# Algorithmic Aspects of Perception-Aware Motion Planning on Resource-Constrained Platforms

by
Igor Spasojevic
B.A., University of Cambridge (2016)
M.Math., University of Cambridge (2016)
Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Autonomous Systems
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
May 2022


Author .....
Department of Aeronautics and Astronautics
May 17, 2022

Certified by .....
Sertac Karaman
Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by .....
John Tsitsiklis
C. J. Lebel Professor of Electrical Engineering and Computer Science
Thesis Committee Member

Certified by .....
Munther Dahleh
W. A. Coolidge Professor of Electrical Engineering and Computer Science
Thesis Committee Member

Accepted by .....
Jonathan P. How
R. C. Maclaurin Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

# Algorithmic Aspects of Perception-Aware Motion Planning on Resource-Constrained Platforms

by

Igor Spasojevic

Submitted to the Department of Aeronautics and Astronautics
on May 17, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Autonomous Systems

## Abstract

Autonomous micro aerial vehicles (MAVs) are becoming an integral tool in numerous applications involving time-critical missions in GPS-denied environments. Due to their small size and lean energy budget, MAVs are often equipped with a camera to aid ego-localization. This introduces at least two fundamental challenges. First, cameras are of little use for state estimation if there is an insufficient quantity of visual information in the environment of the robot. Second, MAVs only display a limited amount of onboard computational resources. Should extracting motion estimates require excessive computational effort, in order to prevent fatal crashes, these agents would be confined to such low speeds that their deployment would be of questionable value.

This thesis studies algorithmic aspects of the question: "How quickly can a vision-driven MAV traverse a given path, while maintaining accurate state estimates at all times?" We seek tractable families of problems involving designing a time-optimal open-loop sequence of controls for a MAV subject to both actuation as well as perception constraints that allow the robot leverage its onboard camera for accurate state estimation. Prior work has either focused on asymptotically optimal search-based approaches which are challenging to implement in real time, or fast local-optimization-based methods with no guarantees on global constraint satisfaction, stability, or optimality.

We present three contributions. First, we extend optimality guarantees of a robust, computationally efficient algorithm for the time-optimal path parametrization problem. Second, we demonstrate the convexity of a general family of perception constraints which require a quadrotor to maintain a sufficient amount of information within field of view of its forward-facing onboard camera. Third, we devise computationally efficient algorithms for guiding the visual attention of a fully-actuated multirotor to traverse a path in minimum time while keeping the computational burden of extracting incremental motion estimates below a set threshold. Together, these contributions serve as stepping stones towards allowing MAVs execute missions autonomously at operational speeds.

Thesis Supervisor: Sertac Karaman
Title: Associate Professor of Aeronautics and Astronautics

# Acknowledgments

Firstly, I would like to extend my deepest gratitude to my advisor Sertac Karaman, not least for always encouraging me to pursue projects I found interesting, and in the way I saw fit. Furthermore, I would like to thank Munther Dahleh and John Tsitsiklis for incisive comments that ultimately led to what I believe is a significant clarification of this thesis. I was also fortunate to have had numerous discussions about robotics with Kaveh Fathian and Vasileios Tzoumas - I found a lot of inspiration in their work.

On the logistic side, none of this would have been possible without the AeroAstro and LIDS staff ensuring things go smoothly from start to finish. I would like to thank Beth Marois and Beata Shuster for sending reminders and instructions about important course milestones, and for promptly answering the myriad of emails I had sent them. On numerous occasions, Brian Jones kindly helped me with tech-related issues, including setting up a hybrid thesis defense in the wake of a difficult COVID-19 period. And, finally on this note, I would like to offer a big thank you to Jin Gao and Francisco Jaimes for helping out with various organizational tasks, ranging from finding available seminar rooms at short notice to getting lab equipment.

I have naturally had the pleasure of meeting many friends and colleagues at MIT and beyond. For a start, a big shoutout to my talented collaborators Varun Murali and Winter Guerra is in order. Same goes for the numerous folks affiliated with AeroAstro: Bai Liu, Soumya Sudhakar, Andres Sisneros, John Aleman, Dave McCoy, Ezra Tal, Gilhyun Ryou, Guilherme Venturelli Cavalheiro, and Thomas Sayre-McCord. It would also be fair to say that I had spent a lot of my time in graduate school in Building 32 - "Stata". Thank you to Elaheh Fata, Flora Meng, Jennifer Tang, Aviv Adler, Fangchang Ma, Oscar Mickelin, David Miculescu, Amir Nouripour, Hajir Roozbehani, and Tuhin Sarkar for all the fun times, especially on the eve of a paper or project deadline.

Ultimately, I cannot find the words to express how grateful I am to my family. Without them, I would not be here.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 MAVs: Capabilities and Constraints

Rapid progress in autonomous micro aerial vehicle (MAV) technology has secured its place in numerous application domains such as package and aid delivery [16], precision agriculture [65], infrastructure inspection [91], disaster response [46], and even scientific exploration. Key to realizing the full potential of these vehicles to boost task productivity by flying faster, accurately execute increasingly delicate missions, and extend our reach by venturing into unknown environments without any human oversight, lies in developing algorithms endowing them with ever higher levels of autonomy [36].

Successful operation of autonomous robotic platforms typically entails the synergy of several components, traditionally divided into estimation, planning, and control modules. Each involves manipulating its own copy of the logical core of the system - its state, which comprises a collection variables capturing the necessary and sufficient information required to predict how the system will evolve. The estimation module fuses measurements from various sensors on board the agent to form an estimate of its pose, how fast it is moving, as well as where the obstacles in its environment lie. The planner supplies a nominal trajectory of states the vehicle should follow in order to execute a specified task, while the controller seeks to mitigate any deviation from the plan that might arise due to inherently noisy, or even unmodelled, aspects of the

dynamics of the agent.

Despite unrelenting technological advancements, MAVs are resource-constrained agents [70]. They can only display a limited supply of power as well as a limited amount of computational resources on board. A state-of-the-art quadrotor platform currently cannot execute aggressive maneuvers for periods longer than 15 minutes without having to recharge its batteries. Similarly, graphical processing units (GPUs) as well as field-programmable gate arrays (FPGAs) have only been a recent addition to the hardware stack of cutting-edge robots emerging out of research labs around the world [67]. Such cyber-physical constraints ultimately shape not only the hardware design of MAVs, but the algorithms that allow them to achieve the limits of their true capabilities.

The limited power supply of MAVs influences their assortment of onboard sensors, rendering a pairing of an inertial measurement unit (IMU) and a camera a common design choice. IMUs are interoceptive sensors, measuring the acceleration and angular velocity of the vehicle. Though accurate over short time scales, their inherent noise and hidden time-varying biases imply they cannot be solely relied upon for longer missions. A second, exteroceptive, sensor such as a camera is necessary for accurate state estimation, although leveraging it effectively does come with a set of challenges. Firstly, the camera can only yield motion estimates in the presence of a sufficient quantity of visual cues in the environment. Second, the raw sensory output of the camera consists of billions of pixel array intensities per frame, now commonly captured at rates exceeding 100 Hz. Extracting information from such a high-dimensional, high-frequency stream of data incurs a non-negligible computational burden.

## 1.2 Towards Optimization-Based Autonomy

This thesis deals with algorithms for perception-aware motion planning. In other words, we seek the sequence of *open-loop* controls that optimizes some notion of performance of the agent subject to both its actuation bounds as well as appropriately chosen perception constraints. Before going further, we pause to elaborate on the

broader place of such a family of problems in the field of control, which primarily addresses the task of synthesizing optimal *feedback policies* [4, 5].

Conceptually, any algorithm may be regarded as a rulebook mapping a query input to the desired output. For example, a state estimation algorithm maps sensory readings of the agent to a representation of the set of (statistically) consistent states. Similarly, a control algorithm or a policy maps the state of the agent to suitable actuator inputs. In the earliest days of embedded systems, such maps were represented as hard-coded circuits. However, with advances in electronic hardware, as well as more challenging mission specifications, attention has steadily shifted to optimization-based control and estimation algorithms [31, 32, 50, 63, 64]. Simply put, the modern rulebook replaces the hard-coded circuit with the solution to an optimization problem parametrized by readings of the sensors in the case of the estimator, or the state of the agent in the case of the controller.

Optimization problems arising in implementation of various autonomy modules involve overcoming a number of computational challenges. In particular, such modules rely on algorithms for dynamic decision making, which often involve solving high-dimensional, non-convex problems. The high-dimensional nature typically comes from the need to consider the trajectory of states throughout a range of time points. The non-convexity usually arises due to nonlinear dynamic and measurement models. As a result, these problems are predominantly addressed using local, gradient-descent-based algorithms, which can in general converge only to local and unsatisfactory optima. Furthermore, such algorithms are often not run until convergence, but for several dozen iterations due to time requirements. Also, some problems entail a mixture of both combinatorial and continuous optimization. Common combinatorial problems in robotics involve instances of the travelling salesman, maximum clique, as well as the set over problem. It is somewhat unfortunate that such problems are not only hard to solve but NP-hard to approximate to within a constant factor by polynomial time algorithms. Moreover, the overarching problem of mapping sensory inputs to optimal actions is PSPACE-hard; this is a common phenomenon when contingency plans are involved [56].

In an effort to compute near-optimal decisions in real time, numerous modelling, architectural, and optimization approximations are employed. In terms of architecture, it is common to adopt hierarchical decision making algorithms. For example, in the case of trajectory planning for quadrotors, a typical scenario involves planning angular velocities which are then delegated to a lower level body-rate controller. Sometimes it is advantageous to speed up motion planning using altered dynamic models that are easier to optimize, and yet capture the leading order trade-offs present in the actual dynamics of the vehicle. Likewise, optimization algorithms often employ convex relaxations of certain cost functions in order to speed up convergence to a local optimum.

## 1.3 Setting and Structure of the Thesis

An approximate motion planning method central to this thesis will be the path-velocity decomposition [38]. From planning trajectories for robotic manipulators [6] to robot teams [57], it decomposes the motion planning task of finding the optimal trajectory from start to goal region by first finding a suitable collision-free path in configuration space, and then endowing the resulting path with a time parametrization that is dynamically feasible for the system. The central problem we will address will be the second, time-optimal path parametrization stage. This is of particular relevance either when the path has been supplied by a higher level module of the task planner, or is naturally given due to the obstacle-rich nature of the environment. In the general case, an optimization algorithm would alternate between selecting the path and endowing it with an optimal time parametrization [89].

The underlying theme of this text will involve bringing to bear the aforementioned methodology to the nascent area of perception-aware motion planning. The latter considers the problem of which trajectory should an agent take from start to goal location in order to optimally execute a given task subject to two classes of constraints. The first class involves hard bounds on actuator inputs, as has been studied studied in the fields of motion planning and trajectory optimization since their inception.

The second, perception-aware, class of constraints will be imposed to ensure the agent maintains a steady stream of percepts that enables it maintain accurate state estimates at all times.

To give a stark example, consider a vision-driven quadrotor navigating an indoor environment. To arrive at a specified goal region it has to choose between two trajectories. With oracle-provided state estimates, the first trajectory has a shorter execution time. However, it involves flying through environments with poor ambient lighting, whereas the second involves navigating well-lit, textured regions of space. Opting for the first trajectory, the vehicle would quickly loose its bearings, and not be able to effectively leverage its controller to damp disturbances to stay on track. This simple example motivates us to consider planning trajectories, that in addition to minimizing execution time, also maintain a sufficient quantity of distinctive visual cues within field of view of the camera on board the agent. Furthermore, we design trajectories that ensure the latest visual percepts can be extracted from the high-dimensional, high-frequency data source such as a camera without undue computational effort.

We shall strive to develop algorithms with provable performance guarantees. From a practical standpoint, this ties back to the requirement that all modules forming an autonomy stack have to yield approximately optimal decisions in real time. Formally, this involves guarantees on underlying algorithms both in terms of precison, or bounds on suboptimality, as well as bounds on execution time to allow the agent to perform at operational speeds. It is natural to seek algorithms that come with such desiderata a priori, without requiring excessive resources to validate them experimentally.

### 1.3.1 Contributions

The following is a chapter-by-chapter outline of the novel ideas in this thesis:

1. We first present the asymptotic optimality of a computationally efficient, numerically robust algorithm for the time-optimal path parametrization problem for a broader class of agents than previously known. We provide a novel way to derive an intuitive characterization of its optimum from first principles using

tools from non-smooth analysis. The results from this chapter are instrumental to the development of Chapter 4 (item 3 below).

2. Second, we demonstrate the convexity of a broad family of perception constraints involving minimizing the traversal time of a specified path by a quadrotor with a forward-facing camera. These include maintaining desired informative regions of the environment within the field of view of the robot's camera, as well as determining which subset of given regions allows for fastest execution time under appropriate assumptions on the environment.

3. The third part of the thesis addresses the problem of designing trajectories that ensure a computationally efficient extraction of visual percepts. In particular, it involves solving a continuous-discrete combinatorial optimization problem, for which we propose simple, polynomial-time algorithms and delineate a relevant subclass of motions for which we prove instance-dependent approximation guarantees.

Finally, we conclude with directions for future work both on the theoretical as well as implementational front.

# Chapter 2

# Asymptotic Optimality of a Time Optimal Path Parametrization Algorithm

## 2.1 Introduction

Seminal works on the time-optimal path parametrization problem (TOPP) dealt with planning dynamically feasible trajectories for robotic manipulators. The latter were typically modelled as linked robots actuated by motors at the joints. Actuation constraints took the form of bounds on their velocities and accelerations, as well as torques exerted by the motors. A key insight of [6] was the existence of a one-to-one correspondence between sets of time parametrizations of a path and *speed profiles*, functions specifying the speed of the agent as a function of distance it has traveled along the path. In this way, TOPP was reduced to an equivalent fixed horizon optimal control problem. In particular, arc length, the speed of the agent, and its acceleration took on the roles of the independent parameter (otherwise most commonly time), the state, and control input, respectively. The requirement of following the path was handled automatically in favour of state and parameter-varying constraints on feasible accelerations.

The first successful class of scalable TOPP algorithms, initiated in [6], are now referred to as numerical integration (NI) approaches. NI algorithms consist of two main steps. The first involves calculating the maximum velocity curve (MVC), a function (taking values in extended non-negative reals) specifying for each point along the path, the infimal speed above which there exists no feasible acceleration. The second step is theoretically grounded on Pontryagin's Maximum Principle, which implies that the optimal profile is either in the regime of maximal acceleration or maximal deceleration in the region below the MVC [6,72]. Therefore, this step involves searching for appropriate parameter-speed coordinates of *switch points* where the optimal profile undergoes a change between the two extremal regimes, or potentially, slides along the MVC. NI methods almost exclusively vary in the second step, striving to make it more computationally efficient and numerically robust.

Before listing the various improvements to the method, we describe the route taken by [6], as it forms of the core of later NI approaches. The method proceeds in iterations until reaching the end of the path as follows. At the start of every iteration it maintains three objects: a path parameter $s_{current}$; the optimal profile $b^*_{current}$ for tracking the restricted path $\gamma|_{[0,s_{current}]}$; and finally $v_{max} \triangleq b^*_{current}(s_{current})$. For example, at the start of the very first iteration $s_{current} = 0$, $b^*_{current} \equiv v_{init}$, and $v_{max} = v_{init}$, where $v_{init}$ represents the specified initial speed of the agent. Then, the method forward integrates the profile of maximal acceleration from $(s_{current}, v_{max})$, until first hitting the parameter axis, or the MVC. In the first case, the algorithm terminates prematurely, correctly concluding the path is untraversable. In the second case, the point of intersection with the MVC is denoted by $(s_{next}, v_{temporary})$, where $s_{next} > s_{current}$. At this point, the the method searches for the highest speed $v_{next}$ lying in the interval $[0, v_{temporary}]$ from which the profile of maxima *deceleration* stays below the remainder of the MVC. It updates $b^*_{current}$ by integrating *backward* from $(s_{current}, v_{current})$ the profile of maximum deceleration until hitting $b^*_{current}$, thereby obtaining a new switch point. The iteration concludes by updating $s_{current}$ and $v_{max}$ to $s_{next}$ and $v_{next}$, respectively. The main room for improvement in computational efficiency of the method lies in working around the search for $v_{next}$.

The efficiency of the NI approach was enhanced in [73], leveraging the insight that switch points from maximal acceleration to maximal deceleration can only occur on the MVC. They use the problem data to efficiently calculate the set of candidates for such points which they partition into three classes. The first class of potential switch points are located where the MVC is discontinuous. These typically occur where the specified geometric path has discontinuous curvature. The second class consists of parameter-speed points on the MVC at which the tangent to the MVC exists and is collinear with the unique allowed acceleration. The final set of candidates are *singular points* where the MVC is continuous but not differentiable; at such points there is a whole interval of feasible accelerations due to one of the acceleration constraints being void. The method would proceed to construct *limit curves* (LCs) emanating from aforementioned candidate switch points by integrating forward profiles of maximal acceleration and integrating backward profiles of maximal deceleration until reaching the MVC. The optimal profile was then recovered as the pointwise minimum of the LCs. In spite of reducing the computational complexity of NI methods, an overlooked aspect of this approach was a principled way of handling construction of LCs at singular points.

Selecting the appropriate acceleration at singular points would prove to be a central topic of future improvements and debate in NI algorithms [37, 72]. The matter was settled only recently [61]. With careful analysis, the authors showed that the appropriate value of acceleration at a singular point can be recovered by considering the constraint inducing the singularity in its vicinity. Nonetheless, despite the progress towards ensuring numerical robustness of the NI approach, a major stumbling block for NI algorithms lay in handling direct velocity bounds (not arising solely from the MVC). Some works considered explicitly determining *trap regions*, areas below the bound on maximum speed where no feasible speed profile could venture into without subsequently breaking the speed or acceleration constraints [90]. However the trap regions were determined by a brute force grid search over in the region below the speed bound, which was a time consuming procedure that somewhat defeated the original purpose of a fast algorithm for TOPP.

Another class of approaches to TOPP that bypass issues of numerical instabilities, and which can seamlessly handle direct velocity bounds, is based on convex optimization (CO). Namely, instead of finding the optimal speed profile, these methods recover the optimal *square speed profile*. Although such a reparametrization was discovered early, it saw initial use only as an aid for integration of maximum acceleration and deceleration profiles [59]. Nevertheless, researchers uncovered that such a reparametrization in fact rendered a wide class of TOPP problems convex [82]. Indeed, by reparametrizing the problem in terms of the square speed profile, actuation constraints such as bounds on velocities, accelerations and torques induced convex constraints on the square speed profile. Furthermore, execution time was also a convex function of the latter variable.

The first step of CO approaches involves transforming the continuous TOPP problem into one consisting of a finite set of decision variables. The latter are typically values of the optimal square speed profile at a predetermined set of discretization points along the path. There is small room for variation in how actuation constraints that ought to hold for the whole trajectory translate into constraints on the aforementioned finite set of decision variables, but they are often enforced to hold pointwise at individual discretization points. Regardless of the transcription method, the resulting finite dimensional optimization problem remains convex. Beyond this point, various methods in literature predominantly differ in terms of the efficiency of the particular CO algorithm they employ. Indeed, by exploiting additional structure of the particular problem at hand, some approaches develop faster tailor-made algorithms compared to ones which can handle a broader class of problems.

The CO approach to TOPP was first introduced as a method for optimizing trajectories of a seven degree of freedom robotic manipulator [82]. Furthermore, this work showed the resulting problem of minimizing the execution time of the path, together with other desiderata such as total energy expenditure and smoothness of the trajectory, subject to velocity, acceleration and torque bounds of the joints could be cast as a second order conic program (SOCP). The work of [26] addresses the task of interpolating a sequence of static keyframes for a walking bi-pedal robot by

a dynamically feasible trajectory. It involves connecting the keyframes by a suitable geometric path, which is then endowed with a feasible time parametrization. The author employs a custom trust-region gradient descent method, reducing the corresponding TOPP problem to a sequence of linear programs, for which more efficient solvers exist. This enables the task of computing dynamically feasible trajectories for robots with hundred degrees of freedom in a matter of seconds. The idea was further developed in [51], reducing the original task of TOPP for robotic manipulators to a related linear program. Ultimately, [41] employ a specialized interior point method for solving TOPP problems for a variety of agents, ranging from thrust vectored space vehicles to a model of a four-wheel-drive car. Interior point methods typically work by pushing the convex inequality constraints into the objective function. At every step of the way, the method performs a Newton gradient descent step recovered as the solution to a banded system of KKT equations, which can be extracted in time that scales linearly with the number of discretization points. Nevertheless, the runtime of the latter method is not guaranteed as the number of required descent steps can be large. In fact, [61] note that CO approaches are an order of magnitude slower than NI approaches; what they gain in numerical robustness and generality they loose in computational efficiency.

The most recent class of approaches to TOPP are iterative methods (I). They possesses the speed of NI algorithms, numerical robustness of CO approaches, and, to an extent, the versatility of the latter. There have been two sub-classes of I-methods, differing mainly in the way they were presented. We will first describe one such algorithm, TOPP-RA (TOPP via Reachability Analysis) by [60], which additionally illustrates the workings of others.

The method recovers values of the optimal square speed profile at a specified set of discretization points along the path in a pair of passes. In the first, backward pass, starting from the end of the path, TOPP-RA incrementally calculates at every discretization point the interval of square speeds from which there exists a feasible profile leading to the end of the path. In the second, forward pass, starting from the beginning of the path, TOPP-RA recovers the interval of speeds at every discretiza-

tion point which can additionally be reached by a feasible square speed profile from the start of the path. Ultimately, the method outputs the upper endpoints of such intervals as values of the optimal square speed profile.

There have been several variations and applications of the latter algorithm for solving TOPP problems. The original application of [60] deals with planning trajectories for manipulators and humanoid robots. In similar vein, [12] develop a method using an efficient custom solver for linear programs in two variables arising when incrementally calculating the reachable square speeds at discretization points. They enable optimizing trajectories for manipulators with hundreds of discretization points in less than a millisecond. Relatedly, [13] plan minimum time trajectories for a double integrator point mass model of the car subject to spatially varying bounds on tangential and normal acceleration. The work of [15] solves TOPP for the waiter motion problem, which involves finding the time optimal way to transport a mass on a tray in a way that the force of static friction is sufficient to prevent it from slipping.

However, the issue with I-approaches was that they were only proven optimal for a subclass of problems solved by convex optimization approaches. For example, [13, 60] show optimality in the presence of polytopic acceleration constraints, whereas [12] relax the latter requirement but still require certain monotonicity assumptions, that need not always hold.

## Contributions

In this chapter, we show that the algorithm proposed in [60] is not only numerically robust, and computationally efficient, but it is also asymptotically optimal for all problems amenable to the more computationally-intensive CO approaches. Towards this end, we provide a novel characterization of the solution of the TOPP problem by leveraging tools from non-smooth analysis that have not been previously used in this context to the best of our knowledge. To complement our theoretical results, we also provide empirical evidence of asymptotic optimality of the aforementioned algorithm by comparing its output with a semi-analytic solution. Much of the work in this chapter can be found in the journal paper [74].

## 2.2 Problem Statement

This section gives a mathematical formulation of the time-optimal path parametrization problem. To this end, we elaborate on two key ingredients: the equations of motion of the agent at hand, and ultimately, the specification of the path it has to traverse.

The ensuing framework is rich enough to capture the task for a wide class of actuation-constrained agents with second order dynamics. The discerning characteristic of such vehicles is that the combination of the zeroth, first, and second time derivatives of the trajectory of their configuration directly corresponds to the trajectory of their state (configuration together with its higher order time derivatives) and inputs required to induce them. Examples of such agents are robotic manipulators, the Dubins and kinematic cars, as well as models of certain space vehicles [10]. Throughout the remainder of the chapter, we will consider the time-optimal path parametrization problem for one such agent we will refer to as the *kinematic plane*.

Roughly speaking, the kinematic plane is a direct product of the kinematic car model and the second order integrator along the vertical axis. Its dynamics are given by:

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \\ \dot{v}_{hor} \\ \dot{v}_{ver} \\ \dot{\omega} \end{bmatrix}}_{\dot{q}} = \underbrace{\begin{bmatrix} v_{hor} \cos\theta \\ v_{hor} \sin\theta \\ v_{ver} \\ \omega \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{drift} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ a_{hor} \\ a_{ver} \\ \tau \end{bmatrix}}_{inputs} \tag{2.1}$$

The configuration of the kinematic plane consists of its position $[x, y, z] \in \mathbb{R}^3$, together with its orientation encoded by $\theta \in [0, 2\pi)$, representing the angle the projection of the longitudinal axis of the vehicle forms with the $x$ axis of the world frame. The state of the kinematic plane additionally contains the horizontal speed $v_{hor} \in \mathbb{R}$ along

its longitudinal axis, its vertical speed $v_{ver} \in \mathbb{R}$, as well as its yaw rate $\omega \in \mathbb{R}$. The inputs consist of longitudinal acceleration $a_{hor} \in \mathbb{R}$, vertical acceleration $a_{ver} \in \mathbb{R}$, and slew rate $\tau \in \mathbb{R}$.

In spite of being a simplified model, the kinematic plane retains several appealing properties that capture the actuation capabilities and limitations of a typical aerial vehicle. First, both its orientation and position behave like second order integrators, much like Euclidean counterparts of the two corresponding components of pose in general rigid body dynamics. Second, the model can succinctly represent a range of different actuation constraints including bounds on (components as well as magnitudes of) speed, acceleration, yaw, and slew rate. Moreover, we will soon see that modelling bounds on combinations of, say, acceleration and slew rate comes at no extra cost. The latter aspect is particularly powerful in situations where a given budget of control authority must be optimally allocated in order to effectively control translational and rotational motion of the actual vehicle.

The second ingredient in the problem definition includes the path specification. In particular, a path is a smooth curve $\gamma : [0, S_{end}] \to \mathbb{R}$ the center of mass of the agent ought to traverse in minimal time while respecting stipulated actuation constraints. The two requirements we impose on $\gamma$ are: regularity, meaning $\gamma'(s) \neq 0 \, \forall s \in [0, S_{end}]$; and smoothness, namely $\gamma(\cdot) \in C^2$ is a twice continuously differentiable function. For our running example of the kinematic plane, we will additionally assume we are given a generic path in that its projection onto the horizontal world plane is a regular curve.

### 2.2.1 Square Speed Profile

The key decision variable in the problem is the square speed profile:

$$\begin{aligned} h : [0, S_{end}] &\to [0, \infty) \\ h : s &\mapsto \left( \frac{ds}{dt}(s) \right)^2 \end{aligned} \tag{2.2}$$

the function giving the speed of the agent as a function of distance is has travelled along the path. Intuitively, the profile states how much time the agent should spend

on every infinitesimal leg of the journey. As noted earlier, considering the square speed profile over simply the speed profile uncovers a hidden convexity of the time optimal path parametrization problem, as we will now see.

