first node is

$$\begin{aligned}
\tilde{\mathbf{r}}'(0) &= \frac{1}{3\Delta s}\frac{\mathrm{d}\mathbf{r}(0;2)}{\mathrm{d}\tau} \\
&= \frac{1}{3\Delta s}\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}\mathbf{G}\begin{bmatrix} \mathbf{r}_1 + \mathbf{n}_1\delta_1 + \mathbf{b}_1\lambda_1 \\ \mathbf{r}_2 + \mathbf{n}_2\delta_2 + \mathbf{b}_2\lambda_2 \\ \mathbf{r}_3 + \mathbf{n}_3\delta_3 + \mathbf{b}_3\lambda_3 \\ \mathbf{r}_4 + \mathbf{n}_4\delta_4 + \mathbf{b}_4\lambda_4 \end{bmatrix} \\
&= \frac{1}{3\Delta s}\begin{bmatrix} -5.5 & 9 & -4.5 & 1 \end{bmatrix}\begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 + \mathbf{n}_2\delta_2 + \mathbf{b}_2\lambda_2 \\ \mathbf{r}_3 + \mathbf{n}_3\delta_3 + \mathbf{b}_3\lambda_3 \\ \mathbf{r}_4 + \mathbf{n}_4\delta_4 + \mathbf{b}_4\lambda_4 \end{bmatrix}.
\end{aligned}$$

The constraint that $\tilde{\mathbf{r}}'(0)$ is parallel to $\mathbf{r}'_1$ is equivalent to the requirement that $\tilde{\mathbf{r}}'(0)$ is perpendicular to both $\mathbf{b}_1$ and $\mathbf{n}_1$, i.e.,

$$\begin{aligned}
&-5.5\langle \mathbf{r}_1, \mathbf{n}_1\rangle + 9\langle \mathbf{r}_2 + \mathbf{n}_2\delta_2, \mathbf{n}_1\rangle - 4.5\langle \mathbf{r}_3 + \mathbf{n}_3\delta_3, \mathbf{n}_1\rangle + \langle \mathbf{r}_4 + \mathbf{n}_4\delta_4, \mathbf{n}_1\rangle \\
&\qquad + 9\langle \mathbf{r}_2 + \mathbf{b}_2\lambda_2, \mathbf{n}_1\rangle - 4.5\langle \mathbf{r}_3 + \mathbf{b}_3\lambda_3, \mathbf{n}_1\rangle + \langle \mathbf{r}_4 + \mathbf{b}_4\lambda_4, \mathbf{n}_1\rangle = 0, \qquad (57) \\
&-5.5\langle \mathbf{r}_1, \mathbf{b}_1\rangle + 9\langle \mathbf{r}_2 + \mathbf{n}_2\delta_2, \mathbf{b}_1\rangle - 4.5\langle \mathbf{r}_3 + \mathbf{n}_3\delta_3, \mathbf{b}_1\rangle + \langle \mathbf{r}_4 + \mathbf{n}_4\delta_4, \mathbf{b}_1\rangle \\
&\qquad + 9\langle \mathbf{r}_2 + \mathbf{b}_2\lambda_2, \mathbf{b}_1\rangle - 4.5\langle \mathbf{r}_3 + \mathbf{b}_3\lambda_3, \mathbf{b}_1\rangle + \langle \mathbf{r}_4 + \mathbf{b}_4\lambda_4, \mathbf{b}_1\rangle = 0, \qquad (58)
\end{aligned}$$

which are linear constraints on δ_2, δ_3, and δ_4. Similarly, the tangent constraint at the last node can be enforced by

$$\begin{aligned}5.5\langle\mathbf{r}_N,\mathbf{n}_N\rangle &- 9\langle\mathbf{r}_{N-1}+\mathbf{n}_{N-1}\delta_{N-1},\mathbf{n}_N\rangle + 4.5\langle\mathbf{r}_{N-2}+\mathbf{n}_{N-2}\delta_{N-2},\mathbf{n}_N\rangle\\ &- \langle\mathbf{r}_{N-3}+\mathbf{n}_{N-3}\delta_{N-3},\mathbf{n}_N\rangle - 9\langle\mathbf{r}_{N-1}+\mathbf{b}_{N-1}\lambda_{N-1},\mathbf{n}_N\rangle\\ &+ 4.5\langle\mathbf{r}_{N-2}+\mathbf{b}_{N-2}\lambda_{N-2},\mathbf{n}_N\rangle - \langle\mathbf{r}_{N-3}+\mathbf{b}_{N-3}\lambda_{N-3},\mathbf{n}_N\rangle = 0,\end{aligned} \tag{59}$$

$$\begin{aligned}5.5\langle\mathbf{r}_N,\mathbf{b}_N\rangle &- 9\langle\mathbf{r}_{N-1}+\mathbf{n}_{N-1}\delta_{N-1},\mathbf{b}_N\rangle + 4.5\langle\mathbf{r}_{N-2}+\mathbf{n}_{N-2}\delta_{N-2},\mathbf{b}_N\rangle\\ &- \langle\mathbf{r}_{N-3}+\mathbf{n}_{N-3}\delta_{N-3},\mathbf{b}_N\rangle - 9\langle\mathbf{r}_{N-1}+\mathbf{b}_{N-1}\lambda_{N-1},\mathbf{b}_N\rangle\\ &+ 4.5\langle\mathbf{r}_{N-2}+\mathbf{b}_{N-2}\lambda_{N-2},\mathbf{b}_N\rangle - \langle\mathbf{r}_{N-3}+\mathbf{b}_{N-3}\lambda_{N-3},\mathbf{b}_N\rangle = 0,\end{aligned} \tag{60}$$

3.4.6 Connection to Beam Theory

Consider a classical beam subject to pure bending. The bending moment and the local curvature satisfy

$$\kappa\left(s\right)=\frac{M\left(s\right)}{EI(s)},$$

where $\kappa(s)$ is the local curvature of the neutral surface of the beam, $M(s)$ is the bending moment at the cross section at s, and $I(s)$ is the second moment of area of the cross section about its neutral surface, and E is the Young's modulus of the beam material. The product $EI(s)$ is often referred to as the *flexural rigidity* or the *bending stiffness* of the beam.

The total strain energy U of the bending beam can be written as:

$$U=\int_0^{s_f}\frac{M^2\left(s\right)}{2EI\left(s\right)}\mathrm{d}s=\frac{1}{2}\int_0^{s_f}EI(s)\kappa^2(s)\mathrm{d}s,$$

which is exactly the square of the weighted L_2 norm of the curvature function. Hence, the result of the quadratic program essentially corresponds to a minimum bending energy configuration in a neighborhood of the original path. It is also observed that the weight function $w(s)$ in (50) corresponds to the flexural rigidity $EI(s)$.

3.5 Path Smoothing Algorithm

3.5.1 Discrete Evolution and the Path Smoothing Algorithm

Consider a family of smooth paths $\mathcal{P}(s,j)$, where s is the path coordinate parameterizing the path and j is the index parameterizing the family. The path evolves among the family

$\mathcal{P}(s,j)$ at the representative nodes according to the evolution equation

$$\begin{aligned} \mathcal{P}\left(s_i, j+1\right) &= \mathcal{P}\left(s_i, j\right) + X^*_{n_i} \mathbf{n}\left(\mathbf{s}_i, j\right) + X^*_{b_i} \mathbf{b}\left(\mathbf{s}_i, j\right), \\ \mathcal{P}\left(s, 0\right) &= \mathcal{P}^{(0)}\left(s\right), \end{aligned} \tag{61}$$

where $X^*_{n_i}$ and $X^*_{b_i}$ are the i^{th} component of X^*_n and X^*_b, which compose the optimal solution $X^* = [X^{*\text{T}}_n, X^{*\text{T}}_b]$ to the quadratic program with initial path $\mathcal{P}(s,j)$.

The proposed path smoothing algorithm is designed based on the evolution equation (61), and involves solving iteratively a series of Quadratic Programming problems:

1. Let j be the count of iterations, starting from $j = 1$,

2. Discretize the path with N nodes, say, $s_1 = 0, s_2, s_3, \ldots, s_N = s_f$.

3. Determine the bounds of variation, and solve the quadratic programming problem. Interpolate the result with a cubic spline curve to generate the new path,

4. Compute the difference between the new and the old path by

$$\xi_j = \int_0^{s_f} \left\| \mathcal{P}(s,j) - \mathcal{P}(s,j-1) \right\|^2 \mathrm{d}s.$$

   Stop the iteration if ξ_j is smaller than some predetermined threshold, or if j reaches the maximum number of iterations. Otherwise increase j by one and go to Step 2).

The main difference between the above iterative Quadratic Programming method and the standard Sequential Quadratic Programming is that for the later, the cost function and constraints are pre-determined functions of the decision variables, and these functions do not change in the optimization. In the iterative Quadratic Programming approach described above, these functions are updated after each iteration, which means that a new problem is formulated at the beginning of each iteration based on the solution of the previous iteration.

In order to test the smoothing efficiency of the proposed algorithm, we consider a planar path example, and compared the proposed algorithm with the curvature evolution path-smoothing method with zero constant speed, which has the following evolution equation [111]:

$$\begin{aligned}\frac{\partial \mathcal{P}(s,t)}{\partial t} &= \kappa(s,t)\mathbf{n}(s,t), \\ \mathcal{P}\left(s,0\right) &= \mathcal{P}^{(0)}\left(s\right). \end{aligned} \tag{62}$$

Equation (62) can be solved numerically using forward difference

$$\mathcal{P}(s,t+\varepsilon) = \mathcal{P}(s,t) + \varepsilon\kappa(s)\mathbf{n}(s,t), \tag{63}$$

where ε is a small number.

Both methods were started from the same initial path $\mathcal{P}(s,0)$ which in this case was the graph of a sine function, and converge to a straight line at the end. The same stopping rule was used for both methods. The fixed length requirement in the quadratic programming method is relaxed to be comparable to the curvature method. For fast convergence speed while ensuring numerical stability, we chose $\varepsilon = 0.5$ for the curvature evolution method in (63). The quadratic programming method finishes the smoothing in 0.3 sec after 21 iterations, while the curvature method finishes the smoothing in 5.6 sec after 3528 iterations. The results of the two methods are shown in Fig. 15 and Fig. 16.

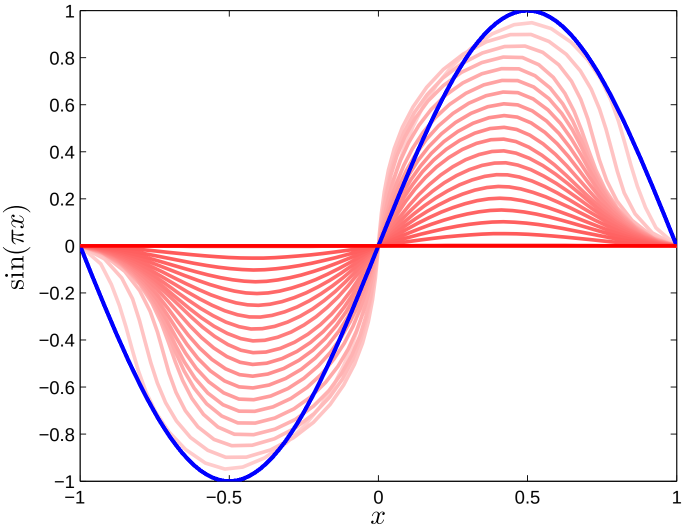

Figure 15: Quadratic programming path smoothing.

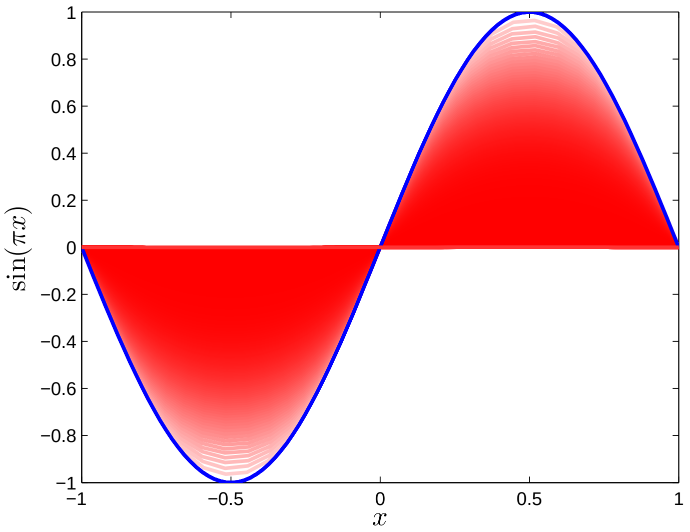

Figure 16: Curvature evolution.

3.5.2 Reconciling Conflicts Between Variation Bounds and Constraints

Due to the bounds on the allowed variation, the domain of optimization in each step of the proposed algorithm is relatively small, and sometimes the variation bounds are in conflict with the boundary conditions and curvature constraints, in the sense that the prescribed boundary conditions and curvature constraints cannot be satisfied by any variation within the bounds during a single iteration.

To resolve such conflicts, the curvature constraints and boundary conditions are enforced progressively during the iterations when necessary, rather than being enforced explicitly in each iteration. For example, suppose the path needs to satisfy the curvature constraints $K_{\min} \leq K \leq K_{\max}$. Then for each iteration j, the following relaxed curvature bounds are used

$$K_{\min} - c_1 e^{-\beta_1 j} \leq K_j \leq K_{\max} + c_2 e^{-\beta_2 j},$$

where $c_1, c_2, \beta_1, \beta_2 > 0$. It is seen that the left and right hand sides in the above inequalities initially provide relaxed curvature bounds when $j = 0$, yet approach the prescribed bounds $K_{\min}$ and $K_{\max}$ asymptotically as j increases. A similar technique is applied for the enforcement of the tangent directional constraints at the start and end points.

3.6 Numerical Examples

3.6.1 Fixed Length Path Smoothing with Collision Avoidance

In this example, a UAV flies from point A to point B. The obstacles are represented by the polytopes in Fig. 17. The original three-dimensional landing path is shown as the red curve in the Figure. This initial path is processed using the path smoothing algorithm introduced in this chapter, and the smoothed path result is shown as the blue curve in Fig. 17. The initial and final tangents of the path are fixed during the optimization, and the path length is also fixed. The path smoothing algorithm finishes in 3.5 sec after 15 iterations. The curvature profiles for the original and smoothed paths are compared in Fig. 18. The L_2 norm of the curvature function with respect to the path coordinate decreased by 67% after smoothing, while the L_∞ norm was reduced by 61%. In Fig. 19, the optimal speed profiles of the original and smoothed paths are compared. It is clear that the smoothed path provides a shorter travel time. The optimal speed profiles are computed using the time-optimal parameterization method introduced in [137] with free final speed at point B.

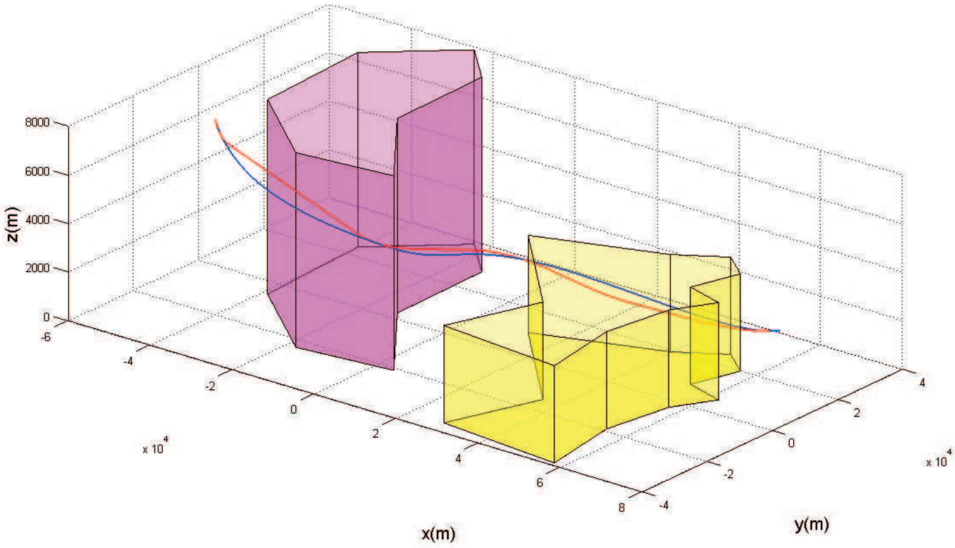

Figure 17: Path smoothing in the presence of obstacles.

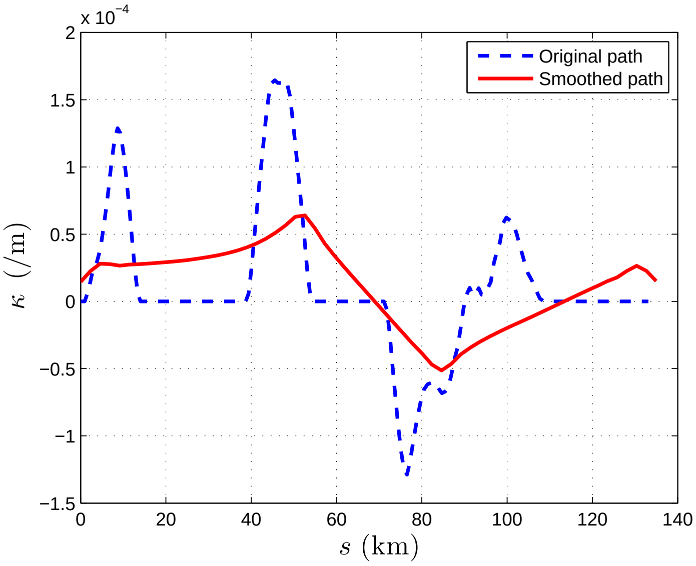

Figure 18: Curvature profile comparison.

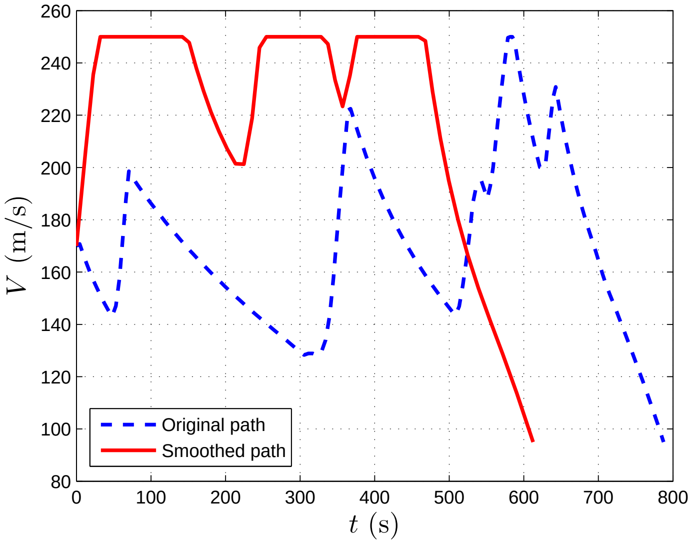

Figure 19: Optimal speed profile.

3.6.2 Path Smoothing with Localized Curvature Bounds

In this example, a ground vehicle starts from point A at one side of a frozen river, avoids the obstacle, crosses the river while passing through point B, and finally reaches the target at point C at the other side of the river. Due to the small coefficient of friction of the icy river surface, it is required that the segment of the path on the ice surface must have zero curvature (no turning allowed). The initial path consists of three line segments. During the smoothing process, the constraint on the total length of the path is relaxed. Furthermore, there exists no directional constraint at the start and the end of the path. In order to ensure that the path passes through point B, a node is added to the path at point B, and the variation at this node is set to be zero during the smoothing process. The result from smoothing is shown in Fig. 20. It is clear that the ground vehicle does not need to perform any turning maneuver on the ice surface.

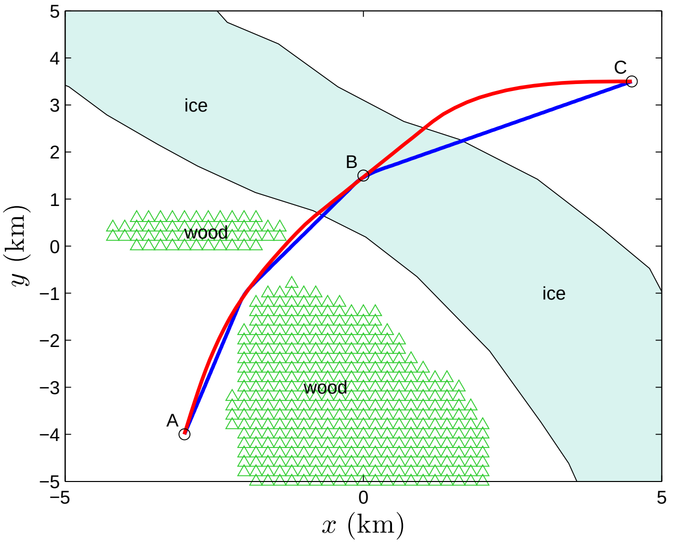

Figure 20: Smoothed path with local curvature constraint.

3.7 Conclusions

In this chapter, we considered the problem of three-dimensional path smoothing with obstacles and local curvature constraints. The problem is formulated as a quadratic program, which minimizes the weighted L_2 norm of the curvature along the path. By incorporating additional linear constraints into the quadratic programming problem, extra constraints on the tangent of the path, path length, and local curvature can also be accommodated. The proposed path smoothing algorithm has been applied to several examples, and its efficiency and effectiveness have been validated.

CHAPTER IV

TIME-OPTIMAL PATH TRACKING OPERATION FOR FIXED-WING AIRCRAFT

In this chapter, we propose a method for the minimum-time travel of a fixed-wing aircraft along a prescribed geometric path. The method checks the feasibility of the path, namely, whether it is possible for the aircraft to travel along the path without violating the state or control constraints. If the path is feasible, the method subsequently finds a semi-analytical solution of the speed profile that minimizes the travel time along the path. The optimal speed profile is used to time-parameterize the path, and generate the state trajectory and control histories via inverse dynamics. Two algorithms for the time-optimal parameterization are proposed. Numerical examples are presented to demonstrate the validity, numerical accuracy and optimality of the proposed method.

4.1 Introduction

Numerous methods including concatenations of Dubins' path primitives, potential field methods, optimal control, etc, have all received considerable attention in the literature for the solution of path-planning problems[93, 123, 44, 39]. Besides trajectory optimization-based methods, the previous approaches typically do not provide the control histories required for maneuvering the vehicle to follow the optimal path. Instead, they generate obstacle-free, geometric paths in the environment. The actual implementation (i.e., path-following) is left to a trajectory tracking controller (or human pilot) which generates the required control commands to follow the path after a suitable time-parameterization along the optimal path is imposed. However, because most of these path-planning methods are at the kinematic level, and do not account for the dynamics of the aircraft, the feasibility of the resulting trajectory is not guaranteed a priori, i.e., it is possible that no control exists that allows the aircraft to follow the proposed path without violating the control or state constraints.

An alternative approach for path-planning of aircraft, which considers more realistic dynamics and incorporates the state and control constraints, is to formulate the flight path-planning problem as an optimal control problem[90]. For the numerical solution of this optimal control problem, the convergence of the solution depends heavily on the quality of the initial guess of the time histories of both the state and control variables. A good initial guess can help the solution converge much faster. A bad initial guess will hinder convergence or lead to divergence of the overall numerical scheme. However, it is not easy to obtain a set of state and control histories that are consistent with the aircraft dynamics and satisfy the given constraints and boundary conditions.

In this chapter we propose a new method to generate time-optimal paramerizations along a given path, which bypasses the solution of the complete time-optimal control problem. The time parameterization is constructed by solving for a time-optimal speed profile such that the state and control constraints are satisfied. The problem is similar to the time-optimal control of robotic manipulators[28, 119, 96, 118, 116]. These references take advantage of the Lagrangian form of the dynamics of a fully-actuated robotic manipulator, to compute the required speed profile for the manipulator to move along a specified path in minimum time. In this work we take a similar approach, and time-parameterize a given geometric path for a fixed-wing aircraft in a way that results in minimum-time optimality, while satisfying the dynamic and control constraints along the given path. The proposed method works equally well as a post-processing tool for pure geometric/kinematic planners for checking the feasibility of the generated path. This method can also be used to construct good initial guesses for a complete trajectory optimization solver. Specifically, the approach may be used as a bridge between geometric path-planning methods and numerical optimal control methods to improve convergence of a Nonlinear Programming (NLP) solver. The geometric path given by the geometric planner can be optimally time-parameterized to obtain the corresponding state and control histories, which can then be passed to the NLP solver as an initial guess.

In the rest of this chapter, we first show that the problem of optimal time-parameterization of a geometric path for a fixed-wing aircraft can be converted to a constrained *scalar* functional optimization problem by decoupling the controls. The analytical solution to this problem is derived using Pontryagin's Maximum Principle. We study the switching structure of the optimal control profile, and propose two algorithms that can be used to generate the optimal speed profile and hence also the profile of the optimal thrust. Numerical examples are included to demonstrate the developed theory.

4.2 Mathematical Preliminaries

In this section, we review a few mathematical definitions that will be needed in the ensuing analysis of the minimum-time travel problem. Since we will be dealing with finite-time problems, we only need to consider functions defined over compact intervals of the real line.

Definition 4.2.1. Let $f : [t_0, t_f] \mapsto \mathbb{R}$. The *left limit* of f at $t \in (t_0, t_f]$ is defined by

$$f(t^-) \overset{\triangle}{=} \lim_{h \uparrow 0} f(t+h)$$

if the limit exists. Similarly, the *right limit* of f at $t \in [t_0, t_f)$ is defined by

$$f(t^+) \overset{\triangle}{=} \lim_{h \downarrow 0} f(t+h)$$

if the limit exists.

Note that, $f(t_0^-)$ and $f(t_f^+)$ do not exist.

Definition 4.2.2. Let $f : [t_0, t_f] \mapsto \mathbb{R}$. Then f is *left continuous at* $t \in (t_0, t_f]$ if $f(t^-) = f(t)$. The function f is *right continuous at* $t \in [t_0, t_f)$ if $f(t^+) = f(t)$. The function f is *left continuous* if f is left continuous at every point in $(t_0, t_f]$. The function f is *right continuous* if f is right continuous at every point in $[t_0, t_f)$.

We will need to consider functions that are continuous or differentiable everywhere on an interval except possibly at a finite number of points.

Definition 4.2.3. Let $f : [t_0, t_f] \mapsto \mathbb{R}$. Then f is *piecewise continuous* if the following conditions hold:

i) Both the limits $f(t_0^+)$ and $f(t_f^-)$ exist.

ii) There exists a finite set of points $\mathcal{S} = \{t_1, t_2, \ldots, t_N\} \subset (t_0, t_f)$ such that f is continuous on $(t_0, t_f)\backslash\mathcal{S}$, and, for every $t \in \mathcal{S}$, both $f(t^-)$ and $f(t^+)$ exist.

The function f is *left piecewise continuous* if f is piecewise continuous and left continuous. It is *right piecewise continuous* if it is piecewise continuous and right continuous.

Note that, by definition, a piecewise continuous function is bounded. Note also that, given a piecewise continuous functions, we can always redefine the value of the functions at the (necessarily finite) points of discontinuities to make it left (or right) continuous.

Definition 4.2.4. Let $f : [t_0, t_f] \mapsto \mathbb{R}$. Then f is *upper semi-continuous* (respectively, *lower semi-continuous*) at $t \in [t_0, t_f]$ if

$$\limsup_{\tau\to t} f(\tau) \leq f(t) \quad \text{(resp. } \liminf_{\tau\to t} f(\tau) \geq f(t)),$$

and, in addition, $\limsup_{\tau\downarrow t_0} f(\tau) \leq f(t_0)$ (resp. $\liminf_{\tau\downarrow t_0} f(\tau) \geq f(t_0)$) and $\limsup_{\tau\uparrow t_f} f(\tau) \leq f(t_f)$ (resp. $\liminf_{\tau\uparrow t_f} f(\tau) \geq f(t_f)$).

Note that if a function is right/left continuous at each point $t \in [t_0, t_f]$, then upper semi-continuity (respectively, *lower semi-continuity*) of f reduces to the conditions $f(t) = \max\{f(t^-), f(t^+)\}$ (resp. $f(t) = \min\{f(t^-), f(t^+)\}$) for all $t \in (t_0, t_f)$ and $f(t_0) = f(t_0^+)$ and $f(t_f) = f(t_f^-)$.

Definition 4.2.5. Let $f : [t_0, t_f] \mapsto \mathbb{R}$. Then f is *piecewise continuously differentiable*[1] or *piecewise* C^1 if the following conditions hold:

i) f is piecewise continuous.

ii) Define $\mathcal{D} \triangleq \{t \in [t_0, t_f] : f'(t)$ exists and f' is continuous at $t\}$. Then $\mathcal{S} \triangleq [t_0, t_f]\backslash\mathcal{D}$ contains a finite number of points.

iii) For every $t \in \mathcal{S} \cap (t_0, t_f)$, both $f'(t^-)$ and $f'(t^+)$ exist.

iv) The limits $f'(t_0^+)$ and $f'(t_f^-)$ exist.

[1] Note that our definition is somewhat different than the most common definition of the derivative of the function being piecewise continuous (often called piecewise smoothness), which assumes that the function is continuous. In particular, our definition of piecewise continuously differentiability allows for the function to be only piecewise continuous.

Note that, by definition, a piecewise continuously differentiable function is bounded.

We can extend Definition 4.2.5 to functions that have derivatives of order k, except possibly at a finite number of points, but at those points the directional left/right derivatives of order k exist. We call these functions *piecewise continuously differentiable of order k* or *piecewise* C^k. In this work we will work with piecewise C^k functions, where $k = 1, 2, 3, \ldots$ can be arbitrarily large and, which, in addition, have the property that at those points where the derivatives[2] of any order exist, they have convergence power (i.e., Taylor) series expansions. We will call such functions *piecewise analytic* (better, *piecewise-defined analytic*). Recall that a function $f : [t_0, t_f] \mapsto \mathbb{R}$ is (real) *analytic* at $t \in [t_0, t_f]$ if its Taylor series converges to $f(t)$ in a neighborhood of t. These are, essentially, functions whose restriction on certain open intervals are analytic. Specifically, we have the following definition.

Definition 4.2.6. Let $f : [t_0, t_f] \mapsto \mathbb{R}$. Then f is *piecewise analytic* if the following conditions hold:

i) f is *piecewise* C^k, for all $k \geq 1$.

ii) There exist $t_0 < t_1 < \cdots < t_N < t_f$ such that f is analytic on each of the open intervals (t_0, t_1), (t_1, t_2), $\ldots$, (t_N, t_f).

It is a straightforward exercise to show that piecewise analytic functions, according to our definition, inherit most of the nice properties of real-analytic functions. In particular, the summation, the subtraction, the multiplication, and the composition of piecewise analytic functions is piecewise analytic[76]. Note, however, that the quotient of two piecewise analytic functions may not be piecewise analytic.

One of the key properties of real-analytic functions is that they cannot become zero at infinitely many points in a compact set, unless they are identically zero[76]. As a result, two distinct real-analytic functions f and g defined on a compact interval can intersect only a finite number of times. It follows that $\max\{f, g\}$ and $\min\{f, g\}$ are real-analytic functions. By restricting the analysis on the (necessarily finite) intervals where the functions

[2] At the boundary points of the interval $[t_0, t_f]$ the higher order derivatives are to be interpreted in the directional sense, from the right or the left, respectively.

are analytic it is not difficult to show that similar properties hold for piecewise analytic functions.

Proposition 4.2.1. Let $f : [t_0, t_f] \mapsto \mathbb{R}$ and $g : [t_0, t_f] \mapsto \mathbb{R}$ be piecewise analytic functions, such that $f \not\equiv g$. Then the following hold:

i) The functions f and g are either identical over a union of compact intervals, or intersect at a finite number of points, or both.

ii) The functions $\max\{f, g\}$ and $\min\{f, g\}$ are piecewise analytic.

Item *i*) of the previous proposition states, essentially, that two distinct piecewise analytic functions cannot intersect at a countably infinite number of points.

Let a path in the three-dimensional space, parameterized by the path coordinate s, be given as follows: $x = x(s)$, $y = y(s)$, $z = z(s)$, where $s \in [s_0, s_f]$. The main objective of this chapter is to find a time-parameterization along the path, i.e., a function $s(t)$, where $t \in [0, t_f]$ such that the corresponding time-parameterized trajectory $(x(s(t)), y(s(t)), z(s(t)))$ minimizes the flight time t_f. It is assumed that $x(s)$, $y(s)$ and $z(s)$ are continuously differentiable and piecewise analytic[3].

Consider the following equations of motion for a point-mass model of a fixed-wing aircraft [89]:

$$\begin{aligned}
\dot{x} &= v \cos\gamma \cos\psi, && (64)\\
\dot{y} &= v \cos\gamma \sin\psi, && (65)\\
\dot{z} &= v \sin\gamma, && (66)\\
\dot{v} &= \frac{1}{m}\left[T - F_D(C_L, v, \rho) - mg\sin\gamma\right], && (67)\\
\dot{\gamma} &= \frac{1}{mv}\left[F_L(C_L, v, \rho)\cos\phi - mg\cos\gamma\right], && (68)\\
\dot{\psi} &= -\frac{F_L(C_L, v, \rho)\sin\phi}{mv\cos\gamma}, && (69)
\end{aligned}$$

where x, y, z are the coordinates defining the position of the aircraft, v is the speed, ρ is the air density (varying with altitude), γ is the flight path angle, ψ is the heading angle, and ϕ

[3] This is a weak assumption. Piecewise polynomial functions or spline functions, for example, satisfy these conditions.

is the bank angle. The aerodynamic lift force $F_L(C_L, v, \rho)$ and drag force $F_D(C_L, v, \rho)$ are given by:

$$\begin{aligned} F_L\left(C_L, v, \rho\right) &= \frac{1}{2}\rho v^2 S C_L, \\ F_D\left(C_L, v, \rho\right) &= \frac{1}{2}\rho v^2 S C_D = \frac{1}{2}\rho v^2 S (C_{D_0} + K C_L^2), \end{aligned}$$

where C_{D_0} and K are constants determined by the aerodynamic properties of the aircraft, and S is the main wing surface area. The effect of wind is not considered. In this model, the lift coefficient C_L, the bank angle ϕ, and the thrust T are the control inputs.

Because the given path is naturally parameterized using the path coordinate s instead of time, the equations of motion can be rewritten with respect to s as follows (where prime denotes differentiation with respect to s:

$$\begin{aligned} x' &= \cos\gamma\cos\psi, && (70) \\ y' &= \cos\gamma\sin\psi, && (71) \\ z' &= \sin\gamma, && (72) \\ v' &= \frac{1}{mv}\left[T - F_D\left(C_L, v, \rho\right) - mg\sin\gamma\right], && (73) \\ \gamma' &= \frac{1}{mv^2}\left[F_L\left(C_L, v, \rho\right)\cos\phi - mg\cos\gamma\right], && (74) \\ \psi' &= -\frac{F_L\left(C_L, v, \rho\right)\sin\phi}{mv^2\cos\gamma}, && (75) \end{aligned}$$

where the following relations have been used for deriving (70)-(75):

$$\begin{aligned} \mathrm{d}t &= \frac{\mathrm{d}s}{v}, && (76) \\ \mathrm{d}s &= \sqrt{\mathrm{d}^2x + \mathrm{d}^2y + \mathrm{d}^2z}, && (77) \\ \psi &= \arctan\frac{\mathrm{d}y}{\mathrm{d}x} = \arctan\frac{y'}{x'}, && (78) \\ \gamma &= \arctan\frac{\mathrm{d}z}{\sqrt{\mathrm{d}x^2 + \mathrm{d}y^2}} = \arctan\frac{z'}{\sqrt{x'^2 + y'^2}}, && (79) \\ \psi' &= \frac{1}{1 + (y'/x')^2}\frac{y''x' - y'x''}{x'^2} = \frac{x'^2}{x'^2 + y'^2}\frac{y''x' - y'x''}{x'^2} = \frac{y''x' - y'x''}{x'^2 + y'^2}, && (80) \\ \gamma' &= \frac{z''x'^2 + z''y'^2 - z'x''x' - z'y''y'}{\sqrt{x'^2 + y'^2}}. && (81) \end{aligned}$$

Note that the flight path angle γ and the heading angle ψ are purely geometric variables, therefore, once a three-dimensional path $(x(s), y(s), z(s))$ is given, these variables and their derivatives with respect to the path coordinate can be computed from (80) and (81). It is clear from the previous expressions that the continuous differentiability of x, y, z implies the continuity of x', y', and z'. We also assume that the fixed-wing aircraft flight-path angle is always between $-\pi/2$ and $\pi/2$, a reasonable assumption for civil fix-wing aircraft, which are the main focus of this work. Note that x'', y'', z'', γ', ψ' and v' may be discontinuous.

In order to time-parameterize an arbitrary path, it is sufficient to obtain the history of the speed $v(s)$ with respect to the path coordinate s. After the optimal speed profile $v^*(s)$ is obtained, the corresponding optimal time-parameterization of the trajectory can be calculated by integrating (76). Specifically, let $t^* : [s_0, s_f] \to [0, t_f]$ be the bijective mapping between the path coordinate and the corresponding time coordinate along the optimal solution. Then $t^*(s)$ denotes the time at which the aircraft arrives at the position corresponding to the path coordinate s. Since $\mathrm{d}t^* = \mathrm{d}s/v^*(s)$, it follows that the optimal time profile along the path is given by

$$t^*(s) = \int_{s_0}^{s} \mathrm{d}t^* = \int_{s_0}^{s} 1/v^*(s)\,\mathrm{d}s, \qquad s_0 \le s \le s_f.$$

The optimal time-parameterization of the geometric trajectory $\big(x(s), y(s), z(s)\big)$ is then given by

$$\big(x^*(t), y^*(t), z^*(t)\big) = \big(x(t^{*-1}(t)), y(t^{*-1}(t)), z(t^{*-1}(t))\big).$$

It will be shown in Section 4.4 that the optimal thrust profile $T^*(s)$ along the path can be determined once $v^*(s)$ is known. Subsequently, the other controls can be recovered through inverse dynamics as follows:

$$\begin{aligned} C_L^*(s) &= \frac{2}{\rho v^{*2}(s) S}\Big(T^*(s) - m v^*(s) v^{*\prime}(s) - m g \sin\gamma(s)\Big), \\ \phi^*(s) &= -\arctan\left(\frac{\cos\gamma(s)\psi'(s)}{\gamma'(s) + g\cos\gamma(s)/v^{*2}(s)}\right). \end{aligned}$$

Obviously, the key to the optimal time-parameterization along a geometric path is the optimization of the speed profile along the given path. Next, we show how the state and control constraints of the problem can be mapped to a set of admissible velocity profiles in the s— $v^2/2$ plane. Later on, we will solve a scalar functional optimization problem to find the optimal speed profile. The solution of the latter problem will provide the optimal time-parameterization along the given path.

4.3 The Admissible Kinetic Energy Set

It is required that the lift coefficient C_L, the bank angle ϕ, and the thrust T must stay within certain ranges during the whole flight, namely,

$$C_L(s) \in [C_{L_{\min}}(s),\, C_{L_{\max}}(s)], \quad \phi(s) \in [\phi_{\min}(s), \phi_{\max}(s)], \quad T(s) \in [T_{\min}(s), T_{\max}(s)], \quad \forall s \in [s_0, s_f], \tag{82}$$

where $C_{L_{\min}}$, $C_{L_{\max}}$, $\phi_{\min}$, $\phi_{\max}$, $T_{\min}$ and $T_{\max}$ are piecewise analytic functions of s. These constraints account for limitations of the control inputs, which may depend on the location along the path. It is also required that the aircraft speed satisfies the bounds $v(s) \in [v_{\min}(s), v_{\max}(s)]$, where $v_{\min}$ and $v_{\max}$ are piecewise analytic functions with $v_{\min}(s) > 0$ for all $s \in [s_0, s_f]$. We will further assume that $C_{L_{\min}}(s) \leq 0 \leq C_{L_{\max}}(s)$, $-\pi/2 < \phi_{\min}(s) < 0 < \phi_{\max}(s) < \pi/2$, and $0 \leq T_{\min}(s) < T_{\max}(s)$, for all $s \in [s_0, s_f]$, and that the flight path angle satisfies $\gamma(s) \in (-\pi/2, \pi/2)$ for all $s \in [s_0, s_f]$. These are generic conditions for a civil fixed-wing aircraft in normal flight conditions. When the aircraft is flying at an abnormal condition (due to malfunction of the control surfaces/servo systems/engines, structure-damage, etc.), some of these assumptions may no longer hold. Nonetheless, the method introduced in this chapter may still be applied with minor modifications. In such cases, the bounds on C_L, ϕ and T in (137) have to be updated to account for the post-failure characteristics of the airplane.

