Real-Time Optimal Slew Maneuver Planning for Small Satellites

Robin Aucoin
Space Flight Laboratory, University of Toronto Institute for Aerospace Studies
4925 Dufferin Street, Toronto, Ontario, Canada, M3H 5T6
raucoin@utias-sfl.net

Faculty Advisor: Dr. Robert E. Zee
Space Flight Laboratory, University of Toronto Institute for Aerospace Studies

ABSTRACT

As the small-satellite market grows, so does the demand for large-scale small-satellite missions with diverse payload functions. To facilitate mission planning, and to enable autonomous transition between payload operations, real-time on-board slew maneuver path planning is often required to reorient the spacecraft. The computed trajectory must take into account a variety of kinematic and dynamic constraints on the spacecraft attitude, angular velocity and actuator limits. It is also desirable to compute this trajectory such that it is optimal in some sense. To simplify computation, current applications typically employ a "rest-to-rest" assumption where the maneuver endpoints are assumed to be inertially fixed, leading to long settling times in practice when dynamic endpoints exist. This settling time prohibits command of maneuvers in quick succession, which can limit operational capabilities. By posing the calculation as a convex semidefinite programming optimization problem, this paper presents a computationally attractive path-planning algorithm that computes an optimal fixed-time trajectory between arbitrary endpoints, while satisfying a variety of common attitude control constraints. In addition, a closed-form iterative solution for a minimum-time maneuver is proposed. Both methods are validated in a simulation case involving the University of Toronto Institute for Aerospace Studies Space Flight Laboratory's (UTIAS-SFL) next-generation DEFIANT-class spacecraft.

I. INTRODUCTION

Over the past decades, small satellites have revolutionized the space industry. Originally developed in small university research labs, they have moved to the aerospace mainstream, providing an alternative to traditional big-space missions while performing the same functions at a fraction of the cost and development time. As the market grows, there is a continuing drive to increase returns from each spacecraft, while maintaining the price point and turn-around time required to remain competitive in the small-satellite revolution. To this end, small spacecraft are now being equipped with multiple payloads to increase return on investment, or designed to work together in large constellations to achieve a common function.

With the growing scale of these missions comes the need to automate operation on board the spacecraft. In particular, when multiple payloads must be operated in sequence, or Earth-observation of several locations in a short time span is desired, path-planning for spacecraft reorientation maneuvers between these functions becomes a non-trivial problem, requiring real-time computation. For instance, minimizing the maneuver time can increase the number of observations in a single pass, providing customers with more payload data, and in turn a more profitable business. Moreover, on-board maneuver planning can enable new functionality, such as autonomously cued observations, where operator intervention and direction is not feasible.

In current on-board path-planning implementations, a "rest-to-rest" condition typically is assumed, where a trajectory is generated as a minimum-angle slew about a single rotation axis between the initial and final attitudes, with the assumption of near-zero angular velocity endpoints. This gives a problem that can be solved quickly in closed-form. Though this approximation is sufficient in many cases, it breaks down when the endpoints are not inertially fixed, leading to long settling times at the end of the maneuver, and limiting the number of maneuvers that can be performed in a given period. This is the case for target-tracking maneuvers, where body-fixed payloads (such as an imager) must track a stationary or moving target on Earth. It is therefore desirable to generate a trajectory that takes into account these dynamic endpoints in order to extend spacecraft capabilities, and enable missions with more demanding operational requirements.

In general, the goal of maneuver path planning is to compute a trajectory that takes the spacecraft from a known initial state (attitude, angular velocity, angular acceleration) to a known final state in an optimal manner. An optimal trajectory is one that minimizes some objective function, without violating a set of kinematic or dynamic constraints. In particular, three common constraints are considered in this work. First, “keep-out” or “keep-in” cones, where some payload is constrained to point outside or inside some subset of the attitude sphere. Second, angular velocity constraints, where the slew rate is bounded to enable operation of rate-limited sensors, or to prevent reaction wheel saturation. Third, angular acceleration constraints, which can be derived from spacecraft actuator limitations.

The problem of optimal maneuver planning has been extensively studied in the past. For instance, [1] proposes a Probabilistic RoadMap solution framework, [2] details a potential function approach, and [3] describes a method used on the Cassini mission involving an iterative convergence to the optimal solution using a genetic algorithm. However, the potential function approach can be prone to non-convergence in certain cases, and the Cassini method still required operator input for maneuver planning. Moreover, all three methods are computationally demanding, making them prohibitive for real-time implementation on board small satellites that often have limited computational power. They are typically concerned with finding a feasible, or locally optimal, point within a large non-convex design space densely populated with constraints. Small satellites, however, often operate under fewer constraints, allowing the problem to be simplified to enable real-time on-board implementation.

More recently, [4] proposed a guidance law for determining attitude trajectories between land-survey targets using low-order spline interpolation. Attitude spline interpolation techniques are also discussed in [5]. Though computationally attractive, these methods do not incorporate the constraints that typically govern spacecraft guidance. To incorporate these constraints, [6], [7], [8] pose the attitude maneuver calculation as a quadratically-constrained quadratic programming problem, solved using Semidefinite Programming (SDP) techniques [9]. This technique is of particular interest, as SDP problems can be efficiently solved using interior-point methods, potentially enabling on-board computation. However, the proposed discretization method, which involves determining a set of torque, angular velocity and attitude setpoints at discrete times throughout the maneuver, can lead to a large set of optimization variables for long slews, complicating computation.

Therefore, there exists a gap in the current literature to provide an attitude path-planning solution geared toward small satellites that is both computationally viable, and able to incorporate the various constraints that govern spacecraft operation. This paper seeks to fill this gap in two ways. Firstly, a computationally attractive attitude path-planning algorithm is introduced involving high-order spline interpolation solved as an SDP problem. This makes use of the computational reliability and efficiency of SDP optimization to ensure constraint satisfaction, while eliminating the quadratic programming element and taking advantage the added simplicity of spline interpolation. Secondly, an iterative closed-form solution for minimum-time attitude maneuvers is proposed that incorporates common attitude, angular velocity and angular acceleration constraints into its formulation. In addition to providing a minimum-time solution, this formulation is more computationally simple for cases where processing power is severely limited and on-board optimization is not possible.

The University of Toronto Institute for Aerospace Studies Space Flight Laboratory (UTIAS-SFL) has over 20 years of experience in designing, manufacturing, and operating small spacecraft for both research and commercial purposes. Its most recent development, the DEFIANT platform [10], leverages this experience to provide a bus that can deliver a multitude of payload functions including spacecraft formation control, radio frequency (RF) signal based target geolocation, and ground observation with optical payloads. Though the proposed path-planning algorithm is applicable to nearly all future UTIAS-SFL missions, for the purposes of demonstration, it will validated in a hypothetical DEFIANT mission involving a controlled formation of satellites geolocating an Earth-based target and cuing follow-up observation from a trailing spacecraft with an optical payload.