To begin with, as $x \mapsto x^{-\frac{1}{2}}$ is convex on $(0, \infty)$, the execution time of the path given by

$$\int_0^{S_{end}} dt = \int_0^{S_{end}} \frac{dt}{ds} ds = \int_0^{S_{end}} \left(\frac{ds}{dt}\right)^{-1} ds = \int_0^{S_{end}} \frac{1}{\sqrt{h(s)}} ds \tag{2.3}$$

is a convex functional of $h$. Furthermore, numerous actuation constraints also take the form of convex constraints on $h$. Indeed, noting the relation $\frac{d}{dt} = \frac{ds}{dt}\frac{d}{ds} = \sqrt{h(s)}\frac{d}{ds}$ and denoting differentiation by $s$ with a prime($'$), we have the following expressions for the velocity and acceleration of the vehicle at point $s \in [0, S_{end}]$:

$$\begin{aligned} \mathbf{v}(s) &= \gamma'(s)\sqrt{h(s)} \\ \mathbf{a}(s) &= \frac{1}{2}\gamma'(s)h'(s) + \gamma''(s)h(s). \end{aligned} \tag{2.4}$$

Let $\Pi_{hor} : \mathbb{R}^3 \to \mathbb{R}^2$ ($\Pi_{ver} : \mathbb{R}^3 \to \mathbb{R}$), defined by $\Pi_{hor}([x, y, z]) = [x, y]$ ($\Pi_{ver}([x, y, z]) = z$), denote the projection onto the world $x - y$ plane ($z$ axis). In this case, bounds on components of the state of the vehicle of the form

$$v_{hor} \leq v_{hor}^{max} \quad \text{and} \quad v_{ver} \leq v_{ver}^{max} \tag{2.5}$$

translate into

$$h(s) \leq \left(\frac{v_{hor}^{max}}{||\Pi_{hor}\gamma'(s)||_2}\right)^2 \quad \text{and} \quad h(s) \leq \left(\frac{v_{ver}^{max}}{||\Pi_{ver}\gamma'(s)||_2}\right)^2, \tag{2.6}$$

respectively. Note that squaring the constraints in Equation (2.5) was lossless since the agent can only move "forward" along the path, and as a result $ds/dt(s) \geq 0 \; \forall s \in [0, S_{end}]$. Similarly, a bound on the magnitude of the speed of the vehicle

$$v^{max} \geq ||\mathbf{v}(s)||_2 = \sqrt{||v_{hor}(s)||_2^2 + v_{ver}(s)^2} \tag{2.7}$$

translates into

$$h(s) \leq \left( \frac{v^{max}}{||\gamma'(s)||_2} \right)^2. \tag{2.8}$$

For the kinematic plane, we also have the relation

$$\theta = \theta(\gamma') = \arctan\left( \frac{e_2 \cdot \gamma'}{e_1 \cdot \gamma'} \right) \Rightarrow \omega = \dot{\theta} = \theta' \sqrt{h(s)} \tag{2.9}$$

and as a result, a bound on yaw rate

$$|\omega| \leq \omega^{max} \tag{2.10}$$

becomes upon squaring

$$h(s) \leq \left( \frac{\omega^{max}}{\theta'} \right)^2. \tag{2.11}$$

Next we turn to bounds on inputs of the agent. For example, bounds on components of translational inputs, and combinations thereof, such as

$$|a_\beta| \leq a_\beta^{max}, \tag{2.12}$$

where $\beta \in \{hor, ver, \emptyset\}$, with $\Pi_\emptyset = I_3$ and $a_\emptyset^{max}$ enforcing an upper bound on the norm of total acceleration, are equivalent to

$$\begin{aligned}
&||\Pi_\beta \mathbf{a}(s)||_2 \leq a_\beta^{max} \quad \Leftrightarrow \\
&||\frac{1}{2}\Pi_\beta \gamma'(s) h'(s) + \Pi_\beta \gamma''(s) h(s)||_2 \leq a_\beta^{max} \quad \Leftrightarrow \\
&h'(s) \in \left[ -2\frac{\gamma'(s)^T \Pi_\beta \gamma''(s)}{\gamma'(s)^T \Pi_\beta \gamma'(s)} \pm 2\sqrt{\frac{\gamma'(s)^T \Pi_\beta \gamma''(s)}{\gamma'(s)^T \Pi_\beta \gamma'(s)} - \frac{\gamma''(s)^T \Pi_\beta \gamma''(s) h(s)^2 - (a_\beta^{max})^2}{\gamma'(s)^T \Pi_\beta \gamma'(s)}} \right]
\end{aligned} \tag{2.13}$$

respectively, where we used the abbreviation $[a \pm b] := [a - b, a + b]$. Similarly, due to the following expression for slew rate

$$\tau = \dot{\omega} = \frac{1}{2}\theta'(s) h'(s) + \theta''(s) h(s) \tag{2.14}$$

an actuation bound of the form

$$|\tau| \leq \tau^{max} \tag{2.15}$$

becomes

$$\begin{aligned} -\tau^{max} \leq \frac{1}{2}\theta'(s)h'(s) + \theta''(s)h(s) \leq \tau^{max} \quad \Leftrightarrow \\ h'(s) \in \left[ -\frac{2h(s)\theta''(s)}{\theta'(s)} \pm \frac{2\tau^{max}}{|\theta'(s)|} \right]. \end{aligned} \tag{2.16}$$

### 2.2.2 Excursion into Non-Smooth Analysis

The equations above clearly show that we can express all actuation constraints on the vehicle for the problem at hand in terms of a pair of inequalities

$$\begin{aligned} h'(s) \leq f^+(s, h(s)) \\ h'(s) \geq f^-(s, h(s)) \end{aligned} \tag{2.17}$$

where, roughly speaking, $f^+$ ($f^-$ ) encodes the minimum (maximum) of the allowed upper (lower) bounds on $h'$ across all the various constraints derived earlier.

However, there is a problem. It may happen there does not exist an optimal profile $h$ that is differentiable on the whole interval $[0, S_{end}]$. Indeed, consider a vehicle following a straight line trajectory at level altitude from hover to hover state. It is easy to see that its optimal acceleration is of a bang-bang nature: accelerating full throttle before breaking as hard as possible. This results in an optimal square speed profile that has a discontinuous slope at the point where the control switches from maximal acceleration to maximal deceleration. For this reason, we have to consider a wider class of continuous functions for representing the square speed profile, and reinterpret constraints (2.17) accordingly.

We allow the class of functions $h$ to be the set of all continuous functions on $[0, S_{end}]$. The key change we make is turning to a special kind of derivative - the Dini derivative, as we recall next.

**Definition 1.** *[33] For a continuous function $h : [a, b] \to \mathbb{R}$, we define functions*

$D^+h,\ D^-h : [a,b) \to \mathbb{R} \cup \{\pm\infty\}$ *given by*

$$D^+h(s) = \limsup_{s' \downarrow s} \frac{h(s') - h(s)}{s' - s},\ D^-h(s) = \liminf_{s' \downarrow s} \frac{h(s') - h(s)}{s' - s},$$

*for all* $s \in [a,b)$. *Additionally,* $h$ *is called Dini differentiable if both* $D^+h$ *and* $D^-h$ *take on values strictly in* $\mathbb{R}$.

Dini derivatives are defined for all continuous functions, which are a strict superset of the class of differentiable functions. In particular, a differentiable function is also Dini-differentiable but not necessarily vice versa. Mathematically, the distinction comes from loosening a limit in favour of a supremal and infimal limit. Clearly, $D^+h(s) \geq D^-h(s)$ for all $s \in [a,b)$, and $h$ is right differentiable at $s$ if and only if $D^+h(s) = D^-h(s) \in \mathbb{R}$, in which case its right derivative equals $D^+h(s)$. Focusing on the right-hand limit is also intuitive from a control-theoretic perspective. Indeed, viewing the square speed profile as a variable to be controlled along the extent of the path, with distance playing the role of time, and derivative the role of the virtual control, the one sided limit allows the control to vary independently of its trajectory of values in its infinitesimal past.

Nevertheless, the Dini differentiation operator shares many characeristics of its smooth counterpart. For every pair of Dini differentiable functions $h_1$ and $h_2$, non-negative $\theta \in \mathbb{R}$, and right differentiable function $f$:

1. $D^+(h_1 + h_2) \leq D^+h_1 + D^+h_2$
2. $D^-(h_1 + h_2) \geq D^-h_1 + D^-h_2$
3. $D^\pm(\theta h_1) = \theta\ D^\pm h_1$
4. $D^+(-h_1) = -D^-h_1$
5. $D^\pm(h + f) = D^\pm h + f'$.

The first and third property above directly follow from subadditivity and positive homogeneity of supremal limits, whereas the fifth property effectively says that there

is no distinction between the action of the actual and Dini differentiation operator on (right) differentiable functions.

One of the fundamental theorems of real analysis is Cauchy's Mean Value Theorem, which relates the infinitesimal rate of change of a differentiable function to its variation over a longer interval of time. The following theorem may be viewed as a generalization of the former, being one of the key results of non-smooth analysis we shall heavily rely on in the remainder of the chapter.

**Theorem 1.** *[33] Let $h : [a, b] \to \mathbb{R}$ be a continuous function. The following are equivalent:*

1. *$h$ is monotonically decreasing (increasing) on $[a, b]$*
2. *$D^- h(s) \in [-\infty, 0]$ ($D^- h(s) \in [0, \infty]$) for all $s \in [a, b)$*
3. *$D^+ h(s) \in [-\infty, 0]$ ($D^+ h(s) \in [0, \infty]$) for all $s \in [a, b)$.*

We now provide an illustrative application of the latter claim we shall later need. To begin with, let us recall

**Definition 2.** *A function $h : [a, b] \to \mathbb{R}$ is said to be $\lambda$-Lipschitz if*

$$|h(s_2) - h(s_1)| \leq \lambda |s_2 - s_1| \quad \forall s_1, s_2 \in [a, b]. \tag{2.18}$$

**Example 1.** *Let $h : [a, b] \to \mathbb{R}$ be a continuous function. We then have*

$$h \text{ is } \lambda\text{-Lipschitz} \Leftrightarrow -\lambda \leq D^- h(s) \leq D^+ h(s) \leq \lambda \ \forall s \in [a, b]. \tag{2.19}$$

*Proof.* Consider an arbitrary pair of numbers $s_1 < s_2 \in [a, b]$, and define the function

$$g : s \mapsto h(s) - h(s_1) - \lambda(s - a) \quad \forall s \in [a, b].$$

By Property 5, we have $D^+ g(s) = D^+ h(s) - \lambda \leq 0$. As a result, by Theorem 1, $g$ is a decreasing function, which implies $h(s_2) \leq h(s_1) + \lambda(s_2 - s_1)$. Similarly, by redefining $g : s \mapsto h(s) - h(s_1) + \lambda(s - a)$, we conclude $h(s_2) \geq h(s_1) - \lambda(s_2 - s_1)$. Since the

pair $(s_1, s_2)$ was arbitrary, the claim follows. As an aside, by virtue of Theorem 1, the statement would be in place without explicit bounds on *both* $D^+$ and $D^-$. $\square$

### 2.2.3 Putting the Ingredients Together

Assembling the dynamic constraints within the analytical framework in the previous section, our problem becomes $P(B_u, B_l, f^+, f^-)$:

$$\begin{aligned} \underset{h:[a,b]\to[0,\infty)}{\text{minimize}} \quad & \int_a^b \frac{ds}{\sqrt{h(s)}} \\ \text{subject to} \quad & D^+h(s) \le f^+(s, h(s)),\ s \in [a, b), \\ & D^-h(s) \ge f^-(s, h(s)),\ s \in [a, b), \\ & B_l(s) \le h(s) \le B_u(s),\ s \in [a, b]. \end{aligned} \tag{2.20}$$

## 2.3 Algorithm

We now turn to an algorithm for numerically solving Problem (2.20), which in its current form is an infinite dimensional optimization problem. To find, represent, and ultimately use its solution on a digital computer, we seek a finite dimensional approximation of the optimal profile in terms of values it attains at a discrete mesh of points along the interval $[0, S_{end}]$.

A *discretization* $D = D([a, b], (s_i)_{i=0}^n)$ of interval $[a, b]$ is an increasing sequence of points $(s_i)_{i=0}^n$ satisfying $a = s_0 < ... < s_n = b$. We denote its cardinality by $|D| = n + 1$, and its resolution by $\Delta(D) = \max_{1\le i\le n} |s_i - s_{i-1}|$. Thus, given a discretization $D$ and problem $P(B_u, B_l, f^+, f^-)$, a numerical procedure aims to find approximations $(\hat{h}_i)_{i=0}^n$ to the optimal solution $\overline{h} = \overline{h}(P)$ at points $(s_i)_{i=0}^n$. Its *error* is defined as $\rho((\hat{h}_i)_{i=0}^n, P, D) = \max_{0\le i\le n} |\hat{h}_i - \overline{h}(s_i)|$, and it is said to be *asymptotically optimal* if $\rho \to 0$ as $\Delta(D) \to 0$. Below we present the algorithm in [60] that was designed to produced time optimal trajectories for robotic manipulators and legged robots.

The algorithm operates in a pair of passes along the discretization points in $D$. First, for every discretization point, the backward pass recovers the highest square

**Algorithm 1:** Backward-Forward Algorithm

**Data:** $D = (s_i)_{i=0}^n$, $(B_l(s_i))_{i=1}^n$, $(B_u(s_i))_{i=1}^n$, $f^+$, $f^-$
**Result:** $(\hat{h}_i)_{i=0}^n$
$h_n^{(b)} = B_u(s_n)$
**for** $i = n-1$ *to* $0$ **do**
  $h_i^{(b)} \leftarrow \max\{h \mid h \leq B_u(s_i), h + f^-(s_i, h)(s_{i+1} - s_i) \leq h_{i+1}^{(b)}\}$
  **if** $h_i^{(b)} = -\infty$ **then**
    return null
  **end**
**end**
$h_0^{(f)} = h_0^{(b)}$
**for** $i = 1$ *to* $n$ **do**
  $h_i^{(f)} \leftarrow \max\{h \mid h \leq h_i^{(b)}, h \leq h_{i-1}^{(f)} + f^+(s_{i-1}, h_{i-1}^{(f)})(s_i - s_{i-1})\}$
  **if** $h_i^{(f)} = -\infty$ **then**
    return null
  **end**
**end**
return $(\hat{h}_i)_{i=0}^n = (h_i^{(f)})_{i=0}^n$

speed which extends to a feasible profile for the remaining leg of the path. The forward pass, effectively mirroring the former, obtains the highest square speeds which can be reached by a feasible profile from the start of the path. Key to the computational efficiency of the algorithm is its incremental way of using the value of the optimal profile calculated for one discretization point, to calculate its corresponding value at a subsequent, adjacent discretization point.

The memory complexity of the algorithm is $O(|D|)$. Its running time is also

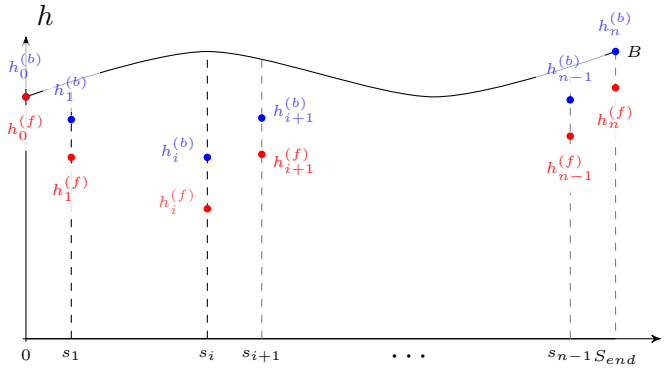



Figure 2-1: Backward-Forward Algorithm

$O(|D|)$, where the asymptotic $O$-notation hides a multiplicative factor equal to the amount of time necessary to solve the convex optimization problem at each iteration of the two passes. In general, this depends on the functional form of $f^{\pm}$. Often $f^{+}$ ($f^{-}$) is specified as a minimum (maximum) of $m$ concave (convex) functions, in which case the latter factor is $O(m)$.

The main advantage of the algorithm over other convex optimization-based approaches is its computational efficiency. However, it was initially only proved to converge to the optimal solution when $f^{\pm}$ are linear in their second variables. This is not the case for various planning constraints we might wish to impose. The key contribution of this chapter is that optimality continues to hold when $f^{+}$ and $f^{-}$ are merely required to be concave and convex in their second arguments, respectively.

## 2.4 Analysis

### 2.4.1 Analytical Characterization of Optimum

The main result of this section is presented in Theorem 3. We prove that the function defined as the pointwise supremum of all functions that are feasible for problem $P$ is also feasible and therefore optimal (Theorem 3(a)). We use this characterization to show continuity of the optimum with respect to a natural parameter quantifying the degree of relaxation of constraints of $P$ (Theorem 3(b)). Finally, we prove that the feasible set of $P$ is convex (Theorem 3(c)). To begin with, we note a useful result from Lipschitz analysis, whose proof we include for the sake of completeness.

**Theorem 2.** *[28] Let $\{h_\alpha\}_{\alpha\in A}$ be an arbitrary non-empty family of uniformly bounded $\lambda$-Lipschitz functions defined on interval $[a,b]$. Functions $\overline{h}, \underline{h} : [a,b] \to \mathbb{R}$, defined by*

$$\overline{h}(s) = \sup_{\alpha\in A} h_\alpha(s),\ \underline{h}(s) = \inf_{\alpha\in A} h_\alpha(s),$$

*for all $s \in [a,b]$, are well-defined and $\lambda$-Lipschitz.*

*Proof.* Consider an arbitrary pair of numbers $s_1, s_2 \in [a,b]$. By definition, we have

$$\begin{aligned} & h_\alpha(s_2) - h_\alpha(s_1) \le \lambda|s_2 - s_1| \ \forall \alpha \in A \\ & \Rightarrow h_\alpha(s_2) \le h_\alpha(s_1) + \lambda|s_2 - s_1| \ \forall \alpha \in A \\ & \Rightarrow h_\alpha(s_2) \le \sup_{\alpha' \in A} h_{\alpha'}(s_1) + \lambda|s_2 - s_1| \ \forall \alpha \in A \\ & \Rightarrow \sup_{\alpha' \in A} h_{\alpha'}(s_2) \le \sup_{\alpha' \in A} h_{\alpha'}(s_1) + \lambda|s_2 - s_1| \ \forall \alpha \in A \\ & \Rightarrow \overline{h}(s_2) - \overline{h}(s_1) \le \lambda|s_2 - s_1|. \end{aligned} \tag{2.21}$$

By interchanging $s_1$ and $s_2$ above, we also obtain $\overline{h}(s_2) - \overline{h}(s_1) \le \lambda|s_2 - s_1|$ and the claim for $\overline{h}$ follows. By an analogous argument, one may derive the same conclusion for $\underline{h}$. □

**Theorem 3.** *Let $B_l, B_u : [a,b] \to \mathbb{R}$ be a pair of continuous functions with $B_u(s) \ge B_l(s)$ for all $s \in [a,b]$. Define region $F := \{(s,h) \mid s \in [a,b],\ B_l(s) \le h \le B_u(s)\}$. Suppose $f^+, f^- : F \to \mathbb{R}$ are a pair of continuous functions with $f^+(s,h) \ge f^-(s,h)$ for all $(s,h) \in F$. In particular, $|f^\pm| \le B$ for some $B > 0$. For a real number $\xi \ge 0$, a function $h : [a,b] \to \mathbb{R}$ is called $\xi$-feasible if it satisfies the following conditions:*

1. *$h$ is continuous*
2. *$B_l(s) \le h(s) \le B_u(s)$ for all $s \in [a,b]$*
3. *$D^- h(s) \ge f^-(s, h(s)) - \xi$ for all $s \in [a,b]$*
4. *$D^+ h(s) \le f^+(s, h(s)) + \xi$ for all $s \in [a,b]$.*

*Let $A_\xi$ denote the set of $\xi$-feasible functions.*

*a) Assume $\emptyset \ne \{h_\alpha\}_{\alpha \in C} \subseteq A_\xi$ for some (possibly uncountable) index set $C$. Then, $\overline{h}, \underline{h} : [a,b] \to \mathbb{R}$, defined by*

$$\overline{h}(s) = \sup_{\alpha \in C} h_\alpha(s),\ \underline{h}(s) = \inf_{\alpha \in C} h_\alpha(s),$$

*for all $s \in [a,b]$, are $\xi$-feasible functions.*

*b) Assume $A_0 \neq \emptyset$. Define $\overline{h}_\xi = \sup_{h \in A_\xi} h$, with the understanding that suprema of functions is interpreted pointwise. Then,*

$$\left|\left|\overline{h}_\xi - \overline{h}_0\right|\right|_\infty \to 0 \text{ as } \xi \to 0.$$

*c) Assume functions $f^+$ and $f^-$ are concave and convex in their second arguments, respectively. For every $\xi \geq 0$, for every pair of $\xi$-feasible functions $h_1$ and $h_2$, and for every $\theta \in [0,1]$, the function $h_\theta = \theta h_1 + (1-\theta)h_2$ is also $\xi$-feasible.*

*Proof.* (a) We only give detailed proof of the claim for $\xi = 0$ and $\overline{h}$. The corresponding result for $\overline{h}$ when $\xi > 0$ can be recovered from the result for $\xi = 0$ by redefining $f^\pm \to f^\pm \pm \xi$. Similarly, the result for $\underline{h}$ can be recovered from the result for $\overline{h}$ by redefining $B_l \to -B_u$, $B_u \to -B_l$, $f^\pm \to -f^\mp$ and using Properties (1)-(5) of Dini derivatives.

Since functions $B_u$ and $B_l$ are continuous on $[a,b]$, they are bounded. As $C \neq \emptyset$, $\overline{h}$ is well defined. For arbitrary $s \in [a,b]$, taking the supremum over $\alpha \in C$ of the inequality $B_l(s) \leq h_\alpha(s) \leq B_u(s)$, we verify $\overline{h}$ satisfies Condition (2). In particular, $f^+$ and $f^-$ are defined at all points $\left(s, \overline{h}(s)\right)$ for $s \in [a,b]$.

By using the ($\Leftarrow$) component of the equivalence from Example 1, the fact that $|f^\pm| \leq B$ implies $h_\alpha$ is $B$-Lipschitz for all $\alpha \in C$. By Theorem 2, $\overline{h}$ is also $B$-Lipschitz; in particular $\overline{h}$ is continuous, verifying Condition (1).

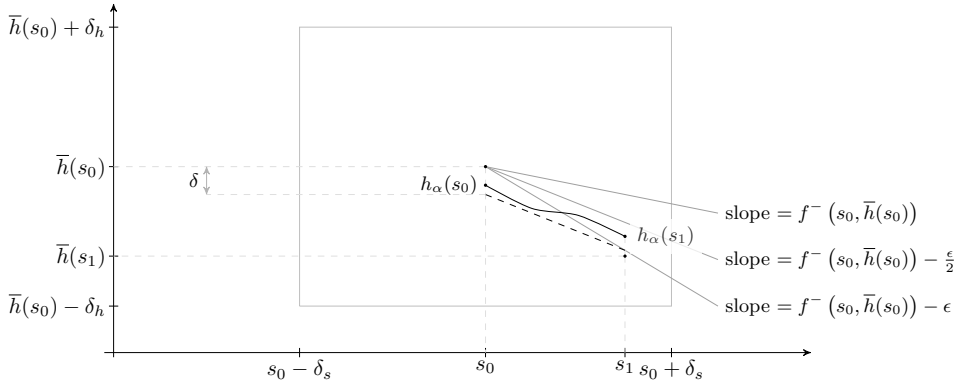



Figure 2-2: Illustration of the Proof of Condition (3) of Theorem 3

Next, we show $\overline{h}$ satisfies Condition (3). Assume, for a contradiction,

$$D^-\overline{h}(s_0) < f^-(s_0, \overline{h}(s_0)) \tag{2.22}$$

for some $s_0 \in [a, b)$. Hence, there exists $\epsilon > 0$ such that $D^-\overline{h}(s_0) < f^-\left(s_0, \overline{h}(s_0)\right) - \epsilon$. Continuity of $f^-$ implies there exist $\delta_s, \delta_h > 0$ such that

$$f^-(s, h) \geq f^-\left(s_0, \overline{h}(s_0)\right) - \frac{\epsilon}{2} \tag{2.23}$$

for all $(s, h) \in F \cap [s_0 - \delta_s, s_0 + \delta_s] \times \left[\overline{h}(s_0) - \delta_h, \overline{h}(s_0) + \delta_h\right]$. Keeping $\delta_h$ intact while shrinking $\delta_s$ if necessary, we may assume

$$\delta_h > (B + \epsilon)\delta_s. \tag{2.24}$$

The definition of $D^-$ implies there exists a strictly decreasing sequence $(s_n)_{n\geq 1}$ of real numbers such that

$$s_n \downarrow s_0, \quad \text{and} \quad \frac{\overline{h}(s_n) - \overline{h}(s_0)}{s_n - s_0} \leq f^-\left(s_0, \overline{h}(s_0)\right) - \epsilon \ \ \forall n \geq 1. \tag{2.25}$$

In particular, we may assume $s_1 \in (s_0, s_0 + \delta_s)$ satisfies

$$\overline{h}(s_1) \leq \overline{h}(s_0) + \left(f^-\left(s_0, \overline{h}(s_0)\right) - \epsilon\right)(s_1 - s_0). \tag{2.26}$$

Define

$$\delta = \min\left(\frac{\epsilon}{2}(s_1 - s_0), \delta_h - B\delta_s\right), \tag{2.27}$$

where we note $\delta > 0$ by virtue of Equation (2.24). The definition of $\overline{h}$ implies there exists $\alpha \in C$ such that

$$\overline{h}(s_0) - \delta < h_\alpha(s_0) \leq \overline{h}(s_0). \tag{2.28}$$

This will ultimately let us apply the non-smooth Mean Value Theorem, namely Theorem 1 to the function $h_\alpha$ on the interval $[s_0, s_1]$. From here, a contradiction will easily follow.

Since $h_\alpha$ is $B$-Lipschitz, it follows that for all $s \in [s_0, s_1]$ we have

$$\begin{aligned}
\left|h_\alpha(s) - \overline{h}(s_0)\right| &= \left|h_\alpha(s) - h_\alpha(s_0) + h_\alpha(s_0) - \overline{h}(s_0)\right| \\
&\leq |h_\alpha(s) - h_\alpha(s_0)| + \left|h_\alpha(s_0) - \overline{h}(s_0)\right| \\
&\leq B(s - s_0) + \delta \\
&\leq B\delta_s + \delta_h - B\delta_s = \delta_h.
\end{aligned} \tag{2.29}$$

Hence, for all $s \in [s_0, s_1]$

$$(s, h_\alpha(s)) \in F \cap [s_0 - \delta_s, s_0 + \delta_s] \times \left[\overline{h}(s_0) - \delta_h, \overline{h}(s_0) + \delta_h\right],$$

implying

$$D^- h_\alpha(s) = f^-\left(s, h_\alpha(s)\right) \geq f^-\left(s_0, \overline{h}(s_0)\right) - \frac{\epsilon}{2}. \tag{2.30}$$

By another application of Theorem 1,

$$\begin{aligned}
h_\alpha(s_1) &\geq h_\alpha(s_0) + \left(f^-\left(s_0, \overline{h}(s_0)\right) - \frac{\epsilon}{2}\right)(s_1 - s_0) \\
&> \overline{h}(s_0) - \delta + \left(f^-(s_0, \overline{h}(s_0)) - \frac{\epsilon}{2}\right)(s_1 - s_0) \\
&\geq \overline{h}(s_0) - (s_1 - s_0)\frac{\epsilon}{2} + \left(f^-(s_0, \overline{h}(s_0)) - \frac{\epsilon}{2}\right)(s_1 - s_0) \\
&\geq \overline{h}(s_1),
\end{aligned} \tag{2.31}$$

where the last inequality follows from Equation (2.26). However, Equation (2.31) violates the definition of $\overline{h}$ at $s_1$. This gives the desired contradiction, and shows $\overline{h}$ satisfies Condition (3). The proof that $\overline{h}$ satisfies Condition (4) is omitted since it can be derived analogously.

(b) Consider arbitrary real numbers $0 \leq \xi_1 \leq \xi_2$. According to part (a) of Theorem 3, $\overline{h}_{\xi_1} \in A_{\xi_1} \subseteq A_{\xi_2}$. This implies $\overline{h}_{\xi_1} \leq \overline{h}_{\xi_2}$. Hence, for every $s \in [a, b]$, $\overline{h}_\xi(s)$ is monotonically increasing in $\xi \geq 0$ and bounded below by $\overline{h}_0(s)$. As a result, function $\tilde{h} : [a, b] \to \mathbb{R}$, given by $\tilde{h}(s) = \inf_{\xi > 0} \overline{h}_\xi(s)$ for all $s \in [a, b]$, is well defined and satisfies $\tilde{h} \geq \overline{h}_0$.

On the other hand, monotonicity of $\overline{h}_\xi(s)$ implies $\tilde{h}(s) = \inf_{0 < \nu \leq \xi} \overline{h}_\nu(s)$ for every

$\xi > 0$. Since $\overline{h}_\nu \in A_\nu \subseteq A_\xi$ for every $\nu \leq \xi$, another application of part (a) of Theorem 3 to the non-empty set of functions $\left(\overline{h}_\nu\right)_{0<\nu\leq\xi} \subseteq A_\xi$ yields $\tilde{h} \in A_\xi$. Since $\xi > 0$ was arbitrary, we have $\tilde{h} \in \cap_{\xi>0} A_\xi = A_0$. By definition of $\overline{h}_0$, we thus have $\overline{h}_0 \geq \tilde{h}$.

Combining previous observations, we get $\tilde{h} = \overline{h}_0$. Thus, $\overline{h}_\xi(s) \downarrow \overline{h}_0(s)$ as $\xi \downarrow 0$ for all $s \in [a, b]$. Since functions $(\overline{h}_\xi)_{\xi\geq 0}$ are continuous on interval $[a, b]$, uniform convergence follows.

(c) Consider any $\xi \geq 0$ and $\theta \in [0, 1]$. Function $h_\theta$ clearly satisfies Conditions (1) and (2) of Theorem 3, so we turn to deriving Condition (3). As in part (a), the proof of Condition (4) is omitted as it can be derived analogously. We have:

$$\begin{aligned} D^- h_\theta(s) &= D^-(\theta h_1 + (1-\theta)h_2)(s) \\ &\geq \theta D^- h_1(s) + (1-\theta) D^- h_2(s) \\ &\geq \theta(f^-(s, h_1(s)) - \xi) + (1-\theta)(f^-(s, h_2(s)) - \xi) \\ &\geq f^-(s, \theta h_1(s) + (1-\theta)h_2(s)) - \xi \\ &= f^-(s, h_\theta(s)) - \xi. \end{aligned} \tag{2.32}$$

The first inequality above follows from Properties (2) and (3) of Dini derivatives, whereas the second inequality follows from $\xi$-feasibility of $h_1$ and $h_2$. Finally, the last inequality follows from convexity of $f^-$ in its second argument. □

Two remarks are in order. The first pertains to continuity of problem data. The theorem holds without any continuity assumptions on $B_l$ and $B_u$ provided functions $f^\pm$ can be bounded uniformly on $F$. Furthermore, closer examination of the proof of part (a) shows that $f^\pm$ need only be assumed continuous from the right in that for all $\epsilon > 0$, and for all $(s_0, h_0) \in F$, there exist $\delta_s, \delta_h > 0$ such that

$$(s, h) \in [s_0, s_0 + \delta_s) \times (h_0 - \delta_h, h_0 + \delta_h) \Rightarrow |f^\pm(s, h) - f^\pm(s_0, h_0)| < \epsilon. \tag{2.33}$$

The second is the role of convexity in the existence and nature of optimal solutions to the problem considered. In particular, it is worth emphasizing that the results of

part (a) and (b) place no assumptions on convexity properties of $f^{\pm}$ in their second argument. Furthermore, the region $F$ need not be delimited by only two functions - it can also have arbitrary disallowed regions without altering the conclusion of the theorem. However, it is critical that the Dini derivative of $h$ be allowed values in a whole interval. Suppose, for the sake of argument, it was allowed to take only two values, $+B$ and $-B$. In such a scenario, taking $B_l \equiv 0$ and $B_u < \frac{b-a}{2B}$, it is easy to see that an optimal solution in the space of continuous profiles would not exist.