Let $E \overset{\triangle}{=} v^2/2$ be the kinetic energy per unit mass of the aircraft. Also, let $E_{\max}(s) = v_{\max}^2(s)/2$ and $E_{\min}(s) = v_{\min}^2(s)/2$. In the sequel the *specific kinetic energy* E will be used in lieu of the aircraft speed v to simplify the ensuing analysis. The constraint on the speed of the aircraft requires that $E_{\min}(s) \leq E(s) \leq E_{\max}(s)$ for all $s \in [s_0, s_f]$.

4.3.1 Lift Coefficient Constraint

From equations (74) and (75), we have

$$\begin{aligned}\gamma' &= \frac{1}{2m}\rho S C_L \cos\phi - \frac{g\cos\gamma}{v^2}, &(83)\\ \psi' &= -\frac{\rho v^2 S C_L \sin\phi}{2mv^2\cos\gamma} = -\frac{\rho S C_L \sin\phi}{2m\cos\gamma}, &(84)\end{aligned}$$

which can be rewritten as:

$$\begin{aligned}C_L\cos\phi &= \frac{2m}{\rho S}\left(\gamma' + \frac{g\cos\gamma}{v^2}\right), &(85)\\ C_L\sin\phi &= -\frac{2m\psi'\cos\gamma}{\rho S}. &(86)\end{aligned}$$

Eliminating ϕ from equation (85) and (86), and replacing v^2 with $2E$, one obtains

$$E = g_1(C_L;\gamma,\gamma',\psi') \triangleq \frac{mg\cos\gamma}{\rho S}\left(\sqrt{C_L^2 - \left(\frac{2m\psi'\cos\gamma}{\rho S}\right)^2} - \frac{2m\gamma'}{\rho S}\right)^{-1}. \qquad (87)$$

The other solution is omitted because it is always negative. Note that the constraint $0 < E_{\min}(s) \leq E(s) \leq E_{\max}(s) < \infty$ for all $s \in [s_0, s_f]$ implies that there exists $C_L(s) \in [C_{L_{\min}}(s), C_{L_{\max}}(s)]$ such that

$$0 < \sqrt{C_L^2(s) - \left(\frac{2m\psi'(s)\cos\gamma(s)}{\rho(s)S}\right)^2} - \frac{2m\gamma'(s)}{\rho(s)S} < \infty, \qquad \forall s \in [s_0, s_f]. \qquad (88)$$

This is equivalent to the condition

$$\bar{C}_L(s) > \tilde{C}_L(s), \qquad \forall s \in [s_0, s_f], \qquad (89)$$

where

$$\bar{C}_L(s) = \max\{-C_{L_{\min}}(s), C_{L_{\max}}(s)\}, \qquad (90)$$

and

$$\tilde{C}_L(s) = \begin{cases} \dfrac{2m}{\rho(s)S}\,|\psi'(s)|\cos\gamma(s), & \text{if } \gamma'(s) < 0,\\[2ex] \dfrac{2m}{\rho(s)S}\sqrt{\gamma'^2(s) + \psi'^2(s)\cos^2\gamma(s)}, & \text{if } \gamma'(s) \geq 0.\end{cases} \qquad (91)$$

The given path $(x(s), y(s), z(s))$ is infeasible if (89) is not satisfied, owing to insufficient lift. When (89) holds, and because the right hand side of equation (87) is a monotonically decreasing function with respect to C_L^2, the limits on the lift coefficient impose a lower bound on the kinetic energy E as follows

$$E(s) \geq \underline{g}_{w1}(s) \overset{\triangle}{=} \max\{E_{\min}(s), g_1(\bar{C}_L(s); \gamma(s), \gamma'(s), \psi'(s))\}. \tag{92}$$

In other words, if the problem is feasible, (92) provides a lower bound on the allowable speed, whereas the bounds $C_{L_{\min}}(s) \leq C_L(s) \leq C_{L_{\max}}(s)$ on the lift coefficient do not impose any constraint on the maximum value of $E(s)$. Finally, note from (92) that if $\underline{g}_{w1}(s)$ is unbounded, then the path is not feasible. Feasibility implies, in particular, that $\underline{g}_{w1}$ in (92) is a (possibly discontinuous) piecewise analytic function of s.

4.3.2 Bank Angle Constraint

In order to consider the effect of the bank angle constraint on the specific kinetic energy E, we need to eliminate C_L from equations (85) and (86) and form an algebraic equation involving ϕ and E. However, two special cases need to be considered before proceeding with such an elimination: the case when $C_L(s) = 0$, and the case when $2\gamma'(s) + g\cos\gamma(s)/E(s) = 0$, for some $s \in [s_0, s_f]$.

If $C_L(s) = 0$ for some $s \in [s_0, s_f]$, then the lift is zero and the bank angle ϕ is indeterminate. In this case, the bounds $\phi_{\min}(s) \leq \phi(s) \leq \phi_{\max}(s)$ on the bank angle ϕ do not constrain the specific kinetic energy at s. Similarly, note that $2\gamma'(s) + g\cos\gamma(s)/E(s) = 0$ may hold only if $\gamma'(s) < 0$. If $2\gamma'(s) + g\cos\gamma(s)/E(s) = 0$, then $E(s)$ is uniquely determined, regardless of the value of the bank angle at s, i.e., the bank angle has no effect on E. Therefore, we only need to consider the cases with $C_L(s) \neq 0$ and $2\gamma'(s) + g\cos\gamma(s)/E(s) \neq 0$ for some $s \in [s_0, s_f]$ in order to eliminate C_L from equations (85) and (86), thus obtaining the following equation:

$$\tan\phi = -\frac{2\psi'\cos\gamma}{2\gamma' + g\cos\gamma/E}. \tag{93}$$

Solving for E from equation (93) yields:

$$E = g_2(\phi; \gamma, \gamma', \psi') \overset{\triangle}{=} -\frac{1}{2}\frac{g\cos\gamma\tan\phi}{\gamma'\tan\phi + \psi'\cos\gamma}. \tag{94}$$

The positivity of $E(s)$ requires that $g_2(\phi(s); \gamma(s), \gamma'(s), \psi'(s)) > 0$ for all $s \in [s_0, s_f]$, otherwise the path is infeasible. If $g_2(\phi; \gamma, \gamma', \psi') > 0$ along the given path, the constraints on E due to the bank angle bounds can be determined as follows:

(i) When $\psi'(s) = 0$, equation (93) implies that $\phi(s) = 0$, and the bounds of ϕ impose no constraints on $E(s)$.

(ii) When $\psi'(s) \neq 0$, two cases need to be considered:

(iia) If $\gamma'(s) = 0$, and since $\gamma \in (-\pi/2, \pi/2)$, it follows that $\cos\gamma \neq 0$, and we have

$$E(s) = g_2(\phi(s); \gamma(s), \gamma'(s), \psi'(s)) = -\frac{g\tan\phi(s)}{2\psi'(s)}.$$

The condition $g_2(\phi(s); \gamma(s), \gamma'(s), \psi'(s)) > 0$ requires that $\phi(s)\psi'(s) < 0$. The constraint on ϕ then leads to the following upper bound on the specific kinetic energy E

$$E(s) \leq \mu_0(s) \overset{\triangle}{=} \max\left\{\frac{g\tan\phi_{\min}(s)}{2\psi'(s)}, \frac{g\tan\phi_{\max}(s)}{2\psi'(s)}\right\}. \tag{95}$$

(iib) If $\gamma'(s) \neq 0$, rewrite (94) as follows:

$$\gamma'(s)\tan\phi(s) + \psi'(s)\cos\gamma(s) = -\frac{g\cos\gamma(s)}{2E(s)}\tan\phi(s). \tag{96}$$

The bank angle constraint $\phi(s) \in [\phi_{\min}(s), \phi_{\max}(s)]$ limits the admissible value of $E(s)$ via equation (96). A necessary and sufficient condition for the satisfaction of this constraint is

$$\frac{g\cos\gamma(s)}{2E(s)} \leq \mu_1(s) \overset{\triangle}{=} \min\left\{h(s; \phi_{\min}, \gamma, \gamma', \psi'), h(s; \phi_{\max}, \gamma, \gamma', \psi')\right\}, \tag{97}$$

or

$$\frac{g\cos\gamma(s)}{2E(s)} \geq \mu_2(s) \overset{\triangle}{=} \max\left\{-h(s; \phi_{\min}, \gamma, \gamma', \psi'), -h(s; \phi_{\max}, \gamma, \gamma', \psi')\right\}, \tag{98}$$

where

$$h(s;\phi,\gamma,\gamma',\psi') \triangleq \gamma'(s) + \psi'(s)\cos\gamma(s)/\tan\phi(s). \tag{99}$$

In order to characterize the constraint on E induced by the bank angle, three subcases are analyzed, and the results are given below:

(iib.1) If $\mu_1(s) \leq 0$ and $\mu_2(s) \leq 0$, then equation (98) always holds as long as $E(s) > 0$.

(iib.2) If $\mu_1(s) \leq 0$ and $\mu_2(s) > 0$, then equation (97) does not hold, and equation (98) must be satisfied, which is equivalent to the following constraint on $E(s)$

$$E(s) \leq \frac{1}{2} g \cos\gamma(s)/\mu_2(s). \tag{100}$$

(iib.3) Finally, if $\mu_1(s) > 0$, then it is required that either (100) holds, or the following inequality holds:

$$E(s) \geq \frac{1}{2} g \cos\gamma(s)/\mu_1(s). \tag{101}$$

Equations (95), (100) and (101) define the admissible values of $E(s)$ limited by the bank angle.

4.3.3 Summary of Algebraic Constraints

In the previous two sections it has been shown that the lift coefficient and the bank angle constraints can be reduced to a series of algebraic constraints on the value of the specific kinetic energy E along the path. Summarizing these results, for feasibility, the specific kinetic energy profile E must satisfy either one, or both, of the following two constraints. The first constraint is defined according to the inequalities

$$\underline{g}_{w1}(s) \leq E(s) \leq \overline{g}_{w1}(s), \qquad s \in [s_0, s_f], \tag{102}$$

where $\underline{g}_{w1}(s)$ from (92) and $\overline{g}_{w1}(s)$ from

$$\overline{g}_{w1}(s) \triangleq \begin{cases} \min\left\{E_{\max}(s), \mu_0(s)\right\}, & s \in \Gamma_1, \\ \min\left\{E_{\max}(s), g\cos\gamma(s)/2\mu_2(s)\right\}, & s \in \Gamma_2 \cup \Gamma_3, \\ E_{\max}(s), & \text{otherwise,} \end{cases}$$

where

$$\Gamma_1 = \left\{s|\psi'(s) \neq 0, \gamma'(s) = 0, s \in [s_0, s_f]\right\},$$

$$\Gamma_2 = \left\{s|\psi'(s) \neq 0, \gamma'(s) \neq 0, \mu_1(s) \leq 0, \mu_2(s) > 0, s \in [s_0, s_f]\right\},$$

$$\Gamma_3 = \left\{s|\psi'(s) \neq 0, \gamma'(s) \neq 0, \mu_1(s) > 0, s \in [s_0, s_f]\right\},$$

The second constraint is defined according to the inequalities

$$\underline{g}_{w2}(s) \triangleq \max\left\{\underline{g}_{w1}(s), \underline{g}_{w3}(s)\right\} \leq E(s) \leq \bar{g}_{w2}(s), \qquad s \in [s_0, s_f]. \tag{103}$$

where,

$$\underline{g}_{w3}(s) \triangleq \begin{cases} \max\{E_{\min}(s), g\cos\gamma(s)/2\mu_1(s)\}, & s \in \Gamma_3, \\ E_{\min}(s), & s \in [s_0, s_f]/\Gamma_3, \end{cases} \tag{104}$$

and

$$\overline{g}_{w2}(s) \triangleq \begin{cases} E_{\max}(s), & s \in \Gamma_3, \\ \overline{g}_{w1}(s), & s \in [s_0, s_f]/\Gamma_3. \end{cases} \tag{105}$$

and where $\mu_0(s), \mu_1(s)$ and $\mu_2(s)$ are given in (95), (97) and (98), respectively.

The collection of points $(s, E(s))$ satisfying either (102) or (103) correspond to the set $\mathcal{W} = \mathcal{W}_1 \cup \mathcal{W}_2$ in the $s - E$ plane, where $\mathcal{W}_1$ and $\mathcal{W}_2$ are given by

$$\mathcal{W}_1 = \{(s, E) \mid \underline{g}_{w1}(s) \leq E \leq \overline{g}_{w1}(s), s \in [s_0, s_f]\}, \tag{106}$$

$$\mathcal{W}_2 = \{(s, E) \mid \underline{g}_{w2}(s) \leq E(s) \leq \overline{g}_{w2}(s), s \in [s_0, s_f]\}, \tag{107}$$

Consequently, the given geometric path is feasible only if there exists a continuous function E, whose graph lies entirely in $\mathcal{W}$, while connecting the initial and final boundary conditions. We will thus always assume that $(s_0, E(s_0)) \in \mathcal{W}$ and $(s_f, E(s_f)) \in \mathcal{W}$ otherwise the problem is clearly infeasible.

4.3.4 Topological Properties of the Admissible Velocity Set

Before proceeding with the determination of the optimal velocity profile inside the admissible velocity set $\mathcal{W}$, some observations regarding the topological properties of $\mathcal{W}$ and its boundary are in order.

1. If $\mathcal{W}$ is not connected, then the given path is not feasible.

2. Even if the admissible velocity set $\mathcal{W}$ is connected, it may not be simply connected. If $\mathcal{W}$ is simply connected, then exists two piecewise analytic unctions $\underline{g}_w$ and $\overline{g}_w$ such that

$$\mathcal{W} = \{(s, E) | \underline{g}_w(s) \leq E(s) \leq \overline{g}_w\,(s)\,, s \in [s_0, s_f]\}. \tag{108}$$

For instance, one can simply take $\underline{g}_w = \min\{\underline{g}_{w1}, \underline{g}_{w2}\}$ and $\overline{g}_w = \max\{\overline{g}_{w1}, \overline{g}_{w2}\}$.

3. In case $\mathcal{W}$ is not simply connected, then it cannot be characterized by inequalities involving only two piecewise analytic functions as in (108). Such a situation will occur if there exist points $s \in [s_0, s_f]$ such that $\underline{g}_{w1}(s) > \overline{g}_{w2}(s)$ or $\underline{g}_{w2}(s) > \overline{g}_{w1}(s)$, for instance. Nonetheless, owing to the piecewise analyticity of the functions involved in (106) and (107), which represent the boundaries of $\mathcal{W}_1$ and $\mathcal{W}_2$ between s_0 and s_f, respectively, these functions may intersect at only at a finite number of points in $[s_0, s_f]$ (see Proposition 4.2.1). Consequently, there can only be a finite number of "holes" in $\mathcal{W}$.

4. Suppose $\mathcal{W}$ is not simply connected, but it rather has m holes. In this (rather rare) case, $\mathcal{W}$ can be decomposed as the union of 2^m simply connected subsets, as illustrated in Fig. 21 for the case when $m = 1$. After such a decomposition, each subset is searched for an optimal kinetic energy profile candidate using the approach described later on in the chapter. Once all possible (at most 2^m) candidates have been obtained, they are compared to identify the unique optimal kinetic energy profile for the original set $\mathcal{W}$.

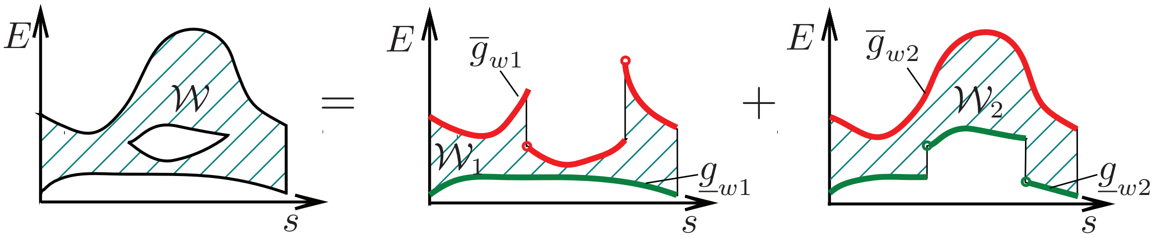

Figure 21: Decomposition of $\mathcal{W}$ when it is not simply connected.

In this work, we focus on the simple—and most common—case when $\mathcal{W}$ is simply connected and hence $\mathcal{W}$ is defined by algebraic constraints of the form $\underline{g}_w(s) \leq E(s) \leq$

$\overline{g}_w(s)\,, s \in [s_0, s_f]$, where $\underline{g}_w$ and $\overline{g}_w$ are appropriately defined piecewise analytic functions.

4.3.5 Thrust Constraint

From equations (73), (85) and (86) we have the following equation:

$$\begin{aligned} T &= mvv' + \left(\frac{1}{2}C_{D_0}\rho S + \frac{2Km^2\gamma'^2}{\rho S} + \frac{Km^2\cos^2\gamma\psi'^2}{\rho S}\right)v^2 \\ &\quad + \frac{2Km^2g^2\cos^2\gamma}{\rho S}\frac{1}{v^2} + \frac{4Km^2\gamma' g\cos\gamma}{\rho S} + mg\sin\gamma. \end{aligned} \tag{109}$$

Note that $vv' = v\frac{\mathrm{d}v}{\mathrm{d}s} = \frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{v^2}{2}\right) = E'$ and the above equation can be rewritten as a constraint on the derivative of E as follows

$$E'(s) = \frac{T(s)}{m} + c_1(s)E(s) + \frac{c_2(s)}{E(s)} + c_3(s), \tag{110}$$

where

$$c_1(s) \triangleq -\frac{C_{D_0}(s)\rho(s)S}{m} - \frac{4Km\gamma'^2(s)}{\rho(s)S} - \frac{2Km\cos^2\gamma(s)\psi'^2(s)}{\rho(s)S}, \tag{111}$$

$$c_2(s) \triangleq -\frac{Kmg^2\cos^2\gamma(s)}{\rho(s)S}, \tag{112}$$

$$c_3(s) \triangleq -\frac{4Km\gamma'(s)g\cos\gamma(s)}{\rho(s)S} - g\sin\gamma(s). \tag{113}$$

Note that c_1, c_2 and c_3 are piecewise analytic functions with respect to the path length coordinate s.

4.4 Optimal Control Formulation

The extensive analysis of the previous section reveals that instead of working with the original dynamical system described by (70) - (75), we only need to solve an optimal control problem with a single state variable E and a single control input T. For the case of robotic manipulators[118] it has been proved that the control is bang-bang when the speed limit is not active. In this section we show a similar result for the thrust control of a fixed-wing aircraft. Although the bang-bang form of the control for robotic manipulators has been proved in Ref. [118], the switching structure between the upper and lower control bounds has not been studied, despite the fact that the appropriate structure has been used implicitly in the algorithms proposed in Refs. [28, 119, 96, 118, 116]. In this section we prove that for the case of a fixed-wing aircraft, the thrust control switching structure is unique when the speed constraint is not active.

The optimal thrust profile $T^*(s)$ and the corresponding optimal speed $v^*(s) = \sqrt{2E^*(s)}$ for the minimum-time travel of a fixed-wing aircraft are given by the solution to the following optimal control problem:

Problem 4.4.1 (Minimum-Time Path-Tracking Problem). Consider the following optimal control problem in Lagrange form:

$$\begin{aligned}
\min_T \quad & J(s_0, s_f, E(s_0), E(s_f), T) = t_f = \int_{s_0}^{s_f} \frac{\mathrm{d}s}{\sqrt{2E(s)}} && \text{(114a)}\\
\text{subject to} \quad & E'(s) = \frac{T(s)}{m} + c_1(s)E(s) + \frac{c_2(s)}{E(s)} + c_3(s), && \text{(114b)}\\
& \underline{g}_w(s) \leq E(s) \leq \overline{g}_w(s), && \text{(114c)}\\
& E(s_0) = v_0^2/2, && \text{(114d)}\\
& E(s_f) = v_f^2/2, && \text{(114e)}\\
& T_{\min} \leq T(s) \leq T_{\max}, && \text{(114f)}
\end{aligned}$$

where v_0 and v_f are the required initial and final speed at s_0 and s_f, respectively, and $\overline{g}_w$ and $\underline{g}_w$ are piecewise analytic functions, computed in Section 4.3.

Note that we can always redefine the value of $\overline{g}_w$ and $\underline{g}_w$ at their (necessarily finite) points of discontinuities to make them either left or right continuous. In particular, and without loss of generality, in this work we assume that at the point of discontinuity, the value of $\overline{g}_w$ is defined so that it is lower semi-continuous and the value of $\underline{g}_w$ is defined so that it is upper semi-continuous. The reasons for such an assumption will be explained later in Section 4.5. The functions c_1, c_2 and c_3 are also piecewise analytic, and are given in equations (141), (142) and (143). They can be readily computed once the path is given.

Consider the case when the state constraint (114c) is not active. The Hamiltonian of the optimal control problem is

$$\begin{aligned}
H(E, \lambda, T, s) &= \frac{1}{\sqrt{2E}} + \lambda\left(\frac{T}{m} + c_1 E + \frac{c_2}{E} + c_3\right)\\
&= \frac{2\lambda}{m}T + c_1\lambda E + c_2\frac{\lambda}{E} + \frac{1}{\sqrt{2E}} + c_3\lambda,
\end{aligned}$$

The costate equation is

$$\lambda' = -\frac{\partial H}{\partial E} = -c_1\lambda + c_2\lambda E^{-2} + \frac{1}{2\sqrt{2}}E^{-3/2}. \tag{115}$$

The optimal control consists of constrained (i.e., $E(s) = \underline{g}_w(s)$ or $E(s) = \overline{g}_w(s)$) and unconstrained (i.e., $\underline{g}_w(s) < E(s) < \overline{g}_w(s)$) arcs. Furthermore, the control T enters linearly into the Hamiltonian, so a singular control may exist. The switching function is

$$\frac{\partial H}{\partial T} = \frac{2\lambda}{m}. \tag{116}$$

According to the Pontryagin's Maximum Principle, depending on the sign of the switching function, the optimal control may switch between the two bounds $T_{\min}$, $T_{\max}$ and the singular control when the state constraints are not active. Correspondingly, in general, the optimal control T^* of Problem 4.4.1 may contain bang-bang control, singular control, and control arcs associated with active state constraints, as described by the following expression

$$T^*(s) = \begin{cases} T_{\min}, & \text{for } \lambda > 0, s \in [s_0, s_f] \setminus \mathcal{K}, \\ \text{singular control}, & \text{for } \lambda = 0, s \in [s_0, s_f] \setminus \mathcal{K}, \\ T_{\max}, & \text{for } \lambda < 0, s \in [s_0, s_f] \setminus \mathcal{K}, \\ \overline{T}_w(s), & \text{for } s \in \mathcal{K}_U, \\ \underline{T}_w(s), & \text{for } s \in \mathcal{K}_L. \end{cases} \tag{117}$$

where $\mathcal{K}_U = \{s | E^*(s) = \overline{g}_w(s),\ s \in [s_0, s_f]\}$, $\mathcal{K}_L = \{s | E^*(s) = \underline{g}_w(s),\ s \in [s_0, s_f]\}$, and $\mathcal{K} = \mathcal{K}_U \cup \mathcal{K}_L$. At the points where the function $\overline{g}_w$ (respectively, $\underline{g}_w$) is differentiable, the value of the thrust $\overline{T}_w(s)$ (respectively, $\underline{T}_w$) is computed by

$$\overline{T}_w(s) = m\left(\overline{g}'_w(s) - c_1(s)\overline{g}_w(s) - c_3(s) - c_2(s)/\overline{g}_w(s)\right), \tag{118}$$

and respectively,

$$\underline{T}_w(s) = m(\underline{g}'_w(s) - c_1(s)\underline{g}_w(s) - c_3(s) - c_2(s)/\underline{g}_w(s)). \tag{119}$$

At the points where $\overline{g}_w$ (respectively, $\underline{g}_w$) is discontinuous and/or non-differentiable, the thrust is discontinuous, and can be computed by

$$\overline{T}_w(s^\pm) = m\left(\overline{g}'_w(s^\pm) - c_1(s)\overline{g}_w(s^\pm) - c_3(s) - c_2(s)/\overline{g}_w(s^\pm),\right) \tag{120}$$

and

$$\underline{T}_w(s^{\pm}) = m(\underline{g}_w'(s^{\pm}) - c_1(s)\underline{g}_w(s^{\pm}) - c_3(s) - c_2(s)/\underline{g}_w(s^{\pm})), \tag{121}$$

for the two cases. Note that owing to the piecewise continuous differentiability of $\overline{g}_w$ and $\underline{g}_w$ the limits $\overline{g}_w'(s^{\pm}), \overline{g}_w(s^{\pm})$ and $\underline{g}_w'(s^{\pm}), \underline{g}_w(s^{\pm})$ exist for all $s \in [s_0, s_f]$. Furthermore, the number of points at which $\overline{g}_w'(s^+) \neq \overline{g}_w'(s^-)$ or $\underline{g}_w'(s^+) \neq \underline{g}_w'(s^-)$ or $\overline{g}_w(s^+) \neq \overline{g}_w(s^-)$ or $\underline{g}_w(s^+) \neq \underline{g}_w(s^-)$ is finite.

Proposition 4.4.1. The optimal control solution of Problem 4.4.1 does not contain any singular control.

Proof. We only need to show that there does not exist any sub-interval $[s_a, s_b] \subseteq [s_0, s_f]$ on which $\lambda(s) \equiv 0$ and $\underline{g}_w(s) < E(s) < \overline{g}_w(s)$ (strict inequalities) for all $s \in [s_a, s_b]$. Suppose, ad absurdum, that $\lambda(s) \equiv \lambda'(s) \equiv 0$ for all $s \in [s_a, s_b]$, and the state constraints are not active on $[s_a, s_b]$. It follows that on $[s_a, s_b]$, equation (115) becomes

$$0 = \frac{1}{2\sqrt{2}} E^{-3/2} > 0,$$

which is impossible. Hence λ cannot remain constantly zero on any nontrivial interval, and the proof is complete. □

Proposition 4.4.2. The optimal control $T^*(s)$ is bang-bang, and does not contain any switch from $T_{\min}$ to $T_{\max}$ on $[s_0, s_f] \setminus \mathcal{K}$.

Proof. Since we have shown that a singular control does not exist, the control history must be bang-bang on $[s_0, s_f] \setminus \mathcal{K}$. We only need to prove that, when the constraint (114c) is inactive, there does not exist a switching from $T_{\min}$ to $T_{\max}$ in the optimal control history.

To this end, suppose, on the contrary, that T^* contains a switching from $T_{\min}$ to $T_{\max}$ at some $s_m \in (s_a, s_b] \subset ([s_0, s_f] \setminus \mathcal{K})$, such that

$$T^* = \begin{cases} T_{\min}, & s_a < s \leq s_m, \\ T_{\max}, & s_m < s \leq s_b. \end{cases}$$

For simplicity, and without loss of generality, we will assume that the functions c_1, c_2 and c_3 are continuous at s_m.

Let η be a small positive scalar, and let $E_m^-(s;\eta)$ and $E_m^+(s;\eta)$ denote the trajectories passing through $(s_m, E^*(s_m)+\eta)$, with control $T_{\min}$ and $T_{\max}$, respectively. From the definitions of $E_m^-(s;\eta)$ and $E_m^+(s;\eta)$, we have the following expressions

$$\begin{aligned} {E_m^-}'(s;\eta) - {E_m^+}'(s;0) &= (T_{\min} - T_{\max})/m + c_1(s)\left(E_m^-(s;\eta) - E_m^+(s;0)\right) + c_2(s)\left(\frac{1}{E_m^-(s;\eta)} - \frac{1}{E_m^+(s;0)}\right) \\ &= (T_{\min} - T_{\max})/m + \left(c_1(s) - \frac{c_2(s)}{E_m^-(s;\eta)E_m^+(s;0)}\right)\left(E_m^-(s;\eta) - E_m^+(s;0)\right) \\ &\le (T_{\min} - T_{\max})/m + \left|c_1(s) - \frac{c_2(s)}{E_m^-(s;\eta)E_m^+(s;0)}\right|\left|E_m^-(s;\eta) - E_m^+(s;0)\right|. \end{aligned} \tag{122}$$

and

$$\begin{aligned} {E_m^-}'(s;0) - {E_m^+}'(s;\eta) &= (T_{\min} - T_{\max})/m + c_1(s)\left(E_m^-(s;0) - E_m^+(s;\eta)\right) + c_2(s)\left(\frac{1}{E_m^-(s;0)} - \frac{1}{E_m^+(s;\eta)}\right) \\ &= (T_{\min} - T_{\max})/m + \left(c_1(s) - \frac{c_2(s)}{E_m^-(s;0)E_m^+(s;\eta)}\right)\left(E_m^-(s;0) - E_m^+(s;\eta)\right) \\ &\le (T_{\min} - T_{\max})/m + \left|c_1(s) - \frac{c_2(s)}{E_m^-(s;0)E_m^+(s;\eta)}\right|\left|E_m^-(s;0) - E_m^+(s;\eta)\right|. \end{aligned} \tag{123}$$

Note that $E_m^+(s_m;\eta) = E_m^-(s_m;\eta) = E^*(s_m) + \eta$. We therefore have

$$\begin{aligned} \left|E_m^-(s;\eta) - E_m^+(s;0)\right| &= \left|E_m^-(s;\eta) - E_m^-(s_m;\eta) + E_m^+(s_m;\eta) - E_m^+(s_m;0) + E_m^+(s_m;0) - E_m^+(s;0)\right| \\ &\le |E_m^-(s;\eta) - E_m^-(s_m;\eta)| + |E_m^+(s_m;\eta) - E_m^+(s_m;0)| + |E_m^+(s_m;0) - E_m^+(s;0)|. \end{aligned}$$

Since $E_m^-(s;\eta)$ and $E_m^+(s;0)$ are continuous with respect to s, and $E_m^+(s_m;\eta)$ is continuous with respect to η, and since the coefficient multiplying $|E_m^-(s;\eta) - E_m^+(s;0)|$ in (122) is bounded, it follows that ${E_m^-}'(s;\eta) - {E_m^+}'(s;0) \le (T_{\min} - T_{\max})/m < 0$ for η small enough and for all s in a small enough neighborhood of s_m. By the same token, we can also show that ${E_m^-}'(s;0) - {E_m^+}'(s;\eta) \le (T_{\min} - T_{\max})/m < 0$ for η small enough and for all s in a small enough neighborhood of s_m. Choose now $\epsilon > 0$ and $\eta_0 > 0$ such that for all $\eta < \eta_0$ and all $s \in (s_m - \epsilon, s_m + \epsilon)$ both the previous inequalities are satisfied and, in particular, ${E_m^-}'(s;\eta) - {E_m^+}'(s;0) < (T_{\min} - T_{\max})/2m < 0$ and ${E_m^-}'(s;0) - {E_m^+}'(s;\eta) < (T_{\min} - T_{\max})/2m < 0$ for all $s \in (s_m - \epsilon, s_m + \epsilon)$ and $0 < \eta < \eta_0$.

Notice that in the interval $(s_m - \epsilon, s_m + \epsilon)$, the optimal specific kinetic energy profile can be written equivalently as:

$$E^*(s) = \begin{cases} E_m^-(s;0), & s_m - \epsilon < s < s_m, \\ \\ E_m^+(s;0), & s_m < s < s_m + \epsilon. \end{cases}$$

Consider now the part of $E_m^+(s;\eta)$ with $s < s_m$ and the part of $E_m^-(s;\eta)$ with $s > s_m$. Since $E^*(s_m) < \liminf_{s \to s_m} \overline{g}_w(s)$, and since $\overline{g}_w(s)$ is lower semi-continuous, there exists a small positive real number η_1 such that, for all $\eta < \eta_1$, $E_m^+(s;\eta) < \overline{g}_w(s)$ for all $s_m - \epsilon < s \leq s_m$, and $E_m^-(s;\eta) < \overline{g}_w(s)$ for all $s_m < s \leq s_m + \epsilon$, that is, a sufficiently small change of the initial condition at s_m will not lead to the violation of the constraint $\overline{g}_w(s)$.

Let $\eta_2 = -\epsilon(T_{\min} - T_{\max})/2m > 0$, and let $0 < \eta < \min\{\eta_0, \eta_1, \eta_2\}$. At the point s_m, we have $E_m^-(s_m;\eta) - E^*(s_m) = E_m^-(s_m;\eta) - E_m^-(s_m;0) = E_m^-(s_m;\eta) - E_m^+(s_m;0) = \eta > 0$. Since $E_m^{-\prime}(s;\eta) - E_m^{+\prime}(s;0) < (T_{\min} - T_{\max})/2m$ for all $s \in (s_m, s_m + \epsilon)$, forward integration of $E_m^{-\prime}(s;\eta) - E_m^{+\prime}(s;0)$ from s_m results in $E_m^-(s;\eta) - E_m^+(s;0) < \eta + (T_{\min} - T_{\max})(s - s_m)/2m$ for all $s \in (s_m, s_m + \epsilon)$. Specifically, there exists $s_m^+ \in (s_m, s_m + \epsilon)$ such that $E_m^-(s_m^+;\eta) = E_m^+(s_m^+;0) = E^*(s_m^+)$.

A similar argument shows that there exists $s_m^- \in (s_m - \epsilon, s_m)$ such that $E_m^+(s_m^-;\eta) = E^*(s_m^-)$. See Fig. 22.

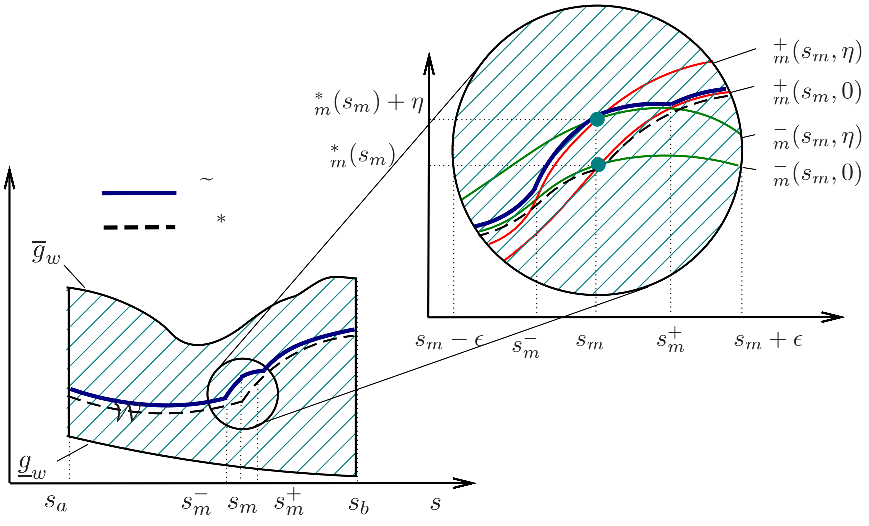

Figure 22: Speed variation for the proof of Proposition 4.4.2.

Now consider the variation of T^* (see Fig. 23) given by

$$\delta T = \begin{cases} T_{\max} - T_{\min}, & s_m^- < s \le s_m, \\ T_{\min} - T_{\max}, & s_m < s \le s_m^+, \\ 0, & \text{otherwise.} \end{cases}$$

Then with the new control $\widetilde{T} = T^* + \delta T$, the new speed profile $\widetilde{E}$ is composed of segments of E^*, $E_m^+(s;\eta)$ and $E_m^-(s;\eta)$, which is given below

$$\widetilde{E}(s) = \begin{cases} E^*(s), & s_a < s \le s_m^-, \\ E_m^+(s;\eta), & s_m^- < s \le s_m, \\ E_m^-(s;\eta), & s_m < s \le s_m^+, \\ E^*(s), & s_m^+ < s \le s_b. \end{cases}$$

The variation of speed is shown in Fig. 22. By construction of s_m^- and s_m^+, we have $E^*(s) < \widetilde{E}(s) < \overline{g}_w(s)$ for $s \in (s_m^-, s_m^+)$. Hence $J(s_a, s_b, E(s_a), E(s_b), T^*) > J(s_b, s_b, E(s_a), E(s_b), \widetilde{T})$, which means that T^* cannot be optimal. □

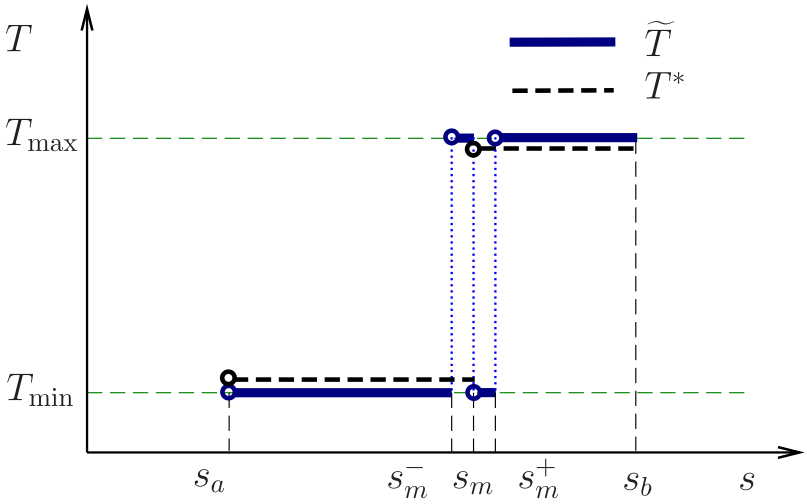

Figure 23: Thrust variation for proof of Proposition 4.4.2.

The next proposition shows that the lower bound $\underline{g}_w$ is generically not part of the optimal specific kinetic energy profile on a nontrivial interval.

Proposition 4.4.3. Assume $\overline{g}_w(s) \neq \underline{g}_w(s)$ and $T^*(s) = \underline{T}_w(s) < T_{\max}(s)$ for all $s \in [s_0, s_f]$. Let $E^*(s)$ be the optimal kinetic energy solution to Problem 4.4.1. Then the set $\mathcal{K}_L$ does not contain any nontrivial interval.

Proof. (Sketch) Assume, on the contrary, that there exists $(s_a, s_b) \in \mathcal{K}_L$ such that $E^*(s) = \underline{g}_w(s)$ for all $s \in (s_a, s_b)$, where $s_a \neq s_b$. Then since $\overline{g}_w(s) \neq \underline{g}_w(s)$ and $\underline{T}_w(s) < T_{\max}(s)$ on (s_a, s_b), one can construct a variation of the thrust T in the interval (s_a, s_b) similar to the proof of Proposition 4.4.2 that does not violate the thrust constraint, and which results in better time optimality, hence leading to a contradiction. We leave the details of the proof to the interested reader. □

Corolory 4.4.1. The time optimal control T^* for Problem 4.4.1 can be constructed as a combination of $T_{\max}$, $T_{\min}$ and $\overline{T}_w$.

Proof. Note that $T^*(s)$ is equal to $T_{\max}$, or $T_{\min}$, or $\overline{T}_w(s)$ on $[s_0, s_f] \setminus \mathcal{K}_L$. We only need to consider the value of $T^*(s)$ on $\mathcal{K}_L$. If $\overline{g}_w(s) = \underline{g}_w(s)$ on some nontrivial interval $[s_a, s_b]$, then clearly $T^*(s) = \underline{T}_w(s) = \overline{T}_w(s)$ for all $s \in [s_a, s_b]$, and the corollary holds on $[s_a, s_b]$. If $\underline{T}_w(s) = T_{\min}(s)$ for some $s \in [s_0, s_f]$, then the corollary trivially holds for such points. If $\overline{g}_w(s) = \underline{g}_w(s)$ only at isolated points, or if $\overline{g}_w(s) \neq \underline{g}_w(s)$ and $\underline{T}_w(s) < T_{\max}(s)$ for all $s \in [s_0, s_f]$, then $\mathcal{K}_L$ has an empty interior according to Proposition 4.4.3. □

4.5 Two Numerical Algorithms for Finding the Optimal Control

Recall that the admissible kinetic energy set $\mathcal{W}$ is determined by the geometry of the given path. Once the path is given, it is possible to find a semi-analytical solution of the optimal control problem (114a) using the necessary conditions introduced in the previous section.

