2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

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

Igor Spasojevic, Varun Murali and Sertac Karaman

***Abstract*— We study a problem in vision-aided navigation in which an autonomous agent has to traverse a specified path in minimal time while ensuring extraction of a steady stream of visual percepts with low latency. Vision-aided robots extract motion estimates from the sequence of images of their on-board cameras by registering the change in bearing to landmarks in their environment. The computational burden of the latter procedure grows with the range of apparent motion undertaken by the projections of the landmarks, incurring a lag in pose estimates that should be minimized while navigating at high speeds. This paper addresses the problem of selecting a desired number of landmarks in the environment, together with the time parametrization of the path, to allow the agent execute it in minimal time while both (i) ensuring the computational burden of extracting motion estimates stays below a set threshold and (ii) respecting the actuation constraints of the agent. We provide two efficient approximation algorithms for addressing the aforementioned problem. Also, we show how it can be reduced to a mixed integer linear program for which there exist well-developed optimization packages. Ultimately, we illustrate the performance of our algorithms in experiments using a quadrotor.**

I. Introduction

Time-critical missions are a key aspect of numerous robotic applications. Examples include disaster response, search and rescue, and aid delivery. In order to execute specified tasks precisely, and, in particular, avoid fatal collisions with objects in their environment, autonomous systems require the ability to recover accurate estimates of their pose by inverting percepts from their on-board sensors. However, the path from raw sensor data to localized estimates via limited on-board computational resources incurs a latency. These delays can preclude rapid maneuvers that lie beyond the perception-actuation limits of the agent. Therefore, planning maneuvers with consideration of such constraints allows us to maximize the effectiveness of robotic platforms while ensuring their safety.

A common assortment of on-board sensors includes a combination of an inertial measurement unit (IMU) and a camera. An IMU supplies measurements of acceleration and angular velocity that can be used to perform dead reckoning over short time intervals. However, over longer horizons, such estimates accrue large drift that can be significantly mitigated using measurements supplied by the vision system. The latter typically take the form of differential bearings to landmarks, visually salient regions in the environment, whose projections are captured as features on the pixel array of the on-board camera. Extracting accurate bearing measurements efficiently is a problem in its own right, and some regimes of robot motion render it more challenging than others.

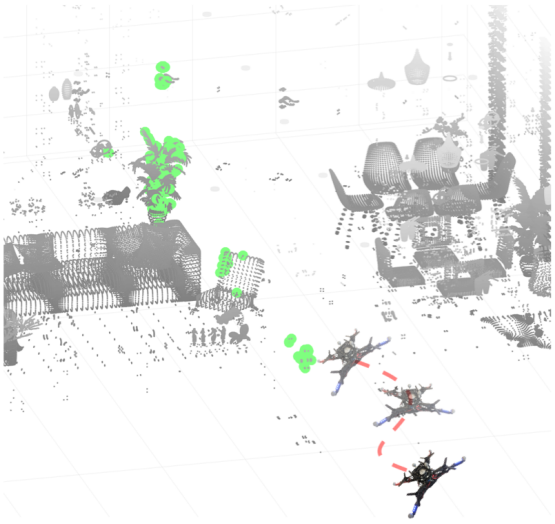


Fig. 1: The figure shows a quadrotor navigating a typical indoor environment. The selection of landmarks in the environment that allow the fastest execution of the path is shown in green.

In order to measure the change in bearing to a particular landmark between successive images, the agent must solve the problem of data association. In visual navigation, the task is commonly solved using an iterative search method such as the KLT feature tracker [1], 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 data association grows with the range of motion underwent by tracked features. The latter, in turn, depends on both the geometry of the scene as well as the trajectory executed by the robot. Tracking a larger set of landmarks or performing more rapid maneuvers increases the cost of data association. However, a greater volume of visual measurements yields more accurate state estimates, whereas faster motion increases the efficacy of the robot. This motivates us to consider a problem at the intersection of visual feature 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 problem of feature selection can be considered de-

*This work was supported in part by the Lockheed Martin Corporation.
All authors are with the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA. {igorspas, mvarun, sertac}@mit.edu

coupled from motion planning. The goal of attention and anticipation [2] is to select landmarks that will remain within field of view throughout the traversal of a segment of the trajectory. This procedure aids the visual navigation process by greedily selecting features ranked by an information gain metric formed by forward simulating the dynamics as a scoring function. [3] propose a stochastic greedy algorithm for feature subset selection for the least squares pose estimation problem. They leverage the submodularity of the problem to select an approximately optimal subset of features using the *max* log det metric in low running time.

There are also numerous works planning trajectories with perception constraints. [4] plan locally optimal dynamically feasible trajectories for quadrotors that minimize execution time and keep a specified target within the on-board camera's field of view. [5] present an algorithm that minimizes the integral of the magnitude of reprojection error of a specified set of landmarks along the planned path from the current position to a given final position. [6] incorporate the perception constraint in their model predictive control framework which minimizes the speed of a particular set of landmarks' projections. [7] formulate an objective function that combines a covisibility constraint, which ensures landmarks remain visible in between keyframes, and trajectory execution time that can leverage differential flatness of the quadrotor. [8] plan time optimal paths for maintaining a specified set of landmarks within field of view by leveraging hidden convexity in the problem.

