



Attitude Trajectory Optimization for Agile Satellites in Autonomous Remote Sensing Constellations

Emmanuel Sin^{*}, Murat Arcaç[†]

University of California, Berkeley, California, 94720, U.S.A.

Sreeja Nag[‡], Vinay Ravindra[§], Alan Li[¶], Richard Levinson^{||}

NASA Ames Research Center, Moffett Field, California, 94035, U.S.A.

Agile attitude maneuvering maximizes the utility of remote sensing satellite constellations. By taking into account a satellite’s physical properties and its actuator specifications, we may leverage the full performance potential of the attitude control system to conduct agile remote sensing beyond conventional reorient-and-scan operation. Employing a constellation of these agile satellites, coordinated by an autonomous and responsive scheduler, can significantly increase overall response rate, revisit time and global coverage for the mission. In this paper, we describe how we use recent advances in sequential convex programming (SCP) based trajectory optimization to enable rapid-target acquisition, pointing and tracking capabilities for a scheduler-based constellation. We present two problem formulations – the first *Minimum-Time Optimal Control Problem* determines the minimum time, required energy, and optimal trajectory to slew between any two orientations given nonlinear quaternion kinematics, gyrostat and actuator dynamics, and state/input constraints. By gridding the space of 3D rotations and efficiently solving this problem on the grid, we may produce lookup tables or parametric fits offline that can be used by the scheduler to compute accurate estimates of minimum-time and maneuver energy online for real-time constellation scheduling. The estimates allow the optimization-based scheduler to produce target-remote-sensing and data-downlinking schedules that are dynamically feasible for each satellite and optimal for the constellation. The second *Minimum Attitude-Error-and-Control-Effort Optimal Control Problem* is used online by each satellite to produce continuous attitude-state and control-input trajectories that realize a given schedule while minimizing attitude error and control effort. The optimal trajectory may then be achieved by a low-level tracking controller. This onboard trajectory generation and tracking scheme is possible due to real-time, efficient SCP implementations. We demonstrate our approach with an example that uses simulation data for a reference satellite in Sun-synchronous orbit passing over globally-distributed, Earth-observation targets.

I. Nomenclature

J	=	mass moment of inertia matrix of rigid body in body-fixed frame, [kg· m ²]
J_r	=	mass moment of inertia of actuation rotor about its axis of rotation, [kg· m ²]
A_r	=	actuator Jacobian, each column representing a rotor’s axis of rotation w.r.t. body-fixed frame
\mathbf{q}	=	unit quaternion parameterizing attitude of rigid body’s body-fixed frame w.r.t. inertial frame, []
$\boldsymbol{\omega}$	=	angular velocity of rigid body about body-fixed frame axes, [rad/s]
\mathbf{r}_i	=	angular momentum of rotor i about its axis, [Nms]
$\boldsymbol{\tau}_i$	=	torque of rotor i about its rotor axis, [N]
t	=	time, [s]
τ	=	normalized time, []

^{*}Graduate Student, Department of Mechanical Engineering, emansin@berkeley.edu, AIAA Student Member.

[†]Professor, Department of Electrical Engineering and Computer Sciences, arcak@berkeley.edu.

[‡]Senior Research Scientist, Bay Area Environmental Research Institute, sreeja.nag@nasa.gov, AIAA Senior Member.

[§]Research Scientist, Bay Area Environmental Research Institute, vinay.ravindra@nasa.gov, AIAA Member.

[¶]Research Scientist, Bay Area Environmental Research Institute, alan.s.li@nasa.gov.

^{||}Senior Computer Scientist, KBR Wyle, richard.j.levinson@nasa.gov.

II. Introduction

Due to the proliferation of launch providers to low Earth orbit (LEO) and the trend towards smaller, cost-efficient spacecraft, satellite constellations are enabling scientific missions and commercial applications that are otherwise impossible with a single, larger satellite. For example, a constellation of satellites in LEO may be coordinated to point towards coastal regions around the world to measure ocean color, atmospheric properties, phytoplankton concentrations, and ultimately assess the health of global coral reef ecosystems [1]. LEO constellations may also be employed to measure episodic precipitation and subsequent water flow in flood-prone cities [2]-[3]. Furthermore, constellations may be tasked to measure soil moisture in targeted regions to assess the risk of wildfires [4]. In addition to climate and environment monitoring, Earth-observing constellations are providing commercial and economic value by measuring, for instance, agricultural crop growth, infrastructure development, or logistical activity at airports and shipping container routes [5]. Beyond these Earth observation applications, LEO satellite constellations are also enabling space-based Internet and telecommunication services [6].

To enhance performance as an interconnected system, the satellites of a constellation can be precisely coordinated throughout each stage of a mission. Once a group of satellites are deployed into a desired orbital plane by a launch vehicle, the satellites enter the *acquisition stage* where they must be phased relative to each other to achieve desired angular spacing. In [7], it is assumed that the orientation of the satellites with respect to their orbital velocities may be controlled so that either a minimum or maximum surface area is exposed to the atmosphere that exists in low Earth orbit. By inducing either a low or high atmospheric drag on each satellite using a bang-bang control approach, the resulting differential drag between satellites is used to separate them into an equally-spaced formation. In [8], simulated annealing is used to design time-optimal differential drag commands for a group of up to 100 satellites and the method is demonstrated on an actual constellation deployed in LEO. Assuming accurate attitude pointing, [9] formulates a linear program that produces differential drag commands taking on continuous values between the minimum and maximum values, allowing the constellation to form not only under a minimum-time objective but also with a maximum altitude (or equivalently, maximum constellation lifetime) objective. A distributed controller approach is presented in [10], where it is assumed that the attitude of the satellites can be controlled to apply continuous low-thrust in the appropriate direction. Regardless of where the control authority is derived from (e.g., differential drag commands or thrust commands), these constellation acquisition methods require accurate attitude pointing.

In the subsequent *operational stage*, the satellites must perform various scheduled tasks including targeted remote sensing (e.g., imaging, radiometry), downlinking/uplinking data to/from ground stations, orbital station-keeping and other maintenance activities. In [1]-[4], an automated scheduler is developed to run autonomously on either ground stations (with schedules uplinked to satellites) or onboard in a distributed approach. Based on dynamic programming or mixed integer programming, the scheduler produces imaging schedules for each satellite that maximizes the number of observations and/or observation times for the constellation as a whole. In addition to maximizing spatial imaging coverage, [5] also addresses the problem of maximizing data downlinked by the constellation. We note that the schedulers in these works make inherent assumptions on the agility and pointing performance of the attitude control subsystem on board each satellite. For instance, in order to produce a feasible schedule that provides sufficient time to slew between desired orientations, the scheduler must know the dynamically-feasible, minimum slew time between any two desired orientations. This minimum slew time depends on the physical properties of the spacecraft (i.e., mass moment of inertia, actuator configuration) and actuator constraints (e.g., maximum rotor torque and momentum). Once an optimized schedule is produced for the constellation, each satellite must then generate and track an attitude trajectory that realizes its given schedule. Such an attitude trajectory may be optimized to minimize desired-actual attitude error at specific times of the trajectory and minimize control effort over the course of the trajectory.

Predicting that future spacecraft will require agile attitude control systems that provide rapid multi-target acquisition, pointing, and tracking capabilities, [11] proposes a feedback regulator to conduct large-angle, rest-to-rest slew maneuvers using the Euler's eigenaxis rotation between any two orientations. Resembling feedback linearization, the proposed law introduces a nonlinear term to cancel out the coupling between body angular velocities and replaces it with linear error-quaternion and body rate feedback terms. In [12], a large class of attitude tracking control laws that have the general form of proportional-derivative (PD) feedback and feedforward compensation is obtained with proofs of their global asymptotic stability in the closed-loop. Minimum-time, rest-to-rest slew maneuvers for an inertially symmetric rigid spacecraft with independent three-axis controls are studied in [13], which shows that the optimal maneuver is not, in general, an eigenaxis rotation but one that includes significant nutation of the instantaneous axis of rotation. Furthermore, the structure of the optimal control is different for small and large reorientation angles. While [11]-[13] consider rigid body dynamics under ideal, body-fixed control torques, [14] considers actuator dynamics and presents a

feedback control logic that produces near minimum-time eigenaxis slew maneuvers under physical limits of actuators and sensors. We note that the attitude control strategy in [1] uses a minimum-time, eigenaxis slew maneuver and switches to a PD control law for small angles. Based on closed-loop simulations with this control strategy, a polynomial fit of minimum maneuver time as a function of the eigenaxis slew angle is used in a scheduler [2].

Although eigenaxis-based minimum-time control laws may be applied to general minimum-time problems to produce near-optimal maneuvers, they do not explicitly address nonlinearities that arise from, for instance, actuator dynamics nor do they explicitly consider general state and input constraints (e.g., bounds on momentum, power and energy). Moreover, when directly applying a feedback control strategy to track a desired sequence of discrete orientations, the feedback gains between each pair of orientations must be carefully tuned to achieve settling times that completely satisfy a desired pointing schedule, without missing any targets. In these minimum-time or minimum-attitude-error situations, we may use trajectory optimization methods to not only explicitly deal with nonlinearities and constraints, but to also automate the attitude trajectory generation process for autonomous execution.

As surveyed by Betts [15], there is an expanse of literature on various trajectory optimization methods that may be characterized by, for example, the solution approach (i.e., indirectly satisfying necessary conditions for optimality or directly solving a transcribed version of the problem), or the transcription process (shooting versus collocation). In [16], various direct collocation methods are introduced while [17] focuses on an indirect pseudospectral method that has been used in practice, for example, to control the International Space Station with a zero-propellant maneuver. Regardless of the approach, many trajectory optimization methods treat a problem in its original nonlinear, non-convex form, requiring the use of a nonlinear programming solver [18], which may be computationally inefficient with long solution times.