Assuming that the given path is feasible, then according to Proposition 4.4.3, the lower bound $\underline{g}_w$ cannot be part of the optimal kinetic energy profile, except for the trivial case when $\underline{g}_w(s) = \overline{g}_w(s)$ over some part of $[s_0, s_f]$. The optimal kinetic energy profile is thus composed of three types of segments corresponding to maximum acceleration with $T^* = T_{\max}$, maximum deceleration with $T^* = T_{\min}$, and $T^* = \overline{T}_w$, the latter corresponding to the saturation of the upper state constraint $E(s) = \overline{g}_w(s)$. The most critical step of the optimal synthesis problem is to characterize which parts of $\overline{g}_w$ can possibly be saturated.

If $\overline{g}_w$ is continuous at $s_d \in [s_0, s_f]$ and $E^*(s_d) = \overline{g}_w(s_d)$, since $E^*(s)$ cannot violate the constraint $\overline{g}_w$, i.e., $E^*(s) \le \overline{g}_w(s)$, there exists a control $T^*(s) \in [T_{\min}, T_{\max}]$ such that $E^*(s)$ satisfies the following inequality

$$\frac{E^*(s_d + h) - E^*(s_d)}{h} \le \frac{\overline{g}_w(s_d + h) - \overline{g}_w(s_d)}{h}, \tag{124}$$

where h is a small positive real number. By taking the limits of both sides of (124) with $h \to 0$, the last expression leads to the existence of $T^*(s) \in [T_{\min}, T_{\max}]$ such that

$$E^{*\prime}(s_d^+) \le \overline{g}_w'(s_d^+). \tag{125}$$

On the other hand, we have

$$E^{*\prime}(s_d^+) \in \left[\frac{T_{\min}}{m} + c_1(s_d^+)E^*(s_d) + \frac{c_2(s_d^+)}{E^*(s_d)} + c_3(s_d^+), \frac{T_{\max}}{m} + c_1(s_d^+)E^*(s_d) + \frac{c_2(s_d^+)}{E^*(s_d)} + c_3(s_d^+)\right].$$

Therefore, (125) implies

$$\overline{g}_w'(s_d^+) \ge \frac{T_{\min}}{m} + c_1(s_d^+)E^*(s_d) + \frac{c_2(s_d^+)}{E^*(s_d)} + c_3(s_d^+) = \frac{T_{\min}}{m} + c_1(s_d^+)\overline{g}_w(s_d) + \frac{c_2(s_d^+)}{\overline{g}_w(s_d)} + c_3(s_d^+). \tag{126}$$

Similarly, the constraint $E^*(s) \le \overline{g}_w(s)$ for $s \in (s_d - \epsilon, s_d]$ implies

$$\overline{g}_w'(s_d^-) \le \frac{T_{\max}}{m} + c_1(s_d^-)E^*(s_d) + \frac{c_2(s_d^-)}{E^*(s_d)} + c_3(s_d^-) = \frac{T_{\max}}{m} + c_1(s_d^-)\overline{g}_w(s_d) + \frac{c_2(s_d^-)}{\overline{g}_w(s_d)} + c_3(s_d^-). \tag{127}$$

Therefore, $E^*(s_d) = \overline{g}_w(s_d)$ is possible only if both (126) and (127) are satisfied. In particular, when $\overline{g}_w$ is continuously differentiable at s_d, then $\overline{g}_w'(s_d^-) = \overline{g}_w'(s_d^+) = \overline{g}_w'(s_d)$ and hence, the inequalities (126) and (127) are reduced to

$$\frac{T_{\min}}{m} + c_1(s_d^+)\overline{g}_w(s_d) + \frac{c_2(s_d^+)}{\overline{g}_w(s_d)} + c_3(s_d^+) \le \overline{g}_w'(s_d) \le \frac{T_{\max}}{m} + c_1(s_d^-)\overline{g}_w(s_d) + \frac{c_2(s_d^-)}{\overline{g}_w(s_d)} + c_3(s_d^-). \tag{128}$$

If $\overline{g}_w$ is discontinuous at s_d, then either $\overline{g}_w(s_d) = \overline{g}_w(s_d^+)$ or $\overline{g}_w(s_d) = \overline{g}_w(s_d^-)$. In this case, the conditions $E^*(s_d) = \overline{g}_w(s_d)$ and $E^*(s) \le \overline{g}_w(s)$ in a neighborhood of s_d can be satisfied only if $\overline{g}_w(s)$ is lower semi-continuous (which we assume it is) and, in addition, (126) holds if $\overline{g}_w(s_d^+) < \overline{g}_w(s_d^-)$, and (127) holds if $\overline{g}_w(s_d^+) > \overline{g}_w(s_d^-)$.

Let $\widetilde{W}$ be the graph of all points in the interval $[s_0, s_f]$ such that $\overline{g}_w$ is continuous, and, in addition, (126) and (127) hold, that is,

$$\widetilde{W} \triangleq \{(s_d, \overline{g}_w(s_d)) | (126) \text{ and } (127) \text{ hold, } s_d \in [s_0, s_f]\} .$$

These are the points on the graph of $\overline{g}_w(s)$ which could possibly be part of the optimal kinetic energy profile $E^*(s)$. Furthermore, let $\widetilde{W}_d$ be the points on the graph of $\overline{g}_w$ where $\overline{g}_w$ is discontinuous (but necessarily lower semi-continuous), and either (126) or (127) holds. The points in $\widetilde{W}_d$ are the points of discontinuity of $\overline{g}_w$ which could be part of the optimal $E^*(s)$ profile.

Let $\overline{W} = \widetilde{W} \bigcup \widetilde{W}_d$ and let $\overline{W}^{\mathrm{c}} = \{(s, \overline{g}_w(s)),\ s \in [s_0, s_f]\} \backslash \overline{W}$. Generally, $\overline{W}$ is disconnected. Depending on the path, $\overline{W}$ may consist of multiple arcs and single points, as shown in Fig. 24. By the piecewise analyticity assumption of the given path, all functions involved in (126) and (127) are piecewise analytic, and it follows that the equality in (126) and (127) can only hold for a finite number of points on $[s_0, s_f]$. Hence, $\overline{W}$ is composed of only a finite union of disjoint components. That is, $\overline{W} = \bigcup_{j=1}^{N-1} \overline{W}_j$ for some positive integer N, where $\overline{W}_j$ are connected, and with $\overline{W}_i \bigcap \overline{W}_j = \varnothing$ for $i \neq j$. Let (s_j^-, E_j^-) and (s_j^+, E_j^+) denote the left and right end points of $\overline{W}_j$ for each $j = 1, \ldots, N-1$, where $E_j^- = \overline{g}_w(s_j^-)$ and $E_j^+ = \overline{g}_w(s_j^+)$ correspond to the "trajectory sink" and the "trajectory source" in Ref. [96]. Also, define two points $\overline{W}_0 = (s_0, E_0)$ and $\overline{W}_N = (s_f, E_f)$. Note that, in general, $\overline{W}_0 \neq \overline{W}_1$ and $\overline{W}_N \neq \overline{W}_{N-1}$. It is obvious that $\overline{W}_0$ and $\overline{W}_N$ must be part of the graph of the optimal kinetic energy profile.

For each $j = 1, \ldots, N-1$, let $\mathcal{S}_j^+$ denote the trajectory obtained by forward integration with maximum thrust, starting from s_j^+ with the initial value $\mathcal{S}_j^+(s_j^+) \triangleq E_j^+$, and similarly, let $\mathcal{S}_j^-$ be the trajectory obtained by backward integration using minimum thrust, starting from s_j^- with the initial value $\mathcal{S}_j^-(s_j^-) \triangleq E_j^-$. Forward integration with $T_{\max}$ and backward integration with $T_{\min}$ are also computed from the boundary points s_0 and s_f with initial conditions E_0 and E_f respectively, and the resulting trajectories are denoted with $\mathcal{S}_0^+$ and $\mathcal{S}_N^-$.

All current algorithms, including those in Refs. [28, 96, 118, 119], use a "search, integrate

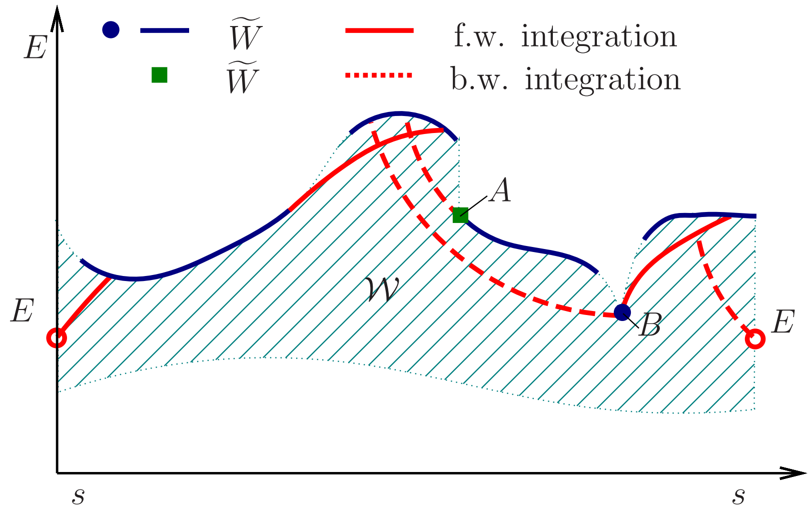

Figure 24: Elements for the optimal E

and check" procedure, which gradually extends the optimal speed profile from the initial point to the final point. Following this procedure, it is possible that during the search process, part of the already constructed trajectory has to be discarded because it cannot intersect $\overline{W}$ later on for any allowable thrust value.

In order to avoid such unnecessary computations, and to also improve the overall computational efficiency of the numerical scheme, it is necessary to characterize the elements in $\overline{W}$ which are part of E^*. Assuming feasibility of the problem, when the boundary conditions cannot be satisfied by a bang-bang control with no more than one switch from $T_{\max}$ to $T_{\min}$, some elements in $\overline{W}$ corresponding to the smaller values of $\overline{g}_w(s)$ must be active (at least at a single point) in the optimal solution, since these correspond to the most stringent/binding part of the constraint.

Following this observation, we introduce two new algorithms, which improve the numerical efficiency of the procedure for searching the optimal speed profile. The first algorithm is designed for parallel computation, while the second algorithm reduces the amount of computations devoted to the "search, integrate and check" process.

4.5.1 Algorithm I

Step 1 Compute $\overline{g}_w$, $\underline{g}_w$ as in Section 4.3.3 and check the feasibility of the geometric path.

Stop if the path is not feasible, otherwise proceed to the next step.

Step 2 Compute the feasible segments $\overline{W}_j$ on the graph of $\overline{g}_w$ following the procedure outlined in the previous section.

Step 3 Calculate $\mathcal{S}_j^+$ for $j = 0, 1, 2, \ldots, N-1$, with the integration terminated when $\overline{g}_w(s) = \mathcal{S}_j^+(s)$, or $s = s_f$. Let I_j^+ denote the interval of integration associated with $\mathcal{S}_j^+$. Also calculate $\mathcal{S}_j^-$ for $j = 1, 2, \ldots, N$, with the integration terminated when $\overline{g}_w(s) = \mathcal{S}_j^-(s)$, or $s = s_0$ and denote by I_j^- the corresponding intervals of integration of $\mathcal{S}_j^-$.

Step 4 Let

$$S^{\pm}(s) = \begin{cases} \mathcal{S}^{\pm}(s), & s \in I_j^{\pm}, \\ \overline{g}_w(s), & s \in [s_0, s_f] \backslash I_j^{\pm}, \end{cases} \tag{129}$$

for all $j = 0, 1, \ldots, N$, and let

$$E(s) \overset{\triangle}{=} \min\{S_0^+(s), S_1^+(s), \ldots, S_{N-1}^+(s), S_1^-(s), S_2^-(s), \ldots, S_N^-(s)\}. \tag{130}$$

If $E(0) = E_0$, $E(s_f) = E_f$ and $E(s) \geq \underline{g}_w(s)$ for all $s \in [s_0, s_f]$, then the optimal speed profile is given by (130). Otherwise the given path is not feasible.

The optimal speed profile is given by $v^*(s) = \sqrt{2E^*(s)}$, and the corresponding optimal thrust profile $T^*(s)$ can be computed by equation (109). By construction, the optimal thrust profile $T^*(s)$ satisfies the necessary conditions given by Proposition 4.4.2 and Theorem 4.4.1. The control T^* is indeed optimal because it maximizes point-wise the speed, and any further increase in speed results in the violation of the speed constraint.

Note that the "search, integrate and check" process is avoided in this algorithm. This algorithm can be implemented in parallel owing to the following reasons: (i) Step 1 and Step 4 can be performed point-wise for different $s \in [s_0, s_f]$; (ii) in Step 2 and 3 the computations of $\mathcal{S}_j^-$ and $\mathcal{S}_j^+$ are independent, hence they can be computed in parallel for different j at the same time.

The following algorithm still preserves the "search, integrate and check" process, but the repetition of the process is reduced to a minimum.

4.5.2 Algorithm II

Step 1 Compute $\overline{g}_w$, $\underline{g}_w$, and check the feasibility of the geometric path. Stop if the path is not feasible, otherwise proceed to the next step.

Step 2 Compute $\mathcal{S}_0^+(s)$ and $\mathcal{S}_N^-(s)$ with stopping criteria $\mathcal{S}_0^+(s) = \overline{g}_w(s)$ and $\mathcal{S}_N^-(s) = \overline{g}_w(s)$, or $s = s_0$, or $s = s_f$. Update $\overline{g}_w(s) \leftarrow \mathcal{S}_0^+(s)$ and $\overline{g}_w(s) \leftarrow \mathcal{S}_N^-(s)$ on the corresponding domain of integration.

Step 3 Compute $\overline{W}$ and its segments $\overline{W}_j$ on the graph of $\overline{g}_w$ following the procedure outlined previously. If $\overline{g}_w$ is continuous and $\overline{W}^c$ is empty, or if $\overline{g}_w(s_0) \neq E(s_0)$, or if $\overline{g}_w(s_f) \neq E(s_f)$, then go to Step 5. Otherwise, go to the next step.

Step 4 Among those $\overline{W}_j$ for which no integration has been performed at s_j^+ and s_j^-, select the one whose distance to the s axis is the smallest. Let its index be k. Compute $\mathcal{S}_k^-(s)$ and $\mathcal{S}_k^+(s)$ with the stopping criteria $\mathcal{S}_k^-(s) = \overline{g}_w(s)$ and $\mathcal{S}_k^+(s) = \overline{g}_w(s)$, or $s = 0$, or $s = s_f$. Update $\overline{g}_w(s) \leftarrow \mathcal{S}_k^-(s)$ and $\overline{g}_w(s) \leftarrow \mathcal{S}_k^+(s)$ on the corresponding domain of integration, and go to Step 3.

Step 5 If $\overline{g}_w(s_0) \neq E(s_0)$ or $\overline{g}_w(s_f) \neq E(s_f)$, then the given path is infeasible. Otherwise, the optimal speed profile is given by $E^* = \overline{g}_w$.

The difference between Algorithm II and Algorithm I (as well as the other time-optimal control algorithms in Refs. [28, 119, 96, 118]) is illustrated in Fig. 25. While Algorithm II computes only the integrations which are involved in the construction of the optimal speed profile, the algorithms in Refs. [28, 119, 96, 118] integrate the trajectory along arcs which may be discarded later on, when extending the optimal speed profile to the final point. Hence, they are in general less efficient when compared to Algorithm II.

4.6 Numerical Examples

In this section, two examples are used to test the feasibility and optimality of the proposed approach. Both examples implement Algorithm I, for simplicity. The first example focuses

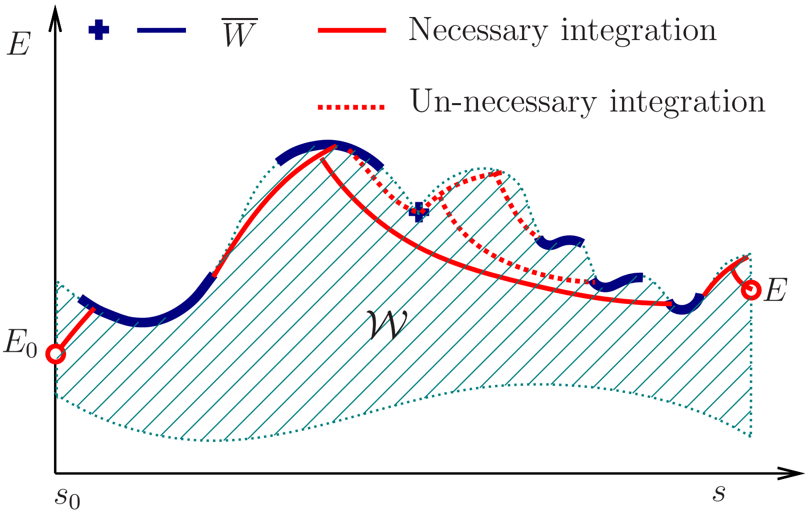

Figure 25: Algorithm comparison

on checking the feasibility of the algorithm, i.e., whether the controls given by the optimal parameterization method satisfy the prescribed bounds, and whether the aircraft can follow the path when using these control inputs. In the second example, the given path is a minimum-time path with known time parameterization, and is used to examine the optimality of the proposed method.

4.6.1 Landing Path with Two Turns

A three-dimensional path is used to test the feasibility of the trajectories obtained using the proposed time parameterization method. The trajectory is shown in Fig. 43. The initial position of the aircraft is $(0, 0, 6)$ km, the aircraft flies with $v_0 = 220\,\text{m/s}$, at $\gamma(0) = 0°$ path angle and $\psi(0) = 0°$ heading. The final position is (111.0,17.3,0) km, with final speed $v(s_f) = 95\,\text{m/s}$, path angle $\gamma(s_f) = 0°$ and heading $\psi(s_f) = -25°$. The horizontal projection of the trajectory contains two constant rate turning maneuvers. The atmospheric density data are taken from Ref. [91]. For simplicity, the change of gravity with altitude is neglected.

The control bounds are given as follows: the lift coefficient $C_L \in [-0.067, 1.9]$, the bank angle $\phi \in [-15°, 15°]$ and the thrust $T \in [0, 1126.3]$ kN. The maximum speed limit is 0.8 Mach, while the minimum speed limit is $v_{\min}$=60 m/s (134.2 mph). These data correspond approximately to a typical civilian airliner. Using the optimal time parameterization

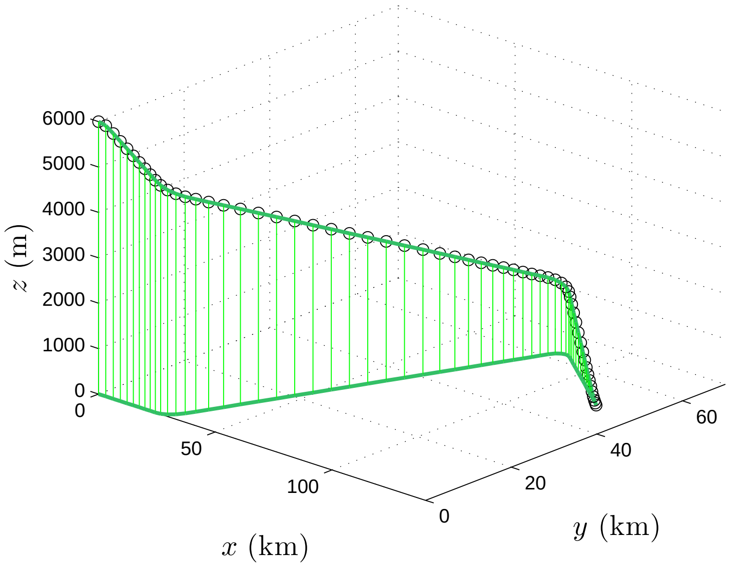

Figure 26: 3D Geometric Trajectory.

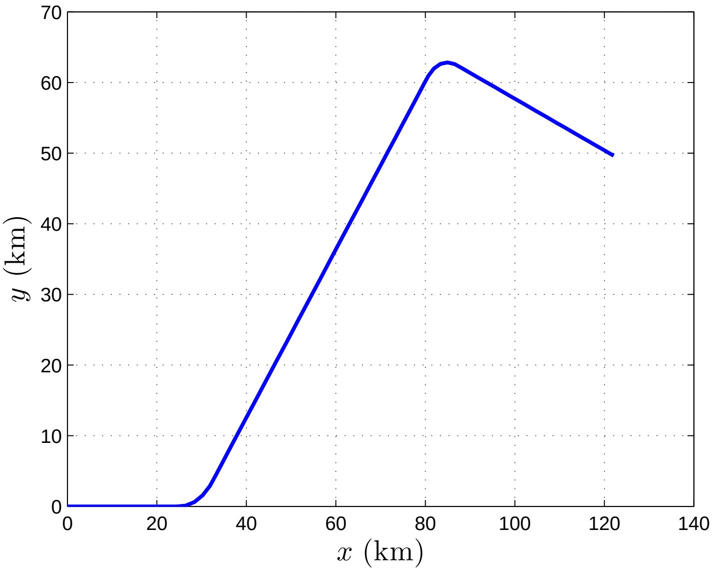

Figure 27: X-Y plane projection of the geometric trajectory.

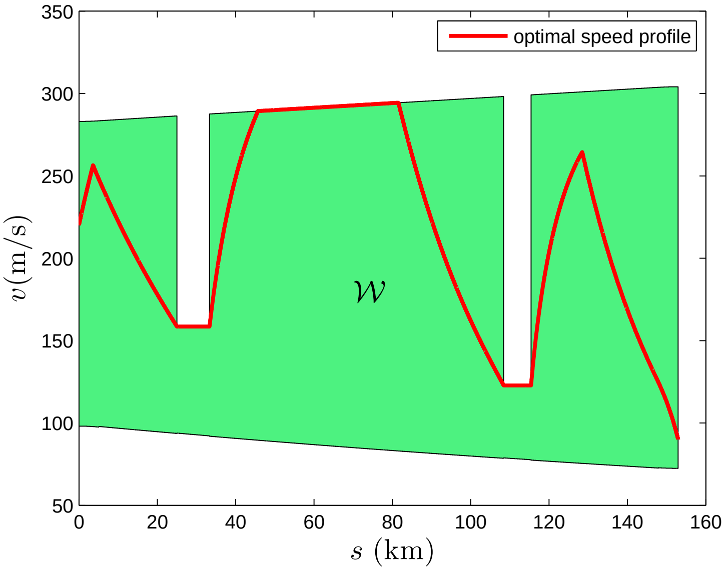

Figure 28: Optimal speed profile under path coordinate.

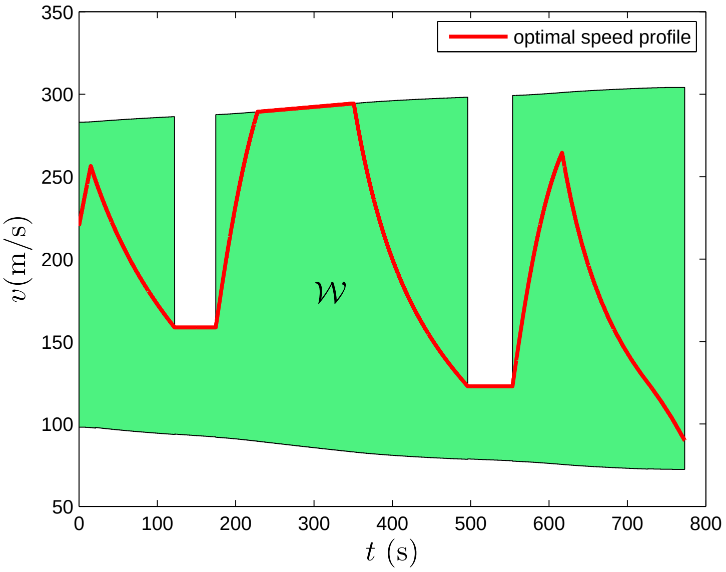

Figure 29: Time history of optimal speed.

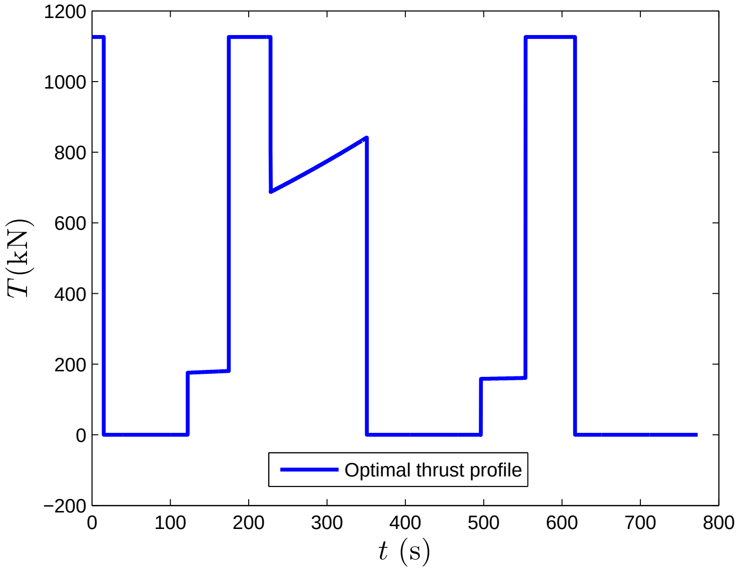

Figure 30: Optimal thrust.

method, the minimum-time speed profile $v^*(s)$ is computed following the approach developed in this chapter and is shown in Fig. 28. The same profile in terms of time is shown in Fig. 29. To arrive at the final position in minimum time, the aircraft should fly as fast as possible, however, due to the limited acceleration and deceleration capability, the optimal velocity profile cannot necessarily stay at $v_{\max}$ all the time. Within $0 \leq s \leq 25\,\mathrm{km}$, the upper limit of speed is higher than $270\,\mathrm{m/s}$, but the aircraft cannot travel at the maximum speed because it would not be able to decelerate sufficiently fast, thus violating the speed upper limit within $25 \leq s \leq 33\,\mathrm{km}$, which is induced by the first left turning maneuver. Similar scenarios exist before the second turning maneuver and the final point. The total length of the path is $152.9\,\mathrm{km}$, and the aircraft finishes in $771\,\mathrm{s}$ using the optimal thrust with an average speed of $170.4\,\mathrm{m/s}$.

The state and control histories recovered from the optimally time-parameterized trajectory are shown in Fig.31. The red dotted lines in the figures represent the control bounds. As shown in the figure, the thrust and bank angle saturate during some phases of the flight. The saturations of the bank angle are caused by the turning maneuvers. The saturation of the thrust leads to maximum acceleration which improves optimality.

To check the validity of this result, inverse dynamics are used to recover the state and

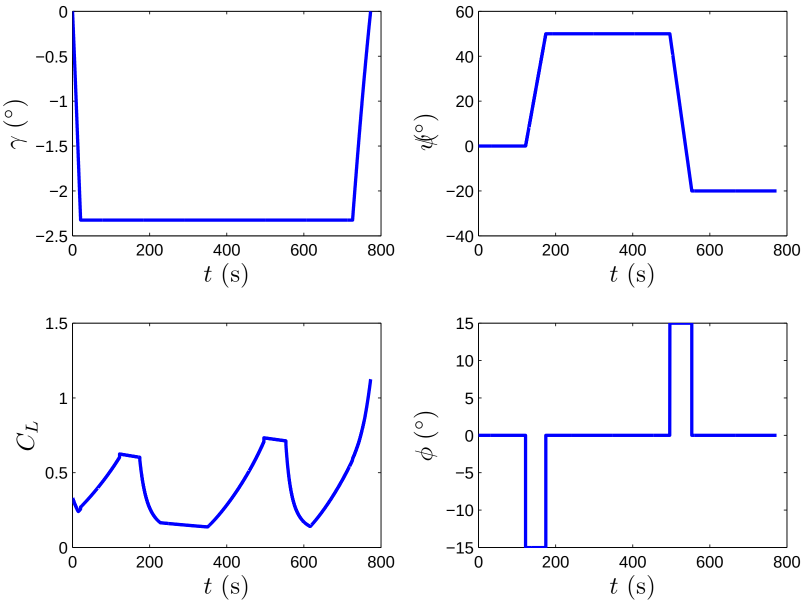

Figure 31: The states and control histories of the time parameterized trajectory.

control histories from the optimal time-parameterized trajectory $\big(x^*(t), y^*(t), z^*(t)\big)$. For the purpose of validation, after the control histories are calculated from inverse dynamics, they are used as the control inputs to simulate the trajectory. Specifically, the ordinary differential equations (131)-(136) are solved using the resulted control histories. The new simulated trajectory $(\hat{x}, \hat{y}, \hat{z})$ is compared with (x^*, y^*, z^*) in Fig. 32.

The discrepancy between the simulated trajectory and the original input trajectory is estimated using the following relative error index

$$\Delta_r = \max_t \sqrt{\left(\frac{\hat{x}\,(t) - x^*\,(t)}{\max\limits_t x^*\,(t) - \min\limits_t x^*\,(t)}\right)^2 + \left(\frac{\hat{y}\,(t) - y^*\,(t)}{\max\limits_t y^*\,(t) - \min\limits_t y^*\,(t)}\right)^2 + \left(\frac{\hat{z}\,(t) - z^*\,(t)}{\max\limits_t z^*\,(t) - \min\limits_t z^*\,(t)}\right)^2}$$

For this example, $\Delta_r = 4.1 \times 10^{-4}$, which is quite acceptable.

4.6.2 Time Optimal Path

In order to validate the optimality of the time-parameterized trajectory, a minimum-time landing path for a large civil aircraft is used to test the proposed method. The path is generated using DENMRA, which is a numerical algorithm solving optimal control problems with an automatic multiresolution mesh refinement scheme [138]. The accuracy and

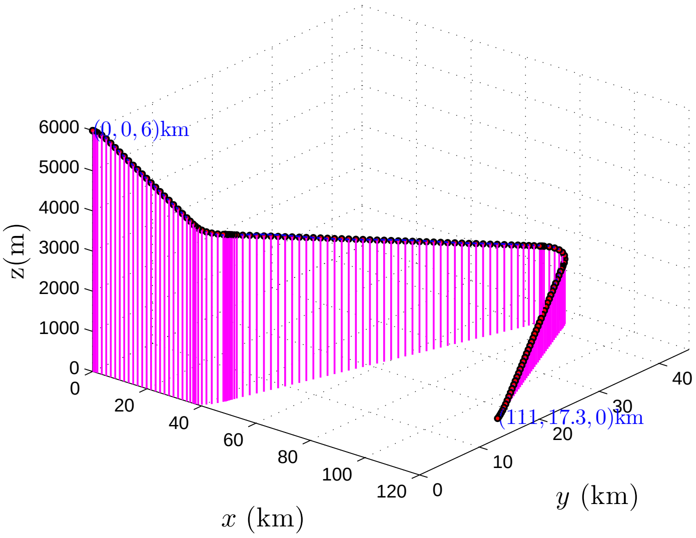

Figure 32: Comparison of the original geometric path(dots) and the path generated using time parameterization and inverse dynamics(line).

robustness of the DENMRA have been demonstrated in the same reference.

The aircraft starts at an initial position of $(0, 0, 10)$km, and lands at an airport with position (110,-60,0) km. The initial conditions are: speed $v(0) = 240\,\mathrm{m/s}$, heading angle $\psi(0) = 0°$ and the path angle $\gamma(0) = 0°$; the final conditions are: speed $v(s_f) = 95\,\mathrm{m/s}$, heading angle $\psi(s_f) = 80°$, and path angle $\gamma(s_f) = -3°$. The aircraft considered in this example is a Boeing-747. During the whole flight, the following constraints need to be satisfied: $v \leq 270\,\mathrm{m/s}$, $\phi \in [-15, 15]°$, $C_L \in [-0.31, 1.52]$, and $T \in [0, 1126.3]$ kN. The path is shown in Figs. 33 and 34.

Because the state and control histories obtained from DENMRA are already time optimal, it is expected that the application of the time-parameterization method to the path corresponding to the DENMRA solution should yield the same optimal solution as that of DENMRA. This is indeed the case, as it is evident from Figs. 37-40.

The optimal parameterization method gives a total travel time of 534.1 s, which matches very well with the final time of 533.8 s given by the DENMRA. The small discrepancy observed is attributed to numerical issues with the solvers. The admissible speed set $\mathcal{W}$ in terms of the path coordinate and time are shown in Figs. 35 and 36, respectively. The time history of the speed and the controls are shown in Figs. 37-40. As mentioned before, the

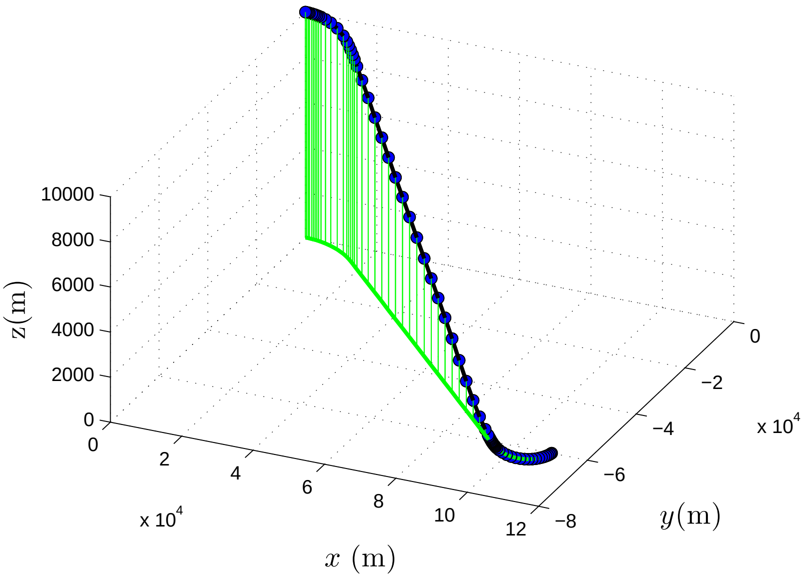

Figure 33: the min-time trajectory.

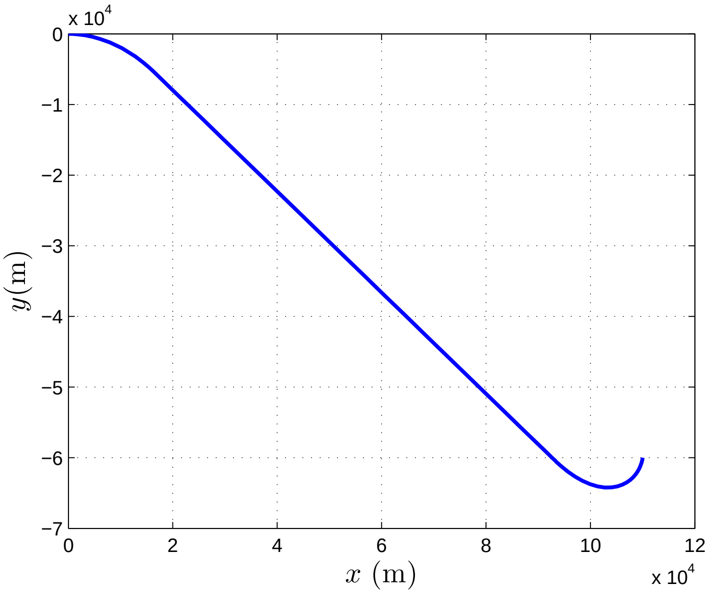

Figure 34: X-Y plane projection of the min-time trajectory.

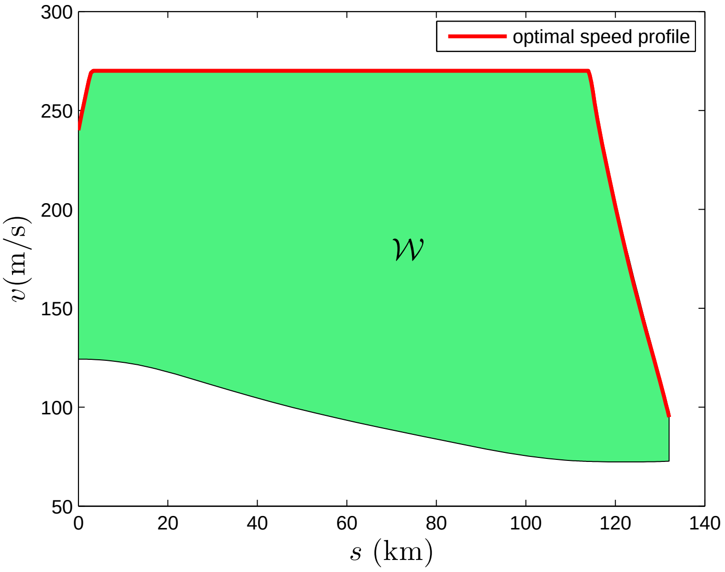

Figure 35: Optimal speed profile under path coordinate (DENMRA).

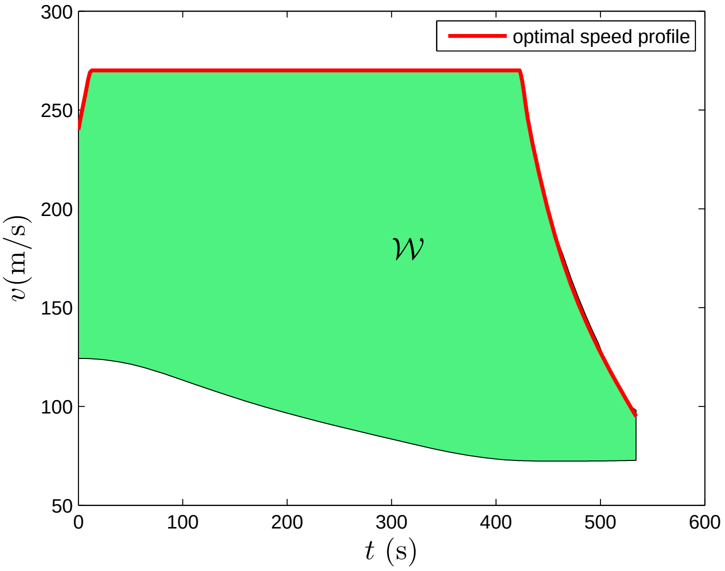

Figure 36: Time history of optimal speed (DENMRA).

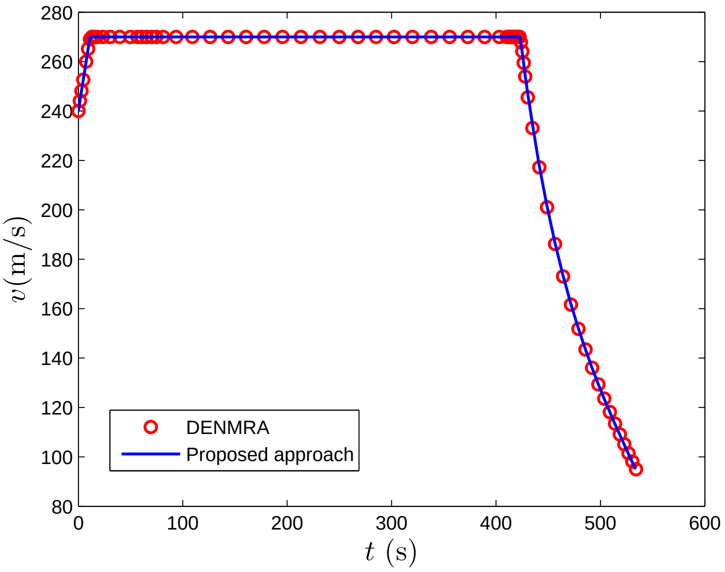

Figure 37: Speed comparison.

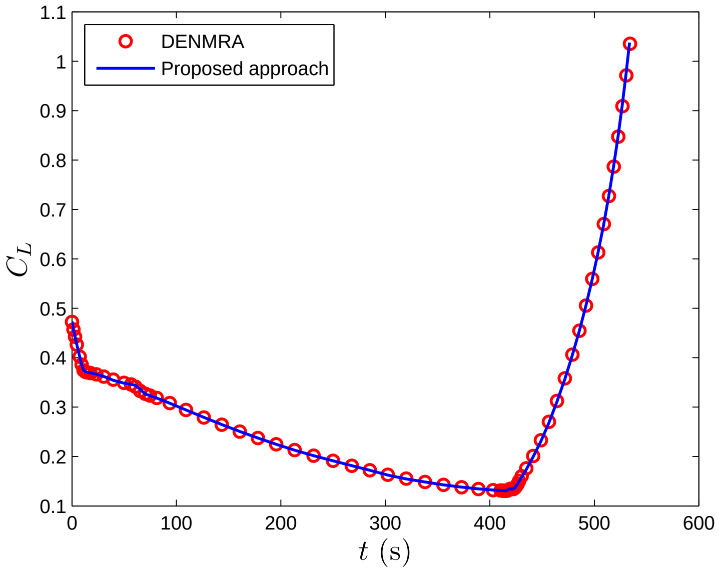

Figure 38: Control comparison: C_L.