A common approximate approach to the computationally challenging problem of planning trajectories for agents with complicated dynamics involves the path-velocity decomposition. It consists of first planning a collision-free geometric path which is then endowed with a time parametrization to yield a dynamically feasible trajectory. Our focus here is on the second stage of this procedure, the time optimal path parametrization problem (TOPP). Some of the first methods based on numerical integration were in the context of planning trajectories for robotic manipulators [9]–[11]. A more recent class of approaches uncovered a suitable parametrization of the problem which rendered it convex for a broad class of agents [12], including robotic manipulators, bipedal robots [13], and even some models of cars and aerial vehicles [14], [15]. The most recent class of algorithms [16], [17] further exploited the special structure of the problem to provide an extremely efficient iterative algorithm.

In this work, we consider the task of selecting a desired number of landmarks, together with planning an optimal time parametrization of a specifed path, that is both dynamically feasible and ensures computationally efficient extraction of changes in bearing of selected landmarks. Our contributions are: ***(1)*** efficient approximation algorithms for addressing the problem of joint feature selection and time optimal path parametrization, together with an analysis of their performance guarantees, and ***(2)*** a reduction of the problem to a mixed integer linear program, for which existing optimization packages supply the optimal solution within a matter of minutes for the size of problems that occur in practice.

II. Problem Formulation

A. Coordinate Frame Conventions

We denote coordinates of point $p \in \mathbb{A}^3$ with respect to reference frame A by ${}^A\mathbf{p} \in \mathbb{R}^3$. Particularly relevant will be the world frame, the body-fixed frame of the agent, and its camera-fixed frame, denoted by W, B, and C, respectively. For a generic pair of reference frames A and $\tilde{A}$, the transformation from A to $\tilde{A}$ will be denoted by $({}^A\mathbf{t}_{\tilde{A}}, {}^A\mathbf{R}_{\tilde{A}}) \in \mathbb{R}^3 \times SO(3)$; the equation ${}^A\mathbf{p} = {}^A\mathbf{R}_{\tilde{A}}{}^{\tilde{A}}\mathbf{p} + {}^A\mathbf{t}_{\tilde{A}}$ relates coordinates of the same physical point with respect to the two frames.

B. Problem Description

The problem under consideration consists of the following. Given an agent $\mathcal{A}$ with known dynamics $\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \mathbf{u}) \in \mathbb{R}^{n_q}$, where $\mathbf{q} \in \mathbb{R}^{n_q}$ and $\mathbf{u} \in C(\mathbf{q}) \in \mathbb{R}^{n_u}$ encode its state and actuation constraints, respectively, a set of landmarks with known static positions with respect to the world frame $\mathcal{L} = \{{}^W\mathbf{l}_1, ..., {}^W\mathbf{l}_n\}$, the desired number $k \in \{1, 2, ..., n\}$ of landmarks to track, and a regularly parametrized path $(\gamma, \mathbf{R}) : [0, S_{end}] \to \mathbb{R}^3 \times SO(3)$, find the shortest execution time of γ by $\mathcal{A}$ that is (i) dynamically feasible, and (ii) maintains the projection speed of each landmark in some *fixed* subset of $\mathcal{L}$, of cardinality k, onto the canvas of the camera on-board $\mathcal{A}$ below a specified threshold π_{max}.

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

$$ {}^C\mathbf{l}_i(s) = \underbrace{{}^C\mathbf{R}_B{}^B\mathbf{R}_W(s)({}^W\mathbf{l}_i - \gamma(s) - \mathbf{R}(s){}^B\mathbf{t}_C)}_{\text{function of } s \text{ and } {}^W\mathbf{l}_i \text{ only}} \tag{1} $$

with respect to C, and is captured as the point $\left[\frac{{}^C\mathbf{l}_i(s)_1}{{}^C\mathbf{l}_i(s)_3}, \frac{{}^C\mathbf{l}_i(s)_2}{{}^C\mathbf{l}_i(s)_3}\right]^T$ on the canvas of the camera. Equation 1 readily implies that for all $1 \le i \le n$, the agent maintains the projection speed of landmark i below threshold π_{max} if and only if

$$ \frac{ds}{dt}(s) \le U_i(s) \triangleq \frac{\pi_{max}}{\left|\left| \frac{d}{ds}({}^C\mathbf{l}_i(s)) \right|\right|_2} \quad \forall s \in [0, S_{end}]. \tag{2} $$

Note that $\frac{ds}{dt}$ represents the rate of increase of the parameter of the path $(\gamma(\cdot), \mathbf{R}(\cdot))$; it coincides with the translational speed of the agent when s corresponds to arc length of γ.

Methods in this paper are applicable to a broad class of agents with second order dynamics [14], including models of space vehicles, cars [15], [18], and Dubins' planes [19]. As a running example, we consider a fully actuated multicopter, whose dynamics are given by [20]

$$ \underbrace{\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{v}} \\ \dot{R} \\ \dot{\omega} \end{bmatrix}}_{\dot{\mathbf{q}}} = \begin{bmatrix} \mathbf{v} \\ \mathbf{g} \\ R[\omega]_\times \\ -J^{-1}\ \omega \times J\omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ R & 0 \\ 0 & 0 \\ 0 & J^{-1} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{c} \\ \tau \end{bmatrix}}_{\text{effective input } \mathbf{u}}, \tag{3} $$

where the effective input $\mathbf{u} \in \mathbb{R}^6$ is subject to convex membership constraints. An example of such a platform is a hexacopter with tilted motors.