Recent advances in sequential convex programming (SCP) have enabled efficient computation of locally optimal trajectories for nonlinear systems with non-convex constraints and objectives. SCP is an iterative method that repeatedly formulates and solves a convex, finite-dimensional parameter optimization problem that approximates the original non-convex optimal control problem. A convex formulation is typically achieved by linearizing the nonlinear system around a nominal trajectory (i.e., the solution from the previous iteration) and approximating any non-convex constraints and objective with Taylor series expansions. Fast and reliable Interior Point Method algorithms [19] may be used to solve these convex subproblems. In the early works of [20] and [21], successive convexification, a specific implementation of the SCP method, is introduced to find minimum-fuel and minimum-time trajectories in a 6-DOF rocket landing problem. In [22]-[25], certain details of the implementation, including choice of discretization method, constraint satisfaction between temporal nodes, convexification of non-convex constraints, and algorithm convergence properties are explored. Finally, the works of [26]-[32] introduce state-triggered constraints and address real-time, onboard implementations that may produce solutions in a fraction of a second.

A. Main Contributions

The main contributions of this work are:

- 1) An offline method to produce accurate estimates of the minimum time and required energy for a reference satellite to maneuver between any two orientations. The estimates may be used by a scheduler to produce dynamically-feasible pointing schedules for each satellite of a constellation.
- 2) An online method to produce optimal attitude trajectories that satisfy desired target-pointing schedules with minimum error and control effort. Once a schedule is determined, this method may be run on-board to produce an optimal trajectory for a low-level tracking controller to then follow.

The first contribution allows a constellation scheduler to make informed scheduling decisions based on an accurate assessment of the spacecraft's maneuvering capabilities. The second contribution allows the spacecraft to realize scheduling decisions.

B. Organization of Paper

After brief comments on the notation, dynamical model, and attitude parameterization conventions assumed in this paper, we present two problem formulations with minimum-time and minimum-attitude-error-and-control-effort objectives. We describe the process of transcribing the continuous-time trajectory optimization problems into convex, finite-dimensional parameter optimization problems. We then review state-of-the-art techniques in sequential convex programming (SCP) that are used to approximately solve the original problems. Finally, we apply the problem formulations to numerical examples and discuss the solutions found using the SCP method.

C. Preliminaries

To study attitude motion, a satellite may be modeled as a gyrostat [33], consisting of the platform (i.e., spacecraft bus) and actuation rotors (e.g., momentum/reaction wheels). The state vector for this system of rigid bodies may be represented as:

$$\mathbf{x} = [\mathbf{q}^\top \ \boldsymbol{\omega}^\top \ \mathbf{r}^\top]^\top = [q_1 \ q_2 \ q_3 \ q_4 \ \omega_x \ \omega_y \ \omega_z \ r_1 \ r_2 \ r_3 \ r_4]^\top \quad (1)$$

where \mathbf{q} is the unit quaternion (i.e., $\|\mathbf{q}\|_2 = 1$) that describes the orientation of the spacecraft's body fixed frame with respect to an inertial frame, $\boldsymbol{\omega}$ is the angular velocity vector of the spacecraft with components expressed about the body-fixed axes, and \mathbf{r} represents the angular momentum of each spinning rotor about its axis of rotation. The input vector consists of the torques produced by each rotor about its axis:

$$\mathbf{u} = \boldsymbol{\tau} = [\tau_1 \ \tau_2 \ \tau_3 \ \tau_4]^\top \quad (2)$$

The differential equations describing the motion of this 11-state, 4-input system consist of the quaternion kinematics, gyrostat equation, and a single-integrator model for the rotors:

$$\dot{\mathbf{x}} := \frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}) := \begin{bmatrix} \frac{1}{2}\boldsymbol{\Omega}\mathbf{q} \\ -J^{-1}(\boldsymbol{\omega}^\times (J\boldsymbol{\omega} + A_r\mathbf{r}) + A_r\boldsymbol{\tau}) \\ \boldsymbol{\tau} \end{bmatrix} \quad (3)$$

The positive definite matrix J represents the mass moment of inertia of the spacecraft in the body-fixed frame, $A_r \in \mathbb{R}^{3 \times 4}$ is the actuator Jacobian where each column represents a rotor's axis of rotation with respect to the body-fixed frame, and $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}^\times$ are skew-symmetric matrices defined as:

$$\boldsymbol{\Omega} := \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \quad \boldsymbol{\omega}^\times := \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (4)$$

In this paper, we follow the unit quaternion (or Euler Symmetric Parameters) convention used in [34] and [35] for attitude parameterization, where the vector part is stacked on top of the scalar part:

$$\mathbf{q} := [\mathbf{q}_v^\top \ q_s]^\top := [q_1 \ q_2 \ q_3 \ q_4]^\top \quad (5)$$

Furthermore, we denote the quaternion conjugate and the identity quaternion, respectively, as:

$$\mathbf{q}^+ := [-\mathbf{q}_v^\top \ q_s]^\top = [-q_1 \ -q_2 \ -q_3 \ q_4]^\top \quad (6)$$

$$\mathbf{q}^I := [\mathbf{0}^\top \ 1]^\top = [0 \ 0 \ 0 \ 1]^\top \quad (7)$$

As defined in [35] and used in [11] and [14], we may measure the attitude error between a given quaternion \mathbf{q} and a desired quaternion $\bar{\mathbf{q}}$ by computing the error-quaternion:

$$\mathbf{q}^e = \begin{bmatrix} q_1^e \\ q_2^e \\ q_3^e \\ q_4^e \end{bmatrix} := \bar{\mathbf{q}}^+ \mathbf{q} = \begin{bmatrix} \bar{q}_4 & \bar{q}_3 & -\bar{q}_2 & -\bar{q}_1 \\ -\bar{q}_3 & \bar{q}_4 & \bar{q}_1 & -\bar{q}_2 \\ \bar{q}_2 & -\bar{q}_1 & \bar{q}_4 & -\bar{q}_3 \\ \bar{q}_1 & \bar{q}_2 & \bar{q}_3 & \bar{q}_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (8)$$

We note that the Hamilton product (or any quaternion multiplication operation) between two quaternions is associative and distributive, but not commutative in general, i.e., $\bar{\mathbf{q}}^+ \mathbf{q} \neq \mathbf{q} \bar{\mathbf{q}}^+$.

III. Problem Formulations

In this section we introduce two optimal control problem (OCP) formulations to be solved with sequential convex programming and applied to numerical examples involving rotor-actuated satellites under agile-pointing operations. As stated, these basic formulations have the same constraints and differ only in the objective and decision variables. We discuss how the objectives and constraints may be modified depending on the application.

A. Minimum Time

The following *Minimum-Time OCP* addresses the time-optimal, large-angle, rest-to-rest slew problem. The integral term in (9) captures the minimum-time objective and we note that the decision variables consist of the gyrostat's state and input as well as the final time t_f . The gyrostat equation (11) may be modified to include other actuators, such as magnetorquers and propulsive thrusters. We may also include modeled environmental disturbances, including moments due to gravity gradient, atmospheric drag, solar radiation pressure, and the magnetic field of Earth. The actuator dynamics in (12) may use higher-fidelity models that consider, for example, brushless DC motor dynamics and rotational friction. Maximum rotor momentum and maximum rotor torque bounds are enforced by (13) and (14), respectively. Finally, the initial and terminal conditions are given in (15) and (16).

$$\underset{\mathbf{q}, \omega, \mathbf{r}, \tau, t_f}{\text{minimize}} \quad \int_{t_0}^{t_f} 1 dt \quad (9)$$

$$\text{s.t. } \dot{\mathbf{q}}(t) = \frac{1}{2} \Omega(t) \mathbf{q}(t) \quad \forall t \in [t_0, t_f] \quad (10)$$

$$\dot{\omega}(t) = -J^{-1}(\omega^\times(t) (J\omega(t) + A_r \mathbf{r}(t)) + A_r \tau(t)) \quad \forall t \in [t_0, t_f] \quad (11)$$

$$\dot{\mathbf{r}}(t) = \tau(t) \quad \forall t \in [t_0, t_f] \quad (12)$$

$$|\mathbf{r}_i(t)| < r_{max} \quad \forall i = 1, \dots, 4 \quad \forall t \in [t_0, t_f] \quad (13)$$

$$|\tau_i(t)| < \tau_{max} \quad \forall i = 1, \dots, 4 \quad \forall t \in [t_0, t_f] \quad (14)$$

$$\mathbf{q}(t_0) = \bar{\mathbf{q}}_0, \quad \omega(t_0) = \mathbf{0} \quad (15)$$

$$\mathbf{q}(t_f) = \bar{\mathbf{q}}_f, \quad \omega(t_f) = \mathbf{0} \quad (16)$$

B. Minimum Attitude Error and Control Effort

In the following *Minimum-Attitude-Error-and-Control-Effort OCP*, we maintain the same constraints as the problem above but change the objective. Furthermore, the final time t_f is no longer a decision variable but a fixed parameter. Given a schedule (i.e., sequence) of desired quaternions and angular velocities at specified times, $\{t_k, \bar{\mathbf{q}}(t_k), \bar{\omega}(t_k)\} \forall k \in \mathbb{K}$, where \mathbb{K} is a finite set of indices, the objective in (17) minimizes the error at those discrete time points while also minimizing the continuous control effort. The user-defined parameter $\gamma > 0$ weighs the angular velocity error relative to the quaternion error while $\rho > 0$ weighs the control penalty term relative to the total attitude error.