The structure of the paper is as follows. In Section II relevant background information and mathematical preliminaries are provided. Following this, Section III provides a mathematical derivation of the relevant dynamic and kinematic constraints. Sections IV and V then detail the optimal-spline, and iterative closed-form solutions respectively. Finally, simulation results demonstrating the utility of the proposed technique are provided in Section VI for the hypothetical DEFIANT mission.

II. Background

Reference Frames

Four reference frames are required to fully represent the spacecraft operations described in this paper. A brief description of each frame is given below, and the relationship between them is shown in Figure 1.

Earth Centered Inertial (ECI): An ECI frame $\mathcal{F}_i$ is a frame with an origin at the center of the Earth that is inertially fixed, meaning that the axes do not rotate with respect to the stars. The most commonly used ECI frame, is the J2000 frame, which has its x-axis aligned with the vernal equinox at 12:00 January 1st 2000, Terrestrial Time, the z-axis aligned with the Earth spin axis at that time, and the y-axis completing the orthonormal triad.

Earth Centered Earth Fixed (ECEF): The ECEF frame $\mathcal{F}_e$ is a rotating frame with an origin at the center of the Earth. The z-axis is aligned with true north, the x-axis is aligned with the vector from the origin to 0° latitude and longitude, and the y-axis completes the orthonormal triad.

Orbit Frame: An orbit frame $\mathcal{F}_o$ is one that rotates with the spacecraft's position in its orbit. By convention, we will refer to the Local Vertical Local Horizontal (LVLH) orbit frame, which has the x-axis aligned with the spacecraft position vector relative to the center of the Earth, z-axis aligned with the orbit angular momentum vector, and the y-axis completing the orthonormal triad. For circular orbits, this aligns the y-axis with the spacecraft velocity vector.

Body Frame: A spacecraft body frame $\mathcal{F}_b$ is any frame that is fixed to, and rotates with, the spacecraft. It is used to denote the spacecraft's attitude relative to any other frame.

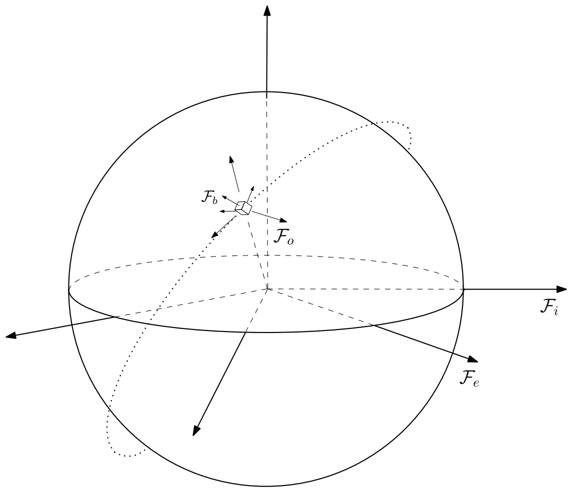


Fig. 1: Relationship between the ECI frame $\mathcal{F}_i$, ECEF frame $\mathcal{F}_e$, orbit frame $\mathcal{F}_o$, and body frame $\mathcal{F}_b$.

Attitude Parametrization

The orientation of some arbitrary frame $\mathcal{F}_c$ relative to another arbitrary frame $\mathcal{F}_d$ is fully described by a rotation matrix $\mathbf{C}_{cd} \in SO(3)$, where $SO(3) := \{\mathbf{C} \in \mathbb{R}^{3\times 3} \mid \mathbf{C}\mathbf{C}^{\mathsf{T}} = \mathbf{1},\ \det(\mathbf{C}) = 1\}$. However, for computational simplicity, parametrizations with fewer elements are typically used, such as Euler angles, quaternions and rotation vectors [11]. As noted in [11], the rotation vector parametrization permits three-element attitude representation without the singularities of Euler angles, and without the quadratic constraint of quaternions. In addition, as will be described in later sections, it allows for a convenient kinematic relationship between the derivatives of the attitude parametrization and the spacecraft angular velocity and acceleration. As such, the rotation vector parametrization is used herein.

Consider a spacecraft with a body-fixed frame at time t_j denoted by $\mathcal{F}_{b_j}$ whose attitude relative to some inertial frame $\mathcal{F}_i$ is given by the rotation matrix $\mathbf{C}_{b_j i}$. The attitude can be represented by the rotation vector $\mathbf{\Phi}^{b_j i}$ such that

$$\begin{aligned}\mathbf{C}_{b_j i} &= e^{-(\mathbf{\Phi}^{b_j i})^\times} \\ &= \cos\left(\phi^{b_j i}\right)\mathbf{1} + \left(1 - \cos(\phi^{b_j i})\right)\left(\frac{\mathbf{\Phi}^{b_j i}}{\phi^{b_j i}}\right)\left(\frac{\mathbf{\Phi}^{b_j i}}{\phi^{b_j i}}\right)^{\mathsf{T}} \\ &\quad - \sin\left(\phi^{b_j i}\right)\left(\frac{\mathbf{\Phi}^{b_j i}}{\phi^{b_j i}}\right)^\times, \qquad (1)\end{aligned}$$

where

$$\phi^{b_j i} = \sqrt{\left(\mathbf{\Phi}^{b_j i}\right)^{\mathsf{T}}\left(\mathbf{\Phi}^{b_j i}\right)},$$

$$\mathbf{v}^\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}.$$

Note that the apparent singularity at $\phi^{b_j i} = 0$ can be avoided by realizing that as $\mathbf{\Phi}^{b_j i} \to \mathbf{0}$, $\mathbf{C}_{b_j i} \to \mathbf{1}$.

Attitude Kinematics and Dynamics

At time t_j, the angular velocity of $\mathcal{F}_{b_j}$ relative to the inertial frame $\mathcal{F}_i$, resolved in $\mathcal{F}_{b_j}$, is denoted $\boldsymbol{\omega}_{b_j}^{b_j i}$. Its angular acceleration, as resolved in $\mathcal{F}_{b_j}$, is then $\dot{\boldsymbol{\omega}}_{b_j}^{b_j i}$. The relationship between the angular velocity and acceleration, and the derivatives of the rotation vector is

$$\dot{\mathbf{\Phi}}^{b_j i} = \mathbf{\Gamma}\ \boldsymbol{\omega}_{b_j}^{b_j i}, \qquad (2)$$

$$\ddot{\mathbf{\Phi}}^{b_j i} = \mathbf{\Gamma}\ \dot{\boldsymbol{\omega}}_{b_j}^{b_j i} + \dot{\mathbf{\Gamma}}\ \boldsymbol{\omega}_{b_j}^{b_j i}, \qquad (3)$$

where,

$$\mathbf{\Gamma} = \left(\frac{\mathbf{\Phi}^{b_j i}}{\phi^{b_j i}}\right)\left(\frac{\mathbf{\Phi}^{b_j i}}{\phi^{b_j i}}\right)^{\mathsf{T}} + \frac{\phi^{b_j i}}{2}\left(\left(\frac{\mathbf{\Phi}^{b_j i}}{\phi^{b_j i}}\right)^\times - \cot\left(\frac{\phi^{b_j i}}{2}\right)\left(\frac{\mathbf{\Phi}^{b_j i}}{\phi^{b_j i}}\right)^\times\left(\frac{\mathbf{\Phi}^{b_j i}}{\phi^{b_j i}}\right)^\times\right).$$