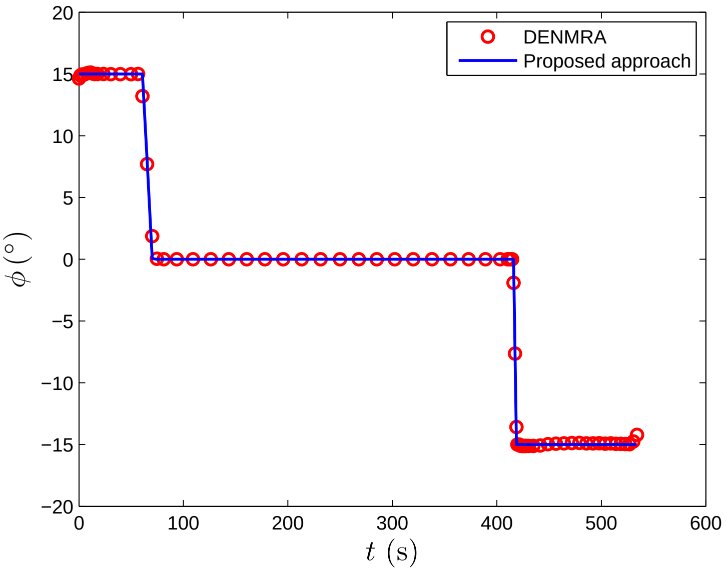

Figure 39: Control comparison: ϕ.

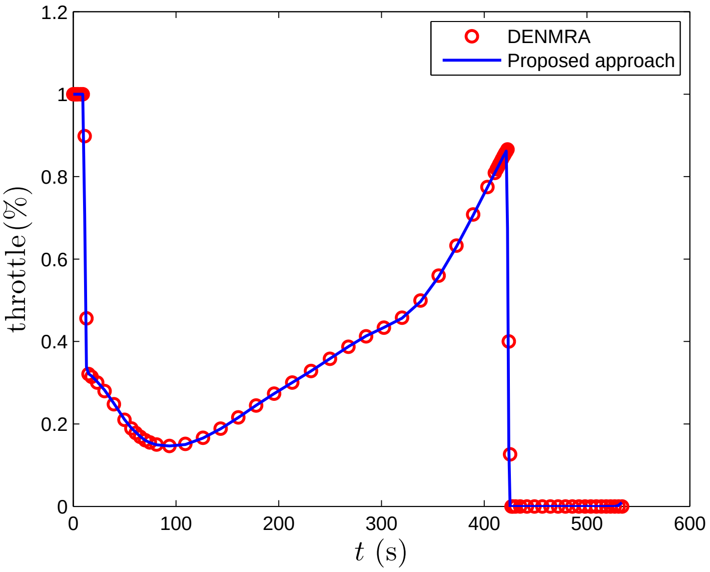

Figure 40: Control comparison: throttle.

other two states—the path angle γ and the heading angle ψ—are pure geometric variables, and are independent of parameterization, so they are not used for checking the optimality of the proposed method. As shown in Figs. 37-40, the numerical optimization result agrees very well with that of the time-parameterization method. This agreement validates the optimality of the time-parameterization method and, to some extent, that of DENMRA as well.

4.7 Conclusions

This chapter studies the problem of minimum-time-travel of a fixed-wing aircraft along a specified path. It has been proved that in an interval where the speed constraint is not active, there exists at most one switching, which is from maximum thrust to minimum thrust, hence the switching structure for the time-optimal control problem is unique. Constrained arcs riding on the upper bound of the admissible velocity are also part of the optimal trajectory. The admissible specific kinetic energy set is introduced to characterize the domain within which the optimal specific kinetic energy profile is searched. The main control (thrust) optimal history is then immediately determined from the optimal specific kinetic energy profile. The admissible specific kinetic energy set is generated by considering the constraints involving the aircraft speed and the remaining two controls, namely the lift coefficient and the bank angle. Hence, a search within the admissible specific kinetic energy set naturally satisfies these constraints. Two algorithms are proposed to solve for the thrust switching structure. The first algorithm can be implemented in parallel, which is difficult for other algorithms involving a sequential "search, integrate and check" pattern. The second algorithm is based on the "search, integrate and check" pattern, but improves its numerical efficiency by eliminating unnecessary integrations. Both algorithms are very efficient, and are thus amenable to real-time implementation.

It has been verified through numerical examples that the optimally time-parameterized trajectory satisfies the specified control bounds, and is indeed flyable with control histories obtained from the time-parameterized solution. The time-parameterization method, when combined with other fast geometric path-planning methods, leads to feasible trajectories with certain optimality characteristic, as opposed to just feasible paths.

CHAPTER V

ENERGY-OPTIMAL LANDING PATH TRACKING WITH FIXED TIME OF ARRIVAL

In this chapter, we presents a method for the energy-optimal operation of a fixed-wing aircraft tracking a prescribed landing path in the three-dimensional space with fixed Time Of Arrival (TOA). Following the same approach in Chapter 4, such a problem is converted to an optimal control problem with one state variable, subject to state and control input constraints along the path. It is shown that the solution to this energy-optimal tracking problem provides a good approximation to the minimum-fuel problem. The switching structure of the optimal solution is analyzed, and a semi-analytical method is proposed for computing the optimal solution. Compared to standard numerical optimization methods, the proposed method is guaranteed to converge to the optimal solution, and is computationally much more efficient. Numerical examples are presented to demonstrate the validity of the proposed method. As verified by these numerical results, the proposed energy-optimal solution can help improve aircraft fuel efficiency during the landing phase.

5.1 Introduction

With climbing fuel cost, it is desirable to improve the fuel efficiency of current aircraft operations subject to aircraft performance and scheduling constraints. Such a problem can be naturally cast as an optimal motion planning problem, which is a common problem encountered in many industrial and transportation systems, including robotic arms[28, 119, 96, 118, 116, 117, 40, 25, 52, 130, 36], ground vehicles[8, 61, 74, 59, 129], aircraft[73, 83], etc. Although optimal motion planning problems can be solved directly using numerical optimization techniques[30, 31, 122, 36, 38, 70], the number of the required computations may grow to impractical levels, especially for real-time applications. Hence, a hybrid approach is commonly adopted in practice, according to which the motion planning task is decomposed into multiple levels[50, 98]. At the higher level, only the geometric aspects of the path are considered, while the lower (path-tracking) level deals with the system dynamics and the state and control constraints, and generates the time-parameterization of the path provided by the higher (geometric) level planner. This chapter focuses on the aircraft path tracking problem at the lower level. Therefore, throughout the chapter, it is assumed that the path to be followed is given by the geometric level path planner.

Given a path, the minimum-time path-tracking problem for robotic manipulators, ground vehicles, and aircraft has been studied in [28, 119, 96, 118, 116, 129, 137]. The optimal solution to these problems can help improve plant productivity[28, 119, 96, 118, 116], racing car performance[129], or achieve faster aircraft landing in case of an emergency[137]. These solutions maximize pointwise the speed along the path, and do not contain any singular arcs [1]. When tracking time is not of primary concern, it is often desirable to minimize the energy or the fuel consumption of the system. Along this direction, the minimum-work train operation problem has been studied in Refs. [8, 61, 74, 59]. Unlike the solution to the minimum-time problem, minimum-work or minimum-energy solutions usually contain singular control arcs, in addition to the bang-bang control arcs. As it is typically the case for problems with singular arcs, it is difficult to determine the optimal sequence in which these singular arcs appear–in combination with the bang-bang arcs–in the optimal solution, as well as the corresponding optimal switching times. Numerical techniques are usually required for solving optimal control problems involving both bang-bang and singular arcs. When the travel time is free, the explicit expression of a singular arc can be solved analytically. In the case of fixed travel time, which is most important for scheduled operations[8, 61, 74, 59], the singular arc(s) cannot be computed directly, and a numerical procedure must be used to compute the singular arc(s) such that the desired travel time and boundary conditions are satisfied.

When using numerical methods to solve singular optimal control problems, an approximate solution is usually obtained at first, using standard numerical optimal control techniques, and then a control switching structure is guessed based on the approximate solution

[1] The “singular arcs” in Ref. [116] actually refer to segments of the speed profile with active speed constraints, which is different from the traditional term used in optimal control [29].

and the analytic expression of the singular control. Finally, the guessed switching structure is applied to solve the singular control problem[131]. These numerical methods are time-consuming, and require extensive knowledge and experience from the part of the user to obtain the actual optimal solution. On the other hand, an analytical optimal control approach (such as in [8, 61, 74, 59]), although less general than purely numerical methods, can provide more accurate information about the singular arcs and switching times in the optimal solution, and thus it is more reliable and efficient.

The path-tracking methods in Refs. [8, 61, 74, 59, 28, 119, 96, 118, 116, 117, 129, 137] share the same key steps of solving a scalar functional optimization problem. For train operations, for instance[8, 61, 74, 59], the point-mass train model has a single degree of freedom along the rail, hence the corresponding path-tracking problem is naturally a speed optimization problem. Similarly, the path-following problem for robotic arms, ground vehicles, and aircraft can also be simplified to a speed optimization problem along a prescribed path[28, 119, 96, 118, 116, 117, 129, 137], which can be solved analytically.

In this chapter, we address the problem of minimum-energy path-tracking for fixed-wing aircraft with fixed time of arrival (TOA). As in Ref. [137], a scalar functional optimization problem is formulated and solved semi-analytically using optimal control theory. Because fuel consumption is closely related to the engine's mechanical work counteracting the effects of air drag and gravity, the issue of fuel efficiency can also be addressed (at least approximately) by solving this minimum-energy problem. Compared to the somewhat similar minimum-work problem for train operations[8, 61, 74, 59], in which the initial and final speed are both zero and only the upper speed limit can be active in the middle of the optimal solution, in the aircraft path-tracking problem considered in this chapter both the initial and final values of the speed are non-zero, and both upper and lower non-zero speed bounds exist, and can be active along the path. Hence, the aircraft minimum-energy solution exhibits a more complicated switching structure than the one in Refs. [8, 61, 74, 59].

The rest of this chapter is organized as follows: We first formulate the aircraft minimum-energy fixed TOA path-tracking problem as an optimal control problem in Section 5.2. Then, in Section 5.3 we provide some new results, along with the corresponding proofs regarding the optimal switching structure of the minimum-energy solution. We also present a formula for computing the energy-optimal solution. A minimum-energy path tracking algorithm is proposed in Section 5.4. The validity of the proposed method is tested using numerical experiments, and the results are presented at the end of the chapter.

5.2 *Aircraft Dynamics and Simplified Problem*

A point-mass model of a fixed-wing aircraft is given by the following equations of motion:

$$\dot{x} = v\cos\gamma\cos\psi, \tag{131}$$

$$\dot{y} = v\cos\gamma\sin\psi, \tag{132}$$

$$\dot{z} = v\sin\gamma, \tag{133}$$

$$\dot{v} = \frac{1}{m}\left[T - F_D(C_L, v, z) - mg\sin\gamma\right], \tag{134}$$

$$\dot{\gamma} = \frac{1}{mv}\left[F_L(C_L, v, z)\cos\phi - mg\cos\gamma\right], \tag{135}$$

$$\dot{\psi} = -\frac{F_L(C_L, v, z)\sin\phi}{mv\cos\gamma}, \tag{136}$$

where x and y denote the position of the aircraft in the horizontal plane, z is the altitude, v is the aircraft speed, γ is the flight path angle, ψ is the heading angle, and ϕ is the aircraft bank angle. The aerodynamic lift force $F_L(C_L, v, z)$ and the drag force $F_D(C_L, v, z)$ are given by:

$$\begin{aligned} F_L\left(C_L, v, z\right) &= \frac{1}{2}\rho(z)v^2 S C_L, \\ F_D\left(C_L, v, z\right) &= \frac{1}{2}\rho(z)v^2 S C_D = \frac{1}{2}\rho(z)v^2 S(C_{D_0} + KC_L^2), \end{aligned}$$

where $\rho(z)$ is the air density given as a function of z, C_{D_0} and K are constants describing the aerodynamic properties of the aircraft, and S is the main wing surface area. The control inputs in this model are the lift coefficient C_L, the bank angle ϕ, and the thrust T. It is required that the aircraft speed satisfies the bounds $v(s) \in [v_{\min}(z), v_{\max}(z)]$, where $v_{\min}(z)$ and $v_{\max}(z)$ are altitude-dependent minimum and maximum speeds, respectively, and

$$C_L \in [C_{L_{\min}}, C_{L_{\max}}], \quad \phi \in [\phi_{\min}, \phi_{\max}], \quad T \in [T_{\min}, T_{\max}], \tag{137}$$

where $C_{L_{\min}}$, $C_{L_{\max}}$, $\phi_{\min}$, $\phi_{\max}$, $T_{\min}$ and $T_{\max}$ are (possibly, path-dependent) bounds on the associated control inputs. It is assumed that $C_{L_{\min}} \leq 0 \leq C_{L_{\max}}$, $-\pi/2 < \phi_{\min} < 0 <$

$\phi_{\max} < \pi/2$, $0 \leq T_{\min} < T_{\max}$, and $\gamma \in (-\pi/2, \pi/2)$. These conditions are generic for a civil fixed-wing aircraft in normal/maneuverable flight.

Let now $(x(s), y(s), z(s))$ denote a three-dimensional geometric path, parameterized by its natural path length coordinate $s \in [s_0, s_f] \subset \mathbb{R}_+$. The main objective of this chapter is to find a time-parameterization of the path, or equivalently, a function $s(t)$ with $s(0) = s_0$ and $s(t_f) = s_f$, where $t \in [0, t_f]$, and t_f is the desired TOA, such that the corresponding time-parameterized trajectory $\big(x(s(t)), y(s(t)), z(s(t))\big)$ minimizes the total energy, or mechanical work, while flying along the path, and without violating any state or control constraints. Because the path coordinate s is related to the speed v as follows

$$s(t) = \int_{t_0}^{t} v(\tau)\,\mathrm{d}\tau,$$

the key step for solving this problem is the optimization of the speed profile $v(s)$ along the path. For convenience of notation, let $E \triangleq v^2/2$ denote the specific kinetic energy per unit mass of the aircraft. It has been shown in Ref. [137] that the lift coefficient, the bank angle, and the speed constraints can be reduced to lower and upper bounds on the specific kinetic energy E as follows:

$$E(s) - \overline{g}_w(s) \leq 0, \tag{138}$$

$$\underline{g}_w(s) - E(s) \leq 0, \tag{139}$$

for all $s \in [s_0, s_f]$, where $\overline{g}_w(s)$ and $\underline{g}_w(s)$ are path-dependant bounds on the specific kinetic energy, which are determined from the path geometry, and the constraints on the speed, the bank angle and the lift coefficient. The derivative of E satisfies the following ordinary differential equation[137]:

$$E'(s) = \frac{T(s)}{m} + c_1(s)E(s) + \frac{c_2(s)}{E(s)} + c_3(s), \tag{140}$$

where the prime denotes the derivative with respect to s, and

$$c_1(s) \triangleq -\frac{C_{D_0}(s)\rho(s)S}{m} - \frac{4Km\gamma'^2(s)}{\rho(s)S} - \frac{2Km\cos^2\gamma(s)\psi'^2(s)}{\rho(s)S}, \tag{141}$$

$$c_2(s) \triangleq -\frac{Kmg^2\cos^2\gamma(s)}{\rho(s)S}, \tag{142}$$

$$c_3(s) \triangleq -\frac{4Km\gamma'(s)g\cos\gamma(s)}{\rho(s)S} - g\sin\gamma(s). \tag{143}$$

Once the optimal specific kinetic energy $E^*(s)$ is obtained, the optimal thrust profile $T^*(s)$ along the path can be determined using equation (140). Subsequently, the other optimal control inputs can also be computed using inverse dynamics as follows:

$$\begin{aligned} C_L^*(s) &= \frac{1}{\rho E^*(s) S}\Big(T^*(s) - mE^{*\prime}(s) - mg\sin\gamma(s)\Big), && (144)\\ \phi^*(s) &= -\arctan\left(\frac{2\cos\gamma(s)\psi'(s)}{2\gamma'(s) + g\cos\gamma(s)/E^*(s)}\right). && (145)\end{aligned}$$

5.3 *Energy-Optimal Path Tracking with Fixed Time of Arrival*

In this section, we first introduce a formulation of the energy-optimal aircraft path-tracking problem with fixed TOA. This problem provides an approximate solution to the minimum-fuel problem. We then present a semi-analytic solution to the energy-optimal path-tracking problem.

5.3.1 Fuel-Optimal and Energy-Optimal Problem Formulation

ost modern civil airliners are powered by high-bypass turbofan engines for better fuel economy. The fuel consumption rate for this type of engine is given by[87]

$$\dot{f} = -\eta T, \tag{146}$$

where f is the fuel weight, η is the *installed thrust specific fuel consumption*, which varies with airspeed, altitude, type of engine, and throttle conditions, and it is given by

$$\eta = (a + bM_a)\sqrt{\eta_0/(1 + cM_a^2)}, \tag{147}$$

where M_a is the Mach number and a, b, c are constants depending on the engine type. In (147), $\eta_0 = \eta_0(z, M_a)$ varies with altitude and Mach number and can be determined from look-up data tables[87]. The fuel consumption models for other types of jet engines are similar to equations (146) and (147), but with different parameters.

With the above model, the fuel consumption during the landing phase can be estimated by

$$J_f = \int_{t_0}^{t_f} -\dot{f}(t)\,\mathrm{d}t = \int_{t_0}^{t_f} \eta(t)T(t)\,\mathrm{d}t. \tag{148}$$

From (148) it is clear that the minimum-fuel problem is equivalent to the minimization of the weighted thrust history, where the weight $\eta(t)$ is given in (147). The solution to this problem requires the use of purely numerical techniques. To avoid this difficulty, here we will minimize, instead, the total energy (mechanical work) required to fly along the path, which is given by

$$J_w = \int_{t_0}^{t_f} v(t)T(t)\,\mathrm{d}t = \int_{s_0}^{s_f} T(s)\,\mathrm{d}s. \tag{149}$$

As demonstrated in Ref [33], the optimal speed profile of the minimum-fuel optimization problem contains singular arcs on which most of the fuel-saving is achieved. It was observed in our numerical studies that the air speed changes slowly along these singular arcs, in which case the singular arcs of the fuel-optimal problem can be approximated by those of the energy-optimal problem. As a result, the minimization of the energy cost function (149) is expected to provide a reasonably good approximation to the fuel optimization problem (148). This is verified by numerical results in Section 5.5. Henceforth, we focus on minimizing the energy for the landing path-tracking problem.

During the landing process, the change of mass due to fuel consumption is usually negligible when compared to the total mass of the aircraft. Hence, we may neglect the effect of mass change on the specific kinetic energy dynamics (140), and assume that m is constant during the landing phase. The validity of such an assumption is justified in Ref. [33], which reported that the mass change has little influence on the fuel-optimal trajectory during the climb and descent phases. It needs to be noted however that this assumption would be invalid during the long cruise phase [49].

To account for the fixed final time, the flight time t is treated as a state variable in an augmented system with the additional differential equation

$$t'(s) = \frac{1}{\sqrt{2E(s)}}.$$

With the above assumptions, the minimum-energy aircraft path-tracking problem with fixed TOA can be formulated as an optimal control problem involving two differential equations, two algebraic constraints, two boundary conditions, and two control constraints, as follows:

Problem 5.3.1 (Minimum-energy path-tracking problem with fixed TOA). Consider the following optimal control problem in Lagrange form:

$$\begin{aligned}
\min_{T} \quad & \int_{s_0}^{s_f} T(s)\,\mathrm{d}s, && (150)\\
\text{subject to} \quad & E'(s) = \frac{T(s)}{m} + c_1(s)E(s) + \frac{c_2(s)}{E(s)} + c_3(s), && (151)\\
& t'(s) = \frac{1}{\sqrt{2E(s)}}, && (152)\\
& E(s) - \overline{g}_w(s) \le 0, && (153)\\
& \underline{g}_w(s) - E(s) \le 0, && (154)\\
& E(s_0) = v_0^2/2, && (155)\\
& E(s_f) = v_f^2/2, && (156)\\
& T_{\min}(s) \le T(s) \le T_{\max}(s), && (157)\\
& t(s_f) = t_f. && (158)
\end{aligned}$$

To solve this problem, we apply the necessary conditions for optimality to screen the allowable thrust profile candidates. This is done next.

5.3.2 Optimality Conditions

First, consider the case when the state constraints (153) and (154) are not active. The Hamiltonian for Problem 5.3.1 is given by

$$\begin{aligned}
H &= T + \lambda_E\left(\frac{T}{m} + c_1E + \frac{c_2}{E} + c_3\right) + \frac{\lambda_t}{\sqrt{2E(s)}}\\
&= \left(1 + \frac{\lambda_E}{m}\right)T + \lambda_E\left(c_1E + \frac{c_2}{E} + c_3\right) + \frac{\lambda_t}{\sqrt{2E(s)}},
\end{aligned}$$

where λ_E and λ_t are the costates corresponding to the dynamics for E and t, respectively. The costate dynamics are given by:

$$\lambda_E' = -\frac{\partial H}{\partial E} = -c_1\lambda_E + c_2E^{-2}\lambda_E + \frac{1}{2\sqrt{2}}E^{-3/2}\lambda_t, \tag{159}$$

$$\lambda_t' = -\frac{\partial H}{\partial t} = 0. \tag{160}$$

Therefore, the costate λ_t is constant. The switching function is given by

$$\frac{\partial H}{\partial T} = 1 + \frac{\lambda_E}{m}. \tag{161}$$

By Pontryagin's Maximum Principal (PMP), the extremal control is given by

$$T = \begin{cases} T_{\max}, & 1 + \lambda_E/m < 0, \\ \tilde{T}, & 1 + \lambda_E/m = 0, \\ T_{\min}, & 1 + \lambda_E/m > 0, \end{cases} \tag{162}$$

where $\tilde{T}$ is the singular control. On singular arcs, the switching function (161) is identically zero. Hence, the derivative of the switching function must also vanish on singular arcs, which yields

$$\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\partial H}{\partial T}\right) = \frac{\lambda_E'}{m} = \frac{1}{m}\left(-c_1\lambda_E + c_2E^{-2}\lambda_E + \frac{1}{2\sqrt{2}}E^{-3/2}\lambda_t\right) = c_1 - c_2E^{-2} + \frac{1}{2\sqrt{2}m}E^{-3/2}\lambda_t \equiv 0, \tag{163}$$

from which the singular specific kinetic energy profile can be computed. For notational convenience, equation (163) is rewritten as

$$P(E(s), s) = \lambda_t, \tag{164}$$

where

$$P(E(s), s) = -2\sqrt{2}m\left(c_1(s)E^{3/2}(s) - c_2(s)E^{-1/2}(s)\right). \tag{165}$$

Let $E^*(s)$ be the optimal specific kinetic energy profile for Problem 5.3.1 with the corresponding optimal costate value λ_t^*, and suppose that $E^*(s)$ contains a singular arc on a subinterval $[s_a, s_b] \subseteq [s_0, s_f]$. Because the switching function vanishes on singular arcs, we must have $P(E^*(s), s) = \lambda_t^*$ for all $s \in [s_a, s_b]$.

Henceforth, we assume that the optimal solution to Problem 5.3.1 is unique, and we will focus on the energy-optimal path-tracking problem.

Proposition 5.3.1. Let $E^*(s)$ be the optimal specific kinetic energy profile for Problem 5.3.1 with corresponding optimal costate value λ_t^*. Let the function $\tilde{E} : [s_0, s_f] \to \mathbb{R}_+$ be defined via the equation $P(\tilde{E}(s), s) = \lambda_t^*$ for all $s \in [s_0, s_f]$. Then, for all $s \in [s_0, s_f]$, we have that $P(E^*(s), s) > \lambda_t^*$ if and only if $E^*(s) > \tilde{E}(s)$, and $P(E^*(s), s) < \lambda_t^*$ if and only if $E^*(s) < \tilde{E}(s)$.

Proof. Note that

$$\begin{aligned}P\left(E^*(s),s\right)-\lambda_t^* &= 2\sqrt{2}m\left(-c_1(s)E^{*3/2}(s)+c_2(s)E^{*-1/2}(s)\right)-2\sqrt{2}m\left(-c_1(s)\tilde{E}^{3/2}(s)+c_2(s)\tilde{E}^{-1/2}(s)\right)\\ &= 2\sqrt{2}m\left(-c_1(s)\left(E^{*3/2}(s)-\tilde{E}^{3/2}(s)\right)+c_2(s)\left(E^{*-1/2}(s)-\tilde{E}^{-1/2}(s)\right)\right).\end{aligned}$$

Since $c_1(s)<0$ and $c_2(s)<0$ for all $s\in[s_0,s_f]$, according to (141) and (142), the claim of this proposition can be easily verified based on the monotonicity of the power functions appearing in the right hand side of the above expression. □

Remark 5.3.1. It is clear that if $E^*(s)$ contains a singular arc on $[s_a,s_b]\subseteq[s_0,s_f]$, then the function $\tilde{E}(s)$ defined in Proposition 5.3.1 satisfies $\tilde{E}(s)=E^*(s)$ for all $s\in[s_a,s_b]$.

With $E^*(s)$, λ_t^* and $\tilde{E}(s)$ as in Proposition 5.3.1, the singular control $\tilde{T}$ can be obtained by taking the derivative of equation $P(\tilde{E}(s),s)=\lambda_t^*$, and replacing $\tilde{E}'(s)$ with the right hand side of equation (151), that is,

$$\tilde{T}(s)=\frac{2m\left(c_2'(s)\tilde{E}(s)-c_1'(s)\tilde{E}^3(s)\right)}{3c_1(s)\tilde{E}^2(s)+c_2(s)}-c_1(s)\tilde{E}(s)m-\frac{c_2(s)m}{\tilde{E}(s)}-c_3(s)m. \tag{166}$$

Suppose there exists $(s_a,s_b)\subseteq[s_0,s_f]$ such that $E^*(s)=\tilde{E}(s)$ but $\tilde{T}(s)>T_{\max}$ or $\tilde{T}(s)<T_{\min}$. It follows that the corresponding optimal thrust profile cannot contain any singular thrust subarc. Therefore, in the sequel we will assume that $\tilde{T}(s)\in[T_{\min},T_{\max}]$ for all $s\in(s_a,s_b)$. This assumption is valid as long as the aircraft is in a normal flight condition, and the path is smooth enough, in the sense that the path angle and the heading angle change slowly along the path.

According to the PMP, when the state constraints (138) and (139) are not active, the optimal control is composed of extremals $T_{\max}$, $T_{\min}$ and $\tilde{T}$. The singular specific kinetic energy $\tilde{E}$ and the corresponding thrust profile $\tilde{T}$ are not readily known since they depend on the unknown parameter λ_t^*, which further depends on the final time t_f. Furthermore, although there is only a finite number of extremal controls, the possible combinations of the resulting extremals can be large. Hence, it is necessary to identify the switching structure for the different extremals along with the associated switching times in order to obtain the optimal solution.

5.3.3 Optimality of the Singular Arcs

An admissible singular control $\tilde{T}(s)$, in addition to the constraint $T_{\min} \leq \tilde{T}(s) \leq T_{\max}$, must satisfy the generalized Legendre-Clebsch condition[29]

$$\frac{\partial}{\partial T}\left[\frac{\mathrm{d}^2}{\mathrm{d}s^2}\left(\frac{\partial H}{\partial T}\right)\right] \leq 0. \tag{167}$$

if it is to be part of the optimal trajectory. Differentiating (163) with respect to s, one obtains

$$\frac{\mathrm{d}^2}{\mathrm{d}s^2}\left(\frac{\partial H}{\partial T}\right) = c_1'(s) - c_2'(s)\tilde{E}^{-2}(s) + 2c_2(s)\tilde{E}^{-3}(s)\tilde{E}'(s) - \frac{3}{4\sqrt{2}m}\tilde{E}^{-\frac{5}{2}}(s)\lambda_t^*\tilde{E}'(s).$$

Using (151), it follows that

$$\frac{\partial}{\partial T}\left[\frac{\mathrm{d}^2}{\mathrm{d}s^2}\left(\frac{\partial H}{\partial T}\right)\right] = \frac{1}{m}\left(2c_2(s)\tilde{E}^{-3}(s) - \frac{3}{4\sqrt{2}m}\tilde{E}^{-\frac{5}{2}}(s)\lambda_t^*\right). \tag{168}$$

Since $\tilde{E}(s)$ satisfies (163), it follows that

$$c_1(s) - c_2(s)\tilde{E}^{-2}(s) + \frac{1}{2\sqrt{2}m}\tilde{E}^{-3/2}(s)\lambda_t^* = 0, \tag{169}$$

By eliminating λ_t^* from (168), and by using equation (163), equation (168) can be written as:

$$\frac{\partial}{\partial T}\left[\frac{\mathrm{d}^2}{\mathrm{d}s^2}\left(\frac{\partial H}{\partial T}\right)\right] = \frac{1}{mE(s)}\left(2c_2(s)\tilde{E}^{-2}(s) - \frac{3}{4\sqrt{2}m}\tilde{E}^{-\frac{3}{2}}(s)\lambda_t^*\right) \tag{170}$$

$$= \frac{1}{2mE(s)}\left(3c_1(s) + c_2(s)\tilde{E}^{-2}(s)\right), \tag{171}$$

which is indeed negative because $c_1(s) < 0$ and $c_2(s) < 0$ according to (141) and (142). Hence, along the singular arcs, the generalized Legendre-Clebsch condition is satisfied, and hence these arcs can be part of the optimal trajectory.

5.3.4 Optimal Switching Structure Involving Singular Arcs

When solving an optimal control problem with singular arcs, and since the optimal switching structure is not known in advance, it is a common practice to assume initially a certain fixed switching structure according to which the switching times are computed. This approach, although straightforward, may lead to a suboptimal solution. The switching structure of the optimal solution to Problem 5.3.1 can be uniquely determined owing to the special properties of this problem. The following theorem is key regarding the switching structure of the solution of Problem 5.3.1.

Theorem 5.3.1. Let $E^*(s)$ be the optimal specific kinetic energy profile for Problem 5.3.1 with the optimal costate value λ_t^*, and let $\tilde{E} : [s_0, s_f] \to \mathbb{R}_+$ be the function defined by $P(\tilde{E}(s), s) = \lambda_t^*$. Consider a subinterval $(s_a, s_b) \subset [s_0, s_f]$ such that $\underline{g}_w(s) < E^*(s) < \overline{g}_w(s)$ for all $s \in (s_a, s_b)$. If $E^*(s) < \tilde{E}(s)$ (respectively, $E^*(s) > \tilde{E}(s)$) for all $s \in (s_a, s_b) \subset [s_0, s_f]$, then the corresponding optimal control $T^*(s)$ does not contain any switching from $T_{\min}$ to $T_{\max}$ (respectively, $T_{\max}$ to $T_{\min}$) on (s_a, s_b).

Proof. Assume that $E^*(s) < \tilde{E}(s)$ for all $s \in (s_a, s_b)$, and let $T^*(s) = T_{\min}$ on (s_a, τ) , and assume, ad absurdum, that $T^*(s) = T_{\max}$ on (τ, s_b), where $\tau \in (s_a, s_b)$ is the switching point from $T_{\min}$ to $T_{\max}$. Because the state constraints are not saturated on (s_a, s_b), the optimal costate λ_E^* is continuous on (s_a, s_b). Since $T^*(s) = T_{\min}$ on (s_a, τ) , and $T^*(s) = T_{\max}$ on (τ, s_b), we have $1 + \lambda_E^*(s)/m > 0$ on (s_a, τ) and $1 + \lambda_E^*(s)/m < 0$ on (τ, s_b) following (162). By the continuity of $\lambda_E^*(s)$, it follows that $\lambda_E^*(\tau) = -m$.

According to equation (159), the derivative of the costate at τ is given by

$$\begin{aligned}\lambda_E^{*\prime}(\tau) &= -\frac{1}{2\sqrt{2}}(E^*)^{-3/2}(\tau)\left(\left[-2\sqrt{2}m\left(c_1(E^*)^{3/2}(\tau) - c_2(E^*)^{-1/2}(\tau)\right)\right] - \lambda_t^*\right)\\ &= -\frac{1}{2\sqrt{2}}(E^*)^{-3/2}(\tau)\left(P(E^*(\tau),\tau) - \lambda_t^*\right) > 0,\end{aligned}$$

where the last inequality holds because $P(E^*(\tau), \tau) < \lambda_t^*$ when $E^*(\tau) < \tilde{E}(\tau)$, following Proposition 5.3.1. Because $\lambda_E^{*\prime}$ is continuous following the continuity of E^*, there exists $\epsilon > 0$ such that $\lambda_E^{*\prime}(s) > 0$ for all $s \in (\tau, \tau + \epsilon) \subseteq (\tau, s_b)$. It follows that, since $\lambda_E^*(\tau) = -m$, we have $1 + \lambda_E(s)/m > 0$ for all $s \in (\tau, \epsilon)$, a contradiction. Therefore, if $E^*(s) < \tilde{E}(s)$ on (s_a, s_b), the optimal thrust contains no switch from $T_{\min}$ to $T_{\max}$ on (s_a, s_b). The proof for the case $E^*(s) > \tilde{E}(s)$ is similar, hence, is omitted. □

Theorem 5.3.1 narrows down the possible switching combinations of the optimal control T^* for Problem 5.3.1. The valid switching structures above and below $\tilde{E}$ are illustrated in Fig. 41. In contrast, the switching structures in Fig. 42 are not optimal.

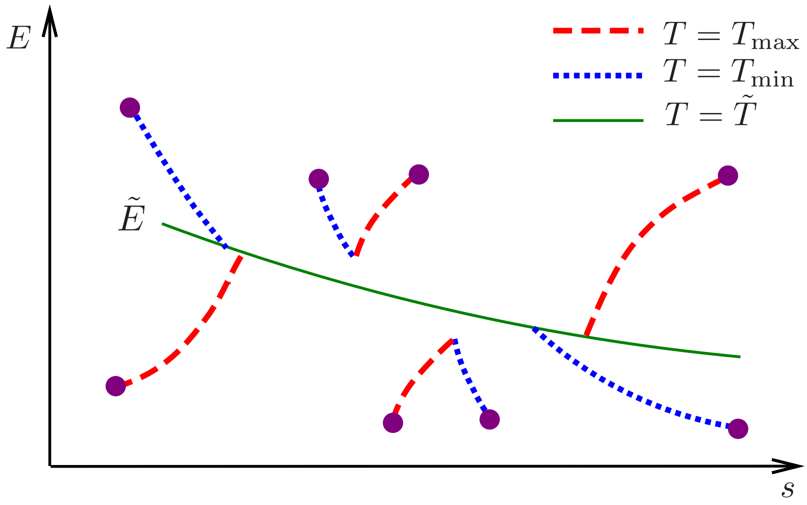

Figure 41: Optimal switching structures

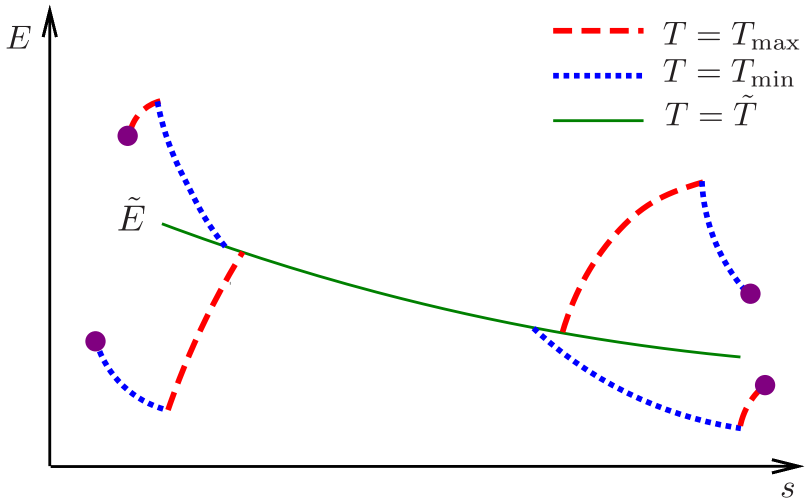

Figure 42: Non-optimal switching structures

Given the optimal costate value λ_t^*, $\tilde{E}(s)$ can be computed using the expression $P(\tilde{E}(s), s) = \lambda_t^*$ for all $s \in [s_0, s_f]$. If the optimal specific kinetic energy E^* contains a singular arc on a subinterval, then it must be true that $E^* = \tilde{E}$ on this subinterval. Hence, the optimal specific kinetic energy E^* can be obtained by first identifying the segments of $\tilde{E}$, and then choosing the optimal structure and the corresponding switching times.

5.3.5 Optimal Switching Structure Involving State-Constrained Arcs

The previous analysis is valid when the state constraints (138) and (139) are inactive. In this section we analyze the case when the state either the constraints (138) or (139) are active on part of the optimal trajectory. When the state constraint (138) or (139) is active along a certain part of the optimal specific kinetic energy solution E^*, we call this part of E^* a *state constrained arc*. The corresponding control is referred to as a *state constrained control*. If the upper state constraint is saturated, then $T^* = \overline{T}_w$, which is the control corresponding to $E^* = \overline{g}_w$. Similarly, if the lower state constraint is saturated, then $T^* = \underline{T}_w$, which is the control corresponding to $E^* = \underline{g}_w$. Clearly, it is required that $\underline{T}_w, \overline{T}_w \in [T_{\min}, T_{\max}]$ on the corresponding domain for feasibility. For an arbitrary geometric path, the optimal control T^* for the minimum-energy path-following problem is composed of bang-bang control $T_{\min}$ and $T_{\max}$, singular control $\tilde{T}$, and state constrained control $\overline{T}_w$ and $\underline{T}_w$.

The minimum-time path-following problem has been solved in Ref. [137]. This method can be modified to provide the maximum flight time along a given geometric path. The maximum flight time scheme corresponds to the point-wise minimization of the specific kinetic energy along the path. This is the opposite of the minimum-time problem, which seeks to maximize pointwise the specific kinetic energy along the path. Note that, for any given path, an upper bound of the flight time exists because the speed of a fixed-wing aircraft must be higher than a certain value to avoid stall.

Lemma 5.3.1. Let $E_U^*(s)$ be the minimum-time path-following specific kinetic energy profile with flight time $t_{\min}$, and let $E_L^*(s)$ be the maximum-time path-following specific kinetic energy profile with flight time $t_{\max}$. Let $E^*(s)$ be the optimal specific kinetic energy profile for the minimum-energy path-following problem with fixed flight time t_f. Then the following inequalities hold

$$
\begin{aligned}
&t_{\min} \leq t_f \leq t_{\max}, \\
&E_L^*(s) \leq E^*(s) \leq E_U^*(s), \quad s \in [s_0, s_f].
\end{aligned}
$$

Proof. The inequalities involving $t_{\min}$ and $t_{\max}$ are obvious. To show the other inequalities, suppose, without loss of generality, that $E^*(s_a) > E_U^*(s_a)$ for some $s_a \in [s_0, s_f]$. Since both E^* and E_U^* are feasible specific kinetic energy profiles, $\bar{E} = \max\{E^*, E_U^*\}$ is also a feasible specific kinetic energy profile, i.e., $\bar{E}(s)$ satisfies the boundary conditions, and can be tracked with the available control inputs. Then $\bar{E} \geq E_U^*$ on $[s_0, s_f]$, and $\bar{E}(s) > E_U^*(s)$ on at least one interval containing s_a following the continuity of E^*. Hence, for $\bar{E}$ the total flight time would be smaller than $t_{\min}$, which is a contradiction since $t_{\min}$ is the minimum-time solution. The inequality $E_L^*(s) \leq E^*(s)$ can be proved similarly. □

According to Lemma 5.3.1, the fixed-time energy-optimal specific kinetic energy E^* is bounded by the minimum-time solution E_U^* and the maximum-time solution E_L^*. Furthermore, based on Theorem 5.3.1, it can be shown that $E^*(s) = E_U^*(s)$ or $E^*(s) = E_L^*(s)$ on certain subintervals. This property of E^* is characterized by the following Lemma.