$$\underset{\mathbf{q}, \omega, \mathbf{r}, \tau}{\text{minimize}} \quad \sum_{k \in \mathbb{K}} \{ \|\bar{\mathbf{q}}^+(t_k) \mathbf{q}(t_k) - \mathbf{q}^1\|_2 + \gamma \|\omega(t_k) - \bar{\omega}(t_k)\|_2 \} + \rho \int_{t_0}^{t_f} \|\tau(t)\|_2 dt \quad (17)$$

$$\text{s.t. } \dot{\mathbf{q}}(t) = \frac{1}{2} \Omega(t) \mathbf{q}(t) \quad \forall t \in [t_0, t_f] \quad (18)$$

$$\dot{\omega}(t) = -J^{-1}(\omega^\times(t) (J\omega(t) + A_r \mathbf{r}(t)) + A_r \tau(t)) \quad \forall t \in [t_0, t_f] \quad (19)$$

$$\dot{\mathbf{r}}(t) = \tau(t) \quad \forall t \in [t_0, t_f] \quad (20)$$

$$|\mathbf{r}_i(t)| < r_{max} \quad \forall i = 1, \dots, 4 \quad \forall t \in [t_0, t_f] \quad (21)$$

$$|\tau_i(t)| < \tau_{max} \quad \forall i = 1, \dots, 4 \quad \forall t \in [t_0, t_f] \quad (22)$$

$$\mathbf{q}(t_0) = \bar{\mathbf{q}}_0, \quad \omega(t_0) = \mathbf{0} \quad (23)$$

$$\mathbf{q}(t_f) = \bar{\mathbf{q}}_f, \quad \omega(t_f) = \mathbf{0} \quad (24)$$

We note that an alternative ‘‘constraint formulation’’ of the *Minimum-Attitude-Error-and-Control-Effort OCP* is to remove the attitude penalty terms in the objective and implement them as constraints:

$$\|\bar{\mathbf{q}}^+(t_k)\mathbf{q}(t_k) - \mathbf{q}^l\|_2 \leq \epsilon_q \quad \forall k \in \mathbf{K} \quad (25)$$

$$\|\boldsymbol{\omega}(t_k) - \bar{\boldsymbol{\omega}}(t_k)\|_2 \leq \epsilon_\omega \quad \forall k \in \mathbf{K} \quad (26)$$

where $\epsilon_q \geq 0$ and $\epsilon_\omega \geq 0$ are user-defined error tolerances. However, if the error tolerance values are set too tight for a given attitude schedule, then the problem may be infeasible. Hence, when specific error tolerance values are not required, we may choose to solve the original ‘‘penalty formulation’’ of the problem.

Additional constraints that we may include in our problem formulations are bounds on the maximum instantaneous power drawn P_{max} and maximum energy consumed E_{max} by the attitude control system (i.e., all four rotors combined):

$$\sum_{i=1}^4 \left| \tau_i(t) \cdot \frac{1}{J_r} r_i(t) \right| < P_{max} \quad \forall t \in [t_0, t_f] \quad (27)$$

$$\int_{t_0}^{t_f} \left\{ \sum_{i=1}^4 \left| \tau_i(t) \cdot \frac{1}{J_r} r_i(t) \right| \right\} dt < E_{max} \quad (28)$$

where J_r is the rotor inertia. Apart from the equations of motion, we note that the instantaneous power and energy constraints are non-convex due to being bilinear in the decision variables of $\boldsymbol{\tau}$ and \mathbf{r} . In the following section we discuss how such non-convex constraints can be approximated as convex constraints.

IV. Trajectory Optimization

In this section, we first describe how a general optimal control problem (OCP) is transcribed into a convex, finite-dimensional parameter optimization problem (OPT). We then review the sequential convex programming method used in this paper. Consider the following continuous-time dynamical system:

$$\dot{\mathbf{x}}(t) := \frac{d}{dt} \mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad \forall t \in [t_0, t_f] \quad (29)$$

defined over the given time span, where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the state of the system and $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is the input to the system. As recommended in [17], [18], [32], and [36], it is important to scale a dynamical system before numerical trajectory optimization. Let us assume that our system (29) is in scaled form.

A. Time Normalization

In [21], a procedure is introduced to transform a free-final-time optimal control problem into a finite-dimensional parameter optimization problem by normalizing time in the dynamical system (29) by a normalization factor t_f :

$$\tau := \frac{t}{t_f} \quad (30)$$

We treat t_f as a decision variable in a free-final-time problem (e.g., minimum-time OCP) and as a constant parameter in a fixed-final-time problem (e.g., minimum-attitude-error-and-control-effort OCP). We now express the time span of the dynamical system in terms of the normalized time: $\frac{t_0}{t_f} \leq \tau \leq 1$. Since $t = t_f \tau \Rightarrow dt = t_f d\tau \Rightarrow \frac{dt}{d\tau} = t_f$, the derivative of the scaled state with respect to normalized time is:

$$\mathbf{x}'(t) := \frac{d}{d\tau} \mathbf{x}(t) = \frac{dt}{d\tau} \frac{d}{dt} \mathbf{x}(t) = t_f \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad \forall t \in [t_0, t_f] \quad (31)$$

Since t can be expressed as a function of τ and assuming $t_0 = 0$, we can represent (31) in terms of normalized time:

$$\mathbf{x}'(t(\tau)) = \mathbf{x}'(\tau) = t_f \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau)) =: \mathbf{F}(\mathbf{x}(\tau), \mathbf{u}(\tau), t_f) \quad \forall t \in [0, 1] \quad (32)$$

In this final expression, it is clear that the dynamical system is a function of state \mathbf{x} , input \mathbf{u} , and final time t_f .

B. Linearization

Assuming the dynamical system described by (32) is non-convex but differentiable, it can be approximated as convex in \mathbf{x} , \mathbf{u} , and t_f with a first-order Taylor expansion about a given trajectory:

$$\mathbf{x}'(\tau) = \mathbf{F}(\mathbf{x}(\tau), \mathbf{u}(\tau), t_f) \approx A(\tau)\mathbf{x}(\tau) + B(\tau)\mathbf{u}(\tau) + \Sigma(\tau)t_f + \mathbf{e}(\tau) \quad \forall t \in [0, 1] \quad (33)$$

where we denote the first-order partial derivative matrices of $\mathbf{F}(\mathbf{x}(\tau), \mathbf{u}(\tau), t_f)$ evaluated about $\{\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau), \bar{t}_f\}$ as

$$A(\tau) := D_{\mathbf{x}}\mathbf{F}(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau), \bar{t}_f) \quad (34)$$

$$B(\tau) := D_{\mathbf{u}}\mathbf{F}(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau), \bar{t}_f) \quad (35)$$

$$\Sigma(\tau) := D_{t_f}\mathbf{F}(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau), \bar{t}_f) = \mathbf{f}(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau)) \quad (36)$$

with the following dynamical approximation offset term:

$$\mathbf{e}(\tau) := -(A(\tau)\bar{\mathbf{x}}(\tau) + B(\tau)\bar{\mathbf{u}}(\tau)) \quad (37)$$

C. Discretization

As described in [37], the process of converting an optimal control problem into a parameter optimization problem begins by dividing the time duration of the optimal control problem into intervals using K nodes. The nodes may be chosen to be equally spaced, creating $K - 1$ equally-sized time intervals:

$$0 =: \tau_1 < \tau_2 < \dots < \tau_k < \dots < \tau_{K-1} < \tau_K := 1 \quad (38)$$

We refer to state, input, and offset terms at each node with shorthand notation:

$$\mathbf{x}_k := \mathbf{x}(\tau_k), \quad \mathbf{u}_k := \mathbf{u}(\tau_k), \quad \mathbf{e}_k := \mathbf{e}(\tau_k) \quad \forall k = 1, \dots, K \quad (39)$$

For use in the following section, we collectively refer to the decision variables that we have influence over as:

$$\mathbf{z}_k := [\mathbf{x}_k^\top, \mathbf{u}_k^\top, t_f]^\top \quad \forall k = 1, \dots, K \quad (40)$$

We use the First-Order Hold (FOH)-interpolation based discretization method in our implementation. As demonstrated in [23], FOH discretization provides the fastest computational time and achieves similar accuracy when compared to more advanced pseudospectral methods. Furthermore, it was shown that if convex input constraints are satisfied at the nodes, then inter-nodal convex input constraint satisfaction is also guaranteed. Reviewing the treatment shown in [20]-[31], the FOH interpolation represents the input within each of the $K - 1$ intervals as:

$$\mathbf{u}(\tau) = \lambda_k^- \mathbf{u}_k + \lambda_k^+ \mathbf{u}_{k+1} \quad \forall \tau \in [\tau_k, \tau_{k+1}], \quad k = 1, \dots, K - 1 \quad (41)$$

where

$$\lambda_k^- := \frac{\tau_{k+1} - \tau}{\tau_{k+1} - \tau_k}, \quad \lambda_k^+ := \frac{\tau - \tau_k}{\tau_{k+1} - \tau_k} \quad (42)$$

The exact discretization of (33) is then:

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k^- \mathbf{u}_k + B_k^+ \mathbf{u}_{k+1} + \Sigma_k t_f + \mathbf{e}_k \quad \forall k = 1, \dots, K - 1 \quad (43)$$

where

$$A_k := \Phi(\tau_{k+1}, \tau_k) \quad (44)$$

$$B_k^- := A_k \int_{\tau_k}^{\tau_{k+1}} \Phi^{-1}(\tau, \tau_k) \lambda_k^-(\tau) B(\tau) d\tau \quad (45)$$

$$B_k^+ := A_k \int_{\tau_k}^{\tau_{k+1}} \Phi^{-1}(\tau, \tau_k) \lambda_k^+(\tau) B(\tau) d\tau \quad (46)$$

$$\Sigma_k := A_k \int_{\tau_k}^{\tau_{k+1}} \Phi^{-1}(\tau, \tau_k) \Sigma(\tau) d\tau \quad (47)$$

$$\mathbf{e}_k := A_k \int_{\tau_k}^{\tau_{k+1}} \Phi^{-1}(\tau, \tau_k) \mathbf{w}(\tau) d\tau \quad (48)$$

The state transition matrix satisfies the following differential equation and initial condition within each interval:

$$\frac{d}{d\tau}\Phi(\tau, \tau_k) = A(\tau)\Phi(\tau, \tau_k), \quad \Phi(\tau_k, \tau_k) = I^{n_x} \quad (49)$$

In practice, the integrands of (45)-(48) along with (49) and (32) are numerically integrated from the start to the end of each interval using a nominal trajectory $\{\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k, \bar{t}_f, \bar{\mathbf{e}}_k\}$ for $k = 1, \dots, K-1$. Note that since we initialize the numerical integration for each interval with points from a nominal trajectory, rather than the terminal points found by integration in the previous temporal interval, this approach resembles a multiple-shooting discretization method, which has shown to improve convergence of the SCP algorithm by keeping solutions closer to the nominal trajectory. In contrast, a single shooting method allows approximation errors to grow in later temporal intervals.