Again, the singularity is avoided at $\phi^{b_j i} = 0$ by imposing that when $\mathbf{\Phi}^{b_j i} = 0$, $\dot{\mathbf{\Phi}}^{b_j i} = \boldsymbol{\omega}_{b_j}^{b_j i}$, and $\ddot{\mathbf{\Phi}}^{b_j i} = \dot{\boldsymbol{\omega}}_{b_j}^{b_j i}$.

The rigid-body dynamics that govern spacecraft attitude control with reaction wheels are then given by Euler's equation [12], [13],

$$\mathbf{I}_{b_j}\dot{\boldsymbol{\omega}}_{b_j}^{b_j i} + \dot{\mathbf{h}}_{b_j}^{w} + \left(\boldsymbol{\omega}_{b_j}^{b_j i}\right)^{\times}\left(\mathbf{I}_{b_j}\boldsymbol{\omega}_{b_j}^{b_j i} + \mathbf{h}_{b_j}^{w}\right) = \boldsymbol{\tau}_{b_j} \quad (4)$$

where $\mathbf{I}_{b_j}$ is the moment of inertia of the spacecraft with respect to the center of mass, as resolved in $\mathcal{F}_{b_j}$, $\mathbf{h}_{b_j}^{w}$ is the angular momentum of the reaction wheels with respect to, and expressed in $\mathcal{F}_{b_j}$, and $\boldsymbol{\tau}_{b_j}$ is the net external torque on the spacecraft.

Semidefinite Programming

SDP is a form of convex optimization (and an extension of linear programming) originally developed in the 1990s [9], that is concerned with minimizing a linear objective function subject to Linear Matrix Inequality (LMI) constraints. In general, an SDP problem is one of the form,

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^m} \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x}, \\ \text{s.t.} \quad & \mathbf{F}(\mathbf{x}) \geqslant 0, \end{aligned}$$

where

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + \sum_{l=1}^{m} x_l \mathbf{F}_l.$$

In other words, the problem is to solve for the optimization variable $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix}^{\mathsf{T}}$ that minimizes a linear objective function involving a known set of parameters $\mathbf{c} \in \mathbb{R}^m$, subject to a positive semidefiniteness constraint on a linear combination of known symmetric matrices $\mathbf{F}_0, \ldots, \mathbf{F}_m \in \mathbb{R}^{n\times n}$. The positive semidefiniteness constraint $\mathbf{F}(\mathbf{x}) \geqslant 0$ implies that $\mathbf{F}(\mathbf{x}) = \mathbf{F}(\mathbf{x})^{\mathsf{T}}$, $\mathbf{v}^{\mathsf{T}}\mathbf{F}(\mathbf{x})\mathbf{v} \geqslant 0 \ \forall \mathbf{v} \in \mathbb{R}^n$.

Problems of this form give a convex feasible solution set. When the solution set is nonempty and paired with a linear objective function, there is a guarantee of convergence to the globally optimal solution. Moreover, the problem can be solved quickly and efficiently using interior-point methods. This can be done using custom solvers, or off-the-shelf solver packages such as MOSEK [14] combined with the YALMIP toolbox [15].

III. Dynamic and Kinematic Constraints

Throughout a slew maneuver, the spacecraft is subject to several kinematic and dynamic constraints in order to ensure uninterrupted operation. As described in Section I, these include constraints on permissible attitudes, angular velocity constraints, reaction wheel speed limitations, or actuator torque limitations. The purpose of this section is to describe these constraints in more detail, and to reformulate or approximate them as LMI constraints so that them can be incorporated into an SDP optimization problem.

Attitude Constraints

In attitude path planning, it is often desirable, or even mandatory, to enforce constraints on permissible spacecraft attitudes. Typically this can be modeled as "keep-in" or "keep-out" cones, where a payload vector is constrained to point within a certain subset of the attitude sphere. For instance, there could be a desire to point light-sensitive payloads (such as cameras or star trackers) a minimum angular distance from the Sun to prevent damage or permit operation. Similarly, radiators may be constrained to point toward deep space for thermal control purposes. Alternatively, in order to maximize power generation, the body face with the largest number of solar cells may be constrained to point toward the Sun, within some tolerance. Moreover, during operation, communication payloads often must maintain contact with ground stations, necessitating a constraint on the antenna pointing.

Mathematically, these constraints are

$$(\mathbf{v}_a^c)^{\mathsf{T}}\mathbf{C}_{b_j a}^{\mathsf{T}}(\mathbf{v}_{b_j}^p) \leqslant \cos(\theta_{\text{min}}), \quad (5)$$

and

$$(\mathbf{v}_a^c)^{\mathsf{T}}\mathbf{C}_{b_j a}^{\mathsf{T}}(\mathbf{v}_{b_j}^p) \geqslant \cos(\theta_{\text{max}}), \quad (6)$$

where $\mathbf{v}_{b_j}^p$ is the payload vector resolved in $\mathcal{F}_{b_j}$ at time t_j, $\mathbf{v}_a^c$ is the constraint vector resolved in some frame $\mathcal{F}_a$, $\mathbf{C}_{b_j a}$ is the rotation matrix between the two frames in question. Note that (5) and (6) represent the "keep-out" and "keep-in" cases respectively, and θ_{max} and θ_{min} denote the maximum and minimum permissible cone half angles. Typically, we want $\mathcal{F}_a$ to be an inertial frame, however, not all constraint vectors are conveniently expressed in an inertial frame. For instance, throughout the duration of a slew maneuver, the Sun is approximately fixed in the ECI frame, whereas a nadir constraint would be best expressed in the LVLH frame, and a fixed ground target would be best expressed in the ECEF frame. If the constraint vector is not defined in the inertial frame, the known rotation matrix between the inertial frame and the constraint frame must be included in (5) and (6). For notational simplicity, a "keep-out" constraint for a constraint vector expressed in the ECI frame will be explored herein, without loss of generality.