### 2.4.2 Asymptotic Optimality

The main result of this section is Theorem 5 which proves asymptotic optimality of Algorithm 1 for all feasible problems $P$ amenable to convex optimization approaches. First, in Theorem 4 we recall an important result, which:

a) characterizes a lower bound on the length of the interval on which a solution to an ordinary differential equation is defined

b) proves that a continuous function can never exceed a differentiable function whose derivative upper bounds the former's Dini derivative.

**Theorem 4.** *[35] In addition to the setup of Theorem 3, let:*

1. *$B_u$ and $B_l$ satisfy $B_u > B_l$*

2. *for every pair of continuous functions $U, L : [a, b] \to \mathbb{R}$ such that $B_l < L < U < B_u$, there exist $\lambda_s, \lambda_h > 0$ such that $f^+$ and $f^-$ are $\lambda_s$-Lipschitz and $\lambda_h$-Lipschitz on $\{(s, h) | s \in [a, b], L(s) \leq h \leq U(s)\}$ in their first and second arguments respectively.*

*Consider arbitrary $g \in \{f^+, f^-\}$, $s_0 \in [a, b)$, and $h_0$ such that $L(s_0) < h_0 < U(s_0)$.*

*a) There exists $\delta > 0$ such that the initial value problem*

$$h'(s) = g(s, h(s)) \text{ subject to } h(s_0) = h_0$$

*admits a unique solution on interval* $[s_0, s_0 + \delta]$. *Furthermore, we may choose*

$$s_0 + \delta = \min\left(b, \inf\{s \geq s_0 | h(s) \notin (L(s), U(s))\}\right).$$

*b) Every continuous function* $\tilde{h} : [s_0, s_0 + \tilde{\delta}] \to \mathbb{R}$, *such that* $L(s) < \tilde{h}(s) < U(s)$ *and* $D^+\tilde{h}(s) \leq g(s, \tilde{h}(s))$ *for all* $s \in [s_0, s_0 + \tilde{\delta})$, *satisfies*

$$\tilde{h}(s) \leq h(s)$$

*for all* $s \in [s_0, s_0 + \min(\delta, \tilde{\delta})]$.

Before turning to the main result of the section, we give a definition. For a problem $P(B_u, B_l, f^+, f^-)$ and discretization $D([a, b], (s_i)_{i=0}^n)$, we call a sequence $(h_i)_{i=0}^n$ *admissible* if:

$$B_l(s_i) \leq h_i \leq B_u(s_i)$$

for all $0 \leq i \leq n$, and

$$f^-(s_i, h_i) \leq \frac{h_{i+1} - h_i}{s_{i+1} - s_i} \leq f^+(s_i, h_i)$$

for all $0 \leq i \leq n - 1$. Additionally, we will denote by $\overline{h}(P)$ ($\underline{h}(P)$) the pointwise supremum (infimum) of all feasible functions for $P$.

**Theorem 5.** *Assume in addition to the setup of Theorem 4, problem* $P(B_u, B_l, f^+, f^-)$ *is feasible and* $\overline{h} := \overline{h}(P) > \underline{h}(P) =: \underline{h}$. *For every* $\epsilon > 0$, *there exists an* $\eta > 0$ *such that for every discretization* $D([a, b], (s_i)_{i=0}^n)$ *with resolution* $\Delta(D) \leq \eta$, *Algorithm 1 returns an admissible sequence* $(\hat{h}_i)_{i=0}^n$ *with* $\rho((\hat{h}_i)_{i=0}^n, P, D) < \epsilon$.

Before embarking on the rigorous proof of the stated theorem, we provide high-level intuition of our line of approach. The proof is split across two steps. The first involves showing that Algorithm 1 never produces a solution which drops far below the optimum. Roughly, this is accomplished by showing that sufficiently fine discretizations allow squeezing an admissible sequence of profile values between any

two feasible solutions, provided one dominates the other (see Figure 2-3). The second involves showing that by considering a small downward translation of the fictive profile, obtained by piecewise linear interpolation of computed profile values, results in a near-feasible profile for the original problem (see Figure 2-4). Having the latter two figures at hand might streamline picturing the formal proof to follow.

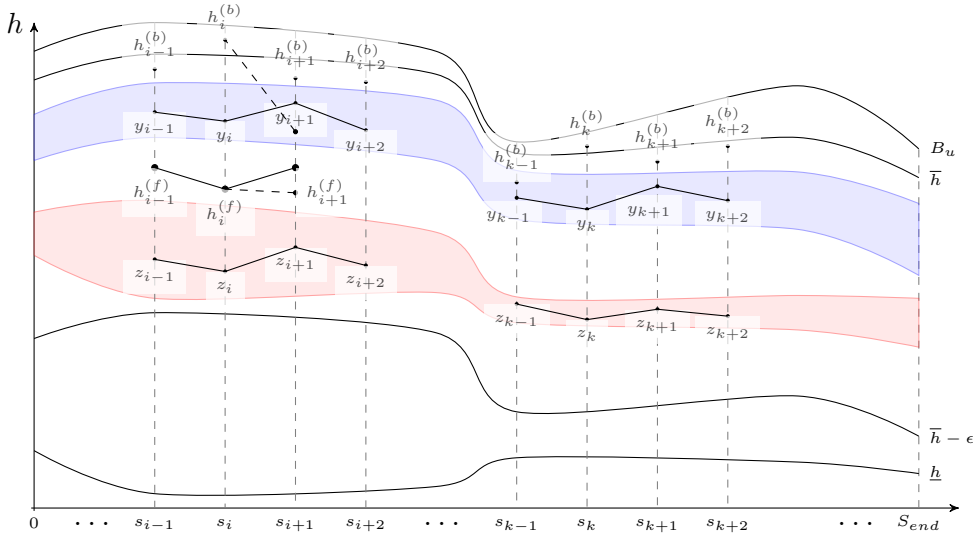



Figure 2-3: Overview of Step One of the Proof of Asymptotic Optimality of the Backward-Forward Algorithm

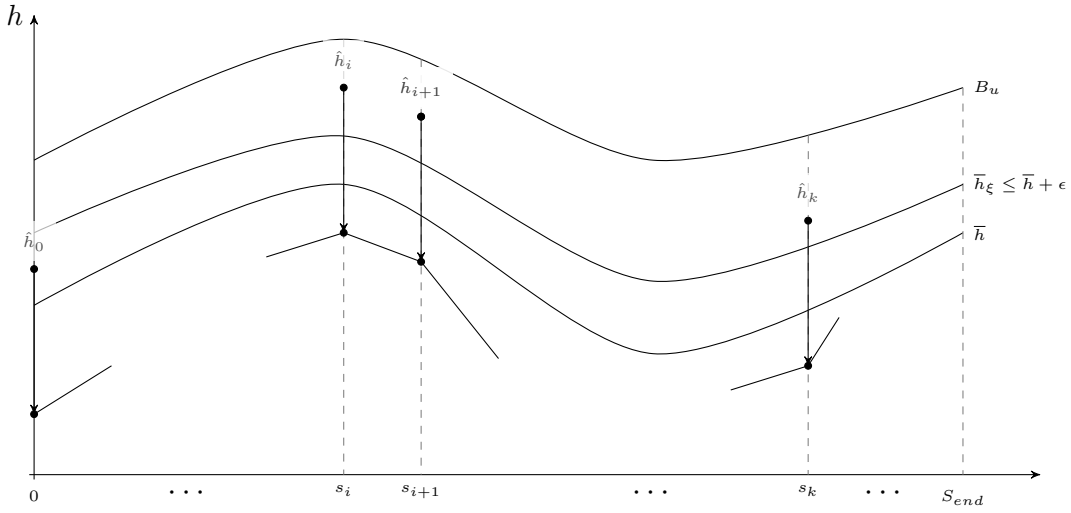



Figure 2-4: Overview of Step Two of the Proof of Asympotic Optimality of the Backward-Forward Algorithm

*Proof.* Since $\underline{h} < \overline{h}$ are continuous functions defined over the compact interval $[a, b]$, the function $\overline{h} - \underline{h}$ achieves its maximum $M$ and minimum $m$ satisfying

$$0 < m < M. \tag{2.34}$$

Fix an arbitrary $\epsilon > 0$ where, without loss of generality, $\epsilon < m$.

The proof will consist of two parts. The first (second) part will show there exists $\eta_1 > 0$ ($\eta_2 > 0$) such that for every discretization $D([a, b], (s_i)_{i=0}^n)$ with resolution at most $\eta_1$ ($\eta_2$), Algorithm 1 produces an admissible sequence $(\hat{h}_i)_{i=0}^n$ satisfying

$$\hat{h}_i \geq \overline{h}(s_i) - \epsilon \quad (\hat{h}_i \leq \overline{h}(s_i) + \epsilon) \quad \forall 0 \leq i \leq n. \tag{2.35}$$

Setting $\eta = \min(\eta_1, \eta_2)$ will yield the proof of the theorem.

For proving the first part, consider feasible functions $h_l$ and $h_u$ such that

$$\overline{h} - \epsilon < h_l < h_u < \overline{h}. \tag{2.36}$$

To see such functions exist, according to part (c) of Theorem 3, we may restrict attention to $h_u$ ($h_l$) of the form

$$h_u = (1 - \theta_u)\underline{h} + \theta_u \overline{h} \quad \left(h_l = (1 - \theta_l)\underline{h} + \theta_l \overline{h}\right), \tag{2.37}$$

where $\theta_l, \theta_u \in [0, 1]$. The inequality $h_l < h_u < \overline{h}$ holds whenever $\theta_l < \theta_u \in (0, 1)$, while

$$\overline{h} - \epsilon < h_l \;\Leftrightarrow\; (1 - \theta_l)(\overline{h} - \underline{h}) < \epsilon \;\Leftrightarrow\; \theta_l > 1 - \frac{\epsilon}{M}. \tag{2.38}$$

Thus, setting $\theta_l = 1 - \epsilon/2M$ and $\theta_u = 1 - \epsilon/4M$, relation (2.36) holds.

Define $\delta_1 = \inf_{s\in[a,b]}(\overline{h} - h_u)$, $\delta_2 = \inf_{s\in[a,b]}(h_u - h_l)$, and $\delta_3 = \inf_{s\in[a,b]}(h_l - (\overline{h} - \epsilon))$. Continuity of feasible profiles and compactness of $[a, b]$ imply $\delta_1, \delta_2, \delta_3 > 0$. Set $\delta = \min\left\{\frac{\delta_1}{3}, \frac{\delta_2}{3}, \frac{\delta_3}{3}\right\}$. By assumption, there exist $\lambda_s,\ \lambda_h > 0$ such that for all

$$(s_1, h_1), (s_2, h_2) \in G := \{(s, h) | s \in [a, b], h_l(s) - \delta \leq h \leq h_u(s) + \delta\}, \tag{2.39}$$

we have

$$\left|f^{\pm}(s_1,h_1)-f^{\pm}(s_2,h_2)\right| \le \lambda_s|s_2-s_1|+\lambda_h|h_2-h_1|. \tag{2.40}$$

Recalling that $|f^{\pm}| \le B$, and assuming without loss of generality $B > 1$, we may choose

$$\eta_1 = \delta e^{-\lambda_h B(b-a)} \min\left\{\frac{1}{2B}, \frac{B\lambda_h}{\lambda_s + B\lambda_h}\right\}. \tag{2.41}$$

We claim that for any discretization $D([a,b],(s_i)_{i=0}^n)$ with resolution $\Delta(D) \le \eta_1$, Algorithm 1 produces an admissible sequence $(\hat{h}_i)_{i=0}^n$ satisfying $\hat{h}_i \ge \overline{h}(s_i) - \epsilon$ for all $0 \le i \le n$. The proof of the claim will proceed in two stages. The first will show the sequence $(h_i^{(b)})_{i=0}^n$ generated by the backward pass satisfies $h_i^{(b)} \ge h_u(s_i) - \delta$ for all $0 \le i \le n$. The second will show the sequence $(h_i^{(f)})_{i=0}^n$ generated by the forward pass satisfies $h_i^{(f)} \ge h_l(s_i) - \delta \ge \overline{h}(s_i) - \epsilon$ for all $0 \le i \le n$.

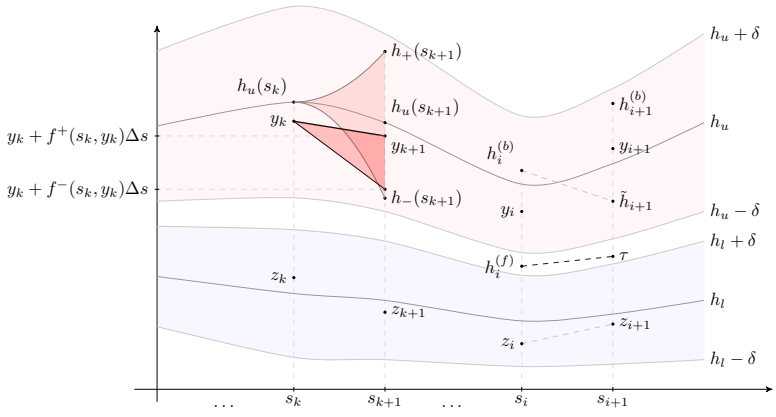



Figure 2-5: Pictorial Summary of Inductive Step of Lemma 1

**Lemma 1.** *There exist admissible sequences $(y_k)_{k=0}^n$ and $(z_k)_{k=0}^n$ such that for every $0 \le k \le n$, we have $|y_k - h_u(s_k)| \le \delta$ and $|z_k - h_l(s_k)| \le \delta$ (see Figure 2-5).*

*Proof.* We only prove existence of $(y_k)_{k=0}^n$ as that of $(z_k)_{k=0}^n$ follows analogously. To this end, define $e : [a,b] \to \mathbb{R}$ given by

$$e(s) = \delta e^{-\lambda_h B(b-a)}\left(e^{\lambda_h B(s-a)} - \frac{1}{2}\right) \tag{2.42}$$

for all $s \in [a,b]$. Clearly $0 < e < \delta$. We will inductively construct an admissible

sequence $(y_k)_{k=0}^n$ which satisfies $|y_k - h_u(s_k)| \leq e(s_k)$, thus proving the lemma.

Set $y_0 = h_u(s_0)$. Assume we have defined an admissible sequence $(y_j)_{j\leq k}$ satisfying $|y_j - h_u(s_j)| \leq e(s_j)$ for all $0 \leq j \leq k \leq n-1$. We now define $y_{k+1}$. Consider the pair of initial value problems

$$h'_{\pm}(s) = f^{\pm}(s, h_{\pm}(s)) \text{ subject to } h_{\pm}(s_k) = h_u(s_k). \tag{2.43}$$

First, we verify they have unique solutions on interval $s \in [s_k, s_{k+1}]$. In what follows, we focus on $h_+$, as the observation for $h_-$ will hold analogously. Part (a) of Theorem 4 implies that the corresponding IVP has a unique solution on $[s_k, s_k + \delta_s]$ where $\delta_s > 0$ is implicitly specified by

$$s_k + \delta_s = min\left(b, \inf\{s \geq s_k \mid h_+(s) \notin (h_l(s) - \delta, h_u(s) + \delta)\}\right). \tag{2.44}$$

It remains to show $s_k + \delta_s \geq s_{k+1}$. Suppose, for a contradiction, this was not the case. Then, let

$$\tilde{s} = \inf\{s \in [s_k, s_{k+1}] \mid h_+(s) \notin (h_l(s) - \delta, h_u(s) + \delta)\}. \tag{2.45}$$

By our former assumption, $\tilde{s} \in (s_k, s_{k+1})$, and by continuity of $h_+, h_u, h_l$ we must have $h_+(\tilde{s}) \in \{h_u(s) + \delta, h_l(s) - \delta\}$, and therefore

$$|h_+(\tilde{s}) - h_u(\tilde{s})| \geq \delta. \tag{2.46}$$

However, by Cauchy's Mean Value Theorem coupled with the fact $|f^{\pm}| \leq B$, we have

$$|h_+(\tilde{s}) - h_+(s_k)| \leq B|\tilde{s} - s_k| \quad \text{and} \quad |h_u(\tilde{s}) - h_u(s_k)| \leq B|\tilde{s} - s_k| \tag{2.47}$$

implying

$$|h_+(\tilde{s}) - h_u(\tilde{s})| \leq 2B|\tilde{s} - s_k| < 2B\eta_1 \leq \delta, \tag{2.48}$$

contradicting Equation (2.46). With existence and uniqueness of solutions to the IVPs in place, part (b) of Theorem 4 implies $h_-(s) \leq h_u(s) \leq h_+(s)$ for all $s \in [s_k, s_{k+1}]$. In particular, $h_-(s_{k+1}) \leq h_u(s_{k+1}) \leq h_+(s_{k+1})$. Define $\Delta s = s_{k+1} - s_k$. Lipschitz

continuity of $f^{\pm}$ and Equation (2.43) imply (see Figure 2-5)

$$\left|h_{\pm}(s_{k+1}) - (h_{\pm}(s_k) + f^{\pm}(s_k, h_u(s_k))\Delta s)\right| \leq \frac{1}{2}\left(\lambda_s + B\lambda_h\right)\Delta s^2. \tag{2.49}$$

Similarly,

$$\begin{aligned} &|(y_k + f^{\pm}(s_k, y_k)\Delta s) - (h_{\pm}(s_k) + f^{\pm}(s_k, h_u(s_k))\Delta s)| \\ &\qquad \leq (1 + B\lambda_h \Delta s)\left|y_k - h_u(s_k)\right|. \end{aligned} \tag{2.50}$$

Since an admissible value of $y_{k+1}$ can take on any value in the interval $[y_k + f^{-}(s_k, y_k)\Delta s, y_k + f^{+}(s_k, y_k)\Delta s]$, Equations (2.49) and (2.50) imply the existence of admissible $y_{k+1}$ satisfying

$$\begin{aligned} |y_{k+1} - h_u(s_{k+1})| &\leq \frac{1}{2}(\lambda_s + B\lambda_h)\Delta s^2 + (1 + B\lambda_h\Delta s)|y_k - h_u(s_k)| \\ &\leq \frac{1}{2}(\lambda_s + B\lambda_h)\Delta s^2 + (1 + B\lambda_h\Delta s)e(s_k) \\ &\leq \frac{1}{2}B\lambda_h\delta e^{-\lambda_h B(b-a)}\Delta s + (1 + B\lambda_h\Delta s)\delta e^{-\lambda_h B(b-a)}\left(e^{\lambda_h B(s_k - a)} - \frac{1}{2}\right) \\ &= \delta e^{-\lambda_h B(b-a)}\left(\frac{B\lambda_h\Delta s}{2} + (1 + B\lambda_h\Delta s)\left(e^{\lambda_h B(s_k - a)} - \frac{1}{2}\right)\right) \\ &= \delta e^{-\lambda_h B(b-a)}\left(\frac{B\lambda_h\Delta s}{2} + (1 + B\lambda_h\Delta s)e^{\lambda_h B(s_k - a)} - \frac{1}{2}(1 + B\lambda_h\Delta s)\right) \\ &\leq \delta e^{-\lambda_h B(b-a)}\left(e^{\lambda_h B\Delta s}e^{\lambda_h B(s_k - a)} - \frac{1}{2}\right) \\ &= e(s_{k+1}). \end{aligned} \tag{2.51}$$

This completes the proof of the inductive step and the lemma.

□

We now return to proofs of stages one and two. Assume sequences $(y_i)_{i=0}^n$ and $(z_i)_{i=0}^n$ have been constructed as in Lemma 1. Existence of $(y_i)_{i=0}^n$ immediately implies the sequence $(h_i^{(b)})_{i=0}^n$ is well-defined and satisfies $h_i^{(b)} \geq y_i \geq h_u(s_i) - \delta$ for all $0 \leq i \leq n$. This finishes the proof of stage one.

For stage two, we prove by induction on $i$ that $h_i^{(f)}$ is well-defined and satisfies $z_i \leq h_i^{(f)} \leq h_i^{(b)}$ for all $0 \leq i \leq n$. The base case $i = 0$ follows from $h_0^{(f)} = h_0^{(b)} \geq y_0 > z_0$. For the inductive hypothesis, assume the statement holds for $i$. We now show it

also holds for $i+1$. The definition of the backward pass implies there exists $\tilde{h}_{i+1}$ such that

$$h_{i+1}^{(b)} \geq \tilde{h}_{i+1} = h_i^{(b)} + f^-(s_i, h_i^{(b)})(s_{i+1} - s_i). \tag{2.52}$$

We recall assumption $|f^{\pm}| \leq B$ along with feasibility of $h_u$ implies $h_u$ is $B$-Lipschitz. Thus,

$$\begin{aligned} \tilde{h}_{i+1} &\geq h_i^{(b)} - B(s_{i+1} - s_i) \\ &\geq h_u(s_i) - \delta - B(s_{i+1} - s_i) \\ &\geq h_u(s_{i+1}) - \delta - 2B(s_{i+1} - s_i). \end{aligned} \tag{2.53}$$

Since $s_{i+1} - s_i \leq \frac{\delta}{2B}$, we have

$$h_{i+1}^{(b)} \geq \tilde{h}_{i+1} \geq h_u(s_{i+1}) - 2\delta \geq h_l(s_{i+1}) + \delta \geq z_{i+1}. \tag{2.54}$$

Since $z_i \leq h_i^{(f)} \leq h_i^{(b)}$ (see Figure 2-5), there exists $\theta \in [0,1]$ such that $h_i^{(f)} = \theta h_i^{(b)} + (1-\theta) z_i$. Consider $\tau = \theta \tilde{h}_{i+1} + (1-\theta) z_{i+1}$. Equation (2.54) implies

$$z_{i+1} \leq \tau \leq h_{i+1}^{(b)}. \tag{2.55}$$

Furthermore,

$$\begin{aligned} \tau - h_i^{(f)} &= \theta(\tilde{h}_{i+1} - h_i^{(b)}) + (1-\theta)(z_{i+1} - z_i) \\ &\geq \left(\theta f^-(s_i, h_i^{(b)}) + (1-\theta) f^-(s_i, z_i)\right)(s_{i+1} - s_i) \\ &\geq f^-(s_i, \theta h_i^{(b)} + (1-\theta) z_i)(s_{i+1} - s_i) \\ &= f^-(s_i, h_i^{(f)})(s_{i+1} - s_i), \end{aligned} \tag{2.56}$$

where the first inequality above follows from Equation (2.52) and admissibility of $(z_i)_{i=0}^n$, and the second inequality from convexity of $f^-$ in its second argument. Similarly, we obtain

$$\tau - h_i^{(f)} \leq f^+(s_i, h_i^{(f)})(s_{i+1} - s_i). \tag{2.57}$$

Equations (2.55), (2.56), and (2.57) imply $h_{i+1}^{(f)}$ is well-defined and satisfies $h_{i+1}^{(f)} \geq z_{i+1}$.

This finishes the proof of the inductive step, the proof of stage two and of the first part of the theorem.

To prove the second part, consider $\xi > 0$ such that $\overline{h}_\xi \leq \overline{h} + \frac{\epsilon}{2}$. Such $\xi$ exists due to part (b) of Theorem 3. Uniform continuity of $f^\pm$ implies there exists

$$\delta_1 \in (0, \epsilon/2) \tag{2.58}$$

such that for all $(s_1, h_1), (s_2, h_2) \in F$ we have

$$|s_1 - s_2| + |h_1 - h_2| \leq \delta_1 \Rightarrow |f^\pm(s_1, h_1) - f^\pm(s_2, h_2)| \leq \xi/2. \tag{2.59}$$

Uniform continuity of $B_u$ implies there exists

$$\eta_2 < \frac{\delta_1}{1 + B} \tag{2.60}$$

such that for all $s_1, s_2 \in [a, b]$ we have

$$|s_1 - s_2| \leq \eta_2 \Rightarrow |B_u(s_1) - B_u(s_2)| \leq \delta_1. \tag{2.61}$$

Consider arbitrary discretization $D([a, b], (s_i)_{i=0}^n)$ with $\Delta(D) \leq \eta_2$. Let $(\hat{h}_i)_{i=0}^n$ be the sequence output by Algorithm 1. Define function $\tilde{h} : [a, b] \to \mathbb{R}$ via $\tilde{h}(s_i) = \hat{h}_i - \delta_1$ for all $0 \leq i \leq n$, and

$$\tilde{h}(s) = \frac{s - s_i}{s_{i+1} - s_i}\tilde{h}(s_{i+1}) + \frac{s_{i+1} - s}{s_{i+1} - s_i}\tilde{h}(s_i) \tag{2.62}$$

for all $s \in [s_i, s_{i+1}]$ and $0 \leq i \leq n - 1$. By construction, $\tilde{h}$ is continuous. In fact, we show $\tilde{h} \in A_\xi$.

First, we will prove $\tilde{h} \leq B_u$. Consider any $0 \leq i \leq n - 1$. Since $\tilde{h}$ is linear on $[s_i, s_{i+1}]$, $\tilde{h}(s) \leq \max(\tilde{h}(s_i), \tilde{h}(s_{i+1}))$ for all $s \in [s_i, s_{i+1}]$. Thus, it suffices to show $\tilde{h}(s_i), \tilde{h}(s_{i+1}) \leq \min_{s \in [s_i, s_{i+1}]} B_u(s)$. To this end, consider arbitrary $s \in [s_i, s_{i+1}]$. Since $|s - s_i| \leq \eta_2$, Equation (2.61) implies $B_u(s) \geq B_u(s_i) - \delta_1$. Admissibility of $(\hat{h}_i)_{i=0}^n$

implies $B_u(s_i) \geq \hat{h}_i$ and so

$$B_u(s) \geq B_u(s_i) - \delta_1 \geq \hat{h}_i - \delta_1 = \tilde{h}(s_i). \tag{2.63}$$

Since $s$ was arbitrary, we obtain $\tilde{h}(s_i) \leq \min_{s\in[s_i,s_{i+1}]} B_u(s)$. The corresponding inequality for $\tilde{h}(s_{i+1})$ follows analogously, and so $\tilde{h} \leq B_u$ holds.

Next, we show $D^+\tilde{h}(s) \leq f^+(s, \tilde{h}(s))+\xi$ for all $s \in [a, b)$. Again, consider arbitrary $0 \leq i \leq n-1$ and $s \in [s_i, s_{i+1})$. We have

$$D^+\tilde{h}(s) = \frac{\tilde{h}(s_{i+1}) - \tilde{h}(s_i)}{s_{i+1} - s_i} = \frac{\hat{h}_{i+1} - \hat{h}_i}{s_{i+1} - s_i} \leq f^+(s_i, \hat{h}_i). \tag{2.64}$$

Also,

$$\begin{aligned} |s - s_i| + |\tilde{h}(s) - \tilde{h}(s_i)| &= |s - s_i| \left(1 + \left|\frac{h_{i+1} - h_i}{s_{i+1} - s_i}\right|\right) \\ &\leq |s - s_i|(1 + B) \leq \delta_1 \end{aligned} \tag{2.65}$$

where the first equality follows from linearity of $\tilde{h}$ on $[s_i, s_{i+1}]$, and the second inequality from admissibility of $(\hat{h}_i)_{i=0}^n$ and the fact $|f^\pm| \leq B$. Equations (2.59) and (2.65) imply

$$f^+(s_i, \tilde{h}(s_i)) \leq f^+(s, \tilde{h}(s)) + \frac{\xi}{2}. \tag{2.66}$$

Similarly, $|\tilde{h}(s_i) - \hat{h}_i| \leq \delta_1$ implies

$$|f^+(s_i, \hat{h}_i) - f^+(s_i, \tilde{h}(s_i))| \leq \frac{\xi}{2}. \tag{2.67}$$

Combining the latter pair of inequalities, we derive

$$D^+\tilde{h}(s) \leq f^+(s, \tilde{h}(s)) + \xi. \tag{2.68}$$

Similarly,

$$D^-\tilde{h}(s) \geq f^-(s, \tilde{h}(s)) - \xi, \tag{2.69}$$

and so we obtain $\tilde{h} \in A_\xi$. As a result, by definition of $\overline{h}_\xi$, we have $\tilde{h} \leq \overline{h}_\xi$. This

implies for every $0 \leq i \leq n$

$$h_i = \tilde{h}(s_i) + \delta_1 \leq \overline{h}_\xi(s_i) + \delta_1 \leq \overline{h}(s_i) + \frac{\epsilon}{2} + \delta_1 \leq \overline{h}(s_i) + \epsilon \tag{2.70}$$

where the last inequality follows from Equation (2.58). This completes the second, and final, part of the proof of the theorem.

□

## 2.5 Simulation Results

In this section, we present a numerical example of the performance of Algorithm 1 on one instance of the TOPP problem that would be solved optimally by CO approaches. To this end, let $\gamma$ be a planar path formed by concatenating a line segment of length $l_1 = 20$ and two semicircular paths with radii $R_1 = 12$ and $R_2 = 4$. In what follows, numerical bounds on actuation values have been set in such a way that the direction (up/down) of the binormal to the two curved segments of the path makes no difference.