D. Sequential Convex Programming Method

For the example in this paper, we use the Penalized Trust Region (PTR) variant of SCP described in [32]. A key difference with Successive Convexification (SCvx) studied in [25] is that PTR treats trust regions as soft constraints placed in the objective whereas SCvx enforces hard trust region constraints that are updated based on a rule. An advantage of SCvx is that convergence of this method is guaranteed. However, the method employs slack variables that may cause the approximately solved problem to be far from the original problem if they take on non-zero values in the solution. Hence, a converged solution may be infeasible for the original problem. In the following subsections we describe the PTR implementation of virtual controls and trust regions, and an approach to constraint convexification.

1. Virtual Controls

While executing sequential convex programming on a trajectory optimization problem, the approximated convex problem may become infeasible. This *artificial infeasibility* [21] is frequently encountered in the early iterations of the algorithm when the dynamics are linearized about a poor initial guess. To alleviate this issue, slack variables called *virtual controls* are added to the discrete-time equations of motion: (43)

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k^- \mathbf{u}_k + B_k^+ \mathbf{u}_{k+1} + \Sigma_k t_f + \mathbf{e}_k + \mathbf{v}_k \quad \forall k = 1, \dots, K-1 \quad (50)$$

These virtual controls act as dynamic relaxation terms that take on nonzero values when necessary to prevent dynamic infeasibility. In turn, use of these slack variables is heavily penalized with a term in the objective:

$$J_{vc} = w_{vc} \sum_{k=1}^K \|\mathbf{v}_k\|_1 \quad (51)$$

where w_{vc} is a large positive weight. Minimization of the 1-norm term encourages sparsity in the virtual control vector.

2. Trust Regions

To ensure that the solver does not stray too far from a nominal trajectory where the linearized model becomes less accurate, we implement a trust region cost term. In PTR, the deviation of decision variables from the solution of the previous iteration is penalized with the 2-norm of weighted deviations:

$$J_{tr} = \sum_{k=1}^K \|W_{tr}(\mathbf{z}_k - \bar{\mathbf{z}}_k)\|_2 \quad (52)$$

Assuming the system has been scaled, the weight matrix W_{tr} may be designed as a diagonal matrix with $w_{tr} > 0$:

$$W_{tr} := w_{tr} \cdot I^{K(n_x+n_u+1)} \quad (53)$$

3. Constraint Convexification

Let us consider a general (non-convex) constraint on the continuous-time variables:

$$\mathbf{g}(\mathbf{x}(\tau), \mathbf{u}(\tau), t_f) \leq 0 \quad \forall \tau \in [0, 1] \quad (54)$$

This constraint can be approximated by linearizing it about a nominal trajectory:

$$\tilde{\mathbf{g}}(\mathbf{x}(\tau), \mathbf{u}(\tau), t_f) := \mathbf{g}(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau), \bar{t}_f) + D\mathbf{g}(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau), \bar{t}_f) \begin{pmatrix} \mathbf{x}(\tau) \\ \mathbf{u}(\tau) \\ t_f \end{pmatrix} - \begin{pmatrix} \bar{\mathbf{x}}(\tau) \\ \bar{\mathbf{u}}(\tau) \\ \bar{t}_f \end{pmatrix} \leq 0 \quad \forall \tau \in [0, 1] \quad (55)$$

We further approximate the constraint by explicitly enforcing it only at the time nodes of our discretization:

$$\tilde{\mathbf{g}}(\mathbf{x}_k, \mathbf{u}_k, t_f) = \tilde{\mathbf{g}}(\mathbf{z}_k) \leq 0 \quad \forall k = 1, \dots, K \quad (56)$$

Inter-nodal constraint satisfaction is not guaranteed. However, the constraints at the nodes can be carefully designed so that constraints are enforced for all time [22]. In a similar fashion, any non-convex cost terms may also be approximated with a Taylor series expansion about a nominal trajectory.

We may also approximate constraints involving integral terms. For example:

$$\int_{t_0}^{t_f} h(\mathbf{x}(t), \mathbf{u}(t)) dt \leq 0 \implies t_f \int_0^1 h(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau \leq 0 \implies \frac{1}{K-1} \sum_{k=1}^K t_f h(\mathbf{x}_k, \mathbf{u}_k) \leq 0 \quad (57)$$

where the nonlinear term to be summed in the last inequality may also be linearized about the nominal trajectory.

4. Transcribed Convex Program

The *Minimum-Time OCP*, described by equations (9) - (16), may be transcribed into the following convex, finite-dimensional parameter optimization problem that we call *Minimum-Time OPT*:

$$\begin{aligned} & \underset{\{\mathbf{z}_k, \mathbf{v}_k\}_{k=1}^K}{\text{minimize}} && t_f + w_{vc} \sum_{k=1}^K \|\mathbf{v}_k\|_1 + \sum_{k=1}^K \|W_{tr}(\mathbf{z}_k - \bar{\mathbf{z}}_k)\|_2 \\ & \text{s.t.} && \end{aligned} \quad (58)$$

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k^- \mathbf{u}_k + B_k^+ \mathbf{u}_{k+1} + \Sigma_k T + \mathbf{w}_k + \mathbf{v}_k \quad \forall k = 1, \dots, K-1 \quad (59)$$

$$|\mathbf{x}_k| < \mathbf{x}_{max} \quad \forall k = 1, \dots, K \quad (60)$$

$$|\mathbf{u}_k| < \mathbf{u}_{max} \quad \forall k = 1, \dots, K \quad (61)$$

$$\mathbf{x}_1 = \bar{\mathbf{x}}_0 \quad (62)$$

$$\mathbf{x}_K = \bar{\mathbf{x}}_f \quad (63)$$

A similar process is conducted to formulate *Minimum-Attitude-Error-and-Control-Effort OPT*:

$$\begin{aligned} & \underset{\{\mathbf{z}_k, \mathbf{v}_k\}_{k=1}^K}{\text{minimize}} && \sum_{k \in \mathbb{K}} \{ \|\bar{\mathbf{q}}_k^+ \mathbf{q}_k - \mathbf{q}^i\|_2 + \gamma \|\omega_k - \bar{\omega}_k\|_2 \} + \rho \sum_{k=1}^K \|\mathbf{u}_k\|_2 + w_{vc} \sum_{k=1}^K \|\mathbf{v}_k\|_1 + \sum_{k=1}^K \|W_{tr}(\mathbf{z}_k - \bar{\mathbf{z}}_k)\|_2 \\ & \text{s.t.} && \end{aligned} \quad (64)$$

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k^- \mathbf{u}_k + B_k^+ \mathbf{u}_{k+1} + \Sigma_k T + \mathbf{w}_k + \mathbf{v}_k \quad \forall k = 1, \dots, K-1 \quad (65)$$

$$|\mathbf{x}_k| < \mathbf{x}_{max} \quad \forall k = 1, \dots, K \quad (66)$$

$$|\mathbf{u}_k| < \mathbf{u}_{max} \quad \forall k = 1, \dots, K \quad (67)$$

$$\mathbf{x}_1 = \bar{\mathbf{x}}_0 \quad (68)$$

$$\mathbf{x}_K = \bar{\mathbf{x}}_f \quad (69)$$

5. Algorithm

The sequential convex programming algorithm used in this paper is listed in Algorithm 1 where \mathbf{S}^0 is an initial guess at the solution. As a notation convention used in the algorithm, superscript i refers to the solution at the i^{th} iteration of

the algorithm:

$$\mathbf{S}^i := \begin{cases} \{\bar{\mathbf{x}}_1^i, \dots, \bar{\mathbf{x}}_k^i, \dots, \bar{\mathbf{x}}_K^i\}, \\ \{\bar{\mathbf{u}}_1^i, \dots, \bar{\mathbf{u}}_k^i, \dots, \bar{\mathbf{u}}_K^i\}, \\ \bar{t}_f^i \end{cases} \quad (70)$$

The algorithm stops when either (1) the user-defined maximum number of SCP iterations N_{SCP} have been executed, or (2) the algorithm has converged on a solution, where we define the convergence as the satisfaction of two conditions:

$$(J_{vc} \leq \epsilon_{vc}) \wedge (J_{tr} \leq \epsilon_{tr}) \quad (71)$$

where ϵ_{vc} and ϵ_{tr} are user-defined convergence tolerances. The first condition ensures that a negligible amount of virtual controls is used, indicating that the converged solution is dynamically feasible. The second condition ensures that the solution remains sufficiently close to the nominal trajectory upon which the trajectory optimization problem was formulated. Using a convergence flag, we keep track of whether the SCP subroutine has converged or not.

Algorithm 1: Sequential Convex Programming

Input : \mathbf{S}^0
Output \mathbf{S}^i , flag
:
for $i = 1 : N_{SCP}$ **do**
 formulate $\mathbf{OPT}^i(\mathbf{S}^{i-1})$
 $\mathbf{S}^i \leftarrow \mathbf{OPT}^i(\mathbf{S}^{i-1})$
 if \mathbf{S}^i converged **then**
 flag = 1
 return
 if $i = N_{SCP}$ **then**
 flag = 0
 return

V. Numerical Example

We demonstrate how our attitude trajectory optimization approach may be used to both inform a constellation scheduler as well as execute a given target-pointing schedule. We model our reference satellite using physical parameters listed in Table (1), representative of Planet's Skysat, a satellite capable of agile maneuvering and imaging with sub-meter (50cm GSD) resolution [38].