Constraints (5) and (6) are in general non-convex, and cannot be directly implemented into an SDP problem in this form. An iterative approach is therefore proposed to apply this constraint in a convex manner. First, the optimal trajectory is computed without considering attitude constraints. If the solution violates a constraint, we introduce the intermediate constraint

$$(\mathbf{v}_i^c)^\mathsf{T}\mathbf{C}_{b_ji}^\mathsf{T}(\mathbf{v}_{b_j}^p) \leqslant \cos(\bar{\theta}_{min}+\delta\theta),$$

where $\bar{\theta}_{min}$ is the minimum angle of approach along the computed trajectory, and $\delta\theta$ is a small perturbation from this angle. We then linearize the constraint along the previously computed trajectory, and enforce

$$(\mathbf{v}_i^c)^\mathsf{T}\bar{\mathbf{C}}_{b_ji}^\mathsf{T}(\mathbf{v}_{b_j}^p) + \mathbf{F}_j(\mathbf{\Phi}^{b_ji}-\bar{\mathbf{\Phi}}^{b_ji}) \leqslant \cos(\bar{\theta}_{min}+\delta\theta), \tag{7}$$

where

$$\mathbf{F}_j = \frac{\mathrm{d}f}{\mathrm{d}\mathbf{\Phi}^{b_ji}}\bigg|_{\mathbf{\Phi}^{b_ji}=\bar{\mathbf{\Phi}}^{b_ji}}, \quad f(\mathbf{\Phi}^{b_ji}) = (\mathbf{v}_i^c)^\mathsf{T}\mathbf{C}_{b_ji}^\mathsf{T}(\mathbf{v}_{b_j}^p). \tag{8}$$

An overbar denotes a value computed on the previous iteration, and $\mathbf{F}_j$ is a Jacobian matrix computed about the previous trajectory. Note that (8) can be expressed in terms of the chosen attitude parametrization using (1). With this constraint added, the optimal trajectory is recomputed, and a new $\bar{\theta}_{min}$ is found. This process is repeated until $\bar{\theta}_{min} = \theta_{min}$, gradually pushing the trajectory out of the avoidance cone. Keeping $\delta\theta$ small (on the order of a few degrees) on each iteration ensures that the linearization is valid. This linearization technique, similar to the one presented in [16], allows (5) and (6) to be expressed as linear convex constraints and be augmented to the SDP optimization problem.

Special attention is required when multiple constraint regions overlap. This could result in no feasible solution with this approach if an iteratively growing constraint cone pushes the trajectory into another constrained region. This case can be avoided by first including an artificial constraint along a vector oriented toward the geometric center of the multiple constraint vectors. A constraint cone is grown iteratively outward following the aforementioned procedure until the trajectory no longer crosses the central region between the multiple constraint vectors, as shown in Figure 2. Once the trajectory is outside of this region, it can safely be pushed out of the true constraint cones.

Angular Velocity Constraints

Spacecraft are often rate-limited during operation for several reasons. For instance, when using reaction wheels, the rate is limited by the ratio of the maximum possible angular momentum of the wheels and the bus moment of inertia. The spacecraft may also carry rate-limited payloads or sensors, such as star trackers, which do not return an accurate attitude solution above a certain angular rate. Mathematically, this constraint is given by

$$(\boldsymbol{\omega}_{b_j}^{b_ji})^\mathsf{T}(\boldsymbol{\omega}_{b_j}^{b_ji}) \leqslant \omega_{\text{max}}^2, \tag{9}$$

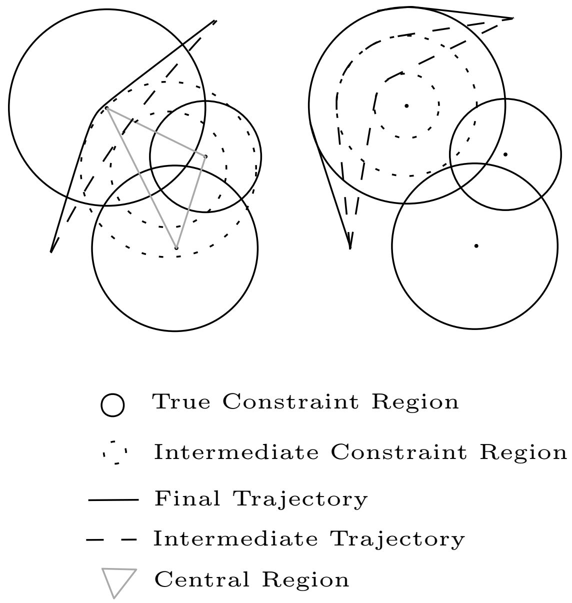


Fig. 2: Planar representation of the iterative attitude constraint application, showing the trajectory being pushed out of the constraint region. Left: first stage in cases with overlapping constraint regions. Right: final stage to push the trajectory out.

where ω_{max} is the maximum permissible angular velocity in the slew, as determined from one of the aforementioned drivers. However, the angular velocity constraint cannot be used in this form. It must be expressed in terms of our chosen attitude parametrization. Consider that from (2), the norm of the angular velocity can be related to the rate of change of the rotation vector by

$$(\dot{\mathbf{\Phi}}^{b_ji})^\mathsf{T}(\dot{\mathbf{\Phi}}^{b_ji}) = (\boldsymbol{\omega}_{b_j}^{b_ji})^\mathsf{T}\mathbf{\Gamma}(\mathbf{\Phi}^{b_ji})^\mathsf{T}\mathbf{\Gamma}(\mathbf{\Phi}^{b_ji})(\boldsymbol{\omega}_{b_j}^{b_ji}),$$

where it can be shown that $\lambda_{\text{min}}(\mathbf{\Gamma}(\mathbf{\Phi}^{b_ji})^\mathsf{T}\mathbf{\Gamma}(\mathbf{\Phi}^{b_ji})) = 1$ with a corresponding eigenvector oriented along $\mathbf{\Phi}^{b_ji}$. Note that the $\lambda(\cdot)$ operator denotes an eigenvalue calculation. Consequentially, the inequality

$$(\mathbf{\Phi}^{b_ji})^\mathsf{T}(\mathbf{\Phi}^{b_ji}) \geqslant (\boldsymbol{\omega}_{b_j}^{b_ji})^\mathsf{T}(\boldsymbol{\omega}_{b_j}^{b_ji}),$$

holds in all cases, and equality is achieved when $\boldsymbol{\omega}_{b_j}^{b_ji}$ is aligned with $\mathbf{\Phi}^{b_ji}$. Therefore, $(\dot{\mathbf{\Phi}}^{b_ji})^\mathsf{T}(\dot{\mathbf{\Phi}}^{b_ji})$ provides an upper bound on $(\boldsymbol{\omega}_{b_j}^{b_ji})^\mathsf{T}(\boldsymbol{\omega}_{b_j}^{b_ji})$, and with some conservatism, we can replace (9) with

$$(\dot{\mathbf{\Phi}}^{b_ji})^\mathsf{T}(\dot{\mathbf{\Phi}}^{b_ji}) \leqslant \omega_{\text{max}}^2, \tag{10}$$

Note that if the maneuver body frame is chosen such that $\mathbf{\Phi}^{b_0i} = \mathbf{0}$ at the slew start time t_0, then the maneuver will typically follow a near single-axis slew that is nearly aligned with the angular velocity vector, minimizing the conservatism introduced with this constraint relaxation. In addition, an iterative approach could be implemented to approximate $\mathbf{\Gamma}(\mathbf{\Phi}^{b_ji})^\mathsf{T}\mathbf{\Gamma}(\mathbf{\Phi}^{b_ji})$ along the optimal maneuver trajectory, but the author found that, in practice, this is not necessary to appropriately bound the angular velocity magnitude.

