Advanced Methods For Intelligent Flight Guidance and Planning In Support Of Pilot Decision Making

Final Report
NASA Grant Number: NNX08AB94A

by

Panagiotis Tsiotras
Principal Investigator

Eric Johnson
Co-Principal Investigator

School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0150

for

National Aeronautics and Space Administration
Intelligent Systems Division
NASA Ames Research Center
Mail Stop: N269-2, Rm: 273
Moffett Field, CA 94035

Attention

Dr. Stephen A. Jacklin
Tel: (650) 604-4567
E-mail: stephen.a.jacklin@nasa.gov

March 2012

School of Aerospace Engineering
Atlanta, Georgia 30332-0150 U.S.A.
PHONE 404-894-9526
FAX 404-894-2760

Advanced Methods For Intelligent Flight Guidance and Planning In Support Of Pilot Decision Making

Final Report
NASA Grant Number NNX08AB94A

Panagiotis Tsiotras (Principal Investigator)
Eric Johnson (Co-Principal Investigator)
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0150

Executive Summary

In this report we summarize the research and educational results developed under NASA Grant number NNX08AB94A. The period of performance for this research award was from November 2007 to November 2011, including a one-year NCE.

The overall objective of this research was to support the NASA activities for a "resilient' aircraft in case of an on-board failure. Of particular interest to this research was the problem of on-board safe and reliable trajectory landing generation. The generated trajectories could be given as suggested options to the pilot (in the most common case) or could be executed autonomously by the next generation of on-board flight management system.

Given the previous overall objective, this research has focused on three areas: (a) first, the development of a cockpit aid system to assist pilots with the tasks of selecting a convenient landing site and developing a safe path to land at this site in the event of an onboard emergency. The results of this research suggest that a particular implementation of the pilot aid that uses a simple dial to sort the most promising landing sites was effective. A survey among professional pilots was also conducted to determine the key issues the pilots have to deal when developing a plan to land the aircraft safely during an emergency; (b) second, the development of multi-resolution path planning strategies to generate obstacle-free paths in case of obstacles (mountains, inclement weather, etc). The proposed scheme guarantees the satisfaction of the vehicle's kinematic and dynamic constraints and focuses the limited computational resources on the part of the trajectory that matters most, namely, in the vicinity of the aircraft's current location. The methodology captures long-term, strategic planning along with sort-term, tactical execution by the pilot; and (c) third, the development of robust, numerically efficient methods to solve on-board time-optimal and fuel-optimal trajectories with execution guarantees. To achieve this 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 same methodology can be used to generate good initial guesses. A good initial guess can help the solution converge much faster. We applied the proposed scheme to two actual cases of aircraft incidents, namely, the US Airways flight 1549 case and the Swissair flight 111.

As a result of the support from this award, 1 student received his M.S. degree and three students received their Ph.D. degrees. Nine (9) archival journal publications and 17 conference papers document the results of this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Overview

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 [3]. 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.

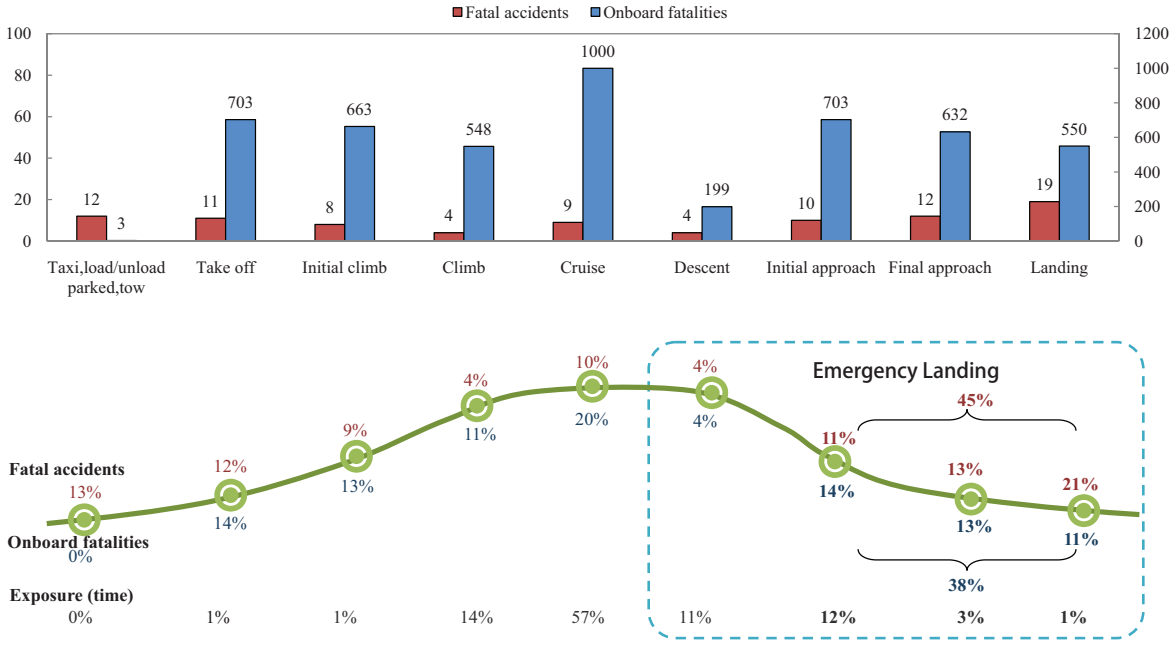


Figure 1.1: Statistical summary of commercial jet airplane accidents.

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 1.1 shows the distribution of fatal accidents and onboard fatalities worldwide

according to the phase of flight from 2000 to 2009 [10] (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 [10] 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 report 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. 1.2, and aims to function as a core component in the Intelligent Flight Planning and Guidance module in the Flight Management System (FMS).

1.2 Problem Statement

In this section we will state the problems addressed in this research. 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

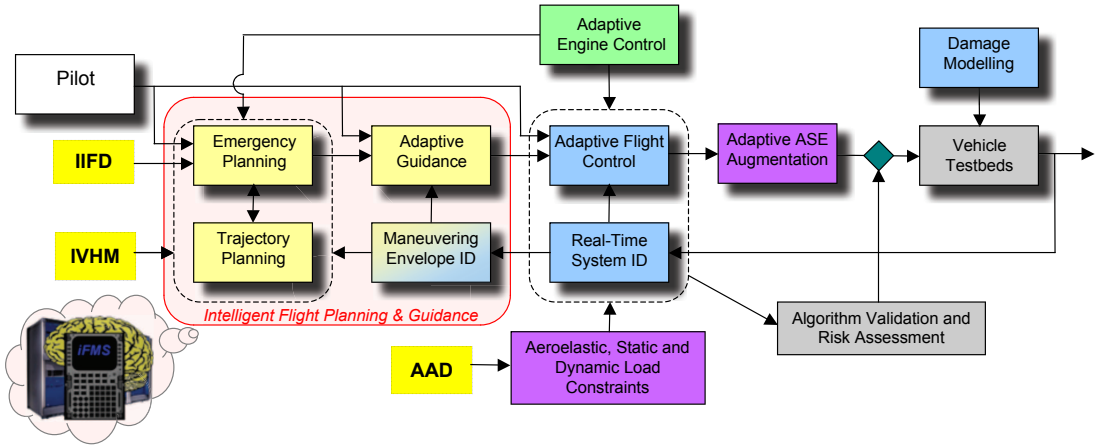


Figure 1.2: NASA Resilient Aircraft Emergency Planning Architecture.

time [171].

In this research 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 $x_{\mathcal{G}}$ 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

$$\dot{x}(t) = f(x(t), u(t)), \tag{1.1}$$

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 1.1 (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 $t \in [t_0, t_f]$,

$$\dot{x}(t) = f(x(t), u(t)), \tag{1.2}$$
$$C(x(t), u(t)) \leq 0, \tag{1.3}$$
$$x_{\mathcal{G}}(t) \in \mathcal{G} \setminus \mathcal{O}. \tag{1.4}$$

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

$$J(x,u) = \int_{t_0}^{t_f} \mathcal{L}(x(t),u(t))\,\mathrm{d}t. \tag{1.5}$$

Hence, we have the following trajectory optimization problem:

Problem 1.2 (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 is not an easy task. More often than not, the required amout of computations prohibit any attempt to solve Problem 1.2 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 by solving Problem 1.1 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 [60, 193]. One way of improving the smoothness of a path is to solve the following variational problem, which regulates the curvature of the path:

Problem 1.3 (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

$$\int_{s_0}^{s_f} w(s)\tilde{\kappa}^2(s)\mathrm{d}s \tag{1.6}$$

Problem 1.3 is a purely geometric problem. The dynamics of the system is not addressed in the process of solving Problem 1.3, although it is expected that the minimization of (1.6) will result in a reasonably good tracking performance, measured by (1.5).

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

Problem 1.4 (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 $t \in [t_0, t_f]$,

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

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 $x_{\mathcal{G}}(\nu(s)) = \mathbf{r}(s)$, $s \in [s_0, s_f]$.

Problem 1.2 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/hiearchical methods can also be applied to trajectory generation and optimization.

1.2.1 The Hierarchical Approach: Path Smoothing and Optimal Path Tracking

Because solving Problem 1.2 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 1.1, which is close to the optimal solution of Problem 1.2 [155, 27, 34]. Hierarchical motion planning methods are sometimes called *hybrid methods* in the literature [76, 77, 141].

A typical hierarchical motion planning process decomposes the task of motion planning into subproblems on several levels [27]. 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 [76].

Reference [141] 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 [150, 74, 20]. Reference [140] used analytical splines and heuristics for smooth path generation. Reference [186] 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. [19] introduced a method for generating curvature-bounded paths in rectangular channels; reference [29] proposed a method for constructing bounded curvature paths traversing a constant width region in the plane, called corridors, and Ref. [103] 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 [121]. 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 1.3. 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 research, Problem 1.4 seeks an optimal solution for tracking a prescribed geometric path subject to dynamics, state, and control constraints. Problem 1.4 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 [35, 160, 139, 158, 156, 176] for shortest travel time along the path subject to control and state constraints. The optimal solutions to these problems can help improve plant productivity [35, 160, 139, 158, 156], racing car performance [176], or faster aircraft landing as will be shown later in this report. 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 [11, 88, 110, 82]. 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 for scheduled operations [11, 88, 110, 82], 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 [11, 88, 110, 82] can provide more reliable and accurate information about the singular arcs in the optimal solution than the numerical optimization approach as in [44, 32, 78, 177].

It is noted that, although originated from different physical systems, the path tracking methods as in Refs. [11, 88, 110, 82, 35, 160, 139, 158, 156, 157, 176] 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 [11, 88, 110, 82]. 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 [35, 160, 139, 158, 156, 157, 176].

In this research, we will solve Problem 1.4 with the aircraft dynamics with two different performance criteria: minimum-time, and minimum-energy with fixed Time Of Arrival (TOA).

1.2.2 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 [40, 41]. The same problem is also studied in [128]. Note that in the physical space, the trajectory is occasionally represented as a four-dimensional flight path, following the tradition of air traffic control [47], with time as the fourth dimension in addition to the normally used three-dimensional representation of a path.

Reference [169] 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. [169] may not be feasible at the junction points between different trajectory segments. A similar approach is used in Ref. [172] to study the emergency flight path

[1]The "singular arcs" in [156] actually refer to segments of speed profile with active speed constraints, which are different from the term's traditional meaning used in optimal control.

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. [169] 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 [181].

Reference [13] 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.

1.2.3 Other Aircraft Trajectory Optimization Problems

The minimum-time, three-dimensional aircraft trajectory optimization prolem was considered in [151] 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 [6]. Not surprisingly, trajectory planning problems have also been studied in the context of air traffic management (ATM) and automation. Reference [95] performed a sensitivity analysis of trajectory prediction for ATM. The aircraft trajectory synthesis problem is studied in [164] to provide some basic tools for air traffic automation.

The aircraft terrain-following (TF) problem is analyzed in [118]. The TF problem is formulated as an optimal control problem that combines short flight time and path-following objectives. The analysis in [118] revealed that the optimal thrust profile is bang-bang in most cases. Inverse dynamics was employed to solve the problem numerically. Reference [119] 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 [166] 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. [96]. 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.

Chapter 2

A Pilot Aid Flight Management System for Emergency Landing

This chapter describes the pilot evaluation of an automated path planning aid concept which can assist pilots with the tasks of selecting a convenient landing site and developing a safe path to land at this site in the event of an onboard emergency. The results suggest that a particular implementation of the pilot aid that uses a simple dial to sort the most promising landing sites was effective. This selectable sorting capability, motivated by the anticipated cognitive mode of the crew, improved the quality of the selected site for the majority of the cases tested. Although this approach increased the time required for the selection, it was also shown that it decreased the time to complete the task in the case of unfamiliar emergencies.

Handling en-route emergencies in modern transport aircraft through an adequate team work of the pilot crew together with the aircraft's automation systems is an ongoing and active field of research. The work presented in this paper highlights the results of a human factors study as part of a proposed automated planning aid which can assist pilots with the tasks of selecting a convenient landing site and developing a safe path to land at this site in the event of an onboard emergency. Focusing on the interactions between the pilot-not-flying and the automated planning aid, the presented results suggest that a particular implementation of the pilot aid interface, which uses a simple dial to sort the most promising landing sites, was effective. This selectable sorting capability, motivated by the anticipated cognitive mode of the pilot crew, improved the quality of the selected site for the majority of the cases tested. Although the presented approach increased the average time required for the selection of an alternate landing site, it is also shown that it decreased the time to complete the task in the case of emergencies unfamiliar to the pilot crew.

2.1 Introduction

Modern air transportation has an excellent flight safety record. When failures do occur in flight, owing to the training and experience of the pilots almost always results in a safe landing. This is evidenced by a rate of only 1.35 accidents per one million hours flown in 2007 by US air carriers [132]. Despite this excellent record, the pilots' responsibility to land safely in case of an emergency can be very demanding. When an emergency situation occurs during a flight, the pilots' workload is very high and a number of tasks demand the pilots' attention. One of the important tasks is the planning and execution of a trajectory resulting in a safe landing. However, this task is

complicated by multiple, often conflicting goals, including reducing time to land, staying within the flight envelope limits of the airplane, weather issues, as well as meeting any relevant regulatory requirements. Moreover, all these tasks must be accomplished in a stressful environment, often under severe time pressure[49].

Although fault tolerant adaptive automation is currently being developed, for the foreseeable future of civil transport aviation, pilots will be the ultimate decision makers, especially in cases of emergencies involving any type of aircraft performance degradation or flight envelope reduction. As a result, current research is being directed at pilot aids that aim at enhancing the pilot's Situation Awareness (SA), as well as at supporting the pilot's decision making process through the provision of relevant, situation-related information.

The purpose of this paper is to report on a human factors study related to efforts to develop an Automated Planning Aid (APA)–in terms of both an acceptable interface and control algorithms–that could assist pilots in generating a plan to safely land at alternative landing sites. In order to do so, the pilots must first determine the "best" landing site and then formulate an expedient and safe trajectory to the ground. This paper presents the results of an evaluation of an APA interface prototype by means of a human-in-the-loop test with commercial airline pilots, focusing on the selection of alternate landing sites during an emergency. Although the implemented APA in the simulator was also able to compute emergency paths to those sites, a detailed description and discussion of this part of the process is omitted in this paper, as it had no immediate effect on the APA interface evaluation.

The results of the study are evaluated in comparison to the opinion and judgment of a single subject matter expert. This expert had more than $20,000$ hours of flight experience in over 20 years of service as a commercial pilot. The authors do acknowledge that this comparison might be improved by incorporating more experts, more test cases and a larger sample. However, given the fact that these scenarios were designed in cooperation with that expert to have an unambiguous "best" solution, it is likely that–given enough time to review each scenario–the vast majority of trained pilots would come to the same conclusion as to which landing site was the best alternative. As such, the authors do believe that the utilized metrics, and the comparison with the experts ranking of the landing sites, is valid for the evaluation of the APA for selecting an alternate landing site under tight time constraints.

2.2 Background

In an emergency situation, the crew must monitor the aircraft systems, detect and resolve any failures, control an aircraft with possibly degraded performance, and coordinate with the cabin crew, airline dispatchers, and air traffic control. In addition to these tasks, the pilots must also plan and execute a trajectory that will result in the safest landing possible. These tasks are made even more difficult by the circumstances during an emergency. For example, the pilots may feel a sense of physical danger, or the cabin environment may be a distraction due to smoke, heat, or noise. Additionally, aircraft performance may be affected, resulting in degraded handling qualities. In order to understand some of the difficulties these circumstances present, a number of cognitive engineering models have been developed in the literature and are reviewed here.

2.2.1 Cognitive Considerations

During an emergency situation a number of contextual features change and alter the pilot's cognitive state. Cognitive Control Theory describes how the context of a situation influences cognition and behavior which change depending on the amount of control the person has [61]. The degree of control a person has is determined, in large part, by the amount of subjectively available time and the familiarity of the situation [84]. Subjectively available time refers to the amount of time that a person perceives that he or she has available to take action. The amount of time perceived may depend on the objective amount of available time, the predicted changes in the system, the person's level of arousal, as well as other factors. In Cognitive Control Theory the degree of control is discretized into four control modes: scrambled, opportunistic, tactical, and strategic. The relationship between the amount of subjectively available time, familiarity of the situation and the control modes is described in [84]. When both familiarity and available time are low, individuals are likely to exhibit behavior associated with a scrambled mode. With more time, but still without familiarity, individuals transition into opportunistic and then into strategic modes; a tactical mode is only expected at moderate to high levels of familiarity. As familiarity becomes greater and time remains low, individuals will transition into opportunistic mode and then to tactical mode. Individuals only transition into the strategic mode when familiarity is low to moderate–otherwise, they remain in the tactical mode regardless of the time available.

The most dangerous mode for a pilot to exhibit is the scrambled mode, which generally corresponds to a person in a state of panic. When a pilot is in this mode, he or she is not able to focus even on a single goal, namely, flying the aircraft. When a pilot has adequate subjectively available time in an emergency, the cognitive state may be better described by the opportunistic mode. In this mode the pilot has a greater sense of control. The pilot is more likely to develop a plan or modify an existing plan in order to fit the current situation. The resulting plan may take into account the potential effects of candidate actions. This mode corresponds to "normal" performance. During an emergency situation, a pilot's cognitive state will likely be somewhere between the scrambled and tactical modes, described by the opportunistic mode. In this mode, pilots are likely to use any plans and procedures available that are deemed to be sufficient; however, these may not be used correctly or most effectively.

The amount of subjectively available time perceived by a pilot may be influenced by a number of factors. The phase of flight during which an emergency occurs, the state and configuration of the aircraft, the type of emergency, the number of actions the pilot is required to complete, the availability of resources, as well as the initial stress level, all contribute to the subjectively available time. Additional stress may be caused by physical factors, such as smoke in the cabin or loud noises, or it may be purely psychological, such as the fear of impending danger. These stress factors affect the manner by which the pilot makes decisions. While the pilot may be able to quickly develop a plan of action based on experience and intuition, stress can lead him or her to fixate on a single solution, and fail to compare alternatives [23, 107]. Additionally, the pilot may simply increase the speed with which he or she processes information, potentially leading to errors. The pilot may also reduce the amount of information that is sought and processed, known as filtration [125, 123]. These stress-related factors can cause pilots to make poor decisions, despite the fact that they would be able to make acceptable decisions under normal circumstances. These inferior decisions may cause incidents to become accidents.

In an emergency situation it is tempting to think that an automated system should be included. However, stress may also lead the pilot to either ignore or rely too heavily on an automated tool. He or she may assume that the plan generated by automation is best, without verifying its feasibility

or exploring other options [129]. In addition, the pilot may seek only information which confirms the automation-generated solution as the best, while discounting other information (confirmation bias). Alternatively, rather than simply discount conflicting information, the pilot may attempt to rationalize and mentally force all available information to fit the automation-generated solution (assimilation bias) [61]. Therefore, care must be taken when devising support systems intended for use in stressful situations.

In addition to the stress, the complex nature of the decision making task is also important. The design presented here is based on a hybrid of the Naturalistic Decision Making (NDM) and Rational, Analytic (RA) decision models. Taking the best of both worlds may allow the decision maker to reach the best result. The NDM framework is often used to describe how experts make complex decisions. Zsambok [194] describes NDM as,

> "the way people use their experience to make decisions in field settings."

While experts are often able to make excellent decisions based on experience and intuition, many of the aforementioned effects of stress can negatively impact the quality of the decision. The RA model of decision making describes how a decision maker proceeds through a set of steps (generating alternatives, envisioning the consequences, evaluating the alternatives against a set of criteria and choosing the best plan) to reach a decision [100]. While under nominal circumstances, a rational decision process may be helpful in determining the safest path to land, it may not the most appropriate model of decision making during an in-flight emergency.

In the NDM/RA hybrid model chosen for the APA, the rational decision process can compensate for some of the weaknesses of NDM. For instance, the rational decision process generates a number of alternatives, which alleviates the tendency to fixate on a single solution. By automating the generation and the evaluation of alternatives, the process can be streamlined. It should be noted that, as Peter Simpson [162, p. 18] warns,

> "a decision aiding system should not become a decision making system, and it should never simply dictate decision courses to the operator."

However, by capitalizing on the automation's fast lookup and simulation abilities, and human pattern recognition and intuition capabilities, the two decision models may be combined to make sound decisions more reliably.

2.2.2 Related Work

The APA must do two things: first, it must be able to accurately predict the most appropriate alternative landing sites, as well as the most suitable trajectory to land at these sites; second, it must provide an intuitive interface for the pilot crew that is appropriate to the task and anticipated context as well as the operator's cognitive state. The completion of the first task requires that the aid determines the overall feasibility of a trajectory. A feasible trajectory must avoid obstacles, which may be static, such as a mountain, or dynamic, such as a severe weather system. The determination of such a trajectory is ideally made by taking into account the aircraft's possibly abnormal characteristics, due to the emergency. Also, the trajectory usually must minimize time to land, which is important in many emergencies. For an appropriate solution to the second task, the aid must also provide an interface with the pilot through which information is effectively communicated in both directions. Most research to date has primarily focused on one or the other of these two tasks.

The landing site selection task has been suggested as a candidate for automation. Atkins, Portillo and Strube [12] have developed a method to complete this task. First, the footprint containing all feasible landing sites is calculated. Then the landing site list within this footprint is prioritized according to a number of weighted criteria, such as runway length, airport facilities available, etc. In [12] chose example values for the criteria weights, but acknowledged that the criteria weights would ultimately be based on expert knowledge and would vary by emergency type.

The need for the pilot and the automated planning aid to interact with each other has also been investigated. The Emergency Flight Planner (EFP) by Chen and Pritchett [49] has been proposed as a prototype interface between the pilot and the pilot aid. The EFP allows the pilot to enter a (flight) plan, the ensuing trajectory is then predicted and evaluated. The EFP also provides an additional mode in which the pilot is presented with a preloaded trajectory, which can then be accepted, modified, or deleted. The results of testing with the EFP emphasized that generated plans must incorporate the structure and objectives used by pilots in order to be effective.

Layton, Smith and McCoy studied human-automation cooperative problem-solving for en route flight planning in [115]. In that study, pilots and air traffic controllers were both used as subjects in the evaluation of three possible modes. The first mode was a sketching-only system, in which a plan devised by the subject was evaluated by the system and feedback was provided. The second was a sketching system with the additional capability for the user to specify constraints on the plan and allow the system to propose a solution, which matched those constraints. In the third mode, the system proposed a plan based on system-specified constraints. The results showed that in the second and third modes, users explored more possible options; however, they were also biased toward the system-generated alternative. The same study also highlighted the fact that the use of a fully automated aid could be detrimental if it performs suboptimally.

The previous results show that in order to increase the usefulness of an APA, the process by which pilots select an alternative landing location and plan a path to, it needs to be better understood. In addition, it needs to be better understood how the pilots' decision making processes can best be assisted by such a tool. It is expected that an aid that accepts and provides information in a manner that is most consonant with the pilot's mental process will be most effective.

2.3 Pilot Aid Tool Design

It should first be noted that pilots are currently not without some form of automated path planning assistance. The modern Flight Management Systems (FMS), which are used on board most major transport aircraft, include pages in the Control Display Unit (CDU) to help the pilot with the task of deciding on a divert landing site. For example, the alternates page `ALTN` of the Boeing 777 FMS displays four possible alternates at a time [86]. These may be input from a list that the pilot creates before the flight, from a database, or can be entered manually. The estimated time of arrival (ETA) and the predicted amount of remaining fuel are displayed. These four alternates are ordered by the ETA.

While these pages are helpful, it is entirely possible the best choice will not be on the list. For instance, the nearest airport list only provides the landing sites at airports at which the aircraft is able to land normally, without taking into account the severity of the emergency. In the case of a severe emergency, the pilot may be willing to land at a runway that is not sufficiently long for a normal landing with normal safety factors. Additionally, in the event of a performance altering emergency, such as a stuck elevator, the FMS cannot presently account for the post-failure flight

dynamics of the aircraft. Thus, the plan generated may not be feasible, given the aircraft's degraded performance. The pilot may alter the recommended plan by altering the waypoints used; however, this requires a non-trivial amount of time and work on the part of the pilot. In the case of an in-flight emergency, both time and cognitive resources may be limited due to the number of other tasks the pilots must address, which suggests the current FMS solutions could be improved for highly time-critical emergencies.

In order to address the shortcomings of the present system, a new APA concept has been developed. The proposed APA is linked to the aircraft's health performance monitoring and alerting system and receives information about failures as they occur in the system. This data is then used to determine the post-failure performance of the aircraft. The updated flight dynamic characteristics of the airplane, combined with terrain and weather information, are then used to compute suited (e.g., time- or fuel-optimal) plans to reach a number of potential divert locations.

Each of the paths to the alternate landing sites is permanently displayed graphically on the ND as well as textually on the CDU. Information about each landing site is collected from precompiled database information, such as data about airports and terrain, as well as live weather information. This additional information is made available to the pilot through the CDU. Based on the information collected, each site is associated with scores from 0 to 1 for different parameters, with a higher score representing a better fit. From these subscores, a cumulative score is calculated based on the system's weighting of the different parameters.[1] The alternates are presented in descending order of the cumulative score in the CDU.

The APA design evaluated here was based on the results of a survey in conjunction with implications suggested by the Cognitive Control Theory survey in [182]. This study investigated the pilots' tasks in the event of an in flight emergency, namely the tasks of choosing a safe landing site, and developing a safe trajectory to reach that site. This survey provides a useful perspective into the methods and priorities pilots use to accomplish these tasks. During an airborne emergency, the need to land quickly is always of high priority. Therefore, the most important factor considered by the pilots when selecting an alternative landing site is proximity in terms of time. Additionally, the weather at the airport, the length of the runway and the distance from the current location are also important criteria. The most important en route factors are the avoidance of severe weather and hazardous terrain.

One of the most important aspects to be considered in an emergency situation is the high workload, time-critical, stressful nature of the situation. Accordingly, one significant feature of any proposed aid is that it should reduce workload, rather than increase it. The aid must provide useful information in a coherent manner, without burdening the pilot with requests. Similarly, pilots view the aid only as a tool, not as a directive. Pilots will use an automatically generated plan in conjunction with their own experience and intuition. Ultimately, the pilot has the final decision-making authority.

A successful design must be closely integrated with interfaces that pilots are currently familiar with. The design of a new tool to be used in the cockpit is a very complex task, as the amount of information and controls available in a modern cockpit is quite large. Also, the physical area in which they must be contained is rather limited. All of the systems' displays and controls must be contained in a small and coherent cockpit layout. With this in mind, no single part of the entire APA system should be designed on its own. It does not exist as a standalone entity, but must work cooperatively with existing systems to allow the pilots to complete all of their responsibilities.

[1] Although the simulated APA was capable of computing polygonal scores and sub-scores, for the purpose of this study, the utilized score values were hardcoded, bases on a subject matter expert's option.

For these reasons, it was decided that the APA be integrated into the existing FMS, utilizing a CDU page which is based on the current `ALTN` page. The alternate routes are continuously displayed on the existing Navigation Display (ND) per survey respondents' preferences. Efforts were also made to ensure that the APA would not adversely contribute to the pilots' workload, but rather present relevant information in a coherent manner. With this in mind, it was determined that the most important function was to help pilots filter information. This was accomplished via the use of a single dial in a slightly modified Navigation Display Control Panel (NDCP), by allowing the pilot to quickly indicate the severity of the emergency by limiting the types of alternates to be considered. Utilizing landing site types as differentiators for the severity of the damage allowed for an intuitive mapping from the pilots' situational analysis to a measure of urgency of the emergency.

In order for the ND to display the routes to the alternate destinations, it must have some method for determining these routes. There are various possible approaches to this research question, among them, for instance, [97, 105]. In the currently implemented design, the approach taken was to calculate these alternate routes in real time, starting from a Dubins path [65]. These Dubins paths serve as initial guesses for a high-fidelity trajectory optimization module whose output can be used by the pilot to get further information about the selected path, along with the corresponding control actions, and can be used to drive an autopilot and/or flight director [21]. The overall trajectory generation step of the APA is shown in Fig. 2.1.

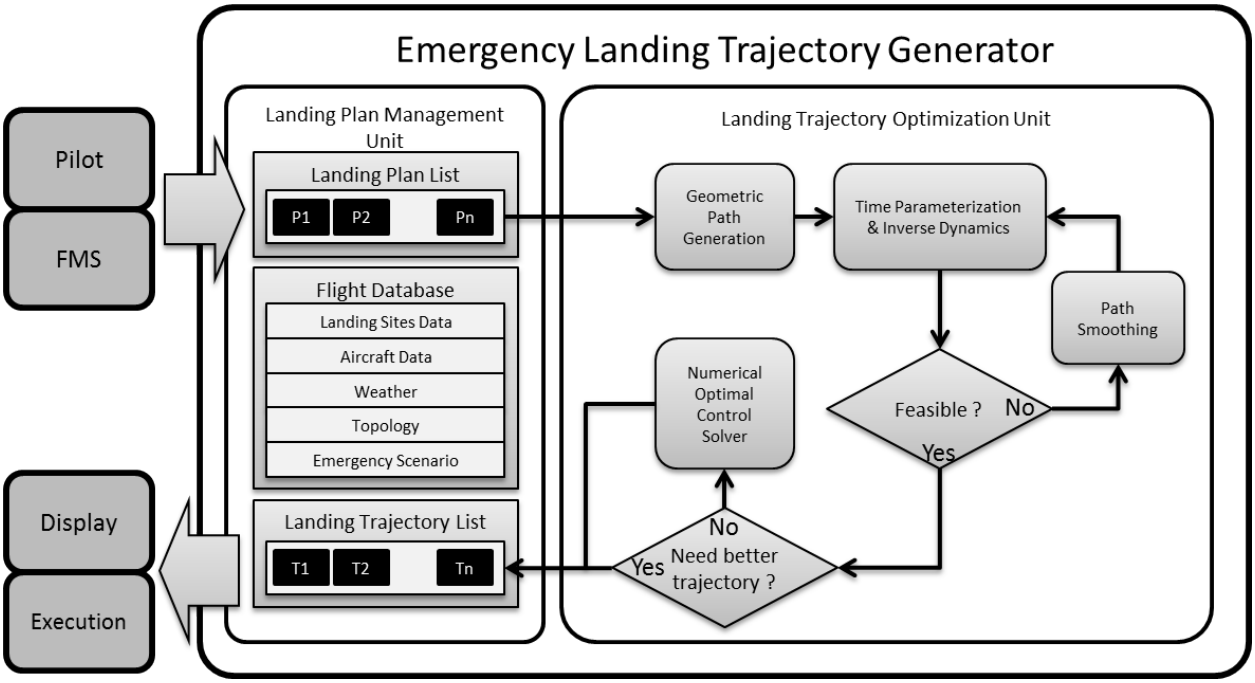


Figure 2.1: Schematic of the overall landing site selection and trajectory optimization process utilized by the APA. The research presented in this paper primarily focuses on the human-machine-interface between the APA and the pilot-not-flying, pictorially represented by the large arrows in the left part of the graphic.

An algorithm for determining appropriate criteria weights based on the type of encountered emergency warrants a study of its own, and is not the focus of the current work. In order to avoid testing the specific criteria weight design, such as those derived from the prior survey results in [182], scores were hard-coded for every site in each scenario, following the advice from a single subject matter

expert. This expert had more than 20,000 hours of flight experience in over 20 years of service as a commercial pilot. The expert was provided all information available about each landing site and, unlike the experiment subjects discussed below, was given an unlimited amount of time to consider each scenario thoroughly. This expert determined scores served to rank the landing sites; these were also the cumulative scores presented to pilots as a weighted combination of the criteria scores and as such presented the expert-determined ranking of the alternate landing sites.

Information from the APA is displayed to the crew through modifications of four displays: Navigation Display, Primary Flight Display (PFD), Navigation Display Control Panel (NDCP), and Control Display Unit. The APA prototype used during the evaluation was built using the Reconfigurable Flight Simulator (RFS) [94]. Each of the display modules used in this simulation is roughly based on the Boeing 777 type displays.

2.3.1 Interface and Setup Description

The ND is on the right of the screen in front of the participant, see Fig. 2.5(b). The display is track up, i.e. the pilot's own aircraft is centered at the bottom of the display and the immediate trajectory is displayed vertically extending from the pictorial representation of the ownship, with the current plan shown as a solid magenta line (see ND detail in Fig. 2.2). The graphical display of the routes to alternate destinations allows the pilot to quickly assess the spatial arrangement of the available alternative landing sites.

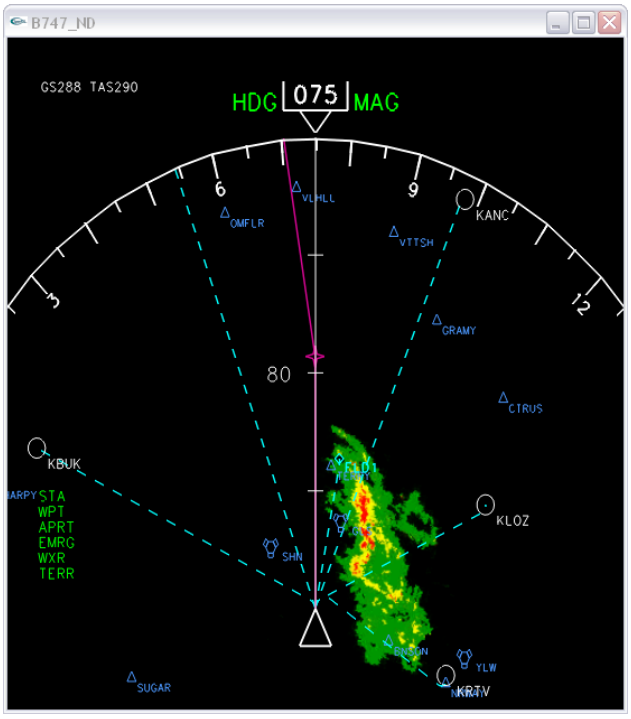


Figure 2.2: Alternate landing sites displayed on the Navigation Display (ND) as dashed, cyan lines. The proposed overlay is the primary way of the APA to present trajectories to alternate sites to the pilots. Details for each trajectory and the corresponding landing sites can be accessed via the CDU, compare Fig. 2.4.

The PFD was located to the left of the ND on the screen in front of the participant, see Fig. 2.5(b).

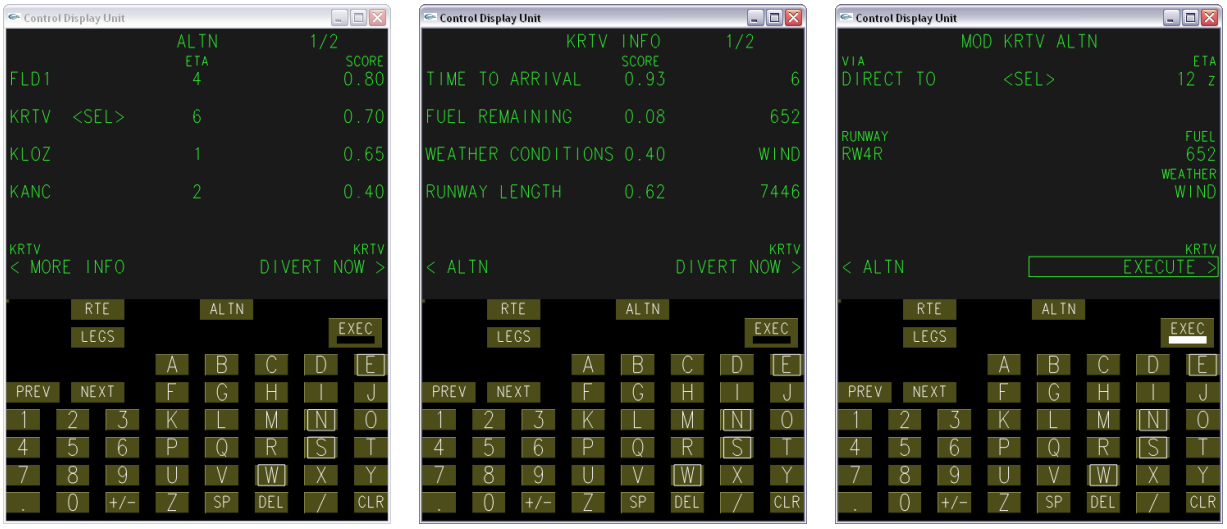
The PFD provides information about the current state of the aircraft such as heading, flight speed, altitude, climb/descent rate, and pitch/roll attitudes. Because the participants were put in the role of a First Officer as the pilot-not-flying, the PFD was provided only as a reference to allow each participant to be aware of the corresponding aspects of the situation. This display provided information which corroborated the emergency that was described, such as loss of altitude as a result of engine failure.

The modified NDCP is the APA's primary input interface and includes seven buttons and two dials, as shown in Fig. 2.3. The buttons toggle data overlay options on the ND: WXR (shows weather systems in the area), STA (shows navigation stations), WPT (shows all waypoints in the area, ARPT (shows airports), EMRG (shows the candidate routes to alternative destinations), TFC (shows traffic), and TERR (shows the terrain). The dial to the right allows the user to set the range (in nautical miles) displayed on the ND (Fig. 2.3 and Fig. 2.2 both reflect a setting of 160 nm), the dial on the left was only present in one of the studied NDCP versions and allowed the pilots to filter possible landing locations. The dial allowed the pilot to quickly indicate the requirements of the landing site, as noted above.

Figure 2.3: A modified Navigation Display Control Panel (NDCP) provides the major input interface to the APA, allowing the pilots to change the APA's configuration while monitoring the effects on the Navigation Display (ND), compare Fig. 2.2. A focus of the study is the effect of presence of the dial in the left, i.e. the newly introduced dial selecting the type of landing sites the pilot wants the APA to consider as alternates.

The Control Display Unit (CDU), see Fig. 2.4, provides a limited subset of the normal CDU functionality required for this evaluation. The route (`RTE`) and legs (`LEGS`) pages provide information about the currently planned FMS route. The alternates (`ALTN`) page was redesigned to provide more information and support more effective use. The destination options, after being filtered by the left dial of the NDCP, were ranked according to the overall scores for each potential landing site. These are the same destination options which are displayed graphically on the ND and may include more than four destinations, in which case the `NEXT` button is used to move further down the list.

The ALTN page allows the pilot to see additional information about each of the options, including time to land, distance, fuel remaining upon arrival, runway length, weather at site, medical services available, and maintenance services available.



(a) The modified ALTN page, displaying landing site identification, estimated time of arrival, and the overall site assigned score.

(b) The MORE INFO page for a particular site, displaying the internal subscore as well as the underlying data for relevant parameters.

(c) Once a site has been selected by the pilot, the CDU can be used to hand over the APA computed plan to the FMS for execution.

Figure 2.4: One part of the APA interface is a modified ALTN page of the Control Display Unit (CDU). Pilots can get an overview via a (ranked) list of potential alternate landing sites, access details for each of the trajectories, and command the execution of any of the proposed plans.

Integrating these landing sites into the exiting ALTN page allows the pilot to select among options in the same manner that is currently available on board modern commercial airliners. After selecting one of the destinations on the list (Fig. 2.4(a) shows the fictional landing site KRTV on the CDU's ALTN page as selected), the pilot was able to view more information about it by pressing MORE INFO [2] , which brings up the corresponding MORE INFO page on the CDU (Fig. 2.4(b) shows the KRTV INFO page.) This page provides information about the landing site and the scores that are used by the ranking system for each of the criteria. After the execution routine has been armed, pressing the lit EXEC button (Fig. 2.4(c)) transfers the computed plan to the FMS, i.e. the autopilot and/or flight director in the PFD.

2.4 Experiment Description

The experiment tested for differences in performance for pilots using two variations of the APA, focusing on the actual human-machine interaction, which is represented by the large arrows in the left of the schematic of the overall process, Fig. 2.1. One variation of the APA included the left dial shown in Fig. 2.3, which facilitates the filtering of landing sites; the other APA version did not include such a dial. The two variations were otherwise identical. In each run, the participant was presented with a scenario in which an emergency occurred. Emergencies that the pilot was

[2] A real CDU provides buttons to the side of the text fields. Due to the implementation on a touchscreen, the subjects in this study had to click the actual field instead.

expected to have been trained to handle, as well as unfamiliar emergencies, were presented. The pilot had the opportunity to use the aid, either with or without the dial, to consider the possible alternate landing sites, and finally select a plan to land. The participating pilots did not actually fly the simulated aircraft, but they participated as a First Officer, that is, the non-flying pilot. The simulation run ended when the subject had selected a route and executed it by selecting the `EXEC` button, shown in Fig. 2.4(c).

2.4.1 Participants

A total of eight pilots participated in the experiment procedure. These pilots all hold an Airline Transport Pilot certificate and were experienced in a variety of aircraft, primarily Boeing, Airbus and Bombardier. One participant had recently retired, while all others currently fly with a commercial airline. The average number of flight experience for the participant pilots was $8,194$ hours.

2.4.2 Independent Variables

The experiment included the variation of two independent variables. Each independent variable had two levels, creating four configurations with two replications. These variables were the aid type, and familiarity of the emergency.

The scenarios were run with two variations of the APA. In one mode, the pilot was able to use the previously described dial to filter possible landing sites (Fig. 2.3, left dial). In the second variant, this dial was not available. In the sequel, this variable is indicated by $(\ )_{\text{dial}}$ and $(\ )_{\text{no dial}}$, respectively.

The evaluation scenarios simulated two general types of emergencies: familiar emergencies and unfamiliar emergencies. In the familiar scenarios, the aircraft's performance was either unaltered or was altered in a manner that pilots had been trained to handle, such as a single engine failure. The second type of scenario was a performance altering emergency in which the failure was one which the pilots had not been specifically trained to manage, such as a stuck elevator. The scenarios were designed such that each scenario was comparable in terms of difficulty and number of options that the participants were expected to consider. The comparison of these two emergency categories is important because pilots may make decisions differently during a familiar scenario than they do during an unfamiliar one, indicated by $()_{\text{familiar}}$ and $(\ )_{\text{unfamiliar}}$, respectively.

2.4.3 Dependent Variables

Two primary metrics were considered. As a first metric, the pilot's ability to choose the "best" landing site was assessed. A subject matter expert was consulted in order to provide aggregate scores for each landing site based on all information available, having unlimited time. These scores served to rank each of the landing sites in each scenario. The second metric was the amount of time pilots spent during the selection process. A reduction in time promotes safe flight by allowing the pilots to focus on other important tasks associated with handling the specific emergency, such as crew coordination, or alerting personnel on the ground. The time required for the pilot to select a path was used to determine the efficiency with which the pilot was able to develop a plan. This time was measured from the moment the emergency occurred, to the time that the pilot selected a plan for execution in the CDU.

In support of these measures, other secondary measures were used to assess the APA. The number of candidate landing sites the pilot reviewed and the number of times the pilot turned the filter dial (when available) were also measured. In addition to simply comparing the total time to complete the task, these measures allowed for a more granular analysis of the task. The following is an overview of all the dependent variables measured:

Time Time was measured from the time the emergency occurred to the time the pilot executed a path in the CDU.

Quality Each scenario involved a number of potential landing site options. These options were ranked according to the appropriateness for the given scenario. This ranking was enabled through an *a priori* evaluation through a subject matter expert. The quality aspect of the participant's performance was based on the rank assigned to the participant-selected landing site in the experts *a priori* determined ranking.

Number of alternates viewed The number of alternates viewed was the number of landing sites for which the pilot viewed the `MORE INFO` page. This was automatically recorded by the system and was the total number of alternates viewed before and after the emergency occurred. The number of alternates viewed was recorded in order to evaluate factors that influenced the amount of time required to reach a selection.

Situation Awareness The participant's situation awareness was assessed immediately after the completion of each run following the Situation Awareness Global Assessment Technique (SAGAT) method [69]. The displays were blanked and the participant was asked ten questions about the current scenario. The questions assessed all three levels of SA [70, 71]. Level 1 assessed the pilot's perception of cues, Level 2 assessed the pilot's comprehension of the situation, and Level 3 assessed the pilot's ability to forecast future events. The ten questions contained five Level 1 questions, three Level 2 questions and two Level 3 questions. These were drawn from a pool of twelve Level 1, ten Level 2 and five Level 3 questions.

Workload The participants were asked to evaluate the perceived workload experienced in each scenario, in order to assess the feasibility of its use in a real emergency. The NASA Task Load Index (TLX) [91] was used in this study to assess the workload for six different sources; mental demand, physical demand, temporal demand, performance, effort, and frustration.

APA Assessment Upon completion of the experiment, the participants were asked to complete a questionnaire, which included questions about the pilot's experience in each simulated flight, as well as an evaluation of both variations of the APA. These included subjective assessments as well as a rating from 1 to 10 based on the modified Cooper-Harper rating for displays [63].

Design of Experiments A 2x2 repeated measure, full-factorial design with one replicate was used to evaluate the effect of APA dial and experiment familiarity on the dependent variables. The design was blocked on the APA type, i.e. the presence or absence of the landing site type filter knob is one of the independent variables in the statistical analysis. As a result, each participant saw four emergencies with one variation, then four with the other variation, denoted by $()_{\text{dial}}$ and $()_{\text{no dial}}$, respectively. An additional "no failure" scenario was included in order to reduce the pilots' expectancy of an emergency. Accordingly, each participant completed a total of nine runs, i.e. each of the four configurations paired with the eight scenarios plus the additional "no failure" run.

Scenarios A total of eight emergency scenarios were created (not counting the "no failure" straight-and-level flight one). Each scenario was characterized by the emergency situation, the phase of flight during which the emergency took place, the alternate landing sites which were available and the time at which the failure occurred. The eight emergencies each occurred in one of three phases of flight: climb, descent or cruise. The emergencies which were repeated occurred in different phases of flight each time. Each scenario had a fixed number of potential landing sites, dependent on the phase of flight in which the emergency occurred. Emergencies occurring during climb had three sites, during cruise had six sites, and during descent had four sites. All identifiers of airports, waypoints, and navigation stations were fictional to prevent any effect due to location familiarity or lack thereof.

There were a total of six types of emergency situations, three familiar situations and three unfamiliar. In each case the Captain, who–as the more experienced aviator–was the flying pilot, described the emergency to the First Officer, who, as the non-flying pilot, was tasked with selecting an alternate landing site. The Captain's description was simulated through an audio playback. The PFD also showed any appropriate changes (such as descent) and the newly introduced alert display (Fig. 2.5(b)) annunciated the appropriate message. The three familiar emergencies used were engine failure, low fuel, and fire onboard. In order to allow for appropriate descriptions and understanding of the failures, flight control failures (for which pilots had not been trained for) were used as the unfamiliar emergency cases. These were: stuck rudder, stuck aileron, and stuck elevator. The amount of time passed after the simulation began until the emergency occurred was different for each scenario, and it was between 45 seconds and 105 seconds, in an attempt to create an element of surprise and–in conjunction with the "no failure" run–minimize pilot readiness. Although real emergencies could happen hours into an otherwise normal flight, the metrics of interest for this study were assumed to not be affected by fatigue resulting from a prolonged period of normal flight time before the occurrence of an emergency.

Procedure Before entering the simulator, each participant had to read a briefing document. This document introduced the pilot to the features of the simulator and the procedures which would be used to conduct the experiment. The introductory material informed (erroneously) participants that the compensation they would receive depended on how well they would perform the emergency planning task (but the compensation did not depend on correctly answering the SAGAT and TLX questions).

After reading this introductory material, the participant entered the simulator to complete at least two practice scenarios. In the first training scenario, no failure occurred. This run simply allowed the pilot to explore the interface and familiarize himself with the tools available. The second run presented a simple engine failure scenario. This allowed the pilot to gain an expectation of how emergencies would be presented and how the tools would allow diversion planning. After each run the participant was given sample situation awareness (SA) questions to familiarize them with the format and types of questions which would be asked. This was done primarily in order to avoid any differences between the first couple of runs and the later runs in terms of answering SA questions. Each participant was also asked to complete the TLX workload questions. The participant was then given the option to run either of the training scenarios again or to begin with the test runs.

After completing the training runs, each pilot completed nine test runs. Eight of these runs contained one of the aforementioned emergencies and one was completed without an emergency. The no-emergency run was included in order to slightly reduce the pilots' expectancy

of an emergency. The pilot joined each flight in progress and was able to use the tools available to gain an understanding of the situation. Between 45 and 105 seconds after the simulator was started, an emergency situation was presented to the pilot through a recorded message. The participant then used the tools to determine the most appropriate landing site for the given situation.

After selecting the route to the alternate landing site for execution, the simulator was closed and the participant was handed a clipboard and asked to complete the ten SA questions. After completing these questions, the participant was asked to complete the TLX questionnaire on the touchscreen.

Upon completion of all nine runs, the pilot was given an additional set of questions pertaining to the experiment as a whole. These included subjective questions about the features and usage of the APA, along with a Modified Cooper-Harper ranking sheet to rate the usability of the APA.

After the completion of all the runs, and after answering all the questions, the participants were made aware of the initial deception regarding the performance-tied compensation and were informed that all of them would receive the entire, same, amount as a token of appreciation for participating in this study.

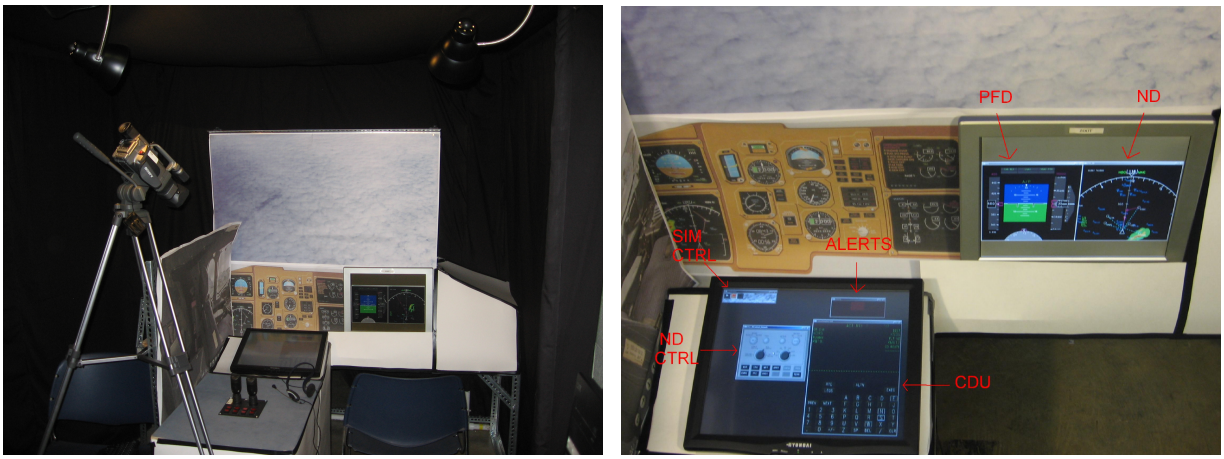
Apparatus The overall experimental setup is shown in Fig. 2.5. An external frame is covered with black cloth to block ambient light sources and isolate the experiment setup. Inside, a mock up of a flight deck consisted of a large shelf in front of the pilot supporting the primary flight monitor and computer, and a center console separating the captain and first officer seats. The touchscreen was placed on the center console in order to allow the pilots to interact primarily with the CDU near its normal position.

The subjects were seated to the right of the center console in the First Officer's seat. Posters were included to provide the pilot with the look and feel of an actual cockpit. These images included the captain's seat, other displays which were not simulated, and a view through the windscreen. The computer screen in front of the pilot showed the PFD on the left and the ND on the right. The screen on the center console was an LCD touchscreen, where the pilots were able to interact with the system. This screen contained the CDU, the NDCP, an alert display and ancillary simulator controls.

2.5 Analysis of Results

The statistical analysis was performed using both parametric and non-parametric statistical analysis, using a repeated-measures Analysis of Variance (ANOVA) or a Spearman's signed rank test, respectively [180, 161]. The significance for all tests was set at $\alpha = 0.05$. In cases where the results from the repeated-measures ANVOVA were found to be significant, the effect size η^2 is reported as a percentage of the overall variance attributed to each predictor. In cases where the results were found to be insignificant, the power for the test, $1 - \beta$, is reported.

The F-distribution and statistic utilized for the ANOVA is characterized by two parameters, the degrees of freedom in the numerator and denominator, respectively. The first represents the number of groups, for this work 1, the second the number of cases minus the number of groups, for this work $8 - 1 = 7$. In the following, $F(1, 7)$ represents the computed value for the observed data.



(a) First-Officer in a pilot-not-flying situation.

(b) The dynamic interfaces of the simulator.

Figure 2.5: The overall experimental setup. Assuming the Captain to be the pilot-in-command during the emergency, the simulator aims at replicating a scenario in which the First Officer is tasked with the landing site selection. The dynamic interfaces the First Officer can access are a the PFD and the ND (conventional LCD display, top right) as well as the CDU and the ND Control Panel (touchscreen LCD, bottom left).

This value is compared to $F_{critical} = F_{critical}(0.05; 1, 7) = 5.591$. If $F(1, 7) > F_{critical}$, the result is assumed to be significant at the 0.05 level. For details, see [180, 161].

2.5.1 Performance

Overall, the average time required to select a landing site was 110 seconds, with a standard deviation of 48 seconds. The unique combination of each phase of flight, failure type and airport availability in each scenario is an important factor which largely accounts for the difficulty in selecting a landing site (see Fig. 2.6). Each test run was completed in a suitable amount of time with the exception of one test run by one subject in which the engine failed during climb. The subject in this run selected an alternate landing site 218 seconds after the failure occurred, which was considered too long by the expert, given the aircraft's altitude at the time. All other test runs by this subject–as well as by all other subjects–were, according to the subject matter expert's opinion, completed within an acceptable amount of time. As shown in Fig. 2.6, the average times varied between the different scenarios. While there was some variation between all scenarios, the times for the aileron failure scenario were significantly higher than for other scenarios. The average time of the aileron failure scenario was 206.8 seconds compared to an average of 96.0 seconds for all other scenarios.

The ability of pilots to select the best landing site (i.e., to pick the same landing site that the subject matter expert *a priori* determined to be the best pick of all alternates) was generally quite good. The median ranking of landing sites selected was 1, and the selected landing site was ranked highest in 57.8 % of runs. Figure 2.7 shows how the participants' selections were classified across scenarios. These results show that the scenarios may have differed in their difficulty. For the engine failure during climb and the rudder control failure in climb, every pilot was able to select the most appropriate landing site. For the fire emergency scenario, only one participant selected the most appropriate landing site. This may not be surprising, considering that time is most limited in the event of an onboard fire. For the elevator failure scenario, a number of different sites were selected,

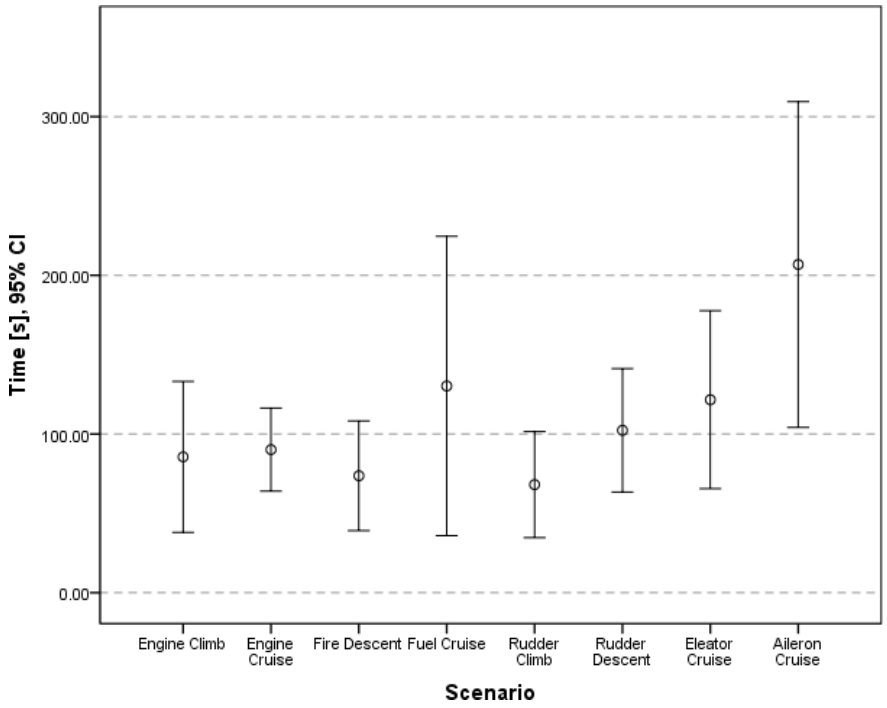


Figure 2.6: The average Time-to-Complete by scenario and the corresponding 95 % confidence intervals.

which may indicate that the differentiation between the best landing site and the other landing sites may not have been substantial enough.

In each scenario, there are some trade-offs between the two measures of performance, time and quality. It was expected that some pilots may spend more time deliberating and coming to a better decision, while others may make decisions more quickly at the expense of the quality of the decision. Spearman's signed rank test [161] was used to determine the correlation between these two measures. The results showed that there was no significant correlation between time and quality ($r_s = 0.203$, $p = 0.107$). With the exception of four out of the total of 64 cases, all runs ended with a selection of a top three ranked site. Focusing on those results (i.e. quality levels 1-3), Figure 2.8 shows that runs which were ranked lower in quality, took slightly longer on average to make and had much greater variance than those classified as being the best, corresponding to a quality of 1.

A repeated-measures ANOVA was conducted to determine the effect of APA type and the familiarity of emergencies on the time to complete. The mean time for cases with the filter dial was slightly higher than those without the dial ($\mu_{\text{dial}} = 115.3$ s, $\mu_{\text{no dial}} = 104.4$ s). The time to select an alternate was lower in the case of familiar emergencies than in cases of unfamiliar emergencies ($\mu_{\text{familiar}} = 94.9$ s, $\mu_{\text{unfamiliar}} = 124.7$ s). The effect of APA type was statistically insignificant ($F(1,7) = 0.217$, $p = 0.655$, $1 - \beta = 0.069$), whereas the effect of the familiarity of the emergency on time to complete was statistically significant ($F(1,7) = 6.979$, $p = 0.033$, $\eta^2 = 0.499$). The interaction between APA type and emergency familiarity indicated that the effect of the dial was not statistically different in the familiar cases than in the unfamiliar cases, Fig. 2.9.

The effect of the APA variation was also tested in regards to the quality of the landing site selected, Fig. 2.10. Cases in which the filter dial was present resulted in slightly better quality of alternates

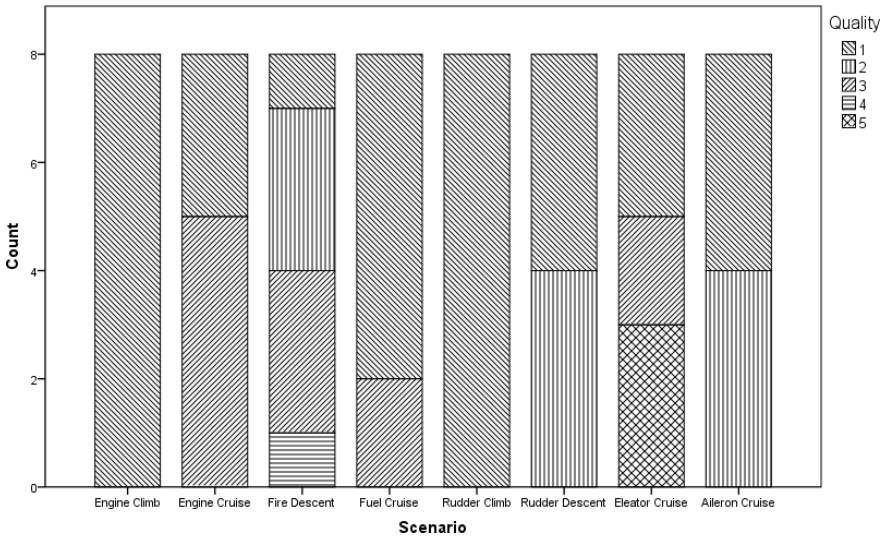


Figure 2.7: Landing site quality by scenario. The quality metric is the rank of the selected option in the expert's solution.

selected. The selected landing site was ranked first or second in 84.4 % of cases which included the dial, compared to only 65.6 % of cases which did not include the dial. Because this dependent variable is ordinal, two Wilcoxon Signed Ranks tests [161] were used. The results from the Wilcoxon tests showed that the difference in quality of landing site selection was not significantly affected by either the APA type ($Z = -1.304$, $p = 0.192$), nor the familiarity of the emergency ($Z = -0.382$, $p = 0.703$). Thus the quality is robust regardless of the familiarity of the emergency or the ability to filter the suggestions.

2.5.2 Workload and Situation Awareness

As previously discussed, the cockpit environment can be very stressful in the event of an emergency. The pilots have a number of tasks which require their attention, and which must be completed in a timely fashion. Therefore, any automation added to the flight deck should not add to the already high workload. A repeated-measures ANOVA was used to determine the effect the independent variables had on the pilots' workload. The effect of the addition of the dial slightly reduced the workload ($\mu_{\text{dial}} = 51.1$ s, $\mu_{\text{no dial}} = 54.9$ s), but this reduction was not statistically significant ($F(1,7) = 1.878$, $p = 0.213$, $1 - \beta = 0.221$). The familiarity of the emergency did not have a significant effect on workload either ($\mu_{\text{familiar}} = 52.6$ s, $\mu_{\text{unfamiliar}} = 53.4$ s, $F(1,7) = 1.277$, $p = 0.296$, $1 - \beta = 0.165$).

A correlation analysis was used to determine if there was indeed a correlation between the pilots' workload and their performance. Spearman's rank correlation coefficient was used to determine that the correlation between the workload and time ($r_s = 0.432$, $p < 0.001$) as well as between workload and quality ($r_s = 0.286$, $p = 0.022$) were both significant. As workload increased, the amount of time taken to determine a solution increased and the quality of the selection decreased.

In order to make a good decision, the pilot must be aware of the situation at hand. The results of the SA questionnaire were used to determine the effect of the aid variation on the pilots' understanding of each situation. Level 1 SA questions were correctly answered at 56 %, Level 2 at 52 % and Level 3 at 47 %. These low percentages of correct responses may be partly attributed to the

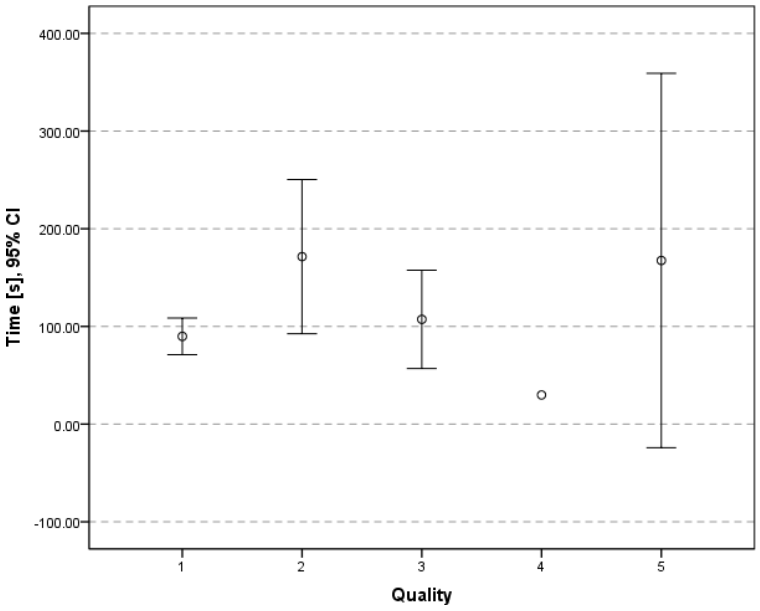


Figure 2.8: Correlation between the average time to get to a decision and the decision quality (1 corresponds to the best quality). Also shown are the 95 % confidence intervals.

type of questions that were asked. For instance, a number of the questions required that pilots recall airport identifiers in order to correctly answer the questions, which were unfamiliar to the participants. Also, pilots were able to keep only the important identifiers in mind, and these were kept only long enough to complete the scenario. These factors likely contributed to low numbers of correct responses.