C. Problem parametrization

In addition to selecting a subset of landmarks for tracking, we will seek the optimal *square speed profile* [21], $h : [0, S_{end}] \to [0, \infty)$ defined, with slight abuse of notation, via $h(s) = (\frac{ds}{dt}(s))^2 \ \forall s \in [0, S_{end}]$. The efficacy of such a parametrization of TOPP in motion planning was first captured in [12]. One of the main insights in [12] uncovered that upper bounds on norms of acceleration, given by $\mathbf{a} = \frac{1}{2}\gamma' h' + \gamma'' h$, induce convex constraints on h, while the execution time $\int_{s=0}^{S_{end}} \frac{ds}{\sqrt{h(s)}}$, is a convex function of h. Furthermore, in our current problem, the constraint of successfully tracking feature i amounts to a bound on the speed profile $h(s) \leq M_i(s) \triangleq U_i(s)^2 \ \forall s \in [0, S_{end}]$.

In general, the optimal h need not be everywhere differentiable. To see this, consider minimizing the traversal time of a straight line segment by an acceleration-constrained agent (encoded by $|h'| \leq 2a_{max}$). The optimal profile involves switching acceleration instantaneously from $+a_{max}$ to $-a_{max}$. To address such behaviour, we enforce actuation constraints in the form $D^+h(s) \leq f^+(s, h(s))$ and $D^-h(s) \geq f^-(s, h(s))$ [22], where [23]

$$\begin{aligned} D^+h(s) &\triangleq \limsup_{\delta \downarrow 0} \frac{h(s+\delta) - h(s)}{\delta} \\ D^-h(s) &\triangleq \liminf_{\delta \downarrow 0} \frac{h(s+\delta) - h(s)}{\delta} \end{aligned} \tag{4}$$

for suitably chosen functions f^+ and f^-. For instance, the example above would amount to setting $f^{\pm} = \pm 2a_{max}$. Therefore, we solve problem $P(f^+, f^-, \mathcal{L} = \{({}^W\mathbf{l}_j)_{j=0}^n\}$:

$$\begin{aligned} \underset{\substack{h:[a,b]\to[0,\infty) \\ \mathcal{F}\subseteq\mathcal{L},\ |\mathcal{F}|=k}}{\text{minimize}} \quad & \int_a^b \frac{ds}{\sqrt{h(s)}} \\ \text{subject to} \quad & D^+h(s) \leq f^+(s, h(s)),\ s \in [a,b), \\ & D^-h(s) \geq f^-(s, h(s)),\ s \in [a,b), \\ & 0 \leq h(s) \leq \min_{j \in \mathcal{F}} M_j(s),\ s \in [a,b]. \end{aligned} \tag{5}$$

III. Method

In this section, we describe three related methods for solving Problem 5. A solution consists of two components: a subset of features of cardinality k, and a dynamically feasible square speed profile which allows tracking selected features. We represent the optimal square speed profile by its values at a set of specified discretization points of the path. In particular, a discretization of interval $[a,b]$, denoted by $D = D([a,b], (s_i)_{i=0}^N)$, comprises an increasing sequence of discretization points $a = s_0 < s_1 < \cdots < s_N = b$, at which we compute the sequence $(\hat{h}_i)_{i=0}^N$ approximating $(h_i^*)_{i=0}^N$, where h^* is the optimal square speed profile. We numerically approximate the execution time of a profile $(h_i)_{i=0}^N$ by $\sum_{i=0}^{N-1} \frac{2(s_{i+1}-s_i)}{\sqrt{h_{i+1}}+\sqrt{h_i}}$.

Algorithm 1: Backward-Forward Algorithm

Data: $D = (s_i)_{i=0}^N$, $(B_u(s_i))_{i=0}^N$, f^+, f^-
Result: $(\hat{h}_i)_{i=0}^N$, τ
$h_N^{(b)} = B_u(s_N)$
for $i = N-1$ *to* 0 **do**
  $h_i^{(b)} \leftarrow \max\{h \mid 0 \leq h \leq B_u(s_i),$
    $h + f^-(s_i, h)(s_{i+1} - s_i) \leq h_{i+1}^{(b)}\}$
  if $h_i^{(b)} = -\infty$ **then**
    return null
  end
end
$h_0^{(f)} = h_0^{(b)}$
for $i = 1$ *to* N **do**
  $h_i^{(f)} \leftarrow \max\{h \mid 0 \leq h \leq h_i^{(b)},$
    $h \leq h_{i-1}^{(f)} + f^+(s_{i-1}, h_{i-1}^{(f)})(s_i - s_{i-1})\}$
  if $h_i^{(f)} = -\infty$ **then**
    return null
  end
end
return $(\hat{h}_i)_{i=0}^N = (h_i^{(f)})_{i=0}^N$, $\tau = \sum_{i=0}^{N-1} \frac{2(s_{i+1}-s_i)}{\sqrt{\hat{h}_{i+1}}+\sqrt{\hat{h}_i}}$