Lemma 5.3.2. Let $E^*(s)$ be the optimal specific kinetic energy solution to Problem 5.3.1. and let $\tilde{E}$ be defined on $[s_0, s_f]$ by $P(\tilde{E}(s), s) = \lambda_t^*$ where λ_t^* is the corresponding optimal costate value. Let $E_U^*(s)$ and $E_L^*(s)$ be the optimal specific kinetic energy solutions to the minimum-time and maximum-time path-tracking problems, respectively. Let $\Gamma_U = \{s | E_U^*(s) < \tilde{E}(s),\ s \in [s_0, s_f]\}$, and $\Gamma_L = \{s | E_L^*(s) > \tilde{E}(s),\ s \in [s_0, s_f]\}$. Suppose that $E^*(s) > \underline{g}_w(s)$ for all $s \in [s_0, s_f] \setminus \Gamma_L$, and $E^*(s) < \overline{g}_w(s)$ for all $s \in [s_0, s_f] \setminus \Gamma_U$, then $E^*(s) = E_U^*(s)$ for all $s \in \Gamma_U$, and $E^*(s) = E_L^*(s)$ for all $s \in \Gamma_L$.

Proof. We first show that $E^*(s) = E_U^*(s)$ for all $s \in \Gamma_U$. Let T_U^* and T^* be the thrust control associated with E_U^* and E^*, respectively. From Lemma 5.3.1, we have that $E^*(s) \leq E_U^*(s)$ for all $s \in [s_0, s_f]$. Assume, ad absurdum, that there exists $\tau \in \Gamma_U$ such that $E^*(\tau) < E_U^*(\tau)$. Then by the definition of Γ_U, we also have $E^*(\tau) < \tilde{E}(\tau)$.

Let $q = \inf\{s | E^*(s) = E_U^*(s), s \in [\tau, s_f]\}$. Since $E^*(s_f) = E_U^*(s_f)$, q is well-defined. Similarly, let $p = \sup\{s | E^*(s) = E_U^*(s), s \in [s_0, \tau]\}$. We have $E^*(s) < E_U^*(s)$ for all

$s \in (p, q)$ by the fact $E^*(\tau) < E_U^*(\tau)$, the definitions of p, q, and the continuity of E^* and E_U^*. Since $E^*(s) < E_U^*(s) \leq \overline{g}_w(s)$ for all $s \in (p, q)$, the state constraint $E(s) \leq \overline{g}_w(s)$ is inactive along E^* for $s \in (p, q)$, hence, $T^*(s)$ can only take the values of $T_{\max}$, $T_{\min}$, $\tilde{T}(s)$, or $\underline{T}_w(s)$ on (p, q). Because $E^*(\tau) < \tilde{E}(\tau)$, we have $E^*(\tau) > \underline{g}_w(\tau)$, and it follows that either $T^*(\tau) = T_{\max}$ or $T^*(\tau) = T_{\min}$.

First, consider the case $T^*(\tau) = T_{\min}$. Then we claim that $E^*(s) < \tilde{E}(s)$ for all $s \in (\tau, q)$. To see this, assume that $E^*(s) \geq \tilde{E}(s)$ for some $s \in (\tau, q)$, then it follows from the fact $E^*(\tau) < \tilde{E}(\tau)$ and the continuity of E^* and $\tilde{E}$ that the equation $E^*(\gamma) = \tilde{E}(\gamma)$ has at least one solution on (τ, q). Let $\gamma = \inf\{s | E^*(s) = \tilde{E}(s), s \in (\tau, q)\}$, it follows that $E^*(\gamma) = \tilde{E}(\gamma)$, and $E^*(s) < \tilde{E}(s)$ for all $s \in (\tau, \gamma)$. Therefore, $(\tau, \gamma) \subseteq [s_0, s_f] \setminus \Gamma_L$, and we have $E^*(s) > \underline{g}_w(s)$ for all $s \in (\tau, \gamma)$. It follows that on (τ, γ), $T^*(s)$ can only take the values of $T_{\min}$ and $T_{\max}$. Because $E^*(s) < \tilde{E}(s)$ for all $s \in (\tau, \gamma)$, $T^*(s)$ can not switch from $T_{\min}$ to $T_{\max}$ according to Theorem 5.3.1, and we have $T^*(s) = T_{\min}$ for all $s \in (\tau, \gamma)$. With $T^*(s) = T_{\min} \leq \tilde{T}(s)$ for $s \in (\tau, \gamma)$, and the initial conditions satisfying $E^*(\tau) < \tilde{E}(\tau)$, it follows from forward integrations of $E^{*\prime}$ and $\tilde{E}'$ from τ to γ that $E^*(\gamma) < \tilde{E}(\gamma)$, which is a contradiction. Hence, the claim is true, i.e., $E^*(s) < \tilde{E}(s)$ for all $s \in (\tau, q)$, and it follows that $T^*(s) = T_{\min}$ for all $s \in (\tau, q)$ according to Theorem 5.3.1. Then, with the initial conditions $E^*(q) = E_U^*(q)$ and $T_U^*(s) \geq T_{\min} = T^*(s)$ for all $s \in (\tau, q)$, backward integrations of $E^{*\prime}$ and $E_U^{*}{}'$ from q to τ lead to $E^*(\tau) \geq E_U^*(\tau)$, which is a contradiction to the assumption $E^*(\tau) < E_U^*(\tau)$.

Similarly, if $T^*(\tau) = T_{\max}$, we can first prove that $E^*(s) < \tilde{E}(s)$ for all $s \in (p, \tau)$. Specifically, suppose this is not true, then $E^*(\gamma) = \tilde{E}(\gamma)$ has at least one solution on (p, τ). By defining $\gamma = \sup\{s | E^*(s) = \tilde{E}(s), s \in [p, \tau)\}$, then the backward integrations of $E^{*\prime}$ and $\tilde{E}'$ lead to $E^*(\gamma) < \tilde{E}(\gamma)$, which is a contradiction. Then it follows that $E^*(\tau) < E_U^*(\tau)$ is not possible, as in the proof for the case with $T^*(\tau) = T_{\min}$.

Hence, there does not exists any $s \in \Gamma_U$ such that $E^*(s) < E_U^*(s)$, and we have $E^*(s) = E_U^*(s)$ on Γ_U.

The proof for the other statement, $E^*(s) = E_L^*(s)$ for all $s \in \Gamma_L$, is similar, hence, is omitted. $\square$

Since the unconstrained solution to an optimal control problem has the same, or better optimality characteristics than a constrained one, a constraint is, in general, not active unless it is violated by the optimal solution of the unconstrained problem [2]. This property is described by the lemma below.

Lemma 5.3.3. Consider the following two optimal control problems

Problem A

$$\begin{aligned} \min_{u} \quad & J(x,u) \\ \text{s.t.} \quad & \dot{x}(t) = f(x(t),u(t)), \\ & g_1(x(t),u(t)) \leqslant 0, \\ & g_2(x(t),u(t)) \leqslant 0, \quad t \in [t_0,t_f], \\ & x(t_0) = x_0,\ x(t_f) = x_f. \end{aligned}$$

Problem B

$$\begin{aligned} \min_{u} \quad & J(x,u) \\ \text{s.t.} \quad & \dot{x}(t) = f(x(t),u(t)), \\ & g_1(x(t),u(t)) \leqslant 0, \quad t \in [t_0,t_f]. \\ & x(t_0) = x_0,\ x(t_f) = x_f. \end{aligned}$$

Let x_A^* be the optimal solution and u_A^* be the corresponding optimal control to Problem A, and let x_B^* and u_B^* be the optimal solution and corresponding optimal control to Problem B. Then the following statements are true:

1. If $g_2(x_B^*(t),u_B^*(t)) \le 0$ for all $t \in [t_0,t_f]$, then $J(x_B^*,u_B^*) = J(x_A^*,u_A^*)$. Furthermore, if either Problem A or Problem B has a unique solution, then $x_A^* = x_B^*$ and $u_A^* = u_B^*$.

2. If Problem B has a unique solution and $g_2(x_B^*(t),u_B^*(t)) > 0$ for some $t \in [t_0,t_f]$, then $J(x_A^*,u_A^*) > J(x_B^*,u_B^*)$.

Proof. We start with the first statement. Since (x_A^*,u_A^*) is the optimal solution to Problem A, and (x_B^*,u_B^*) is a feasible solution to Problem A, we have $J(x_A^*,u_A^*) \le J(x_B^*,u_B^*)$ by the optimality of (x_A^*,u_A^*). On the other hand, (x_A^*,u_A^*) satisfies all constraints in Problem B, so (x_A^*,u_A^*) is a feasible solution to Problem B. Consequently, $J(x_A^*,u_A^*) \ge J(x_B^*,u_B^*)$ by the optimality of (x_B^*,u_B^*) for Problem B. Therefore $J(x_B^*,u_B^*) = J(x_A^*,u_A^*)$. It follows that $x_A^* = x_B^*$ and $u_A^* = u_B^*$, otherwise both Problem A and Problem B have non-unique solutions.

[2] The only exception would be the case when along the unconstrained optimal solution certain constraints are active but not violated, which is considered to be a trivial case.

We now prove the second statement. As in the previous proof, since (x_A^*, u_A^*) is a feasible solution to Problem B, we have $J(x_A^*, u_A^*) \geq J(x_B^*, u_B^*)$ by the optimality of (x_B^*, u_B^*) for Problem B. Since $g_2(x_B^*(t), u_B^*(t)) > 0$ for some $t \in [t_0, t_f]$, and $g_2(x_A^*(t), u_A^*(t)) \leq 0$ for all $t \in [t_0, t_f]$, it follows that (x_B^*, u_B^*) and (x_A^*, u_A^*) are not identical. By the uniqueness of (x_B^*, u_B^*), it follows that $J(x_A^*, u_A^*) > J(x_B^*, u_B^*)$. □

In the following, Lemma 5.3.3 is used to characterize the state constrained arcs in the optimal specific kinetic energy profile $E^*(s)$. Specifically, given the state constraints, we can first compute the optimal solution of a certain relaxed problem to identify the state constrained arcs. Before introducing the relaxed problem, we need some additional notation. For any subset $\Gamma_U \subseteq [s_0, s_f]$, define

$$\overline{g}_{\Gamma_U}(s) = \begin{cases} \overline{g}_w(s), & s \in \Gamma_U, \\ M, & s \in [s_0, s_f] \setminus \Gamma_U, \end{cases}$$

where $M > 0$ is a number large enough such that $E(s) < M$ is always satisfied on $[s_0, s_f]$ by any feasible specific kinetic energy profile $E(s)$. By choosing a subset Γ_U of interest and enforcing the state constraint $E(s) \leq \overline{g}_{\Gamma_U}(s)$ for all $s \in [s_0, s_f]$, we can ensure that the optimal solution E^* satisfies $E^*(s) \leq \overline{g}_w(s)$ on Γ_U, while remaining unconstrained on $[s_0, s_f] \setminus \Gamma_U$. Similarly, we also define

$$\underline{g}_{\Gamma_L}(s) = \begin{cases} \underline{g}_w(s), & s \in \Gamma_L, \\ -M, & s \in [s_0, s_f] \setminus \Gamma_L. \end{cases}$$

By enforcing the constraint $E(s) \geq \underline{g}_{\Gamma_L}(s)$ instead of the constraint $E(s) \geq \underline{g}_w(s)$, the later constraint is relaxed on $[s_0, s_f] \setminus \Gamma_L$. Next, we introduce the following relaxed problem for Problem 5.3.1 by relaxing the original state constraints (153) and (154) on certain subintervals.

Problem 5.3.2 (**Relaxed minimum-energy path-tracking problem with fixed TOA**). Minimize the energy cost (150) while subject to constraints (151), (152), (155), (156), (157), (158), and state bounds

$$E(s) - \overline{g}_\Gamma(s) \leq 0, \tag{172}$$

$$\underline{g}_\Gamma(s) - E(s) \leq 0. \tag{173}$$

for all $s \in [s_0, s_f]$.

Similarly, one can also form the relaxed minimum-time and maximum-time path tracking problems with state constraints (172) and (173) instead of (153) and (154). In general, the minimum-time and maximum-time solutions of the relaxed problems are different from the corresponding solutions of the original (non-relaxed) problem. However, as shown by the following proposition, by choosing carefully where the constraints are relaxed, the the minimum-time and maximum-time solutions do not change on certain subintervals.

Proposition 5.3.2. Consider a function $\tilde{E} : [s_0, s_f] \to \mathbb{R}_+$, which is a solution to the ordinary differential equation (151) with a certain control input $\tilde{T}(s) \in [T_{\min}, T_{\max}]$. Let $\Gamma_U = \{s | E_U^*(s) < \tilde{E}(s), s \in [s_0, s_f]\}$ and $\Gamma_L = \{s | E_L^*(s) > \tilde{E}(s), s \in [s_0, s_f]\}$, where $E_U^*(s)$ and $E_L^*(s)$ are the specific kinetic energy solutions to the minimum-time and maximum-time path-tracking problems, respectively, with constraints (153) and (154). Let $E_{U_r}^*(s)$ and $E_{L_r}^*(s)$ be the specific kinetic energy solutions to the minimum-time and maximum-time path-tracking problems, respectively, with constraints $E(s) \leq \overline{g}_{\Gamma_U}(s)$ and $E(s) \geq \underline{g}_{\Gamma_L}(s)$ instead of (153) and (154). Then we have $E_U^*(s) = E_{U_r}^*(s)$ for all $s \in \Gamma_U$, and $E_L^*(s) = E_{L_r}^*(s)$ for all $s \in \Gamma_L$.

Proof. Define

$$E\,(s) = \begin{cases} \min\{\max\{E_{U_r}^*(s), E_U^*(s)\}, \tilde{E}(s)\}, & s \in \Gamma_U, \\ E_U^*\,(s), & s \in [s_0, s_f] \setminus \Gamma_U. \end{cases} \tag{174}$$

By the definition of $E(s)$ and Γ_U, $E(s) \geq E_U^*(s)$ on $[s_0, s_f]$, and $E(s)$ is continuous. Furthermore, $E(s_0) = E_U^*(s_0) = E_0$, $E(s_f) = E_U^*(s_f) = E_f$, and $\underline{g}_w(s) \leq E(s) \leq \overline{g}_w(s)$ for all $s \in [s_0, s_f]$. Hence, $E(s)$ is a feasible solution to the minimum-time path-tracking problem with constraints (153) and (154). If there exist $\tau \in \Gamma_U$ such that $E_{U_r}^*(\tau) > E_U^*(\tau)$, then by the definition of $E(s)$, we have $E(\tau) > E_U^*(\tau)$, and it follows from the continuity of E and E_U^* that $E(s) > E_U^*(s)$ in a neighborhood of τ. Hence, we have

$$\int_{s_0}^{s_f} \frac{1}{\sqrt{2E(s)}}\,\mathrm{d}s > \int_{s_0}^{s_f} \frac{1}{\sqrt{2E_U^*(s)}}\,\mathrm{d}s,$$

which means that $E(s)$ has a shorter final time than $E_U^*(s)$, which is a contradiction since E_U^* is the minimum-time solution.

Suppose there exist $\tau \in \Gamma$ such that $E_U^*(\tau) > E_{U_r}^*(\tau)$. Let $E(s) = \max\{E_{U_r}^*(s), E_U^*(s)\}$ for $s \in [s_0, s_f]$. Because $E_{U_r}^*(s) \leq \overline{g}_{\Gamma_U}(s)$ and $E_U^*(s) \leq \overline{g}_w(s) \leq \overline{g}_{\Gamma_U}(s)$ are always satisfied, we have $E(s) \leq \overline{g}_{\Gamma_U}(s)$ for all $s \in [s_0, s_f]$, and E is a feasible solution to the minimum-time problem with constraint $E(s) \leq \overline{g}_{\Gamma_U}(s)$. Since $E(\tau) = E_U^*(\tau) > E_{U_r}^*(\tau)$, by the continuity of E and $E_{U_r}^*$, we have $E(s) > E_{U_r}^*(s)$ in a neighborhood of τ. Therefore, following a similar argument as in the proof above, E_1 has a shorter final time than $E_{U_r}^*$, which is a contradiction. Hence, we must have $E_U^*(s) = E_{U_r}^*(s)$ for all $s \in \Gamma_U$. Similarly, one can prove that $E_L^*(s) = E_{L_r}^*(s)$ for all $s \in \Gamma_L$. The proof is omitted for the sake of brevity. □

The optimal solution to Problem 5.3.1 is given by the following theorem. Its proof is based on the optimal solution of the relaxed Problem 5.3.2.

Theorem 5.3.2. Suppose there exists a real number λ_t and a function $\tilde{E}$ given by $P(\tilde{E}(s), s) = \lambda_t$ for all $s \in [s_0, s_f]$, such that the specific kinetic energy E^* given by

$$E^*(s) = \begin{cases} E_L^*(s), & s \in \Gamma_L, \\ \tilde{E}(s), & s \in [s_0, s_f] \setminus (\Gamma_U \cup \Gamma_L), \\ E_U^*(s), & s \in \Gamma_U \end{cases} \tag{175}$$

satisfies the desired TOA, where $\Gamma_U = \{s | E_U^*(s) < \tilde{E}(s),\ s \in [s_0, s_f]\}$, and $\Gamma_L = \{s | E_L^*(s) > \tilde{E}(s),\ s \in [s_0, s_f]\}$. Then E^* is the optimal solution to Problem 5.3.1,

Proof. Consider the relaxed Problem 5.3.1 with the constraints (153) and (154) replaced by $E(s) \leq \overline{g}_{\Gamma_U}(s)$ and $E(s) \geq \underline{g}_{\Gamma_L}(s)$, respectively. Assume that the optimal specific kinetic energy solution of Problem 5.3.1 is E_r^*. Let $\lambda_{t_r}^*$ be the optimal costate value of the relaxed problem, and let $\tilde{E}_r$ be defined on $[s_0, s_f]$ by $P(\tilde{E}_r(s), s) = \lambda_{t_r}^*$. Let $T_r^*(s)$ be the optimal control associated with $E_r^*(s)$.

Let $\Gamma_{U_r} = \{s | E_r^*(s) = \overline{g}_{\Gamma_U}(s), s \in [s_0, s_f]\}$. By definition of $\overline{g}_{\Gamma_U}$, it is clear that $\Gamma_{U_r} \subseteq \Gamma_U$. According to Proposition 5.3.2, the time-optimal solution does not change on Γ_U by enforcing $E(s) \leq \overline{g}_{\Gamma_U}(s)$ instead of $E(s) \leq \overline{g}_w(s)$. Hence, following Lemma 5.3.2, we have $E_r^*(s) = E^*(s)$ for all $s \in \Gamma_U \supseteq \Gamma_{U_r}$. Similarly, let $\Gamma_{L_r} = \{s | E_r^*(s) = \underline{g}_{\Gamma_L}(s), s \in [s_0, s_f]\}$, then we have $E_r^*(s) = E^*(s)$ for all for $s \in \Gamma_L \supseteq \Gamma_{L_r}$.

Next, we will show that $E^*(s) = E_r^*(s)$ for all $s \in [s_0, s_f]$. Suppose, ad absurdum, that $E^*(s) \neq E_r^*(s)$ for some $s \in [s_0, s_f]$. Because E^* and E_r^* have the same TOA, i.e.,

$$\int_{s_0}^{s_f} \frac{1}{\sqrt{2E^*(s)}}\,\mathrm{d}s = \int_{s_0}^{s_f} \frac{1}{\sqrt{2E_r^*(s)}}\,\mathrm{d}s,$$

there must exist τ, $\gamma \in [s_0, s_f]$ such that $E_r^*(\tau) < E^*(\tau)$, and $E_r^*(\gamma) > E^*(\gamma)$.

When $\lambda_{t_r}^* \geq \lambda_t$. It follows from the definition of function P in (165) that $\tilde{E}_r(s) \geq \tilde{E}(s)$ for all $s \in [s_0, s_f]$. Let $q = \inf\{s | E_r^*(s) = E^*(s), s \in [\tau, s_f]\}$. Since $E_r^*(s_f) = E^*(s_f)$, q is well-defined. Similarly, let $p = \sup\{s | E_r^*(s) = E^*(s), s \in [s_0, \tau]\}$. Clearly, $\tau \in (p, q)$ and $(p, q) \cap (\Gamma_{U_r} \cup \Gamma_{L_r}) = \emptyset$ (since it has been shown that $E_r^*(s) = E^*(s)$ on $\Gamma_{U_r} \cup \Gamma_{L_r}$). It follows that $E(s) < \overline{g}_{\Gamma_U}(s)$ and $E(s) > \underline{g}_{\Gamma_L}(s)$ on (p, q), and $T_r^*(s)$ may only take the values of $T_{\max}$, $T_{\min}$ or $\tilde{T}$ on (p, q). Furthermore, we have $E_r^*(s) < E^*(s)$ for all $s \in (p, q)$.

Note that $E_r^*(s) = E^*(s)$ for $s \in \Gamma_L \cup \Gamma_U$ and $E_r^*(s) < E^*(s)$ for all $s \in (p, q)$. We have $(p, q) \subseteq [s_0, s_f] \setminus (\Gamma_L \cup \Gamma_U)$. Since $E^*(s) = \tilde{E}(s)$ on $[s_0, s_f] \setminus (\Gamma_L \cup \Gamma_U)$ by the definition of E^*, we have $E_r^*(s) < E^*(s) = \tilde{E}(s) \leq \tilde{E}_r(s)$ for all $s \in (p, q)$. Hence, $T_r^*(s)$ cannot be singular on (p, q), and either $T_r^*(s) = T_{\min}$ or $T_r^*(s) = T_{\max}$ for $s \in (p, q)$. Specifically, at τ, either $T_r^*(\tau) = T_{\min}$ or $T_r^*(\tau) = T_{\max}$. When $T_r^*(\tau) = T_{\min}$, with $E_r^*(s) < \tilde{E}_r(s)$ for all $s \in (p, q)$, we have $T_r^*(s) = T_{\min}$ for all $s \in (\tau, q)$ since $T_r^*(s)$ can not switch from $T_{\min}$ to $T_{\max}$ on (τ, q) according to Theorem 5.3.1. Note that $T^*(s) \geq T_{\min} = T_r^*(s)$ for all $s \in (p, q)$, by forward integration of $E^{*\prime}$ and $E_r^{*\prime}$ from τ to q with initial conditions satisfying $E_r^*(\tau) < E^*(\tau)$, we have $E_r^*(q) < E^*(q)$, which is a contradiction. Similarly, $T_r^*(\tau) = T_{\max}$ also leads to a contradiction after a backward integration from τ to p. Hence, we have shown that $E_r^*(s) = E^*(s)$ for all $s \in [s_0, s_f]$ when $\lambda_{t_r}^* \geq \lambda_t$.

Similarly, when $\lambda_{t_r}^* < \lambda_t$, by defining $q = \inf\{s | E_r^*(s) = E^*(s), s \in [\gamma, s_f]\}$ and $p = \sup\{s | E_r^*(s) = E^*(s), s \in [s_0, \gamma]\}$, we can also show that $E_r^*(\gamma) > E^*(\gamma)$ leads to a contradiction. Hence, we must have $E_r^*(s) = E^*(s)$ for all $s \in [s_0, s_f]$, i.e., $E^*(s)$ is the optimal solution to the relaxed problem.

Because $E^*(s) = E_U^*(s) \leq \overline{g}_w(s)$ for $s \in \Gamma_U$, $E^*(s) = E_L^*(s) \geq \underline{g}_w(s)$ for $s \in \Gamma_L$, and $\underline{g}_w(s) \leq E_L^*(s) < E^*(s) = \tilde{E}(s) < E_U^*(s) \leq \overline{g}_w(s)$ for $s \in [s_0, s_f] \setminus (\Gamma_U \cup \Gamma_L)$, it is clear that $\underline{g}_w(s) \leq E^*(s) \leq \overline{g}_w(s)$ for all $s \in [s_0, s_f]$, and $E^*(s)$ is feasible for Problem 5.3.1.

Hence, E^* is also the optimal solution to Problem 5.3.1 by Lemma 5.3.3, and the proof is complete. $\square$

5.4 An Energy-Optimal Path-Tracking Algorithm

Theorem 5.3.2 characterizes the switching structure of the optimal solution to the aircraft energy-optimal path-tracking problem. Although E_U^* can be computed using the algorithm in Ref. [137], and E_L^* can be computed in a similar manner, the optimal costate value λ_t^* is unknown. As a result, one is not readily able to choose the correct value of $\tilde{E}(s)$ for each $s \in [s_0, s_f]$ in order to construct the optimal specific kinetic energy as in (175). In this section a numerical algorithm is presented for solving Problem 5.3.1 by identifying the optimal costate value λ_t^*. This allows the computation of the associated function $\tilde{E}(s)$ from (164) and, subsequently, the optimal solution $E^*(s)$ from (175). To identify the constant λ_t^* and the associated singular arcs for a specific TOA, we need to search among a family of extremals associated with the prescribed geometric path for the correct value λ_t^*.

The algorithm for identifying the minimum-energy path-tracking control is given as follows:

Main Algorithm. *Compute the optimal solution for aircraft minimum-energy path-tracking operation with fixed TOA.*

1. Compute the state bounds $\overline{g}_w(s)$, $\underline{g}_w(s)$, and the functions $c_1(s)$, $c_2(s)$, $c_3(s)$ in Problem 5.3.1 as in Ref. [137].

2. Compute and store the values of $P\left(E(s), s\right)$ from equation (164) on a selected mesh $\mathcal{M}$ over the domain $[s_0, s_f] \times [E_{\min}, E_{\max}]$, where $[E_{\min}, E_{\max}]$ covers the possible range of the specific kinetic energy.

3. Compute the minimum-time solution $E_U^*(s)$ and the maximum-time solution $E_L^*(s)$ using the algorithm in Ref. [137]. Let the corresponding minimum and maximum TOA be $t_{\min}$ and $t_{\max}$, respectively. Proceed to the next step if $t_{\min} < t_f < t_{\max}$. Otherwise, quit the algorithm since the desired TOA is not possible and the given problem does not have a solution.

4. Apply a Newton-Raphson algorithm with adjusted bounds of the solution[100] to find the optimal costate value λ_t^* such that $\tau_f = t_f$, where τ_f is given by Algorithm 1 below with $\lambda = \lambda_t^*$. Then the corresponding specific kinetic energy $E^*(s)$ associated with the costate value λ_t^*, which is returned by Algorithm 1, is the optimal solution with TOA equal to t_f.

5. Compute the optimal controls thrust $T^*(s)$, bank angle $\phi^*(s)$, and lift coefficient $C_L^*(s)$ histories using equations (140), (144), and (145), respectively.

Next, we introduce an algorithm that computes the optimal speed solution and the TOA for a specific extremal with costate value λ.

Algorithm 1 *Compute the TOA τ_f and the corresponding optimal specific kinetic energy profile $E^*(s)$ for a given λ value*

1. Solve $P(\tilde{E}_\lambda(s), s) = \lambda$ for the function $\tilde{E}_\lambda(s)$ by interpolating the pre-computed and stored data of $P\left(E(s), s\right)$ for the given path on the mesh $\mathcal{M}$.

2. Compute the optimal specific kinetic energy $E^*(s)$ for the given λ using formula (175) along with the computed maximum-time specific kinetic energy $E_L^*(s)$ and minimum-time specific kinetic energy $E_U^*(s)$.

3. Compute the TOA τ_f for $E^*(s)$ using

$$\tau_f = \int_{s_0}^{s_f} \frac{1}{\sqrt{2E^*(s)}} \mathrm{d}s.$$

4. Return τ_f and $E^*(s)$.

According to the structure of the optimal specific energy profile in (175), it can be easily proved that the travel time τ_f of the energy-optimal solution decreases monotonically with increasing λ_t, since $\tilde{E}(s)$ increases monotonically with respect to λ_t for all $s \in [s_0, s_f]$ according to the definition of $\tilde{E}$ as in (165). In the Newton-Raphson algorithm with adjusted bounds used in Step 4. of the Main Algorithm, a bisection step is taken whenever Newton-Raphson would take the solution out of bounds. Since a bisection method alone is guaranteed to converge given the monotonicity property of the problem, such a hybrid method is also guaranteed to converge, and the Newton-Raphson steps can speed up the convergence.

5.5 Numerical Examples

Next, we validate the proposed energy-optimal tracking algorithm using a three-dimensional landing trajectory, as shown in Fig. 43. The initial position of the aircraft is $(-111, -17.3, 6)$ km and the final position is $(0, 0, 0)$ km. The initial speed is v_0=240 m/s, and the final speed is v_f=95 m/s. Both the initial and final path angles are 0°. The initial heading angle is 0°, and the final heading angle is $-25°$. The horizontal projection of the trajectory contains two turning maneuvers, as shown in Fig. 44.

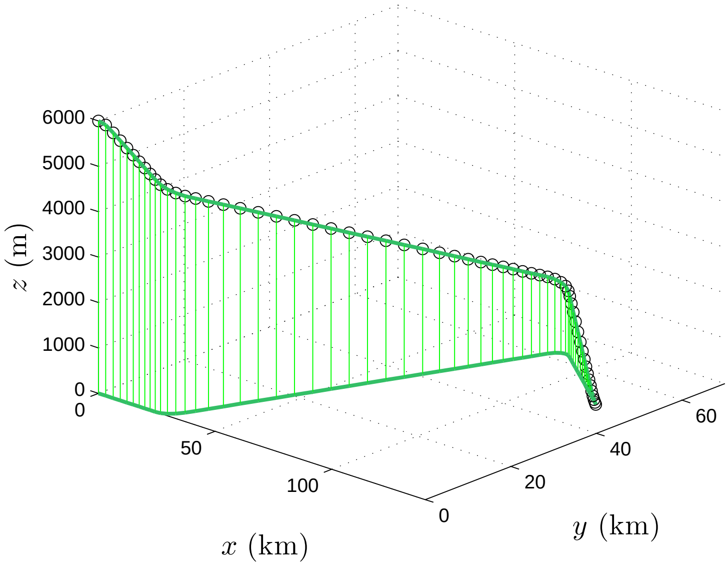

Figure 43: 3D Geometric Trajectory.

The speed and control bounds considered during the time parameterization process are $M_a \leq 0.9$, where M_a is the Mach number, $C_{L_{\min}} = -0.47$, $C_{L_{\max}} = 1.73$, $\phi_{\min} = -15°$, $\phi_{\max} = 15°$, $T_{\min} = 0$, and $T_{\max} = 1126.3$ kN.

The path is processed using the algorithm introduced in the previous section with different TOA requirements. Figures 45 and 46 show the optimal speed profiles for the minimum-energy aircraft path-tracking for several TOA values. It can be seen from these figures that with different TOA values t_f, different parts of the minimum-time and/or the

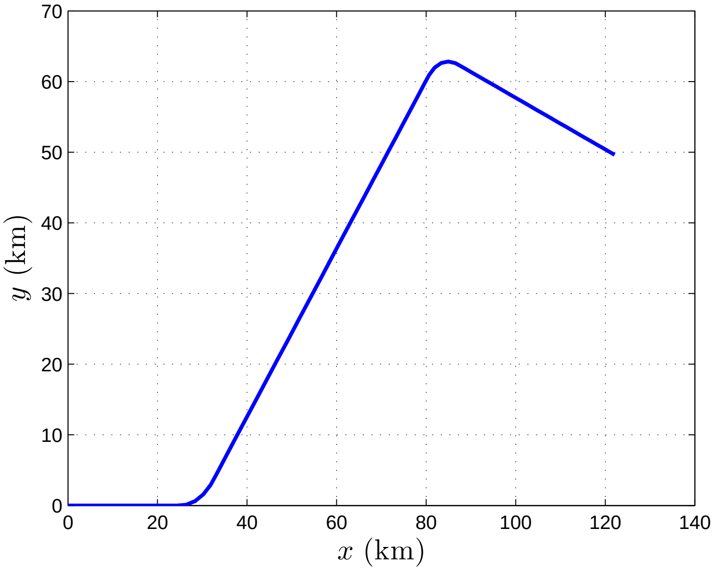

Figure 44: X-Y plane projection of the geometric trajectory.

maximum-time speed profile can be involved in the minimum-energy solution, together with the corresponding singular arcs. Figures 47 and 48 illustrate the minimum-energy control histories for $t_f = 800\,\mathrm{s}$ and $t_f = 1400\,\mathrm{s}$, respectively. In these figures, the throttle is the ratio of the actual thrust to the maximum thrust $T_{\max}$. It is clear that all solutions satisfy the speed and control constraints along the path.

To evaluate the fuel economy of the energy-optimal solution, a fuel-optimal control problem was solved using a numerical optimal control approach with the fuel consumption model (148) as the cost function. The constraints of the fuel-optimal control problem are identical to those of Problem 5.3.1. The fuel-optimal control problem was converted into a nonlinear programming problem via direct transcription[25], and solved using the sparse nonlinear optimization software SNOPT[54]. The density function based mesh refinement method in Ref. [138] was used to generate a mesh such that the state bounds (138) and (139) can be approximated more accurately with a limited number of grid points. The parameters for the computation of η_0 in equation (147) were stored in a look-up table, and were provided to the nonlinear optimization solver.

The same four cases shown in Fig. 45 ($t_f = 800\,\mathrm{s}$, $1000\,\mathrm{s}$, $1200\,\mathrm{s}$, $1400\,\mathrm{s}$) were solved

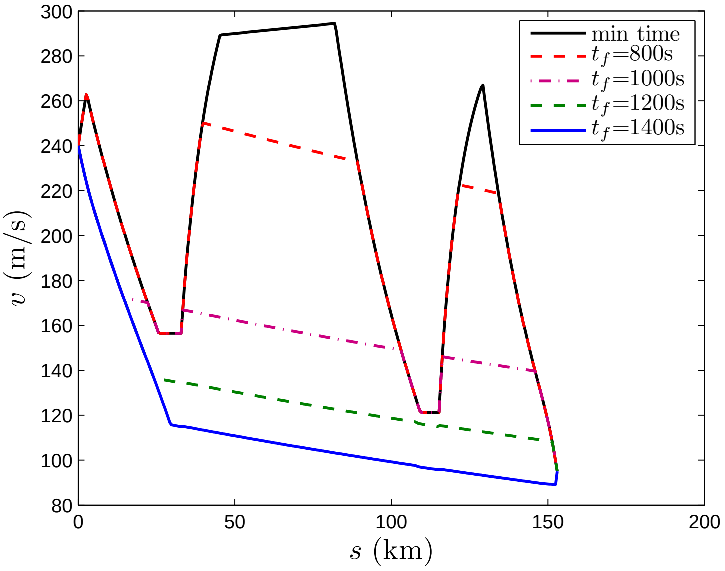

Figure 45: Energy-optimal speed profiles with different TOA, path coordinate domain.

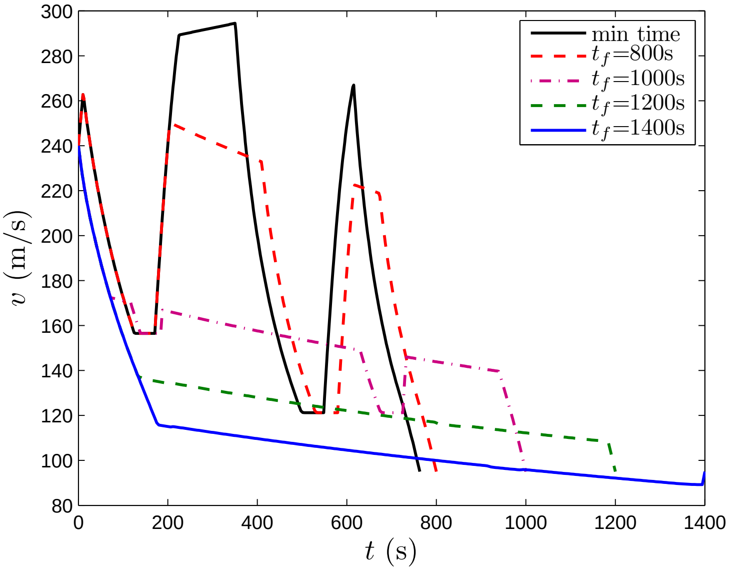

Figure 46: Energy-optimal speed profiles with different TOA, time domain.

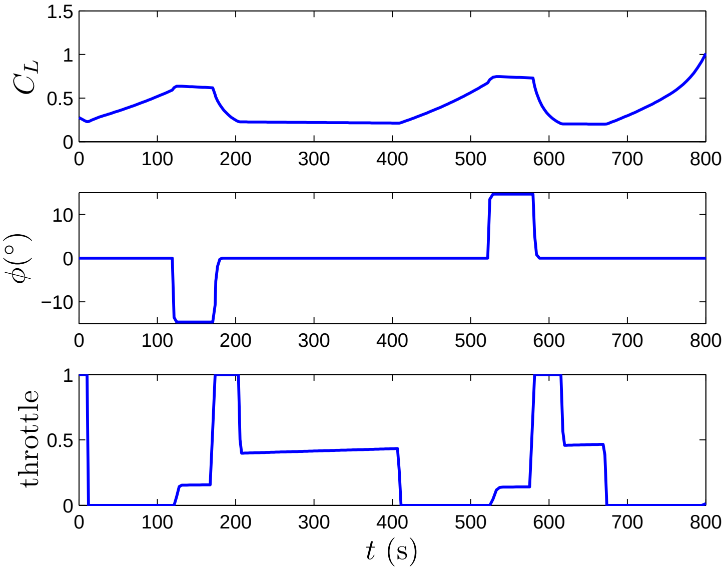

Figure 47: Energy-optimal control histories with $t_f = 800\,\text{s}$.

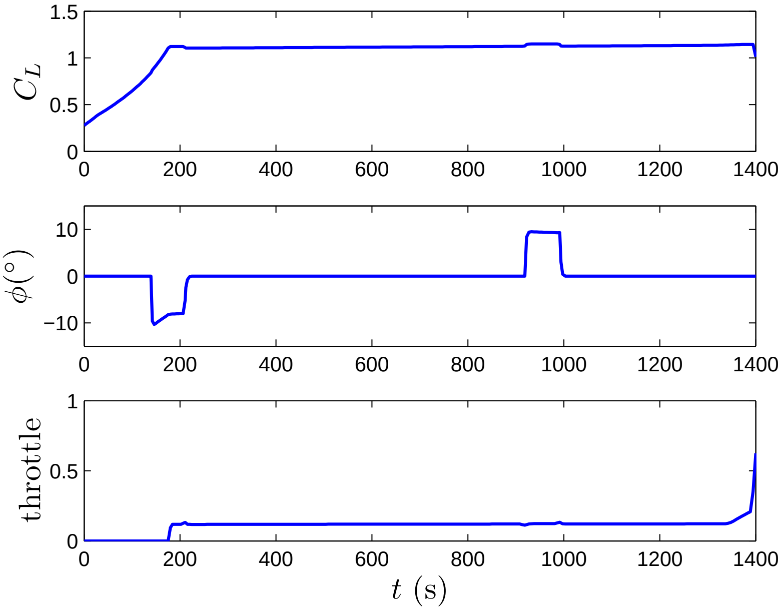

Figure 48: Energy-optimal control histories with $t_f = 1400\,\text{s}$.

using the numerical optimal control approach for minimum-fuel path-tracking, and the optimization results were compared to those given by the energy-optimal path-tracking algorithm. The results of the comparison are shown in Figs. 49 and 50. It is clear from these figures that the energy-optimal solutions are very close to the minimum-fuel solutions. Note that the singular arcs in the minimum-fuel problem cause numerical issues (oscillations along the singular arcs in Figs. 49 and 50). This is a well-known phenomenon observed when computing singular arcs using direct trajectory optimization methods. Furthermore, the computation time of the numerical optimization approach is much longer than the proposed energy-optimal path-tracking algorithm: a Matlab implementation of the energy-optimal path-tracking control algorithm finds the optimal solution in 2-4 seconds, while the Nonlinear Programming solver takes at least 5 minutes (for some cases, more than 20 minutes) to find a convergent fuel-optimal solution.

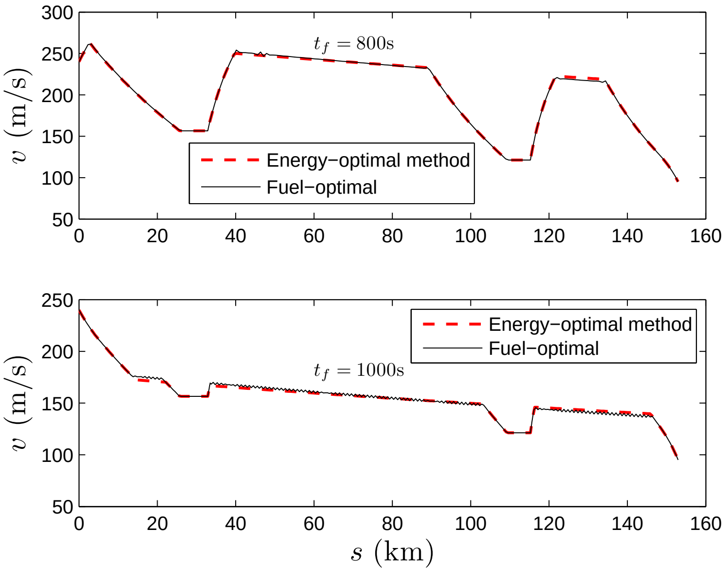

Figure 49: Comparison of fuel-optimal and energy-optimal speed profiles, $t_f = 800\,\mathrm{s}$ and $t_f = 1000\,\mathrm{s}$.