Despite the low absolute scores, three separate repeated-measures ANOVAs were used to test for effects of the independent variables, one for each level of SA. For Level 2 and Level 3 situation awareness the effect of the APA dial and the familiarity of the experiment individually were marginally statistically significant. Level 1 situation awareness was overall marginally significantly affected by the APA dial ($F(1,7) = 3.781$, $p = 0.093$, $\eta^2 = 0.351$) and the familiarity of the experiment ($F(1,7) = 4.515$, $p = 0.024$, $\eta^2 = 0.539$). Level 1 SA was greater without the dial ($\mu_{\text{dial}} = 0.48$, $\mu_{\text{no dial}} = 0.58$) and greater in situations which were familiar ($\mu_{\text{familiar}} = 0.58$, $\mu_{\text{unfamiliar}} = 0.48$). Additionally, Level 2 situation awareness was significantly affected by the interaction of the APA dial and the familiarity of the experiment ($F(1,7) = 9.00$, $p = 0.020$, $\eta^2 = 0.562$). In unfamiliar emergency situations, the Level 2 situation awareness (understanding perceptual cues) is improved with the inclusion of a filter dial ($\mu_{\text{dial}} = 0.604$, $\mu_{\text{no dial}} = 0.438$), whereas in familiar situations the Level 2 situation awareness is decreased ($\mu_{\text{dial}} = 0.417$, $\mu_{\text{no dial}} = 0.500$), Fig. 2.11.

As with workload, changes in situation awareness–regardless of the source–may have an effect on the pilots' performance. In order to test for this correlation, the Spearman's rank correlation coefficient test was again used. The results show that there was no significant correlation between performance and Level 1 or Level 2 SA. There was, however, a marginally significant negative correlation between Level 3 SA and the time to select a landing site ($r_s = -0.212$, $p = 0.092$). Thus the higher the Level 3 SA the more quickly the pilot was able to choose a landing site.

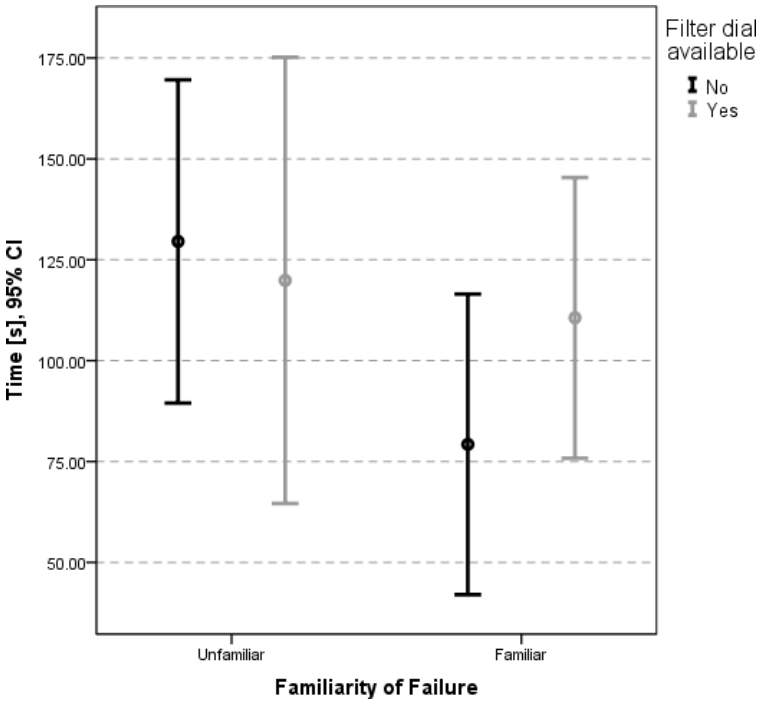


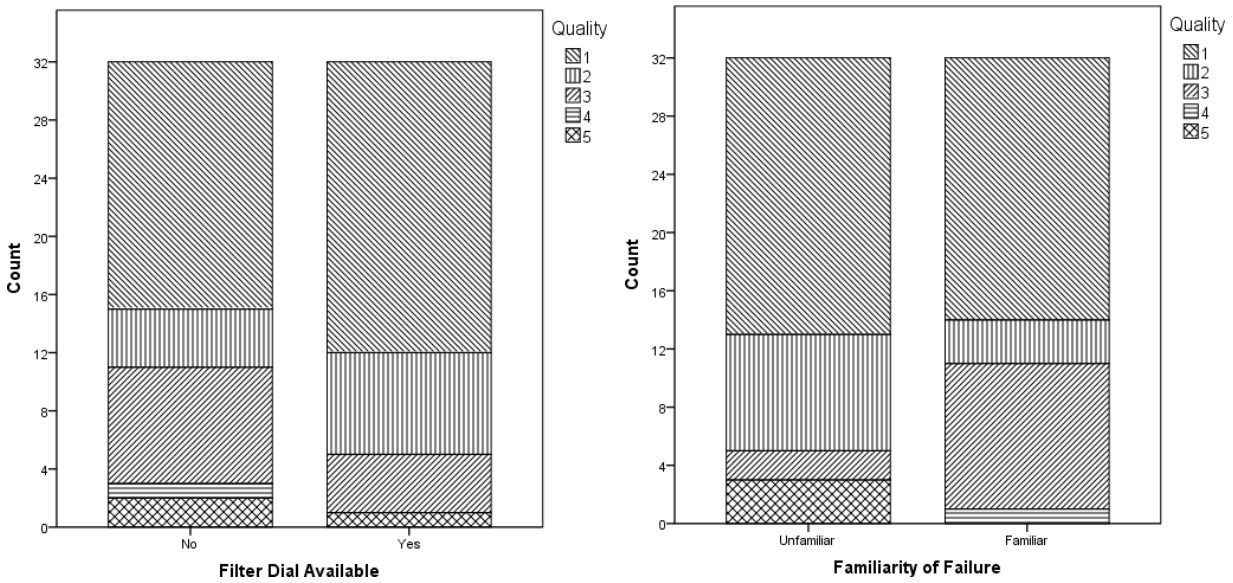
Figure 2.9: Effect of APA variation and familiarity on the average time to make a selection as well as on the variance (indicated by the 95 % confidence intervals).

2.5.3 Secondary Measures

In order to understand the usage of the APA with and without the dial, both the number of landing sites for which the pilot viewed more information, and the number of times the dial setting was changed (for cases in which the dial was available) were analyzed. Viewing more options takes time but may lead to a better understanding of the alternates available and thus a better selection quality.

A repeated-measures ANOVA was conducted on the number of landing sites viewed to determine the effect that the independent variables had on this measure. The average number of solutions viewed was slightly lower for the cases with the dial than for those without the dial ($\mu_{\text{dial}} = 4.5$, $\mu_{\text{no dial}} = 5.0$), however this difference is not statistically significant ($F(1,7) = 0.329$, $p = 0.575$, $1-\beta = 0.084$). The familiarity of the emergency had a significant effect on the number of solutions viewed ($F(1,7) = 5.639$, $p = 0.031$,$\eta^2 = 0.273$). The average number of alternates viewed in familiar emergencies was lower than cases with unfamiliar emergencies ($\mu_{\text{familiar}} = 3.6$, $\mu_{\text{unfamiliar}} = 5.9$). A moderately significant interaction was also found between APA type and emergency familiarity ($F(1,7) = 0.922$, $p = 0.058$, $1-\beta = 0.147$). As seen in Fig. 2.12, in familiar emergencies, the addition of the dial slightly increased the number of solutions viewed from $\mu_{\text{no dial}} = 3.5$ to $\mu_{\text{dial}} = 3.8$. However, in unfamiliar emergencies, the dial reduced the average number of solutions viewed from $\mu_{\text{no dial}} = 6.5$ to $\mu_{\text{dial}} = 5.3$. This may indicate that in familiar cases the filter encouraged the pilot to examine more alternatives than he otherwise would have, and in the unfamiliar cases allowed the pilot to focus on a smaller but more promising set of alternatives.

In order to determine the effect the number of alternates viewed had on the pilots' performance, Spearman's rank correlation coefficient test [161] was used to determine the correlation between



(a) With and without dial.

(b) Familiar and unfamiliar scenarios.

Figure 2.10: Quality of the landing site selection by dial availability and failure familiarity.

the number of alternates viewed, the time to select an alternate, the workload reported and the quality of the landing site selected. There was a significant positive correlation between the number of alternates viewed and the time to complete the task ($r_s = 0.737$, $p < 0.001$). The slight positive correlation between the number of alternates viewed and the quality of the selection made was not significant ($r_s = 0.171$, $p = 0.178$). This implies that participants who viewed more solutions took significantly more time to make a selection, but did select significantly better alternatives. A significant positive correlation was found between the workload reported and the number of alternatives viewed ($r_s = 0.352$, $p = 0.002$). This positive correlation implies that for runs which pilots viewed more alternates, they also encountered a higher workload.

Following all of the data collection runs, the pilots were asked whether they subjectively felt that the filter dial was a useful addition to the APA system. Half of the participants responded that the addition of the dial made the system "Much better," while the other half responded that the dial made the system "Better." This feedback may be reflective of the fact that the dial may be used or ignored in any given situation. A general sentiment was expressed that a tool which can be used or ignored is desirable.

The Modified Cooper-Harper for Displays [63] was used to assess both variations of the aid. Table 2.1 provides the description associated with each level of the rating scale. Fig. 2.13 shows the results from both variations of the aid. Every participant rated the version with the dial the same or higher than the version without the dial. One notable case was the participant who assigned both versions of the aid a score of 10, which is described as "Display is missing critical information; operator is unable to locate essential information..." This subject commented that "runway length is of critical importance and is too hard to find in the pages." All other participants rated the variation with the dial as either "Excellent & Highly Desirable," i.e. 1, or "Good with Negligible Deficiencies," i.e. 2. Only half of the participants rated the variation without the dial in either of these categories.

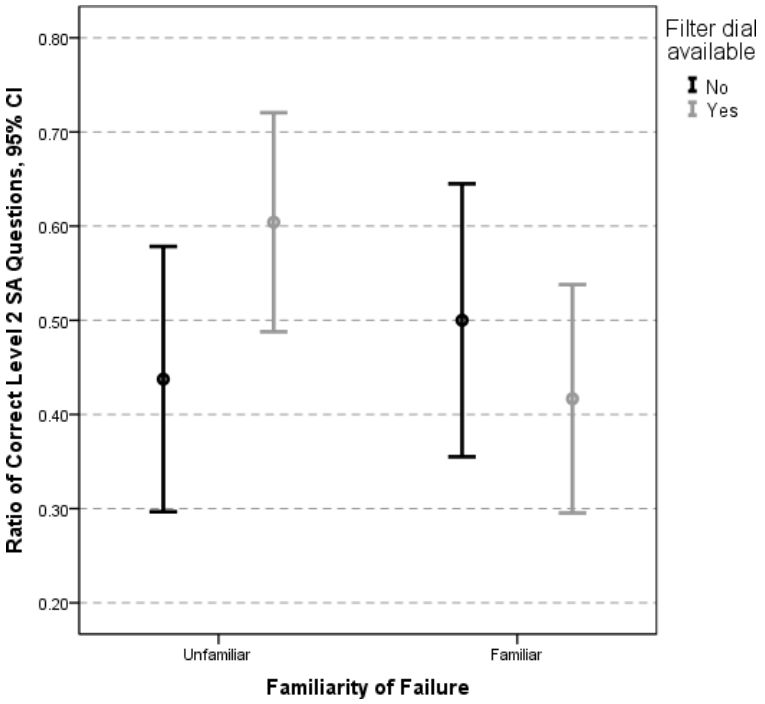


Figure 2.11: Effect of the APA variation and scenario familiarity on the results of the Level 2 situation awareness (SA) questionnaire. Shown is the percentage of correct answers and the 95 % confidence intervals.

2.6 Discussion

This experiment sought to measure pilots' performance in handling emergency situations when using an APA. The two primary measures of performance were the time required to select an alternate landing site and the quality of the landing site which was selected. It was expected that pilots would have to make a trade-off between these two factors. That is, to make a better decision, some pilots may more thoroughly consider their options, resulting in a long time to complete the task. On the other hand, some pilots may choose to act quickly, without investigating all available options, resulting in a lower quality decision. The dial was expected to reduce this negative correlation by reducing the number of options pilots considered. However, the results showed that the correlation between these two measures was, in fact, positive. This suggests that runs in which the pilot took longer to complete the task actually resulted in poorer landing site quality. Though contrary to

Table 2.1: Modified Cooper-Harper Rating Scale for Displays.

Rating	**Description**
1	Excellent & Highly Desirable
2	Good with Negligible Deficiencies
3	Minor but Tolerable Deficiencies
4	Moderately Objectionable Deficiencies
5	Very Objectionable Deficiencies
6,7,8	Deficiencies Require Improvement: Major Deficiencies
9,10	Mandatory Redesign: Major Deficiencies

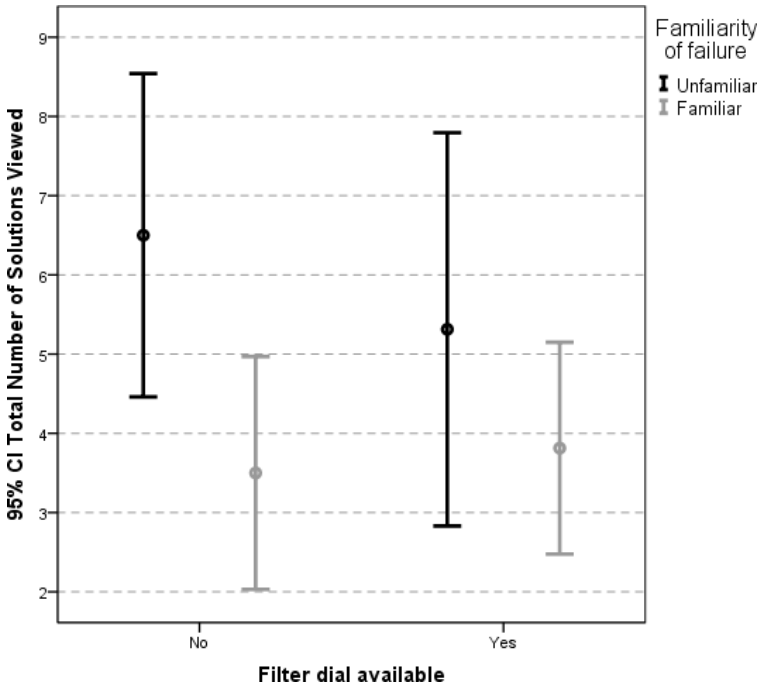


Figure 2.12: Interaction of APA type and familiarity of failure on total number of solutions viewed.

expectation, a number of plausible explanations exist. First the scenario design may have affected this relationship; specifically the number of sites which are similar to the highest ranked site. This similarity may have made some scenarios more difficult than others, requiring more time to consider the options and to differentiate between the highest ranked sites. Second, the benefit of automation may decrease as time pressure is relaxed, suggesting that automation may be less beneficial in lower time pressure scenarios [101]. Third, providing individuals with tools to aid with analytic decision making may result in increased decision time without the associated improvement in decision quality.

A difference in performance was expected for familiar and unfamiliar scenarios. The results supported this hypothesis and showed that pilots made their selections more quickly for familiar scenarios. This supports the results of the pilots' survey in [182], which showed that pilots were more likely to take immediate action in a familiar emergency. This result is also in line with the theory of recognition primed decision making [112] wherein experts upon recognizing a situation immediately understand the implications and are able to make decisions quickly. The effect of the addition of the dial was affected by the familiarity of the scenario. The time to complete the task was increased by the addition of the dial in the case of familiar emergencies, while the time was decreased by the dial in the case of unfamiliar emergencies. This may suggest that the dial proved more useful for filtering out inappropriate options in unfamiliar emergencies. Each participant used his own method to make the APA most useful. However, there were some comments that shed light on how the pilots used the APA. One point of interest is how pilots begin to narrow down the list of possible options. When the dial was available, the first step may be to adjust the filter to an appropriate setting.

In debriefing after the fact, most participants indicated that the ability to filter out unnecessary information and view more detailed information about each landing site were the two most useful features. Pilots emphasized that a key attribute of these features was the speed with which infor-

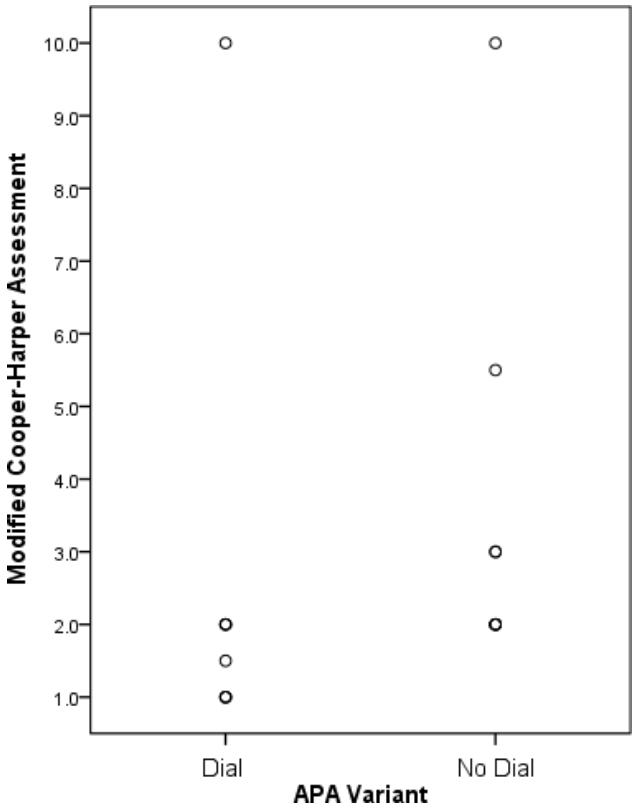


Figure 2.13: Modified Cooper-Harper Ratings of the APA.

mation could be processed. A number of pilots commented that the "ALL FIELDS" dial setting (compare Fig. 2.3) was not useful. The ability to obtain critical information quickly was emphasized by pilots who suggested that more information should be encoded into the graphical display. The addition of a filter setting to show only sites with runways of sufficient length for the aircraft (as currently configured) was suggested by multiple participants. Finally, pilots commented that an APA-type aid should be linked with airline dispatchers. This ability could be used in a number of ways, such as live updates of airline preferred and weighting of evaluation criteria.

Pilots were asked to describe the types of scenarios in which an APA would be most useful, and those in which it would be the least useful. Most pilots expressed that the aid would be most useful in situations where there was high workload and high temporal pressure. These situations are characterized by the need to make a decision quickly. The ability to quickly access large amounts of pertinent information makes the aid a large improvement over current options. Pilots also commented that an APA would be useful in a less intense emergency, in which the aircraft is unable to reach its destination, but the situation is otherwise normal. The aid can be used more deliberately to assess all options and determine the most suitable landing site. The situations in which the aid would not be useful were varied. One participant responded that the aid would not be useful in dire emergencies such as a fire because, "the only piece of information necessary is the nearest runway, all other data is irrelevant." However, another pilot did not identify a situation in which the aid would be least useful, but commented that, "information is always useful in formulating a plan, the more info, the better."

Several caveats regarding realism and training apply to this work. First, while all of the pilots took

the experiment seriously, the experimental conditions could not replicate all of the experiences of an actual in-flight emergency. The pilots were offered additional compensation for improved performance, but this does not entirely replicate the stress and pressures of a real emergency. Second, the pilots were given two training runs, which allowed them to familiarize themselves with the features available. While this was sufficient for the pilots to gain a general understanding of the APA's functionality, if fielded pilots would be trained to a much higher standard on the device.

2.7 Conclusion

This work has sought to evaluate the efficacy of an automated path planning aid (APA), intended to help pilots plan a safe trajectory to land in the event of an in-flight emergency. A prototype was designed and implemented in a cockpit simulation. This simulator was used to test the aid and gather results and further feedback from pilots. The aid which was developed had to be compatible with existing cockpit designs. The aid was designed to be easy to use, without requiring unnecessary time and effort on the part of the pilot. A filter dial was added to allow the pilot to quickly focus only on alternates that were appropriate for a given emergency.

Comparisons between the two variations of the APA showed that the addition of the dial resulted in a small difference in the quality of landing site selected and longer times to select a site. The dial did not significantly reduce the number of alternates viewed, which was strongly correlated with the time metric. This may indicate that the dial did not simplify the task as much as anticipated. However, in the case of unfamiliar emergencies, the dial reduced both time to select a landing site as well as the number of solutions viewed. Every participant scored the variation with the filter dial more highly than the variation without the dial, indicating that they preferred to have the dial, despite the lack of improvement in performance measured in the experiment. The APA was tested using both familiar and unfamiliar emergencies in order to understand if the APA was more useful in one type of scenario than another type. Both survey and simulator results indicated that pilots are likely to act more quickly in a familiar emergency. Pilots found the filter dial and the consolidation of information about landing sites to be very useful features. The ability to quickly and easily access critical information is one of the most important characteristic of an emergency planning aid. This design facilitated the pilots' methods of assessing each landing site throughout a flight, before an emergency has occurred. The ranking system (though not always optimal) gave the pilots aggregate scores for each site and provided a more meaningful starting place when investigating the available options.

Chapter 3

Pilot Feedback for an Automated Planning Aid System in the Cockpit

It is the primary responsibility of the airline pilot to safely complete a flight plan and safely land the airplane. This task can become very difficult in the face of an onboard emergency. One of the challenging tasks faced by the pilots in case of an emergency is the determination of an appropriate landing site as well as the development of a safe trajectory to reach that site. An Automated Planning Aid (APA) can assist the pilot with the tasks of selecting a landing site and developing a suitable trajectory to land. In order to evaluate such an APA, a survey of airline pilots was conducted during the late summer of 2008. The participants were presented with several questions related to the task of planning a path during a performance altering emergency, a non-performance altering emergency and an unforeseen emergency. Participants were also presented with questions about how they would prefer to interact with an APA in the cockpit and the circumstances under which such a device might be most useful. The results of the survey showed that time was the most important criterion to consider, however the methods pilots use to complete the landing site selection and trajectory development tasks vary with the type of emergency and the pilot's familiarity with the circumstances. The results of the survey are used to understand the mental processes currently used by the pilots to complete the path planning task as well as to provide insights to how an APA could be most useful during an onboard emergency.

3.1 Introduction

Modern air transportation has a very good record of flight safety. When failures do occur in flight, the training and experience of the pilots almost always provide for a safe landing. This is evidenced by a rate of only 1.35 accidents per one million hours flown in 2007 by US air carriers [131]. Despite this excellent record, the pilot's responsibility to land safely in case of an emergency can be very demanding. There are a number of tasks which demand the pilots' attention. Among these is the planning and execution of a trajectory that will result in a safe landing. Moreover, all these tasks must be accomplished in a stressful environment under great pressure [48].

The purpose of this research is to better understanding how pilots currently go about the tasks of choosing an appropriate landing site, and planning a safe path to the ground. Additionally, the research will provide suitable directions for the development of an intelligent flight guidance and path planning tool to aid the pilot in this process. To this end, a survey of airline pilots was

conducted to gather information in support of these goals. In this chapter we will first present a review of research related to the task of emergency path planning. Afterwards, the results of the survey will be presented and an analysis of the results will be given. Finally, conclusions will be drawn and suggestions for future research will be made.

3.2 Background

The responsibility of the safe completion of each flight rests with the pilot-in-command. In an emergency situation, this can be a very challenging task. The pilots must monitor the aircraft systems, detect and resolve any failures, control an aircraft with possibly degraded performance, as well as coordinate with the cabin crew, airline dispatchers, and air traffic control. In addition to these tasks, the pilots must also plan and execute a trajectory that will result in a safe landing. These tasks are made even more difficult by the circumstances. For example, the pilots may feel a sense of physical danger, or the cabin environment may be a distraction due to smoke, heat or noise. Additionally, the aircraft's performance may be affected, resulting in degraded or inadequate handling qualities. This limits the relevance of past experience to the current situation.

In order to understand the some of the difficulties that these circumstances present, a number of cognitive engineering models have been developed in the literature and are briefly reviewed in this section. The complex nature of the decision making task can be described using the Naturalistic Decision Making (NDM) framework. Zsambok [194] describes NDM as, "the way people use their experience to make decisions in field settings." Experts are often able to make excellent decisions based on experience and intuition. However, stress can have negative effects on the decision maker's cognition. The Cognitive Control Model (CCM) describes how the context of the emergency dictates the way in which the planning task is handled [83]. This model describes the degree of control a person has as dependent on the context of the situation. The degree of control is determined in large part by the amount of subjectively available time that the pilot perceives, and the familiarity of the situation [85]. Subjectively available time refers to the amount of time that a person perceives that he or she has available to take appropriate action. The amount of time perceived may depend on the objective amount of available time, the predicted changes in the system, the person's level of arousal as well as other factors. The degree of control is discretized into four control modes: scrambled, opportunistic, tactical, and strategic.

The simplest and most dangerous mode is the scrambled mode, which generally represents a person in a state of panic. When a pilot is in this mode, he or she is effectively paralyzed and actions are not part of a plan, and may be unpredictable or irrational, resulting in a lack of control. When a pilot has adequate subjectively available time, his situation may be described by the tactical mode. In this mode the pilot has a greater sense of control. The pilot is more likely to develop a plan or modify an existing plan in order to fit the current situation. The resulting plan may take into account the potential effects of candidate actions. This mode corresponds to "normal" performance. During an emergency situation, a pilot's control mode will be somewhere in between the scrambled and strategic modes, described by the opportunistic mode. In this mode, pilots may use the plans and procedures available to them; however, they may not be used correctly.

The amount of subjectively available time that a pilot perceives may be influenced by a number of factors. These include phase of flight during which the emergency occurs, the type of emergency, the number of actions the pilot is required to complete, and the stress level. This stress may be physical, such as smoke in the cabin or loud noises, or it may be psychological, such as the fear of

impending danger.

The stress of a situation impacts the manner in which the pilot makes decisions. While the pilot may be able to quickly develop a plan of action based on experience and intuition, stress can lead him to fixate on one solution, and fail to compare alternatives [22, 108]. Additionally, the pilot may simply increase the speed with which he processes information, potentially leading to errors. He may also reduce the amount of information that is sought and processed, known as filtration [124, 123].

Stress may also lead the pilot to rely too heavily on an automated tool. He may assume that a plan generated by automation is best, without verifying its feasibility or exploring other options [129]. Also, the pilot may seek only information which confirms the automation-generated solution as the best, while discounting other information (confirmation bias). Alternatively, rather than simply discount conflicting information, he may attempt to mentally force all available information to fit the automation-generated solution (assimilation bias)[62]. These stress-related factors can cause pilots to make poor decisions, despite the fact that they would be able to make acceptable decisions under normal circumstances. These poor decisions may cause incidents to become accidents.

3.3 Review of Accidents

Between 1997 and 2006 there were a total of 89 fatal accidents on board commercial jet aircraft worldwide [9]. More than fourty percent of these accidents categorized as either "Loss of Control - In Flight" or "Controlled Flight into or Toward Terrain." Some of these incidents may not have become fatal accidents if the pilots had been able to quickly plan and execute a satisfactory trajectory in order to complete a safe landing.

One such example is the crash of Swissair flight 111, which encountered smoke in the cockpit during its flight from New York to Geneva (see Figure 3.1). When the pilots noticed the smoke, they declared an emergency. After making an initial turn toward Boston, the controller recommended that they divert to Halifax instead. It was four minutes later when the pilots received the Halifax approach plates and realized that they were too high and needed to lose altitude. So they decided to circle around and dump fuel near Peggy's Cove, Nova Scotia. After the aircraft had turned away from the airport to dump fuel and lose altitude, the fire spread and disabled a number of aircraft systems, which led to the aircraft crashing into the water. It is possible that if the pilots had initially recognized Halifax as the most appropriate landing location and been able to quickly devise a trajectory to land there, then the plane may have been able to land before the fire disabled the avionics systems.

As the previous example illustrates, once an emergency has occurred on board a civil airplane, the selection of a landing site and a safe trajectory to that site are of critical importance. The pilot's high workload and a human's limited computational capacity obviate the need to provide automated assistance. However, the highly complex nature of the selection and planning tasks, as well as the uniqueness of each emergency, makes automation difficult. The input and oversight of a human operator is required.

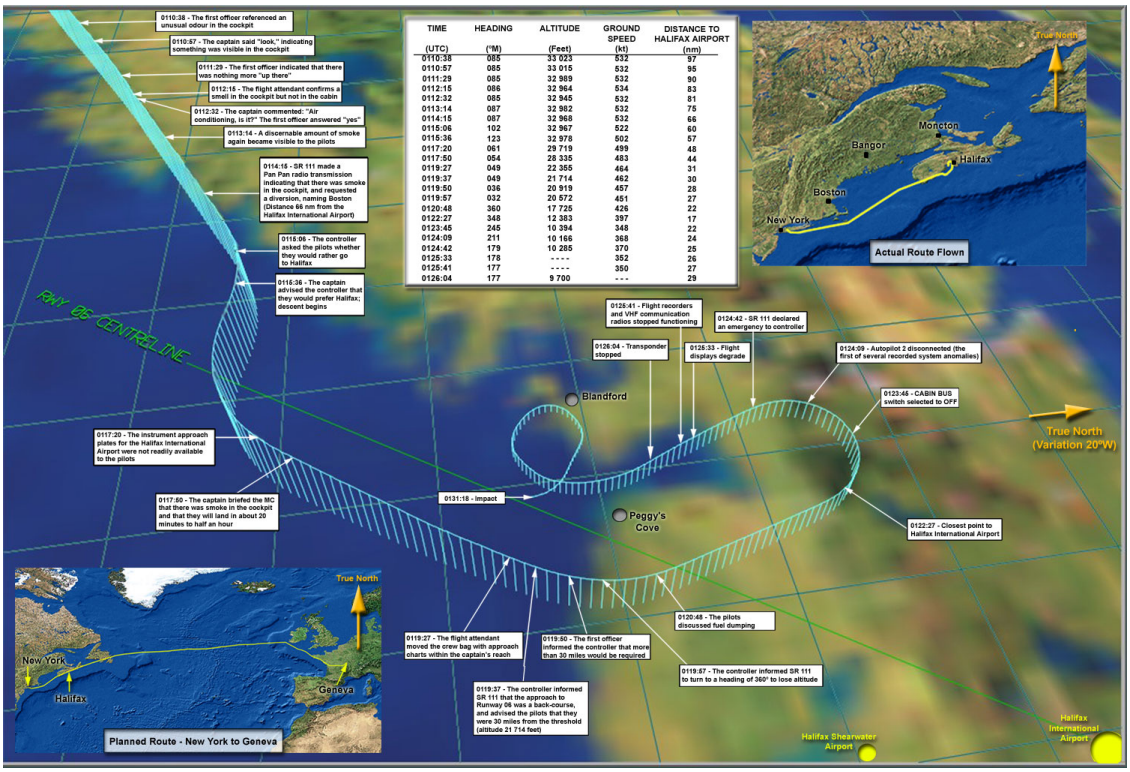


Figure 3.1: Flight path of Swissair Flight 111. Taken from [7].

3.4 Related Work

In order for an automated planning aid to be most useful, there are at least two primary tasks which it must be able to accomplish; first, it must be able to accurately predict the most appropriate alternative landing site, as well as the most efficient trajectory to land at that site. The completion of this task requires that the aid determine the overall feasibility of a trajectory, avoiding the case of a controlled flight into terrain because of not reaching the destination, or overshooting it. A feasible trajectory must also avoid obstacles, which may be static, such as a mountain, or dynamic such as a severe weather system. The consideration of static obstacles avoids controlled flight into terrain, whereas the consideration of dynamic obstacles avoids accidents as a result of flight into convective weather. The determination of such a trajectory must be made by taking into account the aircraft's abnormal aerodynamics. Finally, the trajectory must minimize the time to land, which is important in all cases, such as the aforementioned Swissair accident. The second task an automated planning aid must be able to accomplish is to provide an interface with the pilot through which information is shared in both directions. Most research to date has primarily focused on one or the other of these tasks.

The landing site selection task has been suggested as a candidate for automation. Atkins, Portillo and Strube [14] have developed a method to complete this task. First, the footprint containing all feasible landing sites is calculated. Then the list is prioritized according to a number of weighted criteria, such as runway length, airport facilities available, etc. In their research, the authors choose example values for the criteria weights, but acknowledge that the criteria weights would ultimately be based on expert knowledge and would vary by emergency type.

The need for the pilot and the automated planning aid to interact with each other has also been investigated. The Emergency Flight Planner (EFP) by Chen and Pritchett [48] has been proposed as a prototype interface between the pilot and an aid. The EFP is designed to allow the pilot to easily enter a plan. The ensuing trajectory is then predicted and evaluated. The EFP also provides an additional mode, in which the pilot is presented with a preloaded trajectory, which he may then accept, modify, or delete. The results of testing with the EFP emphasized that generated plans must incorporate the structure and objectives used by pilots in order to be effective.

Research by Layton, Smith and McCoy [116] in their study of a cooperative problem-solving system for en-route flight planning investigated three possible system modes of pilot-system interaction. In that study, pilots and air traffic controllers were both used as subjects. The study evaluated three possible modes. The first mode was a sketching-only system, in which a plan devised by the subject was evaluated by the system and feedback was provided. The second was a sketching system with the additional capability for the user to specify constraints on the plan and allow the system to propose a solution which matched those constraints. In the third mode, the system proposed a plan based on system-specified constraints. The results showed that in the second and third mode, users explored more possible options; however they were also biased toward the system-generated alternative. The same study also highlighted the fact that the use of a fully automated aid could be detrimental if it performs suboptimally.

The previous results show that in order to increase the usefulness of an automated planning aid, the process by which pilots select an alternative landing location and plan a path must be better understood. In addition, it must be understood how the pilot can best be aided by such a tool. It is expected that an aid that accepts and provides information in a manner that is most consonant with the pilot's mental process will be most effective. The results of this research will be useful for the design of a suitable interface by determining the most important inputs to the system, as well as the most useful format for the output.

3.5 Commercial Airline Pilot Survey

A survey was conducted by the authors in order to better understand the tasks and priorities of pilots during an emergency situation. The first section of the survey was intended to elicit information about the primary factors that pilots consider in the process of planning a landing trajectory. This involves choosing the most appropriate destination at which to land, and then determining a trajectory to reach the ground safely. The trajectory planning task also requires attention to certain en route considerations, such as severe weather and hazardous terrain.

The first section of the survey was structured to cover two general types of emergencies: 1) a performance altering scenario, in which the aircraft's performance was non-nominal, and 2) a non-performance altering scenario, in which the aircraft's performance was normal, but an immediate landing was necessary. For the non-performance altering scenario, the participants were presented with the following information:

> You are the captain of a civil transport aircraft. A fire has been detected in the cargo hold. The appropriate checklists have been performed, but the fire has not been completely extinguished. The first officer is controlling the aircraft, allowing you to plan a course of action.

For the performance altering scenario, the participants were presented with the following informa-

tion:

> You are the captain of a civil transport aircraft. The right engine of your twin-engine aircraft has failed. The appropriate checklists have been performed. The first officer is controlling the aircraft, allowing you to plan a course of action.

The same set of questions was used in each of the two scenarios. The performance altering scenario also included an additional question, which addressed how the pilot would assess the feasibility of a trajectory given the aircraft's degraded performance.

The second section introduced the concept of an Automated Planning Aid (APA). The questions built upon the performance altering scenario from the first section, with the following additional information:

> Now an Automated Planning Aid (APA) is available to assist you with the selection (and perhaps execution) of a suitable plan of action.

This section was intended to obtain information about how the participants might use an APA. In particular, how participants prefer to convey information to the APA and furthermore how they prefer to review the information provided by the APA. Finally, this section presented questions meant to ascertain the amount of confidence that participants would have in the APA.

The third section was designed to collect further information about how an APA might be used. The participants were presented with the following information:

> Consider an emergency scenario which is unforeseen (i.e., you have not received any pertinent training). The aircraft's performance is now altered and/or degraded in some way. You are the captain, and the first officer is controlling the aircraft, allowing you to plan a course of action. In this scenario you do not have an Automated Planning Aid (APA) available to assist you.

This scenario was included because it provides some insight into how the participants will make a plan in a situation where they cannot rely on any prior training or procedures to guide them through the process.

The final section included general questions about the participants' opinions of the proposed APA concept. These questions asked about the scenarios under which the participants would be more willing or less willing to seek help from an APA and how the participants would like the plan to be updated. This section also included biographical questions in order to determine the demographic make up of the participants.

3.5.1 Methodology and Participants' Profile

The survey was conducted using an on-line service. The service was used to create, format, and monitor the survey. It was also used to host the survey and collect responses from the participants. In order to generate responses from several airline pilots, a link to the online survey was distributed via email to a number of pilots. In addition to the email, a link was posted on the airline pilot association's message board requesting participation. Responses were collected over the course of approximately six weeks between August and September 2008.

Responses were received from twenty-one participants. One of the respondents declined to include biographical information, however all twenty-one responses were used in the results. The demographic analysis therefore only includes twenty respondents. All participants held the position of either captain or first officer and had been in their current position for an average of 9.5 years. Eighty-five percent were flying a Boeing aircraft (737, 757, 767, or 777) at the time of the survey. All pilots had at least 6,500 flight hours with an average of 12,979 flight hours.

3.6 Results

Due to the small number of responses, slight variations in the number of responses for a given option were neglected as statistically insignificant. However, each question also included an open-ended option where the participants were free to provide more information. These responses often provided additional valuable insights into the participants' thoughts that could not be captured by the multiple choice responses provided.

The first section of the survey included a set of questions about how pilots currently make decisions in an emergency situation. The question set was first given for a non-performance altering scenario, and then repeated for a performance altering scenario, as described previously. Participants were asked to indicate the priority (high, medium, low, or not a consideration) associated with a number of factors when choosing the airport to which the respondent would divert. Under the non-performance altering scenario, the most important factor indicated was the proximity of the airport in terms of time. Weather conditions at the airport, the length of the runway and the proximity of the airport in terms of distance were also given relatively high priority. These results can be seen in Figure 3.2. Under the performance altering scenario, the results were largely the same; however the importance of proximity in terms of time was not differentiable from that of other factors, as seen in Figure 3.3. One free response comment for the non-performance altering case also indicated that runway lighting and the availability of navigational aids were additional important considerations.

A similar question was posed for each scenario in which participants were asked to indicate the priority (high, medium, low or not a consideration) associated with a number of factors when planning a safe path. For both scenarios, en route weather and the avoidance of hazardous terrain were given the same priority, and low priority was given to traffic routes. This is not surprising because once an emergency is declared, the pilot need not comply with ordinary routes and approach procedures.

The performance altering scenario differs from the non-performance altering scenario in that the pilot's experience and knowledge of the aircraft may have limited applicability to the current situation. For this reason, the pilot's "first instinct" may be the best plan given normal performance, but may not be feasible given the degraded capabilities of the aircraft. Participants said that they were most likely to judge the feasibility of a maneuver by running the scenario mentally or seeking help from the dispatcher. Many pilots would also consult the performance manuals. These responses indicate that an automated planning aid may be particularly helpful in situations where the aircraft's performance is not normal.

Participants were then asked to consider whether they would completely determine a plan before taking any action. The alternative would be for the pilot to alter the current course immediately based upon his "first instinct." Many of the respondents took advantage of the open-ended option to describe some other considerations that affect how they would proceed. For instance, when flying

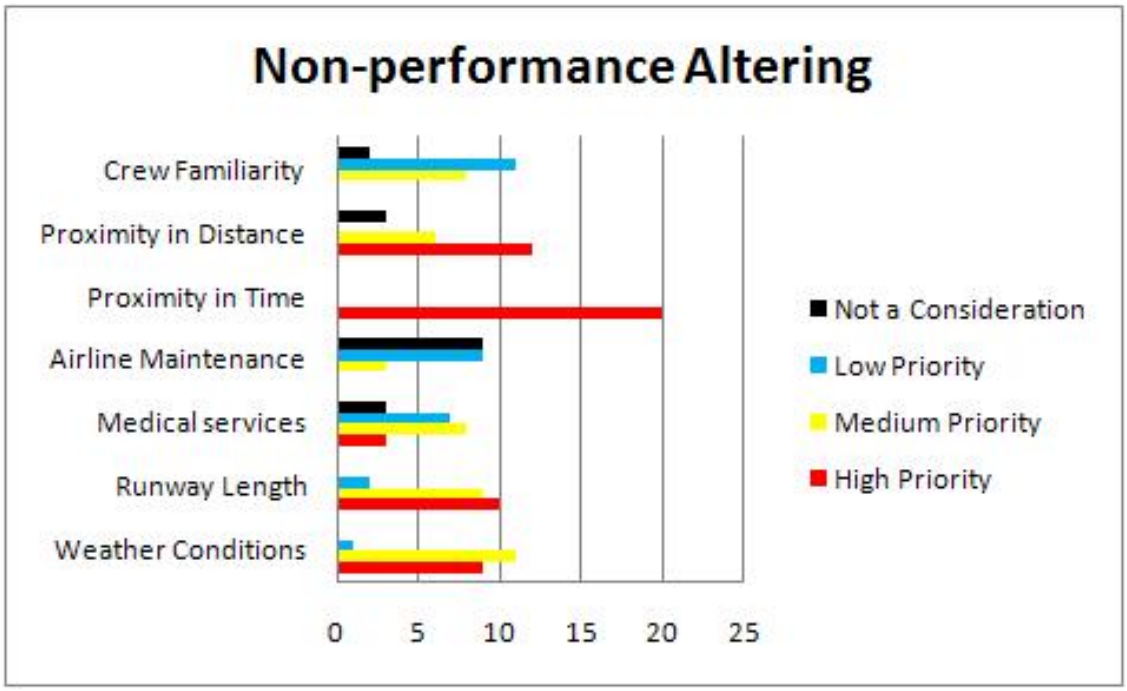


Figure 3.2: Prioritization of landing site criteria for non-performance altering case.

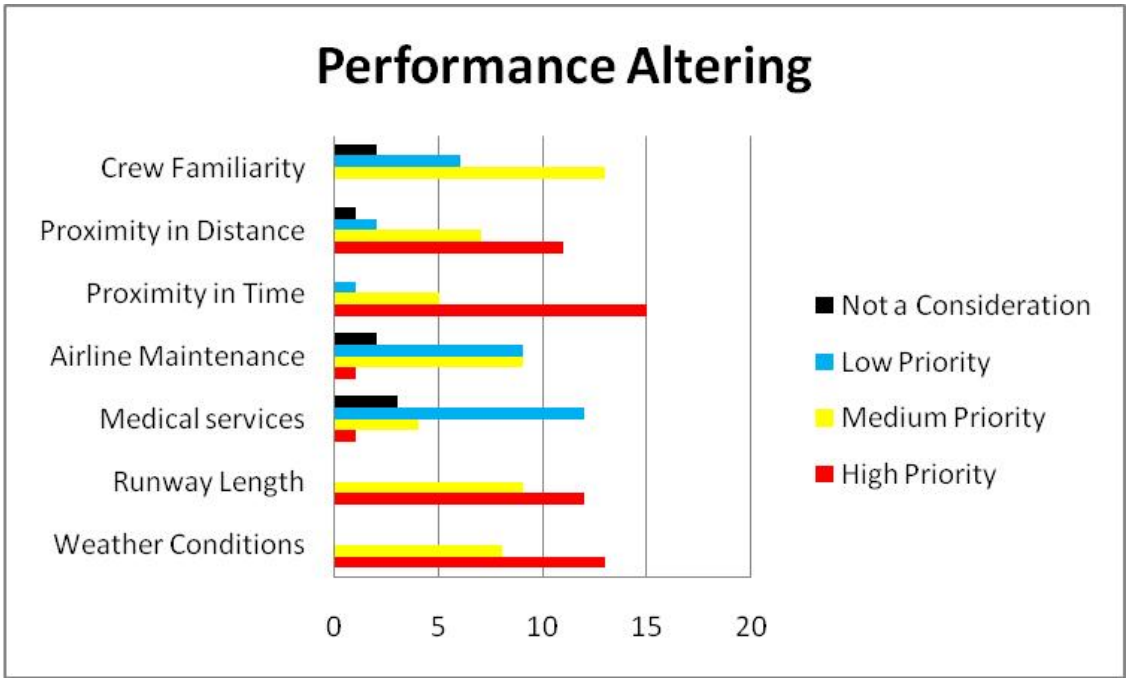


Figure 3.3: Prioritization of landing site criteria for performance altering case.

in a mountainous area such as Quito, Ecuador, planning ahead is essential. Others indicated that they would first coordinate with Air Traffic Control (ATC) before taking action or completing a plan of action. Those who would first take some action responded that they would all turn toward the nearest airport. None would begin descending. For the non-performance altering scenario, most of the pilots reported that they would change the current course immediately. However, in the performance altering scenario, most would develop a plan first before altering the current course. This may indicate that pilots are more comfortable taking immediate action in a more familiar situation, as opposed to a novel scenario. For instance, pilots have trained for an engine failure scenario and would have a relatively good idea of how to control the aircraft. However, in the event of a control surface malfunction, they would not be as familiar with the aircraft's post-failure performance and may be less likely to take immediate action before planning.

The final question of this section addressed the parts of a potential plan with which the participants would be most careful (i.e., provide more specific attention, add more detail, etc.). The pilots were asked to indicate the priority (high, medium, low, not a consideration) associated with portions of the plan. The results are shown in Figures 3.4 and 3.5. In both scenarios, pilots reported that the highest priority is around severe weather and hazardous terrain. Medium priority was given to the area around the aircraft's current location. In the case of the performance altering scenario, medium priority was also given to the area around the destination.

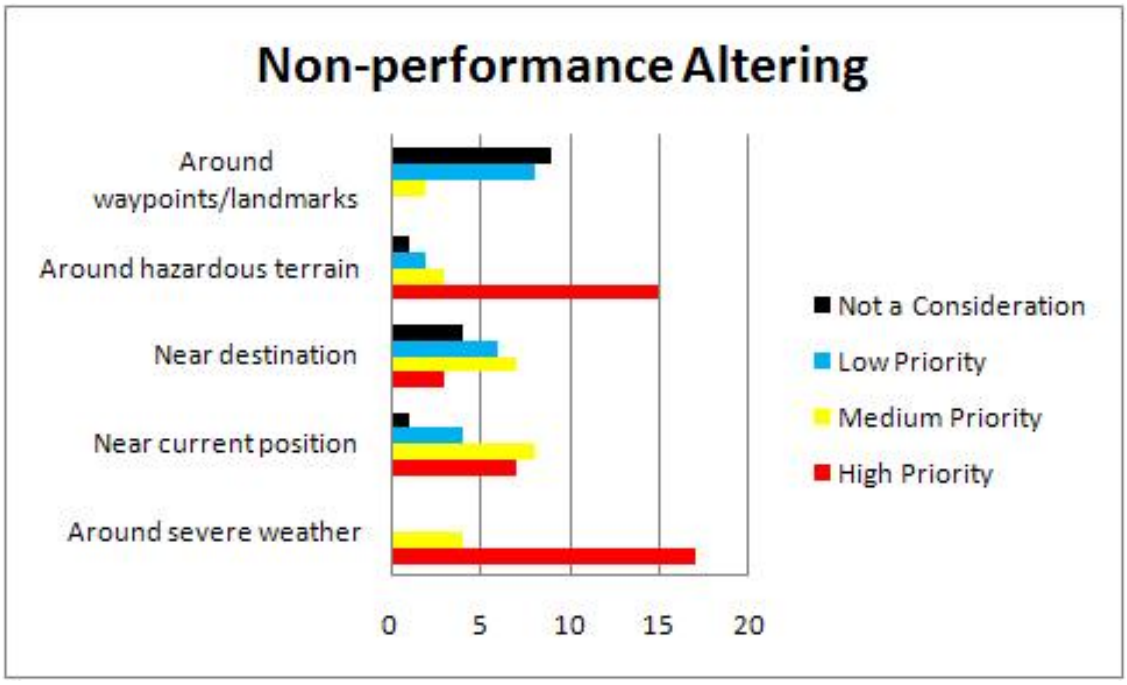


Figure 3.4: Parts of the plan pilots consider with detail for non-performance altering case.

The first questions of the section dealing with an Automated Planning Aid (APA) address the interface between the pilot and the APA. Pilots described a number of inputs as either highly preferable, somewhat preferable or not preferable. Pilots will need to be able to provide priority information to the APA in order for it to be aware of the current situation. Participants indicated that they would prefer to accomplish this either through the Flight Management System (FMS) pages, through a separate dedicated interface, or through a data link from the airline operations center. When the APA has developed a plan, the pilot must be able to review this plan. The

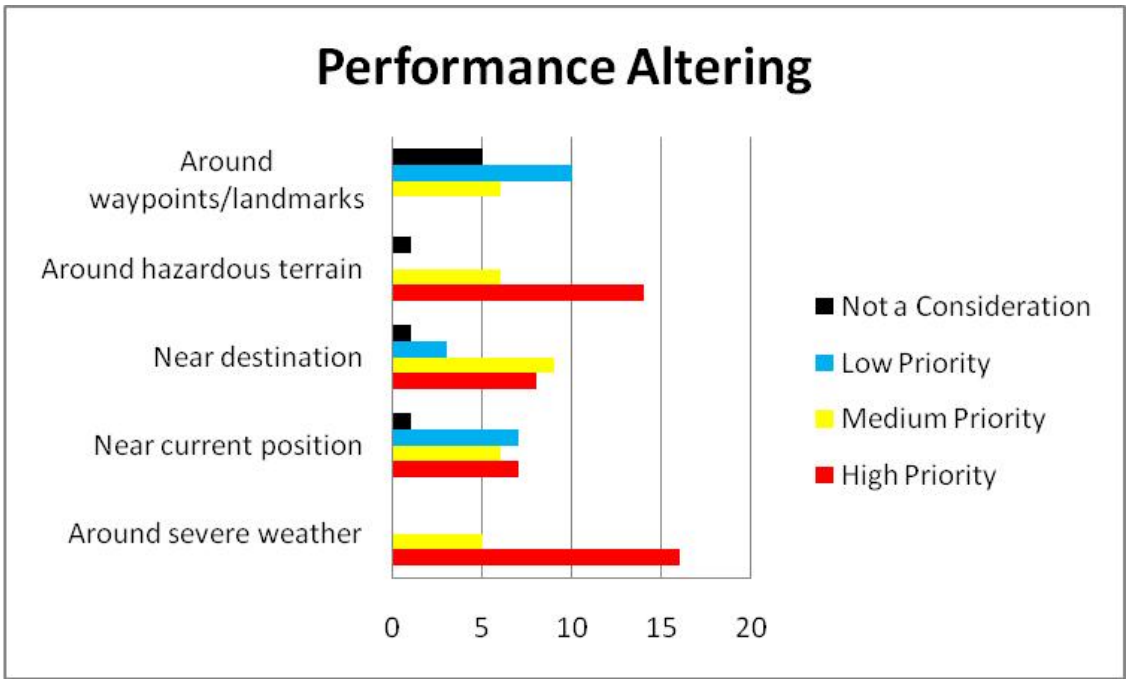


Figure 3.5: Parts of the plan pilots consider with detail for performance altering case.

respondents indicated that they would prefer to review the plan as a set of automatically generated FMS entries. The pilots would also favor a horizontal graphical representation of the proposed trajectory, possibly accompanied by a vertical profile of the proposed trajectory.

In order to effectively evaluate the proposed plan provided by the APA, pilots will have certain metrics in mind which will be used to make the evaluation. Participants were asked to indicate how important certain metrics are when evaluating the plan (highly important, somewhat important or not important). The most important metric was the cumulative time/distance and fuel information. A comparison against alternatives was also considered an important metric.

Responses showed that the pilots will not unconditionally follow the plan generated by the APA, especially if the plan if different from their "first instinct." Most participants said that they would follow an APA-generated plan that was different from their own when the APA plan required a significantly shorter amount of time to execute. Some pilots also said that they would follow the APA plan if it remained well within the flight envelope limitations and encountered significantly less severe weather. Nearly all of the pilots indicated that they would only use the APA-generated plan as an aid; that is, they would take it into account while re-evaluating their own plans of action, but would neither completely accept nor reject an APA-generated plan.

In the final section, the participants were given an unforeseen emergency, as described previously. As in the performance altering scenario, the majority of respondents would completely develop a plan before altering their course. Many participants again took advantage of the open-ended response option to indicate that the primary factor in deciding whether to take immediate action would be the urgency of the situation. Of those who would immediately alter the current course, most would turn toward the nearest airport.

In a situation in which the pilot has accepted the plan, but has deviated from it, a new, more efficient, plan may be calculated as a result of the deviation. Pilots were asked to choose between a

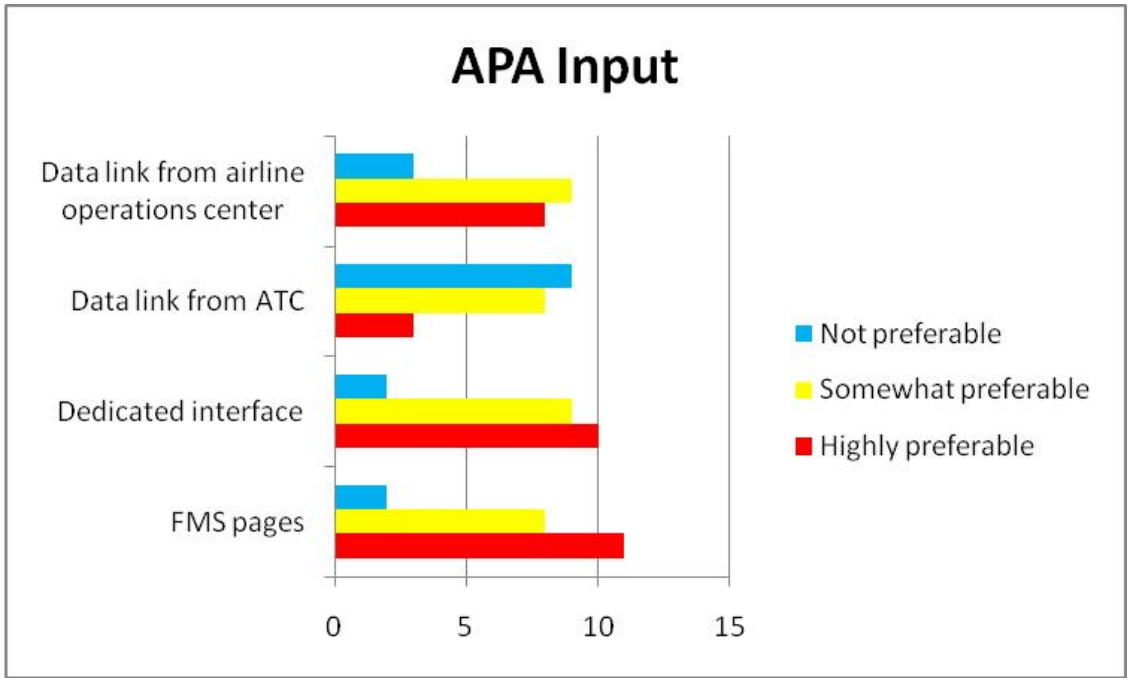


Figure 3.6: Preference for APA method of input.

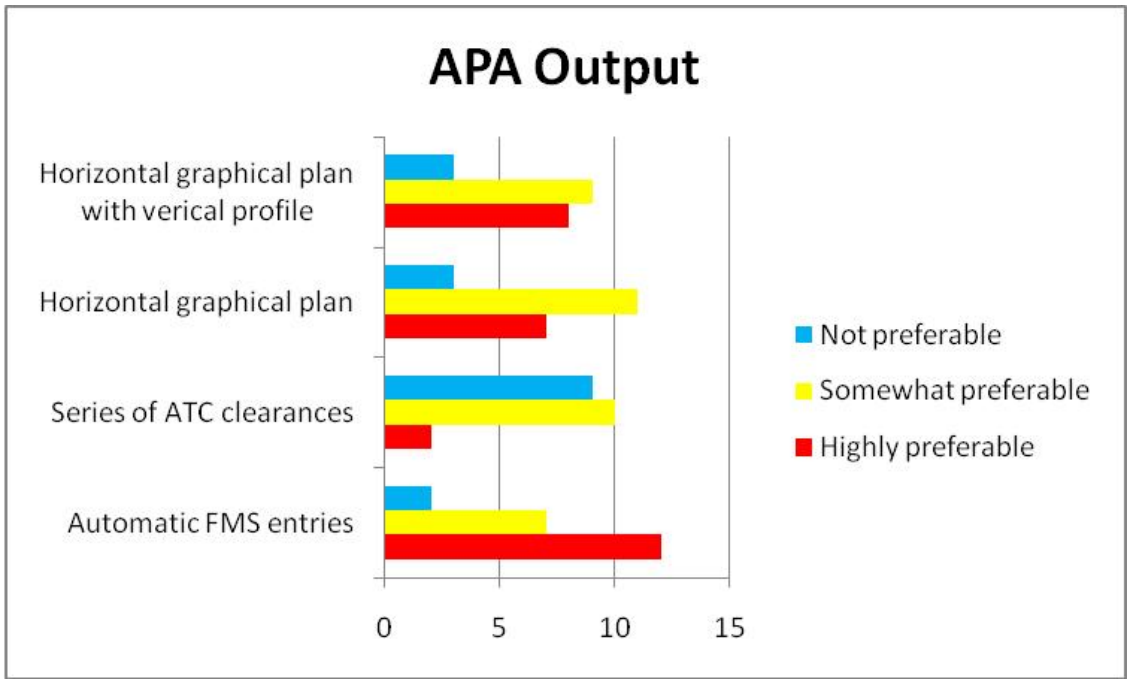


Figure 3.7: Preference for APA method of output.

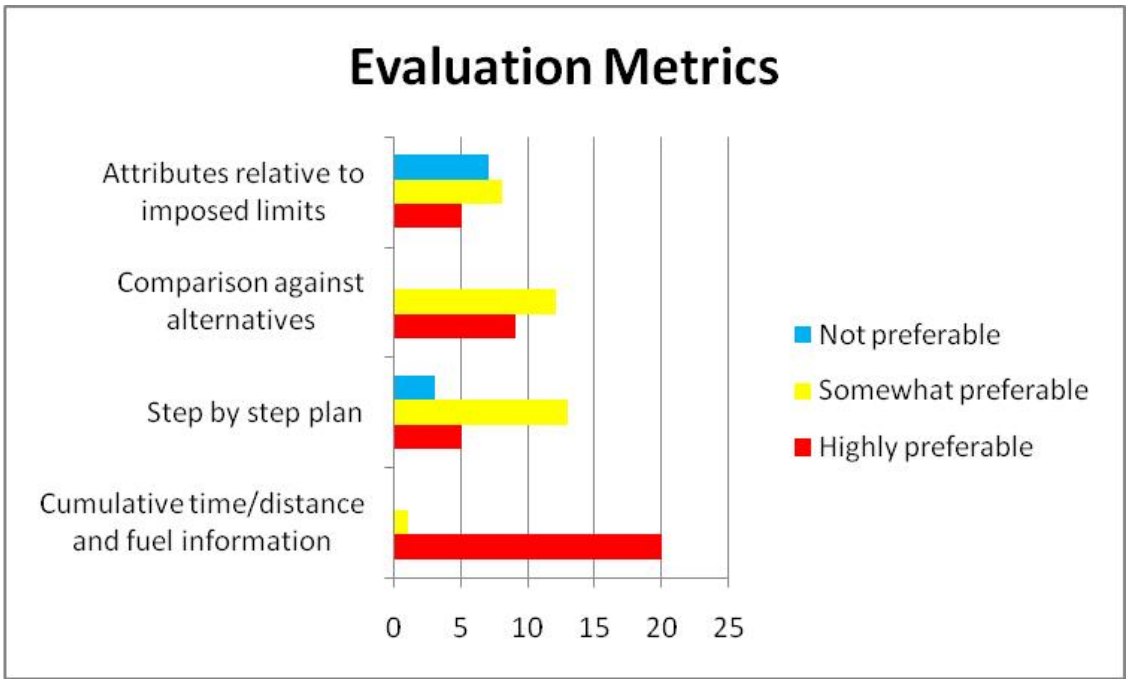


Figure 3.8: Metrics used to evaluate a plan.

number of conditions under which they would like for the APA to provide a new plan. The results are shown in Figure 3.9. Most pilots indicated that they would prefer to only receive a new plan from the APA when they asked for one.

Not surprisingly, the majority of pilots reported that the situation in which they would be most likely to rely heavily on the APA is one for which they have not had any prior training or experience, as shown in Figure 3.10. Some indicated that they would not rely on the APA in either a familiar or unfamiliar situation, while others said that they would rely on the APA in both situations. This may be due to the lack of clarity with regard to reliance. It seems, based on these comments, that many of the pilots would use the APA as an aid, but they would be hesitant to simply follow its plan without some verification of their own.

3.7 Analysis

The multiple choice nature of the responses made the mathematical results simpler to discern; however, the more enlightening portion of the responses were the open-ended options, which allowed the participants to include their thoughts on each of the questions provided. These responses provided insights into the pilots' expectations about how an APA should function and how it could be most useful.

As mentioned previously, the pilot's workload is very high during an emergency. A number of comments indicated that pilots are wary of any factor(s) that would necessitate more work during a stressful time. This is reinforced by the respondents' preference for interacting with the APA through the FMS, a device they use for other purposes, and with which they are familiar. Most participants preferred to review the plan as automatically-generated FMS entries, which would allow the autopilot to follow the APA path with very little additional work required by the pilot.

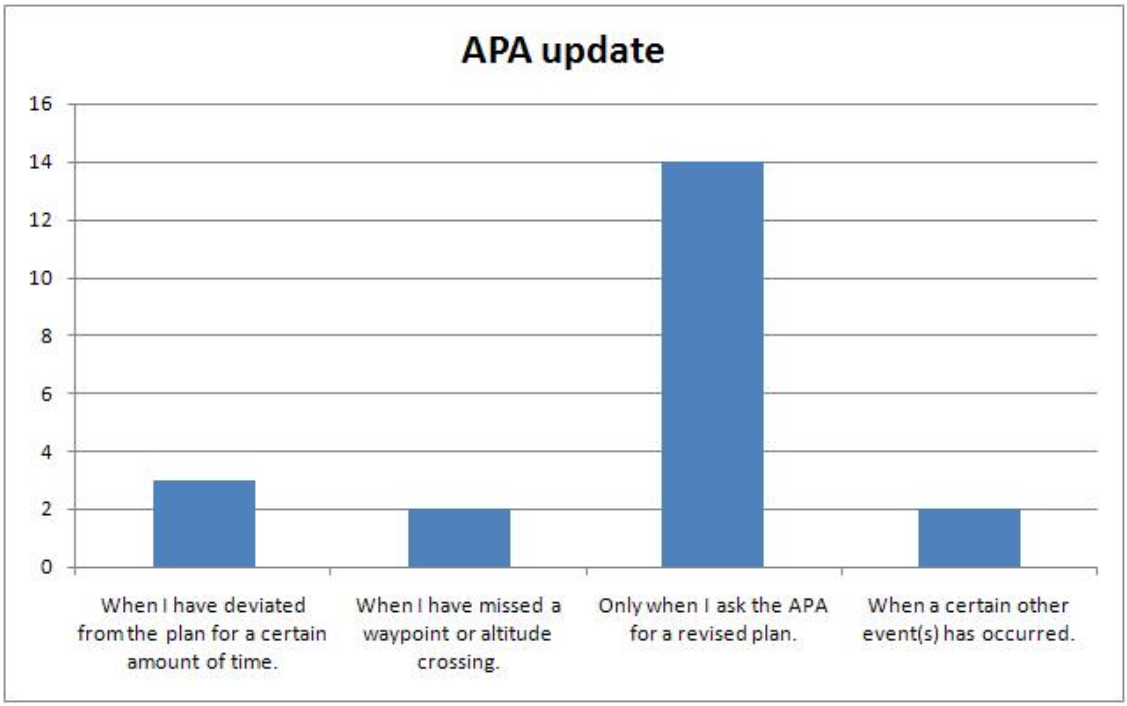


Figure 3.9: Conditions under which the APA should provide an updated plan.

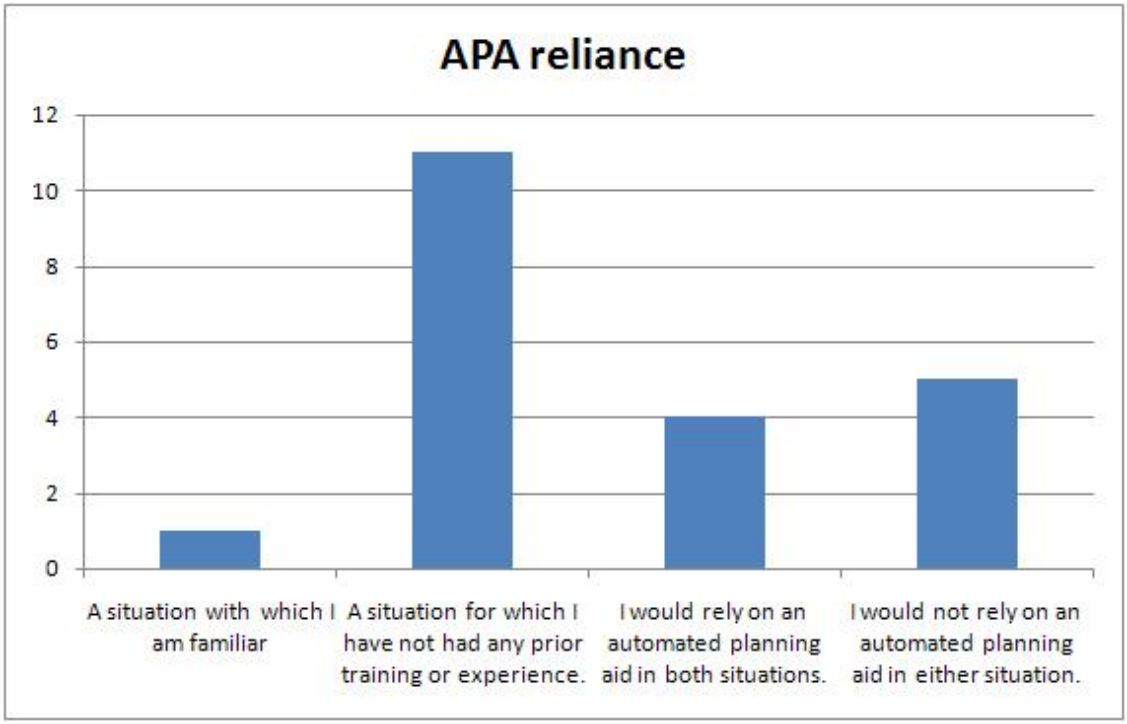


Figure 3.10: Situations in which an APA would be most heavily relied upon.