Set $||\mathbf{a}||_2 \leq 2$, $v_{max} = 5$, and assume, for the sake of simplicity, these constraints automatically imply bounds on $|\omega|$, $|\tau|$ and $|a_{hor}|$ are satisfied (e.g. by choosing them suitably large post hoc). In this case, we can compare the solution of Algorithm 1 with the optimum $\overline{h}$ which admits a semi-analytic form:

$$\overline{h}(s) = \begin{cases} v_{max}^2 & s \in [0, s_0] \\ v_{max}^2 - 2a_{max}(s - s_0) & s \in [s_0, l_1] \\ R_1 a_{max} & s \in [l_1, s_1] \\ R_1 a_{max} \cos\left(\frac{2}{R_1}(s - s_1)\right) & s \in [s_1, l_1 + \pi R_1] \\ R_2 a_{max} & s \geq l_1 + \pi R_1, \end{cases} \tag{2.71}$$

where $s_0 = l_1 - \frac{v_{max}^2 - R_1 a_{max}}{2a_{max}}$, and $s_1 = l_1 + \pi R_1 - \frac{R_1}{2}\left(\frac{\pi}{2} - \arcsin(\frac{R_2}{R_1})\right)$. Empirical evidence of asymptotic optimality of Algorithm 1 may be observed in Figure 2-6. A glance at Figure 2-6 might lead one to suspect that the rate of convergence of

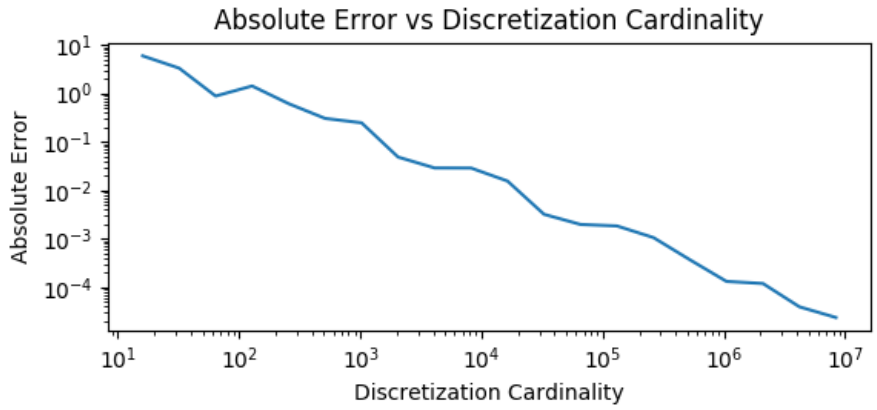



Figure 2-6: Empirical Evidence of Asymptotic Optimality of Algorithm 1

Algorithm 1 satisfies

$$\underbrace{\rho}_{error} \propto |D|^{-1}. \tag{2.72}$$

This would certainly be true in the special case of $f^+ \equiv f^-$ and $B_u \equiv \infty$, as this would then be restating the known rate of convergence of the Forward Euler method for integrating ordinary differential equations. The latter would amount to a single forward pass of Algorithm 1. However, our problem is more involved. We suspect that the rate of convergence of Algorithm 1 predicted by Equation (2.72) is true, but proving it would involve further analysis.

## 2.6 Conclusion

In this chapter we characterized the solution to the TOPP problem for a wide range of agents with second order dynamics. The first novelty was a characterization of the optimal solution, which comes about naturally when the space of decision variables and constraints imposed on them are defined in a suitable way. We then demonstrated the asymptotic optimality of a recently proposed computationally-efficient, robust algorithm for the TOPP problem.

# Chapter 3

# Perception-aware Time Optimal Path Parametrization for Quadrotors

## 3.1 Introduction

State estimation is a critical building block of the software stack enabling autonomous quadrotors. Due to their small size and limited power budget, they often rely on a combination of an inertial measurement unit (IMU) and camera for ego-localization. Information fusion from the two sensor modalities is crucial for the task to succeed. Sole reliance on noisy, biased measurements of acceleration and angular velocity supplied by the IMU results in state estimates that quickly yield to drift. The camera can significantly reduce such errors, provided there is a sufficient quantity of visual cues in the environment. Such cues can include points [17], lines [30], or even whole objects [54].

Due to varying lighting conditions, occluding obstacles, and spatial variation of textured surfaces, some trajectories allow for better localization accuracy than others. This chapter deals with a particular problem in the area of perception-aware trajectory optimization for quadrotors [1, 14, 47, 66]. Conceptually, we solve an optimization problem for which the objective is minimizing execution time of a geometric path subject to hard actuation bounds of the quadrotor, as well as perception constraints necessary for the vehicle to maintain accurate state estimates at all times [76].

Even without perception constraints, trajectory planning for quadrotors is a computationally challenging task. The landmark paper [45] established differential flatness [23] of quadrotor dynamics, showing that the trajectories of four flat outputs consisting of positions and heading angles of the vehicle are functionally independent, and furthermore, uniquely determine the trajectory of states and inputs in which they are embedded. This result allowed planning in a four instead of the ambient twelve dimensional state space of the quadrotor. Additionally, the same work initiated a line of papers using piecewise polynomials to represent trajectories of flat outputs. For example, [45] designed trajectories that visit a set of waypoints at given times by solving a quadratic program. The efficiency of such an approach was later improved in [62] with a clever reparametrization of flat outputs. Adopting a somewhat different paradigm using randomized motion primitives, [27, 48] used the result of [45] for real-time planning of dynamically feasible motions, using the polynomial representation to quickly pick promising trajectories from a large set of randomly sampled ones. The graph-based approach of [42] performed a search in a subspace of flat outputs to generate collision-free trajectories which minimize a linear combination of execution time and the magnitude of jerk.

Numerous works have devised approximate solutions to variations of the trajectory planning problem with perception constraints. For example, [58] planned locally optimal minimum time trajectories between two states by optimizing a family of flat trajectories parametrized by b-splines. Relatedly, [69] designed trajectories that finish at a given position, while minimizing the length of the path traced out by projections of landmarks desired to lie in the final field of view. The work of [19] adopted an MPC framework to incorporate a soft perception constraint minimizing the speed of projections of specified landmarks in the environment while navigating from start to goal position. Some of the most recent works, such as [49], included even more sophisticated perception constraints that optimize the co-visibility of given landmarks between consecutive keyframes, while also minimizing execution time. The elegant work of [80] leveraged the Hopf fibration to plan perception-aware trajectories for multirotors, albeit with simplified dynamic constraints.

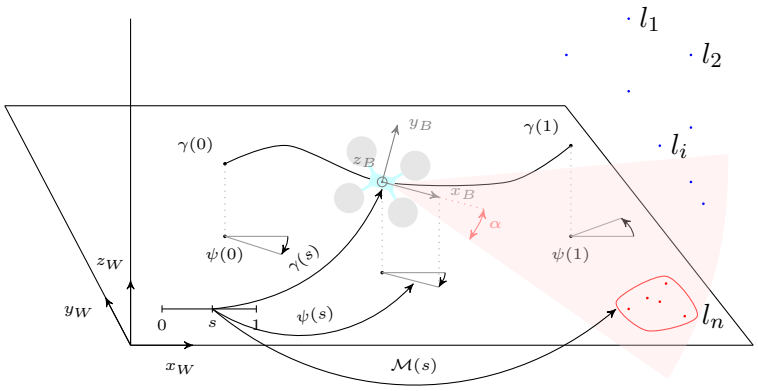



Figure 3-1: Quadrotor tasked with following a user-given path while keeping specified landmarks (in the set $\mathcal{M}(s)$) in field of view (shown in the red cone).

## Contributions

The main contribution of this chapter lies in unravelling the hidden structure of a whole class of perception-aware motion planning problems for quadrotors equipped with a forward-facing camera. In particular, we show that various constraints requiring the agent to maintain a sufficient amount of information within field of view of its onboard camera induce convex (in fact linear) constraints in the resulting TOPP problem. We also show that for a certain, perhaps somewhat restricted, layout of landmarks in the environment, our method can select to have within view subsets of landmarks with a desired amount of information, so as to induce the shortest execution time of the path. Ultimately, we show how to losslessly compress the number of perception constraints by at least an order of magnitude, in an efficient way that is further amenable to parallelized implementation. This is particularly important when applying interior-point methods (IPMs) which typically add a Lagrange multiplier for every constraint.

Finally, we use constructed perception constraints in a competitive solution for the non-convex perception-aware TOPP problem for quadrotors with *hard bounds on thrusts of its individual motors*. We compare the performance of our algorithm to a strong, to the best of our knowledge novel, baseline method based on a general-purpose IPM that takes order of magnitude longer to compute.

## 3.2 Problem Statement

This section provides a mathematical formulation of two related classes of perception-aware time-optimal path parametrization (PA-TOPP) problems. After laying out the dynamics and sensing models of a typical quadrotor agent, we give details of the path-traversal task specification in addition to dynamic constraints imposed on the vehicle. Ultimately, we furnish two classes of perception constraints, each corresponding to its particular PA-TOPP problem we consider.

### 3.2.1 Dynamics Model

We adopt a common framework specifying the rigid body dynamics of a quadrotor. Its state, $\mathbf{q} \in SE(3) \times T_{SE(3)}$, consists of its position $\mathbf{x} \in \mathbb{R}^3$ and orientation $R \in SO(3)$ with respect to an inertial world frame, as well as its translational velocity $\mathbf{v} \in \mathbb{R}^3$ and angular velocity $\omega \in \mathbb{R}^3$. Inputs of the agent are the four thrusts $u_1, u_2, u_3, u_4 \in \mathbb{R}$ of its individual motors, giving rise to the following equation of motion

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{R}} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{g} \\ \mathbf{R}[\omega\times] \\ -\mathbf{J}^{-1}(\omega \times \mathbf{J}\omega) \end{bmatrix} + \begin{bmatrix} 0 \\ R\mathbf{e}_3\ c \\ 0 \\ \mathbf{J}^{-1}\ \tau \end{bmatrix}, \tag{3.1}$$

in which $\mathbf{g} \in \mathbb{R}^3$ represents the gravitational acceleration with respect to the world frame, $[\omega\times] := [\omega \times \mathbf{e}_1 \mid \omega \times \mathbf{e}_2 \mid \omega \times \mathbf{e}_3] \in \mathbb{R}^{3\times 3}$, and the *cumulative thrust* $c \in \mathbb{R}$ and *resultant torque* $\tau \in \mathbb{R}^3$ are given by

$$\begin{bmatrix} c \\ \tau \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ -k_L & k_L & k_L & -k_L \\ -k_L & -k_L & k_L & k_L \\ -k_M & k_M & -k_M & k_M \end{bmatrix}}_{\text{fixed control matrix } \mathbf{F} \in \mathbb{R}^{4\times 4}} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}. \tag{3.2}$$

Last but not least, $\mathbf{J} \in \mathbb{R}^{3\times 3}$ represents the momement of inertia of the quadrotor.

## Characteristic Aspects of Quadrotor Dynamics

Trajectory optimization for quadrotors is challenging for several different reasons. Firstly, their nonlinear dynamics makes it difficult to adapt corresponding analytical results available for linear systems with quadratic or minimum-time objectives. Second, their twelve-dimensional state space precludes grid-based methods that are based on a naive discretization of allowed ranges of individual components of their state. Ultimately, quadrotors live in a non-Euclidean state space; even numerically integrating their state forward in time for a fixed trajectory of inputs requires care.

The landmark paper of [45] established *differential flatness* of quadrotor dynamics. Intuitively, this means there exists a vector function of time derivatives of states and inputs of the quadrotor, which my be specified independently of one another, and whose time derivatives yield the underlying trajectory of states and inputs required to induce them. In particular, for quadrotors such a vector function consists of two interpretable quantities: the position, $\mathbf{x}(t)$, and the heading angle, $\psi(t)$, of the vehicle.

For the purpose of this chapter, it is particularly important to recall the following part of the result of [45]. Indeed, from Equation (3.1), the acceleration of the vehicle is related to its orientation via

$$\underbrace{R\mathbf{e}_3}_{\mathbf{z}_B} \; c = \ddot{\mathbf{x}} - \mathbf{g} \quad \Rightarrow \quad \begin{cases} c = ||\ddot{\mathbf{x}}(t) - \mathbf{g}||_2 \\ \mathbf{z}_B(t) = \frac{\ddot{\mathbf{x}}(t) - \mathbf{g}}{c}. \end{cases} \tag{3.3}$$

If we define the perpendicular to the heading vector as follows

$$\psi^{\perp}(t) := \mathbf{e}_3 \times [\cos(\psi(t)),\ \sin(\psi(t)),\ 0]^T, \tag{3.4}$$

we also get

$$\mathbf{x}_B(t) = \frac{\psi^{\perp}(t) \times \mathbf{z}_B(t)}{||\psi^{\perp}(t) \times \mathbf{z}_B(t)||_2}. \tag{3.5}$$

Combined with the relation $\mathbf{y}_B(t) = \mathbf{z}_B(t) \times \mathbf{x}_B(t)$, we get that the orientation of the quadrotor, $\mathbf{R}(t)$, is fully specified by $\mathbf{x}(t)$, $\dot{\mathbf{x}}(t)$, $\ddot{\mathbf{x}}(t)$, and $\psi(t)$. Differentiating two more times and using Equation (3.1), [45] show there exists a smooth function $\Phi$ so

that for generic, sufficiently smooth trajectories $\mathbf{x}(\cdot)$ and $\psi(\cdot)$, we have

$$\begin{bmatrix} \mathbf{q}(t) \\ \mathbf{u}(t) \end{bmatrix} = \Phi\left(\left(\frac{d^n}{dt^n}\mathbf{x}(t)\right)_{n=0,1,2,3,4}, \left(\frac{d^n}{dt^n}\psi(t)\right)_{n=0,1,2}\right). \tag{3.6}$$

An important corollary of the flatness property, first explicitly used in [62], is that every sufficiently smooth trajectory of positions and heading angles can be "slowed down" so that it is dynamically feasible for the quadrotor, given the sensible assumption that it can operate in "hover" mode. Slowing down a trajectory mathematically amounts to simply dilating its interval of definition in the time domain by the appropriate factor, whereas a hover mode is any state $\mathbf{q}$ which is an equilibrium state for Equation (3.1) for inputs $u_i = \frac{||\mathbf{g}||_2}{4}\ \forall 1 \leq i \leq 4$. We shall also adopt the latter assumption for the remainder of the chapter.

### 3.2.2 Sensing Model

We assume the quadrotor is equipped with a monocular forward-facing camera, a common setup for micro aerial vehicles operating in indoor environments. Geometrically, this means the body $x$-axis of the vehicle is aligned with the $z$(i.e. optical)-axis of the camera. The latter is modelled using the pinhole projection principle [79]. In particular, a point with coordinates $\mathbf{p}^{\mathcal{C}}$ with respect to the camera frame, is registered as a point

$$\left(\frac{\mathbf{p}_1^{\mathcal{C}}}{\mathbf{p}_3^{\mathcal{C}}}, \frac{\mathbf{p}_2^{\mathcal{C}}}{\mathbf{p}_3^{\mathcal{C}}}\right) \tag{3.7}$$

provided it lies inside the field of view (FOV) of the camera. We model the latter as a symmetric circular cone i.e. a point $\mathbf{p} \in \mathbb{R}^3$ lies within the FOV if and only if

$$\mathbf{p}_3^{\mathcal{C}} \geq 0,\ \frac{1}{\mathbf{p}_3^{\mathcal{C}}}\left|\left|\begin{bmatrix} \mathbf{p}_1^{\mathcal{C}} \\ \mathbf{p}_2^{\mathcal{C}} \end{bmatrix}\right|\right|_2 \leq \tan\alpha,$$

where $\alpha \in (0, \frac{\pi}{2})$ denotes the extent of the FOV around the optical axis.

### 3.2.3 Task: Traversing a "Flat" Path in Minimum Time

The objective of our problems will lie in minimizing the traversal time of a specified path of flat outputs

$$(\gamma, \psi) : [0, S_{end}] \to \mathbb{R}^3 \times S^1 \tag{3.8}$$

by a quadrotor with dynamics described in Section 3.2.1 under various dynamic and perception constraints. As in Section 2.2.1, the decision variable of interest will be the square speed profile $h : s \mapsto \left(\frac{ds}{dt}(s)\right)^2$. Using the relation $\frac{d}{dt} = \sqrt{h(s)}\frac{d}{ds}$, we may see that we can express the first four (two) temporal derivatives (i.e. wrt $t$) of $\mathbf{x}$ ($\psi$) in terms of the first three spatial derivatives (i.e. wrt $s$) of $h$. Using the flat transform in Equation (3.6), for an appropriately chosen smooth function $\Phi_{\gamma\times\psi}$, we can therefore write

$$\begin{bmatrix} \mathbf{q}(s) \\ \mathbf{u}(s) \end{bmatrix} = \Phi_{\gamma\times\psi}\left(\left(\frac{d^i h}{ds^i}(s)\right)_{i=0,1,2,3}\right) \quad \forall s \in [0, S_{end}]. \tag{3.9}$$

### 3.2.4 Dynamic Constraints

In addition to dynamic constraints in the form of thrust bounds

$$0 \leq u_i \leq u_{max} \quad \forall 1 \leq i \leq 4, \tag{3.10}$$

we shall assume we are given a path

$$(\mathbf{n}, \beta) : [0, S_{end}] \to S^2 \times \left[0, \frac{\pi}{2}\right) \tag{3.11}$$

of unit vectors and angles, denoting constraints that the body $z$ axis of the quadrotor should form an angle at most $\beta(s)$ with vector $\mathbf{n}(s)$ when its center of mass is located at position $\gamma(s)$. For example, the latter requirement arises in planning trajectories in which the agent, modelled as an oblate spheroid, has to pass through narrow gaps [20]. Additionally, it can represent a constraint preventing the vehicle from tipping over, and thus not encountering the singularity in the differentially flat representation when its $\mathbf{z}_B$ axis is parallel to the horizontal world plane.

### 3.2.5 Perception Constraints

The core theme of this chapter lies in perception constraints. They require the quadrotor to maintain a sufficient amount of visual information within the field of view of its camera so as to be able to localize itself accurately. We shall state two slightly different types of perception constraints. From a high-level view, the first will involve a nonlinear optimization problem with continuous decision variables only, whereas the second, more general one, will introduce additional discrete selection variables.

We assume there is a set

$$\mathcal{L} = \{l_1, l_2, ..., l_N\} \tag{3.12}$$

of $N$ landmarks in the environment of the agent with known, fixed, positions with respect to the world frame. With the setup of a forward-facing camera, a landmark will lie within view precisely when its displacement from the optical center of the camera, assumed to lie at the center of mass of the quadrotor (a negligible approximation), makes an angle at most $\alpha$ with the $\mathbf{x}_B$ axis of the vehicle. Ultimately, we are given a set function

$$\mathcal{M} : [0, S_{end}] \to 2^{\mathcal{L}} \tag{3.13}$$

that takes on slightly different meanings for the two classes of perception constraints we consider.

The first type of constraint, we shall refer to as the *conjunctive* constraint for reasons that will be clarified momentarily, is as follows. At any point $s \in [0, S_{end}]$ along the flat path (see Equation (3.8)), the quadrotor must have all landmarks in the set $\mathcal{M}(s)$ within FOV of its camera. The nomenclature is clear: the agent/planner has no freedom in selecting which subset of landmarks to have within FOV, but merely how navigate as quickly as possible while adhering to stipulated constraints.

The second type of constraint, we call the *disjunctive* constraint, aims to relax the potentially restrictive nature of conjunctive constraints. In this case, for every $s \in [0, S_{end}]$, the set $\mathcal{M}(s)$ represents a set of candidate landmarks from which the agent is additionally allowed to *choose* the subset that contains a sufficient amount of visual information, while offering the agent the opportunity to execute the given path

in less time. To measure the amount of visual information, we introduce a general set function

$$\mathcal{F} : \{(s, V) \mid s \in [0, S_{end}], V \subseteq \mathcal{M}(s)\} \to \mathbb{R} \tag{3.14}$$

specifying the utility of having the collection of landmarks in set $V(s)$ within FOV when the agent is at point $s$ along the path. With the threshold function

$$\lambda_{min} : [0, S_{end}] \to \mathbb{R} \tag{3.15}$$

encoding the minimum acceptable utility, and denoting the subsets of landmarks within field of view by

$$V : [0, S_{end}] \to \mathcal{L}, \tag{3.16}$$

the new perception constraint reads

$$\mathcal{F}(s, V(s)) \geq \lambda_{min}(s) \quad \forall s \in [0, S_{end}]. \tag{3.17}$$

Beyond the minimal requirement that $\mathcal{F}$ be a pointwise increasing set function in that

$$\forall s \in [0, S_{end}] \quad A \subseteq B \subseteq \mathcal{M}(s) \;\Rightarrow\; \mathcal{F}(s, A) \leq \mathcal{F}(s, B), \tag{3.18}$$

we place no other restrictions on $\mathcal{F}$. The monotonicity assumption is a rather natural one, intuitively saying that a larger number of landmarks within FOV allows higher localization accuracy. We have purposely kept the details of the function $\mathcal{F}$ hidden for now. Section 3.4.2 will hilight two natural choices for $\mathcal{F}$ that are useful in vision-driven navigation.

### 3.2.6 Mathematical Formulation

We finally put together previous elements of the PA-TOPP problem for quadrotors. To avoid cluttered notation, we separately state the subclass of problems with conjunctive constraints and those with disjunctive constraints.

**PA-TOPP Problem with Conjunctive Constraints**

$$
\begin{aligned}
&\min_{h:[0,S_{end}]\to[0,\infty)} \int_0^{S_{end}} \frac{ds}{\sqrt{h(s)}} \\
&s.t.\ \forall s \in [0, S_{end}]: \\
&\quad \begin{bmatrix} \mathbf{q}(s) \\ \mathbf{u}(s) \end{bmatrix} = \Phi_{\gamma\times\psi}\left(\left(\frac{d^i h}{ds^i}(s)\right)_{i=0,1,2,3}\right) \\
&\quad \angle(\mathbf{z}_B(s), n(s)) \le \beta(s) \\
&\quad 0 \le \mathbf{u}(s) \le u_{max} \\
&\quad h(s) \le B_u(s) \\
&\quad \angle(\mathbf{x}_B(s), l_i - x(s)) \le \alpha \quad \forall i \in \mathcal{M}(s).
\end{aligned}
\tag{3.19}
$$

**PA-TOPP Problem with Disjunctive Constraints**

$$
\begin{aligned}
&\min_{h,V:[0,S_{end}]\to[0,\infty)\times 2^{\mathcal{L}}} \int_0^{S_{end}} \frac{ds}{\sqrt{h(s)}} \\
&s.t.\ \forall s \in [0, S_{end}]: \\
&\quad \begin{bmatrix} \mathbf{q}(s) \\ \mathbf{u}(s) \end{bmatrix} = \Phi_{\gamma\times\psi}\left(\left(\frac{d^i h}{ds^i}(s)\right)_{i=0,1,2,3}\right) \\
&\quad \angle(\mathbf{z}_B(s), n(s)) \le \beta(s) \\
&\quad 0 \le \mathbf{u}(s) \le u_{max} \\
&\quad h(s) \le B_u(s) \\
&\quad \angle(\mathbf{x}_B(s), l_i - x(s)) \le \alpha \quad \forall i \in V(s) \\
&\quad \mathcal{F}(s, V(s)) \ge \lambda_{min}(s).
\end{aligned}
\tag{3.20}
$$

Problem (3.19) with conjunctive constraints is challenging as it is a nonlinear programming problem with a potentially large number of perception constraints. In Problem (3.20), disjunctive constraints introduce a new layer of difficulty with a mixed integer component. Our algorithms will involve two layers of contributions; the first addressing forming a succint summary of perception constraints, and the second providing a way of dealing with the non-trivial resulting optimization problem.

## 3.3 Algorithm

In this section, we provide our algorithms for addressing Problems (3.19) and (3.20). We start by describing and motivating the way of transcribing aforementioned, *functional*, optimization problems into a form amenable to numerical manipulation.

We consider a discretization $D$ of points $0 = s_0 < s_1 < \cdots < s_N = S_{end}$. The square speed profile $h$ is represented as a third order integrator with constant input on each interval $[s_i, s_{i+1}]$; any such profile is thus a piecewise polynomial function of degree three with continuous second derivatives. One of the motivations for such a representation goes back to the differentially flat transform in Equation (3.6): a third order integrator is in general necessary to render the expression well-defined.

In broad strokes, our method works as follows. We first translate perception constraints into those on the square speed profile. Furthermore, we initially switch off given actuation bounds in favour of requiring the magnitude of the acceleration of the quadrotor to be at most the sum of maximal specific thrusts of its four motors, as well as including limits on the bank angle, preventing the vehicle from 'tipping over'. Enforcing such constraints at the discretization points, we solve the $l_2$-regularized time optimal path parametrization problem stated in Section 3.3.2. Finally, we linearly scale down the resulting square speed profile until all individual motor thrust bounds are satisfied. The following subsections describe of each of these stages in more detail.

### 3.3.1 Perception Constraint Generation

We form perception constraints at each discretization point in turn, noting however that they can be computed in parallel, independently of one another. For any landmark, at an arbitrary point along the path, we extract the interval of elevation angles of the body x axis of the quadrotor (the angle it forms with the horizontal world plane) for which the landmark lies in FOV.

Depending on the nature of the constraint, we then follow slightly different procedures. For the conjunctive constraint, we effectively seek the intersection of all such intervals, which we can accomplish with a single linear pass through their endpoints.

For the disjunctive constraint, we first sort the endpoints of all computed intervals, and then find the extremal angles which satisfy the constraint in a single linear pass.

Assuming the resulting interval of elevations at point $s$ is $[\theta_{min}, \theta_{max}]$, defining the *perception normals* via

$$\begin{aligned} \mathbf{p}_+(s) &:= (\cos(\theta_{max})\psi(s) + \mathbf{e}_3 \sin(\theta_{max})) \\ \mathbf{p}_-(s) &:= (\cos(\theta_{min})\psi(s) + \mathbf{e}_3 \sin(\theta_{min})) \end{aligned} \tag{3.21}$$

with $\psi(s) = [\cos(\psi(s)), \sin(\psi(s)), 0]^T$ being the heading vector, the equivalent constraint on the square speed profile $h$ is given by

$$\mathbf{c}(s) \cdot \mathbf{p}_+(s) \geq 0, \quad \mathbf{c}(s) \cdot \mathbf{p}_-(s) \leq 0 \tag{3.22}$$

where $\mathbf{c}(s) = \mathbf{R}\mathbf{e}_3 c$ is the total thrust from Equation (3.1).

### 3.3.2 Regularized TOPP

Since we assume the square speed profile to be a third order integrator: $h^{(3)}(s) = \rho(s)$, the new objective includes a time-optimality term and a regularization term penalizing the $l_2$ norm of the effective input $\rho$. Computed perception as well as bank angle constraints remain intact, while we substitute bounds on individual motor inputs using a simple upper bound on the norm of the total thrust - the sum of maximal thrusts of the four motors. Thus, for a suitably chosen hyperparameter $\lambda > 0$, using the discretization described at the beginning of the section, this stage solves:

$$\begin{aligned} &\underbrace{\int_0^{S_{end}} \frac{ds}{\sqrt{h(s)}}}_{\text{execution time}} + \underbrace{\lambda \int_0^{S_{end}} \rho(s)^2 ds}_{\text{regularization penalty}} \\ &s.t.\ \forall s \in [0, S_{end}]\ : \\ &\qquad ||\mathbf{c}(s)||_2 \leq 4u_{max} \quad \text{(thrust)} \\ &\qquad \mathbf{c}(s) \cdot \mathbf{n}(\mathbf{s}) \geq ||\mathbf{c}(s)||_2 \cos(\beta) \quad \text{(bank angle)} \\ &\qquad \mathbf{c}(s) \cdot \mathbf{p}_+(s) \geq 0,\ c(s) \cdot \mathbf{p}_-(s) \leq 0 \ \text{(perception)} \end{aligned} \tag{3.23}$$

We solve the discretized version of the latter problem using ECOS [18], a performant, robust interior point solver for second order conic programs. Its running time is $O(N \cdot N_{iters})$, where $O(N_{iters})$ accounts for the number of "gradient descent" steps.

### 3.3.3 Alpha Scaling

The final stage of the procedure finds the largest $\alpha \in [0, 1]$ such that the square speed profile $h_{sol} := \alpha h$ satisfies given motor thrusts. With the natural assumption that any hover state for the quadrotor is dynamically feasible, a strictly positive $\alpha$ is guaranteed to exist. Furthermore, any $\alpha \in [0, 1]$ is guaranteed not to violate the perception and bank angle constraints that have been satisfied by the procedure in section 3.3.2. This statement is a consequence of the convexity of such constraints, which will be elaborated in some detail in section 3.4. In practice, we find $\alpha$ by searching a uniformly discretized grid on $[0, 1]$. Denoting the number of candidates for $\alpha$ by $N_\alpha$, the latter procedure runs in time $O(N_\alpha N)$.