5.6 Conclusions

The method presented in this chapter computes an energy-optimal time-parameterization for an aircraft to follow a given three-dimensional geometric path with fixed time-of-arrival

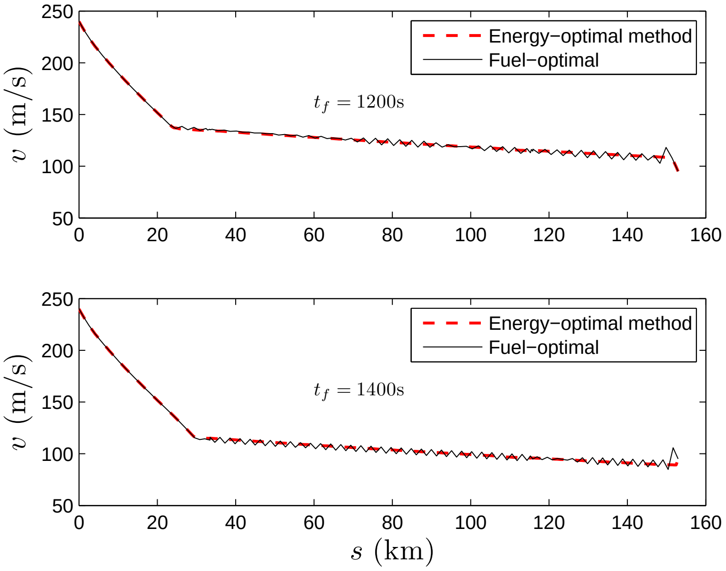

Figure 50: Comparison of fuel-optimal and energy-optimal speed profiles, $t_f = 1200\,\mathrm{s}$ and $t_f = 1400\,\mathrm{s}$.

(TOA). The switching structure of the optimal solution is analyzed using optimal control theory. The switching structure may vary depending on the given TOA. However, for a given path and a fixed TOA, the structure is uniquely determined. It is proved that the energy-optimal solution is a combination of the minimum-time solution, the maximum-time solution, and energy-saving singular arcs. As verified by numerical optimization results, this method is computationally efficient, and can be applied in real-time for improving the fuel efficiency of airline scheduling and terminal phase operations.

CHAPTER VI

INITIAL GUESS GENERATION FOR LANDING TRAJECTORY OPTIMIZATION

As discussed in Chapter 1, a landing trajectory obtained using a Nonlinear Programming (NLP) based numerical optimal control approach via direct transcription provides both feasibility and optimality, which are important for the emergency landing scenario. However, the application of such an approach is limited by the convergence of the optimization algorithm (specifically, the NLP solver working jointly with the numerical optimal control algorithm), which depends extensively on the quality of the initial guess, including the time history of all state and control variables, as well as any unknown parameters. Thus, convergence is not guaranteed, in general. In numerical optimal control algorithms, the initial guess is usually automatically generated by setting the state and control variables to constants, or as simple affine functions. The user may also try different initial guesses if he/she has some insight into the specific problem. In this chapter, we introduce a new scheme for automatic initial guess generation for aircraft landing trajectory optimization problems.

6.1 Feasible Landing Trajectory Generation

The quality of the commonly used affine initial guess is usually unreliable in the sense that such a guess is rarely feasible, i.e., the time histories of the state and control variables of the initial guess do not satisfy the differential equations governing the system dynamics. Throughout this thesis, we say that the NLP solver fails if the result returned by the solver does not satisfy either the feasibility tolerance 1×10^{-6} or the optimality tolerance 1×10^{-3}. Although NLP solvers may proceed to a feasible region by updating decision variables using penalty methods [34], often the solver fails if the initial guess is far away from the feasible region.

For the landing trajectory optimization problem, such failures are commonly observed when affine or constant initial guesses are used. Therefore, to reduce the failure rate of penalty methods for identifying feasible regions, as well as to improve the robustness of the optimization scheme, it is desirable to provide feasible landing trajectory initial guess to the NLP solver.

The generation of a feasible trajectory, as described in Problem 1.2.1 proposed in Chapter 1, is not a trivial task in the case of landing trajectory generation due to complicated aircraft dynamics. With a hierarchical approach as introduced in Chapter 1, the generation of a feasible trajectory can be decomposed into two tasks involving the geometric layer and the dynamics layer, respectively, as illustrated in Fig. 51. Such an approach generates first a purely geometric collision-free path connecting the initial and final positions. After such a path is obtained, in the second step, a certain time parameterization is assigned to the path, which converts the path into a trajectory. It is required that the time parameterization found in the second step must satisfy the dynamics and other state and control constraints.

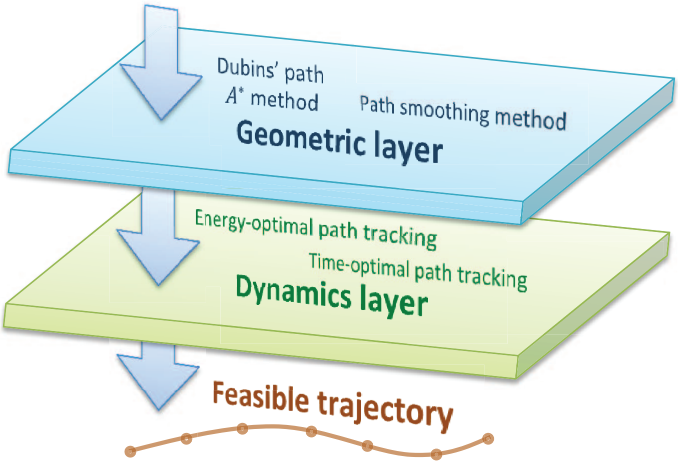

Figure 51: Hierarchical approach to feasible trajectory generation.

Although many efficient collision-free path planning methods are available for the first step in the hierarchical trajectory generation approach, few of them can be directly applied for the generation of aircraft landing trajectories because the generated paths are not smooth enough to be followed by the aircraft. On the other hand, variations of Dubins' paths, although reasonably smooth, cannot meet the requirement for collision avoidance. The path smoothing method introduced in Chapter 3 works as a post-processing technique on the geometric layer for these non-smooth path planning methods. The resultant paths obtained using this method are smooth enough for the aircraft dynamics, while retaining the collision avoidance feature of the original non-smooth paths.

The time-optimal aircraft path tracking method introduced in Chapter 4 and the fixed final time, energy-optimal aircraft path tracking method introduced in Chapter 5 fit into the dynamics layer of the hierarchical approach. Both methods are able to generate a feasible time-parameterization to the prescribed geometric path (if such a parameterization exists) given by the geometric path planning methods employed in the first step of the hierarchical approach. The energy-optimal method can help improve fuel efficiency in the landing phase during normal scheduled flight. The time-optimal formulation provides the shortest landing time, and is more suitable for emergency landing scenarios.

The hierarchical aircraft landing trajectory generation scheme considered in this thesis includes a geometric path planner and a path smoothing method in the geometric layer, and a time-optimal path tracking method in the dynamics layer. The main characteristics of such a hierarchical scheme when compared to a NLP based numerical optimal control approach are described below:

1. **Robustness**

   The robustness of the hierarchical scheme is determined by the robustness of the algorithms employed in the geometric and dynamics layers, and the interaction between these methods. Unlike the NLP approach that often encounters convergence issues, geometric path planning methods such as A^*, D^*, visibility graph, and Dubins' paths are much more reliable. Path smoothing updates the path in a neighborhood by solving a sequence of Quadratic Programming problems. For each problem, the solution is guaranteed to converge. Hence, the generation of a geometric path in the hierarchical scheme is highly reliable.

   On the other hand, this hierarchical approach applies semi-analytic methods on the dynamics layer for the time parameterization of the geometric path. As shown in

Chapter 4 and Chapter 5, these semi-analytic methods are guaranteed to find the time parameterization if such a parameterization exists given the path.

Hence, the individual methods in this hierarchical approach do not cause any robustness issues, and the robustness of the hierarchical scheme, i.e., the feasibility of the generated trajectory, depends on whether the geometric path planner can properly generate the geometric path for which a feasible time-parameterization exists. As will be shown later in this chapter, by properly tuning the geometric path planner (mainly by avoiding aggressive turning maneuvers), the hierarchical scheme can generate a feasible trajectory for the overwhelming majority of cases.

2. **Optimality** Although semi-analytical optimal path tracking methods in the dynamics layer of the hierarchical scheme can compute the exact optimal solution for the given path geometry, the geometric path planners usually do not generate paths with optimal geometry, which depends on system dynamics. Instead, these geometric path planners can only generate paths with reasonably good geometry. Therefore, the landing trajectories generated by a hierarchical scheme are obviously sub-optimal, in general, compared to convergent solutions from a NLP approach. This is especially true when conservatism is introduced in the geometric path planer to ensure the feasibility of the solution.

3. **Computation speed** This is considered to be a major advantage of a hierarchical scheme over a general NLP approach. Current geometric path generation methods are highly efficient. Path smoothing based on Quadratic Programming can also be solved efficiently. In the dynamics layer, optimal path tracking methods are based on semi-analytic solutions, for which the majority of computations deal with the integration of system dynamics, which can also be computed efficiently. Hence, although more subproblems are solved in this hierarchical approach, the overall computation speed is much faster than the NLP, which solves the numerical optimal control problem directly.

Since the landing trajectories generated using the hierarchical scheme are mostly feasible, and, in general, reasonably close to the optimal solution, they are good initial guesses to a numerical optimal control solver. Even if such a hierarchical scheme fails to provide a feasible trajectory, the generated trajectory is still not far away from the feasible region, hence there is a good chance that the feasibility of the solution can be recovered by penalty methods using generic NLP solvers, and thus the optimality can be further improved. The schema of such an initial guess generation technique for numerical optimal control algorithms is illustrated in Fig. 52.

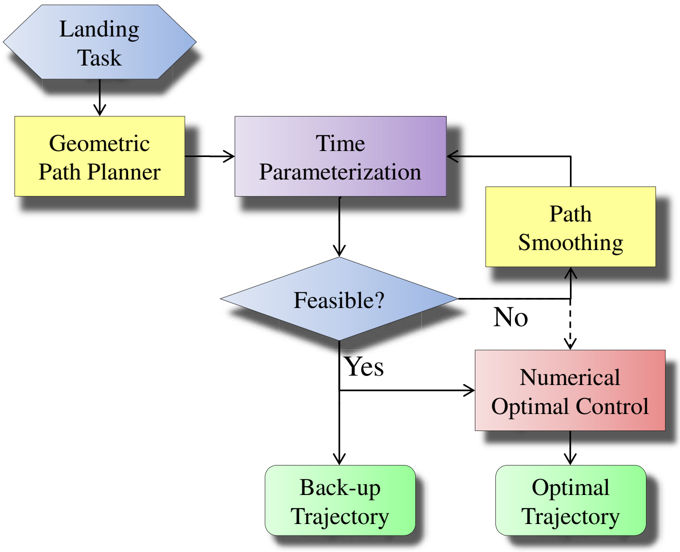

Figure 52: Schematic of landing trajectory optimization.

As shown in Fig. 52, the time-optimal path tracking method first generates a trajectory by assigning a time parameterization along the path given by the geometric path planner. If the trajectory is feasible, then it is used as an initial guess for the numerical optimal control solver. Meanwhile, such a feasible trajectory is also stored as a back-up plan in case of the failure of the NLP solver. If the trajectory generated by the time-optimal path tracking method is not feasible, then the path is revised using the path smoothing method, and optimal path tracking is applied again to the smoothed path. Such a process is repeated until either the trajectory is feasible, or the maximum number of iterations is reached. If no feasible trajectory can be obtained after reaching the iteration limit, the infeasible trajectory is passed to the numerical optimal control algorithm, which makes a last attempt to produce a feasible trajectory. If this last attempt is not successful, then either there does not exists a feasible trajectory to the problem, or both the hierarchical scheme and the NLP solver have failed.

6.2 A Three Dimensional Landing Path Primitive Generation Method

Next, we introduce a landing path primitive generation method based on the suboptimal solution of a three-dimensional variation of the classical Markov-Dubins problem [44], which characterizes curvature constrained paths of minimum length in the plane. Specifically, we consider the generation of a geometric path which connects the initial and terminal configurations of the aircraft subject to the following requirements:

1. The projection of the three-dimensional curve onto the horizontal plane corresponds to a Dubins-like path (that is, it is composed of concatenations of circular arcs and line segments);
2. An aircraft traveling along the path is descending continuously until the final destination is reached.

Such an geometric problem can be formulated, equivalently, as an optimal control problem of a point mass particle of unit mass, with the kinematic model is described by the following equations

$$x' = \cos\psi\cos\gamma, \tag{176}$$

$$y' = \sin\psi\cos\gamma, \tag{177}$$

$$z' = \sin\gamma, \tag{178}$$

$$\psi' = \frac{u}{R_{\min}(z)}, \tag{179}$$

where $(x, y, z) \in \mathbb{R}^3$ is the position vector, $\psi \in [0, 2\pi)$ is the heading of the particle, $R_{\min}$ is a positive number, which may depend on the altitude z, γ is the flight path angle, which is treated as an control input, and u is a control input that determines the rate of change of the heading angle. Prime denotes differentiation with respect to the arc length s. It is furthermore assumed that $\gamma \in [\gamma_{\min}, \gamma_{\max}] \subseteq [-\pi/2, 0]$, and $u \in [-\delta, 1]$, where $\delta \in (0, 1]$ (i.e., the steering constraints may be asymmetric [13]).

Problem 6.2.1. Find the controls u^* and γ^* that steer the system described by Eqs.(176)-(179) from (x_0, y_0, z_0, ψ_0) (prescribed) to (x_f, y_f, z_f, ψ_f) (prescribed) with $z_f < z_0$, such that the total length of the ensuing path s_f (free) is minimum.

In this section we are interested in finding a suboptimal solution to Problem 6.2.1 for any prescribed pair of boundary configurations. A straightforward way to characterize suboptimal solutions for Problem 6.2.1 is to decouple the path planning problem into a steering problem in the x-y plane (or more precisely $\mathbb{R}^2 \times \mathbb{S}^1$), and another steering problem in the vertical plane (one-dimensional problem).

6.2.1 The Minimal Length Curve Problem in the Horizontal Plane

First, we address a path-planning problem in the horizontal x-y plane, which will allow us to address Problem 6.2.1. To this aim, it is assumed that the solution of the steering problem in $\mathbb{R}^2 \times \mathbb{S}^1$ follows the Dubins pattern, that is, the projection of a (suboptimal) solution of Problem 6.2.1 on the x-y plane is a concatenation of two circular arcs of minimum radius interconnected by either a straight line or another circular arc. Note that the radii of different circular arcs of the projection of a path that solves Problem 6.2.1 on the x-y plane may not be equal, as a result of the fact that the steering capacity of the aircraft depends on the altitude.

In order to obtain a simple formula for computing the minimum turning radius of an aircraft as a function of the altitude, we first observe that the rate of change of ψ of an aircraft of mass m traveling with speed v at an altitude z is given by [45]

$$\psi' = -\frac{L(C_L, v, z)\sin\phi}{mv^2\cos\gamma}, \tag{180}$$

where ϕ is the bank angle, $L = L(C_L, v, z)$ is the lift and C_L is the lift coefficient. If we assume that $v = v(z)$, we can obtain a rough approximation of $R_{\min}$ as follows

$$R_{\min}(z;\gamma) = \frac{mv(z)^2\cos\gamma}{L(C_L^{\max}, v(z), z)\sin\phi_{\max}}, \tag{181}$$

where $\phi_{\max}$ and $C_L^{\max}$ denote, respectively, the upper bounds on the bank angle and the lift coefficient.

Equation (181) implies that an aircraft is less maneuverable, in terms of performing sharp turns, at higher altitudes than it is in lower altitudes. Let $R_0 \triangleq R_{\min}(z_0; 0)$, $R_m \triangleq R_{\min}(z_m; 0)$, where $z_m = (z_0 + z_f)/2$, and $R_f \triangleq R_{\min}(z_f; 0)$. In addition, let us assume that along the first and the last circular arc of the Dubins path the quantity $R_{\min}$ in Eq. (179) is constant and equal to R_0 and R_f, respectively. Furthermore, if the Dubins path consists of three circular arcs, then the quantity $R_{\min}$ along the middle arc is constant and equal to R_m. Note that $R_0 \geq R_m \geq R_f$. In order to obtain more conservative estimates of the $R_{\min}$, and thus reduce the risk of selecting a small value for the minimum turning radius that can lead to dynamically infeasible paths for the aircraft, we multiply R_0, R_m, and R_f by a safety factor k_0, k_m, and $k_f > 1$, respectively.

Next, we formulate a minimum-length problem on the horizontal plane x-y plane.

Problem 6.2.2. Given two configurations (x_0, y_0, ψ_0) and (x_f, y_f, ψ_f) in $\mathbb{R}^2 \times \mathbb{S}^1$, find a minimum-length curve that connects the two configurations and belongs necessarily to the following family of paths

$$\mathcal{P} \triangleq \{C^{\pm}(R_0) \circ C^{\mp}(R_m) \circ C^{\pm}(R_f), C^{\pm}(R_0) \circ S \circ C^{\pm}(R_f), C^{\pm}(R_0) \circ S \circ C^{\mp}(R_f)\}, \quad (182)$$

where $C^-(R_\ell)$ $(C^+(R_\ell))$ and S denote a circular arc of radius R_ℓ, where $\ell \in \{0, m, f\}$, traversed clockwise (counterclockwise) and a line segment, respectively, and $\circ$ denotes the concatenation of two consecutive arcs.

6.2.2 Vertical Descent Profile Generation

In this section, we obtain a three dimensional landing path by generating a vertical profile for the two dimensional Dubins' path in the previous section. In the subsequent analysis, it is assumed that the trigonometric sine function of the path angle, which is denoted by $\chi(s) = \sin(\gamma(s))$, is a piecewise linear function with three segments along the path length s of the Dubins' path. It is also assumed that χ is constant along the second segment. We will find a function χ with $\chi(0) = 0$, $\chi(s_f) = \chi_f = \sin(\gamma_f) < 0$, and $\chi(s) \leq 0$, $s \in [0, s_f]$ such that the boundary conditions for the vertical path planning problem, i.e., $z(0) = z_0$ and $z(s_f) = z_f$ are satisfied. We also require that for the first and the third segments,

$|\chi'| = a > 0$. Let s_1 denote the switching point between the first and second segment, and let s_2 denote the switching point between the second and the third segment. The descend profile χ subject to these constraints is given by the following expression

$$\chi(s) = \begin{cases} -as, & s \in [s_0, s_1], \\ \chi_m, & s \in (s_1, s_2), \\ \chi_e, & s \in [s_2, s_f], \end{cases} \tag{183}$$

where χ_m is a negative number to be determined, and either $\chi_e(s) = \chi_f + a(s - s_f)$ or $\chi_e(s) = \chi_f - a(s - s_f)$, depending on the desired boundary condition as will be discussed shortly afterwards.

Note that

$$z_f = z_0 + \int_0^{s_f} z'(s)\,\mathrm{d}s = z_0 + \int_0^{s_f} \sin(\gamma(s))\,\mathrm{d}s = z_0 + \int_0^{s_f} \chi(s)\,\mathrm{d}s, \tag{184}$$

and we have that the signed area enclosed between the image of $\chi(s)$ and the s-axis, which is the integral in the above expression, must equal to a constant $z_f - z_0$. Such a constraint affects the choice of χ_m. We consider four cases for the choice of χ_m as shown below:

Case I: $z_f - z_0 > -\chi_f^2/2a$. In this case, the constraint $|\chi'(s)| = a$ for $s \in [0, s_1] \cup [s_2, s_f]$ and the integral constraint (184) are not compatible, and we relax the previous constraint by choosing $a = \chi_f^2/2(z_0 - z_f)$. Also, then χ is given by (183) with $s_1 = 0$, $s_2 = s_f + \chi_f/a$, $\chi_m = 0$, and $\chi_e = \chi_f + a(s - s_f)$. The profile of χ for this case is shown in Fig. 53, in which $\chi(s)$ is represented by the golden dotted curve.

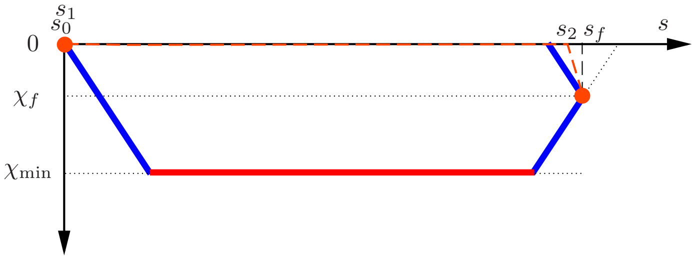

Figure 53: Vertical profile generation, Case 1.

Case II: $s_f\chi_f - \chi_f^2/2a \leq z_f - z_0 \leq -\chi_f^2/2a$. In this case, χ_m is given by the solution to the following equation

$$(s_f - \Delta_1 - \Delta_2)\chi_m - \frac{\chi_m^2}{2a} + \frac{\Delta_2}{2}(\chi_m + \chi_f) = z_f - z_1,$$

where $\Delta_1 = -\chi_m/a$ and $\Delta_2 = (\chi_m - \chi_f)/a$. The solution to the above equation is

$$\chi_m = \frac{(2a(z_f - z_0) + \chi_f^2)}{2as_f + 2\chi_f}.$$

The switching points in (183) are given by $s_1 = \Delta_1$, $s_2 = s_f - \Delta_2$. $\chi_e = \chi_f - a(s - s_f)$. The profile of χ for this case is illustrated in Fig. 54

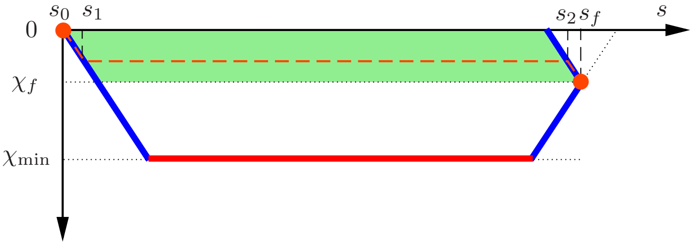

Figure 54: Vertical profile generation, Case 2.

Case III: $-\chi_f^2/2a + s_f\chi_{\min} + 3\chi_{\min}^2/a - \chi_f\chi_{\min}/a \leq z_f - z_0 < s_f\chi_f - \chi_f^2/2a$. In this case, χ_m must satisfy the following equation

$$3\chi_m^2 + (as_f - \chi_f)\chi_m - \chi_f^2/2 - a(z_f - z_0) = 0.$$

The above equation has two solutions. The following solution is used for (183) since the other solution is not feasible:

$$\chi_m = \frac{1}{6}\left(\chi_f - as_f - \sqrt{(as_f - \chi_f)^2 + 6\chi_f^2 + 12a(z_f - z_0)}\right).$$

The switching points s_1 and s_2 in (183) are given by $s_1 = -\chi_m/a$, $s_2 = s_f - (\chi_f - \chi_m)/a$. The third segment of $\chi(s)$ as in (183) is defined by $\chi_e = \chi_f + a(s - s_f)$. A representative solution $\chi(s)$ for this case is shown as the golden dotted curve in Fig. 55.

Case IV: $\chi_f^2/a + s_f\chi_{\min} + \chi_{\min}^2/2a - \chi_f\chi_{\min}/a > z_f - z_0$. In this case, the downward velocity is not sufficiently large to guarantee that an aircraft traversing a path whose projection on the x-y plane is a Dubins path can reach the desired final altitude at the end of its course. In order to increase the length of the descent path without changing the structure

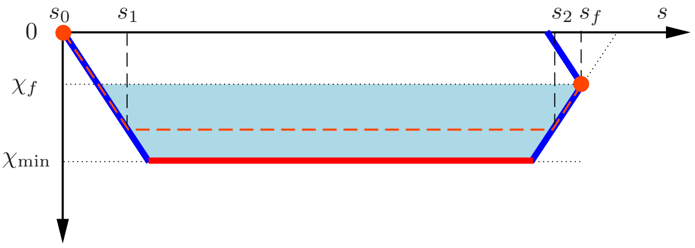

Figure 55: Vertical profile generation, Case 3.

of the path in the x-y plane, we simply add one or more loops along the first helical arc. In this way, the projection of the last arc on the x-y will remain the same but the length of the ensuing path will be increased. Specifically, let s_n denote the total length of a full loop with radius R_0, we find the minimum number of loops $n \geq 1$ such that

$$\frac{\chi_f^2}{a} + (s_f + ns_n)\chi_{\min} + \frac{\chi_{\min}^2}{2a} - \frac{\chi_f \chi_{\min}}{a} \leq z_f - z_0. \tag{185}$$

The total path length is update by $s_f \leftarrow s_f + ns_n$. Then one of the previous three cases can be applied to compute the function χ for the new path including n additional loops at the very beginning. The effect of including additional loops in the Dubins path is illustrated by Fig. 56.

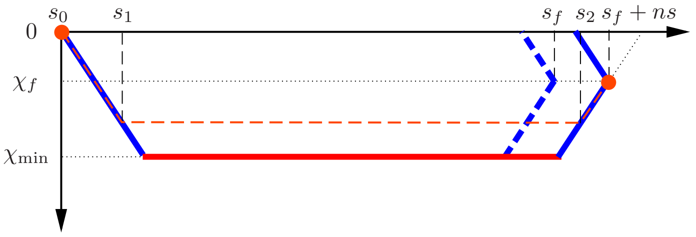

Figure 56: Vertical profile generation, Case 4.

After $\chi(s)$ is obtained for all $s \in [0, s_f]$, the vertical profile of the descent is given by the following integral

$$z(s) = z_0 + \int_0^{s_f} \chi(s)\, \mathrm{d}s.$$

6.3 Simulation Results

In this section, we apply the hierarchical trajectory generation approach to obtain an initial guess for the numerical optimal control software DENMRA for solving the minimum time landing trajectory optimization problem. In particular, the three-dimensional landing path generation method in Section 6.2 is used to generate a geometric path satisfying the boundary conditions, which include the position, path angle, and heading angle at the start and end points of the path. After a landing path is obtained, the time-optimal path tracking method in Chapter 4 is applied to convert the geometric path into a trajectory by assigning a time parameterization to the path and computing the state and control variables using inverse dynamics. If the generated trajectory is feasible, then it is used to generate initial guesses for DENMRA. If the trajectory is not feasible, then a maximum number of two path smoothing iterations are applied, as described in Section 6.1, to modify the path geometry until the trajectory is feasible. If no feasible trajectory can be generated, the trajectory from the last iteration is used to generate initial guesses for DENMRA.

Numerical results show that the initial guess generated using this method usually captures the key features of a local optimal solution, as shown in Figs. 57 and 58. In these plots, the red lines are the initial guess, and the blue lines with markers are the optimization result of DENMRA using the generated initial guess. The difference between the initial guess and the optimal trajectory is observed for some landing cases when the horizontal range of flight (horizontal distance between the aircraft's initial position and the airport) is small, as shown in Fig. 59. Simulation results indicate that the geometry of the optimal landing trajectory is related to the ratio of the horizontal range to the altitude change. When this ratio is large enough, the flight time is mainly determined by the aircraft's movement in the horizontal plane, and the projection of the optimal trajectory to the horizontal plane resembles the typical circle-straight line-circle pattern of the Dubins' path for shorter travel time. When this ratio is small, the total fight time is more limited by the aircraft's dynamics for descent—the aircraft must fly over certain horizontal distance to lose altitude, in which case the optimal landing trajectory tend to exhibit more complex geometry.

A series of numerical experiments were performed to test the effectiveness of the proposed initial guess generation scheme for improving the convergence of the DENMRA for solving the minimum time emergency landing problem. In all experiments, some boundary conditions are fixed, including the initial speed $v_0 = 240\,\mathrm{m/s}$, the final speed $v_f = 95\,\mathrm{m/s}$,

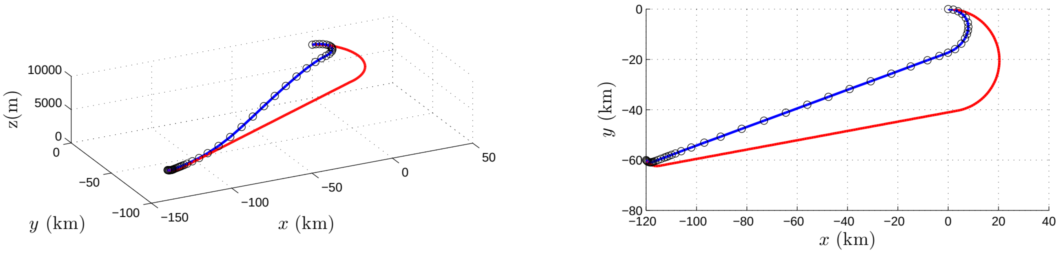

Figure 57: Trajectory comparison, case 1.

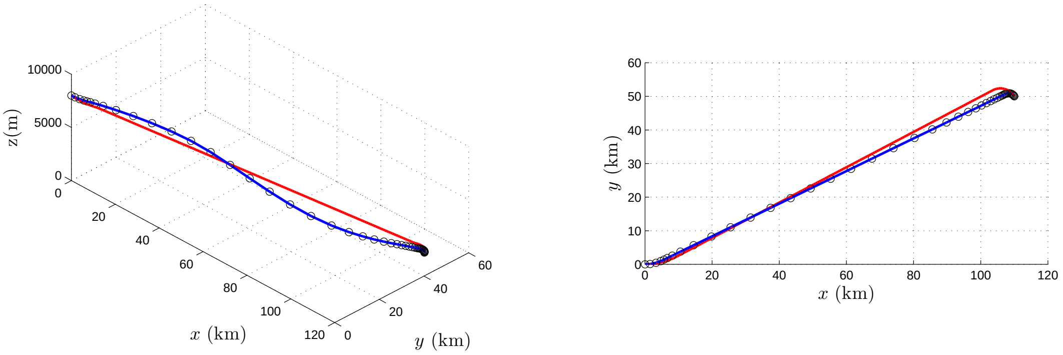

Figure 58: Trajectory comparison, case 2.

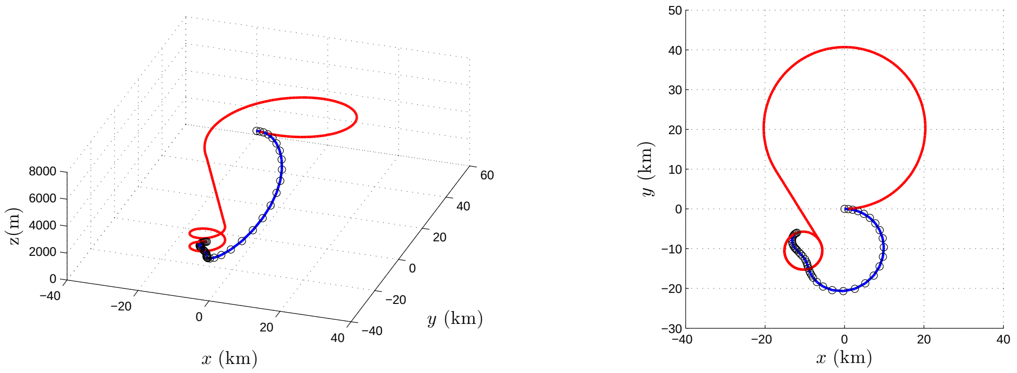

Figure 59: Trajectory comparison, case 3.

the initial path angle $\gamma_0 = 0\,\mathrm{deg}$, the final path angle $\gamma_f = 0\,\mathrm{deg}$, the initial position $x_0 = 0\,\mathrm{km}$, $y_0 = 0\,\mathrm{km}$, and the initial heading angle $\psi_0 = 0$. The other boundary conditions are generated randomly for each experiment. Specifically, the airport position is sampled uniformly from a disc on the ground (zero altitude) with radius $R_{\max} = 200\,\mathrm{km}$, the runway heading is uniformly distributed in $[0, 2\pi]$, and the initial altitude is uniformly distributed between $6\,\mathrm{km}$ and $10\,\mathrm{km}$.

In each experiment, after the boundary conditions are determined, a three-dimensional landing trajectory, which is referred to as the TP trajectory henceforth, is generated by applying the time-optimal tracking method to the Dubins' type landing path generator as introduced in Section 6.1. The TP trajectory is interpolated to obtain the initial guesses for DENMRA. In particular, the DENMRA performs a maximum of three iterations. The DENMRA starts from 50 grid points, and five grid points are added for each additional iteration. If the desired feasibility and optimality tolerance can not be satisfied by the current iteration, then the mesh size is increased, and a subsequent iteration is performed using an initial guess obtained by interpolating the TP trajectory on the new mesh. If the desired tolerances, including the feasibility tolerance (1×10^{-6}) and the optimality tolerance (1×10^{-2}), are satisfied, then the DENMRA is terminated after the current iteration.

The key experimental data and results such as the boundary conditions, and whether DENMRA converged, were recorded. As a comparison, in each experiment, affine initial guesses interpolating the boundary conditions and constant control inputs were also used to start the NLP solver, and the settings of DENMRA were identical to those when the TP trajectory initial guesses are applied.

The details about the boundary condition used in the experiments are shown below:

$$\begin{aligned} x_0 = 0, \quad y_0 = 0, \quad z_f = 0, \quad \psi_0 = 0, \\ \gamma_0 = 0, \quad \gamma_f = 0, \quad v_0 = 240, \quad v_f = 95, \end{aligned}$$

$$\begin{aligned} z_0 \leftarrow \mathrm{U}([6, 10])\mathrm{km}, \qquad \psi_f \leftarrow U([0, 2\pi]), \qquad \theta \leftarrow U([0, 2\pi]), \qquad \varpi \leftarrow U([0, 1]), \\ R = R_{\max}\sqrt{\varpi}, \qquad x_f = x_0 + R_{\max}\cos\theta, \quad y_f = y_0 + R_{\max}\sin\theta, \end{aligned}$$

where $\mathrm{U}([a, b])$, $a, b \in \mathbb{R}$ is a random number uniformly distributed on $[a, b]$, $R_{\max}$ is the maximum cross range during the landing process, which is chosen to be $R_{\max} = 200\,\mathrm{km}$ in the experiments.

A total of 500 experimental cases were performed. DENMRA converged successfully for 68.0% of all cases when an affine initial guess was used. When the hierarchical trajectory generation approach was used to generate initial guesses, the convergence rate shot up to 99.0%, which is a significant improvement compared to those affine initial guesses.

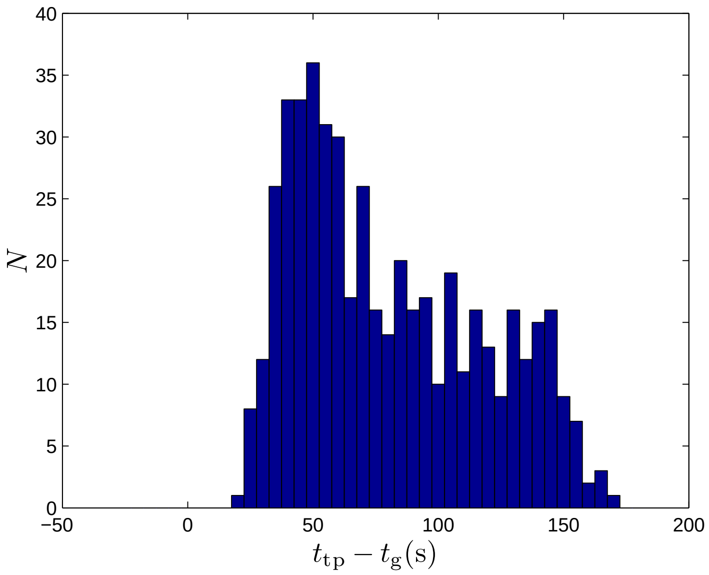

Figure 60: Optimality comparison: time-optimal tracking trajectory v.s. numerical optimization with TP initial guess.

Fig. 60 shows the final time differences of the TP trajectories (t_{tp}) and the corresponding DENMRA trajectories (t_{g}) obtained using initial guesses interpolating the TP trajectories for the 495 cases for which DENMRA converged. N represents the number of cases fitting in the $t_{\mathrm{tp}} - t_{\mathrm{g}}$ ranges corresponding to the blue bars. As expected, $t_{\mathrm{tp}} > t_{\mathrm{g}}$ for all test cases, since the DENMRA can improve both the landing path geometry and the time parameterization of the path for better performance (smaller t_{g}), while the TP trajectory is time-optimal only for a fixed Dubins' type geometric landing path generated in a heuristic way (close to optimal, but not even suboptimal). In other words, DENMRA may further improve the optimality of the initial guesses provided. The maximum value of $t_{\mathrm{tp}} - t_{\mathrm{g}}$ is 168.2 s, and the mean value of $t_{\mathrm{tp}} - t_{\mathrm{g}}$ is 80.6 s. Hence, in a real emergency landing scenario,

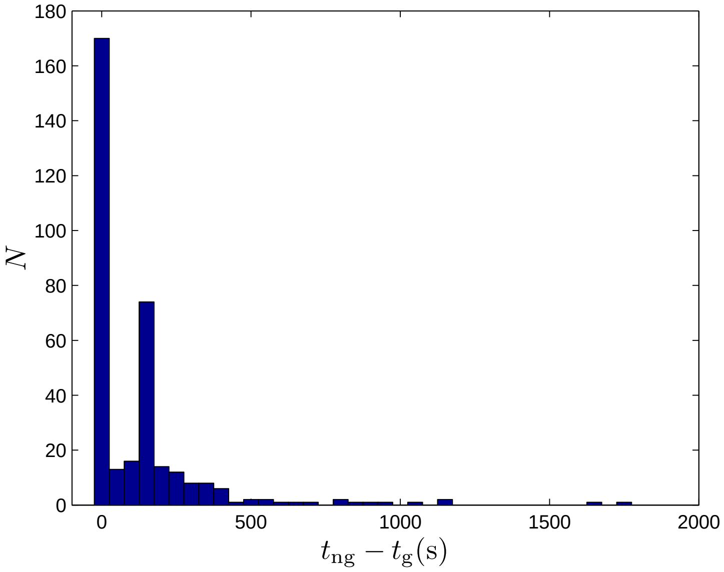

Figure 61: Optimality comparison: numerical optimization results, TP initial guesses v.s. affine initial guesses.

it might be worthwhile to use numerical optimization algorithm to further improve the optimality of the TP trajectory.

Fig. 61 compares the final times t_{g} of DENMRA trajectories generated using TP trajectory initial guesses and final times t_{ng} of DENMRA trajectories obtained using affine initial guesses. Among the 339 cases that DENMRA converged with both types of initial guesses, $-0.71 \leq t_{\mathrm{ng}} - t_{\mathrm{g}} \leq 1767.7$. The mean value of $t_{\mathrm{ng}} - t_{\mathrm{g}}$ is 123.4 s. It was frequently observed that the DENMRA converged to sub-optimal solutions with unacceptably long final times when the affine initial guesses were used, as shown in Fig. 61, which suggests that a simple affine function initial guess generation scheme is not applicable to trajectory generation for real emergency landing scenario.

CHAPTER VII

CASE STUDIES IN EMERGENCY LANDING TRAJECTORY OPTIMIZATION

This Chapter presents the study of two emergency landing cases, including the US Airways flight 1549 case and the Swissair flight 111 case, using the aircraft landing trajectory optimization algorithm introduced in Chapter 6. The purpose of this study is to evaluate the effectiveness of the proposed algorithm for emergency landing, and to characterize the critical factors which affect the success of landing. The aircraft maintains a certain amount of maneuverability for both cases: in the first case, the aircraft maintained full maneuverability for about 11 minutes after the onset of the emergency. In the second case, the thrust of the aircraft was lost, but other controls of the aircraft were operational during the whole flight. In an effort to identify valuable aircraft onboard decision aid tools for improving aviation safety, it would be informative to evaluate the pilots' decisions in terms of trajectory planning in these cases, and how different the outcome of each emergency could have been if a proper landing trajectory had be executed by the pilots.