Respondents emphasized that the APA should be an efficient source of information which is currently disparate, if available at all. One pilot commented that, "it should offer information, but not demand any acceptance or response."

In general, commercial pilots have a good working knowledge of the areas in which they normally fly. For this reason, the pilots' "first instinct" is often very good. The results concur with this conclusion, as evidenced by the fact that pilots were more likely to alter the current course without a complete development of a plan in the non-performance altering case and much less so in the unforeseen case. A number of comments revealed that in some cases pilots simply need a tool to validate their plans and point out to them any options that they may have missed. A tool such as this may have been helpful in the case of Swissair flight 111, which initially turned toward Boston when an emergency was declared despite being closer to Halifax, Nova Scotia [7].

When determining the best landing site, as well as the best path to that site, a number of pilots found the list of factors provided to be insufficient. Certainly, the factors mentioned are important, however, some comments emphasized the reliance on outside sources. Once an emergency has been declared, pilots work very closely with Air Traffic Control (ATC) to receive their input to determine the most appropriate path. Pilots' comments also indicated that they will seek advice from the airline dispatcher in order to determine the most appropriate landing site. The scenario for which the pilot has not had any training or prior experience garnered a number of additional comments. These emphasized the interactive nature of the planning task by pointing out that the process must include ATC, airline dispatcher, other crew members, and possibly the manufacturer.

Many comments addressed the role of the APA in the planning process. In addition to keeping the workload as low as possible, many pilots do not want to view an APA-generated path as a directive. Rather, they prefer to view it simply as one input into the process of developing their own plan. This supports the result seen previously in the literature, in which nearly all participants said that they would take the automated plan into account while re-evaluating their own plan. The comments emphasized that the path planning task is complicated and that the automated tool may not have the ability to gather a complete understanding of the situation at hand. These comments imply that pilots would like to have an APA that makes critical information easy to access in a timely manner, but which does not dictate actions for the pilot to follow.

Due to the uniqueness of each emergency, one respondent proposed an approach which would likely be supported by other pilots. The pilot said, "You should be able to manipulate individual variables and compare solutions." The priority of certain criteria may change, depending on the emergency, and pilots need to be able to indicate this to the APA. For instance, airport fire and rescue services are more important in a fire emergency. Runway length may be more important in a flap or landing gear malfunction. The recognition that there are a number of variables which must be taken into account was echoed by a number of comments. Also, the ability to compare alternatives was considered an important metric.

A fundamental requirement of an APA is that the pilots must trust it and must be willing to use it. One pilot addressed this issue by saying that, "I must know how the automated plan is generated to be able to trust its output. Once I have confidence in the APA planning process, I would be more likely to trust its output, particularly in a time-critical emergency situation." This sentiment would surely be echoed by other pilots who will be hesitant to trust any tool with which they disagree. It is these disagreements that provide the usefulness for an APA; if the generated plan always agrees with the plan the pilot has in mind, then the tool has provided only a very limited service. However, if the pilot and the tool disagree, the tool must be able to demonstrate to the pilot that his plan can be improved upon. It must also be ensured that the pilot, working with

	Nonperformance Altering	Performance Altering
Proximity in Time	0.21	0.19
Proximity in Distance	0.17	0.16
Runway Length	0.18	0.18
Weather Conditions	0.18	0.18
Crew Familiarity	0.10	0.11
Medical services	0.11	0.08
Airline Maintenance	0.05	0.10

Table 3.1: Resulting criteria weights based on survey results

the APA, does in fact generate a better plan than the pilot could on his own. This must be done without causing the pilot to completely rely on the system through biasing or over-reliance [137].

3.7.1 Criteria Weighting

In order for an Automated Planning Aid to develop a recommended path, the automation must be able to develop a prioritization among the possible landing sites. Such a prioritization may be based on the minimization of a utility function, such as Equation 3.1.

$$U = C_1 * \frac{t}{t_{max}} + C_2 * \frac{d}{d_{max}} + C_3 * \frac{r_{l,max} - r_l}{r_{l,max}} + C_4 * w_{wx} + C_5 * w_{cf} + C_6 * w_{med} + C_7 * w_{rep} \quad (3.1)$$

In this equation, the seven parameters are: time required to land t, distance from the current location d, runway length r_l, weather conditions w_{wx}, crew familiarity with the landing site w_{cf}, medical services available w_{med}, and airline maintenance and repairs available w_{rep}. The time t and distance d are nondimensionalized by their maximum possible values given the aircrafts performance t_{max} and d_{max}. The availability of medical services and airline maintenance are static attributes of each airport which could be encoded on a scale from zero to one. The value for the crew familiarity factor could also be assigned before a flight by the pilots. The weather factor must be determined in real-time, based on the probability and severity of adverse weather conditions.

This leaves the determination of the criteria weights C_i. In the survey, each of the landing site criteria were assigned a priority; high, medium, low, or not a consideration. For this analysis, each of these options was assigned a value, three for 'high', two for 'medium', one for 'low' and zero for 'not a consideration.' The responses were summed based on the assigned values, giving a total for each criteria. In order to normalize these values, each was divided by the sum of the total scores for all criteria, as in 3.2. The subscripts of *score* in Equation 3.2 refer to the score assigned by respondent r for criteria i. The resulting values for the criteria weights are shown in Table 3.1.

$$C_i = \frac{\sum_{r=responses} score_{r,i}}{\sum_{j=criteria} \sum_{r=responses} score_{r,j}} \quad (3.2)$$

This is just one possible utility function. There are certainly other criteria that could be considered. For instance, one comment indicated that runway lighting and the availability of instrument landing system equipment may also be taken into consideration. This survey has provided one possible starting point for the relative weighting of these criteria. However, the determination of the most appropriate weighting and utility function must be investigated further before it can be considered for implementation.

Chapter 4

Multi-resolution Motion Planning via Wavelet-Based Cell Decompositions

According to the control hierarchy that we follow in this work, at the top level we need a geometric path planner to construct an obstacle-free path. This path will later be time-parameterized to generate a trajectory to be followed by the aircraft. In this chapter present a path- and motion planning scheme to design such a path that is "multi-resolution," both in the sense of representing the environment with high accuracy only locally, and in the sense of addressing the vehicle kinematic and dynamic constraints only locally. The proposed scheme uses rectangular multi-resolution cell decompositions, generated efficiently using the wavelet transform. The wavelet transform is used widely in signal and image processing, with emerging applications in autonomous sensing and perception systems. The proposed motion planner enables the simultaneous use of the wavelet transform in both the perception and in the motion planning layers of vehicle autonomy, thus potentially reducing online computations. We prove rigorously the completeness of the proposed path planning scheme and we provide numerical simulation results to illustrate its efficacy.

4.1 Introduction

Motion planning for for aerial and other vehicles has been extensively studied [51, 114]. However, important issues such as dealing with uncertain, partially known, and/or dynamically changing environments, and the satisfaction of vehicle kinematic and dynamic constraints are yet to be thoroughly and satisfactorily addressed, especially when considering additional constraints stemming from the limited computational resources on-board the vehicle.

In this chapter, we present a fast multi-resolution motion planning scheme that guarantees the satisfaction of the vehicle's kinematic and dynamic constraints. To introduce the various interrelated aspects of the proposed scheme, we will use the following terminology: We will use the term *path* to refer to the locus of continuous motion of a point, and the term *trajectory* to refer to a path parameterized by time. Depending on the context, we will use the term *path* to also refer to a sequence of successively adjacent vertices in a graph. Finally, we will use synonymously the terms *workspace* and *environment* to refer to a planar region over which the vehicle moves.

Multi-resolution path planning involves the representation of the vehicle's environment with different levels of accuracy. For example, the popular quadtree method [149, 104, 134], generates a

planar cell decomposition consisting of small cell sizes that capture accurately obstacle boundaries, and larger cell sizes that represent efficiently large areas in the free space. Other path planning schemes that use multi-resolution cell decompositions have appeared, for instance, in [25]; in [92] (triangular cells); in [142] (receding horizon path planning scheme using multi-resolution estimates of object locations); in [111] (multi-resolution potential field); and in [178] (hierarchy of imaginary encapsulating spheres for collision avoidance).

We consider planar cell decompositions such that the environment is represented with high accuracy (i.e., using small cell sizes) in the agent's immediate vicinity, and with lower accuracy in regions farther away, similar to the multi-resolution grids considered in [25, 175]. Multi-resolution cell decompositions are compact data structures that encode large environment maps, and thus enable efficient online path- and motion planning. Furthermore, multi-resolution decompositions of the environment capture naturally the graded levels of uncertainty about the environment as a function of the distance from the current location of the agent. In other words, such decompositions encode the notion that the *uncertainty* or *incomplete knowledge* about the environment is higher in regions farther away from the vehicle's current location.

The discrete wavelet transform (dwt) is a mathematical tool used widely in multi-resolution signal processing [64, 145]. Applications of the DWT to vision-based navigation and vision-based SLAM for autonomous vehicles have appeared recently in [120] (appearance-based vision-only SLAM); in [93] and [50] (local feature extraction); and in [159] (stereo image processing). With the plethora of available sensors, and in light of the fact that multiple sensors are typically used for autonomous navigation [175], the wavelet transform may soon become the common standard for representing and analyzing signals [54]. In this context, wavelet-based data representation for path planning problems has been addressed recently in [187] (occupancy grids); in [183] (standardized representation of road roughness characteristics); in [184] (terrain depiction for pilot situational awareness); and in [138] (image registration for vision-based navigation).

In light of the ubiquitous use of the DWT in signal- and image processing, and its emerging applications in autonomous sensing and perception, it is natural to investigate the seamless integration of sensing and path planning using multi-resolution wavelet analysis. To this end, we propose a path planning scheme that directly uses a DWT representation of the environment. Applications of the DWT in multi-resolution path planning schemes have appeared previously, for instance, in [174, 102] (preliminary implementations of the proposed path planning scheme); in [136] (path refinement based on successively finer approximations of a terrain map); and in [45, 163] (multi-resolution schemes for vision-based path planning).

***H*-Cost Motion Planning:** Motion planning schemes often involve a *geometric path planner* that uses an abstract, discrete representation (e.g., graphs associated with cell decompositions) of the workspace and deals with the satisfaction of task specifications such as obstacle avoidance. However, the resultant geometric path may be found to be infeasible or unacceptably sub-optimal if the vehicle's kinematic and dynamic constraints are ignored. To address this issue, we introduced in [59] a motion planning approach based on assigning costs, called *H-costs*, to *multiple* edge transitions in the cell decomposition graph. These H-costs allow the vehicle's kinematic and dynamic constraints to be incorporated in the geometric path planner via the (implicit) construction of a so-called *lifted graph*, which is closely related to the original cell decomposition graph. In this work, we discuss a multi-resolution implementation of this H-cost motion planning approach, such that the overall scheme is "multi-resolution," both in the sense of representing the environment with high accuracy only locally, *and* in the sense of considering the vehicle dynamical model for path planning only locally.

In summary, the proposed motion planning scheme consists of the following main elements: (a) A wavelet-based multi-resolution cell decomposition algorithm that creates and modifies a graph that represents the environment (see Section 4.2); (b) A local trajectory generation algorithm called TilePlan that associates H-costs in the aforesaid cell decomposition graph to (implicitly) construct a "partially" lifted graph (see [58, 57]); and (c) A discrete path planner that finds paths in the "partially" lifted graph, which, in turn, correspond to trajectories that satisfy the vehicle's kinematic and dynamic constraints (see Section 4.4). The interactions between the various models and methods involved in the proposed scheme are illustrated in Fig. 4.1: here, hollow arrows indicate the creations and modifications of various models by the methods shown, whereas bold arrows indicate the dependencies between the various models and methods shown. For example, the hollow arrow from TilePlan to the "partially" lifted graph model indicates modification of the edge transition costs of the latter.

The main contributions of this chapter are as follows. Firstly, we present a multi-resolution cell decomposition technique that is completely encoded in the DWT coefficients of the environment map. This approach allows for the development of highly integrated, efficient navigation and path planning architectures, where the DWT coefficients are used as a common data structure both for scene understanding and for motion planning. Secondly, we demonstrate one such integrated approach by proposing a path planning scheme based on the aforesaid cell decompositions, and we rigorously prove its completeness. To the best of our knowledge, such proofs of completeness are absent from other similar multi-resolution path planning schemes [25, 175]. Finally, we discuss a method of incorporating vehicle dynamic constraints in the multi-resolution path planning scheme using the H-cost motion planning approach of [59]. The issue of consistency between the geometric and dynamic planning layers is well-known in the robotics community [26]. To date, this problem has been addressed by planning in the (high-dimensional) state space, instead of the workspace, where the obstacles naturally lie. We show that, for mobile robotic applications, planning can be restricted to the low-dimensional workspace. The overall motion planning scheme is thus an important step in the development of an hierarchically *consistent* autonomous navigation and motion planning system, i.e., one that guarantees the satisfaction of vehicle kinematic and dynamic constraints while retaining the computational efficiency of discrete multi-resolution path planning.

The rest of this chapter is organized as follows. In Section 4.2, we describe the proposed wavelet-based multi-resolution cell decomposition technique. In Section 4.3, we describe a path planning scheme using this cell decomposition, and we prove its completeness. In Section 4.4, we discuss the inclusion of vehicle dynamical constraints in the multi-resolution path planner via the H-cost approach. In Section 4.5, we provide numerical simulation results illustrating the successful operation of the overall motion planner.

4.2 Multi-resolution Cell Decompositions using the Discrete Wavelet Transform

Cell decomposition is a common technique used in geometric path planning [51], that involves partitioning the workspace into convex regions called *cells*. A graph is associated with this partition, such that each cell corresponds to a unique vertex and each pair of geometrically adjacent cells corresponds to a unique edge. The original path planning problem is thus transformed to the problem of finding a path in this graph, which can be solved, for instance, by the A* algorithm [114]. In what follows, we introduce a multi-resolution cell decomposition technique based on the 2-D discrete wavelet transform. We refer the reader to [64, 145] for further details on the DWT.

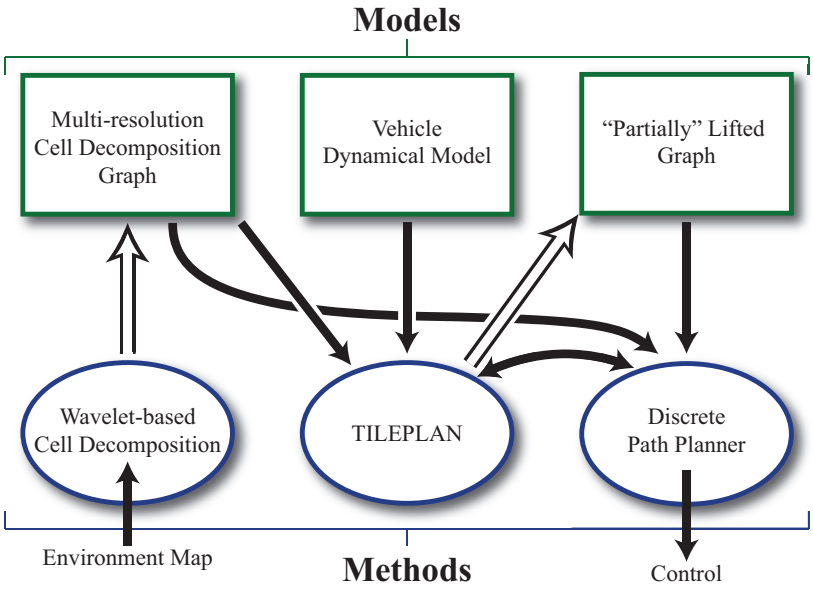


Figure 4.1: Schematic illustration of the proposed motion planning scheme.

4.2.1 Multi-resolution cell decompositions

We define an *image* as a pair (R, F), where $R \subset \mathbb{R}^2$ is a compact, square region, and $F : R \to \mathbb{R}$, $F \in \mathbb{L}^2(R)$, is an intensity map. We will assume that $R = \left[0, 2^D\right] \times \left[0, 2^D\right]$, with $D \in \mathbb{Z}$, and that the image intensity map F is known at a finite resolution $m_{\mathrm{f}} > -D$, i.e., the function F is piecewise constant over each of the square regions $S_{m_{\mathrm{f}},k,\ell}$, for $k, \ell = 0, 1, \ldots, 2^{D+m_{\mathrm{f}}} - 1$. We will assume, without loss of generality, that $m_{\mathrm{f}} = 0$. In the context of path planning, the intensity map F may represent, for instance, terrain elevation [136], a risk measure [174], or a probabilistic occupancy grid [68, 187].

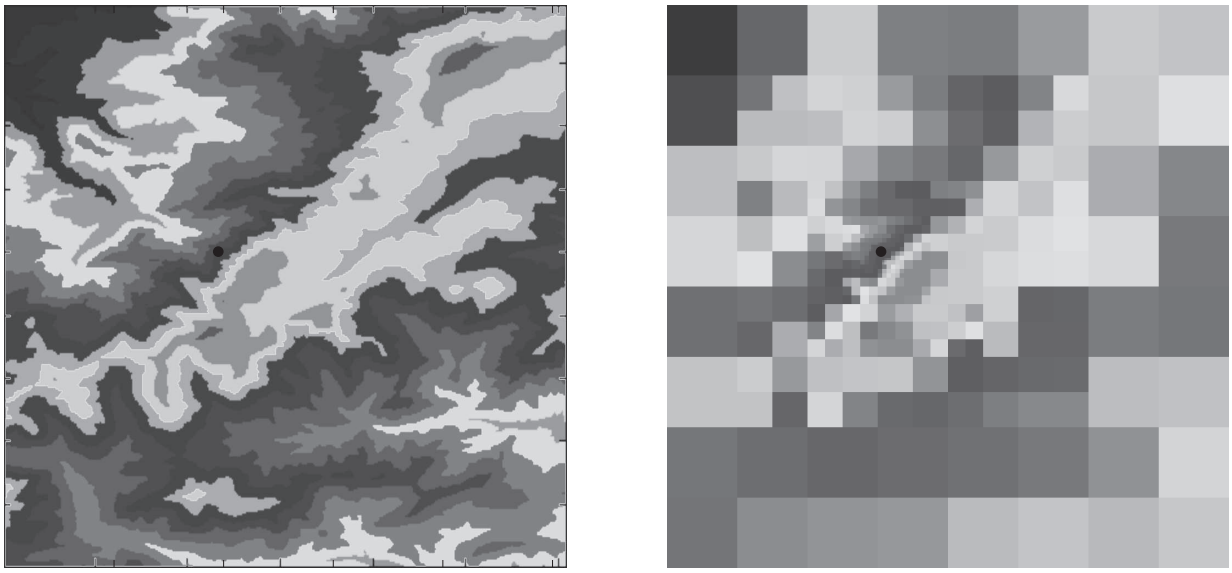
We assume that the least cell size of interest is $2^{-m_{\mathrm{f}}} = 1$, and we define a cell decomposition Ω consisting of uniformly spaced square cells, each of size 1, i.e.,

$$\Omega := \{S_{m_{\mathrm{f}},k,\ell} : k, \ell \in \{0, 1, \ldots, 2^D - 1\}\}.$$

The intention of the geometric path planner is to find a path in the graph associated with Ω. However, the number of cells in Ω is 2^{2D}, which makes the graph search impractical when D is large. To enable fast online computation, multi-resolution cell decompositions may be used to approximate large environment maps. Such decompositions correspond to graphs with significantly fewer vertices, thus requiring lesser computational resources for path planning at each iteration. Furthermore, such decompositions also capture naturally the vehicle's sensing limitations by relaxing progressively, with increasing distance from the vehicle's location, the accuracy at which the intensities of cells in Ω are known.

Let $a_{m_0,k,\ell}$ and $d^p_{m,k,\ell}$ be the DWT coefficients of F, where $m_0 \in \mathbb{Z}$ is pre-specified, let $\mathcal{A} \subset \{(m, k, \ell) \in \mathbb{Z}^3 : m_0 \leq m < 0,\ 0 \leq k, \ell \leq 2^{D+m}\}$ be a set of triplets of integers, and let $\hat{d}^p_{m,k,\ell}$ be defined by

$$\hat{d}^p_{m,k,\ell} := \begin{cases} d^p_{m,k,\ell} & p = 1, 2, 3, \text{ and } (m, k, \ell) \in \mathcal{A}, \\ 0 & \text{otherwise.} \end{cases}$$



(a) Original image (b) Approximation

Figure 4.2: Example of an image and its multi-resolution approximation.

Then the image $(R, \hat{F})$, where $\hat{F}$ is obtained by the reconstruction of $a_{m_0,k,\ell}$ and $\hat{d}^p_{m,k,\ell}$, is called the *approximation* of (R, F) associated with the set $\mathcal{A}$. Informally, an approximation is obtained by neglecting certain detail coefficients in the DWT of F: the set $\mathcal{A}$ contains the indices of detail coefficients that are considered "significant". A specific approximation that is of interest in this chapter is one that retains detail coefficients only in the immediate vicinity of the vehicle's current location $(x_0, y_0) \in R$ and gradually discards them in regions farther away. To define precisely this approximation, let $\varrho : \mathbb{Z} \to \mathbb{N}$ be a "window" function that specifies, for each level of resolution, the distance from the vehicle's location up to which the detail coefficients at that level are significant. The set $\mathcal{A}_{\text{win}}(x_0, y_0)$ of indices of significant detail coefficients is then defined by

$$\begin{aligned}\mathcal{A}_{\text{win}}(x_0, y_0) := \Big\{ & (m, k, \ell) : m_0 \le m < 0, \\ & \lfloor 2^m x_0 \rfloor - \varrho(m) \le k \le \lfloor 2^m x_0 \rfloor + \varrho(m), \\ & \lfloor 2^m y_0 \rfloor - \varrho(m) \le \ell \le \lfloor 2^m y_0 \rfloor + \varrho(m) \Big\}, \end{aligned} \tag{4.1}$$

where $m_0 \in \mathbb{Z}$. An example of an image and its approximation using (4.1) is shown in Fig. 4.2. In this example, $m_0 = -10$, $(x_0, y_0) = (390, 449)$, and $\varrho(m) = 4$ for each $m_0 \le m < 0$.

4.2.2 Computing Cell Locations and Intensities

The cell decomposition Ω^{mr} associated with $(R, \hat{F})$ is a partition of R into square cells of different sizes, such that $\hat{F}$ is constant over each of the cells. In this section, we describe a procedure to determine the locations, the sizes, and the values of $\hat{F}$ over each of the cells in Ω^{mr}.

In this work we use the Haar wavelet family, and the Haar scaling function satisfies the following dilation equation [145]:

$$\phi(t) \quad = \quad \phi(2t) + \phi(2t - 1). \tag{4.2}$$

The dilation equation (4.2) implies, for the 2-D case, that the square support of the scaling functions $\Phi_{m,k,\ell}$ is exactly the union of the supports of $\Phi_{m+1,k,\ell}$, $\Phi_{m+1,k-1,\ell}$, $\Phi_{m+1,k,\ell-1}$, and $\Phi_{m+1,k-1,\ell-1}$. Consequently, a map F is constant over the support of $\Phi_{m,k,\ell}$ if and only if the detail coefficients of F at level m and at higher-resolution levels $m+1, m+2, \ldots$ are all zero. Furthermore, for the Haar scaling function and wavelet, one may associate with each detail coefficient a regions in $\mathbb{R}^2$, such that this coefficient affects the values of the map only in this region. Specifically, we make the following association:

$$d^p_{m,k,\ell} \leftrightarrow S_{m,k,\ell} = 2^{-m} \left([k, (k+1)] \times [\ell, (\ell+1)] \right), \tag{4.3}$$

for each $m_0 \leq m < m_{\mathrm{f}} = 0$, where $m_0 \in \mathbb{Z}$ is pre-specified.

Based on the preceding observations, we formulate the following Rules to determine the locations and the sizes of cells in the cell decomposition Ω^{mr} associated with a set $\mathcal{A}$ of indices of the significant detail coefficients.

1) $\{S_{m_0,\hat{k},\hat{\ell}} : 0 \leq \hat{k}, \hat{\ell} < 2^{D+m_0}\} \in \Omega^{\mathrm{mr}}$. If $\mathcal{A}$ is empty, then these cells form a uniform decomposition.

2) $\{S_{m+1,\hat{k},\hat{\ell}} : \hat{k} \in \{2k, 2k+1\},\ \hat{\ell} \in \{2\ell, 2\ell+1\}\} \in \Omega^{\mathrm{mr}}$ whenever $(m,k,\ell) \in \mathcal{A}$. This Rule arises from the fact that the support of the Haar scaling function at each resolution level is equal to the union of the supports of the scaling functions at the next higher resolution level.

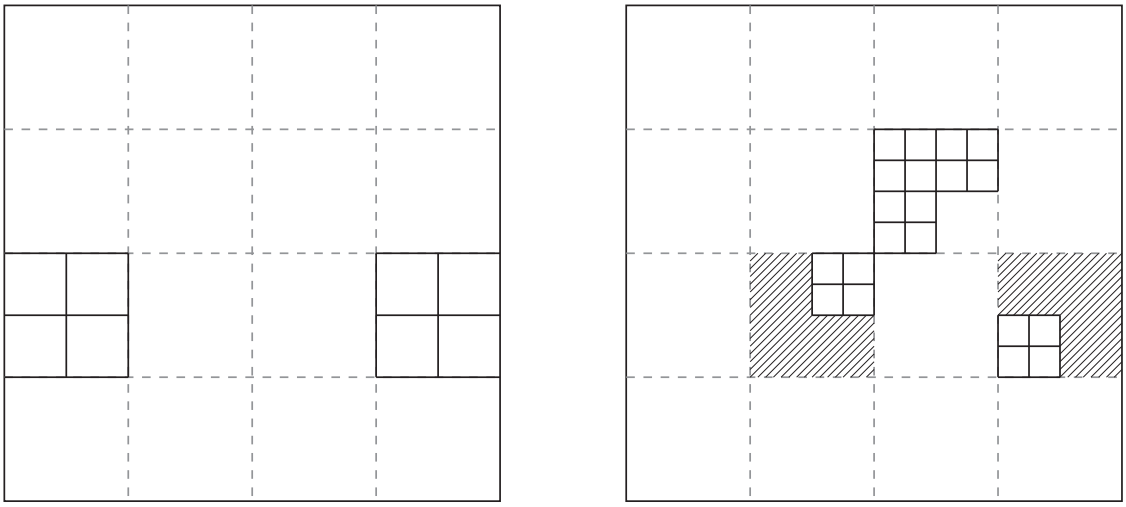
3) $\{S_{\hat{m}+1,\hat{k},\hat{\ell}} : \hat{k} \in \{\lfloor 2^{\hat{m}-m}k \rfloor - 1, \lfloor 2^{\hat{m}-m}k \rfloor\},\ \hat{\ell} \in \{\lfloor 2^{\hat{m}-m}\ell \rfloor - 1, \lfloor 2^{\hat{m}-m}\ell \rfloor\}, m_0 \leq \hat{m} < m\} \in \Omega^{\mathrm{mr}}$, whenever $(m,k,\ell) \in \mathcal{A}$. This Rule decomposes into squares non-convex regions that arise when the indices of a detail coefficient at the level m is in $\mathcal{A}$, but the indices of coefficients associated with the same region (given by (4.3)) at all levels lower than m are not in $\mathcal{A}$.

4) $\{S_{\hat{m},\hat{k},\hat{\ell}} : \hat{k} = \lfloor 2^{\hat{m}-m}k \rfloor,\ \hat{\ell} = \lfloor 2^{\hat{m}-m}\ell \rfloor,\ m_0 \leq \hat{m} \leq m\} \notin \Omega^{\mathrm{mr}}$, whenever $(m,k,\ell) \in \mathcal{A}$. This Rule indicates that a cell, once decomposed, cannot belong to Ω^{mr}.

Exclusions from Ω^{mr} prescribed by Rule 4) take precedence over inclusions prescribed by Rule 3). Note that the preceding Rules are valid only for the Haar system.

Figures 4.3 and 4.4 illustrate the application of the preceding Rules for the approximation associated with $\mathcal{A} = \{(-2,0,2), (-2,3,2), (-1,3,4), (-1,4,2), (-1,4,3),\ (-1,5,2), (-1,6,5)\}$. In Fig. 4.3(a), the cells with dotted borders are due to Rule 1), and the cells with solid borders are due to the Rule 2) for the indices with $m = -2$. The shaded cells in Fig. 4.3(b) illustrate the non-convex regions that may arise due to non-zero coefficients at higher resolution levels, which need to be decomposed using Rule 3). The shaded cells in Fig. 4.4(a) are those which arise twice: due to Rule 2) for level $m = -2$ coefficients and due to Rule 3) for level $m = -1$ coefficients. Figure 4.4(b) shows the final cell decomposition. After determining the elements of the multi-resolution cell decomposition, i.e., the locations and sizes of all the cells, the adjacency relations between cells can be determined by geometric arguments (cf. [102]). To calculate the cell intensities, we use recursively the following relation:

$$\begin{bmatrix} \hat{F}(S_{m+1,2k,2\ell}) \\ \hat{F}(S_{m+1,2k+1,2\ell}) \\ \hat{F}(S_{m+1,2k,2\ell+1}) \\ \hat{F}(S_{m+1,2k+1,2\ell+1}) \end{bmatrix} = 2^{m_0} E \begin{bmatrix} 2^{-m}\hat{F}(S_{m,k,\ell}) \\ d^1_{m,k,\ell} \\ d^2_{m,k,\ell} \\ d^3_{m,k,\ell} \end{bmatrix}, \tag{4.4}$$



(a) Rule 2) for $m = -2$.

(b) Rule 2) for $m = -1$.

Figure 4.3: Computations of cell locations and dimensions from $\mathcal{A}$.

for $0 \leq k, \ell < 2^{D+m}$, with $\hat{F}(S_{m_0,k,\ell}) = 2^{m_0} a_{m_0,k,\ell}$ for $0 \leq k, \ell < 2^{D+m_0}$, where E is a constant matrix. The intensities of the cells due to Rule 2) for a triplet $(m, k, \ell) \in \mathcal{A}$ are given by (4.4). The intensities of cells due to Rule 3) for a triplet $(m, k, \ell) \in \mathcal{A}$, are each equal to $F(S_{m_1,k_1,\ell_1})$, where $(m_1, k_1, \ell_1) \in \mathcal{A}$ is the triplet with the greatest $m_1 < m$ satisfying $S_{m,k,\ell} \subset S_{m_1,k_1,\ell_1}$. If no such triplet exists, then the intensities of these cells are each equal to $F(S_{m_0,k_1,\ell_1})$, where k_1, ℓ_1 are the unique indices satisfying $S_{m,k,\ell} \subset S_{m_0,k_1,\ell_1}$.

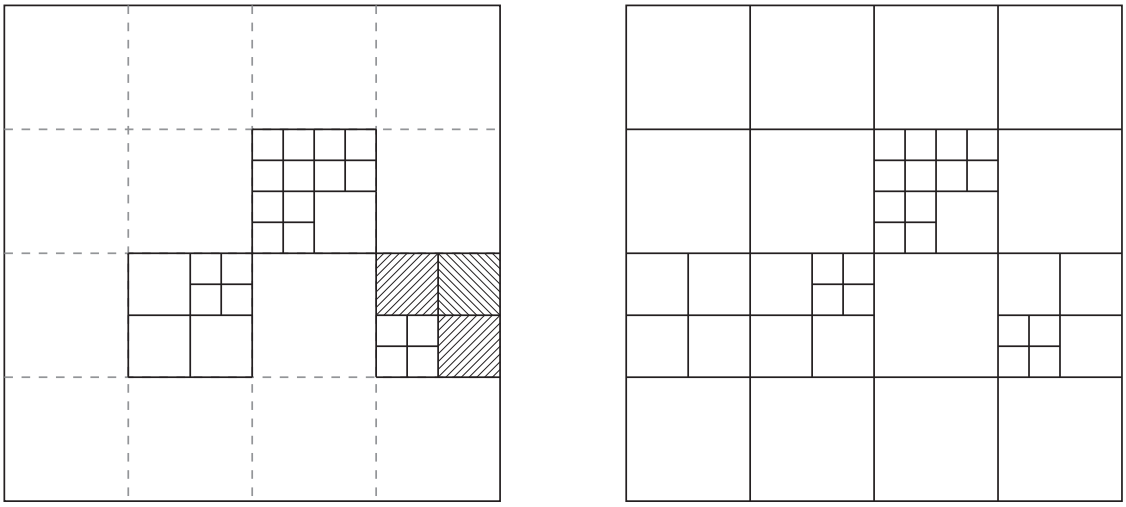
We re-emphasize that *all* information needed to completely define the cell decomposition Ω^{mr} is encoded in the set $\mathcal{A}$, and it is straightforward to extract this information. Furthermore, the expression (4.1) lends itself to a fast update of the set $\mathcal{A}$ in accordance with the changes in the vehicle's position in the environment, as we will demonstrate in Section 4.3.2.

4.3 Multi-resolution Path Planning

We denote by $\bar{\mathcal{G}} = (\bar{V}, \bar{E})$ the graph associated with the cell decomposition Ω, such that each cell in Ω corresponds to a unique vertex in $\bar{V}$. We will denote by $\mathsf{cell}(j; \Omega^{\mathrm{mr}})$ the cell in Ω^{mr} associated with a vertex $j \in \bar{V}$, and by $\mathsf{vert}(c; \bar{\mathcal{G}})$ the vertex of $\bar{\mathcal{G}}$ associated with a cell $c \in \Omega^{\mathrm{mr}}$. Two vertices are adjacent if the corresponding cells are geometrically adjacent, and $\bar{E}$ is the collection of all ordered pairs $(\bar{i}, \bar{j})$ of vertices in $\bar{V}$, such that $\bar{i}$ and $\bar{j}$ are adjacent. In what follows, we will distinguish by an overline symbols denoting vertices, paths, or functions associated with Ω or $\bar{\mathcal{G}}$. We introduce an edge cost function $\bar{g} : \bar{E} \to \mathbb{R}_+$ that assigns to each edge of $\bar{\mathcal{G}}$ a non-negative cost of transitioning this edge.

For given initial and terminal vertices $\bar{i}_{\mathrm{S}}$, $\bar{i}_{\mathrm{G}} \in \bar{V}$, an *admissible path* $\bar{\pi}(\bar{i}_{\mathrm{S}}, \bar{i}_{\mathrm{G}})$ in $\bar{\mathcal{G}}$ is a finite sequence $(\bar{i}_0, \ldots, \bar{i}_{\bar{P}})$ of vertices (with no repetition) such that $\{\bar{i}_{k-1}, \bar{i}_k\} \in \bar{E}$ for each $k = 1, \ldots, P$, with $\bar{i}_0 = \bar{i}_{\mathrm{S}}$ and $\bar{i}_P = \bar{i}_{\mathrm{G}}$. For brevity, and when there is no ambiguity, we will henceforth suppress the arguments in $\bar{\pi}$. The cost $\bar{\mathcal{J}}(\bar{\pi})$ of an admissible path $\bar{\pi}$ in $\bar{\mathcal{G}}$ is the sum of costs of edges in $\bar{\pi}$, and the *path planning problem* is to find an admissible path $\bar{\pi}^*(\bar{i}_{\mathrm{S}}, \bar{i}_{\mathrm{G}})$ with minimum cost.

Next, we associate with the multi-resolution cell decomposition Ω^{mr} a graph $\mathcal{G} = (V, E)$ such that



(a) Rule 3) for $m = -1$.

(b) Overall decomposition.

Figure 4.4: Computations of cell locations and dimensions from $\mathcal{A}$.

each cell in Ω^{mr} corresponds to a unique vertex in V. Note that each vertex $j \in V$ corresponds to a set $W(j, V) \subset \bar{V}$, and the collection $\{W(j, V)\}_{j \in V}$ is a partition of $\bar{V}$. Specifically:

$$W(j, V) := \left\{ \bar{j} \in \bar{V} : \mathsf{cell}(\bar{j}; \Omega) \subseteq \mathsf{cell}(j; \Omega^{\mathrm{mr}}) \right\}.$$

The multi-resolution cell decomposition graph $\mathcal{G}$ approximates the graph $\bar{\mathcal{G}}$ by representing each set of vertices $W(j, V) \subset \bar{V}$ with a single vertex $j \in V$. For the Haar wavelet, it can be shown that for each $j \in V$,

$$\hat{F}(\mathsf{cell}(j; \Omega^{\mathrm{mr}})) = \frac{1}{|W(j, V)|} \sum_{\bar{j} \in W(j,V)} F(\mathsf{cell}(\bar{j}; \Omega)). \tag{4.5}$$

Finally, two vertices $i, j \in V$ are said to be adjacent in $\mathcal{G}$, i.e., $(i, j) \in E$, if and only if there exist $\bar{i} \in W(i, V)$ and $\bar{j} \in W(j, V)$ such that $\{\bar{i}, \bar{j}\} \in \bar{E}$. We will denote by $\mathsf{cell}(j; \Omega^{\mathrm{mr}})$ the cell in Ω^{mr} associated with the vertex $j \in V$, and by $\mathsf{vert}(S; \mathcal{G})$ the vertex in $\mathcal{G}$ associated with the cell $S \in \Omega^{\mathrm{mr}}$.

4.3.1 Path Planning Algorithm

We present an algorithm that finds iteratively an admissible path $\bar{\pi}$ in $\bar{\mathcal{G}}$ by first constructing multi-resolution cell decompositions, then by finding paths in these multi-resolution cell decompositions, and and then by moving along these paths. The proposed algorithm is a modification of the multi-resolution path planning algorithm presented in [174] and these modifications ensure that the proposed algorithm is complete.

We assume that $F(\mathsf{cell}(\bar{j}; \Omega)) \in [0, 1]$ for each $\bar{j} \in \bar{V}$, such that more favorable cells in the environment have a lower intensity, and such that $\mathsf{cell}(\bar{j}; \Omega)$ represents an obstacle if $F(\mathsf{cell}(\bar{j}; \Omega)) > 1 - \varepsilon$, for some pre-specified $\varepsilon \in (0, 1)$. We define the transition cost of an edge $(\bar{i}, \bar{j}) \in \bar{E}$ by

$$\bar{g}((\bar{i}, \bar{j})) := \begin{cases} \lambda_1 F(\mathsf{cell}(\bar{j}; \Omega)) + \lambda_2, & F(\mathsf{cell}(\bar{j}; \Omega)) \leq 1 - \varepsilon, \\ M, & \text{otherwise}, \end{cases} \tag{4.6}$$

where $\lambda_1, \lambda_2 \in (0, 1]$ and $M \gg 2|\bar{V}|$ are constants.

We denote by $\hat{F}_n$ the approximation of the environment constructed at iteration n of the proposed algorithm, by $\mathcal{A}(n)$ the associated set of detail coefficients, by $\Omega^{\mathrm{mr}}(n)$ the associated multi-resolution cell decomposition, and by $\mathcal{G}(n) = (V(n), E(n))$ the associated topological graph. We define the goal vertex $i_{\mathrm{G},n} \in V(n)$ as the unique vertex that satisfies $\bar{i}_{\mathrm{G}} \in W(i_{\mathrm{G},n}, V(n))$.

For each vertex $\bar{j} \in \bar{V}$, the proposed algorithm maintains an estimate $\mathcal{K}_{\mathrm{G}}(\bar{j})$ of the least cost of any path in $\bar{\mathcal{G}}$ from the vertex $\bar{j}$ to the goal vertex $\bar{i}_{\mathrm{G}}$, and a record $\mathcal{K}_{\mathrm{S}}(\bar{j})$ of the least cost of any path in $\bar{\mathcal{G}}$ from the initial vertex $\bar{i}_{\mathrm{S}}$ to the vertex $\bar{j}$. The algorithm also associates with each vertex $\bar{j} \in \bar{V}$ another vertex $b(\bar{j}) \in \bar{V}$ called the *backpointer* of $\bar{j}$. At each iteration, the algorithm performs a computation (Line 18 or Line 20 in Fig. 4.5) whose result is a unique vertex in $\bar{V}$. We refer to this computation as a *visit* to this vertex, and we denot by $\bar{j}_n$ the vertex visited at iteration n, with $\bar{j}_0 := \bar{i}_{\mathrm{S}}$. Finally, let $j_n := \mathsf{vert}(\mathsf{cell}(\bar{j}_n; \Omega^{\mathrm{mr}}(n)); \mathcal{G}(n))$.

An admissible path $\pi_n(j_n, i_{\mathrm{G},n})$ in $\mathcal{G}(n)$ is a finite sequence $(i_0, \ldots, i_{P(n)})$ of vertices (with no repetition) in $V(n)$ excluding $b(\bar{j}_n)$ and excluding vertices in $V(n)$ corresponding to cells in the path from $\bar{i}_{\mathrm{S}}$ to $\bar{j}_n$. Note that this definition precludes cycles in the concatenation of the path $(j_0, \ldots, j_{n-1})$ with the path π_n. We introduce an edge cost function $g_n : E(n) \to \mathbb{R}_+$, which assigns to each edge of $\mathcal{G}(n)$ a non-negative cost of transitioning this edge, defined by

$$g_n((i,j)) := \begin{cases} (\lambda_1 \hat{F}_j + \lambda_2)|W(j, V(n))|, & \hat{F}_j \leq 1 - \varepsilon, \\ M, & \text{otherwise}, \end{cases} \tag{4.7}$$

where $\hat{F}_j := \hat{F}(\mathsf{cell}(j; \Omega^{\mathrm{mr}}(n)))$. The cost $\mathcal{J}(\pi_n)$ of the path π_n is the sum of the costs of edges in the path. Note that, by (4.5) and (4.7), the cost of an obstacle-free path in $\mathcal{G}(n)$ is less than or equal to $2|\bar{V}|$, and hence an admissible path π_n in $\mathcal{G}(n)$ is obstacle-free if and only if $\mathcal{J}(\pi_n) < M$.

The proposed algorithm associates with each vertex $\bar{j} \in \bar{V}$ a binary value $\textsc{Visited}(\bar{j})$ that records whether the vertex $\bar{j}$ has previously been visited by the algorithm: at any iteration of the algorithm's execution, for each $\bar{j} \in \bar{V}$, $\textsc{Visited}(\bar{j}) = 0$ indicates that the algorithm has never visited $\bar{j}$ in any previous iteration, whereas $\textsc{Visited}(\bar{j}) = 1$ indicates that the algorithm has visited $\bar{j}$ during a previous iteration. The algorithm also maintains a cumulative cost $\bar{\mathcal{J}}(\bar{\pi})$ of the path $\bar{\pi}(\bar{i}_{\mathrm{S}}, \bar{j}_n)$ in $\bar{\mathcal{G}}$. The proposed multi-resolution path planning algorithm is described by the pseudo-code in Fig. 4.5. Here $x(\bar{j})$ and $y(\bar{j})$ denote, respectively, the x and y coordinates of the center of $\mathsf{cell}(\bar{j}; \bar{\mathcal{G}})$, and MR-Graph denotes the procedure described in Section 4.2.2 to obtain the cell decomposition graph associated with a set of indices of significant detail coefficients.

Remark 4.1. The constrained optimization problem in Line 8 can be solved by an algorithm that finds the K shortest paths in a graph. Such algorithms have been reported, for instance, in [89]. We assume that the K shortest paths will have strictly increasing costs. This assumption is not required for the algorithm's successful execution, but it enables a concise statement of the algorithm. □

Remark 4.2. Due to Line 9, the cost of "back-tracking" is not added to the cumulative cost $\bar{\mathcal{J}}(\bar{\pi})$. Also, it follows from (4.7) and Line 21 that $\mathcal{K}_{\mathrm{G}}(\bar{j}) = 0$ if and only if $\bar{j} = \bar{i}_{\mathrm{G}}$. □

We may now state the main result of this section as follows.

Proposition 4.1. The proposed algorithm is complete: if there exists an obstacle-free path in $\bar{\mathcal{G}}$ from $\bar{i}_{\mathrm{S}}$ to $\bar{i}_{\mathrm{G}}$, then the algorithm finds an obstacle-free path in a finite number of iterations. Otherwise, the algorithm reports failure after a finite number of iterations.

Proof. See [56]. □

Multi-resolution Path Planning Scheme

procedure MR-APPROX($\bar{j}$)

1: $\mathcal{A} \leftarrow \mathcal{A}_{\text{win}}(x(\bar{j}), y(\bar{j}))$

procedure MAIN

1: $\bar{\pi} \leftarrow \bar{i}_{\text{S}}$, $\bar{j}_0 \leftarrow \bar{i}_{\text{S}}$, $n \leftarrow 0$, $AtGoal \leftarrow 0$, $\bar{\mathcal{J}}(\bar{\pi}) \leftarrow 0$
2: For each $\bar{j} \in \bar{V}$, VISITED($\bar{j}$) $\leftarrow 0$
3: **while** $\neg AtGoal$ and $\bar{\mathcal{J}}(\bar{\pi}) < M$ and $\mathcal{K}_{\text{G}}(\bar{j}_n) < M$ **do**
4: $b(\bar{j}_n) \leftarrow \bar{j}_{n-1}$
5: $\mathcal{A}(n) \leftarrow$ MR-APPROX($\bar{j}_n$)
6: $\mathcal{G}(n) \leftarrow$ MR-GRAPH($\mathcal{A}(n)$)
7: **if** VISITED($\bar{j}_n$) $= 1$ **then**
8: $\pi_n^* \leftarrow \arg\min \{\mathcal{J}(\pi) : \pi$ obstacle-free in $\mathcal{G}(n)\}$, subject to $\mathcal{J}(\pi_n^*) > \mathcal{K}_{\text{G}}(\bar{j}_n)$
9: $\bar{\mathcal{J}}(\bar{\pi}) \leftarrow \mathcal{K}_{\text{S}}(\bar{j}_n)$
10: **else**
11: $\pi_n^* \leftarrow \arg\min \{\mathcal{J}(\pi) : \pi$ obstacle-free in $\mathcal{G}(n)\}$
12: $\mathcal{K}_{\text{S}}(\bar{j}_n) \leftarrow \bar{\mathcal{J}}(\bar{\pi})$
13: VISITED($\bar{j}_n$) $\leftarrow 1$
14: **if** π_n^* does not exist **then**
15: **if** $\bar{j}_n = \bar{i}_{\text{S}}$ **then**
16: Report failure
17: **else**
18: $\bar{j}_{n+1} \leftarrow b(\bar{j}_n)$
19: **else**
20: $\bar{j}_{n+1} \leftarrow \mathsf{vert}(\mathsf{cell}(i_1; \mathcal{G}(n)); \bar{\mathcal{G}})$
21: $\mathcal{K}_{\text{G}}(\bar{j}_n) \leftarrow \mathcal{J}(\pi_n^*)$
22: $AtGoal \leftarrow (\mathcal{K}_{\text{G}}(\bar{j}_n) = 0)$,
23: $\bar{\pi} \leftarrow (\bar{\pi}, \bar{j}_n)$
24: $\bar{\mathcal{J}}(\bar{\pi}) \leftarrow \bar{\mathcal{J}}(\bar{\pi}) + \bar{g}(\bar{j}_n, \bar{j}_{n+1})$
25: $n \leftarrow n + 1$
26: **if** $\bar{\mathcal{J}}(\bar{\pi}) \geq M$ or $\mathcal{K}_{\text{G}}(\bar{j}_n) \geq M$ **then**
27: Report failure

Figure 4.5: Pseudo-code for the proposed path planning algorithm.

4.3.2 Efficient Updates of $\mathcal{A}(n)$ and $\mathcal{G}(n)$

The set $\mathcal{A}(n)$ of the significant detail coefficient indices, and the associated multi-resolution cell decomposition graph both depend on the vehicle's current position. Consequently, both $\mathcal{A}(n)$ and $\mathcal{G}(n)$ are updated (Lines 5–6 in Fig. 4.5) at each iteration of the algorithm. To enable faster computations, we describe, in this section, a method to obtain $\mathcal{A}(n)$ incrementally from $\mathcal{A}(n-1)$. Specifically, we first compute the elements of the sets $\mathcal{B}_1 := \mathcal{A}(n) \cap \mathcal{A}^{\mathrm{c}}(n-1)$ and $\mathcal{B}_{-1} := \mathcal{A}(n-1) \cap \mathcal{A}^{\mathrm{c}}(n)$, and we then evaluate $\mathcal{A}(n) = \mathcal{A}(n-1) \cup \mathcal{B}_1 \backslash \mathcal{B}_{-1}$. To this end, we observe that for each $\bar{j} \in \bar{V}$, $x(\bar{j}) = \lfloor x(\bar{j}) \rfloor + 1/2$. It follows that for every $m \leq 0$,

$$\lfloor 2^m x(\bar{j}_n) \rfloor \quad = \quad \lfloor 2^m (\lfloor x(\bar{j}) \rfloor + 1/2) \rfloor. \tag{4.8}$$

Next, we note that $\lfloor x(\bar{j}_n) \rfloor = \lfloor x(\bar{j}_{n-1}) \rfloor + \Delta_x$, where, at iteration n, $\Delta_x = 1$ if the vehicle moves one cell to the right, $\Delta_x = -1$ if the vehicle moves one cell to the left, and $\Delta_x = 0$ otherwise. From (4.8), it may be shown [57] that

$$\lfloor 2^m x(\bar{j}_{n+1}) \rfloor \quad = \quad \lfloor \lfloor 2^m x(\bar{j}_n) \rfloor + 2^m \Delta_x + r_x^m \rfloor, \tag{4.9}$$

where $r_x^m := 2^m \left(\lfloor 2^m x(\bar{j}_n) \rfloor + 1/2 \right) - \lfloor 2^m x(\bar{j}_n) \rfloor$. Similarly,

$$\lfloor 2^m y(\bar{j}_{n+1}) \rfloor \quad = \quad \lfloor \lfloor 2^m y(\bar{j}_n) \rfloor + 2^m \Delta_y + r_y^m \rfloor, \tag{4.10}$$

where $r_y^m := 2^m \left(\lfloor 2^m y(\bar{j}_n) \rfloor + 1/2 \right) - \lfloor 2^m y(\bar{j}_n) \rfloor$. The elements of the sets $\mathcal{B}_1$ and $\mathcal{B}_{-1}$ are then determined from (4.9)-(4.10) as follows. We first define the scalars δ_x and δ_y by

$$\delta_\alpha := \begin{cases} -1, & 0 > 2^m \Delta_\alpha + r_\alpha^m, \\ 0, & 0 \leq 2^m \Delta_\alpha + r_\alpha^m < 1, \\ 1, & 1 \leq 2^m \Delta_\alpha + r_\alpha^m, \end{cases} \quad \text{for } \alpha \in \{x, y\}, \tag{4.11}$$

and, for $p \in \{-1, 1\}$, we define the sets $\mathcal{B}_p^{m,x}$ and $\mathcal{B}_p^{m,x}$ by

$$\begin{aligned} \mathcal{B}_p^{m,x} \quad &:= \quad \{(m,k,\ell) : k = \lfloor 2^m x(\bar{j}_n) \rfloor + p\delta_x, \\ &\qquad \lfloor 2^m y(\bar{j}_n) \rfloor - \varrho(m) \leq \ell \leq \lfloor 2^m y(\bar{j}_n) \rfloor + \varrho(m) \}, \\ \mathcal{B}_p^{m,y} \quad &:= \quad \{(m,k,\ell) : \ell = \lfloor 2^m y(\bar{j}_n) \rfloor + p\delta_y, \\ &\qquad \lfloor 2^m x(\bar{j}_n) \rfloor - \varrho(m) \leq k \leq \lfloor 2^m x(\bar{j}_n) \rfloor + \varrho(m) \}. \end{aligned}$$

Then the sets $\mathcal{B}_{-1}$ and $\mathcal{B}_1$ are given by the following equation:

$$\mathcal{B}_p \quad = \quad \bigcup_{\alpha \in \{x,y\}} \bigcup_{m_0 \leq m < 0} \mathcal{B}_p^{m\alpha}, \quad p \in \{-1, 1\}. \tag{4.12}$$

The advantage of computing $\mathcal{A}(n)$ using the modified procedure Mod-MR-Approx described in Fig. 4.6 instead of the procedure MR-Approx, is that the sets $\mathcal{B}_{-1}$ and $\mathcal{B}_1$ have significantly fewer elements than $\mathcal{A}(n)$. More precisely, the number of elements in the set $\mathcal{A}(n)$ is $O(\bar{\varrho}^2)$, whereas the numbers of elements in the sets $\mathcal{B}_{-1}$ and $\mathcal{B}_1$ are both $O(\bar{\varrho})$, where $\bar{\varrho} := \max_{m_0 \leq m \leq 0} \{\varrho(j)\}$.

Figures 4.7(a) and 4.7(b) show data that confirm the preceding observations: these figures show the ratio of the execution time required by the combination of the procedures MR-Approx and MR-Graph to the execution time required by the combination of the procedures Mod-MR-Approx and Mod-MR-Graph for computing the graph $\mathcal{G}(n)$. The data shown in Figs. 4.7(a) and 4.7(b) are averages computed over 30 simulations. As it is evident from these results, the multi-resolution path planning algorithm with the modified procedures of construction of $\mathcal{A}(n)$ and $\mathcal{G}(n)$ executes up to 10 times faster than that with the original procedures.

Recomputation of Multi-resolution Cell Decomposition

procedure MOD-MR-APPROX($\mathcal{A}$)

1: Compute $\mathcal{B}_{-1}$ and $\mathcal{B}_1$ with (4.12)
2: $\mathcal{A}(n) \leftarrow \mathcal{A}(n-1) \cup \mathcal{B}_1 \backslash \mathcal{B}_{-1}$

procedure MOD-MR-GRAPH($\Omega^{\mathrm{mr}}(n-1), \mathcal{B}_{-1}, \mathcal{B}_1$)

1: $\Omega^{\mathrm{mr}}_{-1} \leftarrow \varnothing$, $\Omega^{\mathrm{mr}}_{1} \leftarrow \varnothing$
2: **for all** $(m,k,\ell) \in \mathcal{B}_1$ **do**
3: $\quad \Omega^{\mathrm{mr}}_{1} \leftarrow \Omega^{\mathrm{mr}}_{1} \cup \{S_{m+1,\hat{k},\hat{\ell}} : \hat{k} \in \mathcal{K},\ \hat{\ell} \in \mathcal{L}\}$
4: $\quad \Omega^{\mathrm{mr}}_{-1} \leftarrow \Omega^{\mathrm{mr}}_{-1} \cup \{S_{\hat{m},\hat{k},\hat{\ell}} : \hat{k} = \lfloor 2^{\hat{m}-m} k \rfloor,\ \hat{\ell} = \lfloor 2^{\hat{m}-m} \ell \rfloor,\ m_0 \leq \hat{m} \leq m\}$
5: **for all** $(m,k,\ell) \in \mathcal{B}_{-1}$ **do**
6: $\quad \Omega^{\mathrm{mr}}_{-1} \leftarrow \Omega^{\mathrm{mr}}_{-1} \cup \{S_{m+1,\hat{k},\hat{\ell}} : \hat{k} \in \mathcal{K},\ \hat{\ell} \in \mathcal{L}\}$
7: $\quad \Omega^{\mathrm{mr}}_{1} \leftarrow \Omega^{\mathrm{mr}}_{1} \cup \{S_{m,k,\ell}\}$
8: $\Omega^{\mathrm{mr}}(n) \leftarrow \Omega^{\mathrm{mr}}(n-1) \cup \Omega^{\mathrm{mr}}_{1} \backslash \Omega^{\mathrm{mr}}_{-1}$

Figure 4.6: Pseudo-code for the procedure MOD-MR-GRAPH.

4.4 Multi-resolution H-Cost Motion Planning

It has been noted in several previous works [185, 146, 113], including ours [59], that single-edge transition costs in cell decomposition graphs cannot capture adequately the vehicle's kinematic and dynamic constraints. In light of this observation, we introduced in [59] a motion planning approach based on assigning costs to *multiple* edge transitions (called *histories*) in cell decomposition graphs.

Consider the multi-resolution cell decomposition graph[1] $\mathcal{G} = (V, E)$ at any iteration of the path planning algorithm previously discussed. To formalize the concept of cell histories, we define, for every integer $H \geq 0$, the set

$$\begin{aligned} V_H \;:=\; & \{(j_0, \ldots, j_H) : \{j_{k-1}, j_k\} \in E,\ k = 1, \ldots, H, \\ & j_k \neq j_\ell,\ \text{ for } k, \ell \in \{0, \ldots, H\},\ \text{with } k \neq \ell\}\,. \end{aligned}$$

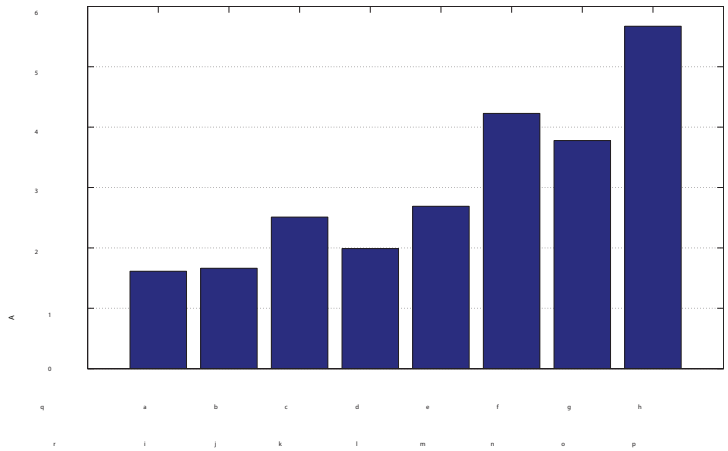
An element of V_{H+1} is called an *H-history*. Let $I \in V_H$ and denote by $[I]_k$ the k^{th} element of this $(H+1)$-tuple, and by $[I]_k^\ell$ the tuple $([I]_k, [I]_{k+1}, \ldots, [I]_\ell)$, for $k < \ell \leq H+1$. We associate with each H a non-negative cost function $g_H : V_{H+1} \to \mathbb{R}_+$, and state a shortest path problem with transition costs defined on histories as follows.

Problem 4.1 (H-Cost Shortest Path Problem)**.** Let $H \geq 0$, and let $i_{\mathrm{S}}, i_{\mathrm{G}} \in V$ be initial and goal vertices such that any admissible path in $\mathcal{G}$ contains at least $H+1$ vertices. The *H-cost* of an admissible path $\pi = (j_0, \ldots, j_P)$ in $\mathcal{G}$ is defined by

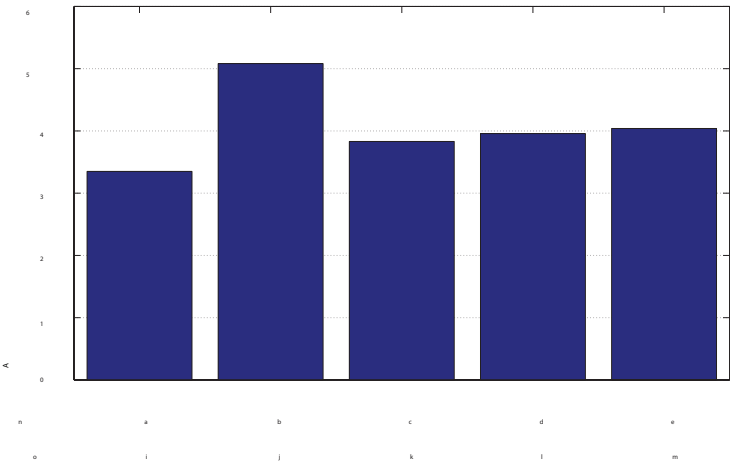
$$\mathcal{J}_H(\pi) := \sum_{k=H+1}^{P} g_H\left((j_{k-H-1}, j_{k-H}, \ldots, j_k)\right). \tag{4.13}$$

Find an admissible path π^* in the graph $\mathcal{G}$ such that $\mathcal{J}_H(\pi^*) \leq \mathcal{J}_H(\pi)$ for every admissible path π in $\mathcal{G}$.

[1] For the sake of clarity, we drop from the notation of the cell decomposition graph the explicit reference to the n^{th} iteration.



(a) Comparisons of execution times for the computations of $\mathcal{A}$ and $\mathcal{G}$.



(b) Comparisons of execution times for overall path planning.

Figure 4.7: Sample data illustrating benefits of incremental updates to $\mathcal{A}$ and $\mathcal{G}$.

Problem 4.1 may be transformed into an equivalent standard shortest path problem on a *lifted graph* $\mathcal{G}_H$. The vertices of $\mathcal{G}_H$ are the elements of V_H, and the edge set E_H of the lifted graph $\mathcal{G}_H$ is the set of all ordered pairs (I, J), such that $I, J \in V_H$, with $[I]_k = [J]_{k-1}$, for every $k = 2, \ldots, H+1$, and $[I]_1 \neq [J]_{H+1}$. For given initial and terminal vertices i_{S}, $i_{\mathrm{G}} \in V$, an admissible path Π in $\mathcal{G}_H$ is a finite sequence $(J_0, \ldots, J_Q)$ of vertices (with no repetition) such that $(J_{k-1}, J_k) \in E_H$, for each $k = 1, \ldots, Q$, with $[J_0]_1 = i_{\mathrm{S}}$, and $[J_Q]_{H+1} = i_{\mathrm{G}}$. Note that every admissible path $\Pi = (J_0, \ldots, J_Q)$ in $\mathcal{G}_H$ corresponds uniquely to an admissible path $\pi = (j_0, \ldots, j_P)$ in $\mathcal{G}$, with $P = Q + H$ and $[J_k]_\ell = j_{kH+\ell-1}$, for each $k = 0, 1, \ldots, Q-1$, and $J_Q = (j_{P-H}, \ldots, j_P)$.

We introduce a non-negative cost function $\tilde{g}_H : E_H \to \mathbb{R}_+$ defined by

$$\tilde{g}_H((I, J)) := g_H\left(([I]_1^{H+1}, [J]_{H+1})\right),$$

for every pair $(I, J) \in E_H$. It follows that Problem 4.1 is equivalent to the standard shortest path problem on $\mathcal{G}_H$, where the cost of an edge $(I, J) \in E_H$ given by $\tilde{g}_H((I, J))$. However, solving the H-cost shortest path problem by first transforming it to the standard problem is computationally intensive, because $|\mathcal{G}_H|$ is large and grows exponentially with H.

In [57], we discuss an efficient and flexible algorithm for solving the H-cost shortest path problem, as well as a motion planning framework that incorporates vehicle kinematic and dynamic constraints by obtaining H-costs from a local trajectory generation algorithm called the *tile motion planner* (TilePlan). A precise statement of the tile motion planning problem and its solution based on model predictive control are available in [57]. Briefly, we specify TilePlan as an algorithm that takes as the input a sequence of cells and an initial state, and returns as the output a control input (if it exists) that enables the vehicle's traversal through the given sequence of cells from the given initial state.

The overall motion planner searches for H-cost shortest paths in the multi-resolution cell decomposition graphs described in Section 4.3. However, it is unnecessary and computationally expensive to consider history-based transition costs on the entire multi-resolution cell decomposition graph due to the following reasons: (a) large cells in Ω^{mr} correspond to coarse information about the environment in the regions associated with those cells, and hence trajectories passing through large cells will be refined and/or replanned in future iterations, and (b) curvature-constrained paths are guaranteed to exist [28] in rectangular channels wider than a certain threshold width (compared to the upper bound on curvature).