Algorithm 2: K-Fastest Algorithm

Data: $D = (s_i)_{i=0}^N$, $({}^W\mathbf{l}_j)_{j=1}^n$, f^+, f^-
Result: $(\hat{h}_i)_{i=0}^N$, $\mathcal{F}$
for $j = 1$ *to* n **do**
  $(B_u^j(s_i))_{i=0}^N \leftarrow$ *UpperBound*$(D, \gamma, {}^W\mathbf{l}_j)$
  $(\hat{h}_i^j)_{i=0}^N, T^j \leftarrow$ Backward-Forward$(D, (B_u^j), f^+, f^-)$
end
$\pi \leftarrow$ *Sort*$((T^j)_{j=1}^n)$
for $i = 0$ *to* N **do**
  $\hat{h}_i \leftarrow \min_{1 \leq j \leq k} \hat{h}_i^{\pi(j)}$
end
return $(\hat{h}_i)_{i=0}^n$, $\mathcal{F} = \{({}^W\mathbf{l}_{\pi(j)})_{j=1}^k\}$

All three methods can be formulated as two-stage procedures, the first stage being common to all three. Indeed, the first step of the approach consists of computing the time optimal feasible square speed profile for tracking every feature *individually*. This is done using the recently proposed Algorithm 1 [16]. Knowledge of the location of the feature allows us to translate the bound on its maximal projection speed to a pointwise bound on the square speed profile as obtained from Equation 2. 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 IV we will see that the optimal profile for tracking a *subset* of landmarks is just the pointwise minimum of the optimal profiles of its constituents. Hence, the second

Algorithm 3: Incremental Algorithm

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

stages of the three methods will primarily differ on how they use the information obtained from the first stage of the algorithm. The first method (Algorithm 2) selects the k landmarks with fastest profiles. The second method (Algorithm 3) starts with an empty set of landmarks and incrementally adds an unselected landmark which induces a minimal increase in execution time. The third method recasts the problem as a binary mixed integer linear program (Claim 1), which can then be fed to any of the off-the-shelf packages such as Gurobi [24].

Claim 1 *The task of optimally selecting k features as well as determining the optimal speed profile can be recovered from the solution of the following binary mixed integer linear program:*

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} (s_i - s_{i-1})\ y_i \\ s.t. \quad & y_i \ \geq \ \zeta_j \frac{1}{\sqrt{\hat{h}^{(j)}(s_i)}} \quad \forall\ 0 \leq i \leq N \\ & \sum_{j=1}^{n} \zeta_j \ \geq \ k \\ & \zeta_i \in \{0,1\}^n, \quad y \in \mathbb{R}^N. \end{aligned}$$

Denoting the optimal value of the solution by $(\zeta^*_{1:n}, y^*_{1:N})$, selected features correspond to indices i such that $\zeta_i^* = 1$, whereas y_j^* yields the amount of time allocated by the optimal profile for the segment $[s_j, s_{j+1}]$ for every $1 \leq j \leq N-1$.

IV. Analysis

A. Continuum Analysis

To analyze proposed algorithms, we draw upon several results in [22] 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. We note that result (a) implies that the time optimal profile for tracking a chosen subset of landmarks is the pointwise maximum of the set of such profiles.

The following claim justifies an efficient method for recovering the optimal speed profile for tracking a desired subset of features from the optimal profiles for tracking each of its elements individually.

Claim 2 *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*

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

Proof: Landmark j can be tracked with profile h iff 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. ■

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. For example, $\tau(p_1, p_2, ..., p_n) = \tau(p_1 \wedge p_2 \wedge \cdots \wedge p_n)$. Furthermore, since $\tau(h) = \int_{s=0}^{S_{end}} \frac{ds}{\sqrt{h(s)}}$, for every $\varepsilon > 0$ we have $\tau(\varepsilon h) = \frac{\tau(h)}{\sqrt{\varepsilon}}$. Also, τ is is monotonically decreasing: if $h_1 \geq h_2$ then $\tau(h_1) \leq \tau(h_2)$.

Claim 3 *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 $\varepsilon > 0$, define $\overline{h}_{M(1+\varepsilon)}$ analogously. Then, we have:*

$$\begin{aligned} \overline{h}_M \ \leq \overline{h}_{M(1+\varepsilon)} \ \leq \ (1+\varepsilon)\overline{h}_M \\ \tau(\overline{h}_{M(1+\varepsilon)}) \ \leq \tau(\overline{h}_M) \ \leq \ \sqrt{1+\varepsilon}\ \tau(\overline{h}_{M(1+\varepsilon)}) \end{aligned}$$

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 $M(1+\varepsilon)$, together with the

characterization of $\bar{h}_{M(1+\varepsilon)}$ 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

$$\frac{\varepsilon}{1+\varepsilon}0+\frac{1}{1+\varepsilon}\bar{h}_{M(1+\varepsilon)}$$

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

Theorem 1 *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 $\varepsilon>0$ be a positive real number such that*

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

for all $1\leq i\leq n$. In addition, assume that Q can be partially 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 k p-profiles and the subset of k individually fastest p-profiles, respectively, we have:

$$\tilde{T}_{P_*^{(k)}}\leq(1+\varepsilon)^2 T_{P_*^{(k)}}. \tag{6}$$

Proof: By assumption, the profiles $(q_i)_{i=1}^n$ have been sorted in order of execution time: $\tau(q_1)\leq\tau(q_2)\leq...\leq\tau(q_n)$. Let $\{p_{i_1},p_{i_2},...,p_{i_k}\}$ be the set of k p-profiles of shortest duration. Let $m=argmax_{1\leq j\leq k}\ \tau(q_{i_j})$, and define $\tilde{q}=q_{i_m}$. In other words, $\tilde{q}$ is the slowest q-profile matched to one of the profiles $p_{i_1},p_{i_2},...,p_{i_k}$. Thus, we have

$$\begin{aligned}\tau(p_{i_1},p_{i_2},...,p_{i_k})&\leq\tau\left(\frac{q_{i_1}}{1+\varepsilon},\frac{q_{i_2}}{1+\varepsilon},...,\frac{q_{i_k}}{1+\varepsilon}\right)\\&=\sqrt{1+\varepsilon}\ \tau(q_{i_1},q_{i_2},...,q_{i_k})\\&=\sqrt{1+\varepsilon}\ \tau(\tilde{q}),\end{aligned} \tag{7}$$

where the last equality follows from the fact that the profile $\tilde{q}$ is dominated by profiles $q_{i_1},q_{i_2},...,q_{i_k}$.