Parameter	Value	Units	Description	First Mention
m	110	[kg]	Satellite mass	
$l \times w \times h$	$60 \times 60 \times 95$	[cm]	Satellite dimensions (cuboid)	
J	$\begin{bmatrix} 8.5 & 0.0 & 0.0 \\ 0.0 & 8.5 & 0.0 \\ 0.0 & 0.0 & 6.0 \end{bmatrix}$	[kg m ²]	Satellite inertia matrix	Eqn (11)
r_{max}	0.80	[Nms]	Maximum rotor momentum	Eqn (13)
τ_{max}	0.06	[Nm]	Maximum rotor torque	Eqn (14)
A_r	$\begin{bmatrix} -0.68 & 0.68 & 0.68 & -0.68 \\ -0.68 & -0.68 & 0.68 & 0.68 \\ 0.26 & 0.26 & 0.26 & 0.26 \end{bmatrix}$	[]	Actuator Jacobian (4 rotors)	Eqn (11)

Table 1 Reference satellite parameters

A. Application of Minimum-Time OCP

In determining a target-remote-sensing or data-downlinking schedule for a particular satellite of a constellation, a scheduler must consider the time it takes for the satellite to slew from one orientation to another. Given the satellite's orbital motion, inertia and actuator constraints, if the relative distance between two scheduled targets on the Earth's

surface is large, it may not be feasible for the satellite to slew from one pointing orientation to the other in sufficient time. To our knowledge, a closed-form expression for computing minimum slew times between any arbitrary orientations does not exist (even for a symmetric rigid body with independent three-axis control).

Our approach is to apply the *Minimum-Time OCP*, described by equations (9) - (16), over a gridded space of 3D rotations. Since any attitude trajectory between any two orientations can be summarized with a single rotation about some axis, we parameterize the rotations using the Euler axis-angle representation. As illustrated in Fig. (1), we consider 100 equidistributed axes of rotation and 88 rotations of increasing magnitude: $\theta \in \{-180, -175, \dots, -10, -9, -1, 1, 2, \dots, 10, 15, \dots, 180\}$ to cover a relatively fine grid of the entire space of 3D rotations.

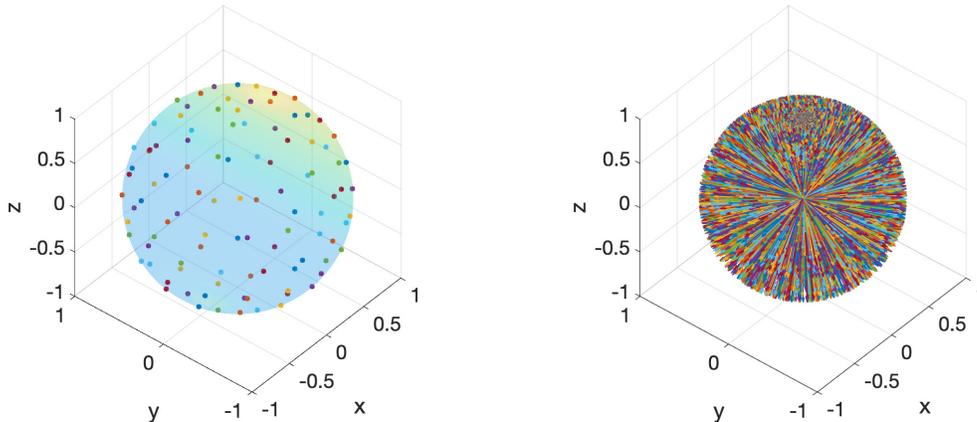


Fig. 1 (Left) 100 equidistributed points on unit sphere, (Right) 8,800 rotations of a pointing vector.

We solve 8,800 instances of *Minimum-Time OCP* where we set the initial value of the quaternion in (15) to be the identity quaternion (i.e., $\mathbf{q}_0 = \mathbf{q}^I$) and the final desired value in (16) to equal the quaternion parameterization of an Euler axis-angle rotation. With the sequential convex programming method, we efficiently solve each problem instance offline with a computation time on the order of seconds. Real-time SCP implementations [32] may solve each problem in milliseconds and the overall procedure can be parallelized for even faster offline computation. With the resulting minimum time and required energy values we can produce lookup tables or data-fitted functions for use by a scheduler.

B. Application of Minimum-Attitude-Error-and-Control-Effort OCP

Given a desired attitude schedule $\{t_k, \bar{\mathbf{q}}(t_k), \bar{\omega}(t_k)\} \forall k \in \mathbb{K}$, we may produce a continuous trajectory that passes through each attitude point in the sequence by using *Minimum-Attitude-Error-and-Control-Effort OCP*, equations (17) - (24). To produce an example attitude schedule to work with, we consider a remote-sensing application described in [2], where a 24-satellite, 3-plane Walker-Delta constellation is simulated at 710 km altitude, 98.5 deg inclined, circular orbits over a 6-hour duration with the orbital mechanics module of the D-SHIELD software suite [4]. Simulation results include the orbital states of the satellites at each time step as well as *access times* when user-defined target *grid points* (described by latitude and longitude coordinates) are not occluded by the Earth's surface and are observable by a satellite. In our example we consider 42 urban regions that experience frequent episodic precipitation and are prone to flooding. Each of these globally-distributed *watersheds* covers an 80 km² area spanned by 121 grid points.

For demonstration, we focus on a single satellite of the constellation and note that it can observe 16 of the 42 watersheds during the course of the 6-hour simulation. As shown in the first plot of Fig. (2), the satellite's groundtrack passes close to the 16 regions, each of which contains 121 green grid points. The second plot of Fig. (2) shows a cluster of 6 regions with a total of 726 potentially observable target grid points. We note that not all grid points in a region may be accessible due to access restrictions or bounds on maximum off-pointing angles. As the satellite passes over this cluster, the satellite may be commanded by the scheduler to perform a rapid sequence of agile slewing maneuvers to acquire, point, and track desired target grid points. The commanded schedule in this part of the orbital trajectory may be the most difficult to execute by an attitude control system, requiring high-frequency intra-region slewing as well as large-angle, inter-region slewing. The numerical results of this paper focus on attempting to observe this high-activity, 6-region cluster, the most challenging section of the 6-hour trajectory.

Figure (3) shows both Earth-Centered Inertial (ECI) and Earth-Centered Earth-Fixed (ECEF) views of the satellite's orbit where the red vectors point from the satellite's center-of-mass to each of the green accessible grid points. At any

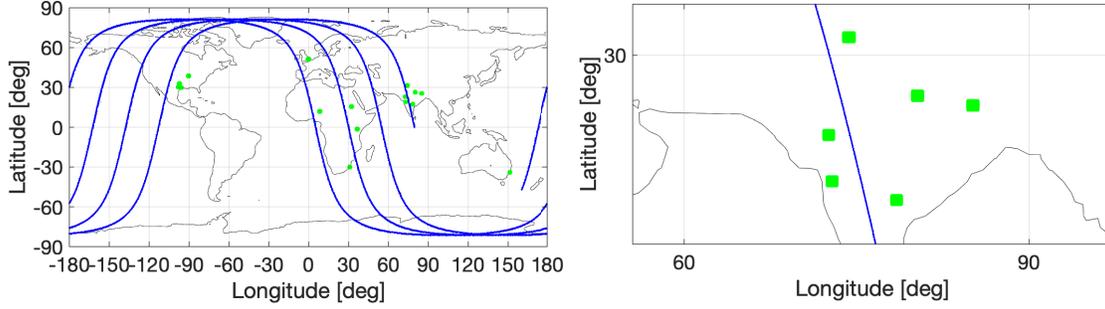


Fig. 2 (Left) Satellite groundtrack and 16 observable watersheds, (Right) Cluster of 6 watersheds

given time, a desired *pointing frame* may be uniquely defined with its z-axis aligned with a pointing vector, y-axis aligned with the negative orbit normal, and x-axis toward the satellite’s velocity vector. The orientation of this frame with respect to the ECI frame is the desired attitude to acquire. When a desired attitude is defined at a particular time, we may also set the angular velocity to equal zero to mitigate motion blur while imaging, i.e., if $\exists \bar{\mathbf{q}}(t_k)$, then $\bar{\boldsymbol{\omega}}(t_k) = \mathbf{0}$. Note that we may also customize the duration of zero angular velocity to accommodate measurement instruments with different observational requirements, e.g., if $\exists \bar{\mathbf{q}}(t_k)$, then $\{\bar{\boldsymbol{\omega}}(t_{k-1}), \bar{\boldsymbol{\omega}}(t_k), \bar{\boldsymbol{\omega}}(t_{k+1})\} = \mathbf{0}$.

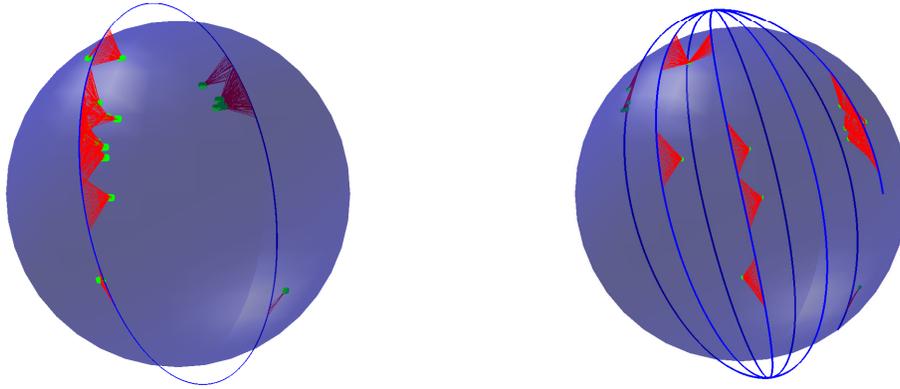


Fig. 3 (Left) Earth-Centered Inertial, (Right) Earth-Centered Earth-Fixed coordinate systems

The trajectory optimization results in the subsequent section are based on an example schedule designed by hand with the following assumptions:

- Attempt to observe all 726 grid points across the cluster of 6 regions.
- Finish observing all grid points in a region before moving to the next region (i.e., cannot slew back-and-forth between any two regions).
- Observe all grid points in a region using a “whisk-broom” scanning pattern, typically implemented by articulating onboard mirrors, similar to early Landsat satellites [39].
- The first accessible grid points in a region are observed first, deciding the direction of the intra-region, whisk-broom slewing pattern.
- The first accessible regions are observed first, deciding the direction of the inter-region slewing.