In light of the preceding observations, and in keeping with the multi-resolution idea of using high-accuracy information only locally, the proposed motion planner searches for H-cost shortest paths on a "partially" lifted graph, such that the vehicle dynamical constraints are considered (via history-based transition costs) only locally. To state precisely this notion of a "partially" lifted graph, we define, for each $J = (j_0, \ldots, j_H) \in V_H$, and each $L \in \{1, \ldots, H-1\}$, the *projection* $\mathcal{P}_L(J)$ of J onto V_L, by $\mathcal{P}_L(J) := (j_0, \ldots, j_L) \in V_L$. For each $L \in \{1, \ldots, H\}$, we define the set $U_L \subseteq V_L$ by

$$\begin{aligned} U_L := \big\{ &(j_0, \ldots, j_L) \in V_L : \mathsf{size}(\mathsf{cell}(j_k)) < \bar{d}, \\ &\text{for } k = 0, \ldots, L-1, \text{ and } \mathsf{size}(\mathsf{cell}(j_L)) \leq \bar{d} \big\}, \end{aligned} \tag{4.14}$$

where $\bar{d}$ is pre-specified, and $\mathsf{size}(\mathsf{cell}(j_k))$ denotes the size of the cell that corresponds to the vertex j_k in the multi-resolution cell decomposition graph. By (4.14), the set U_L consists of $(L+1)$-tuples of vertices in the cell decomposition graph such that the sizes of the first L cells in each $(L+1)$-tuple are strictly lower than $\bar{d}$, whereas the size of the $(L+1)^{\text{th}}$ cell is at most $\bar{d}$. This definition alludes to the previously stated notion of including in the "partially" lifted graph only the cells small enough

Multi-resolution Motion Planning

1: $i \leftarrow \mathsf{vert}(\mathsf{cell}(\bar{i}_{\mathrm{S}}; \Omega); \mathcal{G})$, $AtGoal \leftarrow 0$
2: **while** $\neg AtGoal$ **do**
3: Define $V_{\leq \bar{d}} := \left\{ j \in V : \mathsf{size}(\mathsf{cell}(j)) \leq \bar{d} \right\}$,
$V_{> \bar{d}} := \left\{ j \in V : \mathsf{size}(\mathsf{cell}(j)) > \bar{d} \right\}$
4: Find the set of boundary vertices V_{bd} defined by

$$V_{\mathrm{bd}} := \{ j \in V : \exists\ i_1 \in V_{\leq \bar{d}},\ \ i_2 \in V_{> \bar{d}} \text{ such that } (i_1, j) \in E,\ \ (i_2, j) \in E \}$$

5: For each $j \in V_{\mathrm{bd}}$, define $\mathcal{K}(j) :=$ minimum cost of a path in $\mathcal{G}$ from j to the goal vertex i_{G}
6: Define $V_H^{\mathrm{S}} := \{ I \in \tilde{V}_H : [I]_1 = i \}$, $\quad V_H^{\mathrm{G}} := \{ I \in \tilde{V}_H : [I]_{\mathrm{last}} \in V_{\mathrm{bd}} \}$
7: Find the shortest path $\Pi^* = (I_0, \ldots, I_P)$ in $\tilde{\mathcal{G}}_H$ from any vertex $I_{\mathrm{S}} \in V_H^{\mathrm{S}}$ to any vertex in $I_{\mathrm{G}} \in V_H^{\mathrm{G}}$, with terminal penalty $\mathcal{K}([I_{\mathrm{G}}]_{\mathrm{last}})$
8: $i \leftarrow [I_0]_2$
9: **if** $\mathsf{cell}(i; \Omega^{\mathrm{mr}}) = \mathsf{cell}(\bar{i}_{\mathrm{G}}; \Omega)$ **then**
10: $AtGoal \leftarrow 1$

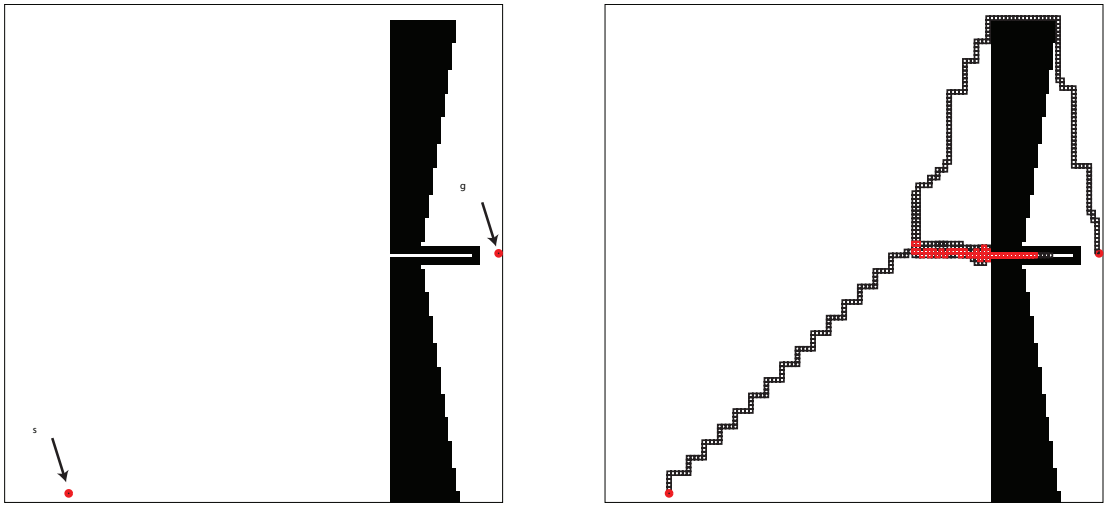
Figure 4.8: Pseudo-code describing the overall motion planner.

for the curvature constraints to be significant. The "partially" lifted graph $\tilde{\mathcal{G}_H} = (\tilde{V}_H, \tilde{E}_H)$ is then defined by

$$\tilde{V}_H := \textstyle\bigcup_{L=1}^{H} U_L \backslash \mathcal{P}_L(U_H),$$
$$\tilde{E}_H := \textstyle\bigcup_{L=1}^{H} \left\{ (I, J) : I \in U_L, J \in U_{L-1},\ [I]_1^L = J \right\}.$$

The overall motion planner then operates as follows. At each iteration, a multi-resolution cell decomposition is first constructed. The cells in this decomposition may be categorized into two classes: cells with sizes at most $\bar{d}$, and cells with sizes greater than $\bar{d}$. We define *boundary cells* as the cells of sizes at most $\bar{d}$ that have at least one neighboring cell in each of the two previously defined classes (see Fig. 4.17(a)). A multiple-source, single-goal implementation of the A* algorithm may be used to determine the costs of optimal paths in the multi-resolution cell decomposition graph from the vertices associated with each of the boundary cells to the goal vertex. These costs are then used as terminal penalty costs in the execution of the H-cost path planner on the "partially" lifted graph previously discussed. This H-cost path planner returns a sequence of cells from the current location to one of the boundary cells, along with an admissible vehicle control input that enables the traversal of this sequence of cells. The vehicle state is advanced by traversing one cell using this control input, and the process is repeated until the vehicle reaches the goal.

The pseudo-code for the overall motion planner is provided in Fig. 4.8. Note that Line 7 of Fig. 4.8 involves local trajectory generation (TilePlan) for the particular vehicle dynamical model considered. We refer the reader to [59] for further details on finding the shortest path in the lifted graph in conjunction with TilePlan.



(a) Map of the environment.

(b) Resultant path.

Figure 4.9: Illustration of the multi-resolution path planning algorithm's ability to recover from a cul-de-sac: the red-colored cells were multiply visited.

4.5 Simulation Results and Discussion

In this section, we present numerical simulation results of implementations of the proposed multi-resolution path- and motion planning schemes. All of these simulations were implemented in the MATLAB® environment, on a computer with an Intel® Core™ i5-2410M 2.3 GHz CPU and 4 GB RAM.

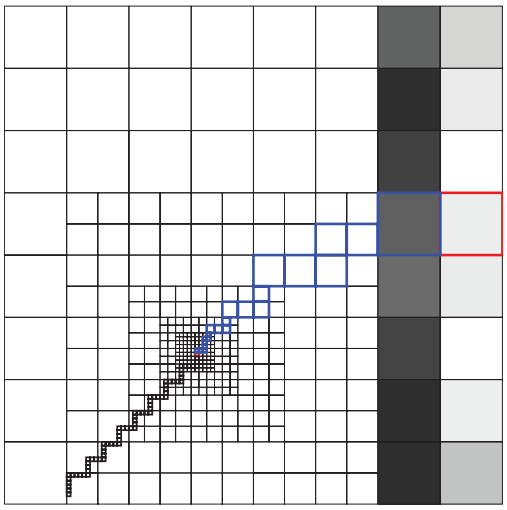
4.5.1 Completeness of the Path Planning Algorithm

First, we focus on the path planning algorithm, which does not consider vehicle dynamics. Figures 4.9 and 4.10 illustrate a simulation example demonstrating the capability of the multi-resolution path planning scheme to recover from a cul-de-sac. Although this situation does not apply for aircraft navigation problems, nonetheless, this simulation "illustrates" the theoretical completeness of the proposed path planner.

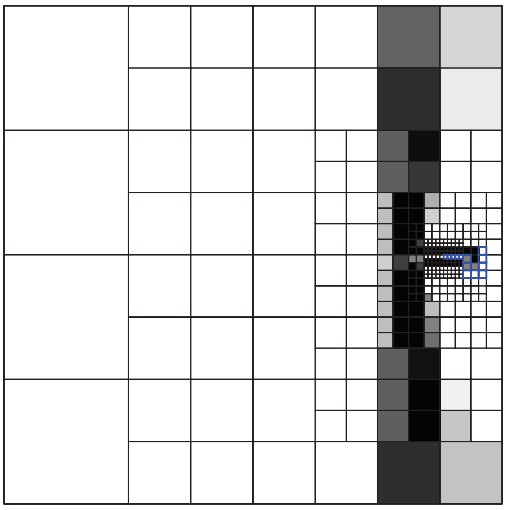
As shown in Fig. 4.9(a), we designed the shape of the obstacle and the location of the goal to lead the multi-resolution path planning algorithm into the cul-de-sac in the "central" region of the obstacle, whereas the goal can only be reached from the "top" region of the obstacle. Figure 4.10 illustrates some intermediate iterations in the execution of the multi-resolution path planning algorithm on this environment. Specifically, the algorithm leads the vehicle into the cul-de-sac but in later iterations, it successfully recovers and finds a path to the goal.

4.5.2 Optimality of the Path Planning Scheme

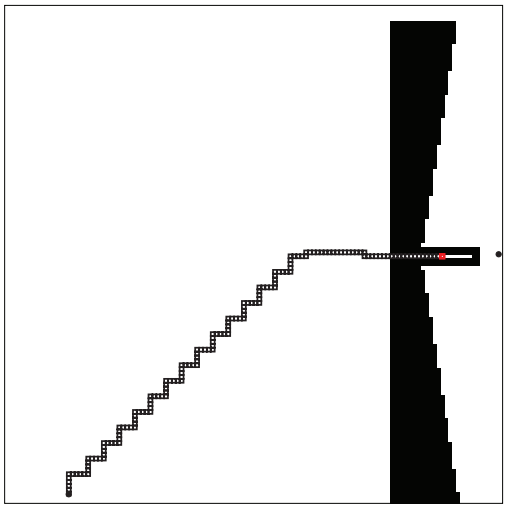
Whereas we can guarantee the algorithm's capability of finding a *feasible* path whenever such a path exists, we do not yet have theoretical results on the *optimality* of the resultant path. Here, we present numerical simulation results concerning the optimality of paths resulting from the multi-resolution path planning algorithm.



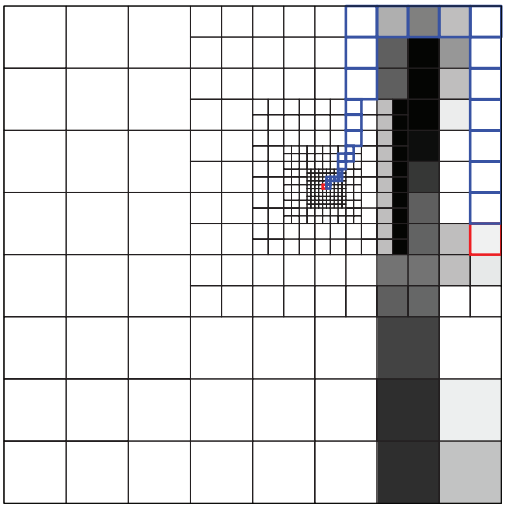
(a) An iteration before the cul-de-sac is explored.



(b) The iteration at which the cul-de-sac is encountered.



(c) The location of the vehicle at the iteration illustrated in Fig. 4.10(b).



(d) The iteration at which the algorithm finds a channel that contains a path to the goal.

Figure 4.10: Intermediate iterations in the multi-resolution path planning algorithm's implementation for the environment shown in Fig. 4.9(a).

Table 4.1: Window function values

m	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
ϱ_1	1	1	2	2	2	2	2	2	3	3	4	4
ϱ_2	1	2	2	3	4	5	6	7	7	7	8	8
ϱ_3	3	3	5	5	6	7	8	8	9	9	10	10

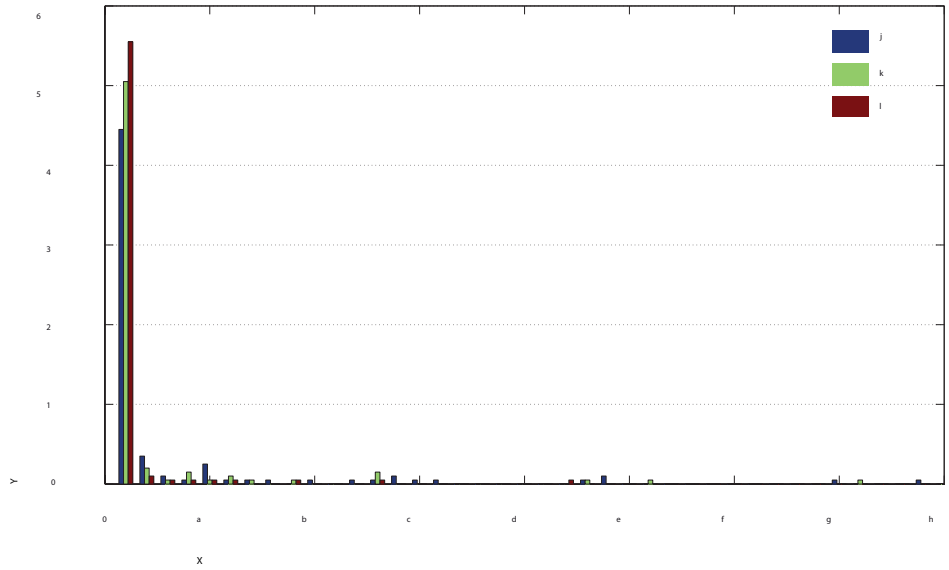


Figure 4.11: Histogram showing the distribution according to percentage sub-optimality of simulated cases, for different window functions.

We compared the cost of the resultant paths with the cost of an optimal path found by executing the A* algorithm on the finest level decomposition graph $\bar{\mathcal{G}}$. For these comparative simulations, we chose an environment represented by the image shown in Fig. 4.2(a), with three different "window" functions, as described in Table 4.1. The window ϱ_1 retains very few significant detail coefficients and results in a multi-resolution cell decomposition with high fidelity representation of the environment in a very small neighborhood of the vehicle's location, whereas the windows ϱ_2 and ϱ_3 result in decompositions with progressively larger neighborhoods of high-fidelity representations. We scaled the environment with $D = 6, 7, 8, 9$, and for each value of D, we performed 30 simulations with the initial and goal cells chosen randomly for each simulation. We executed the multi-resolution path planning algorithm proposed in Section 4.3 with each window function for each simulation (a total of 120 simulations for each window function), with $m_0 = -D$.

Figure 4.11 shows the distribution of the number of simulated cases according to percentage sub-optimality, where the cost of a path in $\bar{\mathcal{G}}$ by (4.6) was defined with $\lambda_1 = 1$ and $\lambda_2 = 0.1$. For all three window functions, the sub-optimality in most cases is under 20%, with window ϱ_3 resulting in the most cases of low sub-optimality, as intuitively expected. Overall, Fig. 4.11 shows that very few cases of extremely high sub-optimality occurred: these cases typically occurred when the algorithm encountered cul-de-sacs.

Figure 4.12 shows the distribution of the number of simulated cases according to sub-optimality for different values of the cost function parameters λ_1 and λ_2, all with the window function ϱ_1. From equation (4.6), note that λ_1 simply scales the image intensity, whereas λ_2 is a constant penalty on each edge in the path. As shown in Fig. 4.12, the proposed multi-resolution path planner results in

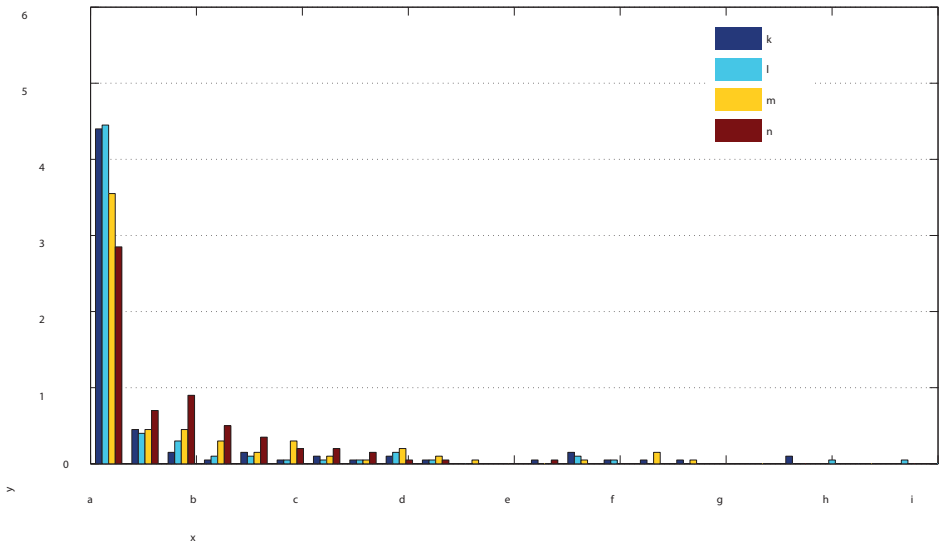


Figure 4.12: Histogram showing the distribution according to percentage sub-optimality of simulated cases, for different values of λ_2 with $\lambda_1 = 1$.

paths of low sub-optimality more often[2] for small values of λ_2. This behavior occurs due to the fact that, for each edge $(i, j) \in E(n)$, the expression (4.7) involves a worst-case estimate[3] of the number of vertices of $\bar{\mathcal{G}}$ in the path in $\bar{\mathcal{G}}$ corresponding to the path searched in $\mathcal{G}(n)$. Furthermore, by (4.7), the multi-resolution path planner's estimate of the cost of the actual path becomes progressively more conservative with increasing values of λ_2.

4.5.3 Performance of the Path Planning Scheme

Figure 4.13 shows the comparison of the (average) number of vertices in the graphs associated with the multi-resolution cell decompositions corresponding to different window functions and with the finest-level cell decomposition Ω. As expected, the window function ϱ_3, which has the largest neighborhood of high fidelity approximation of the environment (i.e., a large number of significant detail coefficients), results in cell decompositions with the largest number of cells among the three multi-resolution decompositions. Note, however, that the numbers of vertices in each of the three multi-resolution decompositions are of the same order of magnitude, whereas the numbers of vertices in $\bar{\mathcal{G}}$ are one to three orders of magnitude greater than those in the multi-resolution cell decomposition graphs. For instance, with $D = 9$, the number of vertices in $\bar{\mathcal{G}}$ was $262,144$, whereas the average number of vertices was only 561 for the multi-resolution cell decomposition with window ϱ_1. In this context, one may recall that the time complexity of the execution on a sparse graph $\mathcal{G} = (V, E)$ of Dijkstra's algorithm and the A* algorithm is $O(|V| \log |V|)$, whereas the memory complexity is $O(|V|)$ [114].

[2]Note that Fig. 4.12 shows a large number of cases of low sub-optimality for *all* values of λ_2.
[3]This worst-case estimate is necessary for completeness of the path planner.

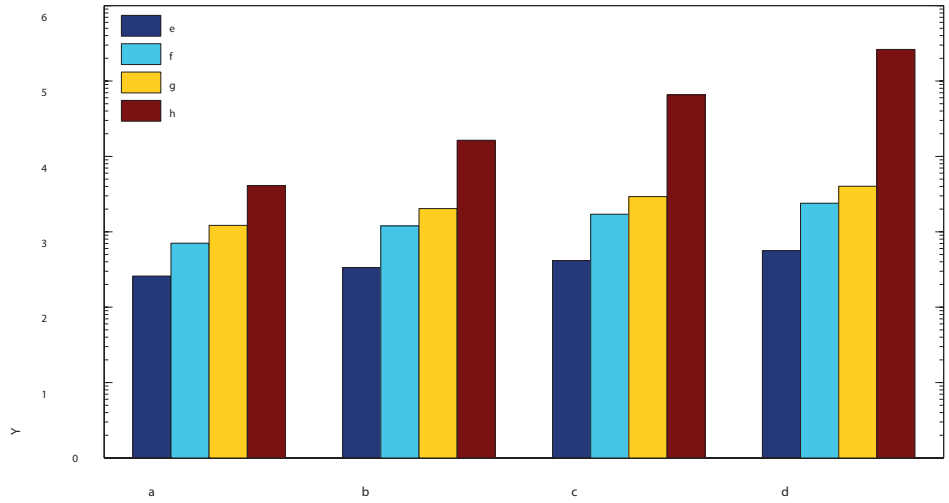


Figure 4.13: Comparison of the number of vertices in the multi-resolution cell decomposition graphs with different window functions and with the finest-level cell decomposition Ω.

4.5.4 Comparisons with Other Multi-resolution Path Planners

In this section we present a comparison between the proposed multi-resolution motion planning scheme and some of the standard multi-resolution planners reported in the literature (e.g. [25, 178, 175]). The comparison will be based on the numbers of vertices and edges of the resulting graph, as the main objective of all multi-resolution planners is to provide graph representations of the environment with low complexity. These graphs are then searched using standard algorithms such as the A* algorithm. This allows a fair comparison of the available multi-resolution motion cell decompositions, as the particular *search* algorithms of the resulting graph are the same across all such schemes.

The multi-resolution approximations of the environment reported in the literature belong to either of the following two broad classes: those governed primarily by the environment map, and those governed primarily by the vehicle's location in the environment. The former ones (such as those based on quadtree decompositions [104]) ensure that the resulting path will be entirely obstacle-free, but they tend to create larger graphs. The latter methods (e.g. [25]) result in smaller graphs, but obstacle-free cells are ensured only in the immediate vicinity of the vehicle's location, and replanning of paths is necessary as the vehicle moves through the environment. The main disadvantage of multi-resolution planners that use such vehicle location-dependent decompositions is that they are prone to a lack of completeness. The proposed approach belongs to the second class of planners, but we provide a guarantee of its completeness.

Table 4.2 provides a qualitative comparison between the proposed work and the various multi-resolution motion planning schemes reported in the literature.

To illustrate quantitatively our claim (echoed also in [25]) that environment-dependent cell decompositions usually consist of significantly more cells than vehicle location-dependent cell decompositions, we chose three environment maps: a terrain-like environment similar to Fig. 4.2(a), an environment consisting of a small number of large obstacles, and an environment consisting of a large number of small obstacles (see Fig. 4.14). We used the basic quadtree decomposition described in [104] as the basic example of an environment-dependent multi-resolution cell decomposition. Fig-

Table 4.2: Comparisons between various multi-resolution motion planners.

	Decomposition depends on	Completeness	Dynamical constraints
Ref. [104]	Environment map	Yes	No
Ref. [134]	Environment map	Yes[a]	No
Ref. [178]	Environment map	Yes[a]	No
Ref. [92]	Environment map	Yes[a]	No
Ref. [25]	Vehicle location	No[a]	No
Ref. [111]	Vehicle location	No[a]	No
Ref. [142]	Environment map and vehicle location	No[a]	Yes[b]
Ref. [175]	Vehicle location	No[a]	Yes
Proposed	Vehicle location	Yes	Yes

a]Not addressed. b]Dynamical constraints addressed separately from high-level path planning.

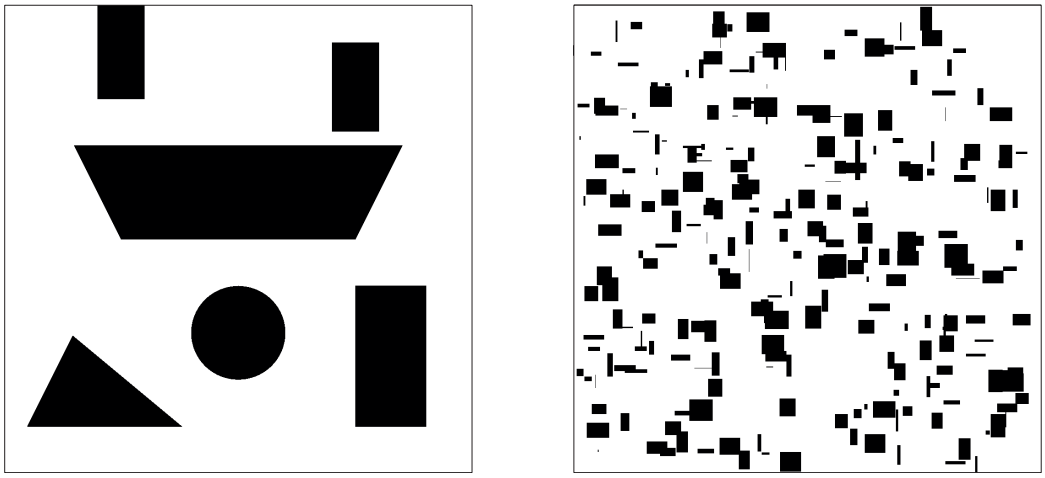
ure 4.15 shows a comparison of the number of vertices in the cell decomposition graphs arising from this quadtree decomposition, using different thresholds[4] and levels of decomposition, against those arising from the proposed wavelet-based decomposition, the latter using different window functions.

The number of cells from the environment-dependent quadtree decomposition for the environment with a sparse obstacle distribution is an order of magnitude larger than that obtained by the proposed vehicle-dependent decomposition. Improvements to environment-dependent decompositions, such as allowing for large "gray" cells, can reduce the number of vertices by an order of magnitude [104], and hence for this environment with sparse obstacles, we may consider the two schemes of decomposition to be equally efficient. However, for the terrain-like environment, where cell intensities take values in the interval $[0, 1]$ (as opposed to binary values in the previous case), the difference in the number of cells is up to three orders of magnitude. A similar observation holds true for the highly cluttered environment (Fig. 4.14(b)). Such a large difference in the number of vertices in the cell decomposition graph may render infeasible the implementation of environment-dependent decompositions (especially for large environments) for on-board computing systems with limited processing and memory resources.

Cell decompositions governed by the vehicle's location in the environment are numerically more efficient. However, as the corresponding graph changes with the vehicle's motion, the completeness of the overall path planning scheme is *not* guaranteed a priori, even though the graph search algorithm used at *each iteration* may be complete. Specifically, the path planner can get trapped in loops, where it visits a certain sequence of cells ad infinitum.

In addition to showing completeness, the most significant difference of our work compared to other similar works on multi-resolution path planning in the literature is the systematic incorporation of vehicle kinematic/dynamic constraints in path planning. In particular, we note in the third column of Table 4.2 that most of the other works do not address vehicle kinematic/dynamic constraints. Reference [142] discusses a receding horizon scheme for incorporating vehicle dynamical constraints, but this scheme is disconnected from the high-level discrete path planner. Consequently, there is no consistency between the two levels of planning (i.e., a guarantee that the path found by the

[4]The threshold τ, applicable for the terrain-like environment map, governs the quadtree decomposition as follows: a cell is decomposed further if and only if the difference in the maximum and minimum intensities of pixels within that cell exceeds τ.



(a) Sparse, large obstacles.

(b) Cluttered, small obstacles.

Figure 4.14: Environment maps used for comparative analysis.

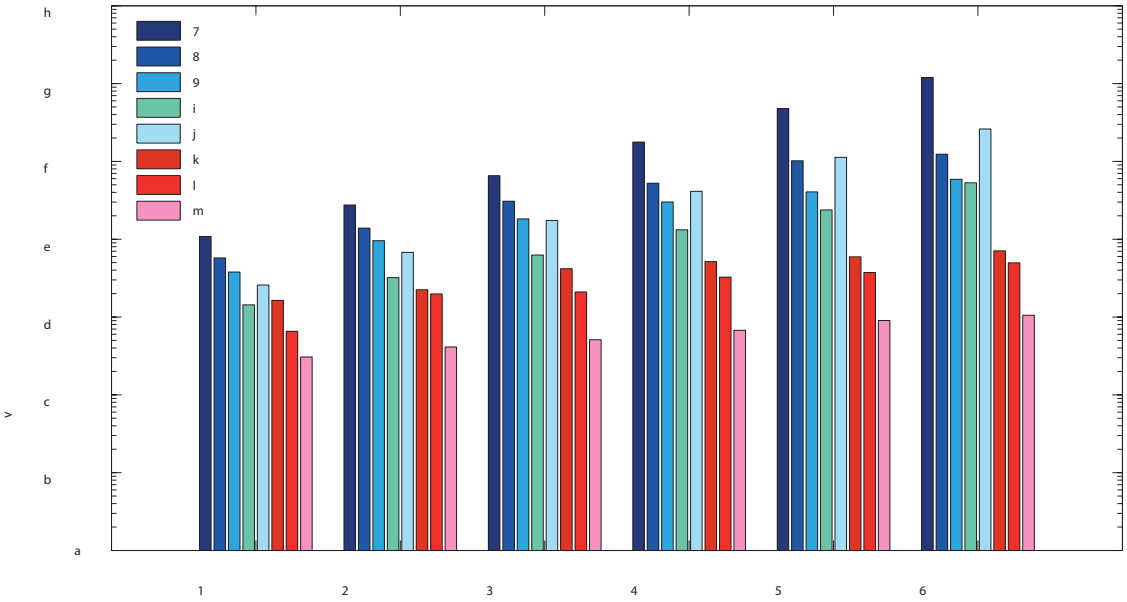


Figure 4.15: Comparison in the numbers of vertices of the resultant cell decompositions of the proposed approach against the quadtree decomposition.

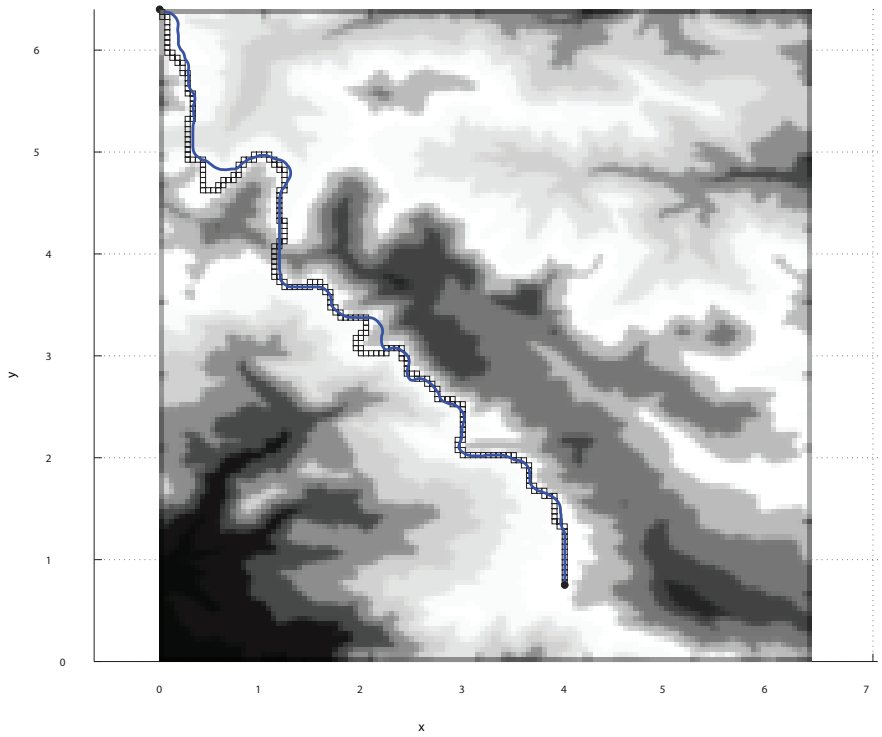


Figure 4.16: Result of motion planning simulation using the aircraft navigational model. The blue curve corresponds to the resultant state trajectory, while the channel of cells in black is the result of executing A* algorithm (without vehicle dynamical constraints). The initial position is at the top left corner.

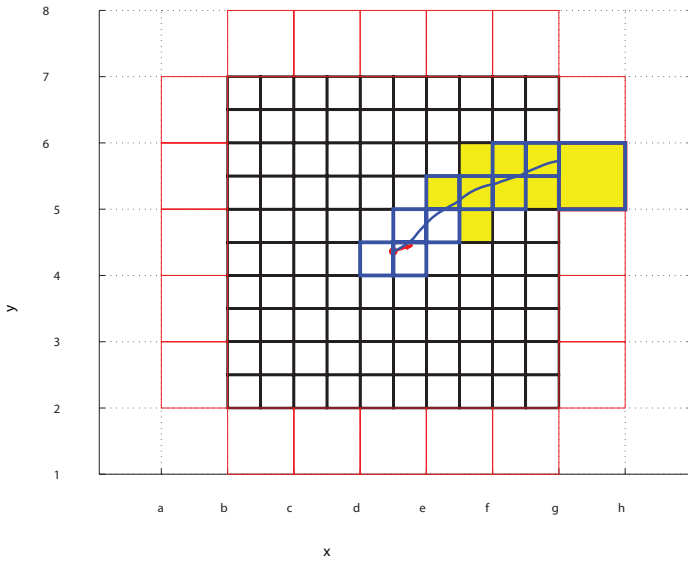
high-level planner can be feasibly traversed by the vehicle). This issue is addressed in this chapter via the H-cost motion planning approach discussed in Section 4.4. In [59], we have also provided extensive comparative analysis establishing the superiority of the H-cost approach using uniform cell decompositions over state-of-the-art randomized sampling-based algorithms.

4.5.5 Multi-resolution Motion Planning Example

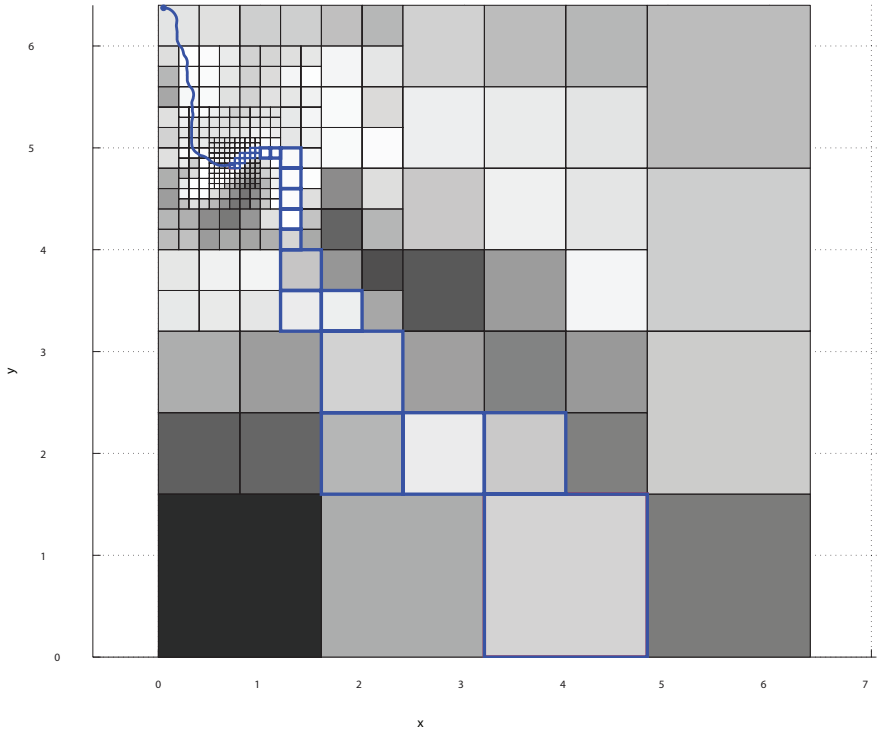
To illustrate a typical application of the overall multi-resolution motion planning scheme that incorporates the vehicle dynamic constraints, we consider the problem of navigating an aircraft amongst a topographic relief of varying elevation. The equations of motion and the implementation of a local trajectory generation algorithm for this vehicle are described in detail in [57].

Figure 4.16 shows the result of the numerical simulation of the proposed motion planner for the aircraft navigational model. The aircraft speed was assumed to be constant, and the control input is the heading angle, which is controlled by the bank angle. To show the flexibility of the algorithm in incorporating dynamic constraints, an asymmetric bound on the bank angle control input was assumed (say, owing to an aileron failure [21]) as follows: $\phi_{\min} = -45^\circ$ and $\phi_{\max} = 20^\circ$. The objective was to minimize a cost defined on the environment (indicated by regions of different intensities in Fig. 4.16, where the darker regions correspond to higher costs).

Figure 4.17 illustrates an intermediate iteration of this simulation example. Figure 4.17(a) shows the cells of size at most $\bar{d}$, with the boundary cells indicated in red. The sequence of cells outlined in blue



(a) Local perspective: the vehicle's configuration is indicated in red.



(b) Global perspective.

Figure 4.17: Illustration of an intermediate iteration of the overall motion planner.

and the blue-colored curve within this cell sequence are the results of the H-cost motion planner. The yellow-colored cells indicate the vertices explored during the H-cost search. Figure 4.17(b) shows the overall multi-resolution cell decomposition at the same iteration. The blue-colored cells indicate the optimal path to the goal from the boundary cell chosen by the H-cost motion planner. The blue-colored curve in Fig. 4.17(b) indicates the geometric path traversed by the vehicle in previous iterations.

Chapter 5

Optimal Synthesis of the Asymmetric Sinistral/Dextral Markov-Dubins Problem

5.1 Introduction

In this chapter, we consider the problem of finding curvature-constrained, planar paths of minimum-length with prescribed positions and tangents when the lower and upper bounds of the curvature are not necessarily equal. The motivation for this problem stems from vehicle navigation applications when the maneuverability of the vehicle when taking a left or a right turn is asymmetric. A typical case would be an aircraft with a damaged aileron as shown in [67]. Henceforth, we shall refer to this generalization of the standard MD problem as the Asymmetric, Sinistral/Dextral[1] Markov-Dubins problem (ASDMD for short). We formulate the ASDMD problem as a minimum-time problem, and we investigate its (time-) optimal synthesis, that is, a) we characterize a family of extremal controls that is sufficient for optimality; b) we provide a state-feedback minimum-time control scheme; and finally c) we compute analytically the level sets of the minimum-time. Additionally, the synthesis problem of the ASDMD when the tangent of the curve at the terminal position is free is also considered, leading us to the analytic characterization of the set of points that can be reached by curves that satisfy asymmetric curvature constraints.

These Dubins paths can be used as geometric path primitives in the case there are no obstacles. We will see in later chapters how these Dubins paths (or suitable modifications thereof) can be used to generate time-optimal paths or (after time-parameterization) as initial guesses for a higher-fidelity trajectory optimizer.

5.2 Kinematic Model and Problem Formulation

We consider a vehicle whose motion is defined by the following kinematic equations

$$\dot{x} = \cos\theta, \quad \dot{y} = \sin\theta, \quad \dot{\theta} = u/\rho, \tag{5.1}$$

[1]The term sinistral (dextral) means "inclined to left (right)" [1].

where (x, y) are the Cartesian coordinates of a reference point of the vehicle, θ is the direction of motion of the vehicle, u is the control input and ρ is a positive constant. We assume that the set of admissible control inputs $\mathcal{U}$ consists of all measurable functions u over $[0, T]$, where $t > 0$, with $u \in U_\delta \triangleq [-\delta, 1]$, where $\delta \in (0, 1]$. If we let $\varrho \triangleq \rho/\delta$, then it follows that ρ and ϱ are the minimum turning radii for counterclockwise and clockwise turns, respectively. The case $U_\delta = [-1, \delta]$ can be treated similarly. We call the system described by (5.1) and input value set U_δ the asymmetric, sinistral/dextral Isaacs-Dubins (ASDID) car.

It is a well-known fact that the standard ID car is completely controllable [170]. Next, it is shown that the ASDID car is also completely controllable. The controllability of the ASDID is established by proving that the equations (5.1) with input value set $U'_\delta = [-\delta, \delta] \subset U$ define a completely controllable system. It suffices to note that the system (5.1) with input value set U'_δ is the standard ID car with minimum turning radius ϱ (for both left and right turns), which is a completely controllable system.

It is worth-noting that the assumption that $\delta \in (0, 1]$, which guarantees that 0 is an interior of the input value set, can be actually relaxed, and it can be assumed instead that $\delta \in [0, 1]$ (in this case the vehicle cannot take right turns). A proof of the complete controllability in this case, which is based on solely geometric arguments, can be found in [17].

Next, we formulate the following minimum-time problem with fixed initial and terminal boundary conditions for the system (5.1).

Problem 5.1. Given the system described by equations (5.1) and the cost functional

$$J(u) = \int_0^{T_\mathrm{f}} 1 \, \mathrm{d}t = T_\mathrm{f}, \tag{5.2}$$

where T_f is the free final time and $\mathsf{x} : [0, T_\mathrm{f}] \mapsto \mathbb{R}^2 \times \mathbb{S}^1$ with $\mathsf{x} = (x, y, \theta)$ is the trajectory generated by the control $u \in \mathcal{U}$, determine the control input $u^* \in \mathcal{U}$ such that

i) The trajectory $\mathsf{x}^* : [0, T_\mathrm{f}] \mapsto \mathbb{R}^2 \times \mathbb{S}^1$ generated by the control u^* satisfies the boundary conditions

$$\mathsf{x}^*(0) = (0, 0, 0), \qquad \mathsf{x}^*(T_\mathrm{f}) = (x_\mathrm{f}, y_\mathrm{f}, \theta_\mathrm{f}). \tag{5.3}$$

ii) The control u^* minimizes the cost functional $J(u)$ given in (5.2).

To show the existence of an optimal solution to Problem 5.1, one can apply Filippov's theorem on minimum-time problems with prescribed initial and terminal states [46], leading to the following proposition.

Proposition 5.1. The minimum-time Problem 1 with boundary conditions (5.3) has a solution for all $(x_\mathrm{f}, y_\mathrm{f}, \theta_\mathrm{f}) \in \mathbb{R}^2 \times \mathbb{S}^1$.

5.3 Analysis of the ASDMD Minimum-Time Problem

In this section, we characterize the structure of the optimal paths using a similar approach as in [170, 173]. To this end, consider the Hamiltonian $\mathcal{H} : \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^3 \mapsto \mathbb{R}$ of Problem 5.1, which is defined by

$$\mathcal{H}(\mathsf{x}, \mathsf{p}, u) = p_0 + p_1 \cos\theta + p_2 \sin\theta + p_3 u/\rho, \tag{5.4}$$

From Pontryagin Maximum Principle (PMP), it follows that if x^* is a minimum-time trajectory generated by the control u^*, then there exists a scalar $p_0^* \in \{0, 1\}$ and an absolutely continuous function $\mathsf{p}^* : [0, T_\mathrm{f}] \mapsto \mathbb{R}^3$, where $\mathsf{p}^* = (p_1^*, p_2^*, p_3^*)$, known as the costate, such that

i) $\|\mathsf{p}^*(t)\| + |p_0^*|$ does not vanish for all $t \in [0, T_\mathrm{f}]$,

ii) $\mathsf{p}^*(t)$ satisfies for almost all $t \in [0, T_\mathrm{f}]$ the canonical equation $\dot{p}^* = -\partial\mathcal{H}(\mathsf{x}^*, \mathsf{p}^*, u^*)/\partial\mathsf{x}$, which for the system (5.1) reduces to

$$\dot{p}_1^* = 0, \quad \dot{p}_2^* = 0, \quad \dot{p}_3^* = p_1^* \sin\theta^* - p_2^* \cos\theta^*, \tag{5.5}$$

iii) $\mathsf{p}^*(T_\mathrm{f})$ satisfies the transversality condition associated with the free final-time Problem 5.1

$$\mathcal{H}(\mathsf{x}^*(T_\mathrm{f}), \mathsf{p}^*(T_\mathrm{f}), u^*(T_\mathrm{f})) = 0. \tag{5.6}$$

Because the Hamiltonian does not depend explicitly on time, it follows from (5.6) that

$$\mathcal{H}(\mathsf{x}^*(t), \mathsf{p}^*(t), u^*(t)) = 0, \tag{5.7}$$

for almost all $t \in [0, T_\mathrm{f}]$, which furthermore implies, that

$$-p_0^* = p_1^*(0) \cos\theta^* + p_2^*(0) \sin\theta^* + p_3^* u^*/\rho. \tag{5.8}$$

Furthermore, the optimal control u^* satisfies

$$\mathcal{H}(\mathsf{x}^*(t), \mathsf{p}^*(t), u^*(t)) = \min_{v \in [-\delta, 1]} \mathcal{H}(\mathsf{x}^*(t), \mathsf{p}^*(t), v), \tag{5.9}$$

for almost every $t \in [0, T_\mathrm{f}]$. It follows that

$$u^*(t) = \begin{cases} +1, & \text{if} \quad p_3^*(t) < 0, \\ \quad v \in [-\delta, 1], & \text{if} \quad p_3^*(t) = 0, \\ -\delta, & \text{if} \quad p_3^*(t) > 0. \end{cases} \tag{5.10}$$

Using similar arguments as in [170, 173] one can show the following proposition.

Proposition 5.2. The optimal control u^* of Problem 5.1 belongs necessarily to U^*, where

$$\mathsf{U}^* \triangleq \{\{\mathsf{u}^\pm, 0, \mathsf{u}^\pm\}, \{\mathsf{u}^\pm, 0, \mathsf{u}^\mp\}, \{\mathsf{u}^\pm, \mathsf{u}^\mp, \mathsf{u}^\pm\}\}, \quad \mathsf{u}^+ \triangleq 1, \quad \mathsf{u}^- \triangleq -\delta. \tag{5.11}$$

Proposition 5.2 implies that the time-optimal paths of Problem 5.1 are necessarily concatenations of at most three segments, namely two bang arcs denoted by b^- (along which $u^* = -\delta$) and b^+ (along which $u^* = 1$), respectively, and a singular arc, denoted as s (along which $u^* = 0$). Note that b^- and b^+ arcs correspond to circular arcs of radius ϱ and ρ respectively, whereas a singular arc s corresponds to a straight line segment. It follows that a minimum-time paths of Problem 5.1 has necessarily one the following structures

i) $\mathsf{b}_\alpha^-\mathsf{s}_\beta\mathsf{b}_\gamma^-$, $\mathsf{b}_\alpha^+\mathsf{s}_\beta\mathsf{b}_\gamma^+$, $\mathsf{b}_\alpha^-\mathsf{s}_\beta\mathsf{b}_\gamma^+$ and $\mathsf{b}_\alpha^+\mathsf{s}_\beta\mathsf{b}_\gamma^-$,

ii) or $\mathsf{b}_\alpha^+\mathsf{b}_\beta^-\mathsf{b}_\gamma^+$ and $\mathsf{b}_\alpha^-\mathsf{b}_\beta^+\mathsf{b}_\gamma^-$,

where the subscripts α, β, γ denote the duration of motion along the first, second, and third path segment, respectively.

Proposition 5.2 provides us with six families of paths that suffice to connect any pair of prescribed initial and terminal configurations in $\Re^2 \times \mathbb{S}^1$ similarly to the solution of the standard MD problem. Although we have significantly reduced the family of candidate paths that solve Problem 5.1, no information regarding the switching times is yet available. In particular, an analysis over the switching times will allow us to significantly refine the family of the extremals before we proceed further to the synthesis problem.

To this end, let us consider an open interval $\mathcal{I} \subset [0, T_\mathrm{f}]$ for which $p_3^*(t) \neq 0$ for all $t \in \mathcal{I}$. The restriction of the optimal control u^* on $\mathcal{I}$ is a piecewise constant function which jumps at most twice, and $u^*(t) \in \{-\delta, +1\}$ for all $t \in \mathcal{I}$. For any subinterval $\mathcal{I}_\mathsf{b}$ of $\mathcal{I}$ where u^* is constant, p_3^* satisfies

$$\ddot{p}_3^* = -\left(\frac{u^*}{\rho}\right)^2 p_3^* - \frac{u^* p_0^*}{\rho}, \tag{5.12}$$

for all $t \in \mathcal{I}_\mathsf{b}$. The general solution of (5.12) and its derivative for all $t \in \mathcal{I}$ are given by

$$p_3^*(t) = C_1 \cos\frac{u^* t}{\rho} + C_2 \sin\frac{u^* t}{\rho} - \frac{\rho p_0^*}{u^* t}, \tag{5.13}$$

$$\dot{p}_3^*(t) = \frac{C_2 u^*}{\rho}\cos\frac{u^* t}{\rho} - \frac{C_1 u^*}{\rho}\sin\frac{u^* t}{\rho}, \tag{5.14}$$

where C_1, C_2 are real constants. It follows readily that

$$\left(\frac{\rho \dot{p}_3^*(t)}{u^*}\right)^2 + \left(p_3^*(t) + \frac{\rho p_0^*}{u^*}\right)^2 = C_1^2 + C_2^2, \quad t \in \mathcal{I}_\mathsf{b}. \tag{5.15}$$

The phase portrait of $(p_3^*, \dot{p}_3^* \rho)$ is illustrated in Fig. 5.1 for the normal case ($p_0^* = 1$) and in Fig. 5.2 for the abnormal case ($p_0^* = 0$). In contrast to the standard MD, the phase portrait of $(p_3^*, \dot{p}_3^* \rho)$ is not symmetric w.r.t. the axis $p_3 = 0$ (compare for example, with Ref. [173]).

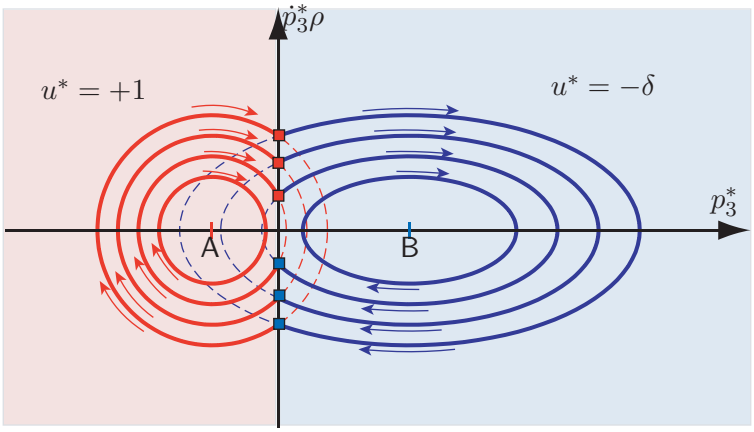


Figure 5.1: Phase portrait of $(p_3^*, \dot{p}_3^* \rho)$. Normal case $p_0^* = 1$.

Proposition 5.3. An $\mathsf{b}_\alpha^- \mathsf{b}_\beta^+ \mathsf{b}_\gamma^-$ $\left[\mathsf{b}_\alpha^+ \mathsf{b}_\beta^- \mathsf{b}_\gamma^+\right]$ path with $\min\{\alpha, \beta, \gamma\} > 0$ corresponds to an optimal trajectory of Problem 5.1 only if

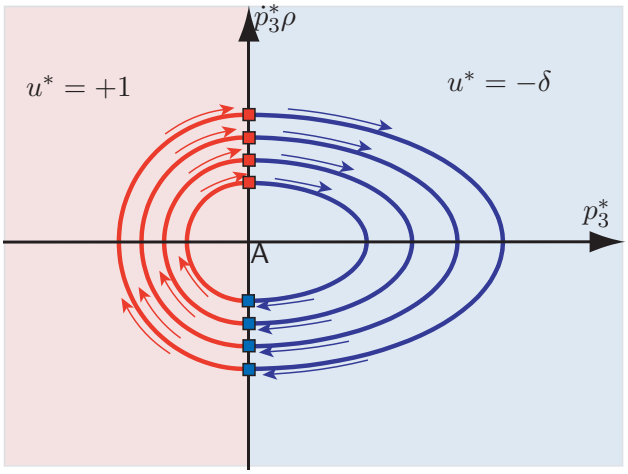


Figure 5.2: Phase portrait of $(p_3^*, \dot{p}_3^*\rho)$. Abnormal case $p_0^* = 0$.

i) $\beta \in (\pi\rho, 2\pi\rho), \quad [\beta \in (\pi\varrho, 2\pi\varrho)]$

ii) $\max\{\alpha, \gamma\} \leq \varepsilon(\delta, \beta)$, where

$$\varepsilon(\delta, \beta) = 2\pi\varrho + 2\varrho \operatorname{atan}\left(\delta \tan\frac{\beta}{2\rho}\right), \quad \left[\varepsilon(\delta, \beta) = 2\pi\rho + 2\rho \operatorname{atan}\left(\delta^{-1} \tan\frac{\beta}{2\varrho}\right)\right], \tag{5.16}$$

iii) $\min\{\alpha, \gamma\} < (\beta - \pi)\varrho, \quad [\min\{\alpha, \gamma\} < (\beta - \pi)\rho]$.

Proof. We consider an $\mathsf{b}_\alpha^-\mathsf{b}_\beta^+\mathsf{b}_\gamma^-$ path. The case of an $\mathsf{b}_\alpha^+\mathsf{b}_\beta^-\mathsf{b}_\gamma^+$ path can be treated similarly. First, we consider the abnormal case $p_0^* = 0$. It follows from Fig. 5.2 that a point in the $(p_3^*, \rho\dot{p}_3^*)$ plane stays in the half plane $p_3 \leq 0$ for exactly $\beta = \pi\rho$ units of time, which is the time required for a particle with coordinates $(p_3^*, \rho\dot{p}_3^*)$ to travel half of the circumference of a circle centered at the origin with constant angular speed $\omega = 1/\rho$. However, using the same geometric argument as in Lemma 23 in [170] we can show that the resulting path with $\beta = \pi\rho$ is not optimal. Hence, all optimal extremals of $\mathsf{b}_\alpha^-\mathsf{b}_\beta^+\mathsf{b}_\gamma^-$ type must be normal.

We therefore let $p_0^* = 1$ in (5.12)-(5.15). In Fig. 5.3, we observe that the phase portrait of $(p_3^*, \rho\dot{p}_3^*)$ consists of a circle centered at A, denoted as C_A and an ellipse centered at B, denoted as E_B, in the $p_3^* < 0$ and $p_3^* > 0$ half-planes respectively, with parameterizations that trace both of them clockwise and such that a full loop along them requires $2\pi\rho$ and $2\pi\varrho$ units of time, respectively. Note that a jump from $u^* = -\delta$ to $u^* = +1$, and vice versa, occurs only if E_B intersects C_A along the axis $p_3^* = 0$. If this intersection does occur, we denote as C and D the points of intersection. Let r and r_δ denote the distance of either C or D from A and B respectively. Then E_B and C_A intersect only if $r \geq \rho$ and $r_\delta \geq \varrho$, and furthermore $r_\delta = \sqrt{r^2 + \varrho^2 - \rho^2}$ as shown in Fig. 5.3.

From Fig. 5.3 it follows that β corresponds to the travel time of the point $(p_3^*, \rho\dot{p}_3^*)$ from D to C along the circle C_A. The times α and γ are upper bounded by the travel time from C to D along the ellipse E_B. We observe that $\pi\rho$ is a strict lower bound for β since $\rho > 0$ (note that β tends to $\pi\rho$ as A gets closer to O but can never reach as far as $\rho > 0$). Furthermore, $2\pi\rho$ and $2\pi\varrho$ are strict uppers bound for β and both α and γ, respectively. To see why the previous remark is true, it suffices to observe that the bang arcs $\mathsf{b}_{2\pi\varrho}^+$ and $\mathsf{b}_{2\pi\rho}^-$ correspond to two full circles driving the system (5.1) to the same state, and thus both $\mathsf{b}_{2\pi\varrho}^+$ and $\mathsf{b}_{2\pi\rho}^-$ cannot be part of an optimal solution.

Next, we improve the upper bound on α, γ. In particular, we observe in Fig. 5.3 that given β, where $\beta = 2(\pi - \widehat{\mathsf{CAO}})\rho$, then α or γ is maximized if the point $(p_3^*, \rho\dot{p}_3^*)$ coincides with C at $t = 0$ or D at $t = T_\mathrm{f}$, respectively. Thus, $\max\{\alpha, \gamma\} \leq 2(\pi - \widehat{\mathsf{DBO}})\varrho$. By using simple geometric arguments, along with the fact that $\delta \in (0, 1]$, it follows that $\widehat{\mathsf{DBO}} = \operatorname{atan}\left(\delta \tan \widehat{\mathsf{CAO}}\right)$. Thus, $\max\{\alpha, \gamma\} \leq 2\left(\pi - \operatorname{atan}\left(\delta \tan \widehat{\mathsf{CAO}}\right)\right)\varrho$, and $\beta = 2(\pi - \widehat{\mathsf{CAO}})\rho$. Equation (5.16) follows immediately.

Finally, the third condition of the Proposition is proved by means of simple geometric arguments as in Lemma 3 of [39]. □

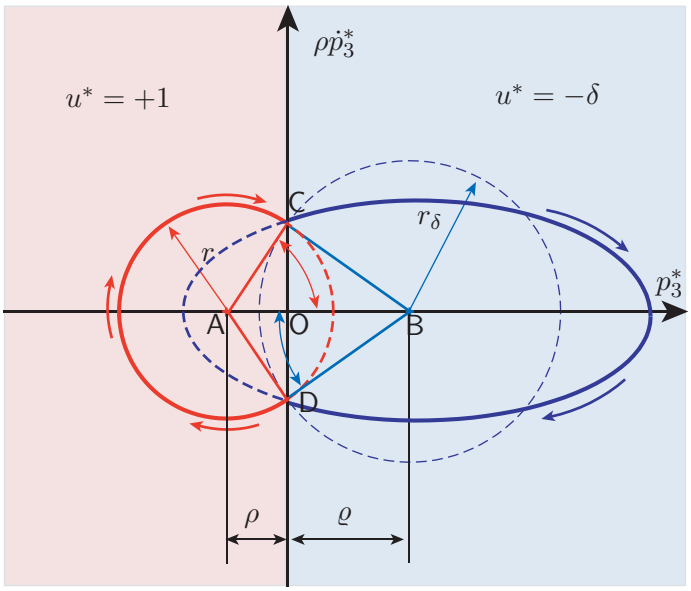


Figure 5.3: Phase portrait $(p_3^*, \rho\dot{p}_3^*)$.

Proposition 5.4. An $\mathsf{b}_\alpha^-\mathsf{s}_\beta\mathsf{b}_\gamma^-$ path corresponds to a time-optimal trajectory of Problem 5.1 only if $\alpha + \gamma \leq 2\pi\varrho$.

Proof. See the proof of Lemma 5 of Ref. [39]. □

Remark 5.1. Notice that Lemma 5 of [39] does not apply for $\mathsf{b}^+\mathsf{s}\mathsf{b}^+$ paths of the ASDMD problem. In particular, as is illustrated in Fig. 5.4, the ASDID car emanating from O reaches the terminal configuration $\mathsf{x}_\mathrm{f} = (x_\mathrm{f}, y_\mathrm{f}, \theta_\mathrm{f})$ by traversing an $\mathsf{b}_\alpha^+\mathsf{s}_\beta\mathsf{b}_\gamma^+$ path with $\alpha + \gamma > 2\pi\rho$. The total elapsed time is the same as if the ASDID car had traversed an $\mathsf{b}_\alpha^-\mathsf{s}_\beta\mathsf{b}_\gamma^-$ with $\alpha + \gamma \leq 2\pi$. Therefore, if the path $\mathsf{b}^-\mathsf{s}\mathsf{b}^-$ is time-optimal, then the $\mathsf{b}_\alpha^+\mathsf{s}_\beta\mathsf{b}_\gamma^+$ path is necessarily time-optimal as well. Thus, we conjecture that there exist $\mathsf{b}_\alpha^+\mathsf{s}_\beta\mathsf{b}_\gamma^+$ paths with $\alpha + \gamma > 2\pi\rho$, which are optimal paths of the ASDMD problem. As it is demonstrated in Section 5.4, our conjecture is indeed correct. Next we provide a conservative bound on the sum of α and γ along $\mathsf{b}_\alpha^+\mathsf{s}_\beta\mathsf{b}_\gamma^+$ paths.

Proposition 5.5. An $\mathsf{b}_\alpha^+\mathsf{s}_\beta\mathsf{b}_\gamma^+$ path corresponds to a time-optimal trajectory of Problem 5.1 only if $\alpha + \gamma \leq (4\pi - \theta_\mathrm{f})\rho$.

Finally, for $\mathsf{b}^-\mathsf{s}\mathsf{b}^+$ and $\mathsf{b}^+\mathsf{s}\mathsf{b}^-$ paths, as in the standard MD problem, we simply take the most conservative bounds. In particular, we have the following proposition.

Proposition 5.6. An $\mathsf{b}_\alpha^+\mathsf{s}_\beta\mathsf{b}_\gamma^-$ and an $\mathsf{b}_\alpha^-\mathsf{s}_\beta\mathsf{b}_\gamma^+$ path corresponds to a time-optimal trajectory of Problem 5.1 only if $\max\{\alpha, \delta\gamma\} < 2\pi\rho$ and $\max\{\delta\alpha, \gamma\} < 2\pi\rho$, respectively.

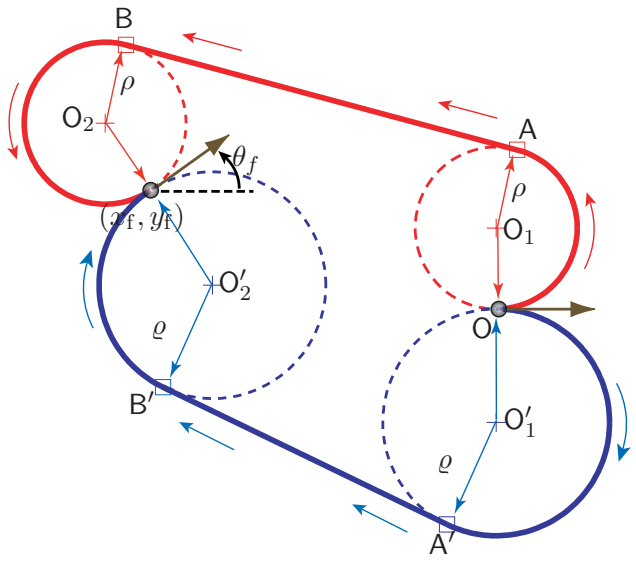


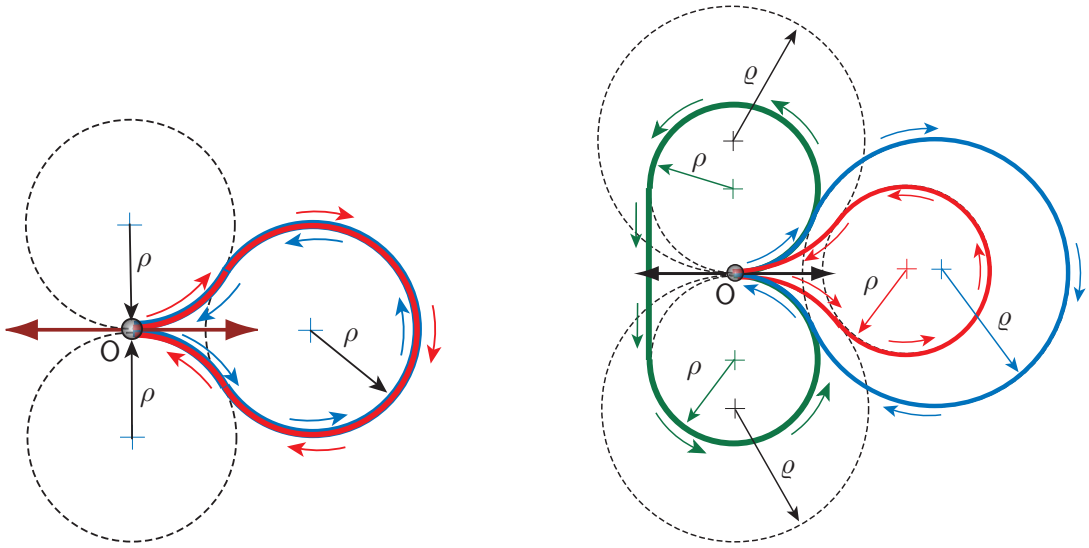
Figure 5.4: A $\mathsf{b}_\alpha^+\mathsf{s}_\beta\mathsf{b}_\gamma^+$ path with $\alpha+\gamma > 2\pi\rho$ can be an optimal solution of the ASDMD problem, in contrast to the solution of the standard MD problem.

5.4 Time-Optimal Synthesis

In this section, we address the time-optimal synthesis problem for the ASDMD problem, and thus provide a complete characterization of the optimal control that solves Problem 5.1 with boundary conditions (5.3), for all $(x_\mathrm{f}, y_\mathrm{f}, \theta_\mathrm{f}) \in \mathbb{R}^2 \times \mathbb{S}^1$.

First, we show by means of an example, that the synthesis of optimal paths for the ASDMD problem may be quite different than for the MD problem. In particular, let us consider the problem of characterizing the minimum-time path from $(0,0,0)$ to $(0,0,\pi)$ for the ID and the ASDID cars. On the one hand, the optimal solution of the standard MD problem is either a $\mathsf{b}_\alpha^+\mathsf{b}_\beta^-\mathsf{b}_\gamma^+$ path or a $\mathsf{b}_\alpha^-\mathsf{b}_\beta^+\mathsf{b}_\gamma^-$ path, where $\alpha = \gamma = \pi\rho/3$ and $\beta = 5\pi\rho/3$, as shown in Fig. 5.5(a) (these two paths have exactly the same length). On the other hand, as is illustrated in Fig. 5.5(b), the optimal path for the ASDMD problem is either an $\mathsf{b}_\alpha^-\mathsf{b}_\beta^+\mathsf{b}_\gamma^-$ path, where $\alpha = \gamma = \varrho\,\mathrm{acos}\,(1/(1+\delta))$ and $\beta = \pi\rho + 2\delta\alpha$ or an $\mathsf{b}_\alpha^+\mathsf{s}_\beta\mathsf{b}_\gamma^+$ path, where $\alpha = \gamma = 3\pi\rho/2$ and $\beta = 2\rho$. The $\mathsf{b}_\alpha^-\mathsf{b}_\beta^+\mathsf{b}_\gamma^-$ and the $\mathsf{b}_\alpha^+\mathsf{s}_\beta\mathsf{b}_\gamma^+$ paths have exactly the same length when $\delta = \tilde{\delta}$, where $\tilde{\delta}$ is the solution of the equation: $1/(1+\delta) + \cos\left((\pi-\delta)/(1+\delta)\right) = 0$. Note that for this specific problem, the $\mathsf{b}_\alpha^+\mathsf{s}_\beta\mathsf{b}_\gamma^+$ path can never be an optimal path of the standard MD problem, in light of Lemma 5 of [39].