Equation (10) is then equivalent to the LMI constraint

$$\begin{bmatrix} \omega_{max}\mathbf{1} & \dot{\boldsymbol{\Phi}}^{b_j i} \\ \left(\dot{\boldsymbol{\Phi}}^{b_j i}\right)^{\mathsf{T}} & \omega_{max} \end{bmatrix} \geqslant 0, \tag{11}$$

which is written in terms of the chosen attitude parametrization, and can be incorporated into the SDP optimization problem.

Angular Acceleration Constraints

The spacecraft slew profile is also constrained by actuator limitations. Typically the spacecraft is controlled using reaction wheels, and the dynamics follow Equation (4). Though the actuator constraint is typically expressed as a torque limit, it is more easily added to the slew-maneuver planning problem as an angular acceleration constraint. To this end, we want to bound $||\dot{\boldsymbol{\omega}}_{b_j}^{b_j i}||$ such that the maximum value of $||\dot{\mathbf{h}}_{b_j}^{w}||$ induced throughout the maneuver does not exceed the wheels' capabilities. This can be done with varying levels of conservatism, but in general an appropriate approach is to choose some nominal bound for the angular velocity cross term in (4), and assume rotation about the principle axis with the largest moment of inertia. Mathematically, this gives

$$\dot{\omega}_{\text{max}} = \frac{\max\left(\dot{h}_{b_j}^{w}\right) - \max\left(\left\|(\boldsymbol{\omega}_{b_j}^{b_j i})^{\times}(\mathbf{I}_{b_j}\boldsymbol{\omega}_{b_j}^{b_j i} + \mathbf{h}_{b_j}^{w})\right\|\right)}{\lambda_{\text{max}}(\mathbf{I}_{b_j})},$$

where the max() operator indicates the maximum possible value given the subset of permissible angular velocities and wheel momenta, and $\dot{h}_{b_j}^{w}$ is the rate of change of momentum of a single reaction wheel.

Given this limiting angular acceleration, we take a similar approach to the angular velocity case, noting that

$$\begin{aligned}(\ddot{\boldsymbol{\Phi}}^{b_j i})^{\mathsf{T}}(\ddot{\boldsymbol{\Phi}}^{b_j i}) &= (\dot{\boldsymbol{\omega}}_{b_j}^{b_j i})^{\mathsf{T}}\boldsymbol{\Gamma}^{\mathsf{T}}\boldsymbol{\Gamma}(\dot{\boldsymbol{\omega}}_{b_j}^{b_j i}) \\ &\quad + 2(\boldsymbol{\omega}_{b_j}^{b_j i})^{\mathsf{T}}\dot{\boldsymbol{\Gamma}}^{\mathsf{T}}\boldsymbol{\Gamma}(\dot{\boldsymbol{\omega}}_{b_j}^{b_j i}) + (\boldsymbol{\omega}_{b_j}^{b_j i})^{\mathsf{T}}\dot{\boldsymbol{\Gamma}}^{\mathsf{T}}\dot{\boldsymbol{\Gamma}}(\boldsymbol{\omega}_{b_j}^{b_j i}).\end{aligned}$$

However, in practice for small spacecraft, slews are nearly performed about a single axis, with angular velocities on the order of a few degrees per second and angular accelerations on the order of fractions of a degree per second squared. In these cases, when the maneuver body frame is chosen such that $\boldsymbol{\Phi}^{b_0 i} = \mathbf{0}$ at the slew start time t_0,

$$\begin{aligned}(\dot{\boldsymbol{\omega}}_{b_j}^{b_j i})^{\mathsf{T}}\boldsymbol{\Gamma}^{\mathsf{T}}\boldsymbol{\Gamma}(\dot{\boldsymbol{\omega}}_{b_j}^{b_j i}) \gg{}& 2(\boldsymbol{\omega}_{b_j}^{b_j i})^{\mathsf{T}}\dot{\boldsymbol{\Gamma}}^{\mathsf{T}}\boldsymbol{\Gamma}(\dot{\boldsymbol{\omega}}_{b_j}^{b_j i}) \\ &+ (\boldsymbol{\omega}_{b_j}^{b_j i})^{\mathsf{T}}\dot{\boldsymbol{\Gamma}}^{\mathsf{T}}\dot{\boldsymbol{\Gamma}}(\boldsymbol{\omega}_{b_j}^{b_j i}).\end{aligned}$$

In fact, in the author's experience, this inequality holds by at least an order of magnitude in nearly all cases, particularly when the spacecraft is operating near its angular acceleration limit. Therefore, we can make the approximation

$$(\ddot{\boldsymbol{\Phi}}^{b_j i})^{\mathsf{T}}(\ddot{\boldsymbol{\Phi}}^{b_j i}) \approx (\dot{\boldsymbol{\omega}}_{b_j}^{b_j i})^{\mathsf{T}}\boldsymbol{\Gamma}^{\mathsf{T}}\boldsymbol{\Gamma}(\dot{\boldsymbol{\omega}}_{b_j}^{b_j i}),$$

and following the same constraint relaxation used to go from (9) to (10), we obtain

$$(\ddot{\boldsymbol{\Phi}}^{b_j i})^{\mathsf{T}}(\ddot{\boldsymbol{\Phi}}^{b_j i}) \leqslant \dot{\omega}_{\text{max}}^2,$$

which can be rewritten as the LMI constraint

$$\begin{bmatrix} \dot{\omega}_{max}\mathbf{1} & \ddot{\boldsymbol{\Phi}}^{b_j i} \\ \left(\ddot{\boldsymbol{\Phi}}^{b_j i}\right)^{\mathsf{T}} & \dot{\omega}_{max} \end{bmatrix} \geqslant 0, \tag{12}$$

and augmented to the SDP optimization problem.

IV. Optimal Spline Interpolation

Formulation

The goal of slew maneuver path planning is to determine an attitude, angular velocity, and angular acceleration profile that takes the spacecraft from a known initial state ($\mathbf{C}_{b_0 i}$, $\boldsymbol{\omega}_{b_0}^{b_0 i}$, and $\dot{\boldsymbol{\omega}}_{b_0}^{b_0 i}$) at time $t_0 = 0$, to a desired final state ($\mathbf{C}_{b_k i}$, $\boldsymbol{\omega}_{b_k}^{b_k i}$, and $\dot{\boldsymbol{\omega}}_{b_k}^{b_k i}$) in some time t_k. This can be done by interpolating the two endpoints using a high-order spline with a rotation vector parametrization. The spline interpolation guarantees satisfaction of the initial and terminal states, and the intermediate profile can be determined though an SDP optimization problem.