We now claim that $\tau(\tilde{q})\leq(1+\varepsilon)\tau(q_k)$. Indeed, assume for a contradiction $\tau(\tilde{q})>(1+\varepsilon)\tau(q_k)$. Then,

$$\begin{aligned}\tau(p_{i_m})&\geq\tau((1+\varepsilon)q_{i_m})\\&=\tau((1+\varepsilon)\tilde{q})\\&=\frac{1}{\sqrt{1+\varepsilon}}\tau(\tilde{q})\\&>(1+\varepsilon)^{1/2}\tau(q_k)\\&\geq(1+\varepsilon)^{1/2}\tau(q_j)\quad\forall\ 1\leq j\leq k,\end{aligned} \tag{8}$$

where the last line follows from the fact that q-profiles have been sorted in order of increasing execution time.

On the other hand, for all j we have

$$\tau(p_j)\leq\tau(\frac{q_j}{1+\varepsilon})=\sqrt{1+\varepsilon}\ \tau(q_j). \tag{9}$$

Hence, $\tau(p_{i_m})>\tau(p_j)$ for all $1\leq j\leq k$, and so p_{i_m} cannot be one of the k p-profiles with the smallest execution time, a contradiction. Thus,

$$\tau(p_{i_1},p_{i_2},...,p_{i_k})\leq(1+\varepsilon)^{3/2}\tau(q_k). \tag{10}$$

Similarly, for any set of profiles $p_{j_1},p_{j_2},...,p_{j_k}$ we have

$$\begin{aligned}\tau(p_{j_1},p_{j_2},...,p_{j_k})&\geq\tau((1+\varepsilon)q_{j_1},(1+\varepsilon)q_{j_2},...,(1+\varepsilon)q_{j_k})\\&=\frac{1}{\sqrt{1+\varepsilon}}\max_{1\leq r\leq k}\tau(q_{j_r})\\&\geq\frac{1}{\sqrt{1+\varepsilon}}\tau(q_k),\end{aligned} \tag{11}$$

where the final inequality follows from the ordering of the q-profiles. Hence, $\tilde{T}_{P_*^{(k)}}=\tau(p_{i_1},p_{i_2},...,p_{i_k})\leq(1+\varepsilon)^2 T_{P_*^{(k)}}$, as desired. ■

Theorem 2 *Consider the task in which the agent has to traverse a path γ with constant orientation R with respect to the world frame. For this theorem, we assume γ is a segment of length L that belongs to line ρ. Assume the distance of all n landmarks which remain within field of view of the on-board camera with field of view θ, are at a distance of at least κL from γ. Then, Algorithm 2 produces a feasible square speed profile whose execution time is within a factor of $((1+\frac{1}{\kappa})/\cos(\theta))^4$ of the optimum.*

Proof: Consider a landmark with coordinates ${}^W\mathbf{l}$. 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. Thus,

$${}^C\mathbf{l}(s)\ =\ \mathbf{R}^{-1}({}^W\mathbf{l}-\gamma(s)). \tag{12}$$

Given the bounded field of view of the camera, the magnitude of the velocity of the projection of the landmark is up to an absolute constant (namely $\frac{1}{\cos^2\theta}$) the rate of change of its bearing vector. Hence, the rate of change of the projection with respect to distance the agent has travelled along the path satisfies

$$\begin{aligned}&\left|\left|\frac{d}{ds}\overbrace{\left(\frac{{}^C\mathbf{l}(s)}{||{}^C\mathbf{l}(s)||_2}\right)}^{bearing}\right|\right|_2\leq\left|\left|\frac{d}{ds}\overbrace{\left(\frac{{}^C\mathbf{l}(s)}{{}^C\mathbf{l}(s)_3}\right)}^{projection}\right|\right|_2\\&\leq\frac{1}{\cos^2\theta}\left|\left|\frac{d}{ds}\left(\frac{{}^C\mathbf{l}(s)}{||{}^C\mathbf{l}(s)||_2}\right)\right|\right|_2\\&=\frac{1}{\cos^2\theta}\left|\left|\frac{{}^C\mathbf{l}(s)\times({}^C\mathbf{l}(s)'\times{}^C\mathbf{l}(s))}{||{}^C\mathbf{l}(s)||_2^3}\right|\right|_2\\&=\frac{1}{\cos^2\theta}\frac{||{}^C\mathbf{l}(s)'\times{}^C\mathbf{l}(s)||_2}{||{}^C\mathbf{l}(s)||_2^2}\end{aligned} \tag{13}$$