To simplify the presentation and without loss in generality, we henceforth consider the minimum trajectories of the ASDID car from $(0,0,0)$ to $(x_\mathrm{f}, y_\mathrm{f}, \theta_\mathrm{f}) \in P_{\theta_\mathrm{f}}$, where $P_{\theta_\mathrm{f}} \triangleq \{(x,y,\theta) \in \mathbb{R}^2 \times \mathbb{S}^1 \;:\; \theta = \theta_\mathrm{f}\}$ as suggested in [38, 39]. Let $\mathfrak{R}_{\theta_\mathrm{f}}(u)$ denote the reachable set that corresponds to the control sequence $u \in \mathsf{U}^*$. The coordinates of all points in P_{θ_f} that can be reached by means of a $\mathsf{b}^+\mathsf{s}\mathsf{b}^+$ control sequence can be expressed as functions of the times of motion along the three arcs of the path, namely α, β, and γ, by simply integrating the equations (5.1), from $t = 0$ to $t = \alpha$ for



(a) Standard MD problem. (b) Asymmetric sinistral/dextral Markov-Dubins problem.

Figure 5.5: The minimum-time paths for the steering problem from $(0,0,0)$ to $(0,0,\pi)$ for the ID and the ASDID cars.

$u=+1$, and subsequently, from $t=\alpha$ to $t=\alpha+\beta$ for $u=0$, and finally, from $t=\alpha+\beta$ to the final time $T_{\mathrm{f}}(\mathsf{b}^+\mathsf{sb}^+)=\alpha+\beta+\gamma$. Note that γ can always be expressed in terms of the parameters α and β (actually for a $\mathsf{b}^+\mathsf{sb}^+$ path γ depends only on α as we shall see shortly later). In particular, since the total change of the velocity direction θ (initially $\theta=0$) along the path $\mod 2\pi$ must equal θ_{f}, it follows readily that $\alpha/\rho+\gamma/\rho \mod 2\pi=\theta_{\mathrm{f}}$, which furthermore implies that

$$\gamma(\alpha)=\begin{cases}\rho\theta_{\mathrm{f}}-\alpha, & \text{if } \theta_{\mathrm{f}}\geq\dfrac{\alpha}{\rho},\\ \rho(2\pi+\theta_{\mathrm{f}})-\alpha, & \text{if } \theta_{\mathrm{f}}<\dfrac{\alpha}{\rho}.\end{cases} \tag{5.17}$$

It follows after routine calculations that

$$x_{\mathrm{f}}(\alpha,\beta)=\rho\sin\theta_{\mathrm{f}}+\beta\cos\frac{\alpha}{\rho},\qquad y_{\mathrm{f}}(\alpha,\beta)=\rho+\beta\sin\frac{\alpha}{\rho}-\rho\cos\theta_{\mathrm{f}}. \tag{5.18}$$

Furthermore, Proposition 5.5 determines the intervals of admissible values of α and β for a $\mathsf{b}^+\mathsf{sb}^+$ control sequence, denoted by $\mathcal{I}_\alpha(\mathsf{b}^+\mathsf{sb}^+)$ and $\mathcal{I}_\beta(\mathsf{b}^+\mathsf{sb}^+)$, respectively. Thus, the reachable set of the control sequence $\mathsf{b}^+\mathsf{sb}^+$ are constructed by determining all points $(x_{\mathrm{f}},y_{\mathrm{f}},\theta_{\mathrm{f}})\in P_{\theta_{\mathrm{f}}}$ for every pairs of $(\alpha,\beta)\in\mathcal{I}_\alpha(\mathsf{b}^+\mathsf{sb}^+)\times\mathcal{I}_\beta(\mathsf{b}^+\mathsf{sb}^+)$.

Conversely, given a point $(x_{\mathrm{f}},y_{\mathrm{f}},\theta_{\mathrm{f}})\in\mathfrak{R}_{\theta_{\mathrm{f}}}(\mathsf{b}^+\mathsf{sb}^+)$ one can determine the parameters α and β such that x_{f} and y_{f} satisfy (5.18). In particular, after some algebra, it follows from (5.18) that

$$\alpha(x_{\mathrm{f}},y_{\mathrm{f}})=\rho\,\mathrm{atan2}(B(y_{\mathrm{f}}),A(x_{\mathrm{f}})),\qquad \beta(x_{\mathrm{f}},y_{\mathrm{f}})=\sqrt{A^2(x_{\mathrm{f}})+B^2(y_{\mathrm{f}})}, \tag{5.19}$$

where $A(x_{\mathrm{f}})=x_{\mathrm{f}}-\rho\sin\theta_{\mathrm{f}}$, $B(y_{\mathrm{f}})=y_{\mathrm{f}}+\rho\cos\theta_{\mathrm{f}}-\rho$, and $\mathrm{atan2}:\mathbb{R}^2\mapsto[0,2\pi[$ is the two-argument arctangent function.

Figure 5.6(a) illustrates the reachable set $\mathfrak{R}_{\theta_{\mathrm{f}}}(\mathsf{b}^+\mathsf{sb}^+)$ of the ASDID car (note that for this path family the value of δ does not affect the geometry of the reachable set), whereas the same reachable

set for the standard ID car is illustrated in 5.6(b). We observe that the former set is a superset of the latter. This is because for the ASDMD problem α satisfies $\alpha+\gamma(\alpha) \leq (4\pi-\theta_{\mathrm{f}})\rho$ (Proposition 5.5), whereas for the standard MD problem it satisfies the stricter condition $\alpha+\gamma(\alpha) \leq 2\pi\rho$ (Lemma 5 of [39]).

Finally, after having established the connection between (α,β) and $(x_{\mathrm{f}},y_{\mathrm{f}})$ the total time $T_{\mathrm{f}}(\mathsf{b}^+\mathsf{sb}^+)$ is given, via (5.17), by

$$T_{\mathrm{f}}(\mathsf{b}^+\mathsf{sb}^+)=\begin{cases}\beta+\rho\theta_{\mathrm{f}}, & \text{if } \theta_{\mathrm{f}}\geq\alpha/\rho,\\ \beta+\rho(2\pi+\theta_{\mathrm{f}}), & \text{if } \theta_{\mathrm{f}}<\alpha/\rho.\end{cases} \tag{5.20}$$

The previous procedure can be applied mutatis mutandis for the rest of the control sequences from U^* (although the algebra, especially in the case of $\mathsf{b}^+\mathsf{b}^-\mathsf{b}^+$ or $\mathsf{b}^-\mathsf{b}^+\mathsf{b}^-$ paths, is significantly more evolved). The equations that give α and β as functions of x_{f} and y_{f}, and vice versa, as well as the minimum-time T_{f} for all the control sequences $u\in\mathsf{U}^*$ can be found in [16].

The next step involves the partitioning of $P_{\theta_{\mathrm{f}}}$ into at most six domains, denoted as $\mathfrak{R}^*_{\theta_{\mathrm{f}}}(u)$, where $u\in\mathsf{U}^*$, such that if $(x_{\mathrm{f}},y_{\mathrm{f}},\theta_{\mathrm{f}})\in\operatorname{int}\left(\mathfrak{R}^*_{\theta_{\mathrm{f}}}(u)\right)$, then $(x_{\mathrm{f}},y_{\mathrm{f}},\theta_{\mathrm{f}})$ cannot be reached faster with the application of $v\in\mathsf{U}^*$, where $v\neq u$. We shall refer to this partition of $P_{\theta_{\mathrm{f}}}$ as the optimal control partition of the ASDMD problem. The number of these domains can be strictly less than six in case the domain associated with a particular control sequence has an empty interior. As we shall see shortly after, such "pathological" cases arise in the time-optimal synthesis of the ASDMD problem in contrast to the optimal synthesis of the standard MD problem. The procedure required for the characterization of the domain over which the control sequence, say $\mathsf{b}^+\mathsf{sb}^+$, is optimal, is summarized below. We denote this domain by $\mathfrak{R}^*_{\theta_{\mathrm{f}}}(\mathsf{b}^+\mathsf{sb}^+)$. In particular, let $(x_{\mathrm{f}},y_{\mathrm{f}},\theta_{\mathrm{f}})\in\mathfrak{R}_{\theta_{\mathrm{f}}}(\mathsf{b}^+\mathsf{sb}^+)$, and let $\mathsf{U}^c(\mathsf{b}^+\mathsf{sb}^+)\subset\mathsf{U}^*$ denote the set of control sequences u that are different from $\mathsf{b}^+\mathsf{sb}^+$ and such that $(x_{\mathrm{f}},y_{\mathrm{f}},\theta_{\mathrm{f}})\in\mathfrak{R}_{\theta_{\mathrm{f}}}(u)$. Then $(x_{\mathrm{f}},y_{\mathrm{f}},\theta_{\mathrm{f}})\in\mathfrak{R}^*_{\theta_{\mathrm{f}}}(\mathsf{b}^+\mathsf{sb}^+)$ if and only if $T_{\mathrm{f}}(\mathsf{b}^+\mathsf{sb}^+)\leq\min_{u\in\mathsf{U}^c(\mathsf{b}^+\mathsf{sb}^+)}T_{\mathrm{f}}(u)$.

Figure 5.7 illustrates the optimal control partition of $P_{\pi/3}$ as well as the level sets of the minimum-time T_{f}, for different values of the ratio $\delta^{-1}=\varrho/\rho$. In particular, each domain of the partition $P_{\pi/3}$ is illustrated by a colored set, whereas the level sets of the minimum time are denoted by solid black lines. We observe that as the ratio ϱ/ρ increases, the domains $\mathfrak{R}^*_{\pi/3}(\mathsf{b}^+\mathsf{sb}^+)$, $\mathfrak{R}^*_{\pi/3}(\mathsf{b}^-\mathsf{sb}^+)$ and $\mathfrak{R}^*_{\pi/3}(\mathsf{b}^+\mathsf{sb}^-)$, primarily, and the domain $\mathfrak{R}^*_{\pi/3}(\mathsf{b}^-\mathsf{b}^+\mathsf{b}^-)$, secondary, expand against the domain $\mathfrak{R}^*_{\pi/3}(\mathsf{b}^-\mathsf{sb}^-)$ as well as the disconnected components of $\mathfrak{R}^*_{\pi/3}(\mathsf{b}^+\mathsf{sb}^-)$ and $\mathfrak{R}^*_{\pi/3}(\mathsf{b}^-\mathsf{sb}^+)$ that are close to the origin of $P_{\theta_{\mathrm{f}}}$. We observe, in particular, that for $\varrho/\rho=1.8$ (Fig 5.7(e)) the partition of $P_{\pi/3}$ consists of five domains since the domain $\mathfrak{R}^*_{\pi/3}(\mathsf{b}^+\mathsf{b}^-\mathsf{b}^+)$ is reduced to the empty set. Similarly, for $\varrho/\rho=2$ (Fig 5.7(f)) only four domains are non-empty since $\mathfrak{R}^*_{\pi/3}(\mathsf{b}^-\mathsf{sb}^-)=\mathfrak{R}^*_{\pi/3}(\mathsf{b}^+\mathsf{b}^-\mathsf{b}^+)=\varnothing$. In addition, we observe in Fig 5.7(a)-5.7(f) that the boundaries of each domain change significantly as the ratio ϱ/ρ varies.

5.5 Time Optimal Synthesis and Reachable Sets of the ASDMD when the Final Tangent of the Path is Free

In this section, we consider the optimal synthesis of Problem 5.1, when θ_{f} is assumed to be free. The solution of this variation of Problem 5.1 will allow us to characterize analytically the set of points in the plane that can be reached by curves with asymmetric curvature constraints. These reachable sets along with the level sets of the minimum time of the ASDMD problem, when θ_{f} is

free, exhibit a few notable features related to the existence/abscence of symmetry planes that are not present neither in the reachable sets nor the syntheses of the standard MD and the ASDMD problems, when θ_{f} is fixed. Favoring the economy of presentation, we shall not discuss in details the analysis of this problem, and we will instead present the solution of the time-optimal synthesis problem directly.

First, we discuss briefly the structure of the family of extremal controls, which is sufficient for optimality for Problem 5.1, when θ_{f} is free. In particular, the new transversality condition for θ is given by $p_3^*(T_{\mathrm{f}}) = 0$. Following the same line of arguments as in [173], where the standard MD, when θ_{f} is free, is addressed in detail, we conclude that a composite path whose final arc is either an b^- or an b^+ arc, that is preceded by an s arc, cannot be part of an optimal path. The following proposition gives us the family of candidate optimal controls for Problem 5.1, when θ_{f} is free (it follows similarly to [173]).

Proposition 5.7. The optimal control u^* of Problem 5.1, when θ_{f} is free, belongs necessarily to U^*, where

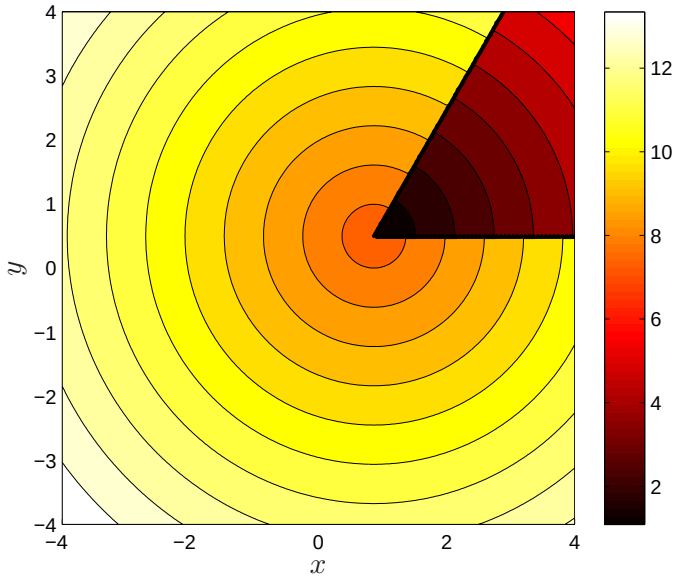
$$\mathsf{U}^* \triangleq \{\{\mathsf{u}^{\pm}, 0\}, \{\mathsf{u}^{\pm}, \mathsf{u}^{\mp}\}\}, \qquad \mathsf{u}^+ \triangleq 1, \qquad \mathsf{u}^- \triangleq -\delta. \tag{5.21}$$

Proposition 5.7 implies that the set of candidate optimal controls of Problem 5.1, when θ_{f} is free, consists of only four control sequences with at most one switching. It follows that the minimum-time paths of Problem 5.1, when θ_{f} is free, necessarily admit one of the following structures: *i)* $\mathsf{b}^+_\alpha \mathsf{b}^-_\beta, \mathsf{b}^-_\alpha \mathsf{b}^+_\beta$, *ii)* $\mathsf{b}^+_\alpha \mathsf{s}_\beta, \mathsf{b}^-_\alpha \mathsf{s}_\beta$.

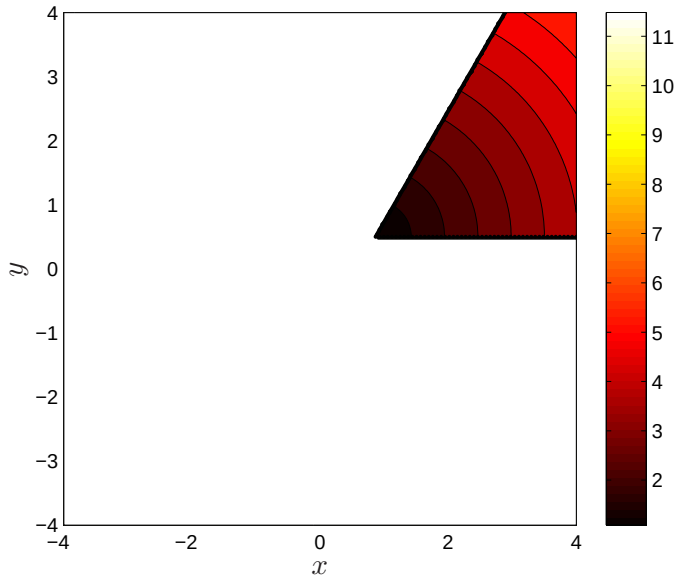
By repeating the analysis carried out in Sections 5.3 and 5.4, we can refine the family of candidate optimal controls (this analysis will lead to a number of propositions similar to Props. 5.3-5.6), and subsequently solve the synthesis problem for Problem 5.1, when θ_{f} is free. Favoring the economy of presentation, we directly show the solution of the synthesis problem. In particular, Figure 5.8 illustrates the optimal control partition of the plane as well as the level sets of the minimum-time T_{f}, when θ_{f} is free (assuming that the ID/ASDID car starts from the origin with $\theta = 0$) for both the standard ID car (Fig. 5.8(a)) and the ASDID car (Figs. 5.8(b)-5.8(d)). Figs. 5.8(b)- 5.8(d) illustrate that as the agility of the ASDID to perform right turns, which is measured by the ratio ϱ/ρ, is reduced, the sets $\mathfrak{R}^*(\mathsf{b}^-\mathsf{s})$ and $\mathfrak{R}^*(\mathsf{b}^+\mathsf{b}^-)$ "shrink" in favor of the set $\mathfrak{R}^*(\mathsf{b}^+\mathsf{s})$, whereas the set $\mathfrak{R}^*(\mathsf{b}^-\mathsf{b}^+)$ remains invariant under the variations of the ratio ϱ/ρ.

It is worth noting that contrary to the synthesis of the ASDMD problem, when θ_{f} is fixed, where both the level sets of the minimum-time and the domains of the optimal control partition are symmetric with respect to some plane of symmetry (also a characteristic of the optimal synthesis of the standard MD problem), both the level sets and the domains of the optimal control partition of the ASDMD problem, when θ_{f} is free, do not enjoy similar symmetry properties. It appears that the term "asymmetric" used in the title of this chapter is more obviously justified in the case when θ_{f} is free rather than when θ_{f} is fixed.

Let $\tau > 0$ and let $\mathfrak{R}^{\mathrm{s}}_{t\leq\tau}$ and $\mathfrak{R}^{\mathrm{asym}}_{t\leq\tau}$ denote the set of points in the plane that can be reached by the ID and ASDID car in time $t \in [0, \tau]$, respectively (assuming again that the ID/ASDID car starts from the origin with $\theta = 0$). The reachable sets $\mathfrak{R}^{\mathrm{asym}}_{t\leq\tau}$ for different values of τ are illustrated in Fig. 5.9. In Figs.5.9(a)-5.9(d), we observe that the reachable sets $\mathfrak{R}^{\mathrm{asym}}_{t\leq\tau}$ are not symmetric with respect to the x-axis by contrast to the sets $\mathfrak{R}^{\mathrm{s}}_{t\leq\tau}$ (see, for example, [36, 165]). This comes at no surprise, since both $\mathfrak{R}^{\mathrm{s}}_{t\leq\tau}$ and $\mathfrak{R}^{\mathrm{asym}}_{t\leq\tau}$ can be interpreted as the union of all the level sets $\{(x, y) : T_{\mathrm{f}} = t\}$, for $t \in [0, \tau]$, which, as we have already mentioned, are symmetric with respect to x-axis for the standard MD problem but not for the ASDMD problem, when θ_{f} is free.

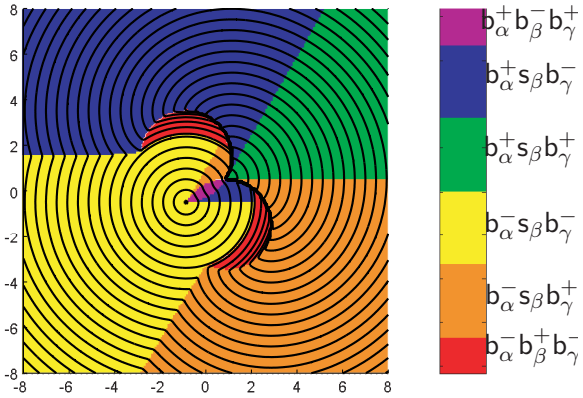


(a) $0 < \delta < 1$

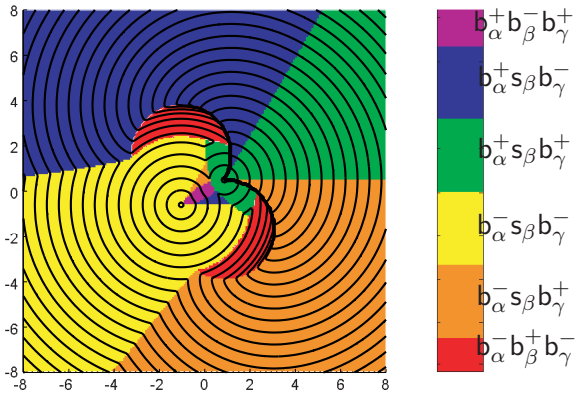


(b) $\delta = 1$ (standard MD)

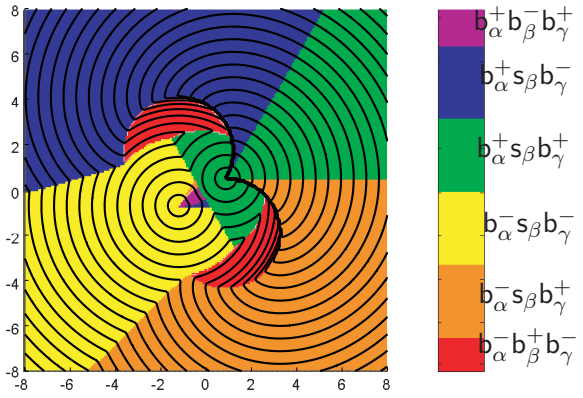
Figure 5.6: Reachable set $\mathfrak{R}_{\theta_f}(\mathsf{b}^+\mathsf{s}\mathsf{b}^+)$ for $\delta \in (0, 1)$ (ASDMD problem) and $\delta = 1$ (standard MD problem). The white colored region corresponds to terminal configurations that cannot be reached in minimum-time by means of a $\mathsf{b}^+\mathsf{s}\mathsf{b}^+$ control sequence for the standard MD problem.



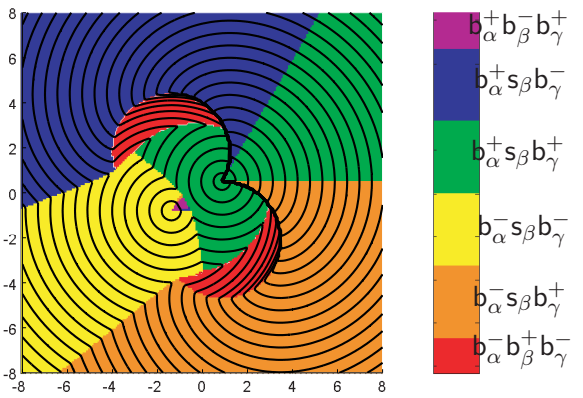
(a) $\varrho/\rho = 1.0$



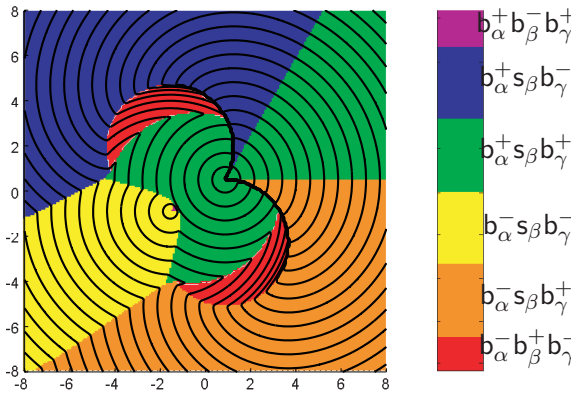
(b) $\varrho/\rho = 1.2$



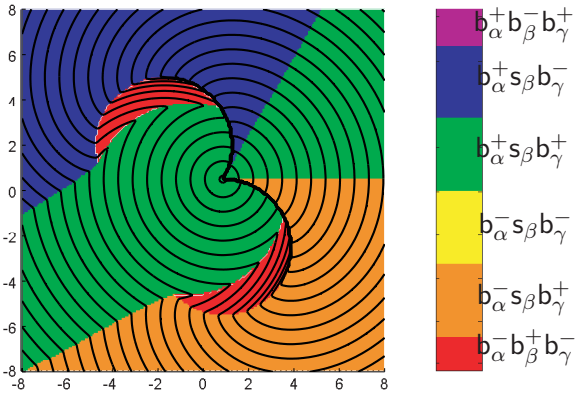
(c) $\varrho/\rho = 1.4$



(d) $\varrho/\rho = 1.6$

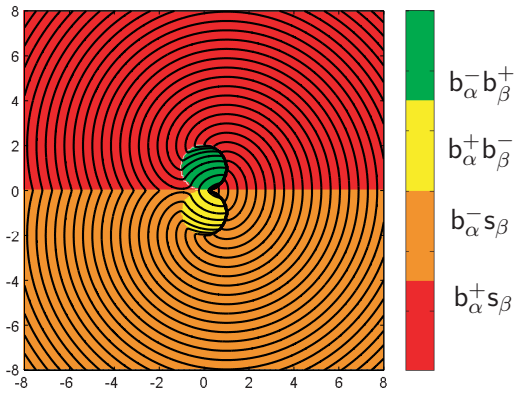


(e) $\varrho/\rho = 1.8$

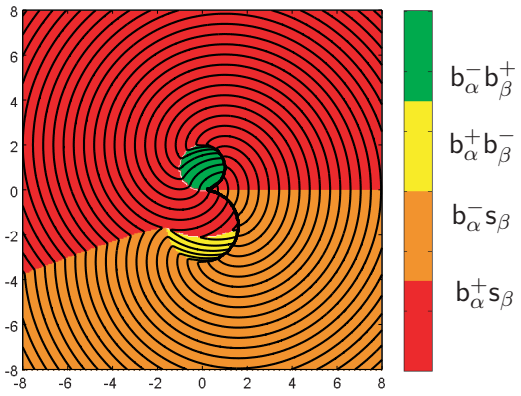


(f) $\varrho/\rho = 2.0$

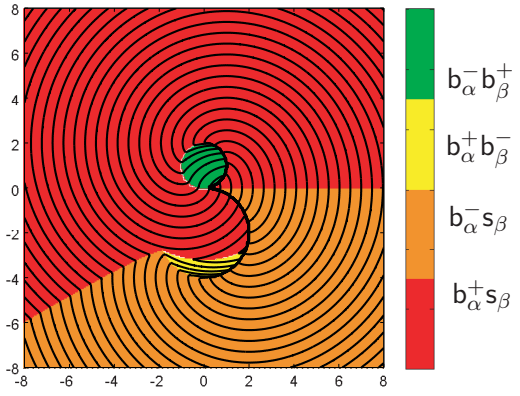
Figure 5.7: Partition of $P_{\pi/3}$ and level sets of $T_{\mathrm{f}} = T_{\mathrm{f}}(x, y)$ for different values of the ratio $\delta^{-1} = \varrho/\rho$.



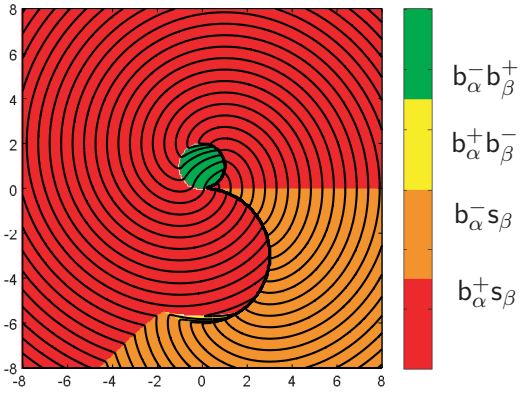
(a) $\varrho/\rho = 1.0$



(b) $\varrho/\rho = 1.6$

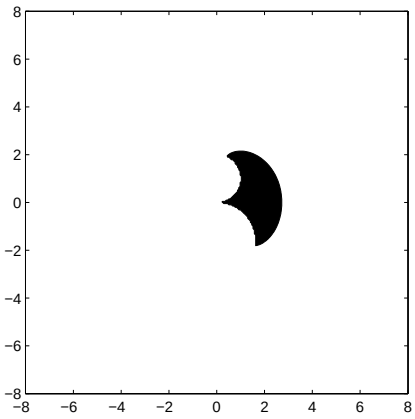


(c) $\varrho/\rho = 2.0$

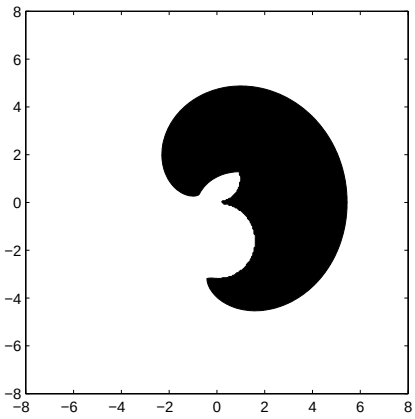


(d) $\varrho/\rho = 3.0$

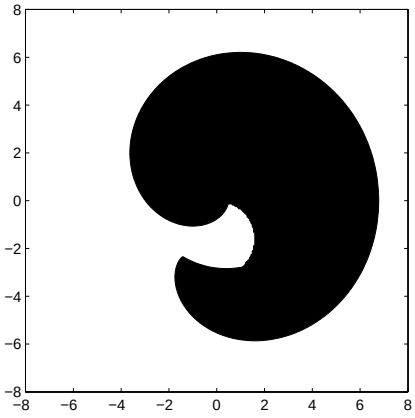
Figure 5.8: Partition of P and level sets of $T_{\mathrm{f}} = T_{\mathrm{f}}(x, y)$ for different values of the ratio ϱ/ρ.



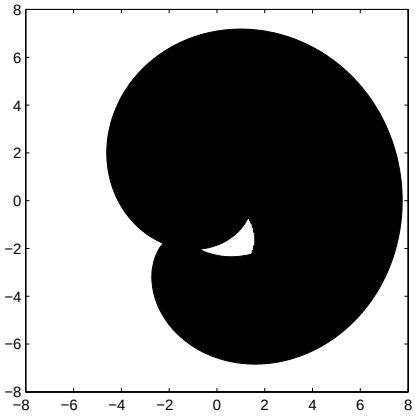
(a) $\mathfrak{R}^{\mathrm{asym}}_{t\leq\tau}$, $\tau = (\rho+\varrho)\pi/3$.



(b) $\mathfrak{R}^{\mathrm{asym}}_{t\leq\tau}$, $\tau = (\rho+\varrho)2\pi/3$.



(c) $\mathfrak{R}^{\mathrm{asym}}_{t\leq\tau}$, $\tau = (\rho+\varrho)(3\pi+1)/4$.



(d) $\mathfrak{R}^{\mathrm{asym}}_{t\leq\tau}$, $\tau = (\rho+\varrho)(3\pi+2.5)/4$.

Figure 5.9: Reachable sets $\mathfrak{R}^{\mathrm{asym}}_{t\leq\tau}$ for different values of τ and for $\varrho/\rho = 1.6$.

Chapter 6

A Mesh Refinement Method Using Density Functions for Solving Numerical Optimal Control Problems

Given a good initial guess for a path or trajectory it will be often necessary to generate optimal trajectories using more realistic aircraft models. This necessitates the use of numerical techniques for the solution of the optimal trajectory. Although trajectory generation has a long and well-established history, the current numerical schemes are plagued by sensitivity to initial conditions, convergence, etc. Some of these problems can be dealt with using multi-resolution grid for the discretization of the continuous optimal control problem. Such discretizations can enhance the numerical stability and convergence properties of the solution algorithm.

This chapter introduces a simple, yet efficient, mesh generation method for solving optimal control problems. The proposed DENsity function-based Mesh Refinement Algorithm (DENMRA) 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 [15, 24, 90]. 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.

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 scheme proposed in Ref. [152]. Numerical results show 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.

6.1 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. [152] demonstrating the advantage of such algorithms. Reference [33] introduced a mesh refinement method in which integer programming is used to minimize the maximum integration error during mesh refinement iterations. Reference [99] 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. [81]. 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 [15, 90]. The concept of density functions is similar to monitor functions used for the numerical solution of PDEs [24]. 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. [31]. 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 [31]. 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).

6.2 Problem Statement and Nonlinear Programming Formulation

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

$$\mathcal{J} = \Phi(x(t_0), t_0, x(t_f), \mathbf{p}, t_f) + \int_{t_0}^{t_f} \mathcal{L}(x(t), u(t), \mathbf{p}, t)\mathrm{d}t, \tag{6.1}$$

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 (6.1) subject to the dynamic constraints

$$\dot{x}(t) = f\big(x(t), u(t), \mathbf{p}, t\big), \quad t_0 \le t \le t_f, \tag{6.2}$$

the boundary conditions

$$\Psi\big(x(t_0), t_0, x(t_f), t_f, \mathbf{p}\big) = 0, \tag{6.3}$$

and the path constraints

$$C\big(x(t), u(t), \mathbf{p}, t\big) \le 0, \quad t_0 \le t \le t_f, \tag{6.4}$$

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 $0 \le i \le N-1$. Let $\mathbf{X}$ be the vector of all decision variables, the corresponding discretization of the continuous time optimal control problem (6.1)–(6.4) can be written as

$$\min_{\mathbf{X}} J(\mathbf{X}) \tag{6.5}$$

subject to

$$|F(\mathbf{X})| \le \zeta_d, \tag{6.6}$$

$$|\tilde{\Psi}(\mathbf{X})| \le \zeta_b, \tag{6.7}$$

and

$$\tilde{C}\big(\mathbf{X}\big) \le \zeta_C, \tag{6.8}$$

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 (6.2), the function J in (6.5) and F in (6.6) 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[152]. For more details the reader may refer to Refs. [152, 189, 33, 98].

6.2.1 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

$$\bar{f}(t) = \frac{f(t)}{\int_a^b f(\tau)\, \mathrm{d}\tau}, \tag{6.9}$$

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

$$F(t) \triangleq \int_a^t \bar{f}(\tau)\, \mathrm{d}\tau. \tag{6.10}$$

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 (6.10). 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

$$F(t_{i+1}) - F(t_i) = \frac{1}{N}. \tag{6.11}$$

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 [31]. 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

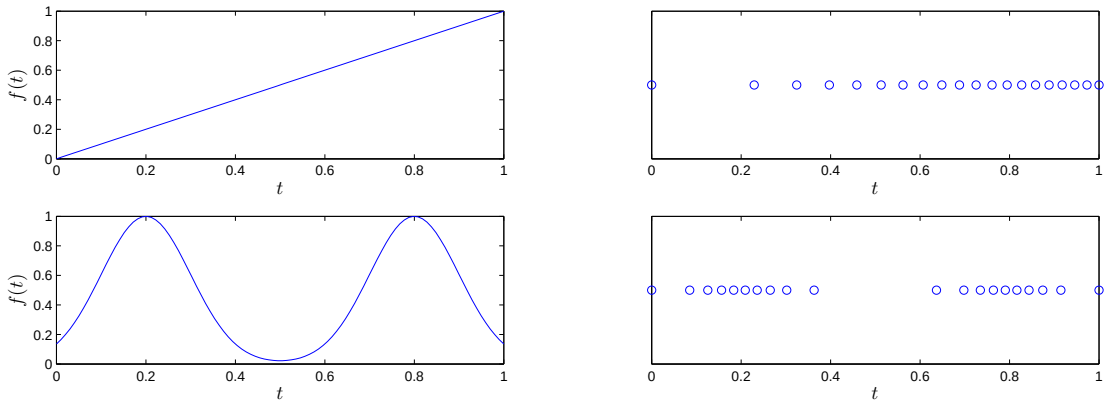


Figure 6.1: Density functions and corresponding distribution of grid points.

with high accuracy. Note that the mesh point distribution is unique once the density function is given, but the converse is not true.

Figure 6.1 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.

6.2.2 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. [152], 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. [31], 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. [190].

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. [33] 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.

6.3 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

$$\int_{t_a}^{t_i} \rho_\Gamma\,(\tau)\, \mathrm{d}\tau = \frac{i-1}{N-1}. \tag{6.12}$$

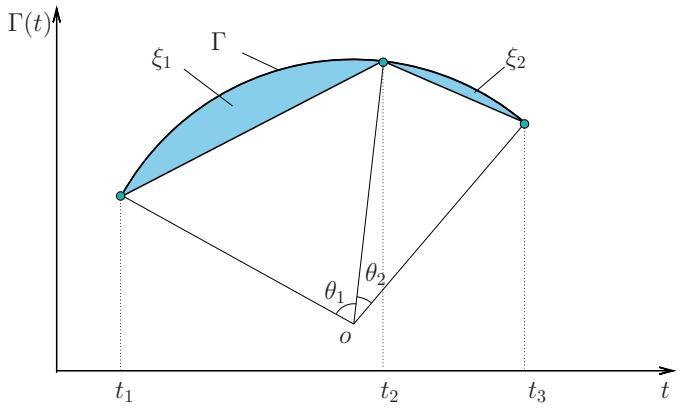


Figure 6.2: Approximation error in terms of the L^1-norm, for a curve Γ of constant curvature.

Proposition 6.1. 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. 6.2, 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. 6.2. The shaded area ξ_1 is given by

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

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 6.1. 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. 6.2), 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 6.1, 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 6.1. 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

$$\frac{\rho_{\kappa_1}}{\rho_{\kappa_2}} = \left(\frac{\kappa_1}{\kappa_2}\right)^{\frac{1}{3}}. \tag{6.13}$$

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:

$$\xi_1(N_1) = \frac{1}{2}\kappa_1^{-2} \sum_{1}^{N_1-1} \left(\frac{\theta_1}{(N_1-1)} - \sin\frac{\theta_1}{N_1-1}\right) = \frac{1}{2}\kappa_1^{-2}\left(\theta_1 - (N_1-1)\sin\frac{\theta_1}{N_1-1}\right). \tag{6.14}$$

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:

$$\xi_2(N_2) = \frac{1}{2}\kappa_2^{-2}\left(\theta_2 - (N_2-1)\sin\frac{\theta_2}{N_2-1}\right). \tag{6.15}$$

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

$$\xi(x,y) = \xi_1(x) + \xi_2(y) = \frac{1}{2}\kappa_1^{-2}\left(\theta_1 - x\sin\frac{\theta_1}{x}\right) + \frac{1}{2}\kappa_2^{-2}\left(\theta_2 - y\sin\frac{\theta_2}{y}\right) \tag{6.16}$$

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

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

$$\kappa_2^{-2}\sin\left(\frac{\theta_2}{y}\right) - \kappa_2^{-2}\frac{\theta_2}{y}\cos\left(\frac{\theta_2}{y}\right) - \kappa_1^{-2}\sin\left(\frac{\theta_1}{x}\right) + \kappa_1^{-2}\frac{\theta_1}{x}\cos\left(\frac{\theta_1}{x}\right) = 0. \tag{6.17}$$

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 (6.17) can be rewritten in terms of infinite series as

$$\kappa_2^{-2} \sum_{n=0}^{\infty} (-1)^{n+1} \frac{2n}{(2n+1)!} \left(\frac{\theta_2}{y}\right)^{2n+1} - \kappa_1^{-2} \sum_{n=0}^{\infty} (-1)^{n+1} \frac{2n}{(2n+1)!} \left(\frac{\theta_1}{x}\right)^{2n+1} = 0. \tag{6.18}$$

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 (6.18) vanishes, and one obtains

$$\kappa_2^{-2} \left(\frac{\theta_2}{y}\right)^3 - \kappa_1^{-2} \left(\frac{\theta_1}{x}\right)^3 = 0. \tag{6.19}$$

The solution of (6.19) yields,

$$\frac{x}{y} = \frac{S_1}{S_2} \left(\frac{\kappa_1}{\kappa_2}\right)^{1/3}, \tag{6.20}$$

from which we have

$$\frac{\rho_{\kappa_1}}{\rho_{\kappa_2}} = \lim_{N_1, N_2 \to \infty} \frac{N_1/S_1}{N_2/S_2} = \lim_{x,y \to \infty} \frac{x/S_1}{y/S_2} = \left(\frac{\kappa_1}{\kappa_2}\right)^{1/3}. \tag{6.21}$$

The solution to (6.19) is indeed the optimal solution since the Hessian of $\xi(x, y)$ for $x, y \to \infty$, given by,

$$H(\xi) = \begin{bmatrix} \frac{\theta_1^2}{\kappa_1^2 x^3} \sin\left(\frac{\theta_1}{x}\right) & 0 \\ 0 & \frac{\theta_2^2}{\kappa_2^2 y^3} \sin\left(\frac{\theta_2}{y}\right) \end{bmatrix} \tag{6.22}$$

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

Although Theorem (6.1) 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 6.1. 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 6.1 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 6.1. 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. [87, 126, 148] 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 [147]. 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. [147] 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. [147] to the estimation of the L^1 norm of the approximation error. Because the arc spline considered here is different from that in Ref. [147], 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. [147] and provide an estimation of the L^1 norm of the approximation error.

Theorem 6.2 (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

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

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

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

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

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

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

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

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. 6.3.

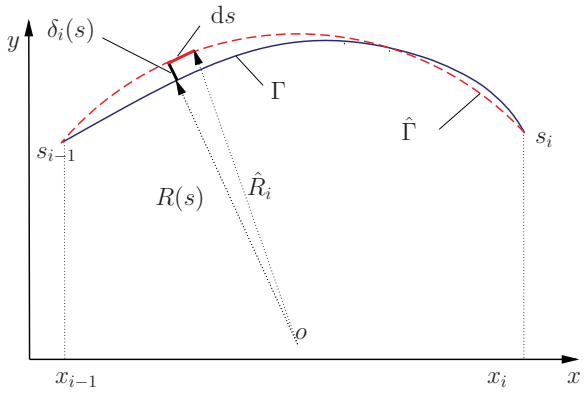


Figure 6.3: L^1-norm of the approximation error on $[s_{i-1}, s_i]$ for a C^3-smooth curve.

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

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

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

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,

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

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

The following result from Ref. [117] extends Corollary 6.1 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 6.3 (The best choice of nodes for approximation using splines in L^p space[117])**.** 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

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

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

$$\int_0^{x_{i,n}^*} \left[\Gamma^{(r)}(t)\right]^{p/(rp+1)} \mathrm{d}t = \frac{i}{n} \int_0^1 \left[\Gamma^{(r)}(t)\right]^{p/(rp+1)} \mathrm{d}t. \tag{6.23}$$

Although it is assumed for Theorem 6.3 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 (6.23) 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 (6.23). 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

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

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.

6.4 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. [154], which was implemented in DENMRA.

Consider the following optimal control problem stated in Mayer form.

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

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 $[t_0, t_f]$.

6.4.1 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

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

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

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

6.4.2 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

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

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

$$\lambda\left(t_i^-\right)^T = \frac{\lambda_{i+1}^T}{2} \left.\frac{\partial f}{\partial x}\right|_{(\bar{x}_{i+1}, u_{i+1}, \bar{t}_{i+1})} + \frac{\lambda_{i+1}^T}{t_{i+1} - t_i} + \frac{\sigma_{i+1}^T}{2} \left.\frac{\partial g}{\partial x}\right|_{(\bar{x}_{i+1}, u_{i+1}, \bar{t}_{i+1})} + \frac{\mu_i^T}{2} \left.\frac{\partial h}{\partial x}\right|_{(\bar{x}_i, \bar{t}_i)}, \quad (6.24)$$

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 (6.24) 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:

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

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 $l_j = \sum_{k=i}^{i_b} \dfrac{\mu_k}{j!}\left(t_k - t_i\right)^j$, $j = 0, \ldots, q-1$.

6.4.3 Numerical Example

Consider the Brachistochrone problem with state constraint as in Ref. [154]. 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

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

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. 6.4 and Fig. 6.5. The Hamiltonian is shown in Fig. 6.6. 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 [154], 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.

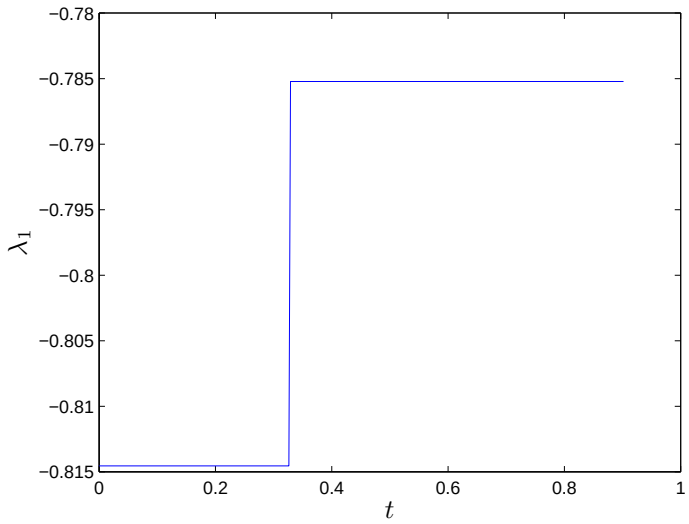


Figure 6.4: Costate history: λ_x.

6.5 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.

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

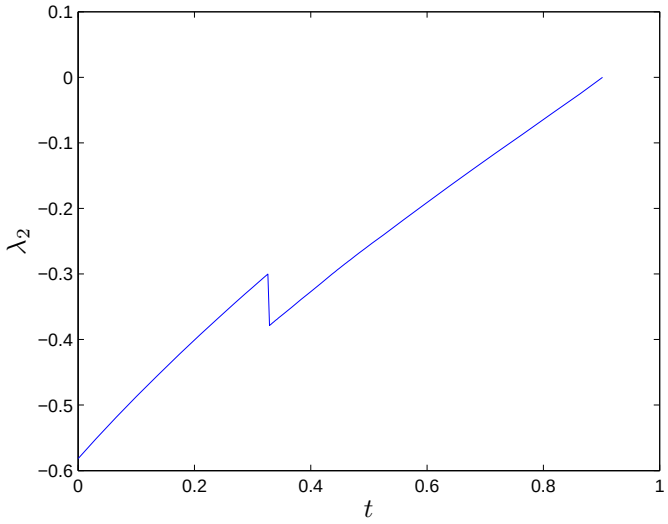


Figure 6.5: Costate history: λ_y.

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.

6.5.1 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. [33]. 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.

6.5.2 Technical Details

The details of these steps are given below.

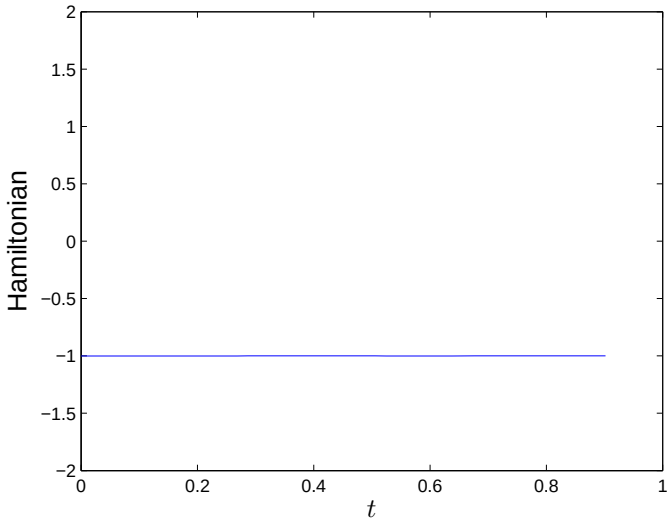


Figure 6.6: Hamiltonian history.

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 [80] 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 [190] 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 $\rho_{i,k} = (\kappa_{i,k}^{1/3} + ilon)\sqrt{1 + \dot{u}_{i,k}^2} = |\ddot{u}_{i,k}|^{1/3} + ilon\sqrt{1 + \dot{u}_{i,k}^2}$, where

$ilon > 0$. 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 $ilon$ 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,

$$f(t) = \left(\sum_{k=1}^{m} \rho_k^2(t) \right)^{\frac{1}{2}}, \tag{6.25}$$

and

$$f(t) = \max_k \rho_k(t) \tag{6.26}$$

are two possible methods to generate the overall density function.

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.

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.

6.6 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 [37]. 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 [144] 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 [30], 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.

6.6.1 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 $0 \le t \le 1$, to minimize

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

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

The solution of the optimal control $u^*(t)$ can be obtained as follows [37]:

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

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

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

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 6.3, 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 [33]. Both algorithms were tested on the same computer, and cold-started using the same linear initial guess.

Table 6.1 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

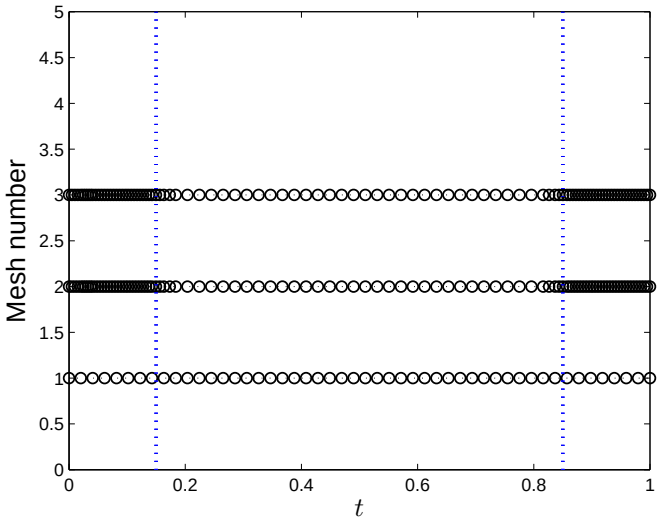


Figure 6.7: Mesh refinement, SOCS, $\ell = 0.05$.

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 $10^{-5} \sim 10^{-6}$.

The mesh refinement histories of the two algorithms for the case with $\ell = 0.05$ are shown in Figs. 6.7-6.8. 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. 6.8, 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 k 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 6.1: 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

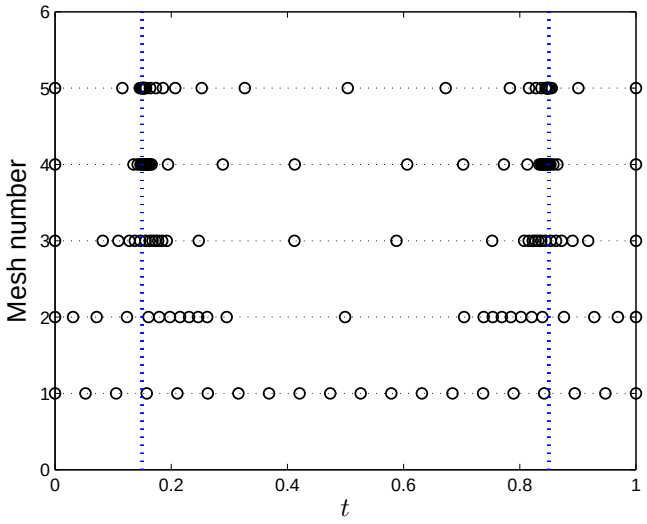


Figure 6.8: Mesh refinement, DENMRA, $\ell = 0.05$.

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 6.2: 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 6.2. 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 6.2, DENMRA provides sharper resolution than SOCS while preserving the accuracy of the solution.

6.6.2 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 [144], 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. [144], 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 6.9 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 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 $10^N \le t_f < 10^{N+1}$, for some positive integer N, and the problem was solved again with the new

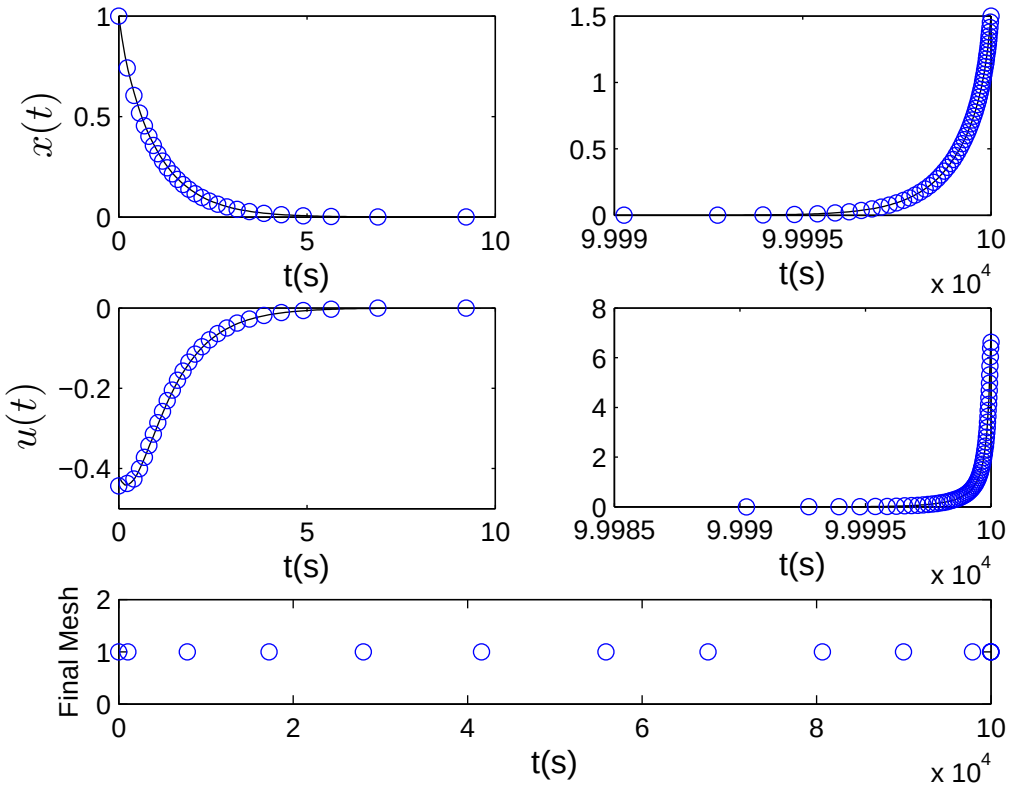


Figure 6.9: DENMRA solution, $t_f = 100,000$.

t_f. This process was repeated until $J \geq 7$. The results are shown in Table 6.3. 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 6.4. 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 6.3: 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

In Ref. [31], a density (monitor) function of the form

Table 6.4: 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

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

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.

6.6.3 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 [73]:

$$\dot{x} = v\cos\gamma\cos\psi, \tag{6.27}$$
$$\dot{y} = v\cos\gamma\sin\psi, \tag{6.28}$$
$$\dot{z} = v\sin\gamma, \tag{6.29}$$
$$\dot{v} = \frac{1}{m}\left(T\cos\alpha - D(\alpha,v,z)\right) - g\sin\gamma, \tag{6.30}$$
$$\dot{\gamma} = \frac{1}{mv}(T\sin\alpha + L(\alpha,v,z)\cos\phi) - \frac{g}{v}\cos\gamma, \tag{6.31}$$
$$\dot{\psi} = -\frac{1}{mv\cos\gamma}L(\alpha,v,z)\sin\phi, \tag{6.32}$$

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:

$$D(\alpha,v,z) = Q(v,z)SC_D(\alpha),$$
$$L(\alpha,v,z) = Q(v,z)SC_L(\alpha),$$

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, $K = 1/(0.95e\pi\mathcal{R})$, where e is the efficiency factor, which is corrected by 0.95 for the assumed landing configuration. $\mathcal{R}$ is the aspect ratio of the aircraft defined by $\mathcal{R} = b^2/S$, 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 [130]. The values of the parameters in the former equations are given in Table 6.6.3, where $T_{\max}$ is the maximum thrust.

Table 6.5: 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 $z = 10\,\mathrm{km}$, cruise speed $v = 240\,\mathrm{m/s}$, 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 6.6.

The zero-thrust emergency landing scenarios in Table 6.6 were solved by minimizing the final arrival time t_f in DENMRA. The corresponding landing trajectories are shown in Fig. 6.10 and Fig. 6.11.

Table 6.6: Runway layout table.

	x_f(km)	y_f(km)	z_f(km)	$\psi_f(^\circ)$
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 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.

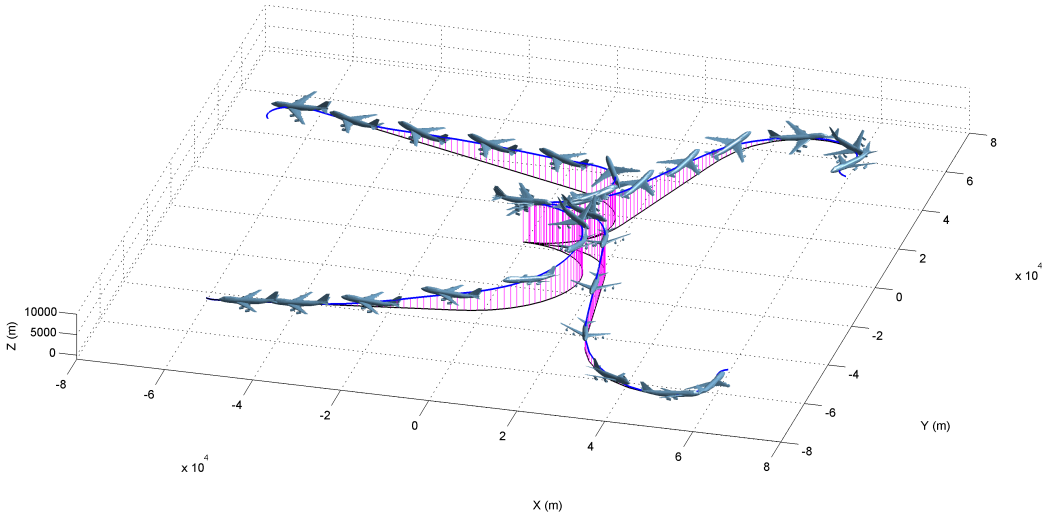


Figure 6.10: 3D glider landing trajectories generated in DENMRA.

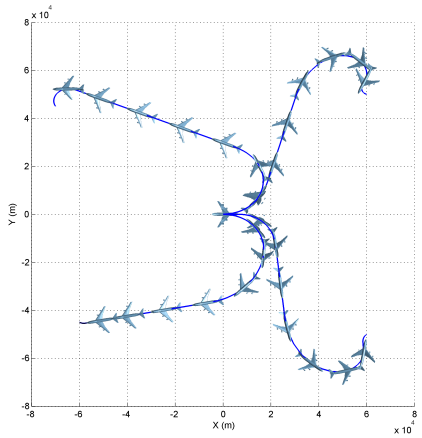


Figure 6.11: 2D projection of glider landing trajectories generated in DENMRA.

Chapter 7

Path Smoothing Using Iterative Quadratic Programming

Path smoothness is a desirable property for the precise tracking of such a path by the aircraft. 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 1.3 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.

7.1 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 [66]. 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 [60]. 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 [150, 74, 20]. Reference [140] used analytical splines and heuristics for smooth path generation. Reference [186] 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, [19] introduced a method for generating curvature-bounded paths in rectangular channels; reference [29] proposed a method for constructing bounded curvature paths traversing a constant width region in the plane, called corridors, and reference [103] 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 [121].

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 [79] 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.

7.2 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 $i-1^{\text{th}}$, i^{th}, $i+1^{\text{th}}$, and $i+2^{\text{th}}$ nodes) is used to construct a local cubic curve for the interpolation between the i^{th} and the $i+1^{\text{th}}$ node, as shown in Fig. 7.1. A local path length coordinate τ is assigned to each group of nodes such that $\tau = 0$ for the $i-1^{\text{th}}$ node, $\tau = 1/3$ for the i^{th} node, $\tau = 2/3$ for the $i+1^{\text{th}}$ node, and $\tau = 1$ for the $i+2^{\text{th}}$ node.

With a slight abuse of notation, the coordinate x of the path between $\tau = 1/3$ and $\tau = 2/3$ is given

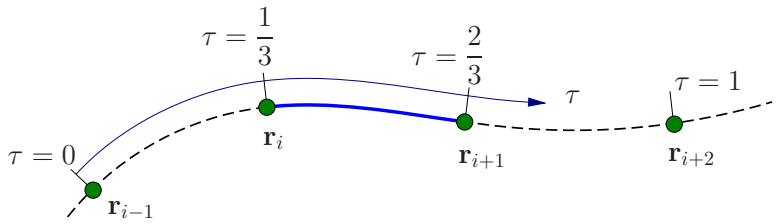


Figure 7.1: Cubic spline interpolation.

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:

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

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

$$x_{i-1} = x(0; i) = d_{x_i}, \tag{7.1}$$

$$x_i = x(\tfrac{1}{3}; i) = \frac{a_{x_i}}{27} + \frac{b_{x_i}}{9} + \frac{c_{x_i}}{3} + d_{x_i}, \tag{7.2}$$

$$x_{i+1} = x(\tfrac{2}{3}; i) = \frac{8a_{x_i}}{27} + \frac{4b_{x_i}}{9} + \frac{2c_{x_i}}{3} + d_{x_i}, \tag{7.3}$$

$$x_{i+2} = x(1; i) = a_{x_i} + b_{x_i} + c_{x_i} + d_{x_i}, \tag{7.4}$$

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

$$x(\tau; i) = \begin{bmatrix} \tau^3 & \tau^2 & \tau & 1 \end{bmatrix} \begin{bmatrix} a_{x_i} \\ b_{x_i} \\ c_{x_i} \\ d_{x_i} \end{bmatrix} = \begin{bmatrix} \tau^3 & \tau^2 & \tau & 1 \end{bmatrix} \mathbf{G} \begin{bmatrix} x_{i-1} \\ x_i \\ x_{i+1} \\ x_{i+2} \end{bmatrix}. \tag{7.5}$$

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

$$\mathbf{r}(\tau; i) = \begin{bmatrix} \tau^3 & \tau^2 & \tau & 1 \end{bmatrix} \mathbf{G} \begin{bmatrix} \mathbf{r}_{i-1} \\ \mathbf{r}_i \\ \mathbf{r}_{i+1} \\ \mathbf{r}_{i+2} \end{bmatrix}. \tag{7.6}$$

By taking derivatives of (7.6), we have

$$\frac{\mathrm{d}\mathbf{r}(\tau;i)}{\mathrm{d}\tau} = \begin{bmatrix} 3\tau^2 & 2\tau & 1 & 0 \end{bmatrix} \mathbf{G} \begin{bmatrix} \mathbf{r}_{i-1} \\ \mathbf{r}_i \\ \mathbf{r}_{i+1} \\ \mathbf{r}_{i+2} \end{bmatrix}, \tag{7.7}$$

and

$$\frac{\mathrm{d}^2\mathbf{r}(\tau;i)}{\mathrm{d}\tau^2} = \begin{bmatrix} 6\tau & 2 & 0 & 0 \end{bmatrix} \mathbf{G} \begin{bmatrix} \mathbf{r}_{i-1} \\ \mathbf{r}_i \\ \mathbf{r}_{i+1} \\ \mathbf{r}_{i+2} \end{bmatrix}. \tag{7.8}$$

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 '$\times$' denotes the cross product.

We also define

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

7.3 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

$$\tilde{\mathbf{r}}_i = \mathbf{r}_i + \mathbf{n}_i\delta_i + \mathbf{b}_i\lambda_i = \mathbf{r}_i + (\mathbf{n}_i\{1\}, \mathbf{n}_i\{2\}, \mathbf{n}_i\{3\})\,\delta_i + (\mathbf{b}_i\{1\}, \mathbf{b}_i\{2\}, \mathbf{b}_i\{3\})\,\lambda_i. \tag{7.9}$$

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 $\tilde{\mathbf{r}}_i$, $i = 1, \ldots, N$.

7.4 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 7.1. 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,

$$J(x) = \frac{1}{2}x^{\mathrm{T}}Hx + F^{\mathrm{T}}x + c, \tag{7.10}$$

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.

7.4.1 A Quadratic Cost Function

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

$$\|\tilde{\kappa}\|_2 \triangleq \left(\int_{s_0}^{s_f} w(s)\tilde{\kappa}^2(s)\mathrm{ds} \right)^{\frac{1}{2}}, \tag{7.11}$$

where $w : [s_0, s_f] \to \mathbb{R}_+ \setminus \{0\}$ is a weight function. With the cubic spline curve representation of the path, the integral in (7.11) 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 (7.11), 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,

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

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

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

and

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

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 (7.9), 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 (7.11) 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 $\mathbf{H}$ and $\mathbf{F}$ are given in [188].