To this end, assume a maneuver duration of t_k such that $\boldsymbol{\Phi}^{b_k i}, \dot{\boldsymbol{\Phi}}^{b_k i}, \ddot{\boldsymbol{\Phi}}^{b_k i}$ are known, and let the optimal set of $\boldsymbol{\Phi}^{b_j i}, \dot{\boldsymbol{\Phi}}^{b_j i}, \ddot{\boldsymbol{\Phi}}^{b_j i}$ be represented by the polynomial series

$$\begin{aligned}\boldsymbol{\Phi}^{b_j i} &= \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1\left(\frac{t_j}{t_k}\right) + \cdots + \boldsymbol{\beta}_n\left(\frac{t_j}{t_k}\right)^n, \\ \dot{\boldsymbol{\Phi}}^{b_j i} &= \left(\frac{1}{t_k}\right)\boldsymbol{\beta}_1 + 2\boldsymbol{\beta}_2\left(\frac{t_j}{t_k^2}\right) + \cdots + n\boldsymbol{\beta}_n\left(\frac{t_j^{n-1}}{t_k^n}\right), \\ \ddot{\boldsymbol{\Phi}}^{b_j i} &= 2\boldsymbol{\beta}_2 + 6\boldsymbol{\beta}_3\left(\frac{t_j}{t_k^3}\right) + \cdots + n(n-1)\boldsymbol{\beta}_n\left(\frac{t_j^{n-2}}{t_k^n}\right), \\ &\forall j = 0, 1, \ldots, k\end{aligned} \tag{13}$$

where n is the order of the spline interpolation. If $\boldsymbol{\Phi}^{b_j i}, \dot{\boldsymbol{\Phi}}^{b_j i}, \ddot{\boldsymbol{\Phi}}^{b_j i}$ are known throughout the maneuver, then one can backsolve for the angular velocity and acceleration profiles by inverting (2) and (3) at each timestep. Rearranging (13) yields

$$\begin{bmatrix} \boldsymbol{\Phi}^{b_j i} \\ \dot{\boldsymbol{\Phi}}^{b_j i} \\ \ddot{\boldsymbol{\Phi}}^{b_j i} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{1} & \left(\frac{t_j}{t_k}\right)\mathbf{1} & \left(\frac{t_j}{t_k}\right)^2\mathbf{1} & \cdots & \left(\frac{t_j}{t_k}\right)^n\mathbf{1} \\ \mathbf{0} & \left(\frac{1}{t_k}\right)\mathbf{1} & 2\left(\frac{t_j}{t_k^2}\right) & \cdots & n\left(\frac{t_j^{n-1}}{t_k^n}\right)\mathbf{1} \\ \mathbf{0} & \mathbf{0} & 2\left(\frac{1}{t_k^2}\right)\mathbf{1} & \cdots & n(n-1)\left(\frac{t_j^{n-2}}{t_k^n}\right)\mathbf{1} \end{bmatrix}}_{\mathbf{T}_j \in \mathbb{R}^{9\times 3(n+1)}} \begin{bmatrix} \boldsymbol{\beta}_0 \\ \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_n \end{bmatrix}, \tag{14}$$

which, can be partitioned as

$$\mathbf{T}_j = \begin{bmatrix} \mathbf{T}_j^{\phi} \\ \mathbf{T}_j^{\omega} \\ \mathbf{T}_j^{\alpha} \end{bmatrix}.$$

When considering initial conditions, this gives

$$\underbrace{\begin{bmatrix} \boldsymbol{\Phi}^{b_0 i} \\ \dot{\boldsymbol{\Phi}}^{b_0 i} \\ \ddot{\boldsymbol{\Phi}}^{b_0 i} \\ \boldsymbol{\Phi}^{b_k i} \\ \dot{\boldsymbol{\Phi}}^{b_k i} \\ \ddot{\boldsymbol{\Phi}}^{b_k i} \end{bmatrix}}_{\boldsymbol{\Phi}\in\mathbb{R}^{18\times 1}} = \underbrace{\begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{21} & \cdots & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \cdots & \mathbf{1} \\ \mathbf{0} & \left(\frac{1}{t_k}\right)\mathbf{1} & \left(\frac{2}{t_k}\right)\mathbf{1} & \cdots & \left(\frac{n}{t_k}\right)\mathbf{1} \\ \mathbf{0} & \mathbf{0} & \left(\frac{2}{t_k^2}\right)\mathbf{1} & \cdots & \left(\frac{n(n-1)}{t_k^2}\right)\mathbf{1} \end{bmatrix}}_{\mathbf{W}\in\mathbb{R}^{18\times 3(n+1)}} \underbrace{\begin{bmatrix} \boldsymbol{\beta}_0 \\ \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \boldsymbol{\beta}_3 \\ \vdots \\ \boldsymbol{\beta}_n \end{bmatrix}}_{\boldsymbol{\beta}\in\mathbb{R}^{3(n+1)\times 1}}. \tag{15}$$

Provided that $n \geqslant 5$, $\mathbf{W} = \begin{bmatrix} \mathbf{T}_0^{\mathsf{T}} & \mathbf{T}_k^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$ is full rank and (15) can be solved for $\boldsymbol{\beta}$. In the case where $n = 5$, (15) is solved by inverting $\mathbf{W}$ such that $\boldsymbol{\beta} = \mathbf{W}^{-1}\boldsymbol{\Phi}$. This gives attitude and corresponding angular velocity/acceleration profiles that satisfy the endpoints exactly, but gives no control over what happens in between. The trajectory could require torques that exceed the limitations of the actuators, or angular velocities that exceed the imposed limits. As such, it is desirable to use a higher-order spline interpolation that is computed optimally subject to these constraints. This is done as follows: first the minimum-norm solution to (15) is computed using the pseudo-inverse of $\mathbf{W}$ such that $\bar{\boldsymbol{\beta}} = \mathbf{W}^{\mathsf{T}}\big(\mathbf{W}\mathbf{W}^{\mathsf{T}}\big)^{-1}\boldsymbol{\Phi}$. The solution for $\boldsymbol{\beta}$ then becomes

$$\boldsymbol{\beta} = \bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta},$$

where

$$\mathbf{N} = \begin{bmatrix} \mathbf{n}_1 & \mathbf{n}_2 & \cdots & \mathbf{n}_{(3n-15)} \end{bmatrix} \in \mathbb{R}^{3(n+1)\times(3n-15)}$$

such that

$$\mathbf{W}\mathbf{n}_l = \mathbf{0}, \quad \forall l = 1, \ldots, 3n-15,$$

and

$$\boldsymbol{\eta} \in \mathbb{R}^{(3n-15)\times 1}$$

is a column matrix that scales and sums the null space components found in $\mathbf{N}$. Manipulating $\boldsymbol{\eta}$ allows us to control the shape of the slew maneuver without affecting the end points. Our goal then becomes to compute $\boldsymbol{\eta}$ in an optimal manner subject to attitude, angular velocity and angular acceleration constraints.