**Algorithm Summary**

**Algorithm 2:** PA-TOPP Algorithm

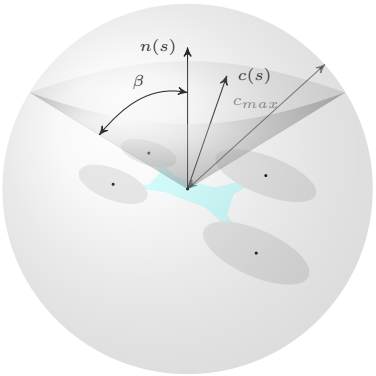
**Data:** $D = (s_i)_{i=0}^N$, $(\mathbf{l}_j)_{j=1}^n$, $u_{max}$, $\beta$, $\lambda$, $\mathcal{F}(\cdot, s_i)_{i=0}^N$, $\lambda_{min}(s_i)_{i=0}^N$, $(\gamma(\cdot), \psi(\cdot))$, $N_\alpha$
**Result:** $h \equiv (h|_{[s_i, s_{i+1}]} \in span\{1, s, s^2, s^3\})_{i=0}^{N-1}$
**for** $i = 1$ *to* $n$ **do**
  $\theta_{min} \leftarrow MinElevation(s_i, \mathcal{F}(\cdot, s_i), w_{min}(s_i))$
  $\theta_{max} \leftarrow MaxElevation(s_i, \mathcal{F}(\cdot, s_i), w_{min}(s_i))$
  $p_+^{(i)}, p_-^{(i)} \leftarrow AnglesToNormals(\theta_{min}, \theta_{max})$
**end**
$h \leftarrow RegTOPP((p_+^{(i)})_{i=0}^N, (p_-^{(i)})_{i=0}^N, u_{max}, \lambda, \beta)$
**for** $i = N_\alpha$ *to* $0$ **do**
  **if** $\underline{h} \leftarrow h\frac{i}{N_\alpha}$ *feasible* **then**
    return $\underline{h}$
  **end**
**end**

The running time of the whole algorithm is $O(N(n \log n + N_{iters} + N_\alpha))$ for disjunctive and $O(N(n + N_{iters} + N_\alpha))$ for conjunctive perception constraints.

## 3.4 Analysis

The key part of this section lies in demonstrating convexity of various perception as well as relaxations of dynamic constraints on the square speed profile.

### 3.4.1 Dynamic Constraints




Using the differentially flat transform in Equation (3.6), we have that the vector thrust of the vehicle is given by

$$\mathbf{c} = \ddot{\mathbf{x}} - \mathbf{g} \quad \Rightarrow \quad \mathbf{c}(s) = \frac{1}{2}\gamma'(s)h'(s) + \gamma''(s)h(s) - \mathbf{g} \quad \forall s \in [0, S_{end}]. \tag{3.24}$$

Since $\gamma(\cdot)$ is fixed, the latter represents an affine operator on $h$. In particular, any convex membership constraints on $\mathbf{c}(\cdot)$ will translate into convex membership constraints on $h$. Examples of such constraints relevant to Problems (3.19) and (3.20) include bounds on the magnitude of total thrust

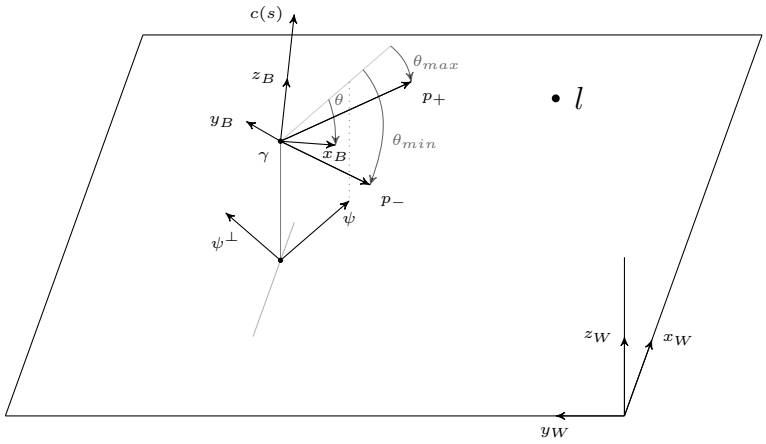
$$||\mathbf{c}(s)||_2 \leq c_{max} \quad \Leftrightarrow \quad \underbrace{\left|\left|\frac{1}{2}\gamma'(s)h'(s) + \gamma''(s)h(s) - \mathbf{g}\right|\right|_2}_{\text{convex in } h} \leq c_{max} \tag{3.25}$$

as well as bank angle constraints

$$\begin{gathered} \angle(\mathbf{c}(s), \mathbf{n}(s)) \leq \beta(s) \quad \Leftrightarrow \\ \underbrace{\left(\frac{1}{2}\gamma'(s)h'(s) + \gamma''(s)h(s) - \mathbf{g}\right) \cdot \mathbf{n}(s)}_{\text{affine in } h} \geq \underbrace{\left|\left|\frac{1}{2}\gamma'(s)h'(s) + \gamma''(s)h(s) - \mathbf{g}\right|\right|_2 \cos\beta(s)}_{\text{convex in } h}. \end{gathered} \tag{3.26}$$

### 3.4.2 Perception Constraints

The aim of this subsection is to derive relations induced by perception constraints on the square speed profile. We shall begin by considering implications of requiring only a single landmark to lie within the FOV at just one point along the path, followed by an argument behind the convexity of such a constraint. After that, we will describe theoretical and application-orientated extensions towards multiple, spatially-varying sets of landmarks. In what follows, the angle between the $\mathbf{x}_B$ axis of the quadrotor and its projection onto the horizontal world plane will be referred to as the elevation angle.




**Theorem 6.** *Let $\gamma$ and $\psi$ be the position and heading angle of the quadrotor with FOV angle $\alpha$. Then, a landmark with position $l$ lies within the FOV for an interval of elevation angles $\theta$, say $[\theta_{min}, \theta_{max}]$, which might be empty.*

*Proof.* With slight abuse of notation, denote the heading vector of the vehicle by $\psi = [\cos(\psi), \sin(\psi), 0]^T$, and let the unit vector parallel to the displacement from its center of mass to the landmark be

$$\hat{\mathbf{d}} := \frac{\mathbf{l} - \gamma}{||\mathbf{l} - \gamma||_2}.$$

Since $\mathbf{x}_B = \psi\cos(\theta) + \mathbf{e}_3\sin(\theta)$, we have $\mathbf{l} \in \text{FOV}$ if and only if:

$$\begin{aligned} \angle(\mathbf{x}_B, \hat{\mathbf{d}}) = \angle(\psi\cos(\theta) + \mathbf{e}_3\sin(\theta), \hat{\mathbf{d}}) &\leq \alpha \\ \Leftrightarrow [\psi\cos(\theta) + \mathbf{e}_3\sin(\theta)] \cdot \hat{\mathbf{d}} &\geq \cos(\alpha) \\ \Leftrightarrow [\psi \cdot \hat{\mathbf{d}}]\cos(\theta) + [\mathbf{e}_3 \cdot \hat{\mathbf{d}}]\sin(\theta) &\geq \cos(\alpha). \end{aligned} \tag{3.27}$$

Define $R = \sqrt{[\psi \cdot \hat{\mathbf{d}}]^2 + [\mathbf{e}_3 \cdot \hat{\mathbf{d}}]^2}$. We distinguish two cases:

1. $R < \cos(\alpha)$. In this case, the landmark lies too far aside, and the interval in question is empty.

2. $R \geq \cos(\alpha)$. Now the constraint becomes

$$\begin{aligned} \frac{\psi \cdot \hat{\mathbf{d}}}{R}\cos(\theta) + \frac{\mathbf{e}_3 \cdot \hat{\mathbf{d}}}{R}\sin(\theta) &\geq \frac{\cos(\alpha)}{R} \\ \Leftrightarrow \angle(\underbrace{\left[\frac{\psi \cdot \hat{\mathbf{d}}}{R}, \frac{\mathbf{e}_3 \cdot \hat{\mathbf{d}}}{R}\right]}_{\text{unit vector}}, [\cos(\theta), \sin(\theta)]) &\leq \arccos\left(\frac{\cos(\alpha)}{R}\right) \end{aligned} \tag{3.28}$$

   Recall we seek the set of $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ satisfying the equation above. To this end, we define

$$\Delta := \arccos\left(\frac{\cos(\alpha)}{R}\right) \in [0, \frac{\pi}{2}),$$

   and

$$\theta_c := atan2\left(\frac{\mathbf{e}_3 \cdot \hat{\mathbf{d}}}{R}, \frac{\psi \cdot \hat{\mathbf{d}}}{R}\right) \in (-\pi, \pi].$$

   We distinguish two sub-cases:

   (a) If $\theta_c > \frac{\pi}{2} + \Delta$ or $\theta_c < -\frac{\pi}{2} - \Delta$, then the interval in question is empty - the landmark lies behind the possible fields of view.

   (b) Otherwise, the interval of feasible elevation angles $\theta$ is

$$[B(\theta_c - \Delta), B(\theta_c + \Delta)], \quad \text{where} \quad B(\theta) := -\frac{\pi}{2} \vee (\frac{\pi}{2} \wedge \theta).$$

□

**Theorem 7.** *For an arbitrary point along the path, say corresponding to parameter $s \in [0, S_{end}]$, the requirement that the elevation angle of the body x axis of the quadrotor lies in interval $[\theta_{min}, \theta_{max}] \subseteq [-\frac{\pi}{2}, \frac{\pi}{2}]$ induces a convex constraint on the square speed profile $h$.*

*Proof.* As noted in Equation (3.24), the total thrust vector $\mathbf{c}(s)$ at a given point is an affine function of the square speed profile. Given $\mathbf{c}(s)$, the body z and x axes of the vehicle are

$$\begin{aligned} \mathbf{z}_B(s) &= \frac{\mathbf{c}(s)}{||\mathbf{c}(s)||_2} \\ &\text{and} \\ \mathbf{x}_B(s) &= \frac{\psi^{\perp}(s) \times \mathbf{c}(s)}{||\psi^{\perp}(s) \times \mathbf{c}(s)||_2}, \end{aligned} \tag{3.29}$$

respectively, where $\psi^{\perp} := \mathbf{e}_3 \times \psi(s)$. The elevation angle of the body x axis lies in interval $[\theta_{min}, \theta_{max}]$ precisely when $\mathbf{x}_B(s)$ belongs to the (pointed) two dimensional convex cone $\mathcal{K}$ delimited by rays

$$\begin{aligned} \eta_{max} &= \{t\ (\psi(s)\cos(\theta_{max}) + \mathbf{e}_3 \sin(\theta_{max})) \mid t \geq 0\}, \\ &\text{and} \\ \eta_{min} &= \{t\ (\psi(s)\cos(\theta_{min}) + \mathbf{e}_3 \sin(\theta_{min})) \mid t \geq 0\}. \end{aligned} \tag{3.30}$$

Hence, the constraint is equivalent to requiring

$$\frac{\psi^{\perp}(s) \times \mathbf{c}(s)}{||\psi^{\perp}(s) \times \mathbf{c}(s)||_2} \in \mathcal{K} \ \Leftrightarrow\ \psi^{\perp}(s) \times \mathbf{c}(s) \in \mathcal{K} \tag{3.31}$$

where the latter equivalence follows from the definition of a cone. Since $\psi^{\perp}(s) \times \mathbf{c}(s)$ is an affine function of the square speed profile, and preimages of convex sets under affine maps are convex, the result follows.

Finally, note that Condition (3.31) can be unravelled further. Indeed, $\psi^{\perp}(s)\times$ may be interpreted as a projection onto the orthogonal complement of $\psi^{\perp}(s)$ followed by a rotation around $\psi^{\perp}(s)$ by $\pi/2$. Thus, we get that Condition (3.31) is equivalent to requiring that the projection of $\mathbf{c}(s)$ onto the plane orthogonal to $\psi^{\perp}(s)$ lies in the

angle delimited by rays $\eta_{min}$ and $\eta_{max}$ rotated by $-\pi/2$ around $\psi^{\perp}(s)$. The latter is in turn equivalent to

$$\mathbf{c}(s) \cdot \mathbf{p}_{+}(s) \geq 0 \quad \text{and} \quad \mathbf{c}(s) \cdot \mathbf{p}_{-}(s) \leq 0,$$

where we recall the definitions

$$\mathbf{p}_{+}(s) = (\cos(\theta_{max})\psi(s) + \mathbf{e}_3 \sin(\theta_{max}))$$
$$\mathbf{p}_{-}(s) = (\cos(\theta_{min})\psi(s) + \mathbf{e}_3 \sin(\theta_{min}))$$

from Equation (3.21) for the sake of completeness. □

Equipped with Theorems 6 and 7, we are in a position to derive a key structural property of the PA-TOPP Problem (3.19) with conjunctive perception constraints.

**Theorem 8.** *The perception constraints in Problem (3.19) induce a convex constraint on the square speed profile.*

*Proof.* Consider a point $s \in [0, S_{end}]$. As before, let $\theta$ denote the elevation angle of the quadrotor. According to Theorem 6, for any $i \in \mathcal{M}(s)$, there exists an interval $[\theta_{min}^{(i)}, \theta_{max}^{(i)}]$ so that $\mathbf{l}_i$ lies within FOV when the quadrotor is at point $s$ along the path if and only if $\theta \in [\theta_{min}^{(i)}, \theta_{max}^{(i)}]$. In case the latter is empty, the task is automatically infeasible, and we henceforth disallow such situations.

As a result, all landmarks in $\mathcal{M}(s)$ lie inside the FOV exactly when

$$\theta \in \bigcap_{i \in \mathcal{M}(s)} [\theta_{min}^{(i)}, \theta_{max}^{(i)}] = [\max_{i \in \mathcal{M}(s)} \theta_{min}^{(i)}, \min_{i \in \mathcal{M}(s)} \theta_{max}^{(i)}]. \tag{3.32}$$

However, by Theorem 7, this amounts to a convex constraint on $h$. The same argument holds regardless of the value of $s \in [0, S_{end}]$: the perception constraint at each individual point $s$ is convex when expressed in terms of $h$. Since an arbitrary conjunction of convex constraints amounts to an intersection of convex feasible regions, which in turn is convex, the claim follows.

□

We now turn to several modelling implications of the statement above, as well as how it allows us to handle a wider class of perception constraints. In particular, the current statement of the problem involves a set of known, spatially-localized landmarks. This is a realistic assumption if the quadrotor is navigating environments thoroughly mapped in previous exploration missions, or if it is fully aware of its artificially structured surroundings such as when performing routine infrastructure monitoring tasks. Nevertheless, numerous missions require the agent to venture into the unknown. In such a scenario, aforementioned assumptions fail, and we now discuss how we might alleviate challenges that arise.

Roughly speaking, we now require having specified regions instead of just designated landmarks within field of view of the on-board camera. For example, consider a quadrotor exploring a new environment. It uses a learned depth estimation module to estimate positions of landmarks for future localization, and to which it initially only receives a bearing [43]. It forms corresponding confidence sets of their individual positions in the form of ellipses or polytopes, and requires those regions to stay within view while traversing a planned path. Alternatively, leveraging recent developments in semantic mapping and localization, we may wish the quadrotor to keep whole objects within its field of view [54]. We will now see that such generalized constraints come at no extra computational cost given the results we had previously derived.

**Theorem 9.** *Let $\mathcal{L} = \{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$ denote a set of $N$ regions in the environment with region $i$ specified as the convex hull of points in the finite set $\mathcal{S}_i$ for all $1 \leq i \leq N$. Then, an arbitrary map $\mathcal{M} : [0, S_{end}] \to 2^{\mathcal{L}}$ designating the indices of regions required to lie within field of view at each point along the path induces a convex constraint on $h$.*

*Proof.* Indeed, defining a set of fictive landmarks $\tilde{\mathcal{L}} = \bigcup_{1 \leq i \leq N} \mathcal{S}_i$, and a modified map $\tilde{\mathcal{M}} : [0, S_{end}] \to 2^{\tilde{\mathcal{L}}}$ defined by $\tilde{\mathcal{M}}(s) = \bigcup_{i \in \mathcal{M}(s)} \mathcal{S}_i$, the statement follows directly from Theorem 8. Here, we have implicitly used the observation that due to the convexity of the field of view, a set of points in space fully lies within it if and only if its convex hull does. □

Next, we uncover some of the structural properties of Problem (3.20) with disjunctive perception constraints. As promised at the end of Section 3.2.5, we shall start by providing two examples of utility functions $\mathcal{F}$. We will delineate natural situations in which these combinatorial, disjunctive constraints, induce a convex constraint on the square speed profile.

**Example.** One simple example of such a function $\mathcal{F}$ involves assigning a weight $w_i > 0$ to every landmark $i \in \mathcal{L}$, and defining

$$\mathcal{F}(s, V) = \sum_{i \in V} w_i, \quad V \subseteq S. \tag{3.33}$$

A "physical" interpretation of $w_i$ might involve visual discernability of the landmark or an estimate of the probability with which it can be tracked successfully.

**Example.** Another illustrative example is the following. In monocular visual-inertial navigation, it is typically more challenging to accurately estimate the translational rather than the rotational component of the pose of the vehicle. Consider, again, a scenario in which there are $N$ landmarks in the environment, with known positions with respect to the world frame. If we assume the agent can accurately estimate its orientation using the gyroscope, its camera may be assumed to yield a bearing measurement. We let

$$f(\mathbf{l}, \mathbf{x}) = \frac{\mathbf{l} - \mathbf{x}}{||\mathbf{l} - \mathbf{x}||_2} \ : \ \mathbb{R}^3 \times \mathbb{R}^3 \to S^2 \tag{3.34}$$

denote the function, giving the unit bearing vector from the vehicle with position $\mathbf{x}$ to a landmark with position $\mathbf{l}$. At any point $s$ along the path, the quadcopter can capture a noisy measurement

$$\underbrace{\mathbf{y}^{(i)}}_{f(\mathbf{l}^{(i)}, \gamma(s)) + \Delta\mathbf{y}^{(i)}} = f(\mathbf{l}^{(i)}, \underbrace{\mathbf{x}}_{\gamma(s) + \Delta\mathbf{x}}) + \underbrace{\epsilon^{(i)}}_{\text{meas. error}} \quad \forall i \in \mathcal{M}(s), \tag{3.35}$$

which upon linearization preceded by projection onto the tangent space of $S^2$ at each

of $f(\mathbf{l}^{(i)}, \gamma(s))$, with a slight abuse of notation, reads

$$\Delta \mathbf{y}^{(i)} = \frac{\partial f}{\partial x}(\mathbf{l}^{(i)}, \gamma(s))\ \Delta \mathbf{x} + \epsilon^{(i)} \quad \forall i \in \mathcal{M}(s). \tag{3.36}$$

Assuming, for the sake of simplicity, that $\epsilon^{(i)}$ are independent random variables distributed according to $N(0, \Sigma^{(i)})$, and an improper "flat" prior on $\Delta \mathbf{x}$, we have that its posterior distribution upon taking bearing measurements of landmarks in set $V \subseteq \mathcal{M}(s)$ is normally distributed with information (inverse covariance) matrix

$$\mathcal{I}(V) = \sum_{i \in V} \frac{\partial f}{\partial x}(\mathbf{l}^{(i)}, \gamma(s))^T (\Sigma^{(i)})^{-1} \frac{\partial f}{\partial x}(\mathbf{l}^{(i)}, \gamma(s)). \tag{3.37}$$

To minimize the error in our state estimate, we would like $\mathcal{I}(V)$ to be as "large" as possible. Since $\mathcal{I}(V) \in S_{3\times 3}$ is a matrix-monotone function of $V$, measures of its size can include determinant, trace, and even the smallest eigenvalue. The latter,

$$\mathcal{F}(s, V) := minEig(\mathcal{I}(V)) \tag{3.38}$$

is particularly appealing since it measures the inverse radius of the uncertainty ball capturing the position of the agent.

**Theorem 10.** *Assume that at any point $s$ along the path, there exists an elevation angle of the body $x$ axis of the quadrotor from which all the candidate landmarks in $\mathcal{M}(s)$ are within FOV. Then, constraint (3.17) induces a convex constraint on the square speed profile.*

*Proof.* By the argument of the proof of Theorem 8, it suffices to show that the set of feasible elevation angles at any particular point along the path is an interval. To this end, consider an arbitrary $s \in [0, S_{end}]$, and let $[\theta^{(i)}_{min}, \theta^{(i)}_{max}]$ be the interval of elevation angles for which landmark $i \in \mathcal{M}(s)$ is visible. By assumption,

$$\theta_0 \in \bigcap_{i \in \mathcal{M}(s)} [\theta^{(i)}_{min}, \theta^{(i)}_{max}]. \tag{3.39}$$

Noting the elementary fact that

$$\theta \ \mapsto \ \{i \in \mathcal{M}(s) | \theta \in [\theta_{min}^{(i)}, \theta_{max}^{(i)}]\} \tag{3.40}$$

is a decreasing (increasing) function on $[\theta_0, \frac{\pi}{2})$ $((-\frac{\pi}{2}, \theta_0])$ with respect to set inclusion, and noting that $\mathcal{F}(s, \cdot)$ is an increasing function, the claim follows. □

**Remark 1.** *The assumption that all candidate landmarks are visible from a single elevation angle at a given point along the path is somewhat strict. However, it will hold if, at every point, we restrict attention to candidate landmarks $\mathcal{M}(s)$ that are visible when the quadrotor is in a near-hover state.*

## 3.5 Simulation Results

We now turn to studying the numerical performance of critical aspects of our algorithm. We start by shedding light on the speed of computation/compression of perception constraints, before venturing to show the relative performance of our proposed algorithm for TOPP to a strong baseline method based on a general purpose constrained trust-region nonlinear optimization solver [7].

### 3.5.1 Perception Constraint Generation Simulations

In this section, we give numerical results on the running time of the perception constraint generation and compression procedure. We generate a simulated environment so that at every point of a densely discretized randomly chosen path of flat outputs, we sample a large quantity of landmarks that are designed to lie within view when the quadrotor is in a state of hover at the corresponding point along the path.

We perform timing results for two classes of constraints. The first involves the conjunctive formulation (Problem 3.19), requiring the quadroter to keep a designated set of landmarks within view. The second, disjunctive formulation (Problem 3.20) involves the quadrotor maintaining at least ten landmarks within field of view at all times - in our framework (Equation (3.17)), this amounts to choosing $\mathcal{F}(s, V) =$

$|V|$ and $\lambda_{min} \equiv 10$. The results for numbers of discretization points up to 240 in increments of 40, and numbers of candidate landmarks up to 300 in increments of 60 are given in Tables 3.1 and 3.2 and corresponding figures lying below them. All times are reported in *microseconds*, and each data point is the average of five runs.

| $N_{\text{ind}}$ \ $N_{\text{disc}}$ | 40 | 80 | 120 | 160 | 200 | 240 |
|---|---|---|---|---|---|---|
| 50 | 298 | 569 | 837 | 1089 | 1315 | 1580 |
| 100 | 562 | 1084 | 1559 | 2072 | 2600 | 2981 |
| 150 | 818 | 1569 | 2336 | 3075 | 3568 | 3955 |
| 200 | 1064 | 2097 | 3049 | 3468 | 4178 | 4875 |
| 250 | 1308 | 2557 | 3385 | 3915 | 4962 | 5828 |
| 300 | 1571 | 2784 | 3632 | 4803 | 5812 | 7010 |

Table 3.1: Conjunctive Constraint Generation [$\mu s$]

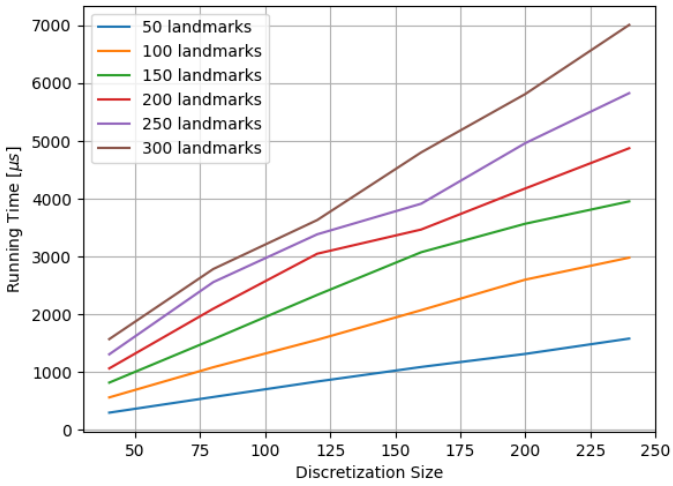



Figure 3-2: Timing Results for Conjunctive Constraint Generation

Plots in Figures 3-2 and 3-3 confirm that the running time of the two algorithms is almost linear in the number of discretization points. Furthermore, the perception constraints can be calculated independently for different points. Extraploting from graphs above, for a single discretization point, and as much as 300 candidate landmarks, the time required to compute perception constraints is tens of *microseconds*.

| $N_{\text{ind}}$ \ $N_{\text{disc}}$ | 40 | 80 | 120 | 160 | 200 | 240 |
|---|---|---|---|---|---|---|
| 50 | 648 | 1036 | 1510 | 1996 | 2351 | 3524 |
| 100 | 1078 | 2078 | 2980 | 4022 | 4783 | 5552 |
| 150 | 1624 | 3046 | 4755 | 5574 | 6712 | 7657 |
| 200 | 2139 | 4326 | 5575 | 6982 | 8345 | 10094 |
| 250 | 2724 | 4987 | 7206 | 8434 | 10158 | 12154 |
| 300 | 3237 | 6171 | 7914 | 10044 | 12203 | 14579 |

Table 3.2: Disjunctive Constraint Generation [$\mu s$]

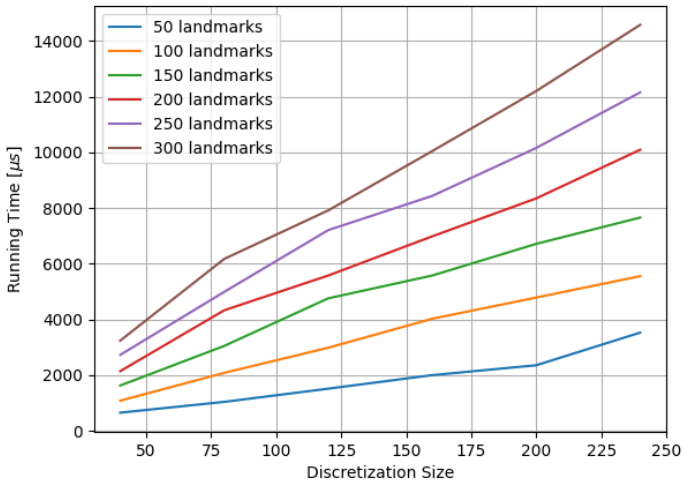



Figure 3-3: Timing Results for Disjunctive Constraint Generation

The practical significance of this result is that by using a highly parallelized architecture, such as a GPU, the running time of perception constraint generation would be reduced further. Nevertheless, even without such sophisticated implementations, the current one is able to compress a nonlinear optimization problem specificiation with tens of thousands of perception constraints into one with just a few hundred. Typically, nonlinear optimization solvers have running time complexity that scales super-linearly with the number of constraints. In these scenarios, our algorithm offers significant downstream computational savings at a neglible cost.

### 3.5.2 Comparison with Generic Nonlinear Optimization Baseline

| path length [m] | 8.3 | 8.3 | 8.4 | 9.7 |
|---|---|---|---|---|
| b-line execution time [s] | 3.14 | 3.36 | 3.58 | 3.76 |
| our execution time [s] | 4.49 | 3.66 | 3.42 | 5.12 |
| b-line compute time [s] | $5.3 \times 10^3$ | $9.0 \times 10^3$ | $5.1 \times 10^3$ | $4.7 \times 10^3$ |
| our compute time [s] | $6.8 \times 10^{-2}$ | $7.3 \times 10^{-2}$ | $9.4 \times 10^{-2}$ | $9.1 \times 10^{-2}$ |

Table 3.3: Performance relative to Baseline Method

The aim of this subsection is a comparison of the quality of solutions to Problems 3.19/3.20 generated by our algorithm with those of a strong baseline method based on a general-purpose nonlinear optimization solver. The latter will produce feasible trajectories that often exceed the performance of ones of our algorithm. However, whereas the compute time of the baseline method is on order of magnitude of hours for a single problem instance, our algorithm runs in a fraction of a second.

First, we describe the algorithm we use for the baseline procedure, which might be of independent interest. We represent the square speed profile as a piecewise polynomial function consisting of $N = 25$ segments (not to be confused with the discretization points at which we enforce various constraints), each formed as a linear combination of Bernstein basis polynomials of degree $d = 5$. In this representation, the problem amounts to finding optimal coefficients encoding the linear combination of bases for each segment. Perception constraints, speed bounds, and bank angle constraints are enforced at all discretization points, whereas thrust bounds are enforced at every second discretization point. Finally using the former ingredients for transcription of Problem 3.19/3.20, we employ a constrained trust region optimization algorithm [7].

In particular, we illustrate the relative performance of the two algorithms on the task of traversing a randomly generated path of length approximately eight meters, requring the quadrotor to start and finish at near-hover states (vanishing speed, and bank angle not exceeding 2°). At each discretization point, we sample a random set

of points in the environment and enforce the perception constraint that at least 10 of these be within field of view when the quadrotor is at the corresponding point along the path. The specific thrust of each of the four motors of the robot is approximately $1.5g$, and the FOV angle is $\frac{\pi}{4}$ radians.