7.4.2 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,

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

Then the segment length D_i can be written as in (7.12).

$$\begin{aligned} D_i &= \|\tilde{\mathbf{r}}_{i+1} - \tilde{\mathbf{r}}_i\| - \|\mathbf{r}_{\mathbf{i+1}} - \mathbf{r}_{\mathbf{i}}\| \\ &= -\|\mathbf{r}_{i+1} - \mathbf{r}_i\| + \sqrt{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|^2 + \|\mu_{i+1} \mathbf{p}_{i+1} - \mu_i \mathbf{p}_i\|^2 + 2 \langle \delta_{i+1} \mathbf{p}_{i+1} - \mu_i \mathbf{p}_i, \mathbf{r}_{i+1} - \mathbf{r}_i \rangle} \\ &= \frac{1}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|} \frac{\|\mu_{i+1} \mathbf{p}_{i+1} - \mu_i \mathbf{p}_i\|^2 + 2 \langle \mu_{i+1} \mathbf{p}_{i+1} - \mu_i \mathbf{p}_i, \mathbf{r}_{i+1} - \mathbf{r}_i \rangle}{1 + \sqrt{1 + \dfrac{\|\mu_{i+1} \mathbf{p}_{i+1} - \mu_i \mathbf{p}_i\|^2}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|^2} + 2 \left\langle \dfrac{\mu_{i+1} \mathbf{p}_{i+1} - \mu_i \mathbf{p}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|}, \dfrac{\mathbf{r}_{i+1} - \mathbf{r}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|} \right\rangle}}. \end{aligned} \tag{7.12}$$

By the small variation assumption, and dropping the square terms, expression (7.12) yields the

following approximation for D_i

$$\begin{aligned} D_i &\approx \left\langle \frac{\mathbf{r}_{i+1}-\mathbf{r}_i}{\|\mathbf{r}_{i+1}-\mathbf{r}_i\|}, \mu_{i+1}\mathbf{p}_{i+1} \right\rangle - \left\langle \frac{\mathbf{r}_{i+1}-\mathbf{r}_i}{\|\mathbf{r}_{i+1}-\mathbf{r}_i\|}, \mu_i \mathbf{p}_i \right\rangle \\ &= \left\langle \frac{\mathbf{r}_{i+1}-\mathbf{r}_i}{\|\mathbf{r}_{i+1}-\mathbf{r}_i\|}, \delta_{i+1}\mathbf{n}_{i+1} \right\rangle - \left\langle \frac{\mathbf{r}_{i+1}-\mathbf{r}_i}{\|\mathbf{r}_{i+1}-\mathbf{r}_i\|}, \delta_i \mathbf{n}_i \right\rangle \\ &+ \left\langle \frac{\mathbf{r}_{i+1}-\mathbf{r}_i}{\|\mathbf{r}_{i+1}-\mathbf{r}_i\|}, \lambda_{i+1}\mathbf{b}_{i+1} \right\rangle - \left\langle \frac{\mathbf{r}_{i+1}-\mathbf{r}_i}{\|\mathbf{r}_{i+1}-\mathbf{r}_i\|}, \lambda_i \mathbf{b}_i \right\rangle \end{aligned} \tag{7.13}$$

$$\mathbf{E}_\delta = \begin{bmatrix} -\langle \mathbf{r}_2-\mathbf{r}_1, \mathbf{n}_1\rangle & \langle \mathbf{r}_2-\mathbf{r}_1, \mathbf{n}_2\rangle & & & 0 \\ & -\langle \mathbf{r}_3-\mathbf{r}_2, \mathbf{n}_2\rangle & \langle \mathbf{r}_3-\mathbf{r}_2, \mathbf{n}_3\rangle & & \\ & & \ddots & \ddots & \\ 0 & & & -\langle \mathbf{r}_N-\mathbf{r}_{N-1}, \mathbf{n}_{N-1}\rangle & \langle \mathbf{r}_N-\mathbf{r}_{N-1}, \mathbf{n}_N\rangle \end{bmatrix} \tag{7.14}$$

$$\mathbf{E}_\lambda = \begin{bmatrix} -\langle \mathbf{r}_2-\mathbf{r}_1, \mathbf{b}_1\rangle & \langle \mathbf{r}_2-\mathbf{r}_1, \mathbf{b}_2\rangle & & & 0 \\ & -\langle \mathbf{r}_3-\mathbf{r}_2, \mathbf{b}_2\rangle & \langle \mathbf{r}_3-\mathbf{r}_2, \mathbf{b}_3\rangle & & \\ & & \ddots & \ddots & \\ 0 & & & -\langle \mathbf{r}_N-\mathbf{r}_{N-1}, \mathbf{b}_{N-1}\rangle & \langle \mathbf{r}_N-\mathbf{r}_{N-1}, \mathbf{b}_N\rangle \end{bmatrix} \tag{7.15}$$

In order to write equation (7.13) 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 (7.14) and (7.15). 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:

$$L_{\min} - L \le \Delta_L(X) \le L_{\max} - L, \tag{7.16}$$

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 element-wise. Alternatively, if the length of the path is fixed, then the linear equality constraint $\Delta_L(X) = 0$ is applied ($L_{\min} = L = L_{\max}$):

7.4.3 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 (7.8), 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 the matrix $\mathbf{C}$ are given in [188]. 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

$$
K_{\min} - K \leq \mathbf{C}X \leq K_{\max} - K. \tag{7.17}
$$

7.4.4 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 (7.17), 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 (7.6).

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.

7.4.5 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.,

$$\begin{aligned}
&-5.5\langle\mathbf{r}_1, \mathbf{n}_1\rangle + 9\langle\mathbf{r}_2 + \mathbf{n}_2\delta_2, \mathbf{n}_1\rangle - 4.5\langle\mathbf{r}_3 + \mathbf{n}_3\delta_3, \mathbf{n}_1\rangle + \langle\mathbf{r}_4 + \mathbf{n}_4\delta_4, \mathbf{n}_1\rangle \\
&\qquad + 9\langle\mathbf{r}_2 + \mathbf{b}_2\lambda_2, \mathbf{n}_1\rangle - 4.5\langle\mathbf{r}_3 + \mathbf{b}_3\lambda_3, \mathbf{n}_1\rangle + \langle\mathbf{r}_4 + \mathbf{b}_4\lambda_4, \mathbf{n}_1\rangle = 0, \qquad (7.18) \\
&-5.5\langle\mathbf{r}_1, \mathbf{b}_1\rangle + 9\langle\mathbf{r}_2 + \mathbf{n}_2\delta_2, \mathbf{b}_1\rangle - 4.5\langle\mathbf{r}_3 + \mathbf{n}_3\delta_3, \mathbf{b}_1\rangle + \langle\mathbf{r}_4 + \mathbf{n}_4\delta_4, \mathbf{b}_1\rangle \\
&\qquad + 9\langle\mathbf{r}_2 + \mathbf{b}_2\lambda_2, \mathbf{b}_1\rangle - 4.5\langle\mathbf{r}_3 + \mathbf{b}_3\lambda_3, \mathbf{b}_1\rangle + \langle\mathbf{r}_4 + \mathbf{b}_4\lambda_4, \mathbf{b}_1\rangle = 0, \qquad (7.19)
\end{aligned}$$

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

$$\begin{aligned}
&5.5\langle\mathbf{r}_N, \mathbf{n}_N\rangle - 9\langle\mathbf{r}_{N-1} + \mathbf{n}_{N-1}\delta_{N-1}, \mathbf{n}_N\rangle + 4.5\langle\mathbf{r}_{N-2} + \mathbf{n}_{N-2}\delta_{N-2}, \mathbf{n}_N\rangle \\
&\qquad - \langle\mathbf{r}_{N-3} + \mathbf{n}_{N-3}\delta_{N-3}, \mathbf{n}_N\rangle - 9\langle\mathbf{r}_{N-1} + \mathbf{b}_{N-1}\lambda_{N-1}, \mathbf{n}_N\rangle \qquad (7.20) \\
&\qquad + 4.5\langle\mathbf{r}_{N-2} + \mathbf{b}_{N-2}\lambda_{N-2}, \mathbf{n}_N\rangle - \langle\mathbf{r}_{N-3} + \mathbf{b}_{N-3}\lambda_{N-3}, \mathbf{n}_N\rangle = 0,
\end{aligned}$$

$$\begin{aligned}
&5.5\langle\mathbf{r}_N, \mathbf{b}_N\rangle - 9\langle\mathbf{r}_{N-1} + \mathbf{n}_{N-1}\delta_{N-1}, \mathbf{b}_N\rangle + 4.5\langle\mathbf{r}_{N-2} + \mathbf{n}_{N-2}\delta_{N-2}, \mathbf{b}_N\rangle \\
&\qquad - \langle\mathbf{r}_{N-3} + \mathbf{n}_{N-3}\delta_{N-3}, \mathbf{b}_N\rangle - 9\langle\mathbf{r}_{N-1} + \mathbf{b}_{N-1}\lambda_{N-1}, \mathbf{b}_N\rangle \qquad (7.21) \\
&\qquad + 4.5\langle\mathbf{r}_{N-2} + \mathbf{b}_{N-2}\lambda_{N-2}, \mathbf{b}_N\rangle - \langle\mathbf{r}_{N-3} + \mathbf{b}_{N-3}\lambda_{N-3}, \mathbf{b}_N\rangle = 0,
\end{aligned}$$

7.4.6 Connection to Beam Theory

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

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

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:

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

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 (7.11) corresponds to the flexural rigidity $EI(s)$.

7.5 Path Smoothing Algorithm

7.5.1 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

$$\begin{aligned} \mathcal{P}(s_i, j+1) &= \mathcal{P}(s_i, j) + X^*_{n_i} \mathbf{n}(\mathbf{s}_i, j) + X^*_{b_i} \mathbf{b}(\mathbf{s}_i, j), \\ \mathcal{P}(s, 0) &= \mathcal{P}^{(0)}(s), \end{aligned} \tag{7.22}$$

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

The proposed path smoothing algorithm is designed based on the evolution equation (7.22), 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

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

   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 [153]:

$$\begin{aligned} \frac{\partial \mathcal{P}(s,t)}{\partial t} &= \kappa(s,t)\mathbf{n}(s,t), \\ \mathcal{P}(s,0) &= \mathcal{P}^{(0)}(s). \end{aligned} \tag{7.23}$$

Equation (7.23) can be solved numerically using forward difference

$$\mathcal{P}(s,t+\varepsilon) = \mathcal{P}(s,t) + \varepsilon\kappa(s)\mathbf{n}(s,t), \tag{7.24}$$

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 (7.24). 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. 7.2 and Fig. 7.3.

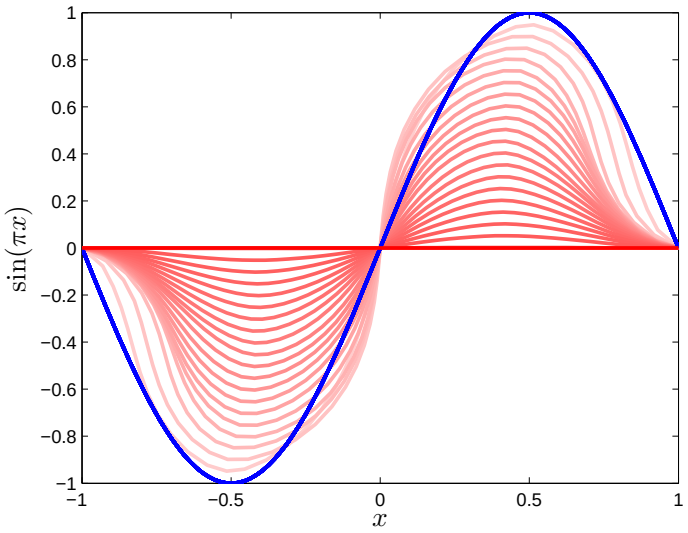


Figure 7.2: Quadratic programming path smoothing.

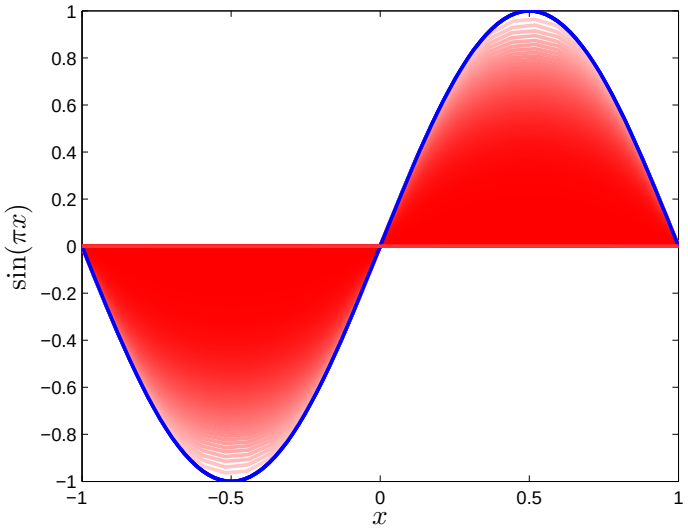


Figure 7.3: Curvature evolution.

7.5.2 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.

7.6 Numerical Examples

7.6.1 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. 7.4. 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. 7.4. 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. 7.5. 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. 7.6, 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 [191] with free final speed at point B.

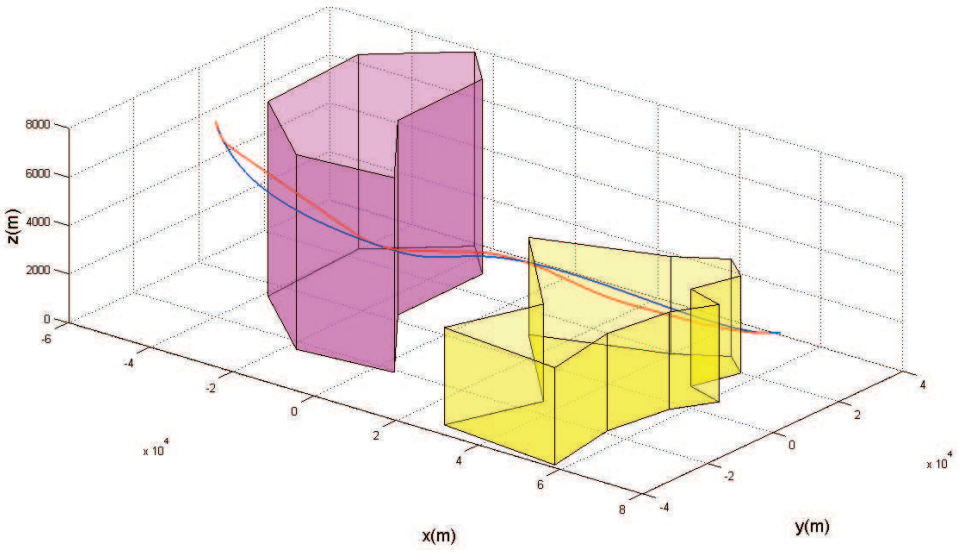


Figure 7.4: Path smoothing in the presence of obstacles.

7.6.2 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. 7.7. It is clear that the ground vehicle does not need to perform any turning maneuver on the ice surface.

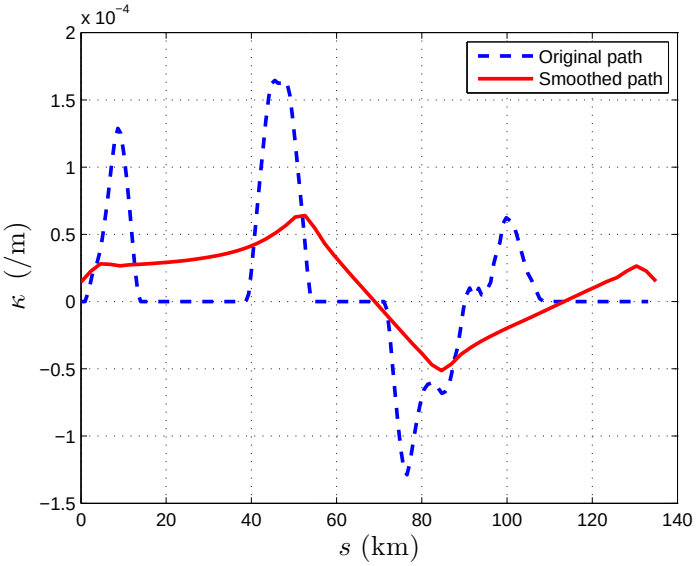


Figure 7.5: Curvature profile comparison.

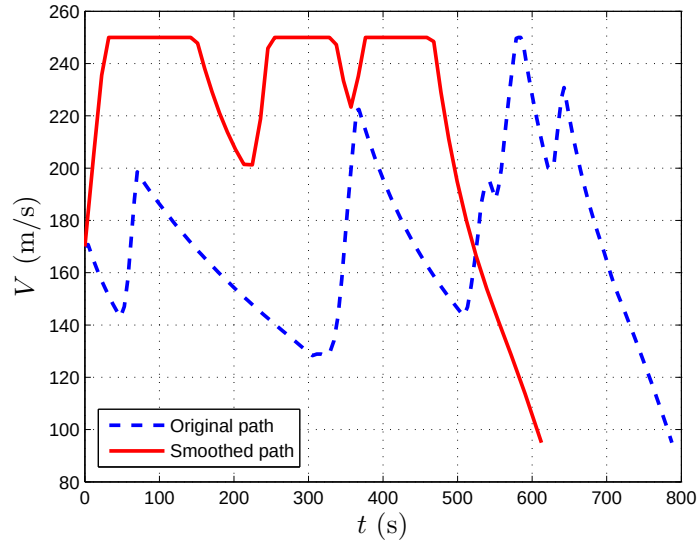


Figure 7.6: Optimal speed profile.

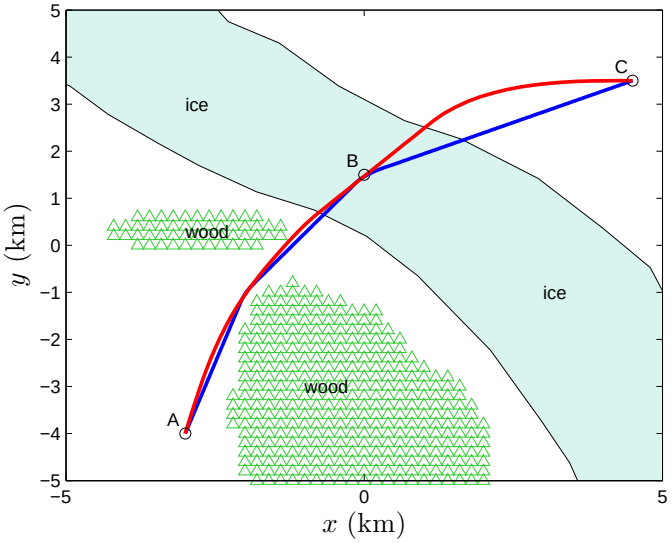


Figure 7.7: Smoothed path with local curvature constraint.

Chapter 8

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.

8.1 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 [133, 168, 66, 53]. 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[128]. 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 [35, 160, 139, 158, 156]. 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.

8.2 Equations of Motion

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[1].

Consider the following equations of motion for a point-mass model of a fixed-wing aircraft [127]:

$$\dot{x} = v\cos\gamma\cos\psi, \tag{8.1}$$

$$\dot{y} = v\cos\gamma\sin\psi, \tag{8.2}$$

$$\dot{z} = v\sin\gamma, \tag{8.3}$$

$$\dot{v} = \frac{1}{m}\left[T - F_D(C_L, v, \rho) - mg\sin\gamma\right], \tag{8.4}$$

$$\dot{\gamma} = \frac{1}{mv}\left[F_L(C_L, v, \rho)\cos\phi - mg\cos\gamma\right], \tag{8.5}$$

$$\dot{\psi} = -\frac{F_L(C_L, v, \rho)\sin\phi}{mv\cos\gamma}, \tag{8.6}$$

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

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 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:

$$x' = \cos\gamma\cos\psi, \tag{8.7}$$

$$y' = \cos\gamma\sin\psi, \tag{8.8}$$

$$z' = \sin\gamma, \tag{8.9}$$

$$v' = \frac{1}{mv}\left[T - F_D\left(C_L, v, \rho\right) - mg\sin\gamma\right], \tag{8.10}$$

$$\gamma' = \frac{1}{mv^2}\left[F_L\left(C_L, v, \rho\right)\cos\phi - mg\cos\gamma\right], \tag{8.11}$$

$$\psi' = -\frac{F_L\left(C_L, v, \rho\right)\sin\phi}{mv^2\cos\gamma}, \tag{8.12}$$

where the following relations have been used for deriving (8.7)-(8.12):

$$\mathrm{d}t = \frac{\mathrm{d}s}{v}, \tag{8.13}$$

$$\mathrm{d}s = \sqrt{\mathrm{d}^2x + \mathrm{d}^2y + \mathrm{d}^2z}, \tag{8.14}$$

$$\psi = \arctan\frac{\mathrm{d}y}{\mathrm{d}x} = \arctan\frac{y'}{x'}, \tag{8.15}$$

$$\gamma = \arctan\frac{\mathrm{d}z}{\sqrt{\mathrm{d}x^2 + \mathrm{d}y^2}} = \arctan\frac{z'}{\sqrt{x'^2 + y'^2}}, \tag{8.16}$$

$$\psi' = \frac{1}{1 + (y'/x')^2}\frac{y''x' - y'x''}{x'^2} = \frac{x'^2}{x'^2 + y'^2}\frac{y''x' - y'x''}{x'^2} = \frac{y''x' - y'x''}{x'^2 + y'^2}, \tag{8.17}$$

$$\gamma' = \frac{z''x'^2 + z''y'^2 - z'x''x' - z'y''y'}{\sqrt{x'^2 + y'^2}}. \tag{8.18}$$

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 (8.17) and (8.18). 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 (8.13). 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 8.4 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:

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

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.

8.3 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,

$$C_L(s) \in [C_{L_{\min}}(s),\, C_{L_{\max}}(s)], \quad \phi(s) \in [\phi_{\min}(s), \phi_{\max}(s)], \quad T(s) \in [T_{\min}(s), T_{\max}(s)], \quad \forall s \in [s_0, s_f], \tag{8.19}$$

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 $C_{L_{\min}}(s) \le 0 \le C_{L_{\max}}(s)$, $-\pi/2 < \phi_{\min}(s) < 0 < \phi_{\max}(s) < \pi/2$, and $0 \le T_{\min}(s) < T_{\max}(s)$, 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 (8.19) have to be updated to account for the post-failure characteristics of the airplane.

Let $E \triangleq v^2/2$ be the kinetic energy per unit mass of the aircraft. Also, let $E_{\max}(s) = v^2_{\max}(s)/2$ and $E_{\min}(s) = v^2_{\min}(s)/2$. 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 $s \in [s_0, s_f]$.

8.3.1 Lift Coefficient Constraint

From equations (8.11) and (8.12), we have

$$\gamma' = \frac{1}{2m}\rho S C_L \cos\phi - \frac{g\cos\gamma}{v^2}, \tag{8.20}$$

$$\psi' = -\frac{\rho v^2 S C_L \sin\phi}{2mv^2\cos\gamma} = -\frac{\rho S C_L \sin\phi}{2m\cos\gamma}, \tag{8.21}$$

which can be rewritten as:

$$C_L \cos\phi = \frac{2m}{\rho S}\left(\gamma' + \frac{g\cos\gamma}{v^2}\right), \tag{8.22}$$

$$C_L \sin\phi = -\frac{2m\psi'\cos\gamma}{\rho S}. \tag{8.23}$$

Eliminating ϕ from equation (8.22) and (8.23), and replacing v^2 with $2E$, one obtains

$$E = g_1(C_L;\gamma,\gamma',\psi') \triangleq \frac{mg\cos\gamma}{\rho S}\left(\sqrt{C_L^2 - \left(\frac{2m\psi'\cos\gamma}{\rho S}\right)^2} - \frac{2m\gamma'}{\rho S}\right)^{-1}. \tag{8.24}$$

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

$$0 < \sqrt{C_L^2(s) - \left(\frac{2m\psi'(s)\cos\gamma(s)}{\rho(s)S}\right)^2} - \frac{2m\gamma'(s)}{\rho(s)S} < \infty, \qquad \forall s \in [s_0, s_f]. \tag{8.25}$$

This is equivalent to the condition

$$\bar{C}_L(s) > \tilde{C}_L(s), \qquad \forall s \in [s_0, s_f], \tag{8.26}$$

where

$$\bar{C}_L(s) = \max\{-C_{L_{\min}}(s), C_{L_{\max}}(s)\}, \tag{8.27}$$

and

$$\tilde{C}_L(s) = \begin{cases} \dfrac{2m}{\rho(s)S}\,|\psi'(s)|\cos\gamma(s), & \text{if } \gamma'(s) < 0, \\ \dfrac{2m}{\rho(s)S}\sqrt{\gamma'^2(s) + \psi'^2(s)\cos^2\gamma(s)}, & \text{if } \gamma'(s) \geq 0. \end{cases} \tag{8.28}$$

The given path $(x(s), y(s), z(s))$ is infeasible if (8.26) is not satisfied, owing to insufficient lift. When (8.26) holds, and because the right hand side of equation (8.24) 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

$$E(s) \geq \underline{g}_{w1}(s) \triangleq \max\{E_{\min}(s), g_1(\bar{C}_L(s); \gamma(s), \gamma'(s), \psi'(s))\}. \tag{8.29}$$

In other words, if the problem is feasible, (8.29) 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 (8.29) that if $\underline{g}_{w1}(s)$ is unbounded, then the path is not feasible. Feasibility implies, in particular, that $\underline{g}_{w1}$ in (8.29) is a (possibly discontinuous) piecewise analytic function of s.

8.3.2 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 (8.22) and (8.23) 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 (8.22) and (8.23), thus obtaining the following equation:

$$\tan\phi = -\frac{2\psi'\cos\gamma}{2\gamma' + g\cos\gamma/E}. \tag{8.30}$$

Solving for E from equation (8.30) yields:

$$E = g_2(\phi; \gamma, \gamma', \psi') \triangleq -\frac{1}{2}\frac{g\cos\gamma\tan\phi}{\gamma'\tan\phi + \psi'\cos\gamma}. \tag{8.31}$$

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 (8.30) 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

$$E(s) \leq \mu_0(s) \triangleq \max\left\{\frac{g\tan\phi_{\min}(s)}{2\psi'(s)}, \frac{g\tan\phi_{\max}(s)}{2\psi'(s)}\right\}. \tag{8.32}$$

(iib) If $\gamma'(s) \neq 0$, rewrite (8.31) as follows:

$$\gamma'(s)\tan\phi(s) + \psi'(s)\cos\gamma(s) = -\frac{g\cos\gamma(s)}{2E(s)}\tan\phi(s). \tag{8.33}$$

The bank angle constraint $\phi(s) \in [\phi_{\min}(s), \phi_{\max}(s)]$ limits the admissible value of $E(s)$ via equation (8.33). A necessary and sufficient condition for the satisfaction of this constraint is

$$\frac{g\cos\gamma(s)}{2E(s)} \leq \mu_1(s) \triangleq \min\left\{h(s;\phi_{\min},\gamma,\gamma',\psi'), h(s;\phi_{\max},\gamma,\gamma',\psi')\right\}, \tag{8.34}$$

or

$$\frac{g\cos\gamma(s)}{2E(s)} \geq \mu_2(s) \triangleq \max\left\{-h(s;\phi_{\min},\gamma,\gamma',\psi'), -h(s;\phi_{\max},\gamma,\gamma',\psi')\right\}, \tag{8.35}$$

where

$$h(s;\phi,\gamma,\gamma',\psi') \triangleq \gamma'(s) + \psi'(s)\cos\gamma(s)/\tan\phi(s). \tag{8.36}$$

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 (8.35) always holds as long as $E(s) > 0$.

(iib.2) If $\mu_1(s) \leq 0$ and $\mu_2(s) > 0$, then equation (8.34) does not hold, and equation (8.35) must be satisfied, which is equivalent to the following constraint on $E(s)$

$$E(s) \leq \frac{1}{2}g\cos\gamma(s)/\mu_2(s). \tag{8.37}$$

(iib.3) Finally, if $\mu_1(s) > 0$, then it is required that either (8.37) holds, or the following inequality holds:

$$E(s) \geq \frac{1}{2}g\cos\gamma(s)/\mu_1(s). \tag{8.38}$$

Equations (8.32), (8.37) and (8.38) define the admissible values of $E(s)$ limited by the bank angle.

8.3.3 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

$$\underline{g}_{w1}(s) \leq E(s) \leq \overline{g}_{w1}(s), \qquad s \in [s_0, s_f], \tag{8.39}$$

where $\underline{g}_{w1}(s)$ from (8.29) 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

$$\underline{g}_{w2}(s) \stackrel{\triangle}{=} \max \left\{ \underline{g}_{w1}(s), \underline{g}_{w3}(s) \right\} \leq E(s) \leq \bar{g}_{w2}(s), \qquad s \in [s_0, s_f]. \tag{8.40}$$

where,

$$\underline{g}_{w3}(s) \stackrel{\triangle}{=} \begin{cases} \max \{ E_{\min}(s), g \cos \gamma(s) / 2\mu_1(s) \}, & s \in \Gamma_3, \\ E_{\min}(s), & s \in [s_0, s_f] / \Gamma_3, \end{cases} \tag{8.41}$$

and

$$\overline{g}_{w2}(s) \stackrel{\triangle}{=} \begin{cases} E_{\max}(s), & s \in \Gamma_3, \\ \overline{g}_{w1}(s), & s \in [s_0, s_f] / \Gamma_3. \end{cases} \tag{8.42}$$

and where $\mu_0(s), \mu_1(s)$ and $\mu_2(s)$ are given in (8.32), (8.34) and (8.35), respectively.

The collection of points $(s, E(s))$ satisfying either (8.39) or (8.40) 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

$$\mathcal{W}_1 = \{ (s, E) \mid \underline{g}_{w1}(s) \leq E \leq \overline{g}_{w1}(s), s \in [s_0, s_f] \}, \tag{8.43}$$
$$\mathcal{W}_2 = \{ (s, E) \mid \underline{g}_{w2}(s) \leq E(s) \leq \overline{g}_{w2}(s), s \in [s_0, s_f] \}, \tag{8.44}$$

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.

8.3.4 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

   $$\mathcal{W} = \{ (s, E) | \underline{g}_w(s) \leq E(s) \leq \overline{g}_w(s), s \in [s_0, s_f] \}. \tag{8.45}$$

   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 (8.45). 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 (8.43) and (8.44), 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 $[s_0, s_f]$. 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. 8.1 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 $\mathcal{W}$.

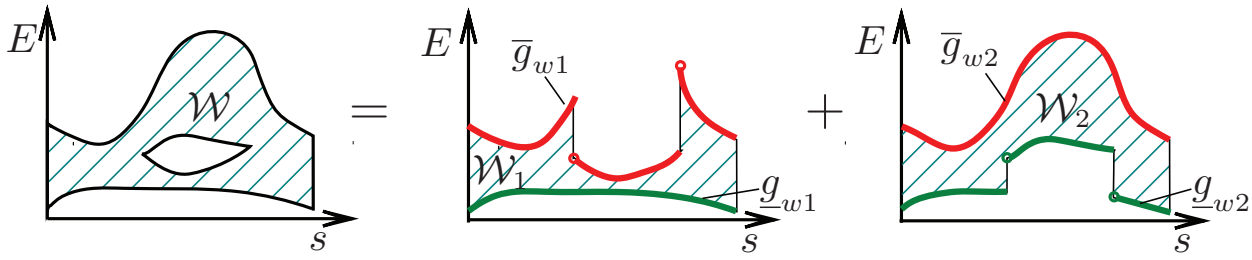


Figure 8.1: Decomposition of $\mathcal{W}$ when it is not simply connected.

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.

8.3.5 Thrust Constraint

From equations (8.10), (8.22) and (8.23) we have the following equation:

$$\begin{aligned} T &= mvv' + \left(\frac{1}{2}C_{D_0}\rho S + \frac{2Km^2\gamma'^2}{\rho S} + \frac{Km^2\cos^2\gamma\psi'^2}{\rho S}\right)v^2 \\ &\quad + \frac{2Km^2g^2\cos^2\gamma}{\rho S}\frac{1}{v^2} + \frac{4Km^2\gamma' g\cos\gamma}{\rho S} + mg\sin\gamma. \end{aligned} \tag{8.46}$$

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

$$E'(s) = \frac{T(s)}{m} + c_1(s)E(s) + \frac{c_2(s)}{E(s)} + c_3(s), \tag{8.47}$$

where

$$c_1(s) \triangleq -\frac{C_{D_0}(s)\rho(s)S}{m} - \frac{4Km\gamma'^2(s)}{\rho(s)S} - \frac{2Km\cos^2\gamma(s)\psi'^2(s)}{\rho(s)S}, \tag{8.48}$$

$$c_2(s) \triangleq -\frac{Kmg^2\cos^2\gamma(s)}{\rho(s)S}, \tag{8.49}$$

$$c_3(s) \triangleq -\frac{4Km\gamma'(s)g\cos\gamma(s)}{\rho(s)S} - g\sin\gamma(s). \tag{8.50}$$

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

8.4 Optimal Control Formulation

The extensive analysis of the previous section reveals that instead of working with the original dynamical system described by (8.7) - (8.12), 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[158] 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. [158], 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. [35, 160, 139, 158, 156]. 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 8.1 (**Minimum-Time Path-Tracking Problem**)**.** Consider the following optimal control problem in Lagrange form:

$$\min_{T} \quad J(s_0, s_f, E(s_0), E(s_f), T) = t_f = \int_{s_0}^{s_f} \frac{\mathrm{d}s}{\sqrt{2E(s)}} \tag{8.51a}$$

$$\text{subject to} \quad E'(s) = \frac{T(s)}{m} + c_1(s)E(s) + \frac{c_2(s)}{E(s)} + c_3(s), \tag{8.51b}$$

$$\underline{g}_w(s) \leq E(s) \leq \overline{g}_w(s), \tag{8.51c}$$

$$E(s_0) = v_0^2/2, \tag{8.51d}$$

$$E(s_f) = v_f^2/2, \tag{8.51e}$$

$$T_{\min} \leq T(s) \leq T_{\max}, \tag{8.51f}$$

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 8.3.

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 8.5. The functions c_1, c_2 and c_3 are also piecewise analytic, and are given in equations (8.48), (8.49) and (8.50). They can be readily computed once the path is given.

Consider the case when the state constraint (8.51c) 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

$$\lambda' = -\frac{\partial H}{\partial E} = -c_1 \lambda + c_2 \lambda E^{-2} + \frac{1}{2\sqrt{2}} E^{-3/2}. \tag{8.52}$$

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

$$\frac{\partial H}{\partial T} = \frac{2\lambda}{m}. \tag{8.53}$$

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 8.1 may contain bang-bang control, singular control, and control arcs associated with active state constraints, as described by the following expression

$$T^*(s) = \begin{cases} T_{\min}, & \text{for } \lambda > 0, s \in [s_0, s_f] \setminus \mathcal{K}, \\ \text{singular control}, & \text{for } \lambda = 0, s \in [s_0, s_f] \setminus \mathcal{K}, \\ T_{\max}, & \text{for } \lambda < 0, s \in [s_0, s_f] \setminus \mathcal{K}, \\ \overline{T}_w(s), & \text{for } s \in \mathcal{K}_U, \\ \underline{T}_w(s), & \text{for } s \in \mathcal{K}_L. \end{cases} \tag{8.54}$$

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

$$\overline{T}_w(s) = m\left(\overline{g}'_w(s) - c_1(s)\overline{g}_w(s) - c_3(s) - c_2(s)/\overline{g}_w(s)\right), \tag{8.55}$$

and respectively,

$$\underline{T}_w(s) = m(\underline{g}'_w(s) - c_1(s)\underline{g}_w(s) - c_3(s) - c_2(s)/\underline{g}_w(s)). \tag{8.56}$$

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

$$\overline{T}_w(s^\pm) = m\left(\overline{g}'_w(s^\pm) - c_1(s)\overline{g}_w(s^\pm) - c_3(s) - c_2(s)/\overline{g}_w(s^\pm),\right) \tag{8.57}$$

and

$$\underline{T}_w(s^\pm) = m(\underline{g}'_w(s^\pm) - c_1(s)\underline{g}_w(s^\pm) - c_3(s) - c_2(s)/\underline{g}_w(s^\pm)), \tag{8.58}$$

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 8.1. The optimal control solution of Problem 8.1 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 (8.52) 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 8.2. 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 (8.51c) 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

$$\begin{aligned} E_m^{-\prime}(s;\eta) - E_m^{+\prime}(s;0) &= (T_{\min} - T_{\max})/m + c_1(s)\left(E_m^-(s;\eta) - E_m^+(s;0)\right) + c_2(s)\left(\frac{1}{E_m^-(s;\eta)} - \frac{1}{E_m^+(s;0)}\right) \\ &= (T_{\min} - T_{\max})/m + \left(c_1(s) - \frac{c_2(s)}{E_m^-(s;\eta)E_m^+(s;0)}\right)\left(E_m^-(s;\eta) - E_m^+(s;0)\right) \\ &\leq (T_{\min} - T_{\max})/m + \left|c_1(s) - \frac{c_2(s)}{E_m^-(s;\eta)E_m^+(s;0)}\right|\left|E_m^-(s;\eta) - E_m^+(s;0)\right|. \end{aligned} \tag{8.59}$$

and

$$\begin{aligned} E_m^{-\prime}(s;0) - E_m^{+\prime}(s;\eta) &= (T_{\min} - T_{\max})/m + c_1(s)\left(E_m^-(s;0) - E_m^+(s;\eta)\right) + c_2(s)\left(\frac{1}{E_m^-(s;0)} - \frac{1}{E_m^+(s;\eta)}\right) \\ &= (T_{\min} - T_{\max})/m + \left(c_1(s) - \frac{c_2(s)}{E_m^-(s;0)E_m^+(s;\eta)}\right)\left(E_m^-(s;0) - E_m^+(s;\eta)\right) \\ &\leq (T_{\min} - T_{\max})/m + \left|c_1(s) - \frac{c_2(s)}{E_m^-(s;0)E_m^+(s;\eta)}\right|\left|E_m^-(s;0) - E_m^+(s;\eta)\right|. \end{aligned} \tag{8.60}$$

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

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

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 (8.59) is bounded, it follows that $E_m^{-\prime}(s;\eta) - E_m^{+\prime}(s;0) \leq (T_{\min} - T_{\max})/m < 0$ 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 $E_m^{-\prime}(s;0) - E_m^{+\prime}(s;\eta) \leq (T_{\min} - T_{\max})/m < 0$ 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, $E_m^{-\prime}(s;\eta) - E_m^{+\prime}(s;0) < (T_{\min} - T_{\max})/2m < 0$ and $E_m^{-\prime}(s;0) - E_m^{+\prime}(s;\eta) < (T_{\min} - T_{\max})/2m < 0$ 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 $s_m - \epsilon < s \le s_m$, and $E_m^-(s;\eta) < \overline{g}_w(s)$ for all $s_m < s \le s_m + \epsilon$, 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. 8.2.

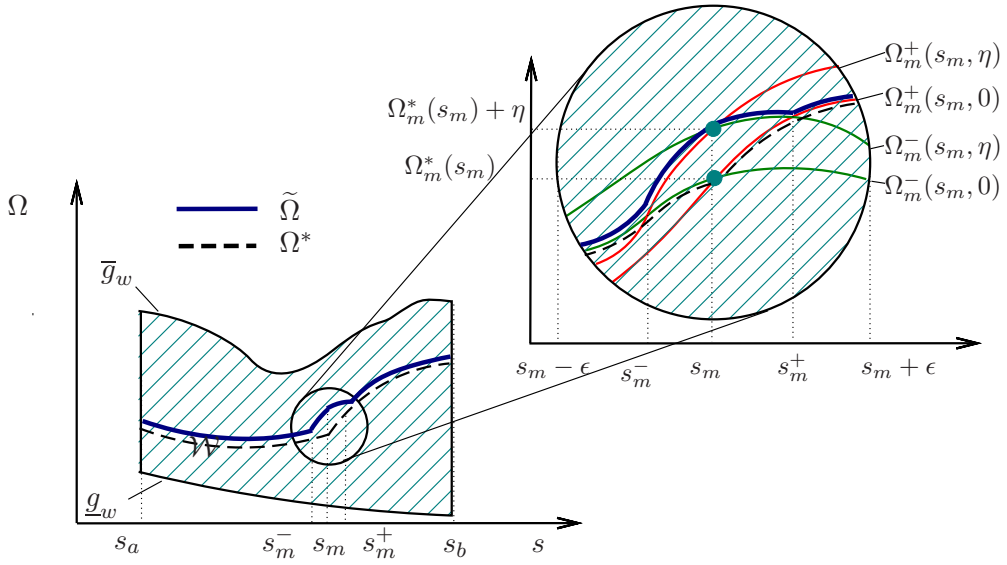


Figure 8.2: Speed variation for the proof of Proposition 8.2.

Now consider the variation of T^* (see Fig. 8.3) 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. 8.2. 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. □

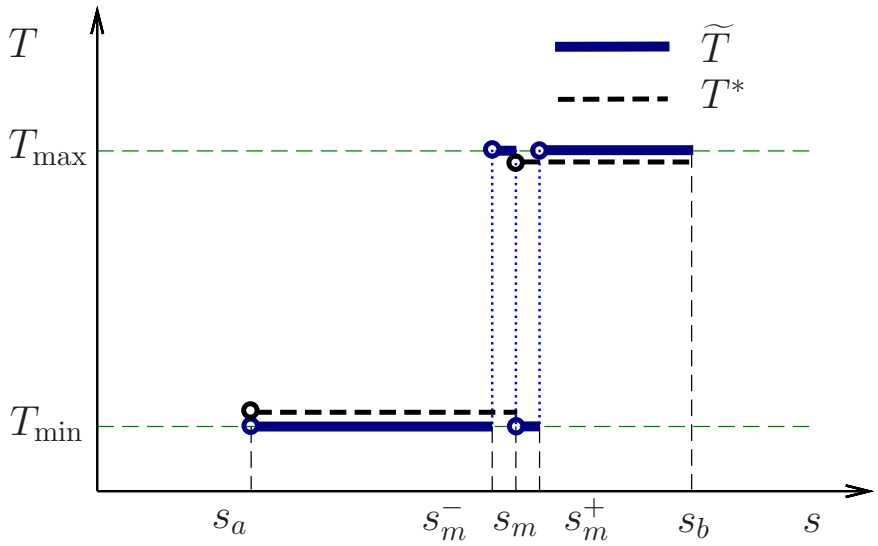


Figure 8.3: Thrust variation for proof of Proposition 8.2.

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 8.3. 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 8.1. 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 8.2 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 8.1. The time optimal control T^* for Problem 8.1 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 8.3. □

8.5 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 (8.51a) using the necessary conditions introduced in the previous section.

Assuming that the given path is feasible, then according to Proposition 8.3, 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., $E^*(s) \leq \overline{g}_w(s)$, there exists a control $T^*(s) \in [T_{\min}, T_{\max}]$ such that $E^*(s)$ satisfies the following inequality

$$\frac{E^*(s_d + h) - E^*(s_d)}{h} \leq \frac{\overline{g}_w(s_d + h) - \overline{g}_w(s_d)}{h}, \tag{8.61}$$

where h is a small positive real number. By taking the limits of both sides of (8.61) with $h \to 0$, the last expression leads to the existence of $T^*(s) \in [T_{\min}, T_{\max}]$ such that

$$E^{*\prime}(s_d^+) \leq \overline{g}'_w(s_d^+). \tag{8.62}$$

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, (8.62) implies

$$\overline{g}'_w(s_d^+) \geq \frac{T_{\min}}{m} + c_1(s_d^+)E^*(s_d) + \frac{c_2(s_d^+)}{E^*(s_d)} + c_3(s_d^+) = \frac{T_{\min}}{m} + c_1(s_d^+)\overline{g}_w(s_d) + \frac{c_2(s_d^+)}{\overline{g}_w(s_d)} + c_3(s_d^+). \tag{8.63}$$

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

$$\overline{g}'_w(s_d^-) \leq \frac{T_{\max}}{m} + c_1(s_d^-)E^*(s_d) + \frac{c_2(s_d^-)}{E^*(s_d)} + c_3(s_d^-) = \frac{T_{\max}}{m} + c_1(s_d^-)\overline{g}_w(s_d) + \frac{c_2(s_d^-)}{\overline{g}_w(s_d)} + c_3(s_d^-). \tag{8.64}$$

Therefore, $E^*(s_d) = \overline{g}_w(s_d)$ is possible only if both (8.63) and (8.64) 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 (8.63) and (8.64) are reduced to

$$\frac{T_{\min}}{m} + c_1(s_d^+)\overline{g}_w(s_d) + \frac{c_2(s_d^+)}{\overline{g}_w(s_d)} + c_3(s_d^+) \leq \overline{g}'_w(s_d) \leq \frac{T_{\max}}{m} + c_1(s_d^-)\overline{g}_w(s_d) + \frac{c_2(s_d^-)}{\overline{g}_w(s_d)} + c_3(s_d^-). \tag{8.65}$$

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 $E^*(s) \leq \overline{g}_w(s)$ 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, (8.63) holds if $\overline{g}_w(s_d^+) < \overline{g}_w(s_d^-)$, and (8.64) 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, (8.63) and (8.64) hold, that is,

$$\widetilde{W} \triangleq \{(s_d, \overline{g}_w(s_d)) | (8.63) \text{ and } (8.64) \text{ hold}, s_d \in [s_0, s_f]\}.$$

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 (8.63) or (8.64) 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. 8.4. By the piecewise analyticity assumption of the given path, all functions involved in (8.63) and (8.64) are piecewise analytic, and it follows that the equality in (8.63) and (8.64) 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. [139]. 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.

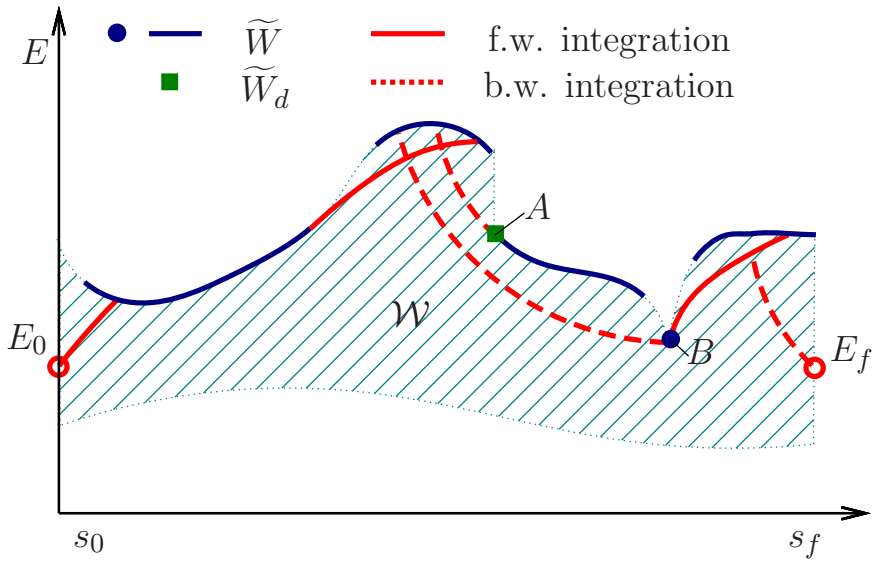


Figure 8.4: Elements for the optimal E

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. [35, 139, 158, 160], use a "search, integrate 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.

8.5.1 Algorithm I

Step 1 Compute $\overline{g}_w$, $\underline{g}_w$ as in Section 8.3.3 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

$$S^{\pm}(s) = \begin{cases} \mathcal{S}^{\pm}(s), & s \in I_j^{\pm}, \\ \overline{g}_w(s), & s \in [s_0, s_f] \backslash I_j^{\pm}, \end{cases} \tag{8.66}$$

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

$$E(s) \triangleq \min\{S_0^+(s), S_1^+(s), \ldots, S_{N-1}^+(s), S_1^-(s), S_2^-(s), \ldots, S_N^-(s)\}. \tag{8.67}$$

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 (8.67). 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 (8.46). By construction, the optimal thrust profile $T^*(s)$ satisfies the necessary conditions given by Proposition 8.2 and Theorem 8.1. 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.

8.5.2 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. [35, 160, 139, 158]) is illustrated in Fig. 8.5. While Algorithm II computes only the integrations which are involved in the construction of the optimal speed profile, the algorithms in Refs. [35, 160, 139, 158] 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.

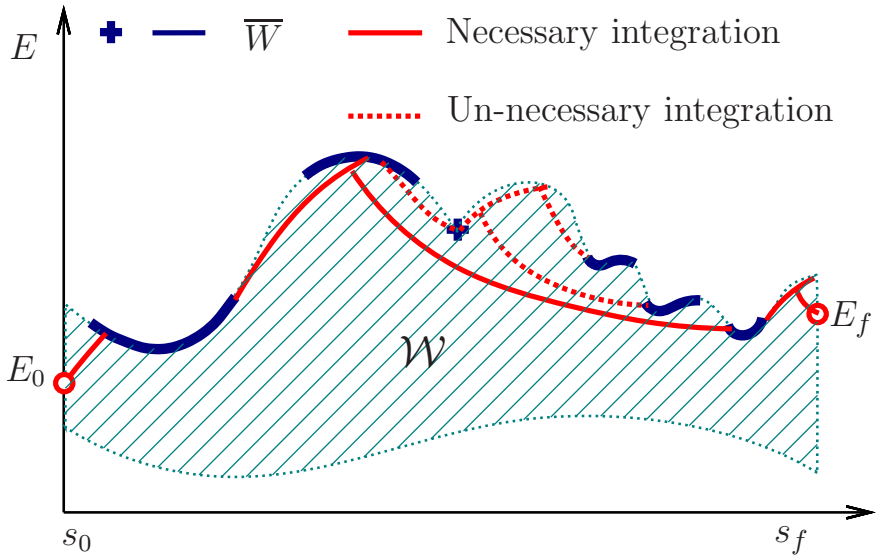


Figure 8.5: Algorithm comparison

8.6 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 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.

8.6.1 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. 8.6. The initial position of the aircraft is $(0, 0, 6)$ km, the aircraft flies with $v_0 = 220$ m/s, at $\gamma(0) = 0^\circ$ path angle and $\psi(0) = 0^\circ$ heading. The final position is (111.0,17.3,0) km, with final speed $v(s_f) = 95$ m/s, path angle $\gamma(s_f) = 0^\circ$ and heading $\psi(s_f) = -25^\circ$. The horizontal projection of the trajectory contains two constant rate turning maneuvers. The atmospheric density data are taken from Ref. [130]. For simplicity, the change of gravity with altitude is neglected.

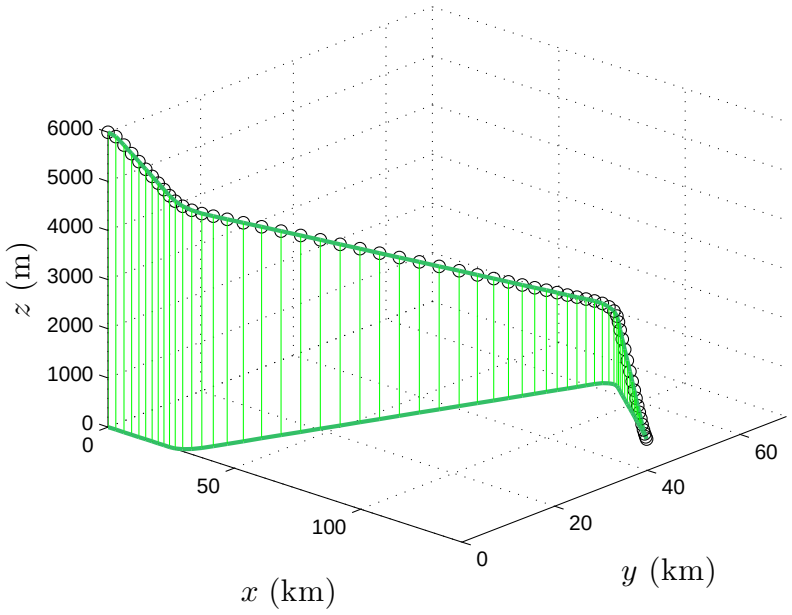


Figure 8.6: 3D Geometric Trajectory.

The control bounds are given as follows: the lift coefficient $C_L \in [-0.067, 1.9]$, the bank angle $\phi \in [-15^\circ, 15^\circ]$ 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 method, the minimum-time speed profile $v^*(s)$ is computed following the approach developed in this chapter and is shown in Fig. 8.8. The same profile in terms of time is shown in Fig. 8.9. To arrive at the final position in minimum time, the aircraft should fly as fast as possible, however, due to the limited acceleration

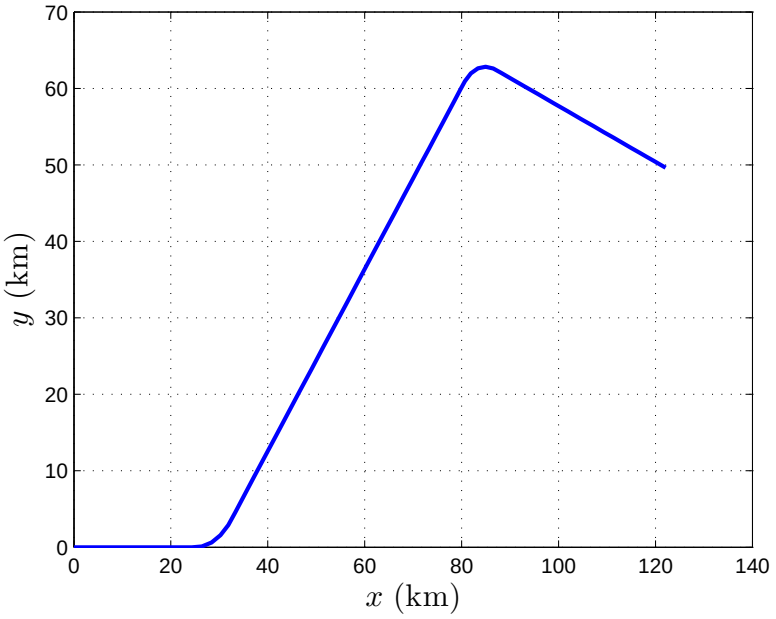


Figure 8.7: X-Y plane projection of the geometric trajectory.

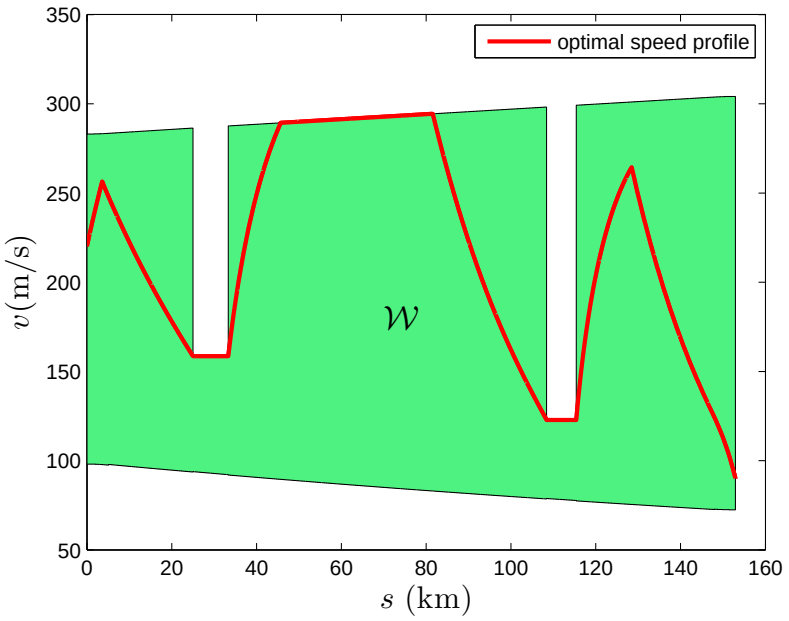


Figure 8.8: Optimal speed profile under path coordinate.

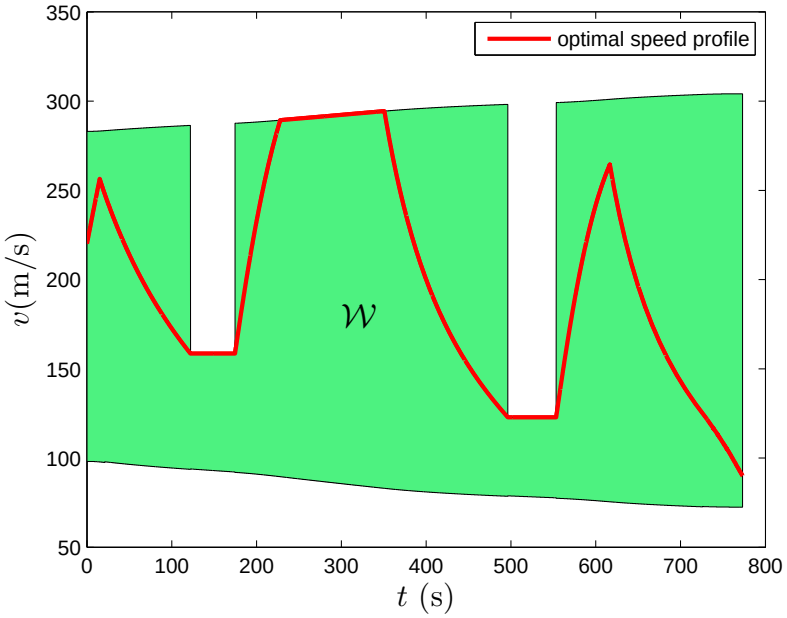


Figure 8.9: Time history of optimal speed.

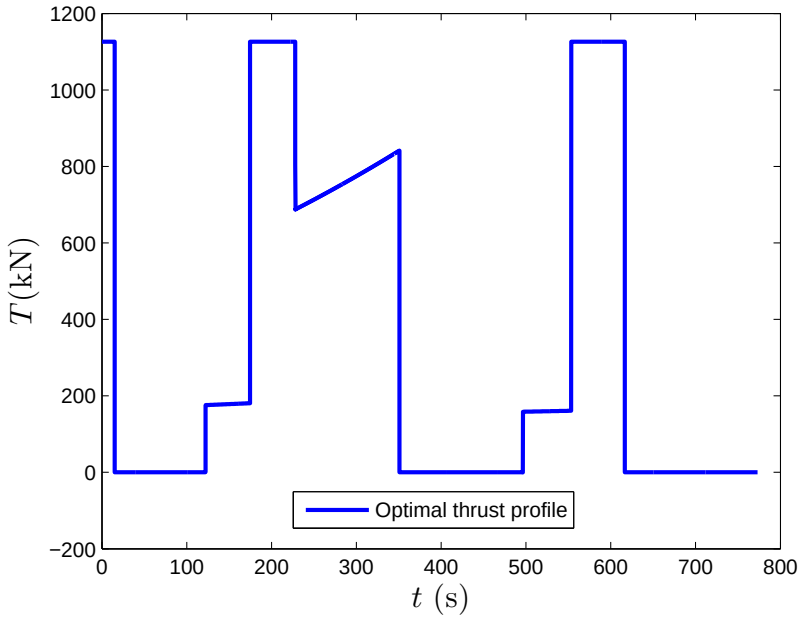


Figure 8.10: Optimal thrust.

and deceleration capability, the optimal velocity profile cannot necessarily stay at $v_{\max}$ all the time. Within $0 \leq s \leq 25\,\text{km}$, the upper limit of speed is higher than $270\,\text{m/s}$, 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 $25 \leq s \leq 33\,\text{km}$, 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 km, and the aircraft finishes in 771 s using the optimal thrust with an average speed of 170.4 m/s.

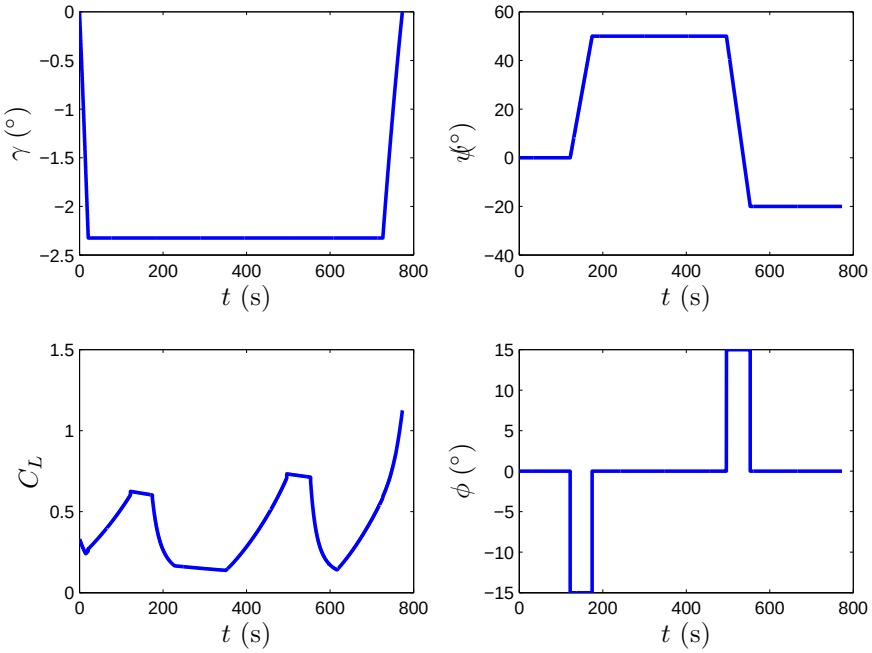


Figure 8.11: The states and control histories of the time parameterized trajectory.

The state and control histories recovered from the optimally time-parameterized trajectory are shown in Fig.8.11. 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 control histories from the optimal time-parameterized trajectory $\left(x^*(t), y^*(t), z^*(t)\right)$. 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 (8.1)-(8.6) 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. 8.12.

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

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

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

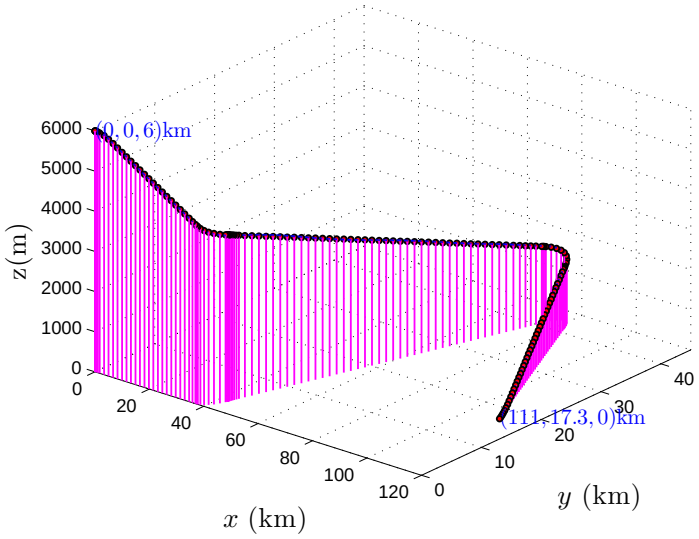


Figure 8.12: Comparison of the original geometric path(dots) and the path generated using time parameterization and inverse dynamics(line).

8.6.2 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 [192]. The accuracy and 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 $v(0) = 240\,\text{m/s}$, heading angle $\psi(0) = 0^\circ$ and the path angle $\gamma(0) = 0^\circ$; the final conditions are: speed $v(s_f) = 95\,\text{m/s}$, heading angle $\psi(s_f) = 80^\circ$, and path angle $\gamma(s_f) = -3^\circ$. The aircraft considered in this example is a Boeing-747. During the whole flight, the following constraints need to be satisfied: $v \leq 270\,\text{m/s}$, $\phi \in [-15, 15]^\circ$, $C_L \in [-0.31, 1.52]$, and $T \in [0, 1126.3]\,\text{kN}$. The path is shown in Figs. 8.13 and 8.14.

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. 8.17-8.20.

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. 8.15 and 8.16, respectively. The time history of the speed and the controls are shown in Figs. 8.17-8.20. As mentioned before, the 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. 8.17-8.20,

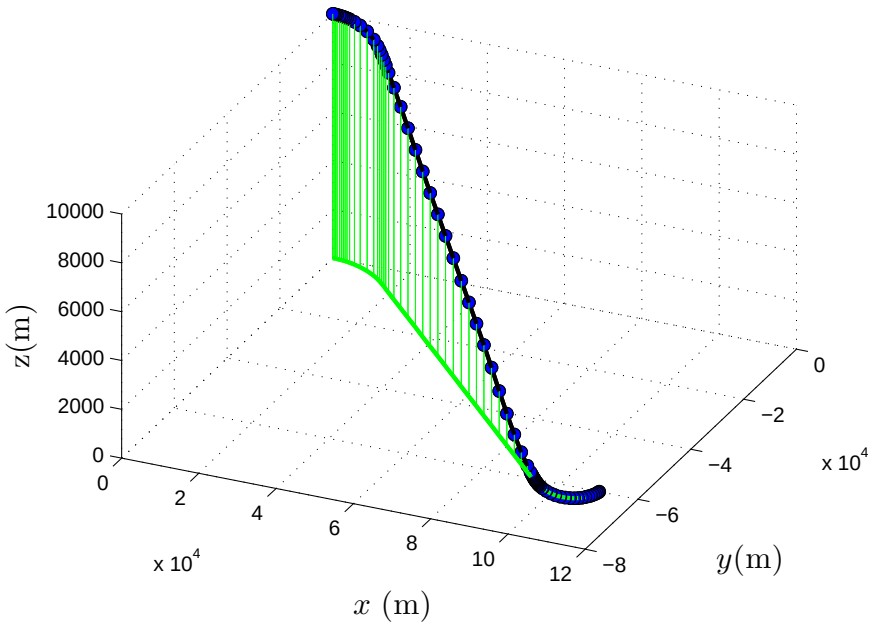


Figure 8.13: the min-time trajectory.