Objective Function Choice

To be considered an optimal maneuver, the trajectory should be chosen such that it minimizes some objective function within the feasible set of solutions. For an SDP problem, we want the objective function to be linear in terms of the optimization variable $\boldsymbol{\eta}$. The choice of objective function is dependent on the application. Several useful choices will be presented herein to demonstrate the versatility of the SDP optimization approach.

Minimum Control Effort: To minimize power consumption and actuator wear, it is often desirable to minimize the total control effort of the spacecraft, which means minimizing the total applied torque, or equivalently, the spacecraft acceleration. To this end, we want to minimize J, where

$$\begin{aligned} J &= \int_0^{t_k} \big(\ddot{\boldsymbol{\Phi}}^{b_j i}\big)^{\mathsf{T}}\big(\ddot{\boldsymbol{\Phi}}^{b_j i}\big)\mathrm{dt}, \\ &= (\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta})^{\mathsf{T}} \underbrace{\left(\int_0^{t_k} (\mathbf{T}_j^{\alpha})^{\mathsf{T}}(\mathbf{T}_j^{\alpha})\mathrm{dt}\right)}_{\boldsymbol{\Psi}\in\mathbb{R}^{3n\times 3n}} (\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta}), \end{aligned}$$

which is quadratic in terms of the design variable. Note that, in general, $\boldsymbol{\Psi}$ is positive semidefinite and can be computed in closed form for a given maneuver duration t_k. Because $\boldsymbol{\Psi}$ is positive semidefinite, J can be rewritten as

$$J = \big(\boldsymbol{\psi}(\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta})\big)^{\mathsf{T}}\big(\boldsymbol{\psi}(\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta})\big), \tag{16}$$

where $\boldsymbol{\Psi} = \boldsymbol{\psi}^{\mathsf{T}}\boldsymbol{\psi}$. To express (16) as a linear function, we introduce the intermediate variable γ, and set

$$J = \gamma, \tag{17}$$

such that,

$$\big(\boldsymbol{\psi}(\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta})\big)^{\mathsf{T}}\big(\boldsymbol{\psi}(\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta})\big) \leqslant \gamma. \tag{18}$$

Using the Schur Complement Lemma [17], (18) can then be expressed as the LMI constraint

$$\begin{bmatrix} \gamma & \big(\boldsymbol{\psi}(\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta})\big)^{\mathsf{T}} \\ \big(\boldsymbol{\psi}(\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta})\big) & \mathbf{1} \end{bmatrix} \geqslant 0, \tag{19}$$

$$\gamma \geqslant 0. \tag{20}$$

The problem then becomes to minimize γ subject to (19) and (20).

Maximum Angular Velocity or Acceleration: Because high angular velocities or angular accelerations may result in trajectories that are difficult to track in practice, it may be desirable to limit the peak values experienced in the maneuver. To this end, we can impose one of the two following objective functions,

$$J = \omega_{\max}, \tag{21}$$

or

$$J = \dot{\omega}_{\text{max}}. \tag{22}$$

If (21) or (22) are used as an objective function, ω_{max} or $\dot{\omega}_{\text{max}}$ become optimization variables, and should be treated as such in the constraints (11) and (12). Note that by definition, (11) and (12) force ω_{max} and $\dot{\omega}_{\text{max}}$ to be greater than zero.

Spline Order: Finally, to obtain a smooth trajectory, one can choose an objective function that penalizes higher-order spline coefficients. This leads to the objective function

$$J = (\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta})^{\mathsf{T}}\mathbf{Q}(\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta}),$$

where $\mathbf{Q} = \mathbf{Q}^{\mathsf{T}} > 0$ is a positive definite weighting matrix that can be chosen to penalize the higher-order coefficients in $\boldsymbol{\beta}$. As with the minimum control effort objective function, we introduce an intermediate design variable γ, and choose (17) as the objective function, subject to the constraint

$$(\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta})^{\mathsf{T}}\mathbf{Q}(\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta}) \leqslant \gamma.$$

Again, using the Schur Complement allows us to re-express this quadratic constraint as the LMI constraint,

$$\begin{bmatrix} \gamma & (\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta})^{\mathsf{T}} \\ (\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta}) & \mathbf{Q}^{-1} \end{bmatrix} \geqslant 0, \tag{23}$$

$$\gamma \geqslant 0. \tag{24}$$

where (23) and (24) are augmented to the optimization problem.

The SDP optimization problem can then be summarized as follows. Minimize (17), (21), or (22), subject to (19) and (20), or (23) and (24) when relevant. In addition, enforce (11), (12), and (when appropriate) (7) at timesteps $t_j,\ j = 1, 2, \ldots, k$ throughout the maneuver duration. The coarseness of the timestep choice can vary, but in general, a choice of half the spacecraft control cycle is an appropriate number. Note that to enforce (7), (11), and (12), the substitution

$$\begin{aligned} \boldsymbol{\Phi}^{b_j i} &= \mathbf{T}_j^{\phi}(\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta}), \\ \dot{\boldsymbol{\Phi}}^{b_j i} &= \mathbf{T}_j^{\omega}(\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta}), \\ \ddot{\boldsymbol{\Phi}}^{b_j i} &= \mathbf{T}_j^{\alpha}(\bar{\boldsymbol{\beta}} + \mathbf{N}\boldsymbol{\eta}), \end{aligned}$$

is required to express the inequality constraints in terms of the design variable $\boldsymbol{\eta}$. Once the optimal value for $\boldsymbol{\eta}$ is computed, it can be used to back solve for the attitude, angular velocity and angular acceleration at any point throughout the maneuver.

V. Minimum-Time Maneuver

The previous section describes the computation of a fixed-time optimal slew maneuver described by a high-order spline, subject to attitude, angular velocity and angular acceleration constraints. It was observed that as the maneuver time was reduced, regardless of cost function, the spacecraft slew approached a "ramp-coast-ramp" (RCR) profile. In other words, the profile had a high-acceleration "ramp" at the start, followed by a low-acceleration "coast" period, followed by a final "ramp" period to bring the spacecraft to the desired final state.