Compared to this heuristics-based schedule, we expect that an optimization-based schedule should result in a lower objective value for the *Minimum-Attitude-Error-and-Control-Effort OCP*. Future work will compare such optimization-based schedules to the baseline schedule used in this paper.

VI. Results

In this section we summarize the numerical example results produced by trajectory optimization with sequential convex programming (SCP). For every problem instance, we provide a crude initial guess that consists of a spherical linear

interpolated (SLERP, [40]) trajectory between desired quaternions, zero body angular velocity, zero rotor momentum, and zero rotor torque. We recall that our reference satellite is modeled after Planet’s Skysat with representative physical parameters listed in Table (1).

A. Application of Minimum-Time OCP

To demonstrate the application of *Minimum-Time OCP*, we consider the example shown in Fig. (4), where we find the minimum-time maneuver to conduct a 60 degree rotation about some arbitrary axis. The blue vector represents the body z-axis, which we assume is aligned with the pointing vector for a radar or imaging sensor. The trajectories traced out by the body axes illustrate the minimum-time 60 degree maneuver. We qualitatively note that the minimum-time slew maneuver does not resemble a simple eigenaxis slew about some axis.

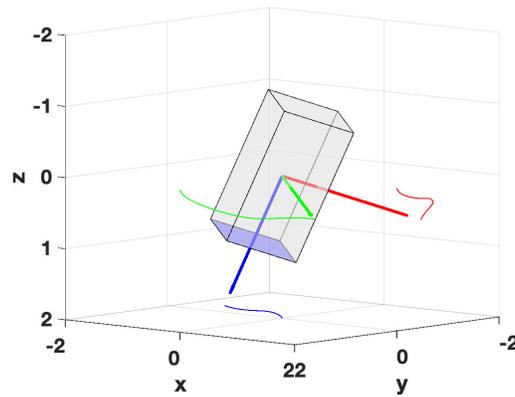


Fig. 4 Minimum-time maneuver (60-degree rotation about an arbitrary axis)

In Fig. (5), we note that the quaternion trajectory reaches the desired endpoint values with zero angular velocities. The optimal maneuver time is approximately 21.3 seconds. Given the same maneuver and actuator constraints, we expect that a spacecraft with less moment of inertia would require less time. Hence, the trend towards small satellites not only result in cost efficiency but also nimble operation.

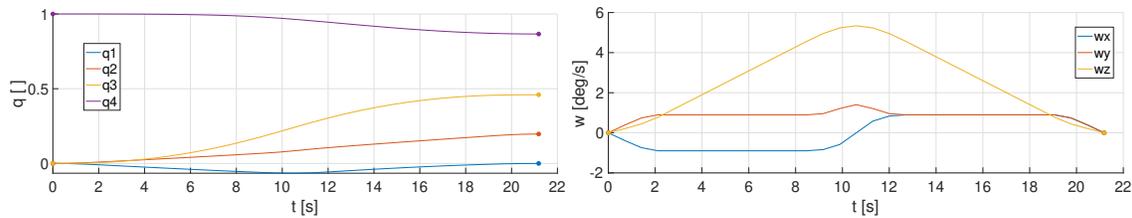


Fig. 5 (Left) Quaternion parameterization of attitude, (Right) Body Angular Velocity

Figure (6) shows that the momentum of the rotors do not saturate for this maneuver. For larger rotation magnitudes, we expect that the momentum trajectories may approach the bounds. We do observe that the rotor torques saturate and follow a “bang-bang” structure, consistent with minimum-time maneuvers [41]-[42].

Finally, in Fig. (7), we note that the peak instantaneous power for all four rotors reaches stays below 15 W and total energy consumption is less than 150 J for the maneuver. Given mission constraints on the instantaneous power drawn by the attitude control system or on the energy used to conduct a maneuver, we may enforce constraints (27) and (28).

After running 8,800 trajectory optimization instances (i.e., 88 rotations about 100 arbitrary axes), we plot the maneuver time and required energy data points in Fig. (8). We observe that maneuver time is a power function of the rotation magnitude and required energy is a linear function of rotation magnitude.

We also produced data for pure body axis rotations (i.e., about x-, y-, and z-axes) and found coefficients for a minimum-time power model, $T = a\theta^b$, and energy linear model, $E = c\theta + d$, valid for $\theta > 1^\circ$ about each axis. Using least-squares fitting, we find the power coefficient in the minimum-time model to take on the value of $b = 0.5$ exactly for

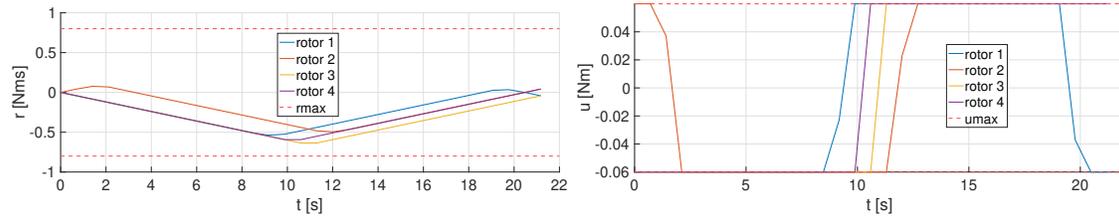


Fig. 6 (Left) Rotor Momentum, (Right) Rotor Torque

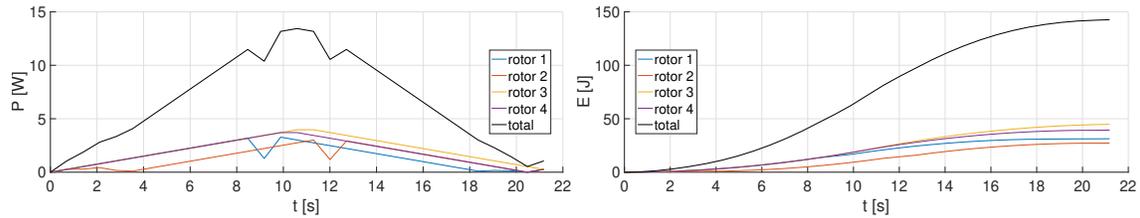


Fig. 7 (Left) Instantaneous Power drawn, (Right) Cumulative Energy Consumed

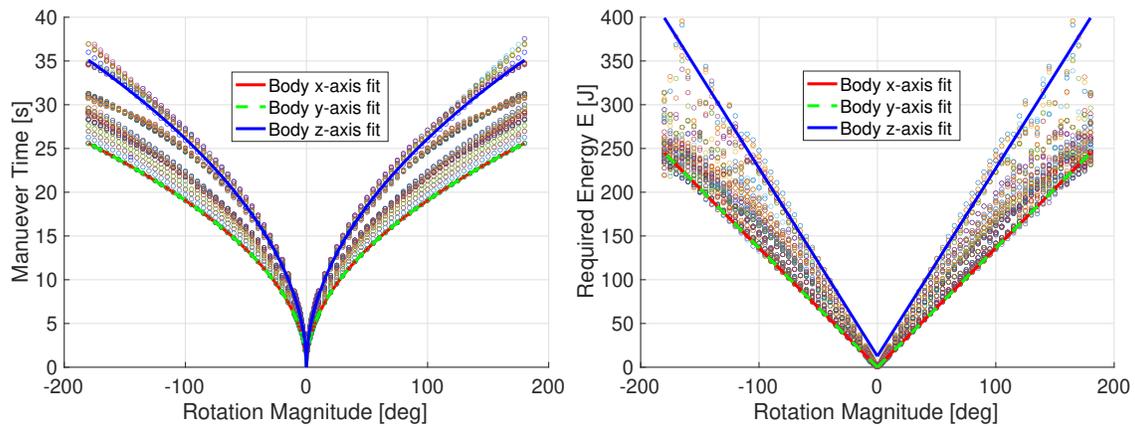


Fig. 8 (Left) Minimum-time (Right) Energy as functions of rotation magnitude θ

x- or y-axis rotations and nearly for all other axes. Hence, the optimal maneuver time is proportional to the square-root of the rotation magnitude in radians. We note that the well-known rest-to-rest, minimum-time, double-integrator problem also reveals this relationship, as mentioned in [13] and proved in [41] and [42]. This result provides assurance that our trajectory optimization method is indeed finding optimal values and solutions.

Body Axis	a	b	c	d
x	14.4371	0.5000	78.0329	-0.0225
y	14.4371	0.5000	78.0329	-0.0225
z	19.7292	0.5033	123.1154	12.5189

Table 2 Coefficients produced by least squares fitting

B. Application of Minimum-Attitude-Error-and-Control-Effort OCP

Of the 726 grid points in the 6-region cluster of our *Minimum-Attitude-Error-and-Control-Effort OCP* example, we produce a schedule based on heuristics that observes 33 grid points (GPs) in each of five regions (165 GPs in total or 22.7 percent of total number of accessible grid points). The schedule is produced based on estimates of the time required to move between neighboring grid points (intra-region slewing) and neighboring regions (inter-region slewing). Observing at an altitude of 710 km with approximately 8 km distance between each grid point, we estimate an optimal slew time of less than 4 seconds between grid points. However, slewing between regions that are several hundreds of kilometers apart can require up to 15 seconds. Given that the satellite passes over this 6-region cluster in approximately 12 minutes, we find that 165 GPs is close to the maximum number of grid points that a satellite may feasibly observe (assuming we also maximize visited regions). The other grid points must be observed on a subsequent revisit or by another satellite.