For each of the two accidents, a flight trajectory is reconstructed based on recorded flight data, which are referred to as the *actual* landing trajectories. Furthermore, a sequence of points evenly distributed in the time domain were selected along each actual trajectory, and the corresponding minimum-time trajectories were computed using the proposed algorithm. Each of these minimum-time trajectories starts descending at a specific point on the actual landing trajectory, and ends up at either the final approach fix point or the runway of a nearby airport. The performance of the optimal trajectories were compared to those of the actual trajectories.

The purpose of this chapter is limited to testing the previously proposed trajectory planning algorithms, and demonstrating the potential of such algorithms for providing decision aid references to pilots and air traffic controllers. The current study is highly preliminary, and the presented results do not lead to any implication on the accidents considered or the previous investigation results. The assumptions and limitations of this study include, but not limited to, the following

1. The effect of wind is not addressed.

2. The actual flight trajectory used in this study are extracted from references [6, 4] and [95], which are different from the actual flight path.

3. The aerodynamics characteristics are assumed to be time-invariant, which is different from the real case when the aerodynamic characteristics are changed by the change of flap setting, elevator deflection, speed brake, and landing gear, etc.

4. It is assumed that the change of mass during the landing process is negligible.

5. For the Swissair 111 case, a maximum airspeed of 250 m/s is assumed. A maximum airspeed of 150 m/s is assumed for the US Airways 1549 case.

6. The dynamics model itself may not be accurate enough. Besides, the model parameters, such as aircraft mass, zero lift drag, and control bounds etc., may be different from those in the real scenario.

7.0.1 The Case of Swissair Flight 111

Swissair Flight 111 (SR-111) departed from John F. Kennedy International Airport in New York City on September 2, 1998, on a scheduled flight to Cointrin International Airport in Geneva, Switzerland. The aircraft type was a McDonnell Douglas MD-11. Due to a fire accident on board, the aircraft crashed into the Atlantic ocean Southwest of the Halifax International Airport.

The projection of the actual trajectory of flight SR-111 is shown in Fig. 62, which is reconstructed using the data in [6, 4]. A three-dimensional plot of the same trajectory is shown in Fig. 63. Because of unusual odor in the cockpit, the pilot declared a *Pan Pan* emergency at point “A” and attempted to go to Boston. Pan Pan is an expression, spoken three times in succession, used in the case of an urgency, which is a condition concerning the safety of the aircraft, or of some person on board or within sight, but which does not require immediate assistance. The pilot later took the advice of the air traffic controller and started approaching the Halifax International Airport. At point “D”, which is about 55.6 km away from the Halifax airport, the altitude of the aircraft was about 6620 m. Considering the altitude to be too high to land in Halifax, the pilot requested and was permitted to circle above the ocean to lose altitude and to dump excessive fuel. However, the fire condition onboard deteriorated rapidly. The fire first affected cockpit avionic systems at point “G” and caused autopilot disengagement, which was 590 seconds after the declaration of the emergency. The pilot declared “Mayday”, a first class emergency, at point “H” and requested immediate landing. However, the aircraft experienced a series of malfunctions immediately afterwards, finally crashing into the Atlantic ocean. Point “I” is generated from the last data received from the transponder of the aircraft. Note that the flight path following point “I” is not shown in the figure. The flight time between the point “A” and point “I” is 634 s.

A total of 11 points were selected along the actual SR-111 flight trajectory and used as the initial point of descent for the minimum-time landing trajectories. These points are evenly distributed temporally with 20 s between adjacent points. All time-optimal trajectories share the same final point, which is the final approach fix point of runway-6 of Halifax airport. This point is 5 nautical miles away from the runway. The initial speed, path angle, and heading angle are interpolated using the reconstructed actual trajectory data. The final speed at the approach fix point is 100 m/s, the final path angle is $-3°$, and the final heading angle is aligned with the runway direction. A maximum speed limit of 250 m/s and a minimum flight path angle of $-8°$ were assumed and enforced during the optimization. It was also assumed that the average aircraft’ speed between the final approach fix point and the runway touch down point is 80 m/s, which corresponds to a flight time of about 116.8 s between these two points. The effect of wind was not considered. The aircraft model used in this study is based on Ref. [3].

To help understand the benefits of using time-optimal trajectories for emergency landing, we introduce two time variables t_s and t_e for the analysis of the SR-111 flight case. The first time variable t_s is the time span between the start time of the time-optimal trajectory and the time corresponding to point "I" in Fig. 62. For example, the t_s value for point "A" would be 634 s. Because the aircraft experienced a series of functional downgrades after point "G", an optimal trajectory is considered to be feasible if by flying such a trajectory the aircraft arrives at the airport before the actual time corresponding to point "G". The second time variable t_e denotes the excessive amount of time, which is the time span between the touch down of the aircraft and the first functional downgrade of the aircraft (point "G"). Hence, a trajectory is considered viable if the associated t_e value is nonnegative, which means that the functional downgrade happens after touchdown, hence it does not affect airplane safety. Otherwise, the trajectory is considered to be nonviable.

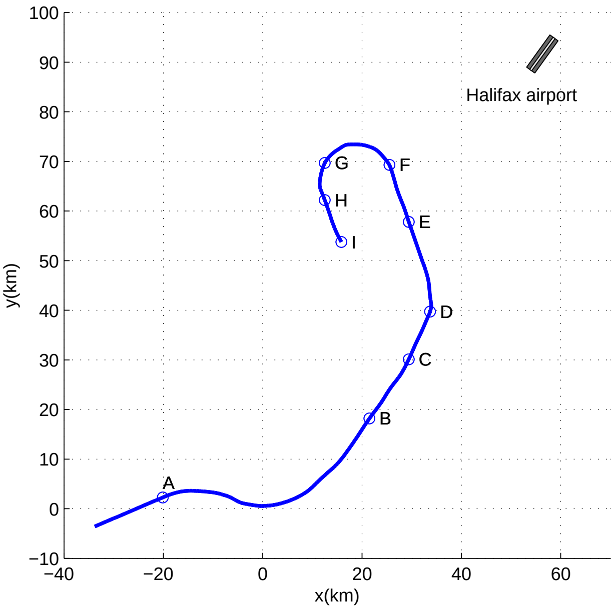

Figure 62: Swissair flight 111 trajectory, a top view.

The landing trajectory optimization algorithm proposed in this thesis converged for all 11 descent start points along the actual flight trajectory, and generated the corresponding time-optimal trajectories. The time-optimal landing trajectories connecting the actual flight path and the airport are shown in Figs. 64 and 65. In these figures, viable trajectories are plotted with green color, and nonviable trajectories are plotted with red color. According

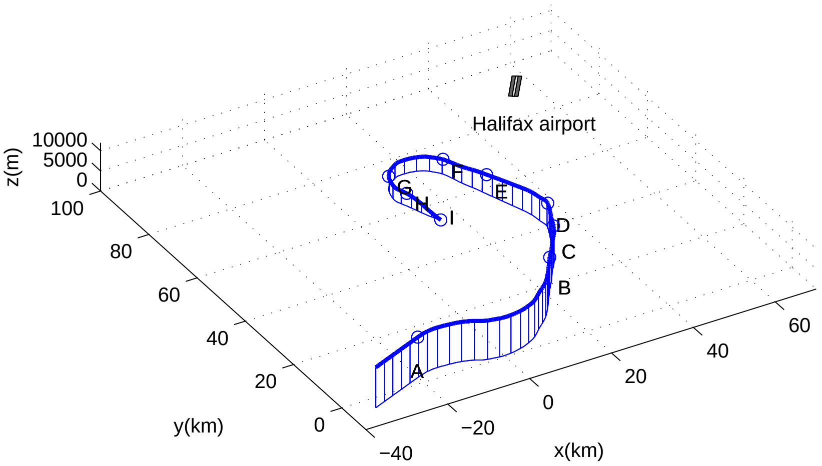

Figure 63: Swissair flight 111 trajectory, a bird view.

to the optimization result, after passing point “C”, the aircraft had been unable to land at the airport before the fire affected its maneuverability of the aircraft, even if a time-optimal trajectory has been pursued. As an example, the minimum-time state and control histories for the case with $t_s = 620$ are shown in Figs. 66-71.

Admittedly, due to imprecise actual flight trajectory data and the discrepancy between the true dynamics of the aircraft and the aircraft model used in the optimization, the optimization results may not be accurate enough, and are surely is not conclusive. However, these results suggest, at least, the possibility that the outcome of this fatal aviation accident may had been reversed if a time-optimal flight trajectory was pursued early enough.

The excessive times t_e for different descent start times t_s are shown in Fig. 72. Note that, by definition, smaller t_s means flying along the actual flight trajectory longer before performing a time-optimal descend. If the actual flight trajectory and the aircraft dynamics model are accurate enough, then t_e should decrease monotonically as t_s decreases. To see this, consider two time-optimal descend start times t_{s_1} and t_{s_2} with $t_{s_1} > t_{s_2}$, and let t_{e_1} and t_{e_2} be the associated excessive times, respectively. If the aircraft starts an optimal descent at t_{s_2}, then between t_{s_1} and t_{s_2}, the aircraft tracks the actual flight trajectory, which is not optimal and results in a longer flight time as compared to the time-optimal

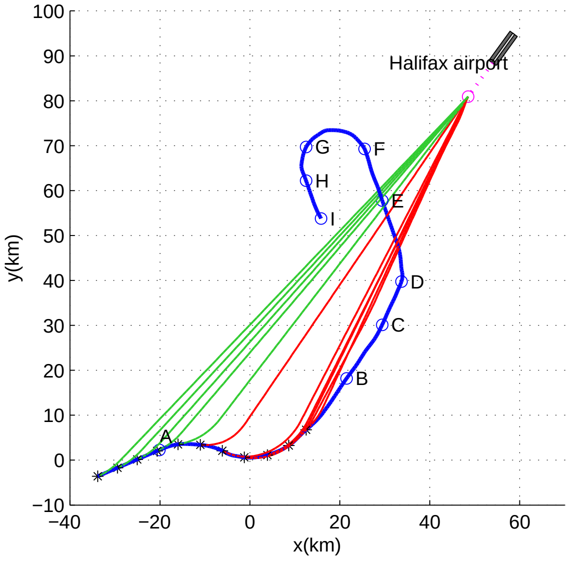

Figure 64: Minimum-time trajectories along the Swissair flight 111 trajectory, a top view.

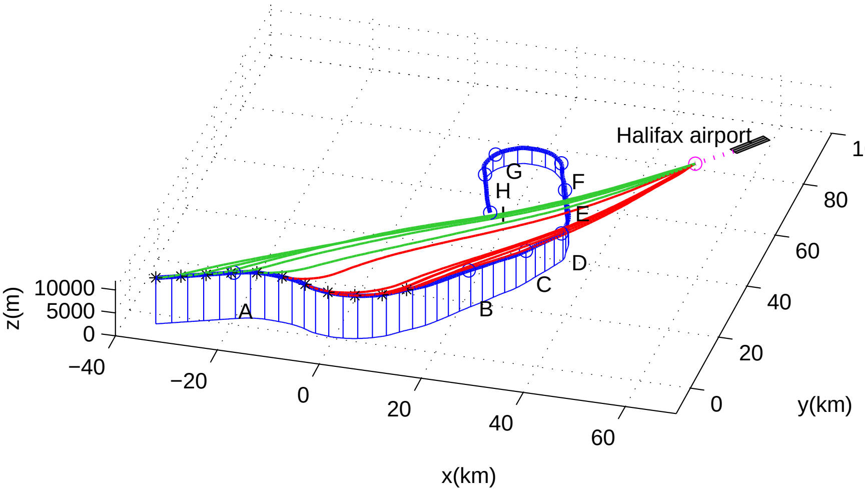

Figure 65: Minimum-time trajectories along the Swissair flight 111 trajectory, a bird view.

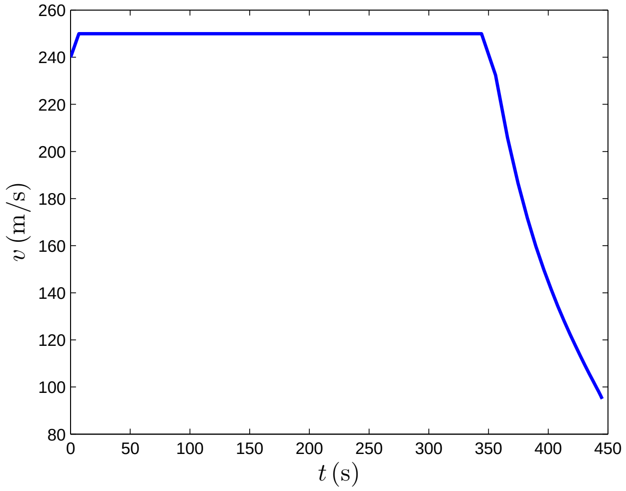

Figure 66: Minimum-time speed profile, $t_s = 620$.

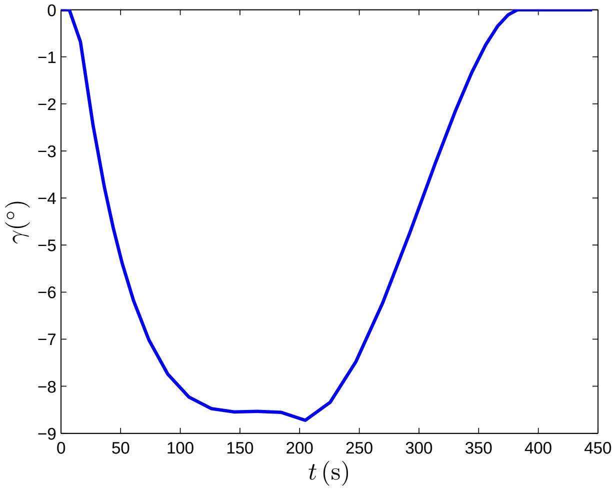

Figure 67: Minimum-time path angle profile, $t_s = 620$.

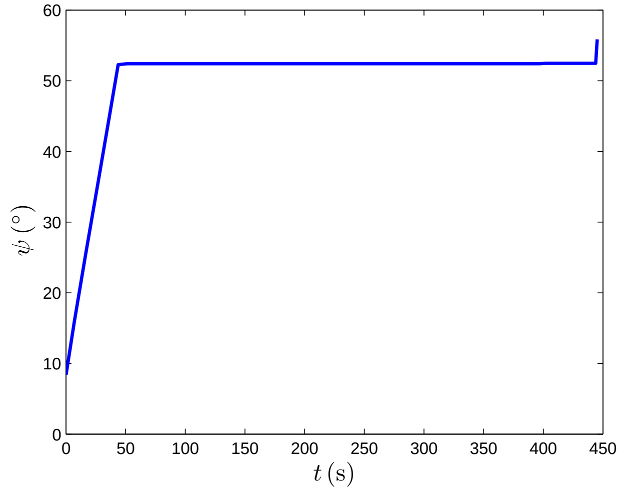

Figure 68: Minimum-time heading angle profile, $t_s = 620$.

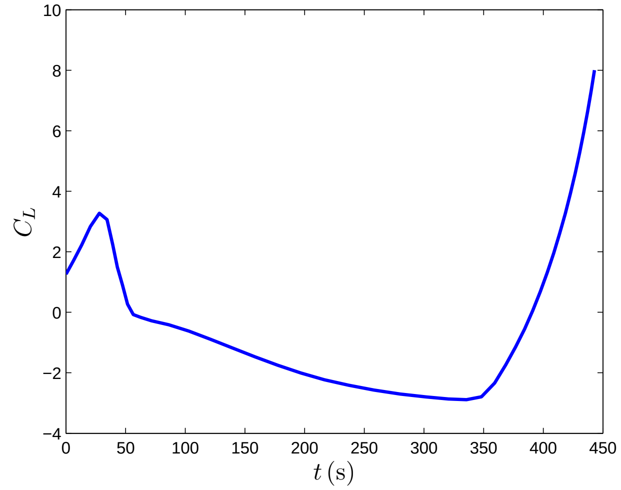

Figure 69: Minimum-time lift coefficient profile, $t_s = 620$.

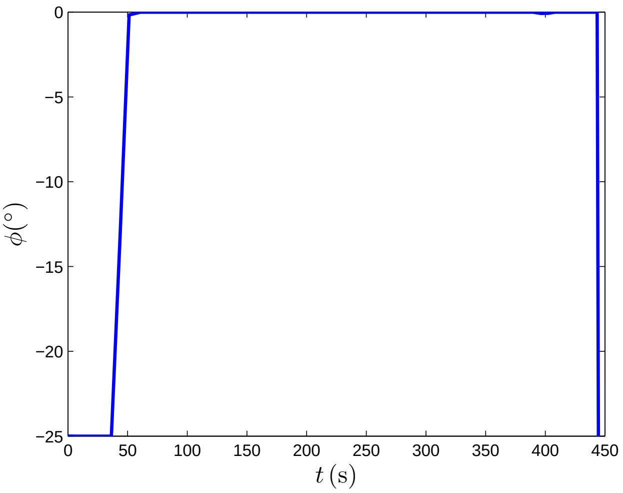

Figure 70: Minimum-time bank angle profile, $t_s = 620$.

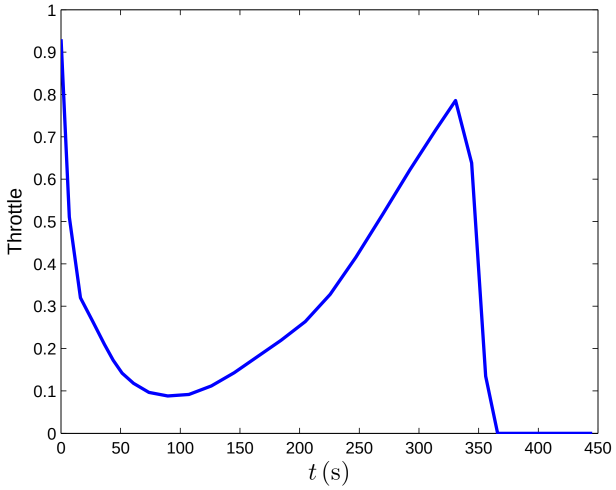

Figure 71: Minimum-time throttle profile, $t_s = 620$.

trajectory starting from t_{s_1}. Therefore, the excessive time t_{e_2} is smaller than t_{s_1}. As shown in Fig. 72, such a monotonicity is not perfectly maintained, which can be caused by various reasons such as position errors in the actual flight path data or imprecise model. However, the overall trend of the $t_s - t_e$ curve is acceptable. It can be seen from this figure that t_e decreases most rapidly around $t_e = 600\,\mathrm{s}$, which happened after the pilot declared a Pan Pan emergency and initiated a right turn to go back to Boston. t_e decreased by about 50 seconds during this incomplete turning maneuver and became negative. Therefore, after the first right turn around $t_s = 600$s, the chance of a safe landing became very slim. The earliest possible landing time is also estimated in [6], which corresponds to $t_s = 683\,\mathrm{s}$ and $t_e = 0$. Such an estimation assumes direct tracking to the Halifax Golf beacon. As a comparison, a time-optimal trajectory starting from the same point provides an excessive time of $t_e = 27.6\,\mathrm{s}$, which suggests that about half a minute might be saved compared to a conventional emergency descent procedure if a time-optimal descent is initiated at this point.

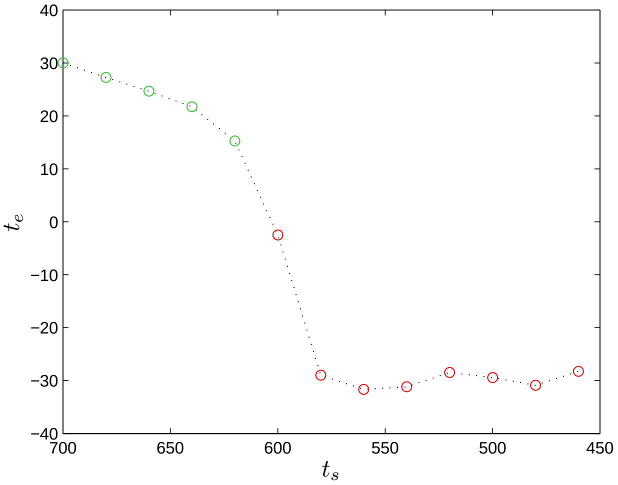

Figure 72: The excessive time t_e v.s. the start time t_s of optimal landing trajectories.

The above analysis suggests that if real-time landing trajectory optimization technologies were available to pilots and air traffic controllers in the future in assistance of landing site selection and landing trajectory generation, they may help reduce the probability of adverse outcomes of emergencies scenarios and improve aviation safety. Landing trajectory optimization algorithms, such as the hierarchical algorithm introduced in this thesis, may provide the desired capabilities and help with the safe landing of aircraft under abnormal conditions.

7.0.2 The Case of US Airways Flight 1549

US Airways Flight 1549 (US-1549) was a A320 aircraft on a domestic flight from New York City's La Guardia Airport (LGA) to Charlotte/Douglas (CLT), North Carolina, on January 15, 2009. La Guardia Airport has two runways perpendicular to each other. The length and width of these runways are almost the same. Two numbers are assigned to each runway for different landing/take-off directions, hence there are four runway labels, which are illustrated in Fig. 73. Flight US-1549 took off from runway-4 and headed Northwest. About two minutes after take-off, the aircraft collided with birds and lost thrust on all engines. The captain, who happened to be an experienced glider pilot, maintained control of the aircraft and successfully performed a ditch landing on the Hudson river. The flight path of the aircraft is also shown in Fig. 73 (the trajectory data is from [95]). The red circle on the flight path corresponds to the point where the aircraft collided with birds and lost thrust.

The US-1549 flight emergency landing case was also used to test the performance of the proposed landing trajectory optimization algorithm. Similar to the SR-111 flight case, time-optimal landing trajectories were computed for different start points along the actual flight trajectory with different start times and corresponding initial positions along the trajectory. Because the aircraft does not have any thrust, the normal landing procedures do not apply to this case, and the final approach fix point is not used in the trajectory optimization. The final condition of the aircraft is chosen such that the aircraft is aligned with the runway with a touch-down speed between 70-85 m/s. Landing scenarios for the four runways were considered separately, and the effect of wind was not taken into account. The aerodynamic data of A320 aircraft in this study obtained using a min-square fitting of the aerodynamic data table in Ref. [1].

If the nonlinear optimization solver converges and generates a trajectory satisfying the specified feasibility (10^{-5}) and optimality (10^{-4}) tolerances, then the trajectory is considered to be optimal. The solver may not converge either because there does not exist a glider landing trajectory given the specified boundary conditions and constraints, which is very common for this zero-thrust landing trajectory optimization problems, or because of numerical difficulties of the NLP solver itself. The feasibility of a landing path for a glider is very sensitive to the path geometry due to the lack of thrust control. For this reason, paths generated by the geometric path planner are often infeasible. As a result, the quality of the initial guess to the numerical optimal control algorithm is not good enough, which may affect the convergence of the NLP solver. In this study, if the NLP solver does not converge for certain landing cases, then the geometric path planner is modified to adjust the geometry of the initial guess. If no convergent solution can be found in this way, then a zero-thrust landing would be considered impossible for this case.

The time-optimal trajectories for four different cases are shown in Figs. 74-77. In these figures, the green circles represent the latest time and the corresponding position of the aircraft such that the aircraft can still land on a particular runway by performing a time-optimal landing starting from that point. For a runway-4 landing, the latest time for starting time-optimal descent is 21 seconds after the bird-strike. For runway-13, runway-31, runway-22, these values are 123 s, 65 s, and 73 s, respectively.

As shown in Figs. 74-77, the zero-thrust time-optimal emergency landing trajectories are much more complicated than the actual flight trajectory. Therefore, even if all four runways are available for landing, the pilot must choose a runway and respond fast enough such that the time-optimal descent would start early enough before those green points in these figures. Besides, the execution of such trajectories poses very high demands on the skill and attention of the pilots, since any deviation from the optimal trajectory may result in an accident. Hence, the pilot's choice of the Hudson River as the landing site was practically a much safer choice than other alternative plans such as landing at any runway of LGA.

On the other hand, if the same emergency happened in an airport without a convenient ditch landing site like the Hudson River, the outcome of the accident might be much severer.

As demonstrated by the optimization result, the choice of runway has a major influence on the emergency landing process. Taking the runway layout of LGA and the US-1539 flight trajectory as an example, a longer time span between the red and green circles in Figs. 74-77 would be favorable, in the sense that the pilot would have more time to plan an emergency descent trajectory. Besides, the earlier the descent start time within this time span, the better the chance of a successful landing. Hence, under the assumption that the LGA runways are the only possible landing sites, it is clear that runway-31 was the best choice for emergency landing in this case, as suggested by the trajectory planning results obtained using the landing trajectory optimization algorithm proposed in this thesis.

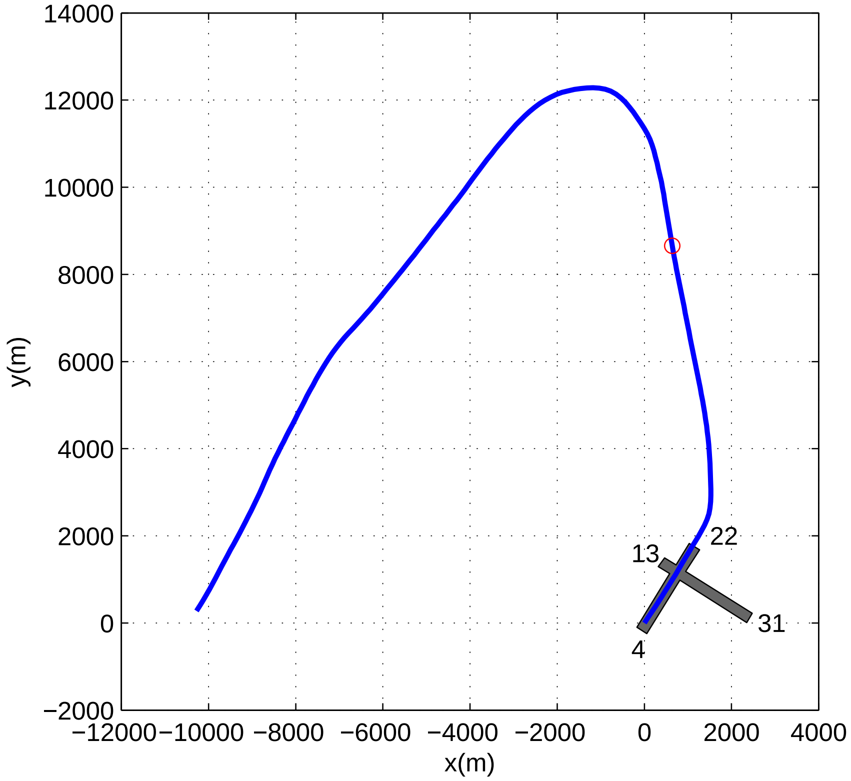

Figure 73: US-1549 Hudson River landing trajectory, a top view.

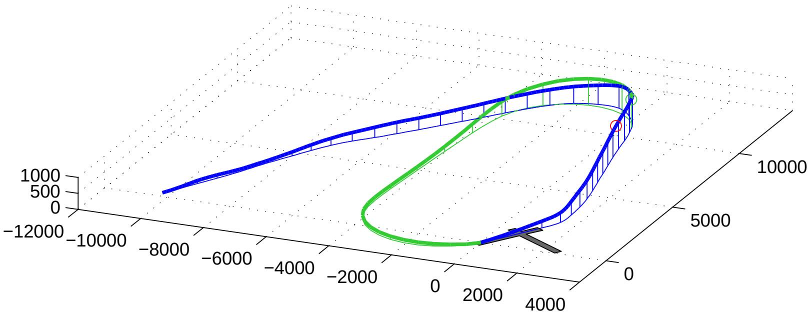

Figure 74: US 1549 Runway-4 landing.

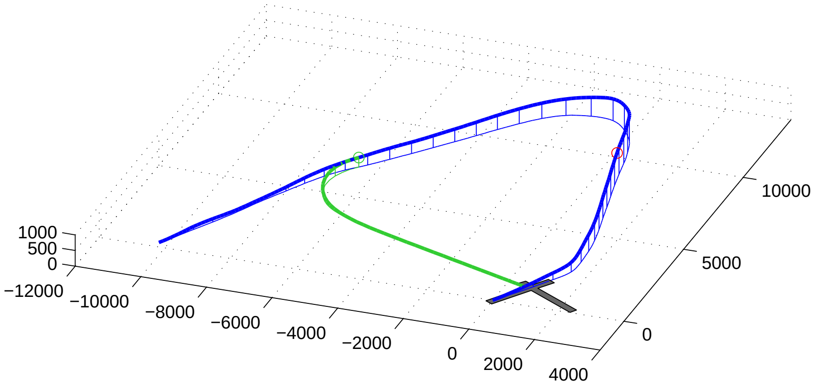

Figure 75: US 1549 Runway-31 landing

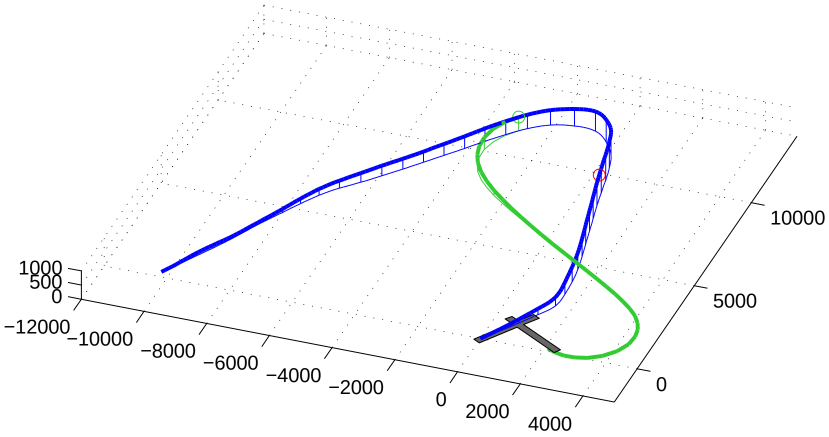

Figure 76: US 1549 Runway-13 landing.

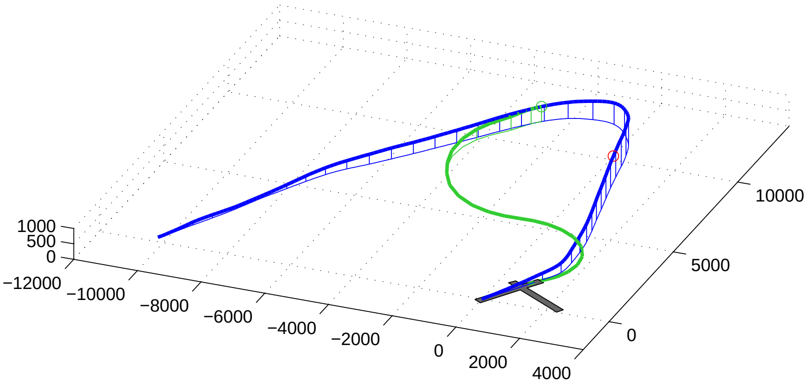

Figure 77: US 1549 Runway-22 landing.

CHAPTER VIII

CONCLUSIONS

In this thesis, we have addressed the aircraft landing trajectory optimization problem, and have presented several computationally efficient and numerically robust techniques for the landing trajectory generation and optimization problem for fixed-wing aircraft. We also introduced a hierarchical scheme based on these techniques for generating good initial guesses for numerical optimal control algorithms, thus, further improving the convergence of the landing trajectory. Although presented in the context of aircraft landing trajectory optimization, the techniques introduced in this thesis can be adapted to solve trajectory optimization problems for other types of electro-mechanical systems, such as locomotive, ground vehicle, elevator, multi-axis machinery, etc.

In this chapter, we will first summarize the techniques presented in this thesis for landing trajectory optimization. Then we will discuss some directions for future research.

8.1 Summary

In Chapter 2, we focused on the problem of mesh refinement for generating a grid in the time domain, on which a continuous time optimal control problem is discretized into a Nonlinear Programming problem. A density function technique was proposed to automatically allocate a fixed number of grid points with the appropriate local temporal resolution to better capture the discontinuities and smoothness irregularities in the solution. It was shown that the problem of mesh generation can be converted equivalently into choosing the density function, and some of the previous mesh refinement methods correspond to certain specific forms of density functions. We also presented a curvature density function for mesh generation, which minimizes the L^1 norm of the error in approximating two-dimensional C^3 curves (or piecewise C^3 curves) using piecewise linear interpolative spline curves. A density function based mesh refinement algorithm was developed and tested using a number of challenging optimal control problems. Numerical results verified the improvement in solution accuracy and optimality by using such a method when compared to existing mesh refinement schemes. A major benefit of introducing the density function is that the improvement of solution accuracy and optimality can be achieved without increasing the number of grid points, hence keeping the size of the Nonlinear Programming problem invariant in the mesh refinement iterations, which is a favorable feature for trajectory optimization applications with limited computation resource.

In Chapter 3, we introduced an iterative Quadratic Programming algorithm for the smoothing of a three-dimensional geometric path by minimizing a weighted L^2 norm of the path's curvature. The path is represented by a number of characteristic nodes and a cubic spline curve interpolation between the nodes. In each iteration, a smoother path is obtained by perturbing the path from the previous iteration in its neighborhood along the normal direction only. The path smoothing method can also address requirements for collision avoidance, path length preservation, local bounds on curvature, etc., by incorporating corresponding constraints. It was demonstrated, using numerical examples, that a smoothed path obtained using the introduced algorithm can provide a shorter tracking time for the aircraft dynamics as compared to the original path. This algorithm is also computationally efficient and reliable.

In Chapter 4, we considered the problem of time-optimal tracking of a prescribed geometric path for fixed-wing aircraft, and a semi-analytic solution to this problem was presented. By transforming the aircraft dynamics, which is originally established in the time domain, into the path coordinate domain and applying inverse dynamics, the constraints on the bank angle and lift coefficient control inputs are converted into simple bounds on the kinetic energy of the aircraft. As a result, the time-optimal path tracking problem is reduced to an equivalent scalar functional optimization problem subject to state constraints, thrust control input constraint, and boundary conditions. The switching structure of the optimal solution to the simplified problem was analyzed using optimal control theory. Two algorithms were developed for computing the time-optimal kinetic energy and the corresponding optimal state and control time histories. Two numerical examples verified the proposed method.

In Chapter 5, we addressed the problem of the energy-optimal aircraft landing path tracking with fixed time-of-arrival (TOA). A reduced problem was formulated in a similar way as in Chapter 4. It was shown that the optimal solution to the energy-optimal problem may contain singular arcs, which typically cause numerical issues for standard numerical optimization approaches. Due to the existence of singular arcs, the energy-optimal solution may exhibit more complicated switching structures than the time-optimal solution. Based on optimal control theory, we analyzed the switching structure in the energy-optimal kinetic energy solution. We also introduced a technique for characterizing state constrained arcs in the optimal kinetic energy profile via a partial relaxation of the state constraints. It was proved that the energy-optimal solution can be computed in a straightforward manner using a combination of a singular arc, the minimum-time solution, and a maximum-time solution (which is a variation of the time-optimal path tracking problem considered in Chapter 4 with maximum tracking time performance). A numerical algorithm was introduced for computing the energy-optimal path tracking solution. As shown by both the theoretical analysis and the numerical optimization results, the energy-optimal solution provides a good approximation to the fuel-optimal solution during the landing process, but is much more computationally efficient than the use of a direct numerical optimization approach.

In Chapter 6, we proposed a hierarchical scheme for the generation of good initial guesses based on the techniques introduced in Chapters 3 and 4. We presented a Dubins-like three dimensional path generation method, which is designed specifically for the hierarchical initial guess generation scheme. Such a hierarchical scheme is integrated into an aircraft landing trajectory optimization algorithm to provide the initial guess for the density function based numerical optimal control routine. The robustness of the overall landing trajectory algorithm was tested using numerical simulations. It was shown that with such a hierarchical approach, the failure rate of the Nonlinear Programming solver was reduced from 51% to less than 1%.

Finally, in Chapter 7, we applied the landing trajectory optimization algorithm to analyze two real emergency landing cases including Swissair flight 111 and US Airways flight

1549. The analysis showed that human pilots may make sub-optimal or even faulty trajectory planning decisions during an emergency, which can be addressed by automated landing trajectory optimization algorithms.

8.2 Challenges for the Application of Onboard Flight Trajectory Optimization

The application of automatic onboard trajectory optimization tools in commercial and general aviation is inevitably a complex and difficult process that involves and affects multiple organizations, agencies, companies, and individuals. Besides, the challenges go beyond the technology aspect to administration, legislation, licensing, training, etc., which are out of the scope of this thesis. In the following, we briefly discuss about the status of the technology for such an application.

On the technology level, the development and application of onboard trajectory optimization techniques bring up numerous challenges in algorithm accuracy, efficiency and reliability, software robustness, hardware development, human-machine interface, system integration, system compatibility, flight testing, etc.

Despite these technical challenges and difficulties, preliminary research and flight tests has been been performed, which leads to promising results. For example, Ref. [125] developed online flight trajectory optimization algorithms based on a combination of A^* path search algorithm with a random tabu search method. The optimized path is presented as tunnel-in-the-sky image to the pilot for tracking. The algorithm was verified using flight simulator and tested in real flight tests. The flight test results showed that the algorithm can help reduce the flight time to reach the terminal point. The same research group also proposed an online trajectory optimization algorithm by using a direct collocation method for generating more smooth trajectories [107]. Because direct collocation method takes more computation time, the total flight trajectory is divided into multiple segments and a receding horizon scheme is applied such that the optimization of each segment can be finished before it is tracked. A Dubins' type of landing trajectory is used as an initial guess for the direct collocation method. Flight test results demonstrated that the optimized 4D landing trajectories, which were generated off-line, can be tracked satisfactorily by the tracking controller. Ref. [128] tested a trajectory optimization algorithm for ground noise abatement in helicopter landing approach using simulation. The optimized trajectories were flied in flight test, and the flight test results confirmed a reduction of mean noise level between 6-10dB.

Considering the progress of onboard landing trajectory optimization as reported in Refs. [125, 107, 128], the main technical challenges to be addressed in the near future include algorithm efficiency, algorithm reliability, human-machine interface, and the development of trajectory tracking controller. The direct collocation method is used in Refs. [107] and [128] in favor of its capability for dealing with more realistic aircraft dynamics. However, in these references, the trajectories are optimized off-line, and tested by simulations before real flight test. The online and realtime application of trajectory optimization algorithms impose very high demand on computation speed and reliability. Because the optimal trajectories are in general more complicated than traditional flight trajectories, proper human-machine interface need to be developed such that the optimization result can be presented to the pilot both efficiently and effectively. It would also be necessary to improve the accuracy and tracking of tracking controllers for the automatic tracking of optimal 4D trajectories such that the pilots can be relieved from exhaustive trajectory tracking and focus on higher level tasks.

8.3 Directions of Future Work

In this section, we present some directions of future research as extensions to the results in this thesis.

8.3.1 Interaction Between Path Smoothing and Time Parameterization Methods

In Chapter 6, we presented a hierarchical scheme for generating a feasible initial guess. In this scheme, a geometric path primitive is converted to a trajectory using the path tracking methods in Chapters 4 and 5. If the generated trajectory is not feasible, then the path smoothing method in Chapter 3 is applied to update the path such that the new path is more likely to be feasible. Although such an approach will typically work well in practice, it does not provide the best path geometry for the given aircraft dynamics and boundary conditions.

The time parameterization methods in Chapters 4 and 5 provide rich information about the influence of the path geometry on tracking performance. For example, when a path is infeasible for the prescribed dynamics and control constraints, based on the time parameterization result, we can tell which constraint is causing infeasibility, and what are the geometric characteristics associated to the activation of these constraints. It was observed in our numerical experiments that the infeasibility of a path primitive was frequently caused by inappropriate geometry along a small part along the path. In this thesis, such problems are addressed by manually tuning several geometry-related parameters in the Dubins-like geometric path planning algorithm based on the time parameterization result. Such an approach is not only inefficient, but also results in further loss of optimality due to the conservative parameter setting.

The three-dimensional path smoothing method, on the other hand, is capable of regulating certain geometric properties of the path such as local curvature and path length. However, these capabilities are not used in the current aircraft landing trajectory optimization algorithm. Instead, the path smoothing algorithm minimizes the norm of the curvature only, and is blind to the system dynamics. By converting the time parameterization result into desirable geometric features for recovering feasibility or further improving the optimality of tracking result, the path smoothing method can address the system dynamics based on feedback information from the time parameterization step, and thus produce paths with "close to optimal" geometry.