The magnitude of the rate of change of the bearing vector may be expressed as follows. Since $\mathbf{R}$ is assumed constant, we have

$${}^C\mathbf{l}(s)'=-\mathbf{R}^{-1}\gamma'(s) \tag{14}$$

implying that

$$\frac{||{}^C\mathbf{l}(s)' \times {}^C\mathbf{l}(s)||_2}{||{}^C\mathbf{l}(s)||_2^2} = \frac{||\gamma'(s) \times ({}^W\mathbf{l} - \gamma(s))||_2}{||{}^W\mathbf{l} - \gamma(s)||_2^2} = \frac{|\sin\angle(\gamma'(s),\ {}^W\mathbf{l} - \gamma(s))|}{||{}^W\mathbf{l} - \gamma(s)||_2} = \frac{dist({}^W\mathbf{l}, \rho)}{||{}^W\mathbf{l} - \gamma(s)||_2^2} \tag{15}$$

where $dist({}^W\mathbf{l}, \rho)$ denotes the orthogonal distance from point ${}^W\mathbf{l}$ to line ρ. Denoting the minimal and maximal distance between points of γ and ${}^W\mathbf{l}$ by $\underline{d}({}^W\mathbf{l}, \gamma)$ and $\overline{d}({}^W\mathbf{l}, \gamma)$ respectively, we have

$$\left(\frac{\overline{d}({}^W\mathbf{l}, \gamma)}{\underline{d}({}^W\mathbf{l}, \gamma)}\right)^{-1} \leq \frac{\frac{dist({}^W\mathbf{l}, \rho)}{||{}^W\mathbf{l} - \gamma(s)||_2^2}}{\frac{dist({}^W\mathbf{l}, \rho)}{\overline{d}({}^W\mathbf{l}, \gamma)\underline{d}({}^W\mathbf{l}, \gamma)}} \leq \left(\frac{\overline{d}({}^W\mathbf{l}, \gamma)}{\underline{d}({}^W\mathbf{l}, \gamma)}\right) \tag{16}$$

From the lower bound on $\underline{d}({}^W\mathbf{l}, \gamma)$, we get that $1 \leq \frac{\overline{d}({}^W\mathbf{l}, \gamma)}{\underline{d}({}^W\mathbf{l}, \gamma)} \leq (1 + \frac{1}{\kappa})$. Putting the latter relations together, we get that the profile $\pi_{max}^2 / || \frac{d}{ds}(\frac{{}^C\mathbf{l}(s)}{{}^C\mathbf{l}(s)_3}) ||_2^2$ is within a factor $\eta := ((1 + \frac{1}{\kappa})\frac{1}{\cos\theta})^2$ of the constant profile $\pi_{max}^2 / || \frac{1}{\cos^2\theta} \frac{dist({}^W\mathbf{l}, \rho)}{\overline{d}({}^W\mathbf{l}, \gamma)\underline{d}({}^W\mathbf{l}, \gamma)} ||_2^2$. Letting $(p_j)_{j=1}^n$ and $(q_j)_{j=1}^n$ be the time optimal feasible profiles not exceeding the former and latter profiles for every landmark ${}^W\mathbf{l} \in \mathcal{L}$, respectively, we can derive three things. Firstly, the set of $(q_j)_{j=1}^n$ may be partially ordered as in the definition of Theorem 1. Second, Claim 3 implies that for all j, profiles p_j and q_j are within a factor of η of each other. Ultimately, the latter two observations set the stage for an application of Theorem 1 which yields the desired result. ■

B. Computational Complexity Analysis

We split the computational complexity analysis of the three proposed methods into two parts. First, we will state the complexity of the first stage of the algorithm(s), the details of which can be found in [16], and [17]. Then we will analyze the complexity of the second stage of each method.

The first stage consists of executing Algorithm 1, whose running time is of order $O(NE)$, where E is the complexity of solving the one-dimensional optimization problem in every iteration of the for loops, and N is the number of discretization points of the path. Typically, $f^+(s,\cdot) = \min_{1 \leq j \leq m} f_j^+(s,\cdot)$ ($f^-(s,\cdot) = \max_{1 \leq j \leq m} f_j^-(s,\cdot)$) is specified as the minimum (maximum) of m concave (convex) functions, where m denotes the number of constraints. In this case, $E = O(Cm\log m)$ [17], where C is the complexity of obtaining lower and upper bounds of sublevel sets of $\pm f_j^{\mp}(s,\cdot)$. Computing the optimal profile for tracking each individual landmark can be done in parallel, so that the complexity of the first stage of the method is $O(p^{-1}nNCm\log m)$, where p denotes the number of threads.

The second stage of the first method proceeds by selecting the k fastest out of n features, requiring $O(k\log n)$ time. Hence, the running time of the first method is $O(p^{-1}nNCm\log m) + O(k\log n)$.

The second stage of the second method iteratively selects from the set of unchosen features the one which incurs the smallest increase in execution time. Calculating the increase for a single feature has complexity $O(N)$, yielding a total of $O(nN)$ for $O(n)$ such features. Thus, selecting k features has running time $O(nkN)$, yielding a $O(p^{-1}nNCm\log m) + O(nkN)$-time algorithm.

The second stage of the third method has running time exponential in k. Although theoretically slower than the first two methods, and as of yet unsuitable for use in real time on embedded systems, it yields the ground truth optimum within minutes for problems of the size we consider.

V. Experiments

A. Experimental Setup

For simplicity we assume that the quadrotor used in the experiments is modeled 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. The navigation environment is generated using the FlightGoggles photorealistic simulator described in [25]. The chosen environment is a living room (see Figure 2) which would be a typical scenario for autonomous robots. For the experiments we generate 4 trajectory segments shown in Figure 3.