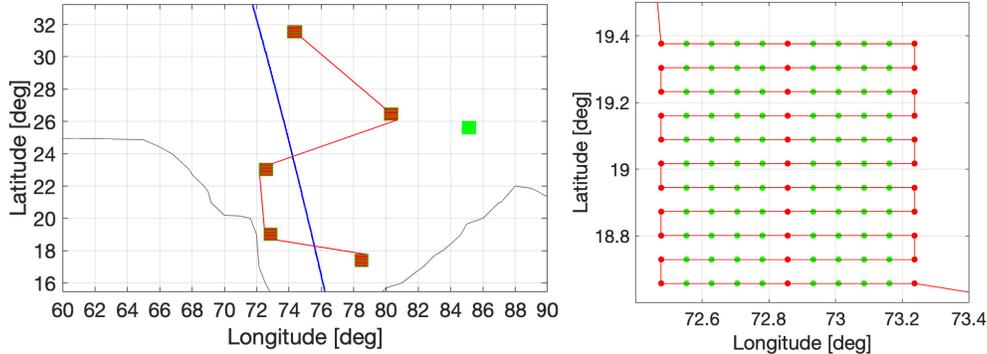


Fig. 9 (Left) Inter-region slewing, (Right) Intra-region whiskbroom slewing with schedule including 33 well-distributed (red) GPs of the 121 accessible (green) GPs for every region.

In Fig. (10), we note that the trajectory optimization finds a trajectory that passes through all desired attitude points with near zero angular velocity.

As shown in Fig. (11), the rotor momentum stays within the bounds and peaks when conducting large angle maneuver as required for slewing between regions. The rotor torques saturate for most of the trajectory, signifying that the spacecraft is working at its performance limits to achieve an aggressive schedule. We would expect this actuator saturation for schedules that maximize the number of observed grid points.

Finally, we observe in Fig. (12) that the total instantaneous power peaks during inter-region slewing maneuvers. A total energy of approximately 550 J is consumed to conduct this 10-minute observation trajectory.

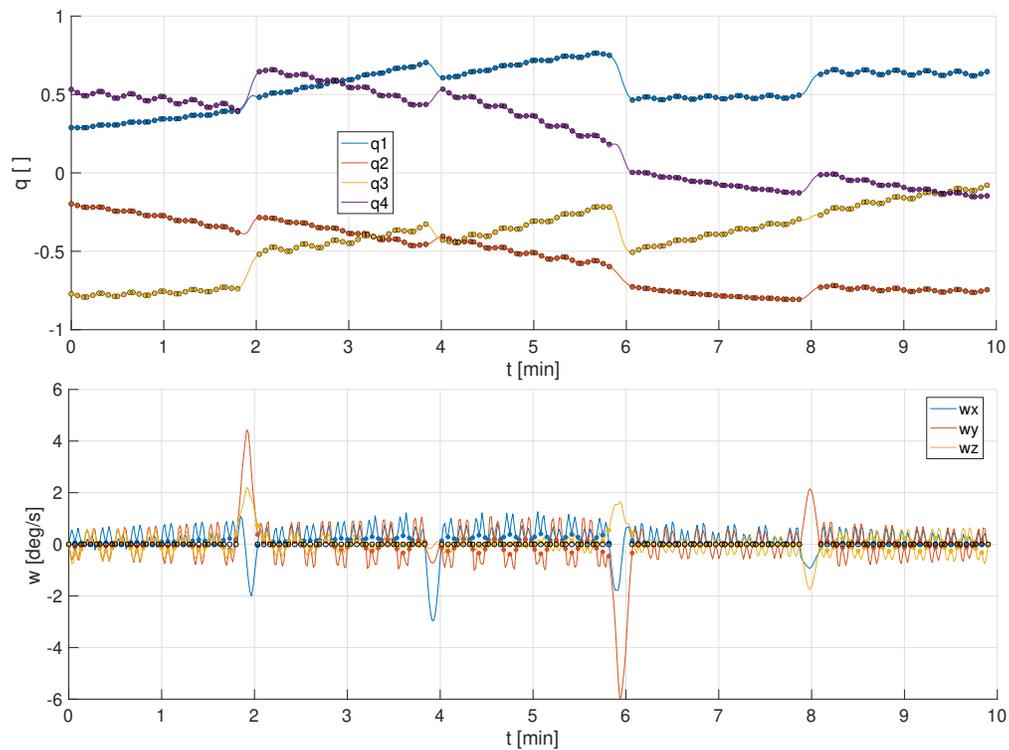


Fig. 10 (Top) Quaternion parameterization of attitude, (Bottom) Body Angular Velocity

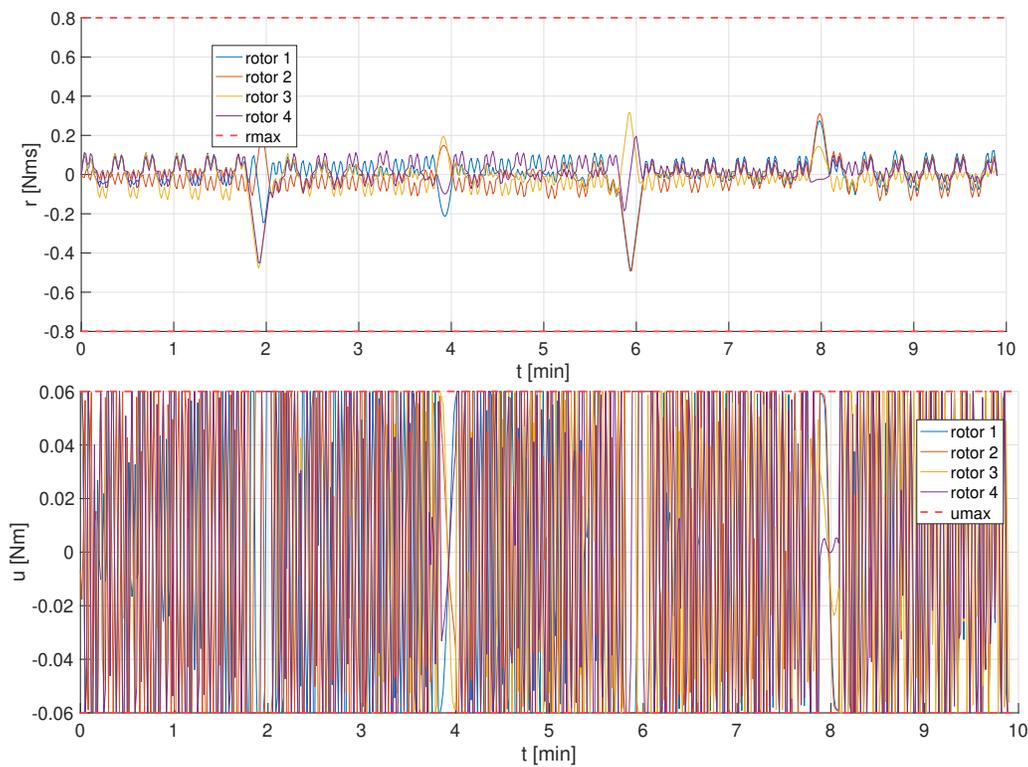


Fig. 11 (Top) Rotor Momentum, (Bottom) Rotor Torque

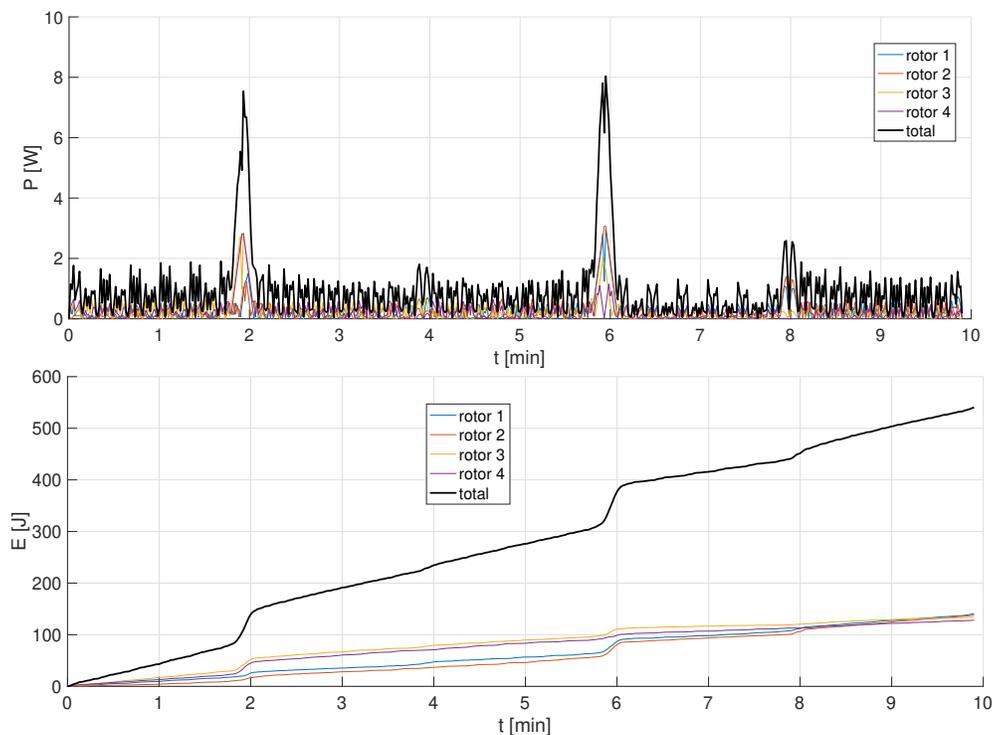


Fig. 12 (Top) Instantaneous Power drawn, (Bottom) Cumulative Energy consumed

VII. Conclusion

We have formulated and demonstrated two attitude trajectory optimization problems that complement each other. The parametric relationships obtained in the results of Section VI.A are used to inform an optimization-based constellation scheduler. The resulting schedules may then be executed using the approach illustrated in Section VI.B.