8.3.2 Path Tracking via Convex Optimization

The path tracking methods in Chapter 4 and Chapter 5 are based on semi-analytic solutions. The advantage of such an analytic approach is superior computation speed, accuracy, and algorithm robustness, compared to the standard NLP approach. However, it is extremely laborious to extend these methods to other types of dynamical systems, or to address new cost functions.

Convex Programming (CP) problems can be solved much more efficiently and reliably than general NLP problems, Hence, convex optimization techniques seem to be a good choice for solving path tracking problems. Indeed, it has been used for the time-optimal tracking control of robotic manipulators [130]. However, the application of CP imposes very strict requirements on the problem to be solved, i.e., both the cost function and the feasible region determined by all constraints must be convex. These are unrealistic assumptions for most physical systems. Hence, in order to fit into the CP framework for its robustness and efficiency, the problematic terms in the system dynamics, which cause the loss of convexity, are neglected. For example, the friction term of the dynamics in the previous research on time-optimal tracking control of robotic manipulators using CP have been omitted. Such treatment on these terms causes the loss of optimality since the solution is obtained for a different system.

To avoid such loss of optimality, *iterative* convex optimization can be applied by finding a series of piecewise constant or piecewise linear approximations to the terms in the dynamics which causes loss of convexity, and solve at each iteration a new CP problem, which is an improved approximation to the original non-convex problem. The solution to the iterative CP problem, if converges, is also the solution to the optimal solution (at least a locally) to the original non-convex problem.

Furthermore, an alternating CP approach can be applied to trajectory optimization. With such an approach, the solution to a non-convex problem is obtained by solving two or more subproblems alternatively. Each subproblem, which is convex by formulation, updates only a subset of the decision variables in the original non-convex problem, while keeping the other variables constant. By alteratively solving all subproblems, all decision variables to the non-convex problem are updated. Note that the landing trajectory optimization problem can be decomposed into two planning tasks including the planning of path geometry and the optimal tracking task. The path smoothing method in Chapter 3 solves a CP (Quadratic Programming) problem at each step for the modification of the path geometry. By formulating the path tracking task as a convex optimization problem, the overall trajectory can be optimized by solving the path smoothing and tracking problems alternatively.

8.3.3 Trajectory Generation in a Environment with Moving Obstacles

When performing flight path and trajectory planning tasks, pilots and air traffic controllers sometimes need to deal with moving obstacles such as other aerial vehicle or severe weather patterns. Such tasks are more complicated than the usual trajectory planning for a single airplane because more time-varying objects are involved in the decision making, and the generation of collision-free trajectories in such a time-evolving environment requires extensive projection and correction.

With valid mathematical models, computers can predict the evolution of dynamical systems much more efficiently and accurately than human beings, and perform well on tasks too complicated, and even impossible for human beings such as weather forecast. Hence, it is expected that computers may also play an important role in flight path and trajectory generation with moving obstacles, and here we propose one possible solution for such an application for future exploration.

The mathematical formulation of the collision avoidance requirement is not convex in general, which induces difficulty for the convergence of the NLP based numerical optimal control approach. However, by updating the path and trajectory only locally (both temporally and spatially), it is possible to enforce the collision avoidance requirement approximately using convex constraints, as in the path smoothing algorithm, therefore, the NLP approach can still be applied. The key step would be the generation of a collision-free feasible trajectory such that a repeated local update can be initiated from its neighborhood. Reference [50] proposed trajectory planning algorithms dealing with moving obstacles and simple vehicle dynamics, which can be used to generate an initial collision-free trajectory. Next,hierarchical trajectory generation scheme based on the path smoothing method and time-parameterization method in this thesis can be applied to refine such a trajectory using more accurate aircraft dynamics. Finally, the NLP based numerical optimal control approach can be applied to further improve the optimality by updating the trajectory locally.

8.3.4 Three-Dimensional Collision-Free Path Primitive Generation

In Chapter 6, we introduced a three-dimensional landing path primitive generation algorithm. Although this method is very efficient, and the result is close to optimal, it does not address the problem of collision avoidance. In practice, collision-free paths are necessary for bypassing severe weather conditions, obstacles, or traffic patterns. Hence, it is necessary to consider the problem of collision-free path generation in the three-dimensional space. Because the path-smoothing algorithm in Chapter 3 can be applied to smooth any collision-free path primitive, while preserving the collision avoidance property, no smoothness is required for the generation of collision-free path primitives, and the main focus would be on the robustness and the speed of computation.

APPENDIX A

COMPUTATION OF MATRICES FOR PATH SMOOTHING

This appendix provides the pseudo code for the computation of matrices **H**, **F** and **C** used in the path smoothing method introduced in Chapter 3.

Pseudo-code 1 Computation of matrices **H** and **F**

```
$\mathbf{H}_\delta \leftarrow \mathbf{0}^{N\times N}$, $\mathbf{H}_\lambda \leftarrow \mathbf{0}^{N\times N}$, $\mathbf{H}_{\delta\lambda} \leftarrow \mathbf{0}^{N\times N}$, $\mathbf{F}_\delta \leftarrow \mathbf{0}^{1\times N}$, $\mathbf{F}_\lambda \leftarrow \mathbf{0}^{1\times N}$;
for $i = 1$ to $N-1$ do
  $H_{\delta,i} \leftarrow \mathbf{0}^{4\times 4}$, $H_{\lambda,i} \leftarrow \mathbf{0}^{4\times 4}$, $H_{\delta\lambda,i} \leftarrow \mathbf{0}^{4\times 4}$, $F_i \leftarrow \mathbf{0}^{1\times 4}$;
  for $k = 1$ to $3$ do
    $H_{\delta,i} \leftarrow H_{\delta,i} + \mathbf{N}_{i,\{k\}}\mathbf{M}_i\mathbf{N}_{i,\{k\}}$, $H_{\lambda,i} \leftarrow H_{\lambda,i} + \mathbf{B}_{i,\{k\}}\mathbf{M}_i\mathbf{B}_{i,\{k\}}$;
    $H_{\lambda\delta,i} \leftarrow H_{\lambda\delta,i} + \mathbf{B}_{i,\{k\}}\mathbf{M}_i\mathbf{N}_{i,\{k\}}$;
    $F_{i,\delta} \leftarrow F_{i,\delta} + \mathbf{R}_i^{\mathrm{T}}\{k\}\mathbf{M}_i\mathbf{N}_{i,\{k\}}$, $F_{i,\lambda} \leftarrow F_{i,\lambda} + \mathbf{R}_i^{\mathrm{T}}\{k\}\mathbf{M}_i\mathbf{B}_{i,\{k\}}$;
    $k \leftarrow k+1$;
  end for
  $\mathbf{H}_\delta(i-1:i+2, i-1:i+2) \leftarrow \mathbf{H}_\delta(i-1:i+2) + w_i H_{\delta,i}$;
  $\mathbf{H}_\lambda(i-1:i+2, i-1:i+2) \leftarrow \mathbf{H}_\lambda(i-1:i+2) + w_i H_{\lambda,i}$;
  $\mathbf{H}_{\delta\lambda}(i-1:i+2, i-1:i+2) \leftarrow \mathbf{H}_{\delta\lambda}(i-1:i+2) + w_i H_{\delta\lambda,i}$;
  $\mathbf{F}_\delta(1, i-1:i+2) \leftarrow \mathbf{F}_\delta(1, i-1:i+2) + w_i F_{\delta,i}$;
  $\mathbf{F}_\lambda(1, i-1:i+2) \leftarrow \mathbf{F}_\lambda(1, i-1:i+2) + w_i F_{\lambda,i}$;
  $i \leftarrow i+1$;
end for
$\mathbf{H}_\delta(1:4, 1:4) \leftarrow \mathbf{H}_\delta((1:4, 1:4) + w_1\sum_{k=1}^{3}\mathbf{N}_{2,\{k\}}\mathbf{M}_1\mathbf{N}_{2,\{k\}}$;
$\mathbf{H}_\lambda(1:4, 1:4) \leftarrow \mathbf{H}_\lambda(1:4, 1:4) + w_1\sum_{k=1}^{3}\mathbf{B}_{2,\{k\}}\mathbf{M}_1\mathbf{B}_{2,\{k\}}$;
$\mathbf{H}_{\delta\lambda}(1:4, 1:4) \leftarrow \mathbf{H}_{\delta\lambda}(1:4, 1:4) + w_1\sum_{k=1}^{3}\mathbf{N}_{2,\{k\}}\mathbf{M}_1\mathbf{B}_{2,\{k\}}$;
$\mathbf{H}_\delta(N-3:N, N-3:N) \leftarrow \mathbf{H}_\delta(N-3:N, N-3:N) + w_{N-1}\sum_{k=1}^{3}\mathbf{R}_{N-2,\{k\}}\mathbf{M}_{N-1}\mathbf{N}_{N-2,\{k\}}$;
$\mathbf{H}_\lambda(N-3:N, N-3:N) \leftarrow \mathbf{H}_\lambda(N-3:N, N-3:N) + w_{N-1}\sum_{k=1}^{3}\mathbf{R}_{N-2,\{k\}}\mathbf{M}_{N-1}\mathbf{N}_{N-2,\{k\}}$;
$\mathbf{H}_{\delta\lambda}(N-3:N, N-3:N) \leftarrow \mathbf{H}_{\delta\lambda}(N-3:N, N-3:N) + w_{N-1}\sum_{k=1}^{3}\mathbf{R}_{N-2,\{k\}}\mathbf{M}_{N-1}\mathbf{N}_{N-2,\{k\}}$;
$\mathbf{H} = \begin{bmatrix}\mathbf{H}_\delta & \mathbf{H}_{\delta\lambda}\\ \mathbf{H}_{\delta\lambda}^{\mathrm{T}} & \mathbf{H}_\lambda\end{bmatrix}$;
$\mathbf{F} = [\mathbf{F}_\delta, \mathbf{F}_\lambda]$;
$\mathbf{H} = 2\mathbf{H}$;
$\mathbf{F} = 2\mathbf{F}$.
```

Pseudo-code 2 Computation of matrix **C**

```
$\mathbf{C} \leftarrow \mathbf{0}^{N\times 2N}$, $\mathbf{C}_\delta \leftarrow \mathbf{0}^{N\times N}$, $\mathbf{C}_\lambda \leftarrow \mathbf{0}^{N\times N}$;
$\mathbf{C}_\delta(1,:) \leftarrow \frac{1}{\Delta_s^2}\begin{bmatrix}0 & -5\langle \mathbf{n}_1, \mathbf{n}_2\rangle & 4\langle \mathbf{n}_1, \mathbf{n}_3\rangle & -\langle \mathbf{n}_1, \mathbf{n}_4\rangle & 0 & \dots & 0\end{bmatrix}$;
$\mathbf{C}_\lambda(1,:) \leftarrow \frac{1}{\Delta_s^2}\begin{bmatrix}0 & -5\langle \mathbf{n}_1, \mathbf{b}_2\rangle & 4\langle \mathbf{n}_1, \mathbf{b}_3\rangle & -\langle \mathbf{n}_1, \mathbf{b}_4\rangle & 0 & \dots & 0\end{bmatrix}$;
for $i = 2$ to $N-1$ do
    $\mathbf{C}_\delta(i, i-1) \leftarrow \langle \mathbf{n}_i, \mathbf{n}_{i-1}\rangle/\Delta_s^2$, $\mathbf{C}_\lambda(i, i-1) \leftarrow \langle \mathbf{n}_i, \mathbf{b}_{i-1}\rangle/\Delta_s^2$;
    $\mathbf{C}_\delta(i, i) \leftarrow -2/\Delta_s^2$;
    $\mathbf{C}_\delta(i, i+1) \leftarrow \langle \mathbf{n}_i, \mathbf{n}_{i+1}\rangle/\Delta_s^2$, $\mathbf{C}_\lambda(i, i+1) \leftarrow \langle \mathbf{n}_i, \mathbf{b}_{i+1}\rangle/\Delta_s^2$;
    $i \leftarrow i+1$;
end for
$\mathbf{C}_\delta(N,:) \leftarrow \frac{1}{\Delta_s^2}\begin{bmatrix}0 & \dots & 0 & -\langle \mathbf{n}_N, \mathbf{n}_{N-3}\rangle & 4\langle \mathbf{n}_N, \mathbf{n}_{N-2}\rangle & -\langle \mathbf{n}_N, \mathbf{n}_{N-1}\rangle & 0\end{bmatrix}$;
$\mathbf{C}_\lambda(N,:) \leftarrow \frac{1}{\Delta_s^2}\begin{bmatrix}0 & \dots & 0 & -\langle \mathbf{n}_N, \mathbf{b}_{N-3}\rangle & 4\langle \mathbf{n}_N, \mathbf{b}_{N-2}\rangle & -\langle \mathbf{n}_N, \mathbf{b}_{N-1}\rangle & 0\end{bmatrix}$;
$\mathbf{C} = [\mathbf{C}_\delta, \mathbf{C}_\lambda]$.
```

REFERENCES

[1] "Airbus 320 aircraft model." `http://www.jsbsim.sourceforge.net`. retrieved Aug., 2011.

[2] "Airline data project." `http://web.mit.edu/airlinedata/www/default.html`. retrieved Nov, 2010.

[3] "Mcdonnell Douglas MD-11 aircraft model." `http://www.flightgear.org`. retrieved Dec., 2011.

[4] "Swissair 111 flightpath map." `http://www.digistar.cl/SR111/srmv3o.html`. retrieved Sep. 2011.

[5] *Control and Dynamic Systems: Advances in Theory and Applications*, ch. Aircraft Maneuver Optimization of Reduced-order Approximation, pp. 131–178. Academic Press, Inc., 1973.

[6] "SR-111 Investigation Report," Tech. Rep., Transportation Safety Board of Canada, 2003.

[7] "Statistical summary of commercial jet airplane accidents worldwide operations 1959-2009," Tech. Rep., The Boeing Company, Seattle, WA, 2010.

[8] Asnis, I. A., Dmitruk, A. V., and Osmolovskii, N. P., "Solution of the problem of the energetically optimal control of the motion of a train by the Maximum principle," *Computational Mathematics and Mathematical Physics*, vol. 25, no. 6, pp. 37–44, 1985.

[9] Athans, M. and Falb, P. L., *Optimal Control : An Introduction to the Theory and Its Applications*. Lincoln Laboratory publications, McGraw-Hill, 1966.

[10] Atkins, E. M., Portillo, I. A., and Strube, M. J., "Emergency flight planning applied to total loss of thrust," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 4, pp. 1205–1216, 2006.

[11] Aurenhammer, F., "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.

[12] Babuska, I. and Gui, W., "Basic principles of feedback and adaptive approaches in the finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 55, pp. 27–42, Apr. 1986.

[13] Bakolas, E. and Tsiotras, P., "Optimal synthesis of the asymmetric sinistral/dextral Markov-Dubins problem," *Journal of Optimization Theory and Applications*, vol. 150, no. 2, pp. 233–250, 2011.

[14] Bakolas, E. and Tsiotras, P., "On-line, kinodynamic trajectory generation through rectangular channels using path and motion primitives," in *47th IEEE Conference on Decision and Control*, (Cancun, Mexico), pp. 3725–3730, 2008.

[15] Bakolas, E. and Tsiotras, P., "On the generation of nearly optimal, planar paths of bounded curvature and bounded curvature gradient," in *Proceedings of the 2009 conference on American Control Conference*, (St. Louis, MO), pp. 385–390, 2009.

[16] Barraquand, J., Langlois, B., and Latombe, J.-C., "Numerical potential field techniques for robot path planning," *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 224–241, 1992.

[17] Becerra, V. M., *PSOPT Optimal Control Solver User Manual*, Jul. 2011.

[18] Beckett, G., Mackenzie, J. A., Ramage, A., and Sloan, D. M., "On the numerical solution of one-dimensional pdes using adaptive methods based on equidistribution," *Journal of Computational Physics*, vol. 167, no. 2, pp. 372–392, 2001.

[19] Bellman, R. E., *Dynamic Programming.* Princeton, NJ: Princeton University Press, 1957.

[20] Belta, C., Bicchi, A., Egerstedt, M., Frazzoli, E., Klavins, E., and Pappas, G., "Symbolic planning and control of robot motion," *IEEE Robotics Automation Magazine*, vol. 14, pp. 61–70, Mar. 2007.

[21] Bereg, S. and Kirkpatrick, D., "Curvature-bounded traversals of narrow corridors," in *SCG '05: Proceedings of the twenty-first annual symposium on Computational geometry*, (Pisa, Italy), pp. 278–287, ACM, 2005.

[22] Betts, J. T., "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[23] Betts, J. T. and Huffman, W. P., "Sparse optimal control software SOCS," Mathematics and Engineering Analysis Technical Document MEALR-085, Boeing Information and Support Services, The Boeing Company, Seattle, WA, Jul 1997.

[24] Betts, J. T., Campbell, S. L., and Kalla, N. N., "Initialization of direct transcription optimal control software," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, (Maui, Hawaii), pp. 3802–3807, Dec. 2003.

[25] Betts, J. T., *Practical Methods for Optimal Control using Nonlinear Programming.* PA: SIAM, 2001.

[26] Betts, J. T. and Huffman, W. P., "Mesh refinement in direct transcription methods for optimal control," *Optimal Control Applications and Methods*, vol. 19, no. 1, pp. 1–21, 1998.

[27] Bhat, P., Kuffner, J., Goldstein, S., and Srinivasa, S., "Hierarchical motion planning for self-reconfigurable modular robots," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Beijing, China), pp. 886 –891, Oct. 2006.

[28] Bobrow, J. E., Dubowsky, S., and Gibson, J. S., "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[29] Bryson, A. E. and Ho, Y., *Applied Optimal Control-Optimization, Estimation and Control.* Hemispere Publishing Corporation, Washington, 1975.

[30] Bulirsch, R., Montrone, F., and Pesch, H. J., "Abort landing in the presence of windshear as a minimax optimal control problem, Part I: Necessary conditions," *Journal of Optimization Theory and Applications*, vol. 70, no. 1, pp. 1–23, 1991.

[31] Bulirsch, R., Montrone, F., and Pesch, H. J., "Abort landing in the presence of windshear as a minimax optimal control problem, Part II: Multiple shooting and homotopy," *Journal of Optimization Theory and Applications*, vol. 70, no. 2, pp. 223–254, 1991.

[32] Burrows, J. W., "Fuel-optimal aircraft trajectories with fixed arrival times," *Journal of Guidance, Control, and Dynamics*, vol. 6, no. 1, pp. 14–19, 1983.

[33] Burrows, J. W., "Fuel-optimal aircraft trajectories with fixed arrival times," *Journal of Guidance, Control, and Dynamics*, vol. 6, pp. 14–19, Jan.-Feb. 1983.

[34] Byrd, R. H., Curtis, F. E., and Nocedal, J., "Infeasibility detection and SQP methods for nonlinear optimization," *SIAM Journal on Optimization*, vol. 20, no. 5, pp. 2281–2299, 2010.

[35] Calise, A. J., "Extended energy management methods for flight performance optimization," *AIAA Journal*, vol. 15, no. 3, pp. 314–321, 1977.

[36] Cao, B., Dodds, G. I., and Irwin, G. W., "Constrained time-efficient and smooth cubic spline trajectory generation for industrial robots," *IEE Proceedings Control Theory and Applications*, vol. 144, no. 5, pp. 467–475, 1997.

[37] Chakravarty, A., "Four-dimensional fuel-optimal guidance in the presence of winds," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 1, pp. 16–22, 1985.

[38] Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., and Thrun, S., *Principles of Robot Motion-Theory, Algorithms, and Implementation.* Cambridge, MA: The MIT Press, Jun. 2005.

[39] Chuang, J. and Ahuja, N., "Path planning using Newtonian potential," in *IEEE Conference on Robotics and Automation*, (Sacramento, CA), pp. 558–563, 1991.

[40] Constantinescu, D. and Croft, E. A., "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths," *Journal of Robotic Systems*, vol. 17, no. 5, pp. 233–249, 2000.

[41] Cowlagi, R. V. and Tsiotras, P., "Shortest distance problems in graphs using history-dependent transition costs with application to kinodynamic path planning," in *American Control Conference*, (St. Louis, MO), pp. 414–419, 2000.

[42] Drury, R., Tsourdos, A., and Cooke, A., "Real-time trajectory generation: Improving the optimality and speed of an inverse dynamics method," in *2010 IEEE Aerospace Conference*, pp. 1–12, march 2010.

[43] Drury, R. G., "Performance of NLP algorithms with inverse dynamics for near-real time trajectory generation," in *AIAA Guidance, Navigation, and Control Conference*, (Portland, OR), Aug. 2011.

[44] Dubins, L. E., "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[45] Etkin, B., *Dynamics of Atmospheric Flight*. Dover Publications, 2005.

[46] Faiz, N., Agrawal, S. K., and Murray, R. M., "Trajectory planning of differentially flat systems with dynamics and inequalities," *Journal of Guidance, Control, and Dynamics*, vol. 24, pp. 219–227, Mar.-Apr. 2001.

[47] Fliess, M., Lévine, J., Martin, P., and Rouchon, P., "On differentially flat nonlinear systems," in *Proceedings of IFAC-Symposium NOLCOS'92*, (Bordeaux, France), pp. 408–412, 1992.

[48] Fraichard, T. and Scheuer, A., "From Reeds and Shepp's to continuous-curvature paths," *IEEE Transaction on Robotics*, vol. 20, no. 6, pp. 1025–1035, 2004.

[49] Franco, A., Rivas, D., and Valenzuela, A., "Minimum-fuel cruise at constant altitude with fixed arrival time," *Journal of Guidance, Control, and Dynamics*, vol. 33, pp. 280–285, Jan.-Feb. 2010.

[50] Frazzoli, E., *Robust Hybrid Control for Autonomous Vehicle Motion Planning*. PhD thesis, Massachusetts Institute of Technology, 2001.

[51] Frazzoli, E., Dahleh, M. A., and Feron, E., "Real-time motion planning for agile autonomous vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 116–129, 2002.

[52] Gasparetto, A. and Zanotto, V., "A new method for smooth trajectory planning of robot manipulators," *Mechanism and Machine Theory*, vol. 42, pp. 455–471, Apr. 2007.

[53] Gilimyanov, R. F., Pesterev, A. V., and Rapoport, L. B., "Smoothing curvature of trajectories constructed by noisy measurements in path planning problems for wheeled robots," *Journal of Computer and Systems Sciences International*, vol. 47, no. 5, pp. 812–819, 2008.

[54] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP algorithm for large-scale constrained optimization," numerical analysis report 97-2, University of California, San Diego, La Jolla, CA, 1997.

[55] Gong, Q., Fahroo, F., and Ross, I. M., "Spectral algorithm for pseudospectral methods in optimal control," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 3, pp. 460–471, 2008.

[56] Grimm, W., Well, K., and Oberle, H., "Periodic control for minimum-fuel aircraft trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 9, no. 2, pp. 169–174, 1986.

[57] Guay, M., Kansal, S., and Forbes, J. F., "Trajectory optimization for flat dynamic systems," *Industrial & Engineering Chemistry Research*, vol. 40, no. 9, pp. 2089–2012, 2001.

[58] Hirschorn, R. M., "Invertibility of multivariable nonlinear control systems," *IEEE Transaction on Automatic Control*, vol. 24, pp. 855–865, Dec. 1979.

[59] Holett, P. G., Pudney, P. J., and Vu, X., "Local energy minimization in optimal train control," *Automatica*, vol. 45, no. 11, pp. 2692–2698, 2009.

[60] Hoschek, J., "Circular splines," *Computer-Aided Design*, vol. 2, no. 11, pp. 611–618, 1992.

[61] Howlett, P. G., Milroy, I. P., and Pudney, P. J., "Energy-efficient train control," *Control Engineering Practice*, vol. 2, no. 2, pp. 193–200, 1994.

[62] Hugger, J., "The theory of density representation of finite element meshes. examples of density operators with quadrilateral elements in the mapped domain," *Computer Methods in Applied Mechanics and Engineering*, vol. 109, no. 1-2, pp. 17–39, 1993.

[63] Jackson, M. R., Zhao, Y., and Slattery, R. A., "Sensitivity of trajectory prediction in air traffic management," *Journal of Guidance, Control, and Dynamics*, vol. 22, no. 2, pp. 219–228, 1999.

[64] Jacobson, D. H., *Differential dynamic programming*. Elsevier Scientific, 1970.

[65] Jacobson, M. and Ringertz, U. T., "Airspace constraints in aircraft emission trajectory optimization," *Journal of Aircraft*, vol. 47, pp. 1256–1265, Jul.-Aug. 2010.

[66] Jain, S. and Tsiotras, P., "Trajectory optimization using multiresolution techniques," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 5, pp. 1424–1436, 2008.

[67] Jain, S., *Multiresolution strategies for the numerical solution of optimal control problems*. PhD thesis, Georgia Institute of Technology, Mar 2008.

[68] Jung, D. and Tsiotras, P., "On-line path generation for small unmanned aerial vehicles using b-spline path templates," in *AIAA Guidance, Navigation, and Control Conference*, AIAA-2008-7135, (Honolulu, HI), 2008.

[69] Karaman, S. and Frazzoli, E., "Incremental sampling-based algorithms for optimal motion planning," in *Robotics: Science and Systems*, (Zaragoza, Spain), 2010.

[70] Karelahti, J., Virtanen, K., and Öström, J., "Automated generation of realistic near-optimal aircraft trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 3, pp. 674–688, 2008.

[71] Kato, O. and Sugiura, I., "An interpretation of airplane general motion and control as inverse problem," *Journal of Guidance, Control, and Dynamics*, vol. 9, no. 2, pp. 198–204, 1986.

[72] Kavraki, L. E., Kolountzakis, M. N., and Latombe, J.-C., "Analysis of probabilistic roadmaps for path planning," in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, (Minneapolis, MN), 1996.

[73] Kelley, H. J., "Flight path optimization with multiple time scales," *AIAA Journal of Aircraft*, vol. 8, no. 4, pp. 238–240, 1971.

[74] Khmelnitsky, E., "On an optimal control problem of train operation," *IEEE Transaction on Automatic Control*, vol. 45, pp. 1257–1265, Jul. 2000.

[75] Kirk, D. E., *Optimal Control Theory: An Introduction.* Prentice Hall, 1970.

[76] Krantz, S. G. and Parks, H. R., *A Primer of Real Analytic Functions.* Birkhäuser Boston, 2002.

[77] Kreyszig, E., "On the calculus of variations and its major influences on the mathematics of the first half of our century. Part I.," *The American Mathematical Monthly*, vol. 101, pp. 674–678, Aug.-Sep. 1994.

[78] Lantoine, G. and Russell, R. P., "A hybrid differential dynamic programming algorithm for robust low-thrust optimization," in *AAS/AIAA Astrodynamics Specialist Conference and Exhibit*, no. AIAA-2008-6615, (Honolulu, HI), 2008.

[79] LaValle, S. M., *Planning Algorithms.* Cambridge University Press, 2006.

[80] LaValle, S. M. and Kuffner, J. J., "Randomized kinodynamic planning," in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, (Detroit, MI), 1999.

[81] Ligun, A. A. and Storchai, V. F., "On the best choice of nodes for approximation by splines in the metric of l^p," *Mathematicheskie Zametki*, vol. 20, pp. 611–618, Otc. 1976.

[82] Lozano-Pérez, T. and Wesley, M. A., "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, vol. 22, no. 10, pp. 560–570, 1979.

[83] Lu, P. and Pierson, B. L., "Optimal aircraft terrain-following analysis and trajectory generation," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 3, pp. 555–560, 1995.

[84] Lu, P. and Pierson, B. L., "Optimal aircraft terrain-following flight with nonlinear engine dynamics," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 1, pp. 240–242, 1996.

[85] Maekawa, T., Noda, T., Tamura, S., Ozaki, T., and Machida, K., "Curvature continuous path generation for autonomous vehicle using b-spline curves," *Computer-Aided Design*, vol. 42, no. 4, pp. 350–359, 2010.

[86] Margraves, C. and Paris, S., "Direct tracjectory optimization using nonlinear programming and collocation," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338–342, 1987.

[87] Mattingly, J. D., Heiser, W. H., and Pratt, D. T., *Aircraft Engine Design.* AIAA, 1987.

[88] Meek, D. and Walton, D., "Approximating smooth planar curves by arc splines," *Journal of Computational and Applied Mathematics*, vol. 59, no. 2, pp. 221–231, 1995.

[89] Miele, A., *Flight Mechanics, Vol. I: Theory of Flight Paths.* Reading, MA: Addison-Wesley, 1962.

[90] Miele, A., "Optimal trajectories and guidance trajectories for aircraft flight through windshears," in *Proceedings of the 29th IEEE Conference on Decision and Control*, vol. 2, (Honolulu, HI), pp. 737–746, Dec 1990.

[91] NASA, "Earth atmosphere model." `http://www.grc.nasa.gov/WWW/K-12/airplane/atmosmet.html`. retrieved Aug, 2009.

[92] Nieuwstadt, M. J. V. and Murray, R. M., "Real-time trajectory generation for differentially flat systems," *International Journal of Robust and Nonlinear Control*, vol. 8, no. 11, pp. 995–1020, 1998.

[93] Nilsson, N. J., *Principles of Artificial Intelligence.* San Francisco: Morgan Kaufmann, 1980.

[94] Oberle, H. and Grimm, W., *BNDSCO: a Program for the Numerical Solution of Optimal Control Problems.* Hamburger Beiträge zur angewandten Mathematik / B: Berichte, Inst. für Angewandte Mathematik, 2001.

[95] O'Callaghan, J., "Group chairman's aircraft performance study," Tech. Rep., National Transportation Safety Board, Dec. 2009.

[96] Pfeiffer, F. and Johanni, R., "A concept for manipulator trajectory planning," *IEEE Journal of Robotics and Automation*, vol. RA-3, pp. 115–123, Apr. 1987.

[97] Piazzi, A., Bianco, C. G. L., and Romano, M., "η^3-splines for the smooth path generation of wheeled mobile robots," *IEEE Transaction on Robotics*, vol. 23, no. 5, pp. 1089–1095, 2007.

[98] Plaku, E., Kavraki, L. E., and Vardi, M. Y., "Motion planning with dynamics by a synergistic combination of layers of planning," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 469–482, 2010.

[99] Polak, E., *Optimization: Algorithms and Consistent Approximations.* Springer, New York, 1997.

[100] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, 3 ed., Aug. 2007.

[101] Rao, A. V., Benson, D., and Huntington, G. T., *User's Manual for GPOPS Version 4.x: A MATLAB Package for Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Pseudospectral Methods*, 2011. `http://gpops.sourceforge.net`.

[102] Rao, A. V. and Mease, K. D., "Eigenvector approximate dichotomic basis method for solving hyper-sensitive optimal control problems," *Optimal Control Applications and Methods*, vol. 20, no. 2, pp. 59–77, 1999.

[103] Ross, I. M., "User's manual for DIDO: A MATLAB application package for solving optimal control problems," NPS Tech. Rep. MAE03005, Naval Postgraduate School, Monterey, CA, Sep. 2003.

[104] Rouchon, P., Fliess, M., Lévine, M., and Martin, P., "Flatness, motion planning, and trailer systems," in *Proceedings of IEEE Conference on Decision and Control*, (San Antonio, TX), pp. 2700–2705, 1993.

[105] Sabitov, I. K. and Slovesnov, A. V., "Approximation of plane curves by circular arcs," *Computational Mathematics and Mathematical Physics*, vol. 50, no. 8, pp. 1279–1288, 2010.

[106] Safonova, A. and Rossignac, J., "Compressed piecewise-circular approximations of 3d curves," *Computer-Aided Design*, vol. 35, no. 6, pp. 533–547, 2003.

[107] Sakai, Y., Suzuki, S., Miwa, M., and Tsuchiya, T., "Flight test evaluation of non-linear dynamc inversion controller," in *46th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2008-209, (Reno, Nevada), Jan. 7-8 2008.

[108] Scheuer, A. and Fraichard, T., "Continuous-curvature path planning for car-like vehicles," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 997–1003, 1997.

[109] Schultz, R. L., "Three-dimensional trajectory optimization for aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 6, pp. 936–943, 1990.

[110] Schwartz, A., *Theory and Implementation of Numerical Methods Based on Runge-Kutta Integration for Solving Optimal Control Problems.* Ph.D. thesis, Berkeley, University of California, 1996.

[111] Sethian, J. A., "Curavature and the evolution of fronts," *Communications in Mathematical Physics*, vol. 101, no. 4, pp. 487–499, 1985.

[112] Seywald, H., "Long flight-time range-optimal aircraft trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 1, pp. 242–244, 1994.

[113] Seywald, H., Cliff, E. M., and Well, K. H., "Range optimal trajectories for an aircraft flying in the vertical plane," *Journal of Guidance, Control, and Dynamics*, vol. 17, no. 2, pp. 389–398, 1994.

[114] Seywald, H. and Kumar, R. R., "Method for automatic costate calculation," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 6, pp. 1252–1261, 1996.

[115] Shiller, Z. and Gwo, Y.-R., "Dynamic motion planning of autonomous vehicles," *Robotics and Automation, IEEE Transactions on*, vol. 7, pp. 241 –249, Apr. 1991.

[116] Shiller, Z., "On singular time-optimal control along specified paths," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 561–566, Aug. 1994.

[117] Shiller, Z., "Time-energy optmal control of articulated systems with geometric path constraints," in *Proceeding of IEEE International Conference on Robotic and Automation*, (San Diego, CA), pp. 2680–2685, 1994.

[118] Shiller, Z. and Lu, H.-H., "Computation of path constrained time optimal motions with dynamic singularities," *Journal of Dynamic Systems, Measurement, and Control*, vol. 114, pp. 34–40, Mar. 1992.

[119] Shin, K. G. and McKay, N. D., "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. AC-30, pp. 531–541, Jun. 1985.

[120] Slattery, R. and Zhao, Y., "Trajectory synthesis for air traffic automation," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 2, pp. 232–238, 1997.

[121] Sridhar, B., Ng, H. K., and Chen, N. Y., "Aircraft trajectory optimization and contrails avoidance in the presence of winds," *Journal of Guidance, Control, and Dynamics*, vol. 34, pp. 1577–1583, Oct. 2011.

[122] Steinbach, M. C., Bock, H. G., and Longman, R. W., "Time-optimal extension and retraction of robots: Numerical analysis of the switching structure," *Journal of Optimization Theory and Applications*, vol. 84, pp. 589–616, Mar. 1995.

[123] Stentz, A., "Optimal and efficient path planning for partially-known environments," in *IEEE International Conference on Robotics and Automation*, 1994.

[124] Strube, M. J., Sanner, R. M., and Atkins, E. M., "Dynamic flight guidance recalibration after actuator failure," in *AIAA 1st Intelligent Systems Technical Conference*, (Chicago, IL), Sep. 20-22 2004.

[125] Suzuki, S., Komatsu, Y., Yonezawa, S., Masui, K., and Tomita, H., "Online four-dimensional flight trajectory search and its flight testing," in *AIAA Guidance, Navigation and Control Conference and Exibit*, AIAA-2005-6475, Aug. 15-18 2005.

[126] Synge, J. L., "On the geometry of dynamics," *Philosophical Transactions of the Royal Society of London. Series A*, vol. 226, pp. 31–106, 1927.

[127] Tang, Y., Atkins, E. M., and Sanner, R. M., "Emergency flight planning for a generalized transport aircraft with left wing damage," in *AIAA Guidance, Navigation and Control Conference*, (Hilton Head, SC), Aug. 2007.

[128] Tsuchiya, T., Ishii, H., Uchida, J., Ikaida, H., Gomi, H., Matayoshi, N., and Okuno, Y., "Flight tracjectory optimization to minimize ground noise in helicopter landing approach," *Journal of Guidance, Control, and Dynamics*, vol. 32, pp. 605–615, Mar.-Apr. 2009.

[129] Velenis, E. and Tsiotras, P., "Minimum-time travel for a vehicle with acceleration limits: Theoretical analysis and receding horizon implementation," *Journal of Optimization Theory and Applications*, vol. 138, no. 2, pp. 275–296, 2008.

[130] Verscheure, D., Demeulenaere, B., Swevers, J., Schutter, J. D., and Diehl, M., "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Transaction on Automatic Control*, vol. 54, pp. 2318–2327, Oct. 2009.

[131] Vossen, G., Rehbock, V., and Siburian, A., "Numerical solution methods for singular control with multiple state dependent forms," *Optimization Methods and Software*, vol. 22, no. 4, pp. 551–559, 2007.

[132] Watts, R., Tsiotras, P., and Johnson, E., "Pilot feedback for an automated planning aid system in the cockpit," in *28th Digital Avionics Systems Conference*, (Orlando, FL), Oct. 2009.

[133] Yakimenko, O. A., "Direct method for rapid prototyping of near-optimal aircraft trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 23, pp. 865–875, Sep.-Oct. 2000.

[134] Yang, K. and Sukkarieh, S., "An analytical continuous-curvature path-smoothing algorithm," *IEEE Transactions on Robotics*, vol. 26, pp. 561–568, Jun. 2010.

[135] Zhao., Y. J., "Optimal patterns of glider dynamic soaring," *Optimal Control Applications and Methods*, vol. 24, no. 2, pp. 67–89, 2004.

[136] Zhao, Y. and Tsiotras, P., "A density-function based mesh refinement algorithm for solving optimal control problems," in *Infotech@Aerospace Conference, Seattle, WA*, AIAA-2009-2019, 2009. AIAA-2009-2019.

[137] Zhao, Y. and Tsiotras, P., "Time-optimal parameterization of geometric path for fixed-wing aircraft," in *Infotech@Aerospace Conference*, AIAA-2010-3352, (Atlanta, GA), 2010.

[138] Zhao, Y. and Tsiotras, P., "Density functions for mesh refinement in numerical optimal control," *Journal of Guidance, Control, and Dynamics*, vol. 34, pp. 271–277, Jan.-Feb. 2011.

[139] Zhao, Y. and Tsiotras, P., "A quadratic programming approach to path smoothing," in *2011 American Control Conference*, (San Francisco, CA, USA), pp. 5324–5329, Jun. 29-Jul. 01 2011.

VITA

Yiming Zhao obtained his Bachelor's (2003) and Master's (2006) degrees from the School of Aeronautic Science and Engineering at Beijing University of Aeronautics and Astronautics (BUAA), Beijing, China. Yiming joined the Ph.D. program in the School of Aerospace Engineering at Georgia Tech in August 2006. He also obtained a Master's degree in Mathematics from the School of Mathematics at Georgia Tech in 2010. His research interest include real-time optimization and control, efficient trajectory optimization algorithms, energy-optimal motion planning for dynamical systems, and aircraft trajectory optimization.

Efficient and Robust Aircraft Landing Trajectory Optimization

Yiming Zhao

211 Pages

Directed by Professor Panagiotis Tsiotras

This thesis addresses the challenges in the *efficient* and *robust* generation and optimization of three-dimensional landing trajectories for fixed-wing aircraft subject to prescribed boundary conditions and constraints on maneuverability and collision avoidance. In particular, this thesis focuses on the airliner emergency landing scenario and the minimization of landing time.

The main contribution of the thesis is two-fold. First, it provides a hierarchical scheme for integrating the complementary strength of a variety of methods in path planning and trajectory optimization for the improvement in efficiency and robustness of the overall landing trajectory optimization algorithm. The second contribution is the development of new techniques and results in mesh refinement for numerical optimal control, optimal path tracking, and smooth path generation, which are all integrated in a hierarchical scheme and applied to the landing trajectory optimization problem.

A density function based grid generation method is developed for the mesh refinement process during numerical optimal control. A numerical algorithm is developed based on this technique for solving general optimal control problems, and is used for optimizing aircraft landing trajectories. A path smoothing technique is proposed for recovering feasibility of the path and improving the tracking performance by modifying the path geometry. The optimal aircraft path tracking problem is studied and analytical results are presented for both the minimum-time, and minimum-energy tracking with fixed time of arrival. The path smoothing and optimal path tracking methods work together with the geometric path planner to provide a set of feasible initial guess to the numerical optimal control algorithm.

The trajectory optimization algorithm in this thesis was tested by simulation experiments using flight data from two previous airliner accidents under emergency landing scenarios. The real-time application of the landing trajectory optimization algorithm as part

of the aircraft on-board automation avionics system has the potential to provide effective guidelines to the pilots for improving the fuel consumption during normal landing process, and help enhancing flight safety under emergency landing scenarios. The proposed algorithms can also help design optimal take-off and landing trajectories and procedures for airports.