B. Algorithm Comparison

In this experiment, we consider the performance of all of the algorithms mentioned in Section III. The key performance indices measured in this experiment are the execution time and the computation time. We report these indices for a selection problem of selecting 50 features from an available 500 features in the environment. For all the timing experiments, the speed profiles of the features are pre-computed and the timing only includes the selection of the features. 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 [24] was allowed to use all available 40 threads. The results of this evaluation are summarized in Table I. As can be seen in the table, the execution times generated by the algorithms are nearly identical but the K-fastest algorithm is orders of magnitude faster to compute.

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
Convex optimization	2.2828	2.8669	258.4733	48.97485

TABLE I: Results of the algorithm comparison.

C. Feature Tracking experiment

For this experiment, the point cloud is generated using stereo matching and the pixels with valid depth values are then used for corner extraction using GoodFeaturesToTrack.

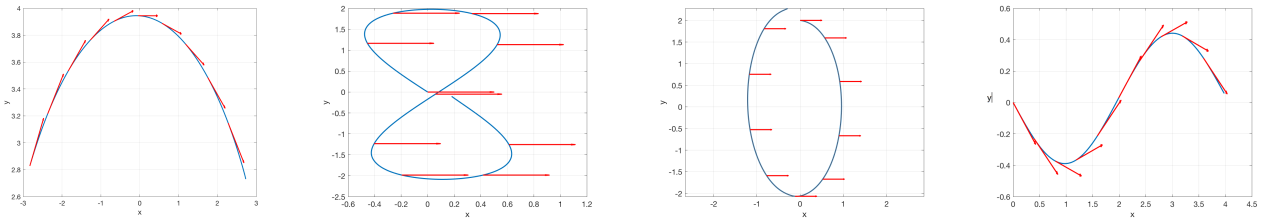
(a) Image captured at the start of the trajectory.

(b) Image captured at the end of the trajectory.

(c) Features chosen using the quality metric of GoodFeaturesToTrack.

(d) Features chosen by the K-Fastest method.

Fig. 2: The figures show the start and end frame of a simulation experiment. The K-Fastest method chooses features that are still visible at the end of the trajectory in comparison to features chosen using the quality from GoodFeaturestoTrack.



(a) The Arc trajectory.

(b) The Figure 8 trajectory segment.

(c) The oval trajectory segment.

(d) The slalom trajectory segment.

Fig. 3: Trajectories considered for the experiments.

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

TABLE II: This table shows the number of successfully tracked features over the trajectory with k=100.

Trajectory	Execution Time [s]			
	$\leq$ 288 pixels s^{-1}			$\leq$ 432 pixels s^{-1}
	$k = 50$	$k = 100$	$k = 150$	$k = 100$
Oval	4.342	4.342	4.342	4.194
Figure 8	4.634	4.634	4.634	4.634
Slalom	3.908	3.929	3.943	2.881

TABLE III: This table shows the variation in execution time among the different experiments performed.

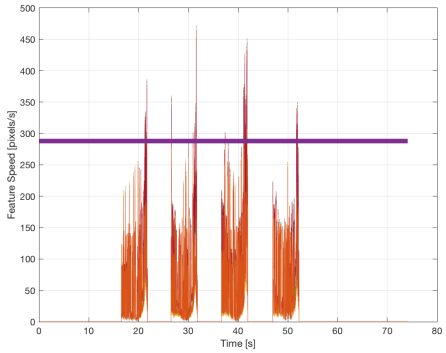
The extracted corners form the candidate point cloud for feature selection. In the first set of experiments, we evaluate the effect on feature tracking. For this experiment, the trajectory and features are specified by the K-Fastest algorithm. We compare the number of features tracked by a standard KLT tracker for the same number of features selected by the quality metric and the K-Fastest method. An example of the selected features is shown in Figure 2. The results are summarized in Table II. As can be seen in the table K-Fastest consistently tracks more features.

D. Robot experiments

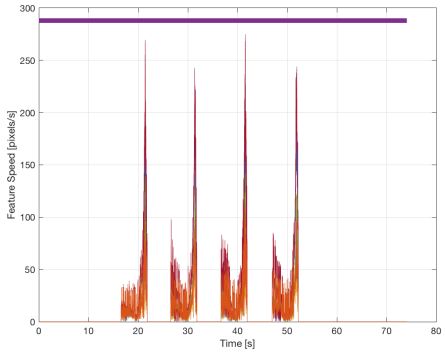
For each of the trajectories, we generate a speed profile for an allowed selection of 50, 100 and 150 landmarks in the environment with a maximum feature speed of 288 pixels s^{-1} and a selection of 100 with 432 pixels s^{-1}. In this experiment, feature selection is performed only using the K-Fastest algorithm. The trajectories are then flown using a custom built quadrotor executing a state of the art INDI controller described in [26] and the ground truth pose and reference command provided to the quadrotor are recorded. The ground truth pose of the quadrotor is then used to project the chosen landmarks back onto the image plane and the consecutive feature velocity along the trajectory segment is measured. Figure 4 shows a qualitative comparison of the feature speeds along the trajectory segment assuming both a gimbaled and non-gimbaled camera. In this case, the trajectory is executed 4 times and the periodic behavior can be observed in the plots. The quadrotor is successfully able to keep the feature speed under the desired threshold and a few features violate the threshold for short periods of time. This can be attributed to the imperfect nature of the trajectory tracking on the real quadrotor. The feature speeds are noticeably higher for the non-gimbaled case as expected but the optimization does compress the speed of the features. The summarization of the execution times for the 16 trajectories flown is shown in Table III. Table IV shows the mean of the maximum feature speed profiles over the trajectory segments.