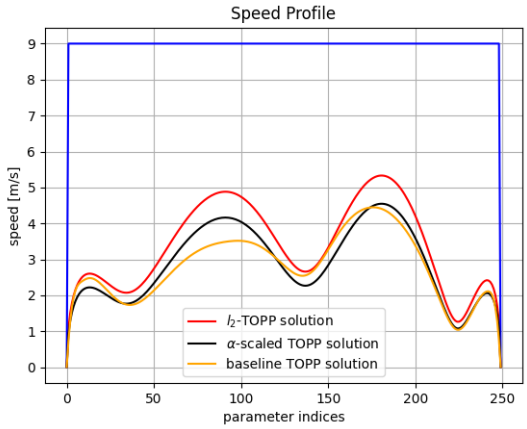



Figure 3-4: Example where Regularized-TOPP algorithm outperforms baseline

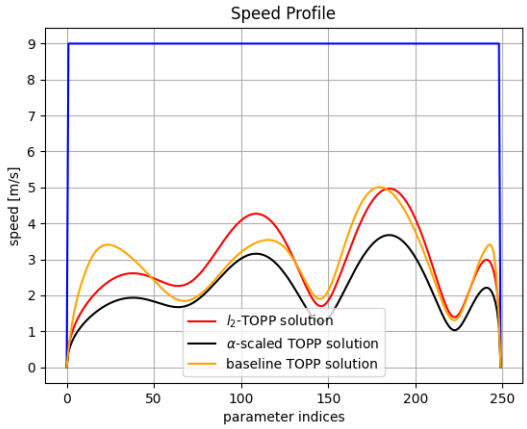



Figure 3-5: Example where baseline outperforms Regularized-TOPP

### 3.5.3 Dependence of Scaling and Penalty Hyperparameter

In this subsection we consider the effects of scaling the square speed profile on a single randomly generated problem instance. Figure 3-6 illustrates the execution time of the trajectory as a function of the regularization parameter, while Figure 3-7 shows

the largest $\alpha$ in the range $[0, 1]$ (discretized into 50 equally spaced points) at which all constraints are satisfied. Note in particular, the elbow-like shape of the graph in Figure 3-6. It shows that when the regularization term is too weak, the intermediate profile it too aggressive at some point along the path and the whole profile has to be scaled down by a significant factor. On the other hand, past a given point, the regularization term precludes the profile from being aggressive where it can afford to be so. The value of $\lambda$ is the only hyperparameter of the algorithm that has to be tuned. In our expriments we used the value of $10^{-3}$.

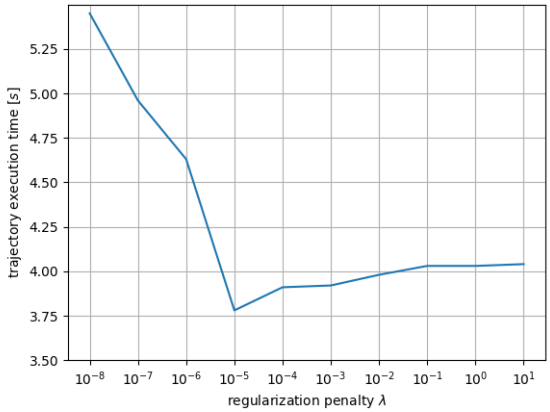



Figure 3-6: Traversal time of path as a function of regularization factor

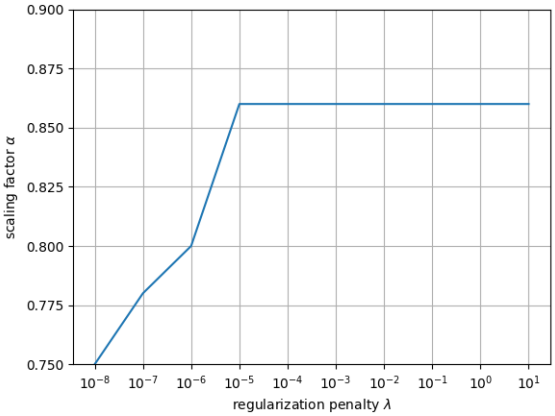



Figure 3-7: Required scaling factor versus regularization factor

## 3.6 Conclusion

We considered the perception-aware time optimal path parametrization problem for quadrotors. We showed how to losslessly compress the number of perceptions constraints in the resulting optimization problem, making this procedure useful for any gradient descent optimization scheme that uses time superlinear in the horizon of the problem to calculate search directions. Finally we showed how to transform the solution of a convex relaxation problem of the original problem to one feasible for the full rigid body dynamics of the quadrotor.

# Chapter 4

# Joint Landmark Selection and Time Optimal Path Parametrization for High-Speed Vision-Aided Navigation

## 4.1 Introduction

To avoid fatal crashes, fast flight in time-critical missions demands maintaining accurate current state estimates at all times. However, relating raw, and often high-dimensional, measurements of sensors on board an MAV to its state involves several stages of processing. Each stage requires computation, which in turn produces a delay. The larger the delay, the smaller the agility of maneuvers within the perception-actuation limits of the agent. This chapter develops efficient algorithms for designing near-time-optimal trajectories with consideration of such limits. It serves as a stepping stone towards allowing robotic platforms execute missions at operational speeds while ensuring their safety.

Data association is one of the core challenges of landmark-based visual navigation. It involves determining which subset of pixels of an image taken at one point in space and time registers the same region in the environment (namely, a landmark) as another, usually different, subset of pixels of an image taken at a different point in

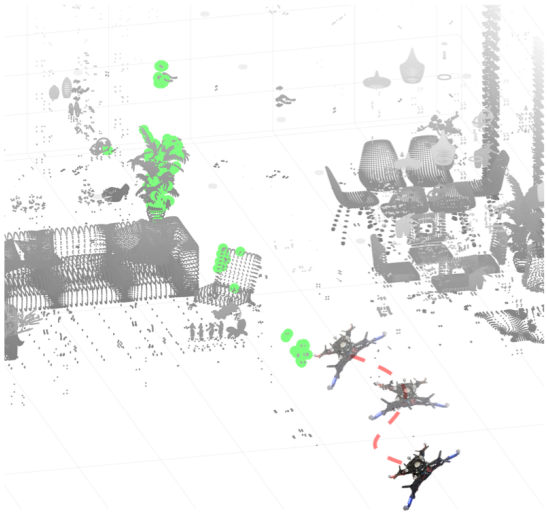


Figure 4-1: An MAV navigating an indoor environment. The selection of landmarks in the environment that allow the fastest execution of the path is shown in green.

space and time. When the vantage points of the pair of images are distant, one often has to resort to re-detecting and matching projections of landmarks across the different frames. Despite tremendous progress in improving the efficiency and robustness of the latter procedure [3, 21, 83, 84], it still carries a non-negligible computational burden. In some applications, such as structure from motion [2, 79], this seems difficult to avoid. However, in visual odometry, successive images are captured from proximate vantage points, allowing for a significantly more efficient approach to data association [44].

The Lucas-Kanade-Tomasi feature tracker [71] is a popular, computationally efficient, approach to measuring the change in bearing to a particular landmark between successive images. It is an iterative search method, with the area of search constrained by some understanding of the maximum distance between projections of landmarks in consecutive frames. As a result, the computational burden of such a method of data association grows with the range of motion underwent by tracked features. In general, higher feature speeds make the tracking task more difficult for two reasons. Firstly, the area of search has to be suitably increased. Second, larger projection speeds are increasingly afflicted by motion blur.

This motivates us to consider a problem at the intersection of landmark selection and motion planning: which subset of landmarks of desired cardinality should the

agent track, so it can traverse a given geometric path in minimum time subject to maintaining projection speeds of selected landmarks below a specified threshold. The upper bound on projection speed serves as a proxy for an upper bound on the computational burden of data association (i.e. feature tracking). Similarly, the cardinality constraint captures the fact that tracking a larger set of landmarks leads to more visual measurements, which in turn yields more accurate state estimates. Therefore, our problem involves both a continuous and discrete optimization component. The continuous one effectively amounts to finding the time-optimal scheduling of accelerations and decelerations along the path that respect the actuation and perception constraints of the vehicle. The discrete part involves choosing the subset of perception constraints which allow for the fastest possible execution time.

Before shedding light on related work, we pause to answer a natural question: why not use additional sensors, such as an inertial measurement unit, to guide the search for reprojections of landmarks? Indeed, such an approach is viable. However, it introduces several practical challenges. Firstly, measurements from the IMU have to be properly synchronized with those of the camera [34]. This makes the state estimation module more complex. Second, such a method would still leave unanswered the problem of dealing with motion blur. Finally, the reliance on IMU would leave open the challenge of how to behave should it fail.

Landmark selection alone pertains to scenarios where the robot must select a set of future visual measurements of given cardinality without altering the trajectory it follows. In this case, upper bounds on cardinality arise from the desire to limit the computational complexity of optimization algorithms that have to run in real time. Some of the earliest works in this direction, such as [40], considered selecting a cardinality-constrained subset of known landmarks lying within field of view of the camera on board a UAV in order to minimize the uncertainty of a weighted combination of components of its pose. Similarly, [87] dealt with the problem of selecting measurements of projections of landmarks triangulated throughout a SLAM mission that allow the most accurate estimate of the future pose of the camera. The heart of their procedure involved selecting a triplet of previously triangulated land-

marks whose stripped observability matrix has the largest minimal singular value. Their subsequent work [86] resulted in an approximation algorithm for selecting a cardinality-constrained subset of landmarks that maximize the latter quantity. Relatedly, [8] addressed the task of enhancing the state estimate obtained from inertial measurements with those of a camera registering the change in bearing to previously unknown landmarks in the environment. Their algorithm chooses measuring the bearing to a near-optimal subset of approximately localized landmarks that are anticipated to remain within field of view.

There have been a couple of recent papers closely related to the problem we consider here. As mentioned in chapter 3, [19, 39] plan trajectories for quadrotors in a receding horizon fashion, incorporating the penalty on the magnitude of the speed of the projection of a specified landmark into the objective function minimized at each decision step. As a result, their method handles the latter class of bounds as a soft constraint, and cannot guarantee the projection speed lies below a set threshold.

In general, problems involving discrete (measurement/landmark) selection components are computationally challenging to solve, with various gradations of hardness. One particularly relevant problem to informative sensor selection is the maximum cover problem (MCP) [55]. MCP addresses selecting a given number of elements of a family of subsets with the largest union. Though efficient approximation algorithms for this NP-hard problem do exist, it is hard to approximate with *arbitrary* precision by a polynomial time algorithm [22]. Other relevant complexity-theoretic hardness results include [11, 78, 85].

Approximation algorithms are ways of dealing with combinatorially-difficult problems. For budgeted maximization problems, one of the most common tools in robotics involves submodularity of objective functions, capturing a notion of diminishing returns [52,68]. In many ways, what concavity is for continuous maximization, submodularity is for discrete maximization. Even when functions are not submodular, it is possible to give approximation guarantees using the notion of approximate submodularity, a measure of how far a given function is from being a submodular one [81]. Another line of research uses the notion of discrete curvature [29, 77]. The latter

intuitively captures the distance of a given set function from simply being the sum of its parts.

### Contributions

This chapter has several contributions on the theoretical and practical front. On the practical end, algorithms developed in this chapter allow for a computationally efficient algorithm for guiding the focus of a vision-driven vehicle on regions of its environment that let it traverse a specified path as fast as possible while maintaining accurate up-to-date state estimates at all times. Unlike previous approaches, our method satisfies hard perception constraints, which we believe to be necessary to ensure safety of the vehicle. We also find that our algorithm produces solutions that are empirically close to ones output by an exponential-time globally-optimal algorithm.

On the theoretical end, we separate two different contributions. The first involves a separation principle, somewhat akin to that of [81], of computing the optimal square speed profile and selecting the optimal subset of landmarks. Second, we provide a novel take on the underlying combinatorial optimization problem, viewing it as a noisy version of a non-combinatorial one. This is to alleviate the fact that we are dealing with a problem whose naive generalizations are NP-hard to approximate to within an arbitrary constant factor [11, 78]. Roughly speaking, the level of noise controls the approximation quality of our algorithm; lower noise leads to stronger optimality bounds. This allows us to develop strong suboptimality guarantees for a subclass of problems we consider. Results of this chapter are a substantial extension of the conference paper [75].

## 4.2 Problem Statement

This section builds up the joint landmark selection and time optimal path parametrization (JLS-TOPP) problem. As in chapter 3, we start by describing dynamical aspects of the path traversal task, followed by a general family of pertinent perception con-

straints. After translating latter requirements into those on the square speed profile, we give a compact formulation of the JLS-TOPP problem.

### 4.2.1 Geometric Path and Dynamic Model

We focus on regularly-parametrized geometric paths

$$(\gamma, \mathbf{R}) : [0, S_{end}] \to \mathbb{R}^3 \times SO(3), \tag{4.1}$$

that specify both positions and orientations of the vehicle. Additionally, we assume the agent can execute any such path in a quasi-static manner. The latter two requirements implicitly restrict both the nature of such paths, as well as the type of vehicles that can traverse them. One viable example of such an agent-path coupling is a kinematic plane (Equation 2.1), together with a translational path which implicitly specifies the orientation of the vehicle by aligning its body x and z axes with the projection of its direction of motion onto the horizontal plane and the z axes of the world frame, respectively. A more general class of agents include fully-actuated MAVs, such as hexarotors with tilted motors [53]. Their dynamics are modelled as

$$\underbrace{\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{v}} \\ \dot{R} \\ \dot{\omega} \end{bmatrix}}_{\dot{\mathbf{q}}} = \begin{bmatrix} \mathbf{v} \\ \mathbf{g} \\ R[\omega]_\times \\ -J^{-1}\ \omega \times J\omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ R & 0 \\ 0 & 0 \\ 0 & J^{-1} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{c} \\ \tau \end{bmatrix}}_{\text{effective input } \mathbf{u}}, \tag{4.2}$$

where the effective input $\mathbf{u} \in \mathbb{R}^6$ is subject to suitable convex membership constraints determined by thrust bounds and the geometric configuration of the motors.

### 4.2.2 Perception Model

We assume the relative pose between the agent and its on-board camera (WLOG with unit focal length), given by the transformation $({}^B\mathbf{t}_C, {}^B\mathbf{R}_C)$, is fixed. When the

pose of the agent is $({}^W\mathbf{t}_B, {}^W\mathbf{R}_B) = (\gamma(s), \mathbf{R}(s))$, landmark ${}^W\mathbf{l}_i$ has coordinates

$$ {}^C\mathbf{l}_i(s) = \underbrace{{}^C\mathbf{R}_B {}^B\mathbf{R}_W(s)({}^W\mathbf{l}_i - \gamma(s) - \mathbf{R}(s){}^B\mathbf{t}_C)}_{\text{function of } s \text{ and } {}^W\mathbf{l}_i \text{ only}} \tag{4.3} $$

with respect to $C$, the reference frame of the camera, and is captured as point $\left[\frac{{}^C\mathbf{l}_i(s)_1}{{}^C\mathbf{l}_i(s)_3}, \frac{{}^C\mathbf{l}_i(s)_2}{{}^C\mathbf{l}_i(s)_3}\right]^T$.

### 4.2.3 Perception Constraints

We are given a set of landmarks with known static positions with respect to the world frame:

$$ \mathcal{L} = \{{}^W\mathbf{l}_1, ..., {}^W\mathbf{l}_n\}. \tag{4.4} $$

For most of this chapter, we assume that all landmarks in $\mathcal{L}$ remain within field of view throughout the traversal of the path. This is a statement about the geometric relationship of $(\gamma(\cdot), R(\cdot))$ with $\mathcal{L}$, and involves no component of decision-making. Such an assumption may fail to hold in obstacle-rich environments where landmarks often get occluded. At the end of Section 4.4, we show how to handle this issue.

The agent can successfully track a landmark whose projection speed lies below a set threshold $\pi_{max}$. Equation (4.3) readily implies that for all $1 \leq i \leq n$, the projection speed of $\mathbf{l}_i$ is below $\pi_{max}$ if and only if

$$ \frac{ds}{dt}(s) \leq U_i(s) := \frac{\pi_{max}}{\left|\left|\frac{d}{ds}({}^C\mathbf{l}_i(s)/({}^C\mathbf{l}_i(s) \cdot \mathbf{e}_3))\right|\right|_2} \quad \forall s \in [0, S_{end}]. \tag{4.5} $$

Defining $M_i := U_i^2$, the latter is equivalent to the following constraint on the square speed profile (recall $h(s) \equiv (\frac{ds}{dt}(s))^2$)

$$ h(s) \leq M_i(s) \ \forall s \in [0, S_{end}]. \tag{4.6} $$

To specify perception constraints, we introduce a monotonically increasing set function

$$ \mathcal{F} : 2^{\mathcal{L}} \to [0, \infty). \tag{4.7} $$

Similarly to chapter 3, $\mathcal{F}(V)$ denotes the utility of being able to track projections of all landmarks in the set $V \subseteq \mathcal{L}$ while traversing the path. The monotonicity of $\mathcal{F}$ captures the intuitive property that tracking a larger set of landmarks yields more measurements, and thus results in more accurate state estimates. Ultimately, the perception constraint amounts to the requirement

$$\mathcal{F}(V) \geq \lambda_{min} \tag{4.8}$$

for a fixed $\lambda_{min} > 0$. The value of $\lambda_{min}$ serves as a lower bound on a suitable localization metric [9].

### 4.2.4 Mathematical Formulation

Denote our agent by $\mathcal{A}$. We seek the shortest execution time of the geometric path $(\gamma(\cdot), R(\cdot))$ by $\mathcal{A}$ that is (i) dynamically feasible, and (ii) maintains the projection speed of each landmark in some *fixed* subset $V \subseteq \mathcal{L}$, with $\mathcal{F}(V) \geq \lambda_{min}$, onto the canvas of the camera on-board $\mathcal{A}$ below a specified threshold $\pi_{max}$. The tradeoff is that larger subsets of landmarks get closer to satisfying the perception constraint, while imposing stricter bounds on the square speed profile, and thus increasing the traversal time of the path. In summary, the problem reads:

$$\begin{aligned}
\underset{\substack{h:[a,b]\to[0,\infty) \\ V\subseteq\mathcal{L}}}{\text{minimize}} \quad & \int_a^b \frac{ds}{\sqrt{h(s)}} \\
\text{subject to} \quad & D^+h(s) \leq f^+(s,h(s)),\ s \in [a,b), \\
& D^-h(s) \geq f^-(s,h(s)),\ s \in [a,b), \\
& 0 \leq h(s) \leq \min_{j\in V}\ M_j(s),\ s \in [a,b] \\
& \mathcal{F}(V) \geq \lambda_{min}.
\end{aligned} \tag{4.9}$$

## 4.3 Algorithms

We describe three related methods for solving Problem 4.9. A solution consists of two components: a subset of landmarks $V$ with $\mathcal{F}(V) \geq \lambda_{min}$, and a dynamically feasible square speed profile which allows tracking selected landmarks. We numerically represent our estimate of the optimal square speed profile, $h^*$, by its sequence of values $(\hat{h}_i)_{i=0}^N$ at a set of discretization points $D = D([a,b],(s_i)_{i=0}^N)$ of the path. We approximate the execution time of a profile $(h_i)_{i=0}^N$ by $\sum_{i=0}^{N-1} \frac{2(s_{i+1}-s_i)}{\sqrt{h_{i+1}}+\sqrt{h_i}}$.

All three methods can be formulated as two-stage procedures, the first stage being common to all three. We first compute the time optimal square speed profile for tracking every landmark *individually*. This is done using Algorithm 1 from chapter 2. Knowledge of the location of the landmark allows us to translate the bound on its maximal projection speed to a pointwise bound on the square speed profile as obtained from Equation (4.5). As a reminder, the algorithm calculates the optimal square speed profile in a pair of backward and forward passes along the sequence of discretization points, incrementally computing for each point the highest speed which can be reached from the start feasibly and from which there exists a feasible trajectory to reach the goal.

In Section 4.4, we will see that the optimal profile for tracking a *subset* of landmarks is just the pointwise minimum of the optimal profiles for tracking its individual elements. Hence, the second stages of the three methods primarily differ on how they use the information obtained from the first stage of the algorithm. The first method, developed specifically for modular functions $\mathcal{F}$ (i.e. for which $\mathcal{F}(V) \equiv \sum_{v\in V} w_v$ with $w_{(\cdot)} > 0$), recasts the problem as a mixed integer linear program, which can then be solved by off-the-shelf packages, such as Gurobi [25]. The second method, "K-Fastest", sorts the optimal profiles associated with landmarks in order of increasing execution time, and selects the $k$ fastest profiles that together satisfy the perception constraint. The third method, "Incremental-Greedy", starts with an empty set of landmarks and incrementally adds an unselected landmark which induces a minimal increase in execution time, until the perception constraint is satisfied.

### 4.3.1 Mixed Integer Formulation

The mixed integer formulation is suitable for producing ground truth solutions in the following family of scenarios. Each landmark $\mathbf{l}_i$ is assigned a weight $w_i > 0$, and we choose some $\lambda_{min} > 0$ and $\mathcal{F}(V) = \sum_{i \in V} w_i$. Let the time-optimal profile for tracking landmark $\mathbf{l}_i$ alone be $\hat{h}^{(i)}$ for $1 \leq i \leq n$.

The task of selecting the optimal subset $V$ together with the optimal speed profile can be recovered from the following binary mixed integer linear program:

$$\begin{aligned} \min_{(\zeta_{1:n},\ y_{1:N})} \quad & \sum_{i=1}^{N} (s_i - s_{i-1})\ y_i \\ s.t. \quad y_i \ &\geq\ \zeta_j \frac{2}{\sqrt{\hat{h}^{(j)}(s_{i-1})} + \sqrt{\hat{h}^{(j)}(s_i)}} \quad \forall\ 1 \leq i \leq N \\ & \sum_{j=1}^{n} w_j \zeta_j \ \geq\ \lambda_{min} \\ & \zeta \in \{0,1\}^n, \quad y \in \mathbb{R}^N. \end{aligned} \tag{4.10}$$

Denoting the optimal solution by $(\zeta^*_{1:n}, y^*_{1:N})$, selected landmarks may be read off via

$$V^* = \{j \mid \zeta^*_j = 1\}, \tag{4.11}$$

whereas the optimal square speed profile may be approximated as

$$h^*(s_i) = \frac{1}{(y^*(s_i))^2} \quad \forall 1 \leq i \leq N. \tag{4.12}$$

Here, the approximation error vanishes as the maximal spacing between consecutive elements of $D$ tends to zero.

This approach, however, has worst-case running time exponential in $n$. The remaining two algorithms will sacrifice optimality in favour of substantial gains in computational efficiency. They will also be able to handle the general class of perception constraints described in Section 4.2.3.

### 4.3.2 K-Fastest Algorithm

The K-Fastest algorithm is the simplest approach we consider. The complexity of computing optimal profiles for tracking all landmarks in $\mathcal{L}$ ($|\mathcal{L}| = n$) individually across $p$ processors is $O(Nn/p)$, and that of sorting them in order of increasing duration is $O(n \log n)$. Starting from an empty set, K-Fastest adds landmarks to $V$ in order of increasing execution time of corresponding profiles until the perception constraint is satisfied. Adding an element to $V$ involves two bookkeeping operations: updating the current profile and computing the incremental gain of $\mathcal{F}$ in $O(N)$ and $O(G)$ time, respectively. For modular functions $\mathcal{F}$, for example, $G = \Theta(1)$. Bringing everything together, the running time of K-Fastest is $O(n(N + \log n + G))$.

**Algorithm 3:** K-Fastest Algorithm

**Data:** $D = (s_i)_{i=0}^{N}$, $({}^{W}\mathbf{l}_j)_{j=1}^{n}$, $f^+$, $f^-$, $\mathcal{F}$
**Result:** $(\hat{h}(s_i))_{i=0}^{N}$, $V \subseteq \mathcal{L}$
**for** $j = 1$ *to* $n$ **do**
  $(M_j(s_i))_{i=0}^{N} \leftarrow UpperBound(D, \gamma, {}^{W}\mathbf{l}_j)$
  $(\hat{h}^{(j)}(s_i))_{i=0}^{N}, T^j \leftarrow$ Backward-Forward$(D, M_j, f^+, f^-)$
**end**
$\pi \leftarrow Sort((T^j)_{j=1}^{n})$
$V \leftarrow \emptyset$
$i \leftarrow 0$
$\hat{h} \leftarrow +\infty$
**while** $\mathcal{F}(V) < \lambda_{min}$ **do**
  $\hat{h} \leftarrow \hat{h} \wedge \hat{h}^{(\pi(i))}$
  $V \leftarrow V \cup \{i\}$
  $i \leftarrow i + 1$
**end**
return $(\hat{h}(s_i))_{i=0}^{N}$, $V$

### 4.3.3 Incremental Greedy Algorithm

A potential source of suboptimality in the K-Fastest algorithm is that it selects elements of $V$ oblivious to how they interact with one another to influence the execution time of the resulting speed profile. The Incremental Greedy algorithm seeks to partially address this apparent blind spot. While the perception constraint still has not

been satisfied, the present algorithm adds the single unselected landmark that induces the smallest increment in *execution time* of the intermediate profile. As a result, the complexity of this stage of the algorithm is quadratic in the number of landmarks, and so its cumulative running time is $O(n^2N + nG)$. Due to the quadratic scaling with $n$, Incremental Greedy is in theory an order of magnitude slower than K-Fastest. However, it has the potential to detect combinatorial interactions between the various landmarks that K-Fastest ignores.

**Algorithm 4:** Incremental Algorithm

**Data:** $D = (s_i)_{i=0}^{N}$, $({}^{W}\mathbf{l}_j)_{j=1}^{n}$, $f^+$, $f^-$, $\mathcal{F}$
**Result:** $(\hat{h}(s_i))_{i=0}^{N}$, $V$
**for** $j = 1$ *to* $n$ **do**
  $(M_j(s_i))_{i=0}^{N} \leftarrow$ *UpperBound*$(D, \gamma, {}^{W}\mathbf{l}_j)$
  $(\hat{h}^{(j)}(s_i))_{i=0}^{N} \leftarrow$ Backward-Forward$(D, M_j, f^+, f^-)$
**end**
$V \leftarrow \emptyset$
$\hat{h} \leftarrow +\infty$
**while** $\mathcal{F}(V) < \lambda_{min}$ **do**
  $j_{opt} \leftarrow$ null
  $\tau'_{opt} \leftarrow +\infty$
  **for** $j' \in \{1, 2, \cdots, n\} \setminus V$ **do**
    $\tilde{h} \leftarrow \hat{h} \wedge \hat{h}^{(j)}$
    $\tau' \leftarrow \sum_{i=0}^{N-1} \frac{2(s_{i+1}-s_i)}{\sqrt{\tilde{h}^{(j)}(s_{i+1})}+\sqrt{\tilde{h}^{(j)}(s_i)}}$
    **if** $\tau' < \tau'_{opt}$ **then**
      $\tau'_{opt} \leftarrow \tau'$
      $j_{opt} \leftarrow j'$
    **end**
  **end**
  $\hat{h} \leftarrow \hat{h} \wedge \hat{h}^{(j_{opt})}$
  $V \leftarrow V \cup \{j_{opt}\}$
**end**
return $(\hat{h}_i)_{i=0}^{N}$, $V$

In the previous two algorithms, we assumed that calculating increments of $\mathcal{F}$ is not the computational bottleneck of the procedure. However, when all known efficient (polynomial-time) algorithms can calculate $\mathcal{F}$ only to within a constant factor $\phi$, we may terminate the selection of landmarks the moment $\mathcal{F}(V) \geq \lambda_{max}/\phi$. This may

readily be translated into bicriteria guarantees [88] using analysis we delve into next.

## 4.4 Analysis

An overview of this section is as follows. First, we present a "separation principle" involving a modular framework for calculating the optimal square speed profile and selecting a favourable subset of landmarks. Then, we show that for straight line motions, both the K-Fastest and Incremental Greedy algorithms have strong instance-dependent suboptimality guarantees. At the end of the section, we show how to adapt our algorithms for scenarios in which the selection of landmarks necessarily involves changes due to occlusions by obstacles.

### 4.4.1 Separation Principle

We draw upon several results in [74] that tell us that for any *fixed* selection of landmarks, the set of feasible square speed profiles is closed under: (a) pointwise suprema and infima; and (b) convex combinations. The following claim justifies an efficient method for recovering the optimal speed profile for tracking a desired subset of landmarks from the optimal profiles for tracking each of its elements individually. One practical significance of this result is that it makes a difference between the linear and quadratic complexity of the K-Fastest algorithm.

**Claim 1.** *Suppose the optimal square speed profile for tracking landmark $j$ is $\overline{h}^{(j)}$, for $j \in \{1, 2, ..., n\}$. Then, the optimal square speed profile for tracking all landmarks in the set $\mathcal{S} = \{j_1 < j_2 < \cdots < j_k\}$ is given by*

$$\overline{h}^{(\mathcal{S})}(s) := \min_{1 \leq r \leq k} \overline{h}^{(j_r)}(s) \quad \forall s \in [0, S_{end}].$$

*Proof.* Landmark $j$ can be tracked with profile $h$ if and only if the latter is dynamically feasible and does not exceed $M_j$. Thus, the optimal such profile, $\overline{h}^j$, is the pointwise supremum of all dynamically feasible profiles which are at most $M_j$. Additionally, landmark $j$ can be tracked with any dynamically feasible profile $h$ such that $h \leq \overline{h}^j$.