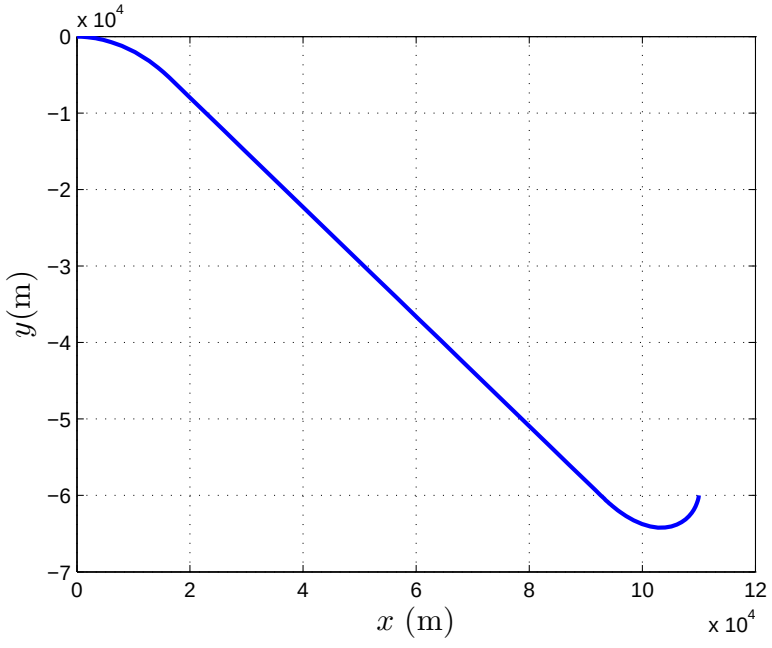


Figure 8.14: X-Y plane projection of the min-time trajectory.

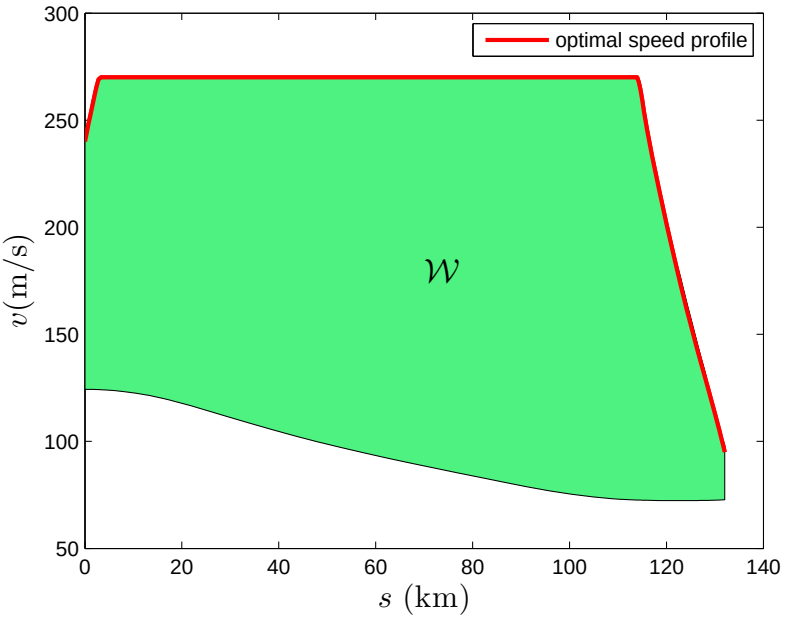


Figure 8.15: Optimal speed profile under path coordinate (DENMRA).

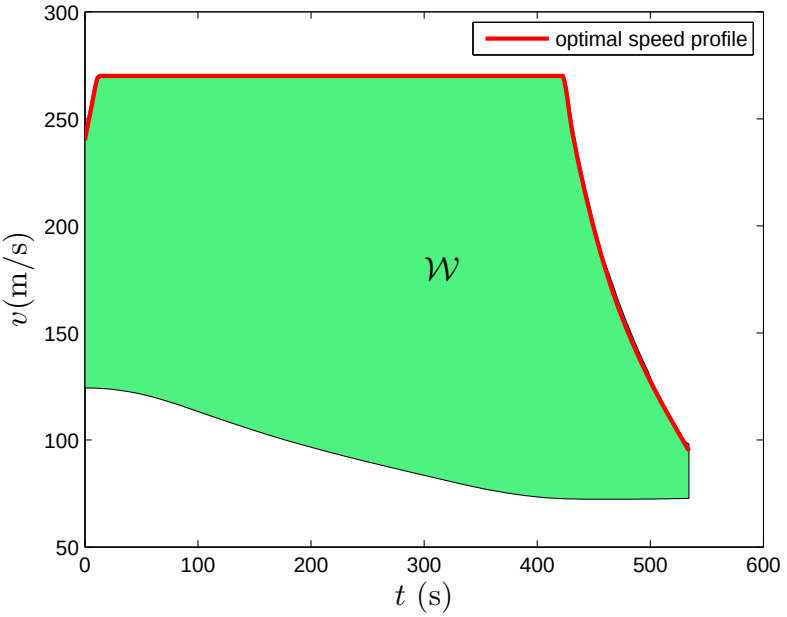


Figure 8.16: Time history of optimal speed (DENMRA).

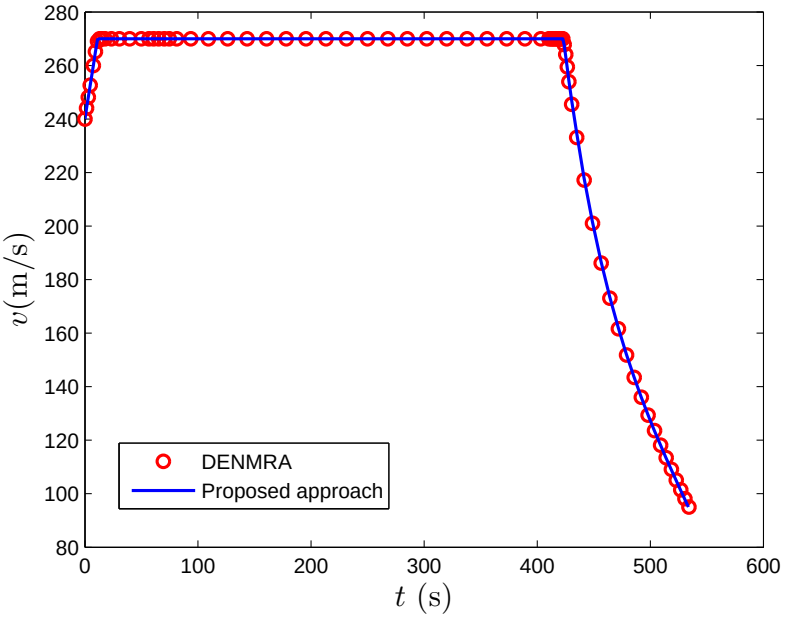


Figure 8.17: Speed comparison.

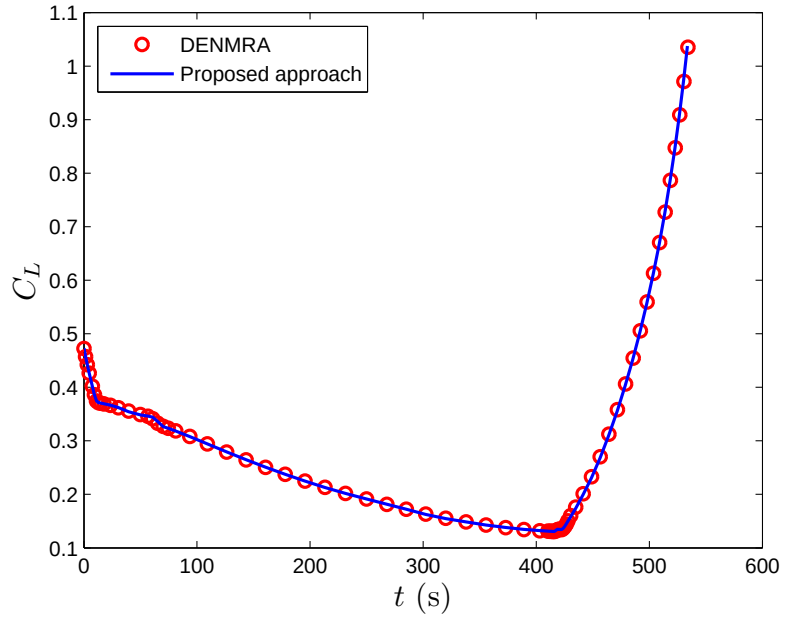


Figure 8.18: Control comparison: C_L.

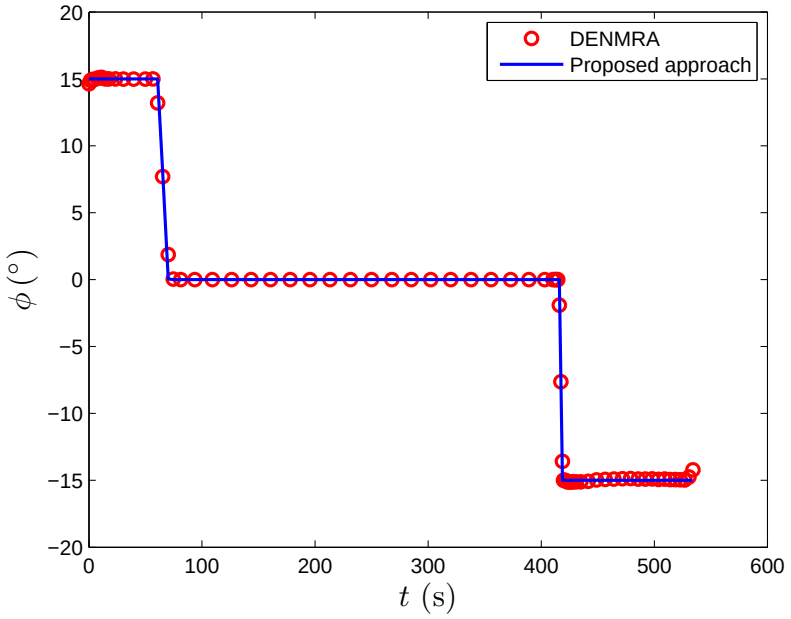


Figure 8.19: Control comparison: ϕ.

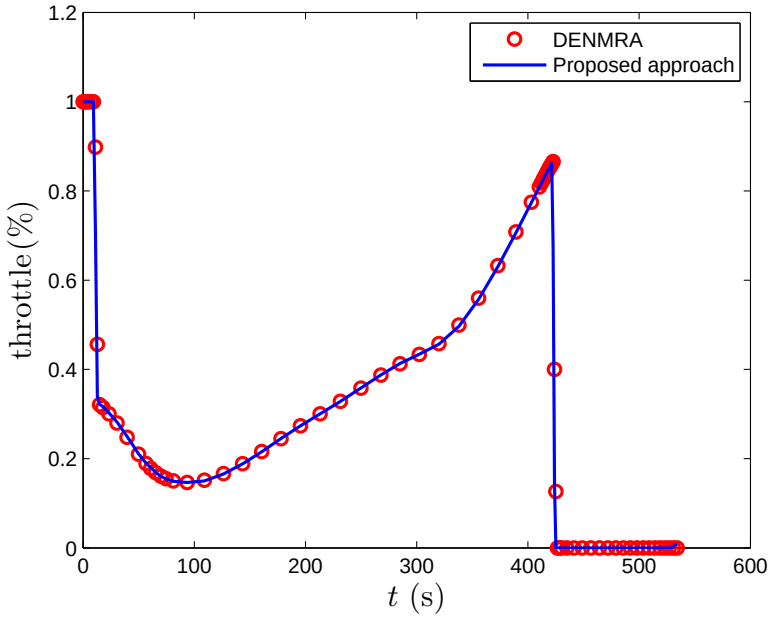


Figure 8.20: Control comparison: throttle.

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.

Chapter 9

Energy-Optimal Landing Path Tracking with Fixed Time of Arrival

In this chapter, we present 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 8, 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.

9.1 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 [35, 160, 139, 158, 156, 157, 55, 32, 78, 177, 44], ground vehicles [11, 88, 110, 82, 176], aircraft [109, 118], etc. Although optimal motion planning problems can be solved directly using numerical optimization techniques [40, 41, 167, 44, 52, 106], 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 [76, 141]. 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 [35, 160, 139, 158, 156, 176, 191]. The optimal solution to these

problems can help improve plant productivity [35, 160, 139, 158, 156], racing car performance [176], or achieve faster aircraft landing in case of an emergency [191]. 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. [11, 88, 110, 82]. 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 [11, 88, 110, 82], 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 and the analytic expression of the singular control. Finally, the guessed switching structure is applied to solve the singular control problem [179]. 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 [11, 88, 110, 82]), 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. [11, 88, 110, 82, 35, 160, 139, 158, 156, 157, 176, 191] share the same key steps of solving a scalar functional optimization problem. For train operations, for instance [11, 88, 110, 82], 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 [35, 160, 139, 158, 156, 157, 176, 191], 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. [191], 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 [11, 88, 110, 82], 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. [11, 88, 110, 82].

The rest of this chapter is organized as follows: We first formulate the aircraft minimum-energy

[1] The "singular arcs" in Ref. [156] actually refer to segments of the speed profile with active speed constraints, which is different from the traditional term used in optimal control [37].

fixed TOA path-tracking problem as an optimal control problem in Section 9.2. Then, in Section 9.3 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 9.4. The validity of the proposed method is tested using numerical experiments, and the results are presented at the end of the chapter.

9.2 Aircraft Dynamics and Simplified Problem

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

$$\begin{aligned}
\dot{x} &= v\cos\gamma\cos\psi, && (9.1)\\
\dot{y} &= v\cos\gamma\sin\psi, && (9.2)\\
\dot{z} &= v\sin\gamma, && (9.3)\\
\dot{v} &= \frac{1}{m}\left[T - F_D(C_L, v, z) - mg\sin\gamma\right], && (9.4)\\
\dot{\gamma} &= \frac{1}{mv}\left[F_L(C_L, v, z)\cos\phi - mg\cos\gamma\right], && (9.5)\\
\dot{\psi} &= -\frac{F_L(C_L, v, z)\sin\phi}{mv\cos\gamma}, && (9.6)
\end{aligned}$$

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:

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

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

$$C_L \in [C_{L_{\min}}, C_{L_{\max}}], \quad \phi \in [\phi_{\min}, \phi_{\max}], \quad T \in [T_{\min}, T_{\max}], \tag{9.7}$$

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. [191] 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:

$$E(s) - \overline{g}_w(s) \leq 0, \tag{9.8}$$

$$\underline{g}_w(s) - E(s) \leq 0, \tag{9.9}$$

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 [191]:

$$E'(s) = \frac{T(s)}{m} + c_1(s)E(s) + \frac{c_2(s)}{E(s)} + c_3(s), \tag{9.10}$$

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

$$c_1(s) \triangleq -\frac{C_{D_0}(s)\rho(s)S}{m} - \frac{4Km\gamma'^2(s)}{\rho(s)S} - \frac{2Km\cos^2\gamma(s)\psi'^2(s)}{\rho(s)S}, \tag{9.11}$$

$$c_2(s) \triangleq -\frac{Kmg^2\cos^2\gamma(s)}{\rho(s)S}, \tag{9.12}$$

$$c_3(s) \triangleq -\frac{4Km\gamma'(s)g\cos\gamma(s)}{\rho(s)S} - g\sin\gamma(s). \tag{9.13}$$

Once the optimal specific kinetic energy $E^*(s)$ is obtained, the optimal thrust profile $T^*(s)$ along the path can be determined using equation (9.10). Subsequently, the other optimal control inputs can also be computed using inverse dynamics as follows:

$$C_L^*(s) = \frac{1}{\rho E^*(s)S}\Big(T^*(s) - mE^{*\prime}(s) - mg\sin\gamma(s)\Big), \tag{9.14}$$

$$\phi^*(s) = -\arctan\left(\frac{2\cos\gamma(s)\psi'(s)}{2\gamma'(s) + g\cos\gamma(s)/E^*(s)}\right). \tag{9.15}$$

9.3 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.

9.3.1 Fuel-Optimal and Energy-Optimal Problem Formulation

Most 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 [122]

$$\dot{f} = -\eta T, \tag{9.16}$$

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

$$\eta = (a + bM_a)\sqrt{\eta_0/(1 + cM_a^2)}, \tag{9.17}$$

where M_a is the Mach number and a, b, c are constants depending on the engine type. In (9.17), $\eta_0 = \eta_0(z, M_a)$ varies with altitude and Mach number and can be determined from look-up data tables [122]. The fuel consumption models for other types of jet engines are similar to equations (9.16) and (9.17), but with different parameters.

With the above model, the fuel consumption during the landing phase can be estimated by

$$J_f = \int_{t_0}^{t_f} -\dot{f}(t)\,\mathrm{d}t = \int_{t_0}^{t_f} \eta(t)T(t)\,\mathrm{d}t. \tag{9.18}$$

From (9.18) 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 (9.17). 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

$$J_w = \int_{t_0}^{t_f} v(t)T(t)\,\mathrm{d}t = \int_{s_0}^{s_f} T(s)\,\mathrm{d}s. \tag{9.19}$$

As demonstrated in Ref [42], 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 (9.19) is expected to provide a reasonably good approximation to the fuel optimization problem (9.18). This is verified by numerical results in Section 9.5. 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 (8.47), and assume that m is constant during the landing phase. The validity of such an assumption is justified in Ref. [42], 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 [75].

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 9.1 (**Minimum-energy path-tracking problem with fixed TOA**). Consider the following optimal control problem in Lagrange form:

$$\min_{T} \quad \int_{s_0}^{s_f} T(s)\,\mathrm{d}s, \tag{9.20}$$

$$\text{subject to} \quad E'(s) = \frac{T(s)}{m} + c_1(s)E(s) + \frac{c_2(s)}{E(s)} + c_3(s), \tag{9.21}$$

$$t'(s) = \frac{1}{\sqrt{2E(s)}}, \tag{9.22}$$

$$E(s) - \overline{g}_w(s) \leq 0, \tag{9.23}$$

$$\underline{g}_w(s) - E(s) \leq 0, \tag{9.24}$$

$$E(s_0) = v_0^2/2, \tag{9.25}$$

$$E(s_f) = v_f^2/2, \tag{9.26}$$

$$T_{\min}(s) \leq T(s) \leq T_{\max}(s), \tag{9.27}$$

$$t(s_f) = t_f. \tag{9.28}$$

To solve this problem, we apply the necessary conditions for optimality to screen the allowable thrust profile candidates. This is done next.

9.3.2 Optimality Conditions

First, consider the case when the state constraints (9.23) and (9.24) are not active. The Hamiltonian for Problem 9.1 is given by

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

where λ_E and λ_t are the costates corresponding to the dynamics for E and t, respectively. The costate dynamics are given by:

$$\lambda_E' = -\frac{\partial H}{\partial E} = -c_1 \lambda_E + c_2 E^{-2} \lambda_E + \frac{1}{2\sqrt{2}} E^{-3/2} \lambda_t, \tag{9.29}$$

$$\lambda_t' = -\frac{\partial H}{\partial t} = 0. \tag{9.30}$$

Therefore, the costate λ_t is constant. The switching function is given by

$$\frac{\partial H}{\partial T} = 1 + \frac{\lambda_E}{m}. \tag{9.31}$$

By Pontryagin's Maximum Principal (PMP), the extremal control is given by

$$T = \begin{cases} T_{\max}, & 1 + \lambda_E/m < 0, \\ \tilde{T}, & 1 + \lambda_E/m = 0, \\ T_{\min}, & 1 + \lambda_E/m > 0, \end{cases} \tag{9.32}$$

where $\tilde{T}$ is the singular control. On singular arcs, the switching function (9.31) is identically zero. Hence, the derivative of the switching function must also vanish on singular arcs, which yields

$$\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\partial H}{\partial T}\right) = \frac{\lambda_E'}{m} = \frac{1}{m}\left(-c_1\lambda_E + c_2E^{-2}\lambda_E + \frac{1}{2\sqrt{2}}E^{-3/2}\lambda_t\right) = c_1 - c_2E^{-2} + \frac{1}{2\sqrt{2}m}E^{-3/2}\lambda_t \equiv 0, \quad (9.33)$$

from which the singular specific kinetic energy profile can be computed. For notational convenience, equation (9.33) is rewritten as

$$P(E(s), s) = \lambda_t, \quad (9.34)$$

where

$$P(E(s), s) = -2\sqrt{2}m\left(c_1(s)E^{3/2}(s) - c_2(s)E^{-1/2}(s)\right). \quad (9.35)$$

Let $E^*(s)$ be the optimal specific kinetic energy profile for Problem 9.1 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 9.1 is unique, and we will focus on the energy-optimal path-tracking problem.

Proposition 9.1. Let $E^*(s)$ be the optimal specific kinetic energy profile for Problem 9.1 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 (8.48) and (8.49), 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 9.1. 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 9.1 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 9.1, 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 (9.21), that is,

$$\tilde{T}(s) = \frac{2m\left(c_2'(s)\tilde{E}(s) - c_1'(s)\tilde{E}^3(s)\right)}{3c_1(s)\tilde{E}^2(s) + c_2(s)} - c_1(s)\tilde{E}(s)m - \frac{c_2(s)m}{\tilde{E}(s)} - c_3(s)m. \quad (9.36)$$

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 (9.8) and (9.9) 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.

9.3.3 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[37]

$$\frac{\partial}{\partial T}\left[\frac{\mathrm{d}^2}{\mathrm{d}s^2}\left(\frac{\partial H}{\partial T}\right)\right] \leq 0. \tag{9.37}$$

if it is to be part of the optimal trajectory. Differentiating (9.33) 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 (9.21), it follows that

$$\frac{\partial}{\partial T}\left[\frac{\mathrm{d}^2}{\mathrm{d}s^2}\left(\frac{\partial H}{\partial T}\right)\right] = \frac{1}{m}\left(2c_2(s)\tilde{E}^{-3}(s) - \frac{3}{4\sqrt{2}m}\tilde{E}^{-\frac{5}{2}}(s)\lambda_t^*\right). \tag{9.38}$$

Since $\tilde{E}(s)$ satisfies (9.33), it follows that

$$c_1(s) - c_2(s)\tilde{E}^{-2}(s) + \frac{1}{2\sqrt{2}m}\tilde{E}^{-3/2}(s)\lambda_t^* = 0, \tag{9.39}$$

By eliminating λ_t^* from (9.38), and by using equation (9.33), equation (9.38) can be written as:

$$\frac{\partial}{\partial T}\left[\frac{\mathrm{d}^2}{\mathrm{d}s^2}\left(\frac{\partial H}{\partial T}\right)\right] = \frac{1}{mE(s)}\left(2c_2(s)\tilde{E}^{-2}(s) - \frac{3}{4\sqrt{2}m}\tilde{E}^{-\frac{3}{2}}(s)\lambda_t^*\right) \tag{9.40}$$

$$= \frac{1}{2mE(s)}\left(3c_1(s) + c_2(s)\tilde{E}^{-2}(s)\right), \tag{9.41}$$

which is indeed negative because $c_1(s) < 0$ and $c_2(s) < 0$ according to (8.48) and (8.49). Hence, along the singular arcs, the generalized Legendre-Clebsch condition is satisfied, and hence these arcs can be part of the optimal trajectory.

9.3.4 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 9.1 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 9.1.

Theorem 9.1. Let $E^*(s)$ be the optimal specific kinetic energy profile for Problem 9.1 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 (9.32). By the continuity of $\lambda_E^*(s)$, it follows that $\lambda_E^*(\tau) = -m$.

According to equation (9.29), 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 9.1. 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 9.1 narrows down the possible switching combinations of the optimal control T^* for Problem 9.1. The valid switching structures above and below $\tilde{E}$ are illustrated in Fig. 9.1. In contrast, the switching structures in Fig. 9.2 are not optimal.

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.

9.3.5 Optimal Switching Structure Involving State-Constrained Arcs

The previous analysis is valid when the state constraints (9.8) and (9.9) are inactive. In this section we analyze the case when the state either the constraints (9.8) or (9.9) are active on part of the optimal trajectory. When the state constraint (9.8) or (9.9) 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

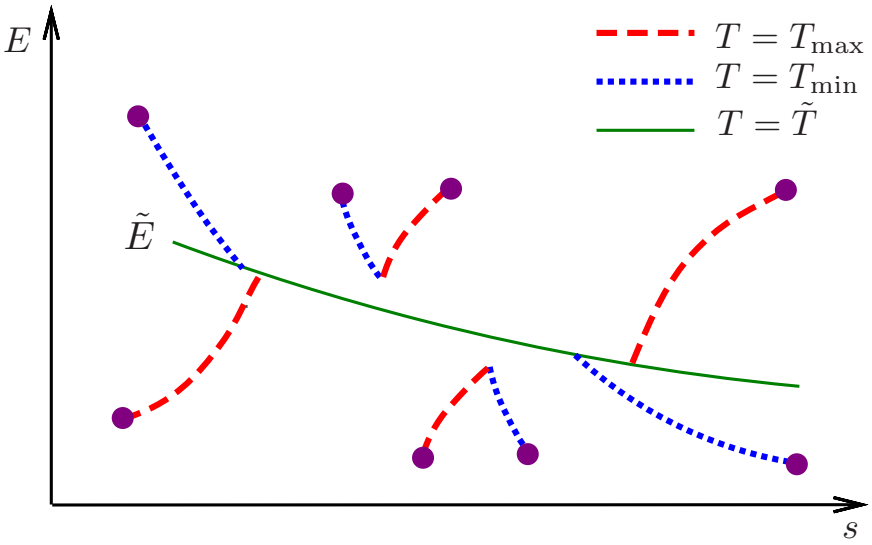


Figure 9.1: Optimal switching structures

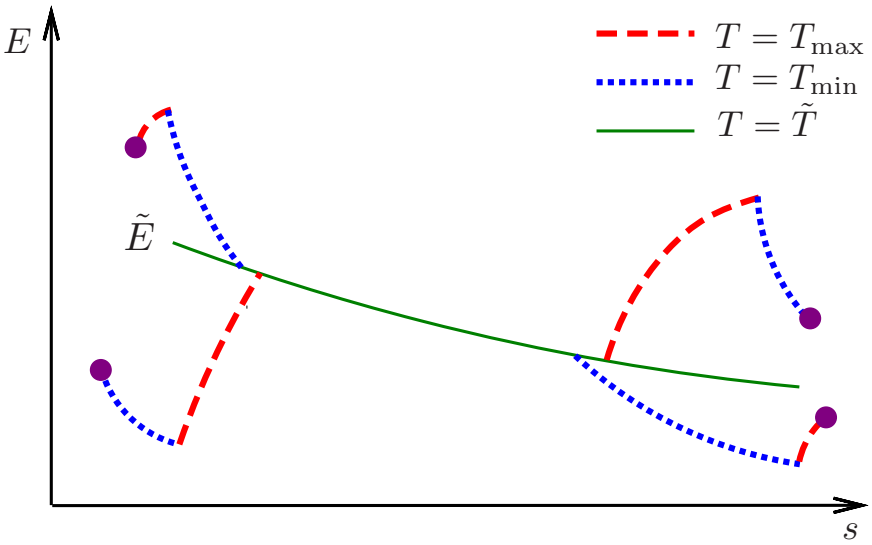


Figure 9.2: Non-optimal switching structures

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. [191]. 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 9.1. 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 9.1, 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 9.1, 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 9.2. Let $E^*(s)$ be the optimal specific kinetic energy solution to Problem 9.1. 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 9.1, 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 9.1, 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 9.1. 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 $E^{*\prime}_U$ 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. □

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 9.3. 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 $g_2(x^*_B(t), u^*_B(t)) \leq 0$ 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] 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.

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 $J(x_A^*, u_A^*) \leq J(x_B^*, u_B^*)$ 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, $J(x_A^*, u_A^*) \geq J(x_B^*, u_B^*)$ 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.

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 9.3 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 9.1 by relaxing the original state constraints (9.23) and (9.24) on certain subintervals.

Problem 9.2 (**Relaxed minimum-energy path-tracking problem with fixed TOA**). Minimize the energy cost (9.20) while subject to constraints (9.21), (9.22), (9.25), (9.26), (9.27), (9.28), and state bounds

$$E(s) - \overline{g}_\Gamma(s) \leq 0, \tag{9.42}$$

$$\underline{g}_\Gamma(s) - E(s) \leq 0. \tag{9.43}$$

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 (9.42) and (9.43) instead of (9.23) and (9.24). 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 9.2. Consider a function $\tilde{E} : [s_0, s_f] \to \mathbb{R}_+$, which is a solution to the ordinary differential equation (9.21) 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 (9.23) and (9.24). 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 $E(s) \le \overline{g}_{\Gamma_U}(s)$ and $E(s) \ge \underline{g}_{\Gamma_L}(s)$ instead of (9.23) and (9.24). 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

$$E(s) = \begin{cases} \min\{\max\{E_{U_r}^*(s), E_U^*(s)\}, \tilde{E}(s)\}, & s \in \Gamma_U, \\ E_U^*(s), & s \in [s_0, s_f] \setminus \Gamma_U. \end{cases} \tag{9.44}$$

By the definition of $E(s)$ and Γ_U, $E(s) \ge E_U^*(s)$ 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 $\underline{g}_w(s) \le E(s) \le \overline{g}_w(s)$ for all $s \in [s_0, s_f]$. Hence, $E(s)$ is a feasible solution to the minimum-time path-tracking problem with constraints (9.23) and (9.24). 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 $E_{U_r}^*(s) \le \overline{g}_{\Gamma_U}(s)$ and $E_U^*(s) \le \overline{g}_w(s) \le \overline{g}_{\Gamma_U}(s)$ are always satisfied, we have $E(s) \le \overline{g}_{\Gamma_U}(s)$ for all $s \in [s_0, s_f]$, and E is a feasible solution to the minimum-time problem with constraint $E(s) \le \overline{g}_{\Gamma_U}(s)$. 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 9.1 is given by the following theorem. Its proof is based on the optimal solution of the relaxed Problem 9.2.

Theorem 9.2. 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

$$E^*(s) = \begin{cases} E_L^*(s), & s \in \Gamma_L, \\ \tilde{E}(s), & s \in [s_0, s_f] \setminus (\Gamma_U \cup \Gamma_L), \\ E_U^*(s), & s \in \Gamma_U \end{cases} \tag{9.45}$$

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 9.1,

Proof. Consider the relaxed Problem 9.1 with the constraints (9.23) and (9.24) 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 9.1 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 9.2, 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 9.2, 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 (9.35) 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 9.1. 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 9.1. Hence, E^* is also the optimal solution to Problem 9.1 by Lemma 9.3, and the proof is complete. □

9.4 An Energy-Optimal Path-Tracking Algorithm

Theorem 9.2 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. [191],

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 (9.45). In this section a numerical algorithm is presented for solving Problem 9.1 by identifying the optimal costate value λ_t^*. This allows the computation of the associated function $\tilde{E}(s)$ from (9.34) and, subsequently, the optimal solution $E^*(s)$ from (9.45). 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 9.1 as in Ref. [191].

2. Compute and store the values of $P\left(E(s), s\right)$ from equation (9.34) 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. [191]. 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[143] 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 (8.47), (9.14), and (9.15), 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 (9.45) 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 (9.45), 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 (9.35). 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.

9.5 Numerical Examples

Next, we validate the proposed energy-optimal tracking algorithm using a three-dimensional landing trajectory, as shown in Fig. 9.3. 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. 9.4.

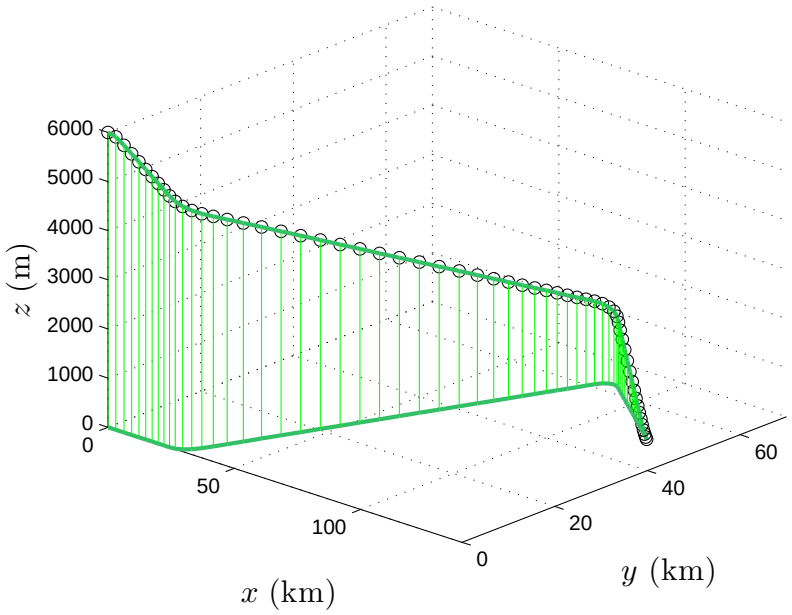


Figure 9.3: 3D Geometric Trajectory.

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 9.5 and 9.6 show the optimal speed profiles for the minimum-energy aircraft

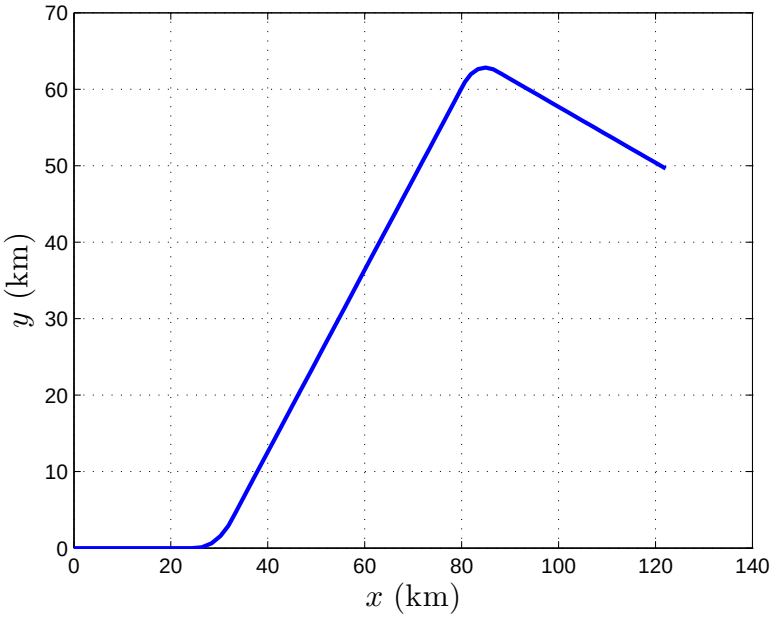


Figure 9.4: X-Y plane projection of the geometric trajectory.

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 maximum-time speed profile can be involved in the minimum-energy solution, together with the corresponding singular arcs. Figures 9.7 and 9.8 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 (9.18) as the cost function. The constraints of the fuel-optimal control problem are identical to those of Problem 9.1. The fuel-optimal control problem was converted into a nonlinear programming problem via direct transcription [32], and solved using the sparse nonlinear optimization software SNOPT [80]. The density function based mesh refinement method in Ref. [192] was used to generate a mesh such that the state bounds (9.8) and (9.9) can be approximated more accurately with a limited number of grid points. The parameters for the computation of η_0 in equation (9.17) were stored in a look-up table, and were provided to the nonlinear optimization solver.

The same four cases shown in Fig. 9.5 ($t_f = 800\,\mathrm{s}$, $1000\,\mathrm{s}$, $1200\,\mathrm{s}$, $1400\,\mathrm{s}$) were solved 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. 9.9 and 9.10. 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. 9.9 and 9.10). 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

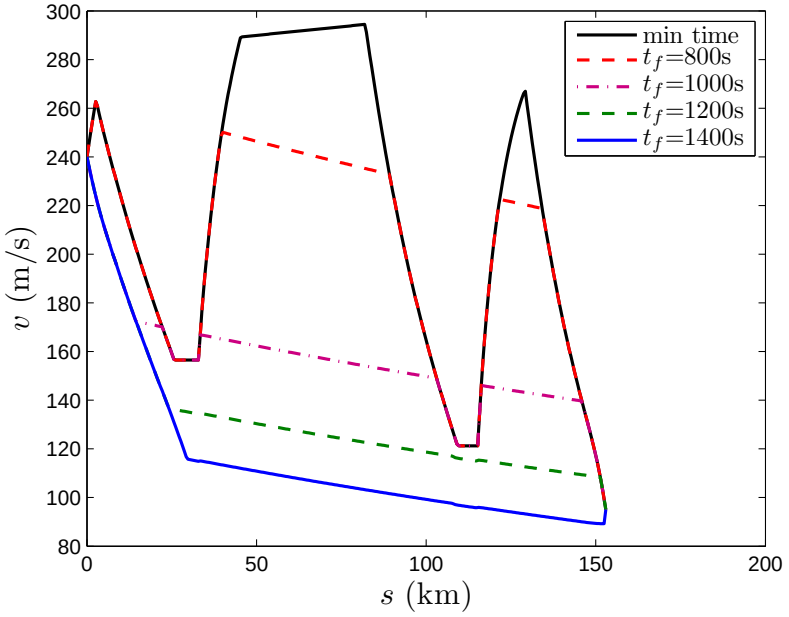


Figure 9.5: Energy-optimal speed profiles with different TOA, path coordinate domain.

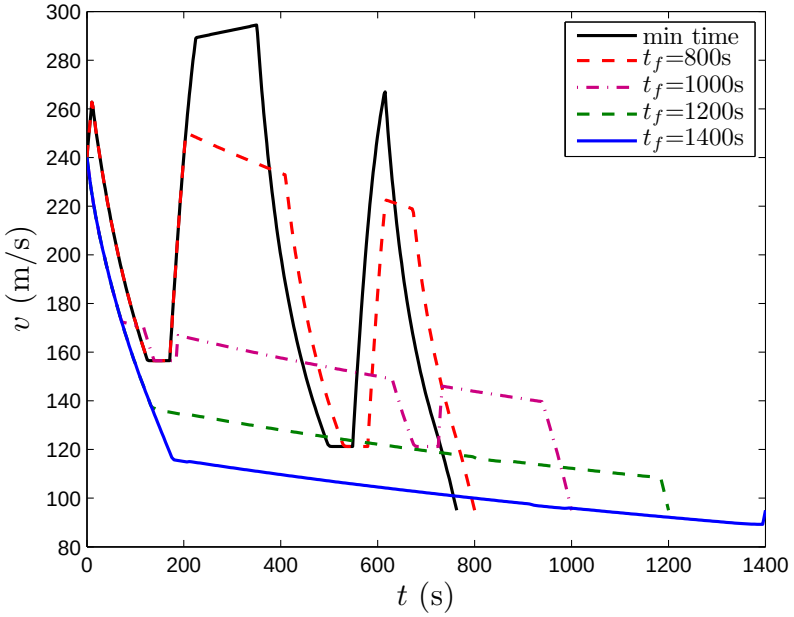


Figure 9.6: Energy-optimal speed profiles with different TOA, time domain.

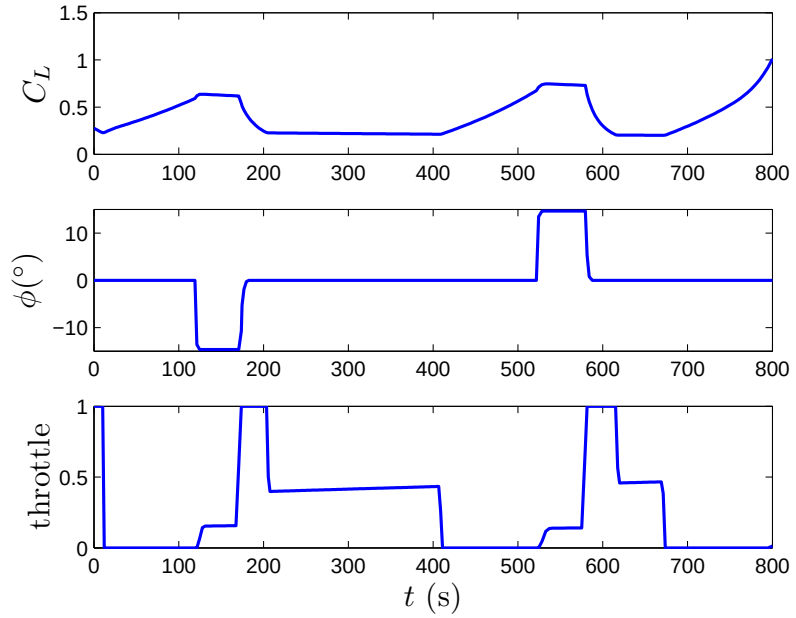


Figure 9.7: Energy-optimal control histories with $t_f = 800\,\mathrm{s}$.

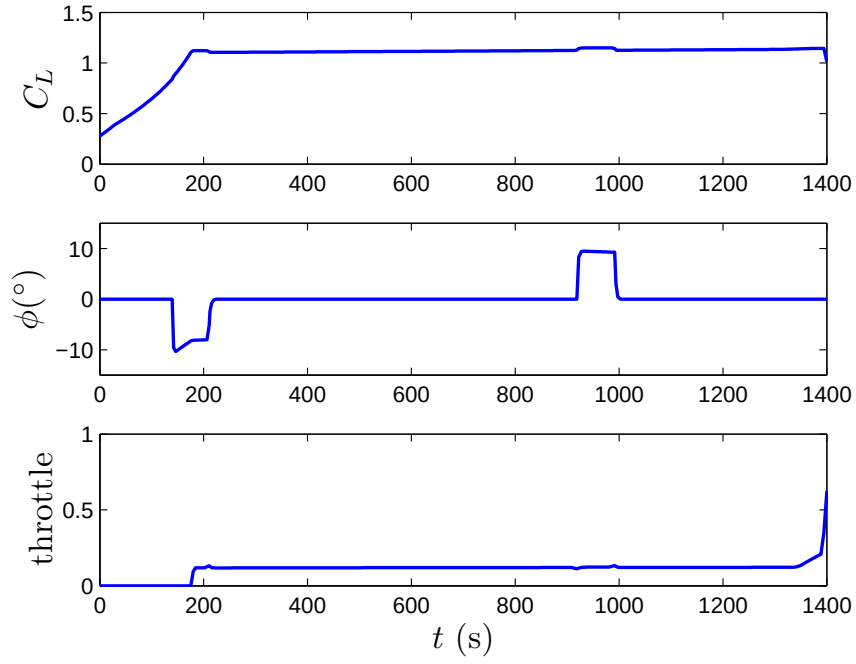


Figure 9.8: Energy-optimal control histories with $t_f = 1400\,\mathrm{s}$.

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.

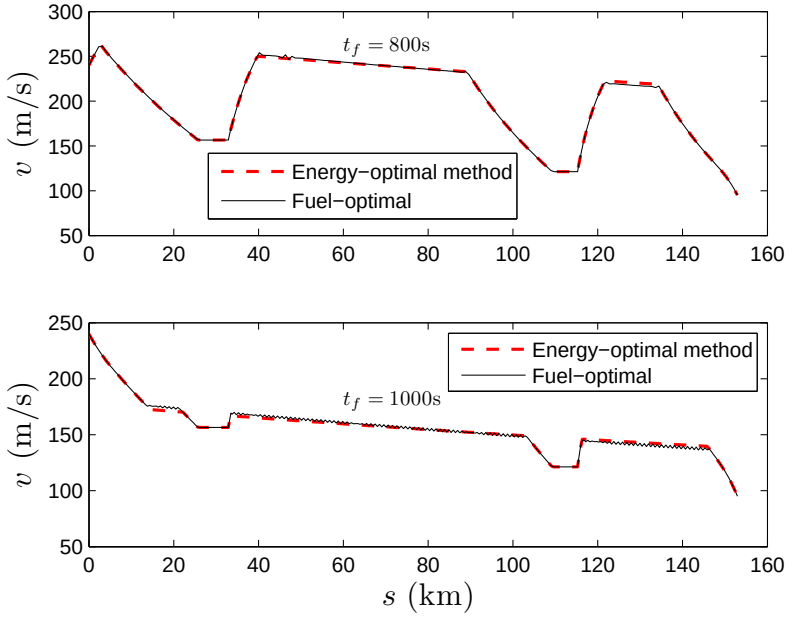


Figure 9.9: Comparison of fuel-optimal and energy-optimal speed profiles, $t_f = 800\,\mathrm{s}$ and $t_f = 1000\,\mathrm{s}$.

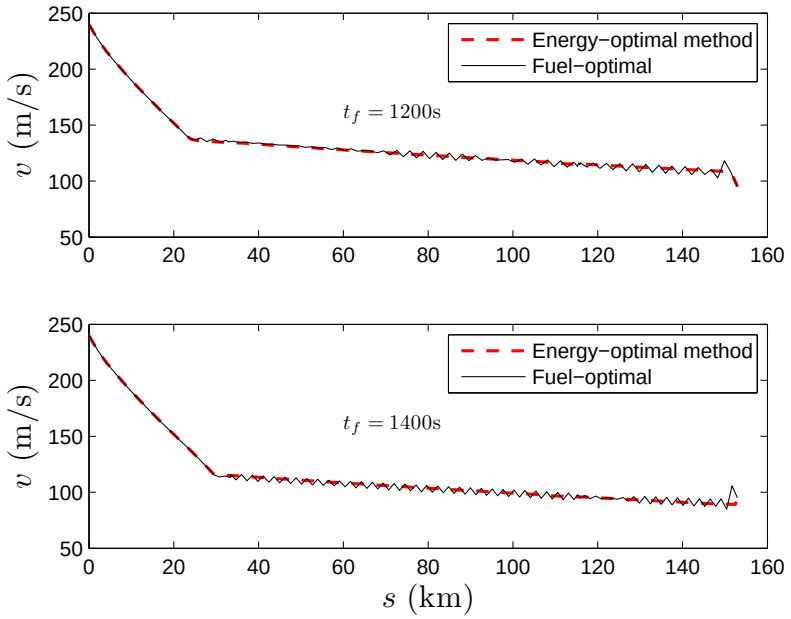


Figure 9.10: Comparison of fuel-optimal and energy-optimal speed profiles, $t_f = 1200\,\mathrm{s}$ and $t_f = 1400\,\mathrm{s}$.

Chapter 10

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.

10.1 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 [43], 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 1.1 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. 10.1. 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.

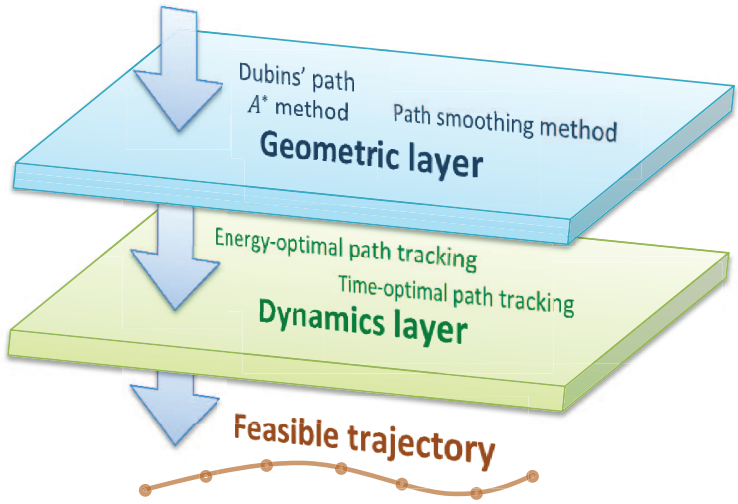


Figure 10.1: Hierarchical approach to feasible trajectory generation.

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 7 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 8 and the fixed final time, energy-optimal aircraft path tracking method introduced in Chapter 9 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 8 and Chapter 9, 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. 10.2.

As shown in Fig. 10.2, 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.

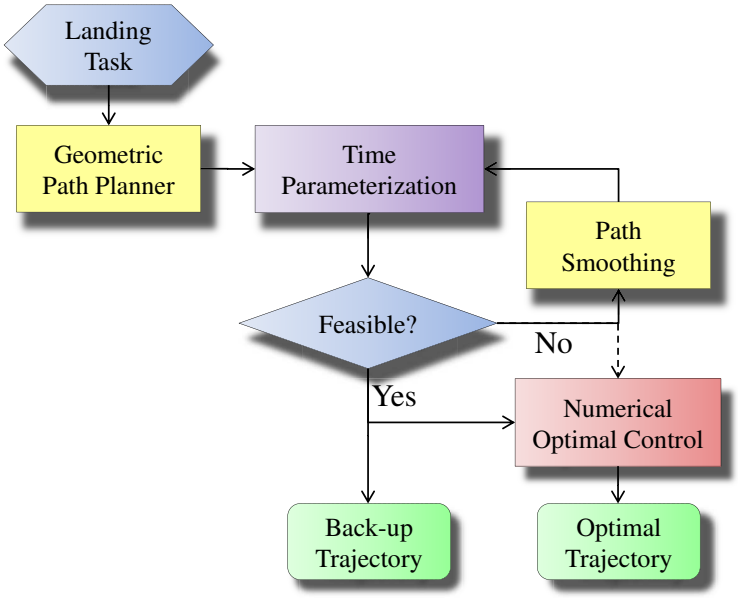


Figure 10.2: Schematic of landing trajectory optimization.

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.

10.2 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 [66], 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

$$x' = \cos\psi \cos\gamma, \tag{10.1}$$
$$y' = \sin\psi \cos\gamma, \tag{10.2}$$
$$z' = \sin\gamma, \tag{10.3}$$
$$\psi' = \frac{u}{R_{\min}(z)}, \tag{10.4}$$

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 [18]).

Problem 10.1. Find the controls u^* and γ^* that steer the system described by Eqs.(10.1)-(10.4) 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 10.1 for any prescribed pair of boundary configurations. A straightforward way to characterize suboptimal solutions for Problem 10.1 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).

10.2.1 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 10.1. 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 10.1 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 10.1 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 [73]

$$\psi' = -\frac{L(C_L, v, z) \sin\phi}{mv^2 \cos\gamma}, \tag{10.5}$$

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

$$R_{\min}(z; \gamma) = \frac{mv(z)^2 \cos\gamma}{L(C_L^{\max}, v(z), z) \sin\phi_{\max}}, \tag{10.6}$$

where $\phi_{\max}$ and $C_L^{\max}$ denote, respectively, the upper bounds on the bank angle and the lift coefficient.

Equation (10.6) 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. (10.4) 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 10.2. 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

$$\mathcal{P} \triangleq \{C^{\pm}(R_0) \circ C^{\mp}(R_m) \circ C^{\pm}(R_f), C^{\pm}(R_0) \circ S \circ C^{\pm}(R_f), C^{\pm}(R_0) \circ S \circ C^{\mp}(R_f)\}, \tag{10.7}$$

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.

10.2.2 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

$$\chi(s) = \begin{cases} -as, & s \in [s_0, s_1], \\ \chi_m, & s \in (s_1, s_2), \\ \chi_e, & s \in [s_2, s_f], \end{cases} \tag{10.8}$$

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

$$z_f = z_0 + \int_0^{s_f} z'(s)\,\mathrm{d}s = z_0 + \int_0^{s_f} \sin(\gamma(s))\,\mathrm{d}s = z_0 + \int_0^{s_f} \chi(s)\,\mathrm{d}s, \tag{10.9}$$

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 (10.9) 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 (10.8) 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. 10.3, in which $\chi(s)$ is represented by the golden dotted curve.

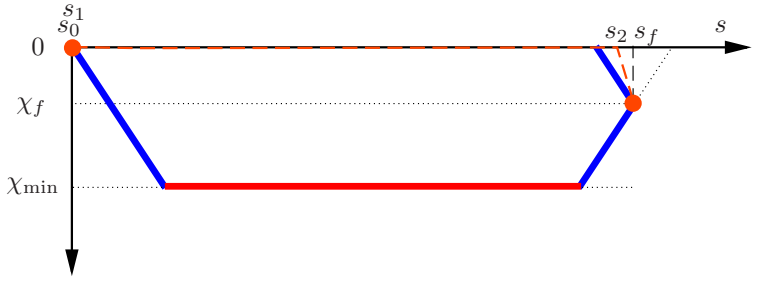


Figure 10.3: Vertical profile generation, Case 1.

Case II: $s_f\chi_f - \chi_f^2/2a \le z_f - z_0 \le -\chi_f^2/2a$. 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 (10.8) 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. 10.4

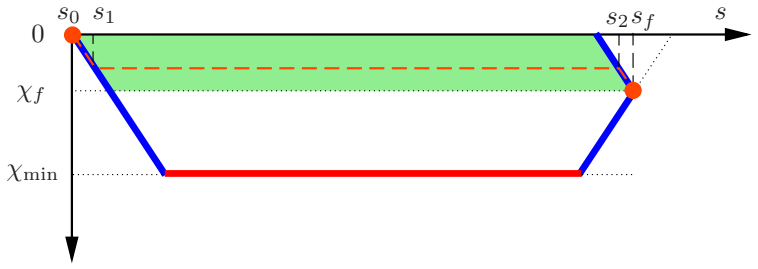


Figure 10.4: Vertical profile generation, Case 2.

Case III: $-\chi_f^2/2a + s_f\chi_{\min} + 3\chi_{\min}^2/a - \chi_f\chi_{\min}/a \le z_f - z_0 < s_f\chi_f - \chi_f^2/2a$. 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 (10.8) 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 (10.8) 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 (10.8) 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. 10.5.

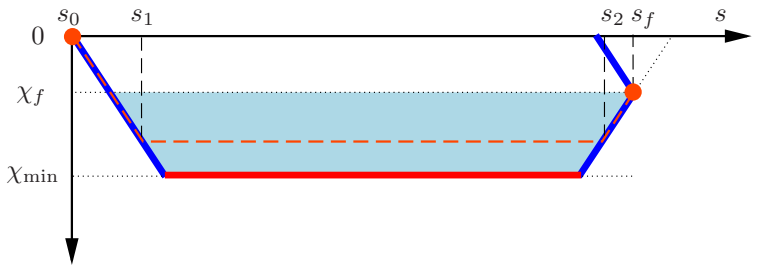


Figure 10.5: Vertical profile generation, Case 3.

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 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

$$\frac{\chi_f^2}{a} + (s_f + ns_n)\chi_{\min} + \frac{\chi_{\min}^2}{2a} - \frac{\chi_f\chi_{\min}}{a} \leq z_f - z_0. \tag{10.10}$$

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. 10.6.

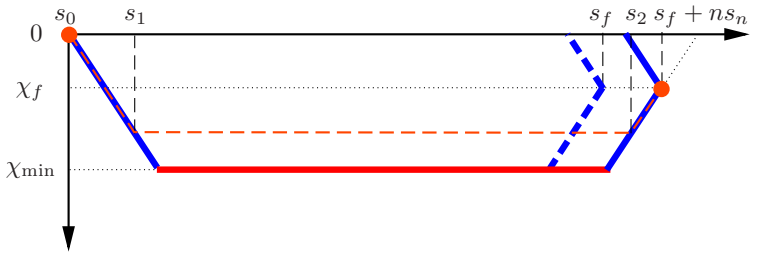


Figure 10.6: Vertical profile generation, Case 4.

After $\chi(s)$ is obtained for all $s \in [0, s_f]$, the vertical profile of the descent is given by the following integral

$$z(s) = z_0 + \int_0^{s_f} \chi(s)\, \mathrm{d}s.$$

10.3 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 10.2 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 8 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 10.1, 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. 10.7 and 10.8. 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. 10.9. 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.

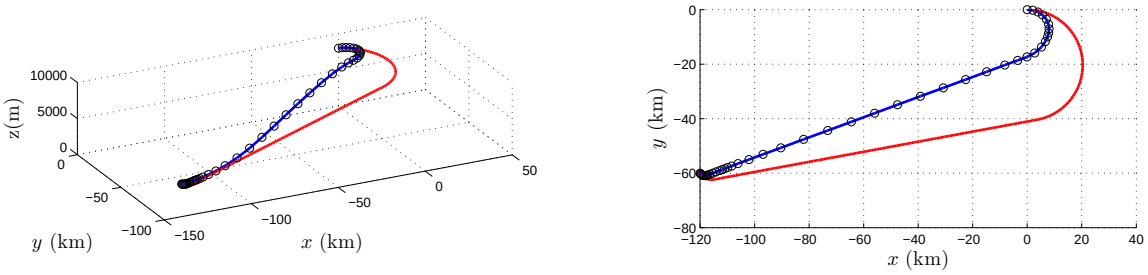


Figure 10.7: Trajectory comparison, case 1.

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 $v_0 = 240\,\text{m/s}$, the final speed $v_f = 95\,\text{m/s}$, the initial path angle $\gamma_0 = 0\,\text{deg}$, the

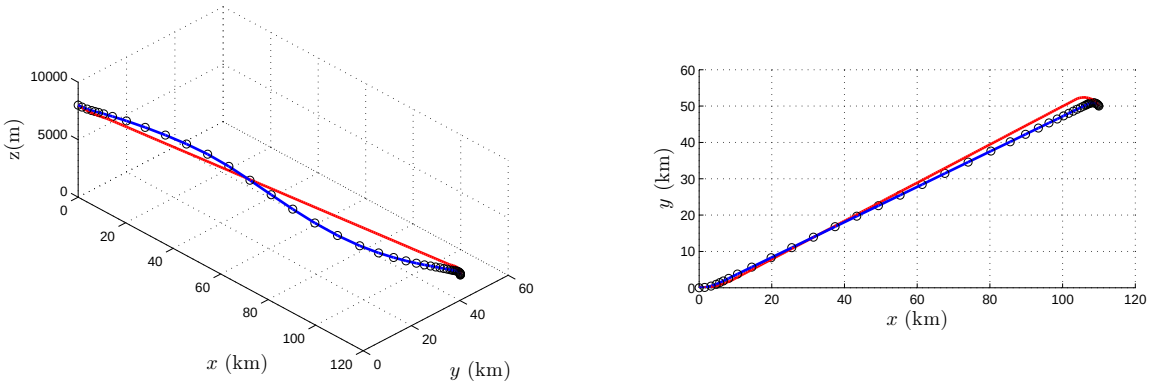


Figure 10.8: Trajectory comparison, case 2.

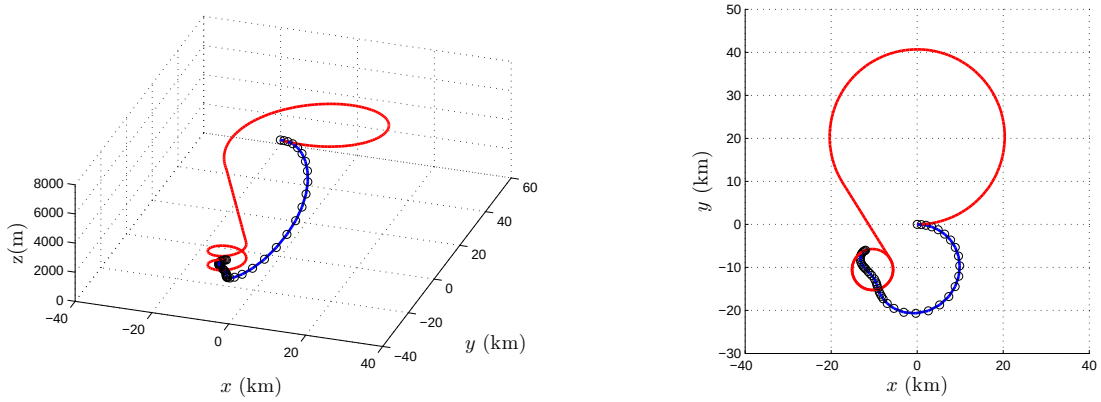


Figure 10.9: Trajectory comparison, case 3.

final path angle $\gamma_f = 0\,\text{deg}$, the initial position $x_0 = 0\,\text{km}$, $y_0 = 0\,\text{km}$, 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 $R_{\max} = 200\,\text{km}$, the runway heading is uniformly distributed in $[0, 2\pi]$, and the initial altitude is uniformly distributed between 6 km and 10 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 10.1. 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:

$$\begin{gathered} x_0 = 0, \quad y_0 = 0, \quad z_f = 0, \quad \psi_0 = 0, \\ \gamma_0 = 0, \quad \gamma_f = 0, \quad v_0 = 240, \quad v_f = 95, \end{gathered}$$

$$\begin{gathered} z_0 \leftarrow \mathrm{U}([6, 10])\text{km}, \qquad \psi_f \leftarrow U([0, 2\pi]), \qquad \theta \leftarrow U([0, 2\pi]), \qquad \varpi \leftarrow U([0, 1]), \\ R = R_{\max}\sqrt{\varpi}, \qquad x_f = x_0 + R_{\max}\cos\theta, \quad y_f = y_0 + R_{\max}\sin\theta, \end{gathered}$$

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 $R_{\max} = 200\,\text{km}$ 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.

Fig. 10.10 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 $t_{\text{tp}} - t_{\text{g}}$ ranges corresponding to the blue bars. As expected, $t_{\text{tp}} > t_{\text{g}}$ 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 $t_{\text{tp}} - t_{\text{g}}$ is 168.2 s, and the mean value of $t_{\text{tp}} - t_{\text{g}}$ is 80.6 s. Hence, in a real emergency landing scenario, it might be worthwhile to use numerical optimization algorithm to further improve the optimality of the TP trajectory.

Fig. 10.11 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

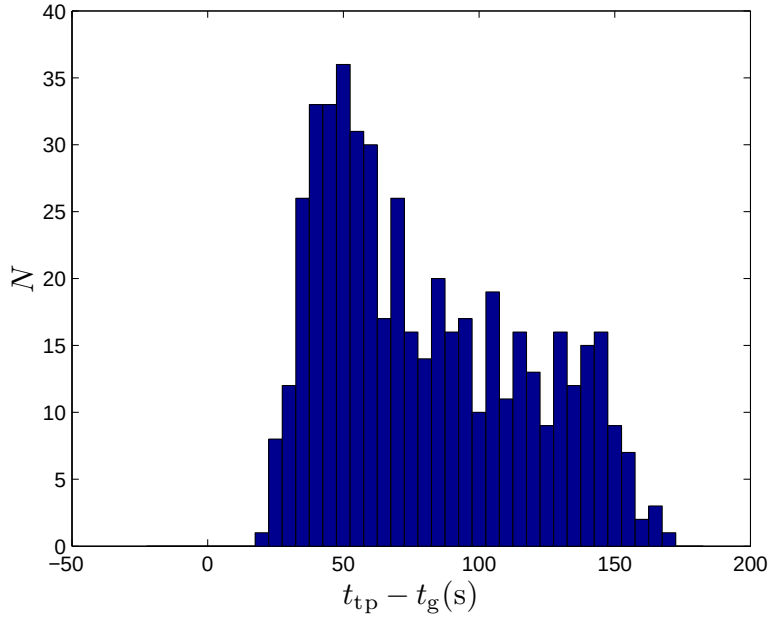


Figure 10.10: Optimality comparison: time-optimal tracking trajectory v.s. numerical optimization with TP initial guess.

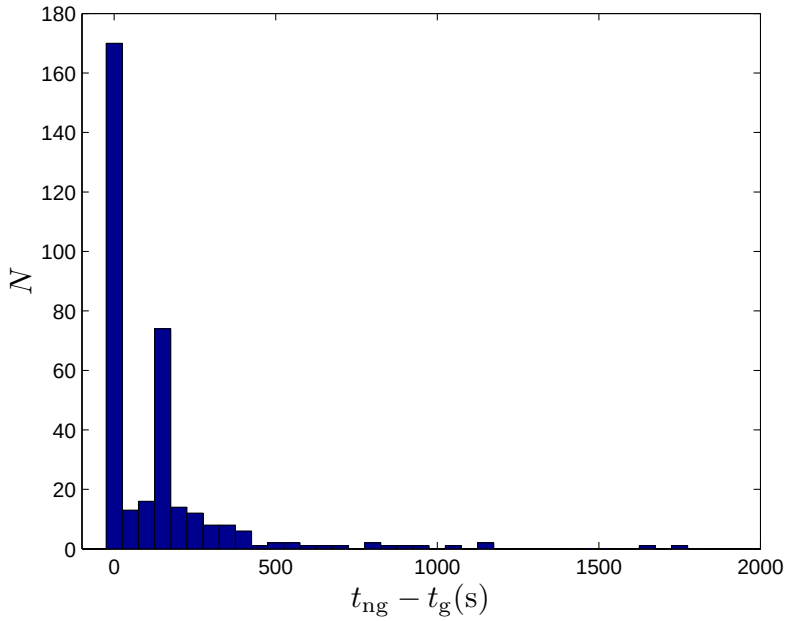


Figure 10.11: Optimality comparison: numerical optimization results, TP initial guesses v.s. affine initial guesses.

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. 10.11, which suggests that a simple affine function initial guess generation scheme is not applicable to trajectory generation for real emergency landing scenario.

Chapter 11

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 10. 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 [8, 5] and [135], 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.

11.0.1 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. 11.1, which is reconstructed using the data in [8, 5]. A three-dimensional plot of the same trajectory is shown in Fig. 11.2. 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. [4].

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. 11.1. 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.