Acknowledgments

This project has been funded and supported by NASA's Earth Science Technology Office via the Advanced Information Systems Technology grant. We thank the Orbital R&D Team at Planet Labs for providing representative parameter values for the reference satellite used in this paper and for sharing insight on the state-of-the-practice in satellite attitude control.

References

- [1] S. Nag, A. Li, J.H. Merrick, "Scheduling algorithms for rapid imaging using agile Cubesat constellations," in *Advances in Space Research*, vol. 61, no. 3, pp. 891-913, November 2017.
- [2] S. Nag, A. Li, V. Ravindra, M.S. Net, K.M. Cheung, R. Lammers, B. Bledsoe, "Autonomous Scheduling of Agile Spacecraft Constellations with Delay Tolerant Networking for Reactive Imaging," in *Proceedings of the International Conference on Planning and Scheduling SPARK Workshop*, Berkeley, CA, USA, July, 2019.
- [3] S. Nag, M.S. Net, A. Li, V. Ravindra, "Designing a Disruption Tolerant Network for Reactive Spacecraft Constellations," in *Proceedings of the ASCEND*, Virtual Event, November 2020.
- [4] S. Nag, A. Aguilar, R. Akbar, A. Azemati, J. Frank, R. Levinson, A. Li, M. Moghaddam, V. Ravindra, D. Selva, "D-SHIELD: Distributed Spacecraft with Heuristic Intelligence to Enable Logistical Decisions," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, Virtual Event, September 2020.
- [5] V. Shah, V. Vittaldev, L. Stepan, C. Foster, "Scheduling the World's Largest Earth-Observing Fleet of Medium-Resolution Imaging Satellites," in *Proceedings of the International Workshops on Planning and Scheduling for Space*, Berkeley, CA, USA, July, 2019.

- [6] M. Harris, "Tech giants race to build orbital internet [News]," in *IEEE Spectrum*, vol. 55, no. 6, pp. 10-11, June 2018.
- [7] A. Li, J. Mason, "Optimal Utility of Satellite Constellation Separation with Differential Drag," in *Proceedings of the AIAA Space Forum*, San Diego, CA, USA, August 2014.
- [8] C. Foster, J. Mason, V. Vittaldev, L. Leung, V. Beukelaers, L. Stephan, R. Zimmerman, "Constellation Phasing with Differential Drag on Planet Labs Satellites," *Journal of Spacecraft and Rockets*, vol. 55, no. 2, pp. 473-483, March 2018.
- [9] E. Sin, M. Arcaç, A. Packard, "Small Satellite Constellation Separation using Linear Programming based Differential Drag Commands," in *Proceedings of the IEEE American Control Conference*, Milwaukee, WI, USA, June 2018.
- [10] E. Sin, H. Yin, M. Arcaç, "Passivity-based Distributed Acquisition and Station-keeping Control of a Satellite Constellation in Areostationary Orbit," in *Proceedings of the ASME Dynamic Systems and Control Conference*, Virtual Event, October 2020.
- [11] B. Wie, H. Weiss, A. Arapostathis, "Quaternion Feedback Regulator for Spacecraft Eigenaxis Rotations," *Journal of Guidance, Control, and Dynamics*, vol. 12, no. 3, pp. 375-380, May 1989.
- [12] J.T. Wen, K. Kreutz-Delgado, "The Attitude Control Problem," *IEEE Transactions on Automatic Control*, vol. 36, no. 10, pp. 1148-1162, October 1991.
- [13] K.D. Bilimoria, B. Wie, "Time-Optimal Three-Axis Reorientation of a Rigid Spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 3, pp. 446-452, May 1993.
- [14] B. Wie, J. Lu, "Feedback Control Logic for Spacecraft Eigenaxis Rotations Under Slew Rate and Control Constraints," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 6, pp. 1372-1379, November 1995.
- [15] J.T. Betts, "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193-207, March 1998.
- [16] M. Kelly, "An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation," *SIAM Review*, vol. 59, no. 4, pp. 849-904, 2017.
- [17] I.M. Ross, "A Primer on Pontryagin's Principle in Optimal Control," San Francisco, CA, USA: Collegiate Publishers, 2015.
- [18] J.T. Betts, *Practical Methods for Optimal Control and Estimation using Nonlinear Programming*, Philadelphia, PA, USA: SIAM, 2010.
- [19] J. Nocedal, S.J. Wright, *Numerical Optimization*, New York, NY, USA: Springer, 2006.
- [20] M. Szmuk, U. Eren, B. Açıkmeşe, "Successive Convexification for Mars 6-DoF Powered Descent Landing Guidance," in *Proceedings of the AIAA SciTech Forum*, Grapevine, TX, USA, January 2017.
- [21] M. Szmuk, B. Açıkmeşe, "Successive Convexification for 6-DoF Mars Rocket Powered Landing with Free-Final-Time," in *Proceedings of the AIAA SciTech Forum*, San Diego, CA, USA, January 2018.
- [22] D. Dueri, Y. Mao, Z. Mian, J. Ding, B. Açıkmeşe, "Trajectory Optimization with Inter-sample Obstacle Avoidance via Successive Convexification," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, Melbourne, Australia, December 2017.
- [23] D. Malyuta, T.P. Reynolds, M. Szmuk, M. Mesbahi, B. Açıkmeşe, "Discretization Performance and Accuracy Analysis for the Powered Descent Guidance Problem," in *Proceedings of the AIAA SciTech Forum*, San Diego, CA, USA, January 2019.
- [24] Y. Mao, M. Szmuk, B. Açıkmeşe, "A Tutorial on Real-time Convex Optimization Based Guidance and Control for Aerospace Applications," in *Proceedings of the IEEE American Control Conference*, Milwaukee, WI, USA, June 2018.
- [25] Y. Mao, M. Szmuk, X. Xu, B. Açıkmeşe, "Successive Convexification: A Superlinearly Convergent Algorithm for Non-Convex Optimal Control Problems," *arXiv:1804.06539 [math.OC]*, February 2019.
- [26] M. Szmuk, T.P. Reynolds, B. Açıkmeşe, "Successive Convexification for Real-Time Six-Degree-of-Freedom Powered Descent Guidance with State-Triggered Constraints," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 8, pp. 1399-1413, August 2020.
- [27] T.P. Reynolds, M. Szmuk, D. Malyuta, M. Mesbahi, B. Açıkmeşe, "Dual Quaternion-Based Powered Descent Guidance with State-Triggered Constraints," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 9, pp. 1584-1599, September 2020.

- [28] M. Szmuk, D. Malyuta, T.P. Reynolds, M.S. Mceowen, B. Açıkmeşe, “Real-Time Quad-Rotor Path Planning Using Convex Optimization and Compound State-Triggered Constraints,” *arXiv:1902.09149 [math.OC]*, February 2019.
- [29] M. Szmuk, T.P. Reynolds, B. Açıkmeşe, M. Mesbahi, “Successive Convexification for 6-DoF Powered Descent Guidance with Compound State-Triggered Constraints,” in *Proceedings of the AIAA SciTech Forum*, San Diego, CA, USA, January 2019.
- [30] D. Malyuta, T.P. Reynolds, M. Szmuk, B. Ackimese, M. Mesbahi, “Fast Trajectory Optimization via Successive Convexification for Spacecraft Rendezvous with Integer Constraints,” *arXiv:1906.04857 [math.OC]*, June 2019.
- [31] M. Szmuk, “Successive Convexification & High Performance Feedback Control for Agile Flight,” PhD dissertation, University of Washington, 2019.
- [32] T.P. Reynolds, D. Malyuta, M. Mesbahi, B. Açıkmeşe, “A Real-Time Algorithm for Non-Convex Powered Descent Guidance,” in *Proceedings of the AIAA SciTech Forum*, Orlando, FL, USA, January 2020.
- [33] P.C. Hughes, *Spacecraft Attitude Dynamics*, Mineola, New York, USA: Dover Publications, 1986.
- [34] J.R. Wertz, *Spacecraft Attitude Determination and Control*, Dordrecht, Holland: D. Reidel Publishing Company, 1978.
- [35] F.L. Markley, J.L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*, New York, USA: Springer, 2014.
- [36] I.M. Ross, Q. Gong, M. Karpenko, R.J. Proulx, “Scaling and Balancing for High-Performance Computation of Optimal Controls,” *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 10, pp. 2086-2097, October 2018.
- [37] D. Hull, “Conversion of Optimal Control Problems into Parameter Optimization Problems,” *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 57-60, January 1997.
- [38] Van Ryswyk, “Planet announces 50cm Skysat imagery, tasking dashboard and up to 12x Revisit,” *Planet Pulse*, June 2020.
- [39] “NASA Earth Observatory: Earth Observing-1 (EO-1),” https://earthobservatory.nasa.gov/features/E01/eo1_2.php, November 2000.
- [40] K. Shoemake, “Animating rotation with quaternion curves,” in *Proceedings of SIGGRAPH Computer Graphics*, San Francisco, CA, USA, July 1985.
- [41] M. Athan, P.L. Falb, *Optimal Control: An Introduction to the Theory and Its Applications*, New York, USA: McGraw Hill, 1966.
- [42] A.E. Bryson JR, Y.C. Ho, *Applied Optimal Control*, New York, NY, USA: Taylor & Francis, 1975.
- [43] B. Anderson, J. Moore, “Optimal Control: Linear Quadratic Methods”, Englewood Cliffs, NJ, USA: Prentice-Hall, 1990.