VI. CONCLUSIONS

The main contribution of this paper is an efficient approximation algorithm for jointly selecting the desired number of features and recovering the time optimal trajectory that allows the agent to track them. We provide performance



(a) Feature speeds of all 100 selected features computed assuming a non gimbaled camera for the slalom trajectory segment with a threshold of 288 pixels s^{-1}



(b) Feature speeds of all 100 selected features assuming a gimbaled camera for the slalom trajectory segment with a threshold of 288 pixels s^{-1}

Fig. 4: Qualitative comparison of the feature speed profiles assuming both a gimbaled camera and non-gimbaled camera.

Trajectory	Feature speed [pixel s^{-1}]			
	$\leq$ 288 pixels s^{-1}			$\leq$ 432 pixels s^{-1}]
	$k = 50$	$k = 100$	$k = 150$	$k = 100$
Arc	23.0246	31.0976	29.5065	30.2752
Oval	98.9926	108.0196	99.8715	142.0394
Figure 8	76.9707	70.5656	79.7371	101.3292
Slalom	24.7777	43.6830	63.5860	31.7051

TABLE IV: This table shows the the mean of the maximum feature speed profile across all the experiments.

guarantees for the algorithm for a certain class of maneuvers, leaving for future work, the extension of the guarantee for more complicated trajectories. Ultimately, our empirical results show near optimal performance of the aforementioned algorithm as well as its demonstration on a robotic platform.

REFERENCES

[1] Jianbo Shi and Tomasi, "Good features to track," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 1994, pp. 593–600.

[2] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3886–3893.

[3] Y. Zhao and P. A. Vela, "Good feature selection for least squares pose optimization in vo/vslam," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1183–1189.

[4] B. Penin, R. Spica, P. R. Giordano, and F. Chaumette, "Vision-based minimum-time trajectory generation for a quadrotor UAV," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 6199–6206.

[5] M. Sheckells, G. Garimella, and M. Kobilarov, "Optimal Visual Servoing for differentially flat underactuated systems," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 5541–5548.

[6] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "PAMPC: Perception-Aware Model Predictive Control for Quadrotors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–8.

[7] V. Murali, I. Spasojevic, W. Guerra, and S. Karaman, "Perception-aware trajectory generation for aggressive quadrotor flight using differential flatness," in *2019 American Control Conference (ACC)*, July 2019, pp. 3936–3943.

[8] I. Spasojevic, V. Murali, and S. Karaman, "Perception-aware time optimal path parameterization for quadrotors," *arXiv preprint arXiv:2005.13986*, 2020.

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

[10] J. J. E. Slotine and H. S. Yang, "Improving the efficiency of time-optimal path-following algorithms," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 118–124, 1989.

[11] Z. Shiller and H.-H. 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*, vol. 114, 03 1992.

[12] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-Optimal Path Tracking for Robots: A Convex Optimization Approach," *IEEE Transactions on Automatic Control*, vol. 54, pp. 2318 – 2327, 11 2009.

[13] K. K. Hauser, "Fast interpolation and time-optimization with contact," *The International Journal of Robotics Research*, vol. 33, pp. 1231 – 1250, 2014.

[14] T. Lipp and S. Boyd, "Minimum-time speed optimisation over a fixed path," *International Journal of Control*, vol. 87, 02 2014.

[15] L. Consolini, M. Locatelli, A. Minari, and A. Piazzi, "An optimal complexity algorithm for minimum-time velocity planning," *Systems & Control Letters*, vol. 103, pp. 50–57, 05 2017.

[16] H. Pham and Q. C. Pham, "A New Approach to Time-Optimal Path Parameterization Based on Reachability Analysis," *IEEE Transactions on Robotics*, vol. 34, pp. 645 – 659, 06 2018.

[17] L. Consolini, M. Locatelli, A. Minari, Á. Nagy, and I. Vajk, "Optimal Time-Complexity Speed Planning for Robot Manipulators," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 790–797, 2019.

[18] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[19] H. Chitsaz and S. M. LaValle, "Time-optimal paths for a Dubins airplane," in *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 2379–2384.

[20] H. Nguyen and Q. C. Pham, "Time-Optimal Path Parameterization of Rigid-Body Motions: Applications to Spacecraft Reorientation," *Journal of Guidance, Control, and Dynamics*, vol. 39, pp. 1–5, 01 2016.

[21] F. Pfeiffer and R. Johanni, "A Concept for Manipulator Trajectory Planning," *IEEE Journal of Robotics and Automation*, vol. RA-3, pp. 115 – 123, 05 1987.

[22] I. Spasojevic, V. Murali, and S. Karaman, "Asymptotic Optimality of a Time Optimal Path Parametrization Algorithm," *IEEE Control Systems Letters*, vol. 3, pp. 835–840, 2019.

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

[24] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2020. [Online]. Available: http://www.gurobi.com

[25] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "Flightgoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality," *arXiv preprint arXiv:1905.11377*, 2019.

[26] E. Tal and S. Karaman, "Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness," in *2018 IEEE Conference on Decision and Control (CDC)*, Dec 2018, pp. 4282–4288.