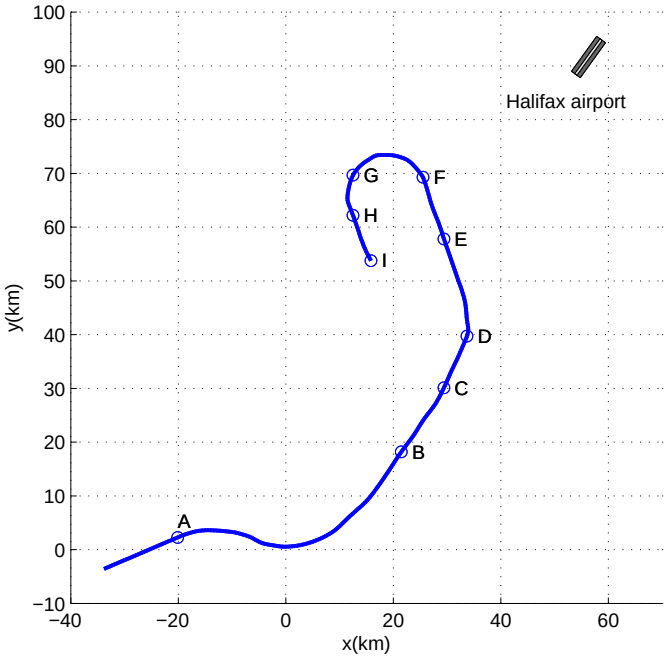


Figure 11.1: Swissair flight 111 trajectory, a top view.

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. 11.3 and 11.4. In these figures, viable trajectories are plotted with green color, and nonviable trajectories are plotted with red color. According 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. 11.5-11.10.

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. 11.11. Note that, by

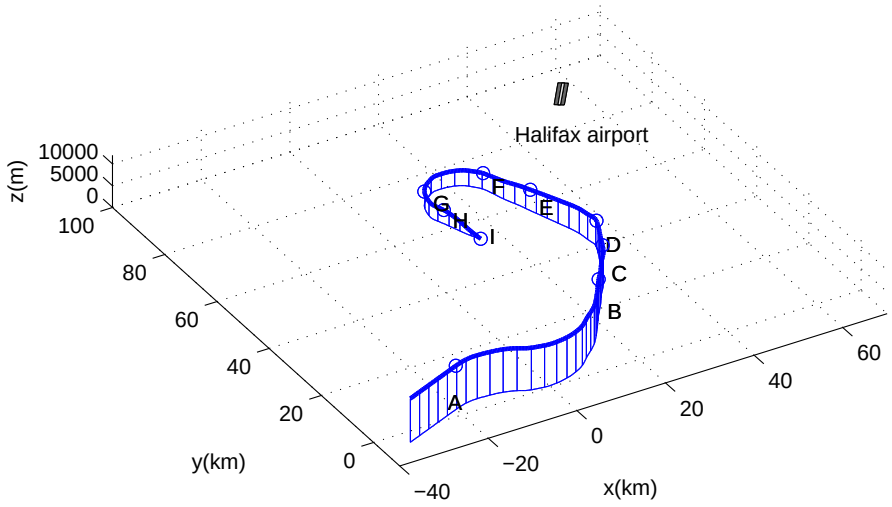


Figure 11.2: Swissair flight 111 trajectory, a bird view.

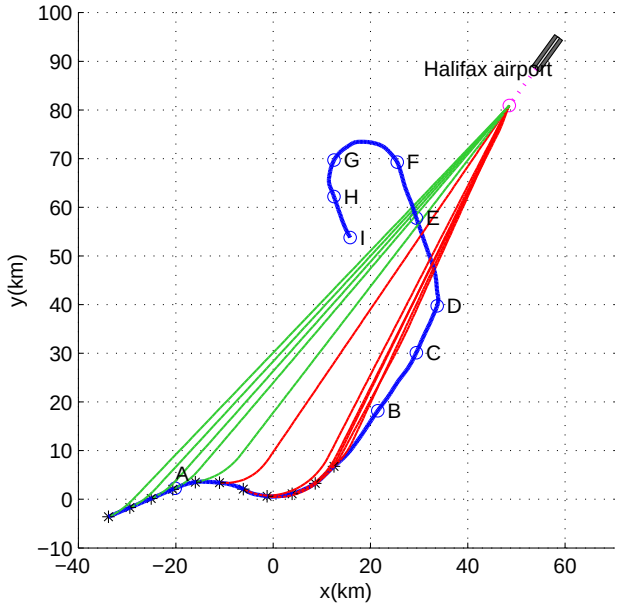


Figure 11.3: Minimum-time trajectories along the Swissair flight 111 trajectory, a top view.

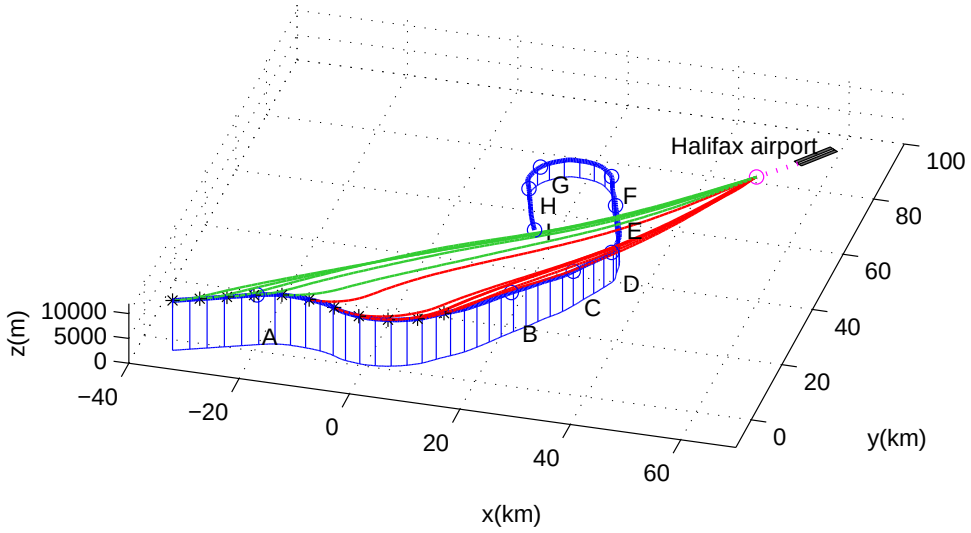


Figure 11.4: Minimum-time trajectories along the Swissair flight 111 trajectory, a bird view.

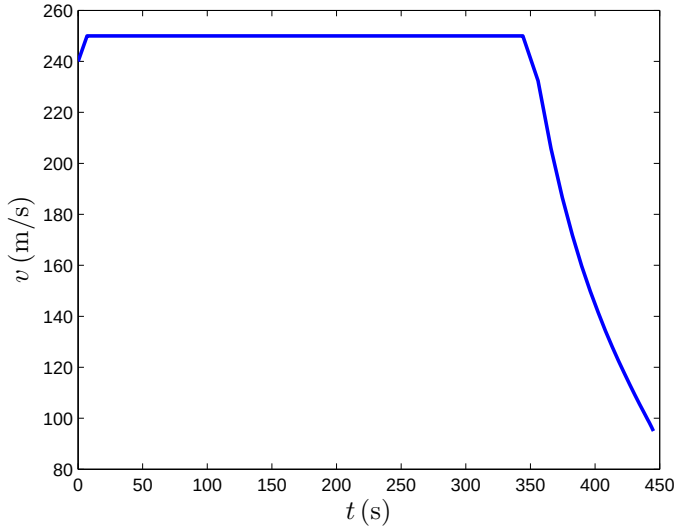


Figure 11.5: Minimum-time speed profile, $t_s = 620$.

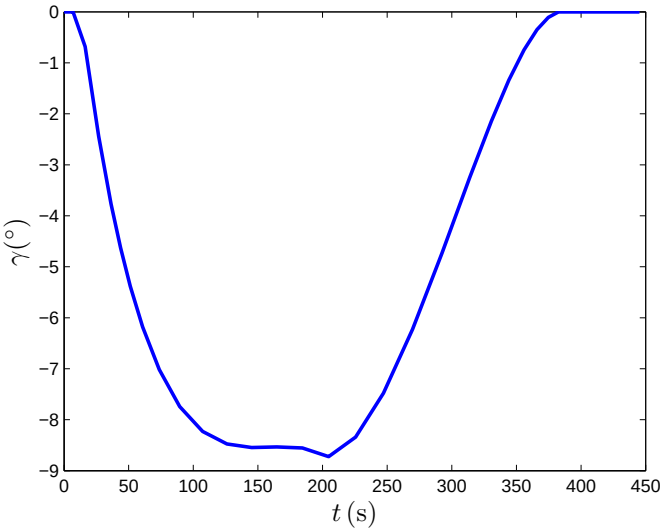


Figure 11.6: Minimum-time path angle profile, $t_s = 620$.

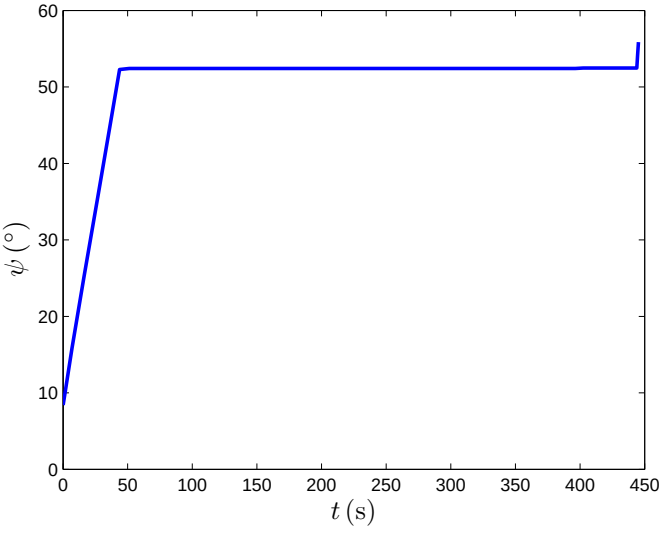


Figure 11.7: Minimum-time heading angle profile, $t_s = 620$.

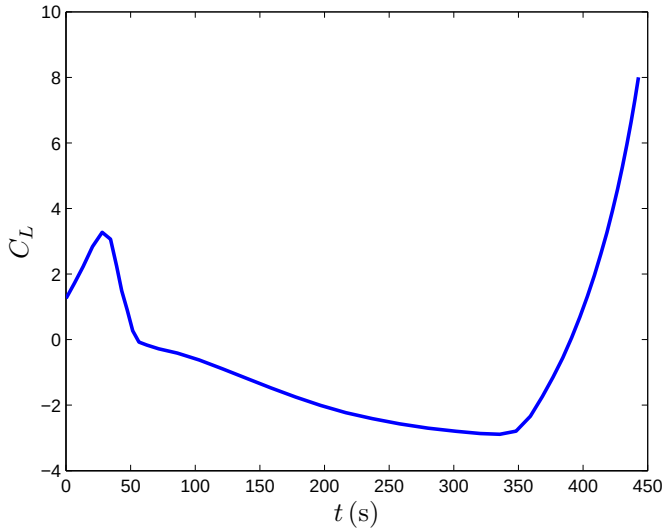


Figure 11.8: Minimum-time lift coefficient profile, $t_s = 620$.

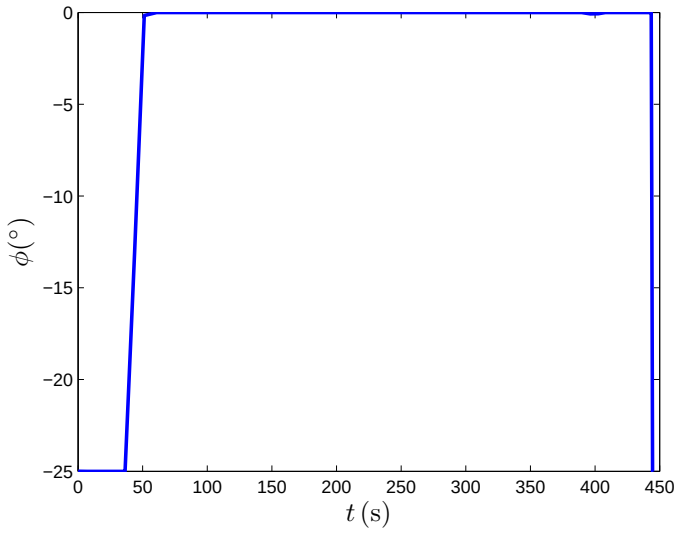


Figure 11.9: Minimum-time bank angle profile, $t_s = 620$.

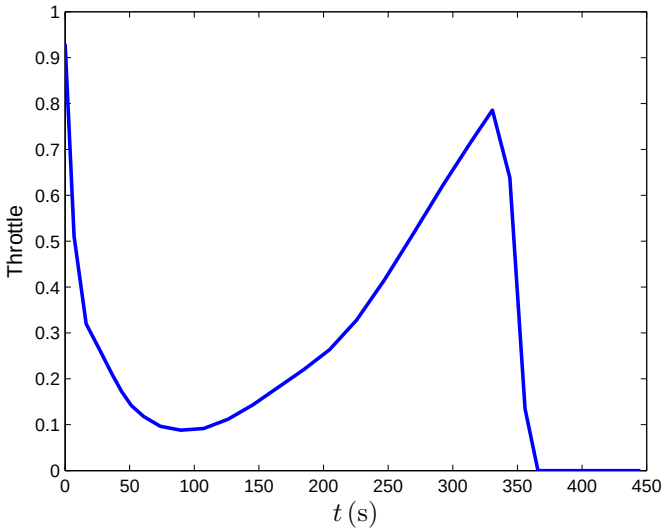


Figure 11.10: Minimum-time throttle profile, $t_s = 620$.

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 trajectory starting from t_{s_1}. Therefore, the excessive time t_{e_2} is smaller than t_{s_1}. As shown in Fig. 11.11, 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 $t_e = 600\,\text{s}$, 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 $t_s = 600\text{s}$, the chance of a safe landing became very slim. The earliest possible landing time is also estimated in [8], which corresponds to $t_s = 683\,\text{s}$ 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 $t_e = 27.6\,\text{s}$, 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.

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.

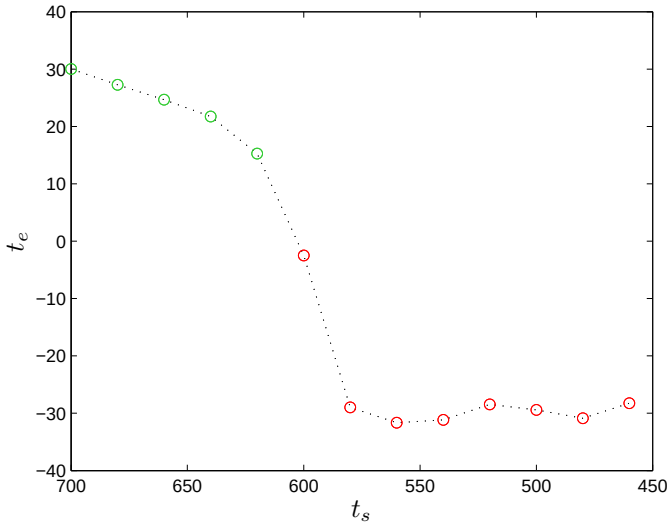


Figure 11.11: The excessive time t_e v.s. the start time t_s of optimal landing trajectories.

11.0.2 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. 11.12. 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. 11.12 (the trajectory data is from [135]). 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. [2].

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. 11.13-11.16. 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. 11.13-11.16, 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. 11.13-11.16 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.

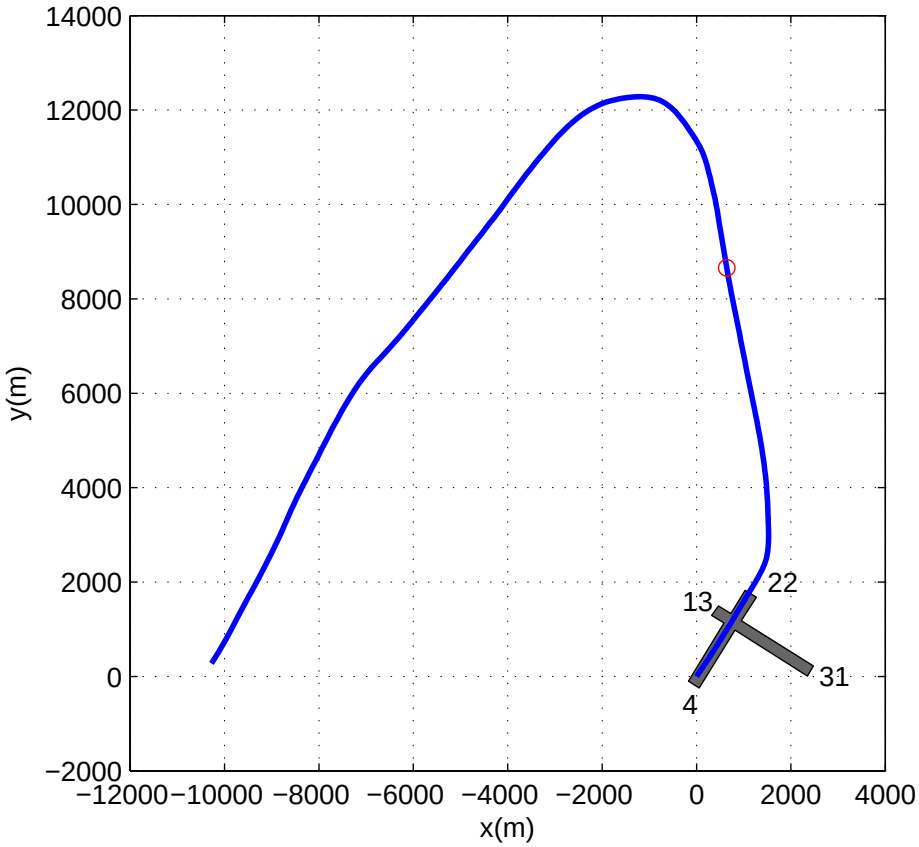


Figure 11.12: US-1549 Hudson River landing trajectory, a top view.

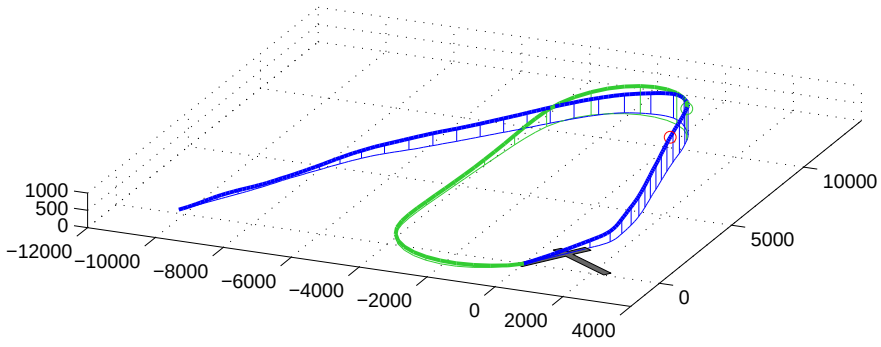


Figure 11.13: US 1549 Runway-4 landing.

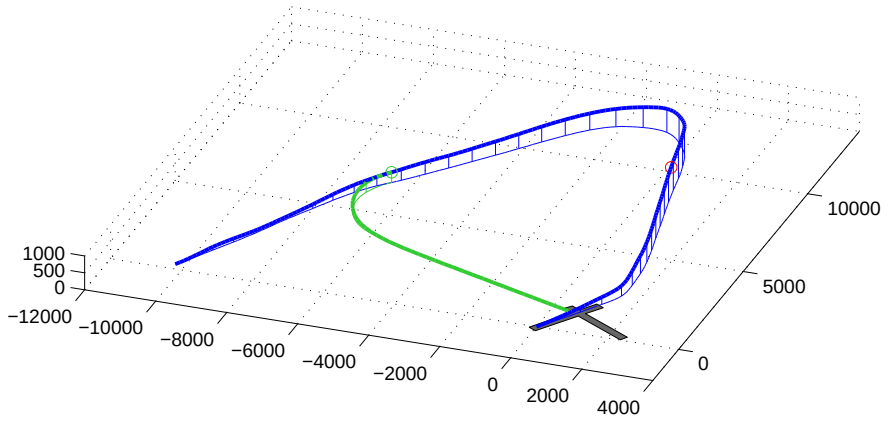


Figure 11.14: US 1549 Runway-31 landing

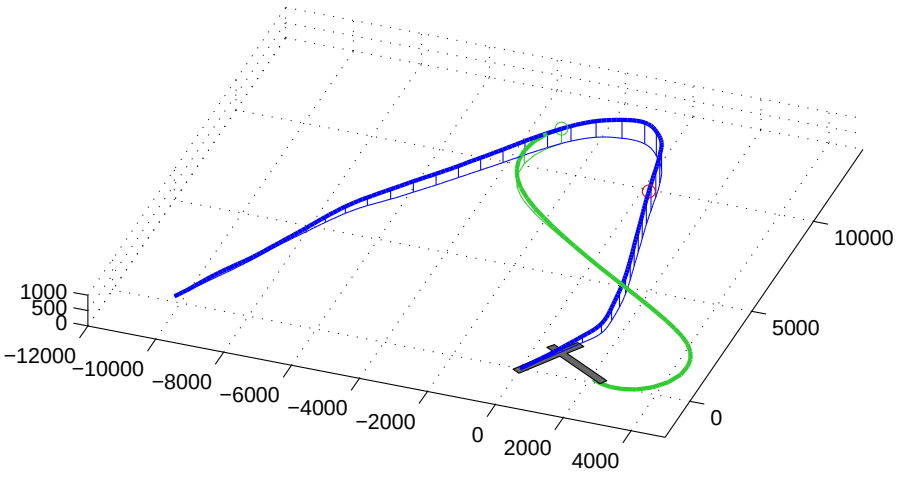


Figure 11.15: US 1549 Runway-13 landing.

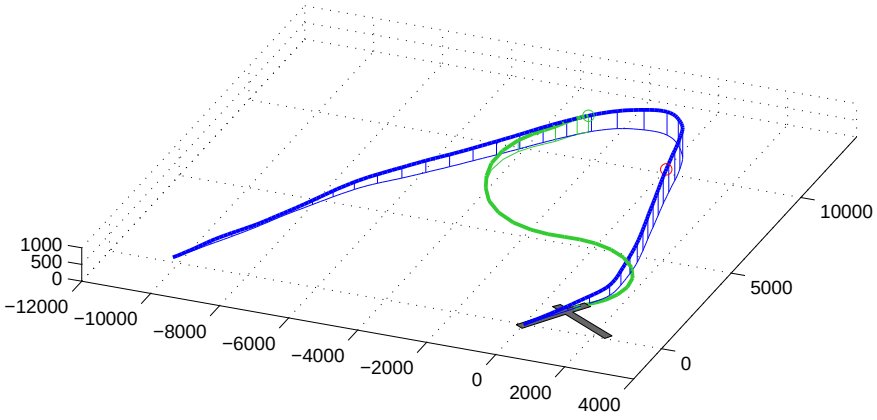


Figure 11.16: US 1549 Runway-22 landing.

References

[1] Merriam Webster Dictionary. http://www.merriam-webster.com/dictionary/.

[2] Airbus 320 aircraft model. `http://www.jsbsim.sourceforge.net`. retrieved Aug., 2011.

[3] Airline data project. `http://web.mit.edu/airlinedata/www/default.html`. retrieved Nov, 2010.

[4] Mcdonnell Douglas MD-11 aircraft model. `http://www.flightgear.org`. retrieved Dec., 2011.

[5] Swissair 111 flightpath map. `http://www.digistar.cl/SR111/srmv3o.html`. retrieved Sep. 2011.

[6] *Control and Dynamic Systems: Advances in Theory and Applications*, chapter Aircraft Maneuver Optimization of Reduced-order Approximation, pages 131–178. Academic Press, Inc., 1973.

[7] In-flight fire leading to collision with water. swissair transport limited mcdonnell douglas md-11 hb-iwf peggys cove, nova scotia 5 nm sw. Technical Report TSB A98H0003, Transportation Safety Board of Canada, Quebec, Canada, 2 September 1998.

[8] SR-111 Investigation Report. Tech. Rep., Transportation Safety Board of Canada, 2003.

[9] Statistical summary of commercial jet airplane accidents worldwide operations 1959-2006. Statistical Summary, Boeing Company, Seattle, WA, July 2006.

[10] Statistical summary of commercial jet airplane accidents worldwide operations 1959-2009. Tech. Rep., The Boeing Company, Seattle, WA, 2010.

[11] I. A. Asnis, A. V. Dmitruk, and N. P. Osmolovskii. Solution of the problem of the energetically optimal control of the motion of a train by the Maximum principle. *Computational Mathematics and Mathematical Physics*, 25(6):37–44, 1985.

[12] Ella Atkins, Igor Alonso Portillo, and Matthew Strube. Emergency flight planning applied to total loss of thrust. *Journal of Aircraft*, 43(4):1205–1216, July–August 2006.

[13] Ella M. Atkins, Igor Alonso Portillo, and Matthew J. Strube. Emergency flight planning applied to total loss of thrust. *Journal of Guidance, Control, and Dynamics*, 43(4):1205–1216, 2006.

[14] Ella M. Atkins, Igor Alonso Portillo, and Matthew J. Strube. Emergency flight planning applied to total loss of thrust. *Journal of Aircraft*, 43(4):1205–1216, July-August 2006.

[15] Ivo Babuska and W. Gui. Basic principles of feedback and adaptive approaches in the finite element method. *Computer Methods in Applied Mechanics and Engineering*, 55(1-2):27–42, Apr. 1986.

[16] Bakolas. *Optimal Steering for Kinematic Vehicles with Applications to Spatially Distributed Agents*. PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, 2011.

[17] E. Bakolas and P. Tsiotras. The asymmetric sinistral/dextral Markov-Dubins problem. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 5649–5654, Shanghai, China, December, 15–18 2009.

[18] E. Bakolas and P. Tsiotras. Optimal synthesis of the asymmetric sinistral/dextral Markov-Dubins problem. *Journal of Optimization Theory and Applications*, 150(2):233–250, 2011.

[19] Efstathios Bakolas and Panagiotis Tsiotras. On-line, kinodynamic trajectory generation through rectangular channels using path and motion primitives. In *47th IEEE Conference on Decision and Control*, pages 3725–3730, Cancun, Mexico, 2008.

[20] Efstathios Bakolas and Panagiotis Tsiotras. 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*, pages 385–390, St. Louis, MO, 2009.

[21] Efstathios Bakolas, Yiming Zhao, and Panagiotis Tsiotras. Initial guess generation for aircraft landing trajectory optimization. In *AIAA Guidance, Navigation, and Control Conference*, Portland, OR, 8–11 August 2011. AIAA 2011-6689.

[22] C. Bearman. Naturalistic decision making. Paper presented at the Centre for Decision Research Seminars (G. Villejoubert, Chair), Leeds University Business School, Leeds, UK., January 2004.

[23] C. R. Bearman. Naturalistic decision making. In *Center for Decision Research Seminar Series*, Leeds, England, January 2004.

[24] G. Beckett, J. A. Mackenzie, A. Ramage, and D. M. Sloan. On the numerical solution of one-dimensional pdes using adaptive methods based on equidistribution. *Journal of Computational Physics*, 167(2):372–392, 2001.

[25] S. Behnke. Local multiresolution path planning. *Lecture Notes in Artificial Intelligence*, 3020:332–43, 2004.

[26] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas. Symbolic planning and control of robot motion. *IEEE Robotics and Automation Magazine*, pages 61 – 70, March 2007.

[27] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G.J. Pappas. Symbolic planning and control of robot motion. *IEEE Robotics Automation Magazine*, 14(1):61–70, Mar. 2007.

[28] S. Bereg and D. Kirkpatrick. Curvature-bounded traversals of narrow corridors. In *Proceedings of the Twenty-first Annual Symposium on Computational Geometry*, pages 278–287, Pisa, Italy, 2005.

[29] Sergey Bereg and David Kirkpatrick. Curvature-bounded traversals of narrow corridors. In *SCG '05: Proceedings of the twenty-first annual symposium on Computational geometry*, pages 278–287, Pisa, Italy, 2005. ACM.

[30] J. T. Betts and W. P. Huffman. 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.

[31] Jhon T. Betts, Stephen L. Campbell, and N. N. Kalla. Initialization of direct transcription optimal control software. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 3802–3807, Maui, Hawaii, Dec. 2003.

[32] John T. Betts. *Practical Methods for Optimal Control using Nonlinear Programming*. PA: SIAM, 2001.

[33] John T. Betts and William P. Huffman. Mesh refinement in direct transcription methods for optimal control. *Optimal Control Applications and Methods*, 19(1):1–21, 1998.

[34] Preethi Bhat, James Kuffner, Seth Goldstein, and Siddhartha Srinivasa. Hierarchical motion planning for self-reconfigurable modular robots. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 886 –891, Beijing, China, Oct. 2006.

[35] J. E. Bobrow, S. Dubowsky, and J. S. Gibson. Time-optimal control of robotic manipulators along specified paths. *The International Journal of Robotics Research*, 4(3):3–17, 1985.

[36] J. D. Boissonnat and X. N. Bui. Accessibility region for a car that only moves forward along optimal paths. Research Note 2181, Institut National de Recherche en Informatique et en Automatique, Sophia-Antipolis, France, 1994.

[37] A. E. Bryson and Y. Ho. *Applied Optimal Control-Optimization, Estimation and Control*. Hemispere Publishing Corporation, Washington, 1975.

[38] X. N. Bui and J. D. Boissonnat. The shortest paths synthesis for nonholonomic robots moving forwards. Research Note 2153, Institut National de Recherche en Informatique et en Automatique, Sophia-Antipolis, France, 1993.

[39] X. N. Bui, P. Souères, J. D. Boissonnat, and J. P. Laumond. Shortest path synthesis for Dubins nonholonomic robot. In *Proceedings of the 11th IEEE Intern. Conf. on Robotics and Automation (ICRA)*, San Diego, California, 1994.

[40] R. Bulirsch, F. Montrone, and H. J. Pesch. Abort landing in the presence of windshear as a minimax optimal control problem, Part I: Necessary conditions. *Journal of Optimization Theory and Applications*, 70(1):1–23, 1991.

[41] R. Bulirsch, F. Montrone, and H. J. Pesch. 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*, 70(2):223–254, 1991.

[42] James W. Burrows. Fuel-optimal aircraft trajectories with fixed arrival times. *Journal of Guidance, Control, and Dynamics*, 6(1):14–19, Jan.-Feb. 1983.

[43] Richard H. Byrd, Frank E. Curtis, and Jorge Nocedal. Infeasibility detection and SQP methods for nonlinear optimization. *SIAM Journal on Optimization*, 20(5):2281–2299, 2010.

[44] B. Cao, G. I. Dodds, and G. W. Irwin. Constrained time-efficient and smooth cubic spline trajectory generation for industrial robots. *IEE Proceedings Control Theory and Applications*, 144(5):467–475, 1997.

[45] L. Carrioli. Unsupervised path planning of many asynchronously self-moving vehicles. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS '91*, pages 555–59, 1991.

[46] M. Cesari. *Optimization - Theory and Applications. Problems with Ordinary Differential Equations*. Springer-Verlag, New York, 1983.

[47] Abhijit Chakravarty. Four-dimensional fuel-optimal guidance in the presence of winds. *Journal of Guidance, Control, and Dynamics*, 8(1):16–22, 1985.

[48] Ted L. Chen and Amy R. Pritchett. Development and evaluation of a cockpit decision-aid for emergency trajectory generation. *Journal of Aircraft*, 38(5):935–943, September-October 2001.

[49] T.L. Chen and A.R. Pritchett. Development and evaluation of a cockpit decision-aid for emergency trajectory generation. *Journal of Aircraft*, 38(5):935 – 943, 2001.

[50] J. T. Cho and B. H. Nam. A study on the fuzzy control navigation and the obstacle avoidance of mobile robot using camera. In *Proceedings of the 2000 IEEE Systems, Man, and Cybernetics Conference*, volume 4, pages 2993 – 2997, 2000.

[51] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. The MIT Press, 2005.

[52] Howie Choset, Kevin Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia Kavraki, and Sebastian Thrun. *Principles of Robot Motion-Theory, Algorithms, and Implementation*. The MIT Press, Cambridge, MA, Jun. 2005.

[53] J. Chuang and N. Ahuja. Path planning using Newtonian potential. In *IEEE Conference on Robotics and Automation*, pages 558–563, Sacramento, CA, 1991.

[54] B. Cipra. Parlez-vous wavelets? In *What's Happening in the Mathematical Sciences*, volume 2. American Mathematical Society, 1994.

[55] D. Constantinescu and E. A. Croft. Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. *Journal of Robotic Systems*, 17(5):233–249, 2000.

[56] R. Cowlagi and P. Tsiotras. Multiresolution path- and motion planning for autonomous agents via wavelet-based cell decompositions. *IEEE Transactions on Systems Man and Cybernetics, Part B: Cybernetics*, 2011. (to appear).

[57] R. V. Cowlagi. *Hierarchical Motion Planning for Autonomous Aerial and Terrestrial Vehicles*. PhD thesis, Georgia Institute of Technology, 2011.

[58] R. V. Cowlagi and P. Tsiotras. Kinematic feasibility guarantees in geometric path planning using history-based transition costs over cell decompositions. In *Proceedings of the 2010 American Control Conference*, pages 5388 – 5393, Baltimore, MD, USA, 30 June– 2 July 2010.

[59] R. V. Cowlagi and P. Tsiotras. Hierarchical motion planning with dynamical feasibility guarantees for mobile robotic vehicles. *IEEE Transactions on Robotics*, 2011. to appear.

[60] Raghvendra V. Cowlagi and Panagiotis Tsiotras. Shortest distance problems in graphs using history-dependent transition costs with application to kinodynamic path planning. In *American Control Conference*, pages 414–419, St. Louis, MO, 2000.

[61] Mary L. Cummings. Automation bias in intelligent time critical decision support systems. In *AIAA 1st Intelligent Systems Technical Conference*, volume 2, pages 557–562, Chicago, IL, United states, 20–22 September 2004. AIAA-2004-6313.

[62] M.L. Cummings. Automation bias in intelligent time critical decision support systems. In *AIAA 1st Intelligent Systems Technical Conference*, September 2004.

[63] M.L. Cummings, K. Meyers, and S.D. Scott. Modified cooper harper evaluation tool for unmanned vehicle displays. In *Unmanned Vehicle Systems Canada Conference*, November 2006.

[64] I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF Lecture Notes, 61, SIAM, 1994.

[65] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516, 1957.

[66] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516, 1957.

[67] K. Ducard, G. Kulling and H. Gering. Evaluation of reduction in the performance of a small uav after an aileron failure for an adaptive guidance system. In *Proceedings of the 2007 American Control Conference*, pages 1793–1798, New York City, N.Y., July, 11-13 2007.

[68] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46 – 57, 1989.

[69] Mica R. Endsley. Measurement of situation awareness in dynamic systems. *Human Factors*, 37(1):65–65, 1995.

[70] Mica R. Endsley. Direct measurement of situation awareness: Validity and use of sagat. In Endsley and Garland [72], chapter 7, pages 147–173.

[71] Mica R. Endsley. Theoretical underpinnings of situation awareness: A critical review. In Endsley and Garland [72], chapter 1, pages 3–32.

[72] Mica R. Endsley and Daniel J. Garland, editors. *Situation Awareness: Analysis and Measurement*. Lawrence Erlbaum Associates, Inc., 2000.

[73] Bernard Etkin. *Dynamics of Atmospheric Flight*. Dover Publications, 2005.

[74] Thierry Fraichard and Alexis Scheuer. From Reeds and Shepp's to continuous-curvature paths. *IEEE Transaction on Robotics*, 20(6):1025–1035, 2004.

[75] Antonio Franco, Damin Rivas, and Alfonso Valenzuela. Minimum-fuel cruise at constant altitude with fixed arrival time. *Journal of Guidance, Control, and Dynamics*, 33(1):280–285, Jan.-Feb. 2010.

[76] Emilio Frazzoli. *Robust Hybrid Control for Autonomous Vehicle Motion Planning*. PhD thesis, Massachusetts Institute of Technology, 2001.

[77] Emilio Frazzoli, Munther A. Dahleh, and Eric Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 20(1):116–129, 2002.

[78] A. Gasparetto and V. Zanotto. A new method for smooth trajectory planning of robot manipulators. *Mechanism and Machine Theory*, 42(4):455–471, Apr. 2007.

[79] R. F. Gilimyanov, A. V. Pesterev, and L. B. Rapoport. Smoothing curvature of trajectories constructed by noisy measurements in path planning problems for wheeled robots. *Journal of Computer and Systems Sciences International*, 47(5):812–819, 2008.

[80] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. Numerical analysis report 97-2, University of California, San Diego, La Jolla, CA, 1997.

[81] Qi Gong, Fariba Fahroo, and I. Michael Ross. Spectral algorithm for pseudospectral methods in optimal control. *Journal of Guidance, Control, and Dynamics*, 31(3):460–471, 2008.

[82] P. G. Holett, P. J. Pudney, and Xuan Vu. Local energy minimization in optimal train control. *Automatica*, 45(11):2692–2698, 2009.

[83] Erik Hollnagel. Contextual control models of cognition. In *Human Reliability Analysis: Context and Control*, pages 159–201. Academic Press, 1993.

[84] Erik Hollnagel. Context, cognition and control. In Yvonne Waern, editor, *Co-operative Process Management*, chapter 4, pages 27–52. Taylor & Francis Ltd., 1998.

[85] Erik Hollnagel. Context, cognition, and control. In Yvonne Wærn, editor, *Co-operative Process Management*, pages 27–52. Taylor & Francis, 1998.

[86] Honeywell. *Boeing 777 Flight Management System Pilot's Guide*, 2001.

[87] J Hoschek. Circular splines. *Computer-Aided Design*, 2(11):611–618, 1992.

[88] P. G. Howlett, I. P. Milroy, and P. J. Pudney. Energy-efficient train control. *Control Engineering Practice*, 2(2):193–200, 1994.

[89] U. Huckenbeck. *Extremal Paths in Graphs*. Akademie Verlag, Berlin, Germany, 1997.

[90] Jens Hugger. 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*, 109(1-2):17–39, 1993.

[91] Human Performance Reserach Group.

[92] J. Y. Hwang, J. S. Kim, S. S. Lim, and K. H. Park. A fast path planning by path graph optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 33(1):121–127, January 2003.

[93] J. Ilkyun, J. Seewong, and K. Youngouk. Mobile robot navigation using difference of wavelet SIFT. In *Proceedings of the 2009 Second International Conference on Machine Vision*, pages 286 – 292, 2009.

[94] C. A. Ippolito and A. R. Pritchett. Software architecture for a reconfigurable flight simulator. In *AIAA Modeling and Simulation Technologies Conference*, Denver, CO, 14–17 August 2000. AIAA-2000-4501.

[95] Michael R. Jackson, Yiyuan Zhao, and Rhonda A. Slattery. Sensitivity of trajectory prediction in air traffic management. *Journal of Guidance, Control, and Dynamics*, 22(2):219–228, 1999.

[96] Marianne Jacobson and Ulf T. Ringertz. Airspace constraints in aircraft emission trajectory optimization. *Journal of Aircraft*, 47(4):1256–1265, Jul.-Aug. 2010.

[97] S. Jain and P. Tsiotras. Trajectory optimization using multiresolution techniques. *Journal of Guidance, Control, and Dynamics*, 31(5):1424–1436, 2008.

[98] S. Jain and P. Tsiotras. Trajectory optimization using multiresolution techniques. *Journal of Guidance, Control, and Dynamics*, 31(5):1424–1436, 2008.

[99] Sachin Jain. *Multiresolution strategies for the numerical solution of optimal control problems*. PhD thesis, Georgia Institute of Technology, Mar 2008.

[100] G. Johannsen and W.B. Rouse. Mathematical concepts for modeling human behavior in complex man-machine systems. *Human Factors*, 21(6):733–47, December 1979.

[101] Kip E. Johnson, James K. Kuchar, and Charles M. Oman. Experimental study of automation to support time-critical replanning decisions. *Human Factors and Ergonomics Society Annual Meeting Proceedings*, 46(5):337–341, 2002.

[102] D. Jung. *Hierarchical Path Planning and Control of a Small Fixed-Wing UAV: Theory and Experimental Validation*. Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, December 2007.

[103] Dongwon Jung and Panagiotis Tsiotras. 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.

[104] S. Kambhampati and L. S. Davis. Multiresolution path planning for mobile robots. *IEEE Journal of Robotics and Automation*, RA-2(3):135–45, September 1986.

[105] Janne Karelahti, Kai Virtanen, and John Öström. Automated generation of realistic near-optimal aircraft trajectories. *Journal of Guidance, Control, and Dynamics*, 31(3):674–688, 2008.

[106] Janne Karelahti, Kai Virtanen, and John Öström. Automated generation of realistic near-optimal aircraft trajectories. *Journal of Guidance, Control, and Dynamics*, 31(3):674–688, 2008.

[107] Giora Keinan. Decision making under stress: Scanning of alternatives under controllable and uncontrollable threats. *Journal of Personality and Social Psychology*, 52(3):639–664, March 1987.

[108] Giora Keinan. Decision making under stress: Scanning of alternatives under controllable and uncontrollable threats. *Journal of Personality and Social Psychology*, 52(3):639–644, 1987.

[109] Henry J. Kelley. Flight path optimization with multiple time scales. *AIAA Journal of Aircraft*, 8(4):238–240, 1971.

[110] Eugene Khmelnitsky. On an optimal control problem of train operation. *IEEE Transaction on Automatic Control*, 45(7):1257–1265, Jul. 2000.

[111] C.-T. Kim and J.-J. Lee. Mobile robot navigation using multi-resolution electrostatic potential field. In *32nd Annual Conference of IEEE Industrial Electronics Society, 2005, IECON 2005*, 2005.

[112] Gary A. Klein, Judith Orasanu, and Roberta Calderwood, editors. *Decision Making in Action: Models and Methods*. Ablex Publishing, 1993.

[113] Y. Kuwata and J. P. How. Stable trajectory design for highly constrained environments using receding horizon control. In *Proceedings of the 2004 American Control Conference*, pages 902 – 907, Boston, MA, June 30 – July 2 2004.

[114] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[115] C. Layton, P.J. Smith, and C.E. Mc Coy. Design of a cooperative problem-solving system for en-route flight planning: An empirical evaluation. *Human Factors*, 36(1):94–119, 1994.

[116] Charles Layton, Philip J. Smith, and C. Elaine McCoy. Design of a Cooperative Problem-Solving System for En-Route Flight Planning: An Empirical Evaluation. *Human Factors*, 36(1):94–119, 1994.

[117] A. A. Ligun and V. F. Storchai. On the best choice of nodes for approximation by splines in the metric of l^p. *Mathematicheskie Zametki*, 20(4):611–618, Otc. 1976.

[118] Ping Lu and Bion L. Pierson. Optimal aircraft terrain-following analysis and trajectory generation. *Journal of Guidance, Control, and Dynamics*, 18(3):555–560, 1995.

[119] Ping Lu and Bion L. Pierson. Optimal aircraft terrain-following flight with nonlinear engine dynamics. *Journal of Guidance, Control, and Dynamics*, 19(1):240–242, 1996.

[120] W. L. D. Lui and R. Jarvis. A pure vision-based approach to topological SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3784 – 3791, Taipei, Taiwan, October 18 – 22 2010.

[121] Takashi Maekawa, Tetsuya Noda, Shigefumi Tamura, Tomonori Ozaki, and Kenichiro Machida. Curvature continuous path generation for autonomous vehicle using b-spline curves. *Computer-Aided Design*, 42(4):350–359, 2010.

[122] Jack D. Mattingly, William H. Heiser, and David T. Pratt. *Aircraft Engine Design*. AIAA, 1987.

[123] A. John Maule and Anne C. Edland. The effects of time pressure on human judgement and decision making. In Rob Ranyard, W. Ray Crozier, and Ola Svenson, editors, *Decision Making*, chapter 11. Routledge, London, 1997.

[124] A. John Maule and Anne C. Edland. The effects of time pressure on human judgement and decision making. In Rob Ranyard, W. Ray Crozier, and Ola Svenson, editors, *Decision Making: Cognitive Models and Explanation*, pages 189–204. Routledge, 1997.

[125] A.John Maule, G.Robert J Hockey, and L Bdzola. Effects of time-pressure on decision-making under uncertainty: changes in affective state and information processing strategy. *Acta Psychologica*, 104(3):283 – 301, 2000.

[126] D.S. Meek and D.J. Walton. Approximating smooth planar curves by arc splines. *Journal of Computational and Applied Mathematics*, 59(2):221–231, 1995.

[127] Angelo Miele. *Flight Mechanics, Vol. I: Theory of Flight Paths*. Addison-Wesley, Reading, MA, 1962.

[128] Angelo Miele. Optimal trajectories and guidance trajectories for aircraft flight through windshears. In *Proceedings of the 29th IEEE Conference on Decision and Control*, volume 2, pages 737–746, Honolulu, HI, Dec 1990.

[129] Kathleen L. Mosier and Linda J. Skitka. Human decision makers and automated decision aids: Made for each other? In Raja Parasuraman and Mustapha Mouloua, editors, *Automation and human performance: theory and applications*, chapter 10, page 201. Lawrence Erlbaum Associates, Inc., 1996.

[130] NASA. Earth atmosphere model. `http://www.grc.nasa.gov/WWW/K-12/airplane/atmosmet.html`. retrieved Aug, 2009.

[131] National Transportation Safety Board. Accidents and Accident Rates by NTSB Classification, 1988 - 2007, 14 CFR 121. http://www.ntsb.gov/aviation/Table2.htm, 2008.

[132] National Transportation Safety Board. Accidents and accident rates by ntsb classification, 1991 through 2010, for u.s. air carriers operating under 14 cfr 121, 2010.

[133] Nils J. Nilsson. *Principles of Artificial Intelligence*. San Francisco: Morgan Kaufmann, 1980.

[134] H. Noborio, T. Naniwa, and S. Arimoto. A quadtree-based path planning algorithm for a mobile robot. *Journal of Robotic Systems*, 7(4):555–74, 1990.

[135] John O'Callaghan. Group chairman's aircraft performance study. Tech. Rep., National Transportation Safety Board, Dec. 2009.

[136] D. K. Pai and L.-M. Reissell. Multiresolution rough terrain motion planning. *IEEE Transactions on Robotics and Automation*, 14(1):19–33, February 1998.

[137] Raja Parasuraman and Victor Riley. Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39(2):230–253, 1997.

[138] C. Paulson, S. Ezekiel, and D. Wu. Wavelet-based image registration. In T. H. O'Donnel, M. Blowers, and K. Priddy, editors, *Evolutionary and Bio-Inspired Computation: Theory and Applications IV, Proceedings of the SPIE*, volume 7704, 2010.

[139] Friedrich Pfeiffer and Rainer Johanni. A concept for manipulator trajectory planning. *IEEE Journal of Robotics and Automation*, RA-3(2):115–123, Apr. 1987.

[140] Aurelio Piazzi, Corrado Guarino Lo Bianco, and Massimo Romano. η^3-splines for the smooth path generation of wheeled mobile robots. *IEEE Transaction on Robotics*, 23(5):1089–1095, 2007.

[141] Erion Plaku, Lydia E. Kavraki, and Moshe Y. Vardi. Motion planning with dynamics by a synergistic combination of layers of planning. *IEEE Transactions on Robotics*, 26(3):469–482, 2010.

[142] R. J. Prazenica, A. J. Kurdila, R. C. Sharpley, and J. Evers. Multiresolution and adaptive path planning for maneuver of micro-air-vehicles in urban environments. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1–12, San Francisco, CA, 2005.

[143] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3 edition, August 2007.

[144] A. V. Rao and K. D. Mease. Eigenvector approximate dichotomic basis method for solving hyper-sensitive optimal control problems. *Optimal Control Applications and Methods*, 20(2):59–77, 1999.

[145] R. M. Rao and A. S. Bopardikar. *Wavelet Transforms - Introduction to Theory and Applications*. Addison-Wesley, 1998.

[146] Eran Rippel, Aharon Bar-Gill, and Nahum Shimkin. Fast graph-search algorithms for general aviation flight trajectory generation. *Journal of Guidance, Control, and Dynamics*, 28(4):801–811, July-August 2005.

[147] I. Kh. Sabitov and A. V. Slovesnov. Approximation of plane curves by circular arcs. *Computational Mathematics and Mathematical Physics*, 50(8):1279–1288, 2010.

[148] Alla Safonova and Jarek Rossignac. Compressed piecewise-circular approximations of 3d curves. *Computer-Aided Design*, 35(6):533–547, 2003.

[149] H. Samet. The quadtree and related hierarchical data structures. *Computing Surveys*, 16(2):187–260, June 1984.

[150] Alexis Scheuer and Thierry Fraichard. Continuous-curvature path planning for car-like vehicles. In *IEEE International Conference on Intelligent Robots and Systems*, pages 997–1003, 1997.

[151] Robert L. Schultz. Three-dimensional trajectory optimization for aircraft. *Journal of Guidance, Control, and Dynamics*, 13(6):936–943, 1990.

[152] A.L. Schwartz. *Theory and Implementation of Numerical Methods Based on Runge-Kutta Integration for Solving Optimal Control Problems*. Ph.D. thesis, Berkeley, University of California, 1996.

[153] James A. Sethian. Curavature and the evolution of fronts. *Communications in Mathematical Physics*, 101(4):487–499, 1985.

[154] Hans Seywald and Renjith R. Kumar. Method for automatic costate calculation. *Journal of Guidance, Control, and Dynamics*, 19(6):1252–1261, 1996.

[155] Z. Shiller and Y.-R. Gwo. Dynamic motion planning of autonomous vehicles. *Robotics and Automation, IEEE Transactions on*, 7(2):241 –249, Apr. 1991.

[156] Zvi Shiller. On singular time-optimal control along specified paths. *IEEE Transactions on Robotics and Automation*, 10(4):561–566, Aug. 1994.

[157] Zvi Shiller. Time-energy optmal control of articulated systems with geometric path constraints. In *Proceeding of IEEE International Conference on Robotic and Automation*, pages 2680–2685, San Diego, CA, 1994.

[158] Zvi Shiller and Hsueh-Hen Lu. Computation of path constrained time optimal motions with dynamic singularities. *Journal of Dynamic Systems, Measurement, and Control*, 114(1):34–40, Mar. 1992.

[159] M. Shim, J. Kurtz, and A. Laine. Multi-resolution stereo algorithm via wavelet representations for autonomous navigation. In *Proceedings of the SPIE*, volume 3723, pages 319 – 328, Orlando, FL, April 1999.

[160] Kang. G. Shin and Neil D. McKay. Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, AC-30(6):531–541, Jun. 1985.

[161] Sidney Siegel and N. John Castellan Jr. *Nonparametric Statistics for The Behavioral Sciences*. McGraw-Hill, 2 edition, 1988.

[162] Peter A. Simpson. Naturalistic decision making in aviation environments. Technical report, Defence Science And Technology Organisation Victoria (Australia), Aeronautical And Maritime Research Lab, January 2001.

[163] B. Sinopoli, M. Micheli, G. Donato, and T. J. Koo. Vision based navigation for an unmanned aerial vehicle. In *Proceedings of 2001 IEEE Conference on Robotics and Automation*, pages 1757–64, 2001.

[164] Rhonda Slattery and Yiyuan Zhao. Trajectory synthesis for air traffic automation. *Journal of Guidance, Control, and Dynamics*, 20(2):232–238, 1997.

[165] J-Y. Souères, P. Fourquet and J-P. Laumond. Set of reachable positions for a car. *IEEE Transactions on Automatic Control*, 39(8):1626–1630, 1994.

[166] Banavar Sridhar, Hok K. Ng, and Neil Y. Chen. Aircraft trajectory optimization and contrails avoidance in the presence of winds. *Journal of Guidance, Control, and Dynamics*, 34(5):1577–1583, Oct. 2011.

[167] M. C. Steinbach, H. G. Bock, and R. W. Longman. Time-optimal extension and retraction of robots: Numerical analysis of the switching structure. *Journal of Optimization Theory and Applications*, 84(3):589–616, Mar. 1995.

[168] Anthony Stentz. Optimal and efficient path planning for partially-known environments. In *IEEE International Conference on Robotics and Automation*, 1994.

[169] Matthew J. Strube, Robert M. Sanner, and Ella M. Atkins. Dynamic flight guidance recalibration after actuator failure. In *AIAA 1st Intelligent Systems Technical Conference*, Chicago, IL, Sep. 20-22 2004.

[170] H. J. Sussmann and G. Tang. Shortest paths for the Reeds-Shepp car: a worked out example of the use of geometric tecnhiques in nonlinear optimal control. Research Note SYCON-91-10, Rutgers University, New Brunswick, NJ, 1991.

[171] J. L. Synge. On the geometry of dynamics. *Philosophical Transactions of the Royal Society of London. Series A*, 226:31–106, 1927.

[172] Yunshen Tang, Ella M. Atkins, and Robert M. Sanner. Emergency flight planning for a generalized transport aircraft with left wing damage. In *AIAA Guidance, Navigation and Control Conference*, Hilton Head, SC, Aug. 2007.

[173] B. Thomaschewski. Dubins' problem for the free terminal direction. *Preprint*, pages 1–14, 2001.

[174] P. Tsiotras and E. Bakolas. A hierarchical on-line path planning scheme using wavelets. In *Proceedings of the European Control Conference*, pages 2806–2812, Kos, Greece, July 2–5 2007.

[175] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, M. N. Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas M. Howard, Sascha Kolski, Alonzo Kelly, Maxim Likhachev, Matt McNaughton, Nick Miller, Kevin Peterson, Brian Pilnick, Raj Rajkumar, Paul Rybski, Bryan Salesky, Young-Woo Seo, Sanjiv Singh, Jarrod Snider, Anthony Stentz, William Whittaker, Ziv Wolkowicki, Jason Ziglar, Hong Bae, Thomas Brown, Daniel Demitrish, Bakhtiar Litkouhi, Jim Nickolaou, Varsha Sadekar, Wende Zhang, Joshua Struble, Michael Taylor, Michael Darms, and Dave Ferguson. Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.

[176] Efstathios Velenis and Panagiotis Tsiotras. Minimum-time travel for a vehicle with acceleration limits: Theoretical analysis and receding horizon implementation. *Journal of Optimization Theory and Applications*, 138(2):275–296, 2008.

[177] Diederik Verscheure, Bram Demeulenaere, Jan Swevers, Joris De Schutter, and Moritz Diehl. Time-optimal path tracking for robots: A convex optimization approach. *IEEE Transaction on Automatic Control*, 54(10):2318–2327, Oct. 2009.

[178] B. J. H. Verwer. A multiresolution workspace, multiresolution configuration space approach to solve the path planning problem. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pages 2107–12, 1990.

[179] Georg Vossen, Volker Rehbock, and Argenes Siburian. Numerical solution methods for singular control with multiple state dependent forms. *Optimization Methods and Software*, 22(4):551–559, 2007.

[180] Rebecca M. Warner. *Applied Statistics: From Bivariate Through Multivariate Techniques*. SAGE Publications, Inc, Thousand Oaks, CA, USA, September 2007.

[181] Robert Watts, Panagiotis Tsiotras, and Eric Johnson. Pilot feedback for an automated planning aid system in the cockpit. In *28th Digital Avionics Systems Conference*, Orlando, FL, Oct. 2009.

[182] Robert Watts, Panagiotis Tsiotras, and Eric N. Johnson. Pilot feedback for an automated planning aid system in the cockpit. In *Digital Avionics Systems Conference*, pages 5.B.21–5.B.213, Orlando, FL, United states, 23–29 October 2009.

[183] L. Wei and T. F. Fwa. Characterizing road roughness by wavelet transform. *Transportation Research Record: Journal of the Transportation Research Board*, 1869:152 – 158, 2004.

[184] T. Wiesemann, J. Schiefele, and J. Bader. Multi-resolution terrain depiction and airport navigation function on an embedded SVS. In J. G. Verly, editor, *Enhanced and Synthetic Vision 2002, Proceedings of the SPIE*, volume 4713, pages 106 – 117, 2002.

[185] S. Winter. Modeling costs of turns in route planning. *GeoInformatica*, 6(4):345 – 361, 2002.

[186] Kwangjin Yang and Salah Sukkarieh. An analytical continuous-curvature path-smoothing algorithm. *IEEE Transactions on Robotics*, 26(3):561–568, Jun. 2010.

[187] M. Yguel, O. Aycard, and C. Laugier. Wavelet occupancy grids: A method for compact map building. In P. Corke and S. Sukkarieh, editors, *Field and Service Robotics, STAR 25*, pages 219 – 230. Springer-Verlag, 2006.

[188] Y. Zhao. *Efficient And Robust Aircraft Landing Trajectory Optimization*. PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, 2011.

[189] Y. J. Zhao. Optimal patterns of glider dynamic soaring. *Optimal Control Applications and Methods*, 24(2):67–89, 2004.

[190] Yiming Zhao and Panagiotis Tsiotras. 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.

[191] Yiming Zhao and Panagiotis Tsiotras. Time-optimal parameterization of geometric path for fixed-wing aircraft. In *Infotech@Aerospace Conference*, AIAA-2010-3352, Atlanta, GA, 2010.

[192] Yiming Zhao and Panagiotis Tsiotras. Density functions for mesh refinement in numerical optimal control. *Journal of Guidance, Control, and Dynamics*, 34(1):271–277, Jan.-Feb. 2011.

[193] Yiming Zhao and Panagiotis Tsiotras. A quadratic programming approach to path smoothing. In *2011 American Control Conference*, pages 5324–5329, San Francisco, CA, USA, Jun. 29-Jul. 01 2011.

[194] Caroline E. Zsambok. Naturalistic decision making: Where are we now? In Caroline E. Zsambok and Gary A. Klein, editors, *Naturalistic Decision Making*. Lawrence Erlbaum Associates, Inc., 1997.

Chapter 12

Student Advising

The following students were either partially of fully supported by this award.

1. Dongwon Jung, post-doctoral student (partially).
2. Yiming Zhao, Ph.D. student, graduated December 2011.
3. Efstathios Bakolas, Ph.D. student, graduated December 2011.
4. Sachin Jain, Ph.D. student (partially), graduated May 2008.
5. Raghvendra Cowlagi, Ph.D. student (partially), graduated May 2011.
6. Claus Christmann, Ph.D. student (partially), expected graduation May 2012.
7. Robert Watts, MS student, graduated May 2010.

The following theses and dissertations have been exclusively or partially supported by this project.

MS1 Robert Watts, Master of Science, thesis title "Development And Evaluation Of An Automated Path Planning Aid," awarded May 2010.

PhD1 Yiming Zhao, Doctor of Philosophy, thesis title "Efficient And Robust Aircraft Landing Trajectory Optimization," December 2011.

PhD2 Efstathios Bakolas, Doctor of Philosophy, thesis title 'Optimal Steering for Kinematic Vehicles with Applications to Spatially Distributed Agents," December 2011.

PhD3 Raghvendra Cowlagi, Doctor of Philosophy, thesis title, "Hierarchical Motion Planning for Autonomous Aerial and Terrestrial Vehicles," May 2011.

Additional Information

The paper "Shortest Distance Problems in Graphs Using History-Dependent Transition Costs with Application to Kinodynamic Path Planning," (co-authored by R. Cowlagi and P. Tsiotras) received the Best Student Paper Award (five finalists) for the 2009 *American Control Conference*, held in St. Louis, MO between June 10–12, 2009.

List of Publications

The following publications describe research that was partially or exclusively supported by this award.

Journal Publications

1. Jain, S. and Tsiotras, P., "Multiresolution-Based Direct Trajectory Optimization," *AIAA Journal of Guidance Control, and Dynamics*, Vol.31, No.5, 2008, pp.1424-1436.
2. Jain, S., and Tsiotras, P., "Sequential Multiresolution Trajectory Optimization Schemes for Problems with Moving Targets," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 2, pp. 488–499, 2009.
3. Bakolas, E. and Tsiotras, P., "The Zermelo-Voronoi Diagram: a Dynamic Partition Problem," *Automatica*, Vol. 46, No. 12, pp. 2059-2067, 2010. doi:10.1016/j.automatica.2010.09.003
4. Cowlagi, R. and Tsiotras, P., "Hierarchical Motion Planning with Kinodynamic Feasibility Guarantees," *IEEE Transactions on Robotics*, (accepted September 2011).
5. Zhao, Y. and Tsiotras, P., "Mesh Refinement for Solving Optimal Control Problems Using Grid Density Functions," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 1, pp. 271–277, 2010, doi:10.2514/1.45852
6. 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, Aug. 2011. doi:10.1007/s10957-011-9841-3
7. Watts, R., Tsiotras, P., Johnson, E., and Feigh, K., "Development and Evaluation of an Automated Path Planning Aid," *AIAA Journal of Aircraft*, (accepted March 2012).
8. Zhao, Y., and Tsiotras, P., "Time-Optimal Path Following for Fixed-Wing Aircraft," *AIAA Journal of Guidance, Control, and Dynamics* (in review, submitted December 2011).
9. Zhao, Y., and Tsiotras, P., "Analysis of Energy-Optimal Aircraft Landing Operation Trajectories," *AIAA Journal of Guidance, Control, and Dynamics* (in review, submitted January 2012).

Conference Publications and Proceedings

1. Jain, S., and Tsiotras, P., "Sequential Multiresolution Trajectory Optimization for Moving Targets," *AIAA Guidance, Navigation, and Control Conference*, Honolulu, HI, Aug. 18-21,

2008, AIAA Paper 2008-6980.

2. Bakolas, E. and Tsiotras, P., "On the Generation of Nearly Optimal, Planar Paths of Bounded Curvature and Curvature Gradient," *American Control Conference*, St. Louis, MO, June 10–12, 2009, pp. 385-390, doi: 10.1109/ACC.2009.5160269.

3. Bakolas, E., and Tsiotras, P., "The Asymmetric Sinistral/Dextral Markov-Dubins Problem," *48th IEEE Conference on Decision and Control*, Shanghai, China, Dec. 16–18, 2009, pp. 5649–5654, doi: 10.1109/CDC.2009.5399851.

4. Bakolas, E. and Tsiotras, P., "Time-Optimal Synthesis for the Zermelo-Markov-Dubins Problem: the Constant Wind Case," *American Control Conference*, Baltimore, MD, June 30–July 2, 2010, pp. 6163–6168.

5. Bakolas, E. and Tsiotras, P., "The Zermelo-Voronoi Diagram: a Dynamic Partition Problem," *American Control Conference*, Baltimore, MD, June 30–July 2, 2010, pp. 1320–1325.

6. Bakolas, E. and Tsiotras, P., "Minimum-Time Paths for a Light Aircraft in the Presence of Regionally-Varying Strong Winds," *Infotech@Aerospace*, Atlanta, GA, April 20–22, 2010, AIAA Paper 2010-3380.

7. Bakolas, E. and Tsiotras, P., "Optimal Pursuit of Moving Targets using Dynamic Voronoi Diagrams," *49th IEEE Conference on Decision and Control*, Atlanta, GA, Dec. 15–17, 2010, pp. 7431–7436.

8. Bakolas, E. and Tsiotras, P., "Optimal Pursuer and Moving Target Assignment using Dynamic Voronoi Diagrams," *American Control Conference*, San Francisco, CA, June 29–July 1, 2011, pp. 5444–5449.

9. Bakolas, E., Zhao, Y. and Tsiotras, P., "Initial Guess Generation for Aircraft Landing Trajectory Optimization," *AIAA Guidance, Navigation, and Control Conference*, Portland, OR, Aug. 8-11, 2011, AIAA Paper 2011-6689.

10. Cowlagi, R. and Tsiotras, P., "Shortest Distance Problems in Graphs Using History-Dependent Transition Costs with Application to Kinodynamic Path Planning," *American Control Conference*, St. Louis, MO, June 10–12, 2009, (best student paper award), pp. 414–419.

11. Cowlagi, R. and Tsiotras, P., "On the Existence and Synthesis of Curvature-Bounded Paths Inside Nonuniform Rectangular Channels," *American Control Conference*, Baltimore, MD, June 30–July 2, 2010, pp. 5382–5387.

12. Cowlagi, R. and Tsiotras, P., "Kinematic Feasibility Guarantees in Geometric Path Planning using History-based Transition Costs over Cell Decompositions," *American Control Conference*, Baltimore, MD, June 30–July 2, 2010, pp. 5388–5393.

13. Cowlagi, R., and Tsiotras, P., "Multi-resolution Path Planning: Theoretical Analysis, Efficient Implementation, and Extensions to Dynamic Environments," *49th IEEE Conference on Decision and Control*, Atlanta, GA, Dec. 15–17, 2010, pp. 1384–1390, doi:10.1109/CDC.2010.5717915.

14. Watts, R., Tsiotras, P. and Johnson, E., "Pilot Feedback for an Automated Planning Aid System in the Cockpit," *28th Digital Avionics Systems Conference*, Orlando, FL, Oct. 25–29, 2009.

15. Zhao, Y. and Tsiotras, P., "A Quadratic Programming Approach to Path Smoothing," *American Control Conference*, San Francisco, CA, June 29–July 1, 2011, pp. 5324–5329.

16. Zhao, Y. and Tsiotras, P., "A Density-Function Based Mesh Refinement Algorithm for Solving Optimal Control Problems," *Infotech@Aerospace Conference*, Seattle, WA, April 6–9, 2009, AIAA Paper 2009-2019 (invited).

17. Zhao, Y. and Tsiotras, P., "Time-Optimal Parameterization of Geometric Path for Fixed-Wing Aircraft," *Infotech@Aerospace*, Atlanta, GA, April 20-22, 2010, AIAA Paper 2010-3352.