Let $\overline{h}$ be the optimal profile for tracking all landmarks in $\mathcal{S}$. Consider any $j \in \mathcal{S}$. Since landmark $j$ can be tracked with $\overline{h}$, from the supremal characterization of $\overline{h}^j$, we know that $\overline{h} \leq \overline{h}^j$. Since $j \in \mathcal{S}$ was arbitrary, $\overline{h}$ is no greater than $\overline{h}^{(\mathcal{S})}$. Conversely, the profile $\overline{h}^{(\mathcal{S})}$ is dynamically feasible as a pointwise minimum of feasible profiles, and furthermore, satisfies the desired upper bounds on the speed at every point of the path, finishing the proof of the claim. □

### 4.4.2 Algorithm Guarantees

In what follows, we will use $\tau(\cdot)$ to denote the function mapping any sequence of square speed profiles to the execution time of their pointwise minimum:

$$\tau(p_1, p_2, ..., p_n) = \tau(p_1 \wedge p_2 \wedge \cdots \wedge p_n).$$

Furthermore, since $\tau(h) = \int_{s=0}^{S_{end}} \frac{ds}{\sqrt{h(s)}}$, for every $\epsilon > 0$ we have

$$\tau(\epsilon h) = \frac{\tau(h)}{\sqrt{\epsilon}}.$$

Also, $\tau$ is is monotonically decreasing: if $h_1 \geq h_2$ then $\tau(h_1) \leq \tau(h_2)$.

**Claim 2.** *Assume the zero profile $h \equiv 0$ is feasible. For an arbitrary strictly positive upper bound on the profile, say $M$, let $\overline{h}_M$ be the optimal feasible square speed profile that does not exceed $M$. For an arbitrary $\epsilon > 0$, define $\overline{h}_{M(1+\epsilon)}$ analogously. Then, we have:*

$$\begin{aligned} \overline{h}_M \;&\leq \overline{h}_{M(1+\epsilon)} \;\leq\; (1+\epsilon)\overline{h}_M \\ \tau(\overline{h}_{M(1+\epsilon)}) \;&\leq \tau(\overline{h}_M) \;\leq\; \sqrt{1+\epsilon}\;\tau(\overline{h}_{M(1+\epsilon)}) \end{aligned} \tag{4.13}$$

*Proof.* The inequality on the left hand side follows from the fact that any dynamically feasible profile that does not exceed $M$ also does not exceed $M(1+\epsilon)$, together with the characterization of $\overline{h}_{M(1+\epsilon)}$ as the pointwise supremum of the set of feasible profiles that do not exceed the latter bound.

The inequality on the right hand side may be argued as follows. The convexity of

the set of feasible square speed profiles implies that

$$\frac{\epsilon}{1+\epsilon}0 + \frac{1}{1+\epsilon}\overline{h}_{M(1+\epsilon)}$$

is a feasible square speed profile that does not exceed $M$. The extremal characterization of $\overline{h}_M$ as the square speed profile that is the pointwise supremum of feasible square speed profiles not exceeding $M$ settles the claim (the inequalities for $\tau$ follow from remarks preceding the claim). □

**Corollary 1.** *In the setting of Claim 2, let $P$ and $Q$ be two arbitrary upper bounds on the square speed profile such that*

$$\left|\left|\frac{P}{Q}\right|\right|_\infty, \left|\left|\frac{Q}{P}\right|\right|_\infty \leq 1+\epsilon, \tag{4.14}$$

*and let $h_P$ and $h_Q$ be the time-optimal dynamically feasible square speed profiles not exceeding $P$ and $Q$, respectively. Then*

$$\left|\left|\frac{h_P}{h_Q}\right|\right|_\infty, \left|\left|\frac{h_Q}{h_P}\right|\right|_\infty \leq 1+\epsilon. \tag{4.15}$$

*Proof.* By virtue of relation (4.13), we have

$$h_P \leq h_{((1+\epsilon)Q)} \leq (1+\epsilon)h_Q$$

and similarly

$$h_Q \leq h_{((1+\epsilon P))} \leq (1+\epsilon)h_P,$$

and so the result follows. □

The following theorem is one of the two key theoretical results of the chapter. It paves the way for viewing our combinatorial problem as a noisy version of a non-combinatorial one, where the level of noise translates into suboptimality guarantees.

**Theorem 11.** *Let $P = \{p_1, p_2, ..., p_n\}$ and $Q = \{q_1, q_2, ..., q_n\}$ be two sets of square speed profiles, referred to as p-profiles and q-profiles, respectively. Let $\mathcal{F} : 2^{[n]} \rightarrow$*

$[0, \infty)$ *be a monotonically increasing set function and let* $\lambda_{min}$ *be a given threshold. Let* $\epsilon > 0$ *be a positive real number such that*

$$\left|\left|\frac{p_i}{q_i}\right|\right|_\infty, \left|\left|\frac{q_i}{p_i}\right|\right|_\infty \leq 1 + \epsilon$$

*for all* $1 \leq i \leq n$*. In addition, assume that* $Q$ *can be completely ordered* $q_1 \geq q_2 \geq ... \geq q_n$*: if* $i < j$ *then* $q_i(s) \geq q_j(s)$ *for all* $s \in [0, S_{end}]$*. Defining* $T_{P_*^{(k)}}$ *and* $\tilde{T}_{P_*^{(k)}}$ *to be the execution times of the pointwise minimum of the optimal subset of p-profiles and the subset of individually fastest p-profiles meeting the set constraint (4.8), respectively, we have:*

$$\tilde{T}_{P_*^{(k)}} \leq (1 + \epsilon) T_{P_*^{(k)}}. \tag{4.16}$$

*Proof.* Let

$$\mathcal{S} = \{p_{i_1}, p_{i_2}, \ldots, p_{i_l}\} \tag{4.17}$$

be the set of $p$-profiles chosen by the K-Fastest algorithm, and let

$$\mathcal{S}^* = \{p_{j_1}, p_{j_2}, \ldots, p_{j_k}\} \tag{4.18}$$

be an optimal set of profiles. Assume

$$\tau(p_{j_k}) \geq \tau(p_{j_r}) \quad \forall 1 \leq r \leq k, \quad \text{and} \quad \tau(p_{i_l}) \geq \tau(p_{i_r}) \quad \forall 1 \leq r \leq l. \tag{4.19}$$

In other words, $p_{j_k}$ and $p_{i_l}$ are the slowest profiles in an optimal set of profiles and those chosen by the K-Fastest algorithm, respectively.

We claim

$$\tau(p_{j_k}) \geq \tau(p_{i_l}). \tag{4.20}$$

Indeed, assume this was not the case: $\tau(p_{j_k}) < \tau(p_{i_l})$. By the definition of the K-Fastest algorithm, this would imply

$$\mathcal{S}^* \subseteq \mathcal{S} \setminus \{p_{i_l}\}, \tag{4.21}$$

and since $\mathcal{S}^*$ satifies the set constraint, the algorithm would have terminated by the time it had to select the $l$-th profile, a contradiction. As a result, we have

$$\tau(\mathcal{S}^*) \geq \tau(p_{j_k}) \geq \tau(p_{i_l}). \tag{4.22}$$

Consider the $q$-profiles $q_{i_1}, q_{i_2}, \ldots, q_{i_l}$ matched with $p_{i_1}, p_{i_2}, \ldots, p_{i_l}$. Let $q_{i_m}$ be the lowest, and therefore slowest, of the latter sequence of $q$-profiles:

$$q_{i_m}(s) \leq q_{i_r}(s) \quad \forall s \in [0, S_{end}],\ \forall 1 \leq r \leq l. \tag{4.23}$$

Then, for every $1 \leq r \leq l$, we have

$$p_{i_r} \geq \frac{q_{i_r}}{1+\epsilon} \geq \frac{q_{i_m}}{1+\epsilon} \tag{4.24}$$

and so

$$p_{i_1} \wedge p_{i_2} \wedge \cdots \wedge p_{i_l} \geq \frac{q_{i_m}}{1+\epsilon} \tag{4.25}$$

implying that

$$\tau(\mathcal{S}) \leq \sqrt{1+\epsilon}\ \tau(q_{i_m}). \tag{4.26}$$

By assumption (4.19), we know

$$\tau(p_{i_m}) \leq \tau(p_{i_l}), \tag{4.27}$$

implying

$$\tau(p_{i_m}) \geq \tau((1+\epsilon)q_{i_m}) = \frac{\tau(q_{i_m})}{\sqrt{1+\epsilon}}. \tag{4.28}$$

As a result,

$$\tau(q_{i_m}) \leq \sqrt{1+\epsilon}\ \tau(p_{i_l}). \tag{4.29}$$

Finally, putting the latter observations together, we obtain

$$\begin{aligned}\tau(\mathcal{S}) &\leq \sqrt{1+\epsilon}\ \tau(q_{i_m}) \\ &\leq (1+\epsilon)\tau(p_{i_l}) \\ &\leq (1+\epsilon)\tau(p_{j_k}) \\ &\leq (1+\epsilon)\tau(\mathcal{S}^*).\end{aligned} \tag{4.30}$$

□

The following theorem is the second core result of the chapter. It sheds light on further structure underlying the apparent combinatorial difficulty of Problem (4.9).

**Theorem 12.** *Consider the task in which the agent has to traverse a path $\gamma$ with constant orientation $\mathbf{R}$ with respect to the world frame. For this theorem, we assume $\gamma$ is a segment of length $L$ that belongs to line $\rho$. Suppose the depth of all $n$ landmarks lying within field of view of the on-board camera lies in the range $[\kappa_1 L, \kappa_2 L]$ at every point along the path, where $0 < \kappa_1 \leq \kappa_2 \leq +\infty$. Then, the K-Fastest algorithm produces a feasible square speed profile whose execution time is within a factor of $\left(\frac{1+1/\kappa_1}{1+1/\kappa_2}\right)$ of the optimum.*

*Proof.* To avoid excessive notation, assume the origin of the reference frame of the camera coincides with the origin of the body frame of the agent. This is also a sound *approximation* in the setting where the dimensions of the agent are negligible compared to its distance from landmarks in the environment. Note that

$$\begin{aligned}\left|\left|\frac{d}{ds}\left(\frac{{}^C\mathbf{l}}{\mathbf{e}_3 \cdot {}^C\mathbf{l}}\right)\right|\right|_2 &= \left|\left|\frac{{}^C\mathbf{l}'(\mathbf{e}_3 \cdot {}^C\mathbf{l}) - {}^C\mathbf{l}(\mathbf{e}_3 \cdot {}^C\mathbf{l}')}{(\mathbf{e}_3 \cdot {}^C\mathbf{l})^2}\right|\right|_2 \\ &= \left|\left|\frac{\mathbf{e}_3 \times ({}^C\mathbf{l}' \times {}^C\mathbf{l})}{(\mathbf{e}_3 \cdot {}^C\mathbf{l})^2}\right|\right|_2.\end{aligned} \tag{4.31}$$

Next, we have

$$\begin{aligned}{}^C\mathbf{l}' &= \frac{d}{ds}(\mathbf{R}^T({}^W\mathbf{l} - \gamma)) \\ &= (\mathbf{R}')^T({}^W\mathbf{l} - \gamma) - \mathbf{R}^T\gamma' \\ &= -\mathbf{R}^T\gamma',\end{aligned} \tag{4.32}$$

where the second and third equalities follow from assumptions that $\mathbf{l}$ is stationary with respect to the world frame and that $\mathbf{R}$ is constant, respectively. Putting the latter two observations together, we get

$$\begin{aligned}\left|\left|\frac{d}{ds}\left(\frac{{}^C\mathbf{l}}{\mathbf{e}_3\cdot{}^C\mathbf{l}}\right)\right|\right|_2 &= \left|\left|\frac{(\mathbf{R}^T\mathbf{z}_C)\times((-\mathbf{R}^T\gamma')\times(\mathbf{R}^T({}^W\mathbf{l}-\gamma)))}{((\mathbf{R}^T\mathbf{z}_C)\cdot(\mathbf{R}^T({}^W\mathbf{l}-\gamma)))^2}\right|\right|_2 \\ &= \left|\left|\frac{\mathbf{z}_C\times(\gamma'\times(\mathbf{l}-\gamma))}{(\mathbf{z}_C\cdot(\mathbf{l}-\gamma))^2}\right|\right|_2.\end{aligned} \tag{4.33}$$

We now claim vector $\mathbf{z}_C\times(\gamma'\times(\mathbf{l}-\gamma))$ is constant for given $\mathbf{l}$ as a function of $s$. Indeed, this follows from noting that

$$\frac{d}{ds}(\mathbf{z}_C\times(\gamma'\times(\mathbf{l}-\gamma)))=\mathbf{z}_C'\times(\gamma'\times(\mathbf{l}-\gamma))+\mathbf{z}_C\times(\gamma''\times(\mathbf{l}-\gamma))+\mathbf{z}_C\times(\gamma'\times(\mathbf{l}-\gamma)')=0. \tag{4.34}$$

The first term vanishes since $\mathbf{z}_C$ is constant, the second since $\gamma$ is a straight line segment parametrized by arc length (and so $\gamma''=0$), and the third due to the fact that $\mathbf{l}$ is fixed (and so $\gamma'\times(\mathbf{l}-\gamma)'=-\gamma'\times\gamma'=0$). As a result, the speed bound imposed by the requirement of tracking landmark $\mathbf{l}_i$ amounts to

$$\begin{aligned}h(s) &\leq \left(\frac{\pi_{max}}{\left|\left|\frac{d}{ds}\left(\frac{{}^C\mathbf{l}}{\mathbf{e}_3\cdot{}^C\mathbf{l}}\right)\right|\right|_2}\right)^2 \\ &= c_i\ (\mathbf{z}_C\cdot(\mathbf{l}_i-\gamma(s)))^4\end{aligned} \tag{4.35}$$

where

$$c_i=\left(\frac{\pi_{max}}{||\mathbf{z}_C\times(\gamma'\times(\mathbf{l}_i-\gamma))||_2}\right)^2\quad\in\mathbb{R}\cup\{\infty\} \tag{4.36}$$

is a constant depending on landmark $i$. To finish off the proof, we will use the following auxiliary result, which allows us to give an upper bound on the level of "noise" of our algorithms in light of the paragraph preceding the statement of Theorem 11.

**Lemma 2.** *Fix $0<d$, $0<k_1<k_2\leq\infty$, and let $\{f_k\}_{k\in[k_1,k_2]}$ be a family of functions on $[0,d]$ defined via*

$$f_k(x)=(x+k)^4\quad\forall x\in[0,d],\ \forall k\in[k_1,k_2]. \tag{4.37}$$

*Then, there exists a function* $g : [0, d] \to (0, \infty)$ *and a family* $\{\alpha_k\}_{k\in[k_1,k_2]}$ *of positive real numbers such that for every* $k \in [k_1, k_2]$,

$$\left|\left|\frac{f_k}{g\alpha_k}\right|\right|_\infty, \left|\left|\frac{g\alpha_k}{f_k}\right|\right|_\infty \leq \frac{1 + d/k_1}{1 + d/k_2}. \tag{4.38}$$

*Proof.* We start with the following observation. Given any pair of strictly positive continuous functions $f$ and $g$ on $[0, d]$, choosing

$$\alpha = \left(\left|\left|\frac{f}{g}\right|\right|_\infty \left|\left|\frac{g}{f}\right|\right|_\infty^{-1}\right)^{1/2} \tag{4.39}$$

we can ensure

$$\left|\left|\frac{f}{\alpha g}\right|\right|_\infty, \left|\left|\frac{\alpha g}{f}\right|\right|_\infty \leq \left(\left|\left|\frac{f}{g}\right|\right|_\infty \left|\left|\frac{g}{f}\right|\right|_\infty\right)^{1/2}. \tag{4.40}$$

Note that the upper bound in the latter inequality remains unchanged by scaling either $f$ or $g$ by an arbitrary positive scalar. As a result, in order to settle the claim, we may redefine $f_k$ via

$$f_k(x) = \frac{(x + k)^4}{k^4} \quad \forall x \in [0, d],\ \forall k \in [k_1, k_2]. \tag{4.41}$$

In the same vein, it suffices to look for $g$ of the form

$$g = f_{\overline{k}} \tag{4.42}$$

for a suitable value of $\overline{k} \in [k_1, k_2]$. To this end, note that for any $k_l < k_h \in [k_1, k_2]$

$$\frac{f_{k_h}(x)}{f_{k_l}(x)} = \left(\frac{k_l}{k_h}\right)^4 \left(\frac{x + k_h}{x + k_l}\right)^4 = \left(\frac{k_l}{k_h}\right)^4 \left(1 + \frac{k_h - k_l}{x + k_l}\right)^4 \tag{4.43}$$

is a decreasing function of $x \in [0, d]$, with $\frac{f_{k_h}(0)}{f_{k_l}(0)} = 1$, implying

$$\left|\left|\frac{f_{k_l}}{f_{k_h}}\right|\right|_\infty = \frac{f_{k_l}(d)}{f_{k_h}(d)} \quad \text{and} \quad \left|\left|\frac{f_{k_h}}{f_{k_l}}\right|\right|_\infty = 1. \tag{4.44}$$

As a result, we have

$$\left|\left|\frac{f_k}{g}\right|\right|_\infty \left|\left|\frac{g}{f_k}\right|\right|_\infty = \begin{cases} f_{\overline{k}}(d)/f_k(d), & \text{if } k \in [\overline{k}, k_2] \\ f_k(d)/f_{\overline{k}}(d), & \text{if } k \in [k_1, \overline{k}]. \end{cases} \tag{4.45}$$

From the relation

$$f_k(d) = \left(1 + \frac{d}{k}\right)^4, \tag{4.46}$$

we immediately get that $f_k(d)$ is a decreasing function of $k$, and due to (4.45):

$$\left|\left|\frac{f_k}{g}\right|\right|_\infty \left|\left|\frac{g}{f_k}\right|\right|_\infty \leq \max\left\{\frac{f_{\overline{k}}(d)}{f_{k_2}(d)}, \frac{f_{k_1}(d)}{f_{\overline{k}}(d)}\right\} \quad \forall k \in [k_1, k_2]. \tag{4.47}$$

The minimum value of the right hand side,

$$\sqrt{\frac{f_{k_1}(d)}{f_{k_2}(d)}}, \tag{4.48}$$

is attained for $\overline{k}$ such that

$$f_{\overline{k}}(d) = \sqrt{f_{k_1}(d) f_{k_2}(d)} \in [f_{k_2}(d), f_{k_1}(d)], \tag{4.49}$$

which exists due to the fact that $f_k(d)$ is a continuous function of $k \in [k_1, k_2]$. Ultimately, putting together 4.40, and 4.49, we get there exists a positive function $g = f_{\overline{k}}$ and positive reals $\{\alpha_k\}_{k\in[k_1,k_2]}$ such that for all $k \in [k_1, k_2]$

$$\left|\left|\frac{f_k}{g\alpha_k}\right|\right|_\infty, \left|\left|\frac{g\alpha_k}{f_k}\right|\right|_\infty \leq \left(\left|\left|\frac{f_k}{g}\right|\right|_\infty \left|\left|\frac{g}{f_k}\right|\right|_\infty\right)^{1/2} \leq \left(\frac{f_{k_1}(d)}{f_{k_2}(d)}\right)^{1/4} = \frac{1 + d/k_1}{1 + d/k_2} \tag{4.50}$$

and the result follows. □

Now, we apply the lemma to functions

$$s \mapsto (\mathbf{z}_C \cdot (\mathbf{l}_i - \gamma(s)))^4 \tag{4.51}$$

expressed in a slightly different form. Defining

$$\mathbf{n} = \frac{\gamma(L) - \gamma(0)}{L} \tag{4.52}$$

to be the unit vector parallel to the path, we may write

$$\gamma(s) = \gamma(0) + s\mathbf{n}, \quad \forall s \in [0, L]. \tag{4.53}$$

Let $\overline{s} = L - s$. Then, we have

$$\begin{aligned}
(\mathbf{z}_C \cdot (\mathbf{l}_i - \gamma(s)))^4 &= (\mathbf{z}_C \cdot \mathbf{l}_i - \mathbf{z}_C \cdot (\gamma(L) - \mathbf{n}\overline{s}))^4 \\
&= ((\mathbf{z}_C \cdot \mathbf{l}_i - \mathbf{z}_C \cdot \gamma(L)) + \mathbf{z}_C \cdot \mathbf{n}\overline{s})^4 \\
&= (\mathbf{z}_C \cdot \mathbf{n})^4 \left( \overline{s} + \frac{\mathbf{z}_C \cdot (\mathbf{l}_i - \gamma(L))}{\mathbf{z}_C \cdot \mathbf{n}} \right)^4 .
\end{aligned} \tag{4.54}$$

From the conditions of the theorem, we know that for all $i$

$$\mathbf{z}_C \cdot (\mathbf{l}_i - \gamma(L)) \in [\kappa_1 L, \kappa_2 L] \quad \Rightarrow \quad \frac{\mathbf{z}_C \cdot (\mathbf{l}_i - \gamma(L))}{\mathbf{z}_C \cdot \mathbf{n}} \in \left[ \kappa_1 \frac{L}{\mathbf{z}_C \cdot \mathbf{n}}, \kappa_2 \frac{L}{\mathbf{z}_C \cdot \mathbf{n}} \right].$$

Hence we may now set

$$k_1 \to \kappa_1 \frac{L}{z_C \cdot \mathbf{n}}, \quad k_2 \to \kappa_2 \frac{L}{z_C \cdot \mathbf{n}}, \quad d \to L \tag{4.55}$$

in Lemma 2 to get that there exists a positive function $g$ and positive reals $\{\alpha_1, \ldots, \alpha_n\}$ such that for all $i$

$$\left|\left| \frac{c_i \ (\mathbf{z}_C \cdot (\mathbf{l}_i - \gamma(s)))^4}{\alpha_i g} \right|\right|_\infty, \left|\left| \frac{\alpha_i g}{c_i \ (\mathbf{z}_C \cdot (\mathbf{l}_i - \gamma(s)))^4} \right|\right|_\infty \leq \frac{1 + \frac{\mathbf{z}_C \cdot \mathbf{n}}{\kappa_1}}{1 + \frac{\mathbf{z}_C \cdot \mathbf{n}}{\kappa_2}}. \tag{4.56}$$

Let $q_i$ and $p_i$ be optimal square speed profiles not exceeding $\alpha_i g$ and $c_i \ (\mathbf{z}_C \cdot (\mathbf{l}_i - \gamma(s)))^4$, respectively. An application of Corollary 2 then yields

$$\left|\left| \frac{p_i}{q_i} \right|\right|_\infty, \left|\left| \frac{q_i}{p_i} \right|\right|_\infty \leq \frac{1 + \frac{\mathbf{z}_C \cdot \mathbf{n}}{\kappa_1}}{1 + \frac{\mathbf{z}_C \cdot \mathbf{n}}{\kappa_2}}, \tag{4.57}$$

as well as the fact that the set of profiles $\{q_i\}_{i=1}^n$ may be *completely* ordered. Thus, a direct application of Theorem 11 implies the K-Fastest algorithm is suboptimal by at most a factor of

$$\left(\frac{1+\frac{z_C\cdot\mathbf{n}}{\kappa_1}}{1+\frac{z_C\cdot\mathbf{n}}{\kappa_2}}\right) \leq \left(\frac{1+\frac{1}{\kappa_1}}{1+\frac{1}{\kappa_2}}\right), \tag{4.58}$$

as desired.

□

We now give instance-dependent approximation guarantees for the Incremental Greedy algorithm under slightly more stringent assumptions than those used for the K-Fastest algorithm. We begin with an analogue of Theorem 11.

**Theorem 13.** *Let $P = \{p_1, p_2, ..., p_n\}$ be a set of square speed profiles, referred to as p-profiles. Suppose the q-profiles, $P = \{q_1, q_2, ..., q_n\}$ are scalar multiples of a single positive profile q: $q_i = \alpha_i q$ with $\alpha_i > 0$ for all $1 \leq i \leq n$. Let $\mathcal{F} : 2^{[n]} \to [0, \infty)$ be a monotonically increasing set function and let $\lambda_{min}$ be a given threshold. Let $\epsilon > 0$ be a positive real number such that*

$$\left|\left|\frac{p_i}{q_i}\right|\right|_\infty, \left|\left|\frac{q_i}{p_i}\right|\right|_\infty \leq 1+\epsilon$$

*for all $1 \leq i \leq n$. Defining $T_{P_*^{(k)}}$ and $\tilde{T}_{P_*^{(k)}}$ to be the execution times of the pointwise minimum of the optimal subset of p-profiles meeting the set constraint (4.8) and the subset of p-profiles chosen by the Incremental Greedy algorithm,*

$$\tilde{T}_{P_*^{(k)}} \leq (1+\epsilon)^2 T_{P_*^{(k)}}. \tag{4.59}$$

*Proof.* Let

$$\mathcal{S} = \{p_{i_1}, p_{i_2}, \ldots, p_{i_l}\} \tag{4.60}$$

be the set (sequence) of $p$-profiles (in order) chosen by the Greedy algorithm, and let

$$\mathcal{S}^* = \{p_{j_1}, p_{j_2}, \ldots, p_{j_k}\} \tag{4.61}$$

be an optimal set of profiles. Since $\mathcal{S}^*$ satisfies the set constraint, we know that

$$\mathcal{S}^* \nsubseteq \mathcal{S} \setminus \{p_{i_l}\},$$

and assume without loss of generality

$$p_{j_k} \in \mathcal{S}^* \setminus (\mathcal{S} \setminus \{p_{i_l}\}). \tag{4.62}$$

We now claim that for all $1 \leq r \leq l$

$$p_{i_r} \geq \frac{p_{j_k}}{(1+\epsilon)^4}. \tag{4.63}$$

Indeed, suppose this was not the case. Then, there exists some $s \in [0, S_{end}]$ such that

$$p_{i_r}(s) < \frac{p_{j_k}(s)}{(1+\epsilon)^4}. \tag{4.64}$$

This implies

$$\begin{aligned} q_{i_r}(s) &\leq (1+\epsilon) p_{i_r}(s) \\ &< \frac{p_{j_k}(s)}{(1+\epsilon)^3} \\ &\leq \frac{(1+\epsilon) q_{j_k}(s)}{(1+\epsilon)^3} \\ &\leq \frac{q_{j_k}(s)}{(1+\epsilon)^2}. \end{aligned} \tag{4.65}$$

However, by the conditions of the theorem, $q_{i_r}$ and $q_{j_k}$ are related by a scalar multiple and so

$$q_{i_r} < \frac{q_{j_k}}{(1+\epsilon)^2}. \tag{4.66}$$

This implies

$$p_{i_r} \leq (1+\epsilon) q_{i_r} < \frac{q_{j_k}}{1+\epsilon} \leq p_{j_k}; \tag{4.67}$$

in other words, the profile $p_{i_r}$ is strictly dominated by profile $p_{j_k}$ and hence cannot

be selected in its stead by the Incremental Greedy algorithm. As a result, we have

$$\begin{aligned}\tau(\mathcal{S}) &= \tau(p_{i_1} \wedge p_{i_2} \wedge \cdots \wedge p_{i_l}) \\ &\leq \tau\left(\frac{p_{j_k}}{(1+\epsilon)^4}\right) \\ &= (1+\epsilon)^2 \tau(p_{j_k}) \\ &\leq (1+\epsilon)^2 \tau(\mathcal{S}*)\end{aligned} \tag{4.68}$$

and the claim follows. □

Ultimately, we have the following analogue of Theorem 12.

**Theorem 14.** *Consider the task in which the agent has to traverse a path $\gamma$ with constant orientation $\mathbf{R}$ with respect to the world frame. For this theorem, we assume $\gamma$ is a segment of length $L$ that belongs to line $\rho$. Suppose the depth of all $n$ landmarks lying within field of view of the on-board camera lies in the range $[\kappa_1 L, \kappa_2 L]$ at every point along the path, where $0 < \kappa_1 \leq \kappa_2 \leq +\infty$. Then, the Incremental Greedy algorithm produces a feasible square speed profile whose execution time is within a factor of $\left(\frac{1+1/\kappa_1}{1+1/\kappa_2}\right)^2$ of the optimum.*

### 4.4.3 Changing Selection of Landmarks

In this section, we consider selecting the optimal subset of landmarks for every given leg of the journey, where the chosen set is allowed to vary from one leg to another. We show that we may use either of K-Fastest or the Incremental Greedy algorithm for every stage separately and combine resulting subsets to give an algorithm with instance-dependent optimality guarantees. An interesting artifact is that the approximation factor approaches unity as the size of the longest leg of the journey tends to zero.

**Theorem 15.** *Let $\mathcal{P}$ be a set of p-profiles defined on $[0, S_{end}]$, partitioned into $H$ groups:*

$$\mathcal{P} = \mathcal{P}_1 \sqcup \mathcal{P}_2 \sqcup \cdots \sqcup \mathcal{P}_H. \tag{4.69}$$

*For every $1 \leq i \leq q$, we assume there exists a set of completely ordered profiles $\{q_j^{(i)} = \alpha_j^{(i)} r_i \mid \alpha_j^{(i)} \in (0,\infty), r_i \in C([a_i, b_i], (0,\infty)))\}_{j \in \mathcal{P}_i}$ such that for every $j \in \mathcal{P}_i$ we have*

$$\left|\left|\frac{p_j}{q_j^{(i)}}\right|\right|_\infty, \left|\left|\frac{q_j^{(i)}}{p_j}\right|\right|_\infty \leq 1 + \epsilon. \tag{4.70}$$

*Consider the task of choosing subsets $\mathcal{S}_1 \subseteq \mathcal{P}_1, \ldots, \mathcal{S}_H \subseteq \mathcal{P}_H$ which minimize the execution time of their pointwise minimum*

$$\tau(\bigwedge_{1 \leq i \leq H} \bigwedge_{j \in \mathcal{S}_i} p_j) \tag{4.71}$$

*subject to the constraint that*

$$\mathcal{F}^i(\mathcal{S}_i) \geq \lambda_{min}^{(i)} \quad \forall 1 \leq i \leq H. \tag{4.72}$$

*Then, applying the K-Fastest/Incremental Greedy algorithm to each partition of $\mathcal{P}$ individually and combining the resulting profiles yields a feasible profile with execution time by at most a factor of $(1+\epsilon)^2$ slower than the optimum.*

*Proof.* Suppose an optimal selection of subsets of profiles consists of

$$\mathcal{S}_i^* \subseteq \mathcal{P}_i. \tag{4.73}$$

By the same argument as in the proof of Theorem 13, we know that

$$\bigwedge_{i \in \mathcal{S}_i} h_i \geq \frac{1}{(1+\epsilon)^4} \bigwedge_{i \in \mathcal{S}_i^*} h_i, \tag{4.74}$$

and so

$$\bigwedge_{1 \leq i \leq H} \bigwedge_{i \in \mathcal{S}_i} h_i \geq \frac{1}{(1+\epsilon)^4} \bigwedge_{1 \leq i \leq H} \bigwedge_{i \in \mathcal{S}_i^*} h_i. \tag{4.75}$$

The result now follows by remarks on properties of $\tau(\cdot)$ given before Claim 2.

□

The specific application of this theorem would be as follows. Let $[0, S_{end}]$ be

partitioned into $H$ intervals $[a_1, b_1], \ldots, [a_H, b_H]$, where

$$0 = a_1 < b_1 = a_2 < b_2 \cdots a_H < b_H = S_{end}.$$

Let $\mathcal{P}_i$ consist of upper bounds on the square speed profile induced by landmarks that are visible for the portion of the path that corresponds to $[a_i, b_i]$. In particular, any such profile $p_j \in \mathcal{P}_i$ is of the form

$$p_j(s) = \begin{cases} M_j(s), & s \in [a_i, b_i] \\ +\infty, & otherwise. \end{cases} \tag{4.76}$$

Then, seeking $q_j^{(i)}$ of the same form (finite on $[a_i, b_i]$ and infinite otherwise), we get via Lemma 2

$$(1+\epsilon) \leq \max_{1 \leq i \leq H} \frac{1 + \frac{z_C \cdot \mathbf{n}(b_i - a_i)}{\kappa_1}}{1 + \frac{z_C \cdot \mathbf{n}(b_i - a_i)}{\kappa_2}}. \tag{4.77}$$

An important property of the relation above is that

$$\max_{1 \leq i \leq H} \frac{1 + \frac{z_C \cdot \mathbf{n}(b_i - a_i)}{\kappa_1}}{1 + \frac{z_C \cdot \mathbf{n}(b_i - a_i)}{\kappa_2}} \to 0 \quad \text{as} \quad \max_{1 \leq i \leq H} (b_i - a_i) \to 0. \tag{4.78}$$

In other words, as the frequency with which the agent can switch its choice of landmarks to track increases, our conceptually simple and efficient algorithm converges to the optimal solution.

## 4.5 Simulation and Experimental Results

### 4.5.1 Experimental Setup

For simplicity, we assume that the quadrotor used in the experiments is modelled as a double integrator with a camera attached on a 2-axis stabilized gimbal. The camera is assumed to be oriented to face the direction of heading of the quadrotor. Thus, the

Figure 4-2: Image captured at the start of the trajectory

Figure 4-3: Image captured at the end of the trajectory

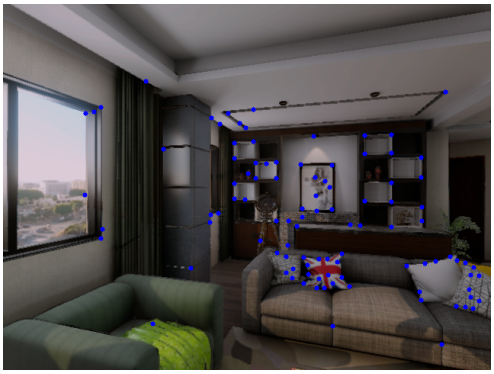


Figure 4-4: Features chosen using the quality metric of GoodFeaturesToTrack

dynamics of the quadrotor can be simply written as:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} a_x - g \\ a_y \\ a_z \end{bmatrix}, \quad R_C^W = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.79}$$

Figure 4-5: Features chosen by the K-Fastest method

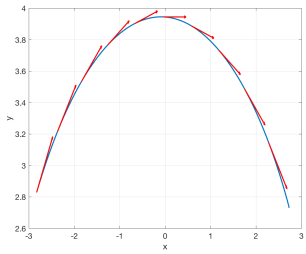



Figure 4-6: The Arc trajectory

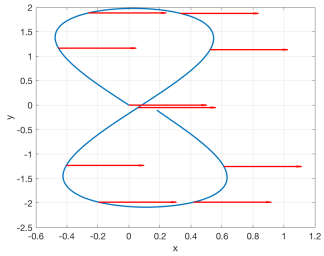



Figure 4-7: The Figure 8 trajectory segment

where $\psi$ is the angle the projection of the body $x$ axis of the quadrotor onto the world $x$-$y$ plane forms with the world $x$ axis. The navigation environment (see Figure 4-2) is generated using the FlightGoggles photorealistic simulator described in [24]. For the experiments we generate four trajectory segments shown in Figures 4-6, 4-7, 4-8, and 4-9.

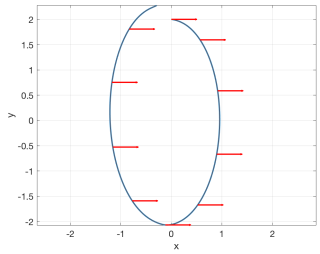



Figure 4-8: The Oval trajectory segment

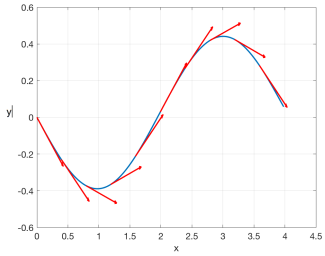



Figure 4-9: The Slalom trajectory segment

### 4.5.2 Algorithm Comparison

In this experiment, we consider the performance of all of the algorithms mentioned in Section 4.3. The key performance indices measured in this experiment are the execution time and the computation time. We report these indices for the problem of selecting 50 landmarks from an available 500 landmarks in the environment. For all the timing experiments, the speed profiles corresponding to individual landmarks are precomputed and the timing only includes the combinatorial selection component of the task. All experiments for computation time are performed using an intel i9 7900X computer using a python implementation of the algorithms presented here, and the optimization package [25] was allowed to use all available 40 threads. The results of this evaluation are summarized in Table 4.1. As can be seen in the table, the execution times generated by the algorithms are nearly identical, but the compute time of the K-Fastest algorithm is orders of magnitude smaller.

| Algorithm | Execution Time [s] | | Computation Time [s] | |
|---|---|---|---|---|
| Trajectory | Arc | Slalom | Arc | Slalom |
| K-fastest | 2.3465 | 2.9094 | 0.0006 | 0.0006 |
| Incremental algorithm | 2.3565 | 2.8889 | 2.1481 | 3.7412 |
| MILP (Gurobi) | 2.2828 | 2.8669 | 258.4733 | 48.97485 |

Table 4.1: Results of the algorithm comparison

### 4.5.3 Feature Tracking experiment

For this experiment, a point cloud is generated using stereo matching and the pixels with valid depth values are then used for corner extraction using GoodFeaturesToTrack. The extracted corners form the candidate point cloud for landmark selection. We use the same double order integrator model of the quadrotor, even though its actual dynamics are significantly more involved. The rationale for this lay in assessing whether our idealized planning procedure would naturally transfer to more complicated, but related dynamics models. The trajectory and features are specified by the K-Fastest algorithm. We compare the number of features tracked by a standard KLT tracker for the same number of features selected by the quality metric and the K-Fastest method. An example of landmarks selected by the two algorithms is shown in Figures 4-4 and 4-5. The results are summarized in Table 4.2. As can be seen in the table, the K-Fastest algorithm consistently results in a higher number of successfully tracked landmarks.

| Trajectory | Number of Tracked Landmarks | |
|---|---|---|
| | GFTT Quality | K-Fastest |
| Oval | 34 | 46 |
| Figure 8 | 23 | 44 |
| Slalom | 45 | 58 |

Table 4.2: Table showing the number of successfully tracked landmarks over the trajectory with k=100

## 4.6 Conclusion

In this chapter, we considered the problem of joint landmark selection and time optimal path parametrization for a wide range of second order integrator agents. The problem is challenging as it involves a mixture of discrete and continuous optimization. For modular criteria on feasibility of selections of landmarks, we reduced the problem to a mixed integer linear optimization problem. For general set functions, we proposed two computationally efficient algorithms for which we developed instance-dependent suboptimality guarantees for a relevant subclass of motions. Future work, however, will address whether guarantees for these algorithms hold up for *all* sufficiently smooth geometric paths.

# Chapter 5

# Conclusion

## 5.1 Recapitulation of Contributions

In this thesis, we considered three problems in the area of perception-aware motion planning on board micro aerial vehicles. The underlying theme lay in discovering and analyzing computationally efficient algorithms with performance guarantees that bring us a step closer to allowing such agile vehicles execute tasks autonomously at operational speeds.

The first problem we considered addressed demonstrating the asymptotic optimality of a time-optimal path parametrization algorithm for a wide class of second order agents. For this, we ventured into the field of non-smooth analysis to characterize the optimal solution from first principles, and used the characterization to show the convergence of the solution output by the algorithm to its true value.

The second of the three problems required maintaining desired landmarks within field of view of the camera on board a quadrotor while minimizing its traversal time of a specified path. We demonstrated the convexity such constraints impose on the time parametrization of the path and we also showed how to losslessly compress them in an efficient, parallelizable way. Ultimately, this let us develop an intuitive algorithm, Regularized-TOPP, that computes feasible time parametrizations with objective values competitive to those output by a generic nonlinear optimization solver taking orders of magnitude longer to compute.

The high level goal of the third problem lay in ensuring computationally-efficient acquisition of future visual measurements via motion planning. Indeed, extracting the change in bearing to desired landmarks between consecutive images incurs a computational burden that grows with the range of their apparent motion on the sensor array of the camera. The latter, in turn, increases with the speed of the agent. As a result, we considered the problem of minimizing the traversal time of a specified path, while maintaining the projection speeds of a sufficient number of landmarks below a specified threshold. We presented two polynomial-time algorithms that are empirically performant with respect to a globally-optimal exponential-time algorithm, and we showed their instance-dependent approximation guarantees for a relevant subclass of problems.

## 5.2 Limitations of Results and Future Work

Nevertheless, the work presented in this thesis is only a stepping stone towards achieving our aim of developing fully autonomous, robust MAVs. While listing necessary future steps for such a problem with many moving parts may be a futile exercise, we do use the opportunity here to discuss ways in which results in this thesis can be strengthened.

The problem from chapter 2 leaves open a natural question: for a given error tolerance, what is the necessary discretization size necessary to attain it? It goes without saying that there is a huge difference between knowing the problem can be solved with one hundred instead of one million discretization points. It is highly likely the key to answering such a question lies in developing convergence rates that were hinted at the very end of chapter 2. We posited, but were not able to show, that for the stated algorithm, the error is inversely proportional to the spacing between consecutive discretization points. To put matters into context, this amounts to increasing the number of discretization points tenfold in order to gain one digit in accuracy. Even if true, such state of affairs might be unsatisfactory for some applications. We believe that one interesting line of work might bring in the analogy from the literature on

numerical integration of ordinary differential equations. Indeed, we mentioned that the algorithm in chapter 2 may be viewed, with a slight stretch of imagination, as a generalization of the Forward Euler method for ODEs. However, we know there are many other, more accurate methods for this procedure, a simple example being the Trapezoidal rule. If the analogy from literature on differential equations were to carry over to our case, an adaptation of the Trapezoidal transcription rule would yield huge savings: *two decimal* digits of additional precision for a tenfold increase in the number of discretization points. Naturally, whether this is actually true is still unknown. However, resolving such a question would further increase the computational efficiency of the studied algorithm.

In chapter 3, we considered the perception-aware TOPP problem for a quadrotor equipped with a forward-facing camera. Such a configuration is common for vehicles designed to navigate indoor, cluttered environments. However, for large-scale applications such as surveillance missions, a downward-facing camera might be more appropriate. In these circumstances, maintaining desired regions within field of view at every point along the path is still a convex problem. However, it seems that losslessly compressing field of view constraints in such a scenario is not amenable to the approach we had described. Dealing with such constraints might be one interesting line of future work that can build off of the insights developed in this thesis. Another aspect of our problem we were not able to solve was answering the following question: given a path of flat outputs, can the quadrotor traverse it subject to bounds on its individual motor thrusts as well as perception constraints. Of course, the key difficulty here is that the quasi-static trajectory might not satisfy perception constraints. In such a scenario, our approach for selecting landmarks would not work, and it is not clear how much more difficult the latter problem would be from a computational standpoint, even if we were to assume the control input of the quadrotor was its cumulative thrust (i.e. its moment of inertia matrix was negligible).

The mathematical core of chapter 4 was a subset selection problem that arose out of a specific geometric setup. The first question we would like to have answered was: is it NP-hard? For example, if the translational path was specified as a concatena-

tion of polynomials with rational coefficients, and if the coordinates of the landmarks were rational too, what would be the minimum execution time of the path subject to maintaining the projection speed of a suitably large fixed subset of landmarks below a given threshold? However, with a more optimistic outlook, the question still remains: is the K-Fastest algorithm suboptimal by a constant factor for a general class of motions, and not just those involving straight line motion with no gyration? How about the Incremental Greedy algorithm? We believe that answering such questions, in addition to being interesting in their own right from a complexity-theoretic standpoint, might serve as guides for future development of algorithms for visual attention of computationally-constrained MAVs.

# Bibliography


[1] Markus W Achtelik, Simon Lynen, Stephan Weiss, Margarita Chli, and Roland Siegwart. Motion- and Uncertainty-aware Path Planning for Micro Aerial Vehicles. *Journal of Field Robotics*, 31(4):676–698, 2014.

[2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building Rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[3] Pasquale Antonante, Vasileios Tzoumas, Heng Yang, and Luca Carlone. Outlier-Robust Estimation: Hardness, Minimally Tuned Algorithms, and Applications. *IEEE Transactions on Robotics*, 38(1):281–301, 2022.

[4] Karl Johan Åström and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2010.

[5] Dimitri Bertsekas. *Dynamic Programming and Optimal Control: Volume I*. Athena Scientific, 2012.

[6] James Bobrow, S. Dubowsky, and J. S. Gibson. Time-Optimal Control of Robotic Manipulators Along Specified Paths. *International Journal of Robotics Research - IJRR*, 4:3–17, 09 1985.

[7] Richard H. Byrd, Mary E. Hribar, and Jorge Nocedal. An Interior Point Algorithm for Large-Scale Nonlinear Programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[8] Luca Carlone and Sertac Karaman. Attention and Anticipation in Fast Visual-Inertial Navigation. *IEEE Transactions on Robotics*, 35(1):1–20, 2019.

[9] Henry Carrillo, Ian Reid, and José A. Castellanos. On the comparison of uncertainty criteria for active SLAM. In *2012 IEEE International Conference on Robotics and Automation*, pages 2080–2087, 2012.

[10] H. Chitsaz and S. M. LaValle. Time-optimal Paths for a Dubins airplane. In *2007 46th IEEE Conference on Decision and Control*, pages 2379–2384, 2007.

[11] Eden Chlamtáč, Michael Dinitz, and Yury Makarychev. Minimizing the Union: Tight Approximations for Small Set Bipartite Vertex Expansion. In *Proceedings*

*of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 881–899. SIAM, 2017.

[12] L. Consolini, M. Locatelli, A. Minari, Á. Nagy, and I. Vajk. Optimal Time-Complexity Speed Planning for Robot Manipulators. *IEEE Transactions on Robotics*, 35(3):790–797, 2019.

[13] Luca Consolini, Marco Locatelli, Andrea Minari, and Aurelio Piazzi. An optimal complexity algorithm for minimum-time velocity planning. *Systems & Control Letters*, 103:50–57, 05 2017.

[14] Gabriele Costante, Jeffrey Delmerico, Manuel Werlberger, Paolo Valigi, and Davide Scaramuzza. *Exploiting Photometric Information for Planning Under Uncertainty*, pages 107–124. Springer International Publishing, Cham, 2018.

[15] Gábor Csorvási, Ákos Nagy, and Istvan Vajk. Near Time-Optimal Path Tracking Method for Waiter Motion Problem. volume 50, pages 4929–4934, 07 2017.

[16] Raffaello D'Andrea. Guest Editorial Can Drones Deliver? *IEEE Transactions on Automation Science and Engineering*, 11(3):647–648, 2014.

[17] Andrew Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1403–1410 vol.2, 2003.

[18] Alexander Domahidi, Eric Chu, and Stephen Boyd. ECOS: An SOCP solver for Embedded Systems. In *2013 European Control Conference (ECC)*, pages 3071–3076, 2013.

[19] Davide Falanga, Philipp Foehn, Peng Lu, and Davide Scaramuzza. PAMPC: Perception-Aware Model Predictive Control for Quadrotors. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8, 2018.

[20] Davide Falanga, Elias Mueggler, Matthias Faessler, and Davide Scaramuzza. Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5774–5781, 2017.

[21] Kaveh Fathian, Kasra Khosoussi, Yulun Tian, Parker Lusk, and Jonathan P. How. CLEAR: A Consistent Lifting, Embedding, and Alignment Rectification Algorithm for Multiview Data Association. *IEEE Transactions on Robotics*, 36(6):1686–1703, 2020.

[22] Uriel Feige. A Threshold of ln n for Approximating Set Cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[23] Michel Fliess, Jean Lévine, Philippe Martin, and Pierre Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. *International Journal of Control*, 61(6):1327–1361, 1995.

[24] Winter Guerra, Ezra Tal, Varun Murali, Gilhyun Ryou, and Sertac Karaman. Flightgoggles: Photorealistic Sensor Simulation for Perception-driven Robotics using Photogrammetry and Virtual Reality. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6941–6948, 2019.

[25] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022.

[26] Kris K. Hauser. Fast interpolation and time-optimization with contact. *The International Journal of Robotics Research*, 33:1231 – 1250, 2014.

[27] Markus Hehn and Raffaello D'Andrea. Real-Time Trajectory Generation for Quadrocopters. *IEEE Transactions on Robotics*, 31(4):877–892, 2015.

[28] Juha Heinonen. Lectures on Lipschitz analysis. *Rep. Dept. Math. Stat*, 100, 01 2005.

[29] Rishabh K. Iyer and Jeff A. Bilmes. Submodular Optimization with Submodular Cover and Submodular Knapsack Constraints. *Advances in Neural Information Processing Systems*, 26, 2013.

[30] Woo Yeon Jeong and Kyoung Mu Lee. Visual SLAM with Line and Corner Features. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2570–2575, 2006.

[31] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

[32] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.

[33] R. Kannan and C.K. Krueger. *Advanced Analysis: on the Real Line*. Universitext. Springer New York, 2012.

[34] Jonathan Kelly, Nicholas Roy, and Gaurav S. Sukhatme. Determining the Time Delay Between Inertial and Visual Sensor Measurements. *IEEE Transactions on Robotics*, 30(6):1514–1523, 2014.

[35] Hassan K Khalil. *Nonlinear systems; 3rd ed.* Prentice-Hall, Upper Saddle River, NJ, 2002.

[36] Vijay Kumar and Nathan Michael. Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research*, 31(11):1279–1291, 2012.

[37] Tobias Kunz and Mike Stilman. Time-Optimal Trajectory Generation for Path Following with Bounded Acceleration and Velocity. 07 2012.

[38] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[39] Keuntaek Lee, Jason Gibson, and Evangelos A. Theodorou. Aggressive Perception-Aware Navigation Using Deep Optical Flow Dynamics and PixelMPC. *IEEE Robotics and Automation Letters*, 5(2):1207–1214, 2020.

[40] Ronen Lerner, Ehud Rivlin, and Ilan Shimshoni. Landmark Selection for Task-Oriented Navigation. *IEEE Transactions on Robotics*, 23(3):494–505, 2007.

[41] Thomas Lipp and Stephen Boyd. Minimum-time speed optimisation over a fixed path. *International Journal of Control*, 87, 02 2014.

[42] Sikang Liu, Kartik Mohta, Nikolay Atanasov, and Vijay Kumar. Search-Based Motion Planning for Aggressive Flight in SE(3). *IEEE Robotics and Automation Letters*, 3(3), 2018.

[43] Fangchang Ma and Sertac Karaman. Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4796–4803, 2018.

[44] Yi Ma, Stefano Soatto, Jana Košecká, and Shankar Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*, volume 26. Springer, 2004.

[45] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE International Conference on Robotics and Automation*, pages 2520–2525, 2011.

[46] Nathan Michael, Shaojie Shen, Kartik Mohta, Vijay Kumar, Keiji Nagatani, Yoshito Okada, Seiga Kiribayashi, Kazuki Otake, Kazuya Yoshida, Kazunori Ohno, Eijiro Takeuchi, and Satoshi Tadokoro. *Collaborative Mapping of an Earthquake Damaged Building via Ground and Aerial Robots*, pages 33–47. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[47] Christian Mostegel, Andreas Wendel, and Horst Bischof. Active Monocular Localization: Towards Autonomous Monocular Exploration for Multirotor MAVs. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3848–3855, 2014.

[48] Mark W. Mueller, Markus Hehn, and Raffaello D'Andrea. A Computationally Efficient Motion Primitive for Quadrocopter Trajectory Generation. *IEEE Transactions on Robotics*, 31(6):1294–1310, 2015.

[49] Varun Murali, Igor Spasojevic, Winter Guerra, and Sertac Karaman. Perception-aware trajectory generation for aggressive quadrotor flight using differential flatness. In *2019 American Control Conference (ACC)*, pages 3936–3943, 2019.

[50] Richard M. Murray. Optimization-Based Control. *California Institute of Technology, CA*, pages 111–128, 2009.

[51] Akos Nagy and Istvan Vajk. LP-based Velocity Profile Generation for Robotic Manipulators. *International Journal of Control*, 91:1–19, 01 2017.

[52] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical programming*, 14(1):265–294, 1978.

[53] Huy Nguyen and Quang Cuong Pham. Time-Optimal Path Parameterization of Rigid-Body Motions: Applications to Spacecraft Reorientation. *Journal of Guidance, Control, and Dynamics*, 39:1–5, 01 2016.

[54] Kyel Ok, Katherine Liu, Kris Frey, Jonathan P. How, and Nicholas Roy. Robust Object-based SLAM for High-speed Autonomous Navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 669–675, 2019.

[55] Alex Olshevsky. Minimal Controllability Problems. *IEEE Transactions on Control of Network Systems*, 1(3):249–258, 2014.

[56] Christos H. Papadimitriou and John N. Tsitsiklis. The Complexity of Markov Decision Processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[57] Jufeng Peng and Srinivas Akella. Coordinating Multiple Robots with Kinodynamic Constraints Along Specified Paths. *I. J. Robotic Res.*, 24:295–310, 04 2005.

[58] Bryan Penin, Riccardo Spica, Paolo Robuffo Giordano, and François Chaumette. Vision-based minimum-time trajectory generation for a quadrotor UAV. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6199–6206, 2017.

[59] Friedrich Pfeiffer and Rainer Johanni. A Concept for Manipulator Trajectory Planning. *IEEE Journal of Robotics and Automation*, RA-3:115 – 123, 05 1987.

[60] Hung Pham and Quang Cuong Pham. A New Approach to Time-Optimal Path Parameterization Based on Reachability Analysis. *IEEE Transactions on Robotics*, 34:645 – 659, 06 2018.

[61] Quang Cuong Pham. A General, Fast, and Robust Implementation of the Time-Optimal Path Parameterization Algorithm. *IEEE Transactions on Robotics*, 30, 12 2013.

[62] Charles Richter, Adam Bry, and Nicholas Roy. *Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments*, pages 649–666. Springer International Publishing, Cham, 2016.

[63] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group. *The International Journal of Robotics Research*, 38(2-3):95–125, 2019.

[64] David M. Rosen, Michael Kaess, and John J. Leonard. RISE: An Incremental Trust-Region Method for Robust Online Sparse Least-Squares Estimation. *IEEE Transactions on Robotics*, 30(5):1091–1108, 2014.

[65] Inkyu Sa, Zetao Chen, Marija Popović, Raghav Khanna, Frank Liebisch, Juan Nieto, and Roland Siegwart. weedNet: Dense Semantic Weed Classification Using Multispectral Images and MAV for Smart Farming. *IEEE Robotics and Automation Letters*, 3(1):588–595, 2018.

[66] Seyed Abbas Sadat, Kyle Chutskoff, Damir Jungic, Jens Wawerla, and Richard Vaughan. Feature-rich path planning for robust navigation of MAVs with Mono-SLAM. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3870–3875, 2014.

[67] Thomas Sayre-McCord, Winter Guerra, Amado Antonini, Jasper Arneberg, Austin Brown, Guilherme Cavalheiro, Yajun Fang, Alex Gorodetsky, Dave McCoy, Sebastian Quilter, Fabian Riether, Ezra Tal, Yunus Terzioglu, Luca Carlone, and Sertac Karaman. Visual-Inertial Navigation Algorithm Development Using Photorealistic Camera Simulation in the Loop. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2566–2573, 2018.

[68] Manohar Shamaiah, Siddhartha Banerjee, and Haris Vikalo. Greedy sensor selection: Leveraging submodularity. In *49th IEEE Conference on Decision and Control (CDC)*, pages 2572–2577, 2010.

[69] Matthew Sheckells, Gowtham Garimella, and Marin Kobilarov. Optimal Visual Servoing for differentially flat underactuated systems. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5541–5548, 2016.

[70] Shaojie Shen, Nathan Michael, and Vijay Kumar. Autonomous multi-floor indoor navigation with a computationally constrained MAV. In *2011 IEEE International Conference on Robotics and Automation*, pages 20–25, 2011.

[71] Jianbo Shi and Tomasi. Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[72] Zvi Shiller and Hsueh-Hen Lu. Computation of Path Constrained Time Optimal Motions With Dynamic Singularities. *Journal of Dynamic Systems Measurement and Control-transactions of The Asme - J DYN SYST MEAS CONTR*, 114, 03 1992.

[73] J. J. E. Slotine and H. S. Yang. Improving the efficiency of time-optimal path-following algorithms. *IEEE Transactions on Robotics and Automation*, 5(1):118–124, 1989.

[74] Igor Spasojevic, Varun Murali, and Sertac Karaman. Asymptotic Optimality of a Time Optimal Path Parametrization Algorithm. *IEEE Control Systems Letters*, 3:835–840, 2019.

[75] Igor Spasojevic, Varun Murali, and Sertac Karaman. Joint Feature Selection and Time Optimal Path Parametrization for High Speed Vision-Aided Navigation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5931–5938, 2020.

[76] Igor Spasojevic, Varun Murali, and Sertac Karaman. Perception-aware time optimal path parameterization for quadrotors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3213–3219, 2020.

[77] Maxim Sviridenko, Jan Vondrák, and Justin Ward. Optimal approximation for submodular and supermodular optimization with bounded curvature. *Mathematics of Operations Research*, 42(4):1197–1218, 2017.

[78] Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM Journal on Computing*, 40(6):1715–1737, 2011.

[79] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010.

[80] Jesus Tordesillas and Jonathan P. How. PANTHER: Perception-Aware Trajectory Planner in Dynamic Environments. *IEEE Access*, 10:22662–22677, 2022.

[81] Vasileios Tzoumas, Luca Carlone, George J. Pappas, and Ali Jadbabaie. LQG Control and Sensing Co-Design. *IEEE Transactions on Automatic Control*, 66(4):1468–1483, 2021.

[82] Diederik Verscheure, Bram Demeulenaere, Jan Swevers, Joris De Schutter, and Moritz Diehl. Time-Optimal Path Tracking for Robots: A Convex Optimization Approach. *IEEE Transactions on Automatic Control*, 54:2318 – 2327, 11 2009.

[83] Heng Yang, Pasquale Antonante, Vasileios Tzoumas, and Luca Carlone. Graduated Non-Convexity for Robust Spatial Perception: From Non-Minimal Solvers to Global Outlier Rejection. *IEEE Robotics and Automation Letters*, 5(2):1127–1134, 2020.

[84] Heng Yang, Jingnan Shi, and Luca Carlone. TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2021.

[85] Lintao Ye, Sandip Roy, and Shreyas Sundaram. On the Complexity and Approximability of Optimal Sensor Selection for Kalman Filtering. In *2018 Annual American Control Conference (ACC)*, pages 5049–5054, 2018.

[86] Guangcong Zhang and Patricio A. Vela. Good features to track for visual SLAM. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1373–1382, 2015.

[87] Guangcong Zhang and Patricio A. Vela. Optimally observable and minimal cardinality monocular SLAM. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5211–5218, 2015.

[88] Haifeng Zhang and Yevgeniy Vorobeychik. Submodular optimization with routing constraints. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[89] Zhijie Zhu, Edward Schmerling, and Marco Pavone. A convex optimization approach to smooth trajectories for motion planning with car-like robots. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 835–842, 2015.

[90] Leon Zlajpah. On time optimal path control of manipulators with bounded joint velocities and torques. *Proceedings of IEEE International Conference on Robotics and Automation*, 2:1572–1577 vol.2, 1996.

[91] Tolga Özaslan, Giuseppe Loianno, James Keller, Camillo J. Taylor, Vijay Kumar, Jennifer M. Wozencraft, and Thomas Hood. Autonomous Navigation and Mapping for Inspection of Penstocks and Tunnels With MAVs. *IEEE Robotics and Automation Letters*, 2(3):1740–1747, 2017.