Therefore, if the desired optimality criteria is to minimize the maneuver time, this can be accomplished by assuming a piecewise RCR angular velocity profile. Mathematically, for a maneuver duration t_k, this gives

$$\boldsymbol{\Phi}^{b_j i} = \begin{cases} \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 t_j + \boldsymbol{\beta}_2 t_j^2, & \text{for } t_j \leqslant \tau_1 \\ \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 t_j + \boldsymbol{\beta}_2 \tau_1 (2t_j - \tau_1), & \text{for } \tau_1 < t_j \leqslant \tau_2 \\ \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 t_j + \boldsymbol{\beta}_2 \tau_1 (2t_j - \tau_1) & \text{for } t_j > \tau_2 \\ +\boldsymbol{\beta}_3 (t_j - \tau_2)^2, & \end{cases}$$

$$\dot{\boldsymbol{\Phi}}^{b_j i} = \begin{cases} \boldsymbol{\beta}_1 + 2\boldsymbol{\beta}_2 t_j, & \text{for } t_j \leqslant \tau_1 \\ \boldsymbol{\beta}_1 + 2\boldsymbol{\beta}_2 \tau_1, & \text{for } \tau_1 < t_j \leqslant \tau_2 \\ \boldsymbol{\beta}_1 + 2\boldsymbol{\beta}_2 \tau_1 + 2\boldsymbol{\beta}_3 (t_j - \tau_2), & \text{for } t_j > \tau_2 \end{cases}$$

$$\ddot{\boldsymbol{\Phi}}^{b_j i} = \begin{cases} 2\boldsymbol{\beta}_2, & \text{for } t_j \leqslant \tau_1 \\ \mathbf{0}, & \text{for } \tau_1 < t_j \leqslant \tau_2 \\ 2\boldsymbol{\beta}_3, & \text{for } \tau_2 < t_j \leqslant t_k \end{cases}$$

where τ_1 and τ_2 are determined iteratively. Note that in this case we no longer enforce a continuous acceleration profile, as rapid torque changes are typically possible with reaction wheels, allowing this piecewise trajectory to be tracked in practice. By inspection, we can determine that $\boldsymbol{\beta}_0 = \boldsymbol{\Phi}^{b_0 i}$ and $\boldsymbol{\beta}_1 = \dot{\boldsymbol{\Phi}}^{b_0 i}$. Next, we rearrange the above equations for the terminal conditions at t_k.

$$\underbrace{\begin{bmatrix} \boldsymbol{\Phi}^{b_k i} - (\boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 t_k) \\ \dot{\boldsymbol{\Phi}}^{b_k i} - \boldsymbol{\beta}_1 \end{bmatrix}}_{\boldsymbol{\Phi} \in \mathbb{R}^{6\times 1}} = \underbrace{\begin{bmatrix} \tau_1 (2t_k - \tau_1)\mathbf{1} & (t_k - \tau_2)^2 \mathbf{1} \\ 2\tau_1 \mathbf{1} & 2(t_k - \tau_2)\mathbf{1} \end{bmatrix}}_{\mathbf{W} \in \mathbb{R}^{6\times 6}} \underbrace{\begin{bmatrix} \boldsymbol{\beta}_2 \\ \boldsymbol{\beta}_3 \end{bmatrix}}_{\boldsymbol{\beta} \in \mathbb{R}^{6\times 1}}, \tag{25}$$

In this case, the system of equations can be solved exactly for the unknown parameters as $\boldsymbol{\beta} = \mathbf{W}^{-1}\boldsymbol{\Phi}$, where $\mathbf{W}^{-1}$ is given by

$$\mathbf{W}^{-1} = \frac{1}{\kappa} \begin{bmatrix} 2(t_k - \tau_2)\mathbf{1} & -(t_k - \tau_2)^2 \mathbf{1} \\ -2\tau_1 \mathbf{1} & \tau_1 (2t_k - \tau_1)\mathbf{1} \end{bmatrix},$$

and,

$$\kappa = 2\tau_1 (2t_k - \tau_1)(t_k - \tau_2) - 2\tau_1 (t_k - \tau_2)^2.$$

To obtain τ_1 and τ_2, an iterative approach is required. First, predict values for them (setting $\tau_1 = \tau_2 = t_k/2$ is a good starting point). Next, compute $\boldsymbol{\beta}$ for the maneuver using (25). Following this, recompute τ_1 and τ_2 as

$$\tau_1 = \frac{\|2\boldsymbol{\beta}_2\bar{\tau}_1\|}{\dot{\omega}_{max}},$$

$$\tau_2 = t_k - \frac{\|2\boldsymbol{\beta}_3(t_k - \bar{\tau}_2)\|}{\dot{\omega}_{max}},$$

where the overbar denotes the previously computed value. Continue with this procedure until τ_1 and τ_2 converge. This drives the "ramp" portions of the slew to approximately follow the maximum allowable angular acceleration of the spacecraft. If the values converge such that $\tau_1 > \tau_2$, or $\|\boldsymbol{\beta}_1 + 2\boldsymbol{\beta}_2\tau_1\| > \omega_{max}$, then the maneuver is not feasible in the given time. After convergence, if $\tau_1 \leqslant \tau_2$ and $\|\boldsymbol{\beta}_1 + 2\boldsymbol{\beta}_2\tau_1\| < \omega_{max}$, then the maneuver is feasible in this time, but it is not the minimum-time solution. However, if $\tau_1 \leqslant \tau_2$ and $\|\boldsymbol{\beta}_1 + 2\boldsymbol{\beta}_2\tau_1\| \approx \omega_{max}$, then the maneuver is possible, and this is approximately the minimum-time solution. Note that for short-duration maneuvers where the spacecraft does not have enough time to ramp up to its maximum angular velocity, an exception can occur. In these cases, the iterative procedure converges to $\tau_1 \approx \tau_2$, $\|\boldsymbol{\beta}_1 + 2\boldsymbol{\beta}_2\tau_1\| < \omega_{max}$, and $\|2\boldsymbol{\beta}_2\| \approx \|2\boldsymbol{\beta}_3\| \approx \dot{\omega}_{max}$. This still constitutes a minimum-time solution, but the "coast" period is removed. Note that further iteration is required to determine the minimum maneuver time. This can be done by a coarse solution space search, or using a structured search (such as a bisection method) to converge to the minimum time.

Note that in this calculation, the angular velocity and angular acceleration constraints are implicit in the formulation. The attitude constraint can be also be implemented by solving this problem iteratively. First, the approximately time-optimal solution is computed. Next, the point of closest approach between the payload vector and the constraint vector is computed. If this point violates the attitude constraint, the closest-approach attitude is "pushed" out of the avoidance cone along the shortest rotation angle. An angular velocity is computed that causes the payload vector to move in a direction tangent to the constraint cone at the new attitude. This new attitude and angular velocity are the added as a waypoint in the maneuver. Thus, the full maneuver becomes a minimum-time RCR maneuver to the waypoint, followed by a minimum-time RCR maneuver to the terminal state. Multiple waypoints can then be added to ensure constraint violations do not occur at any point throughout the maneuver. This approach is shown in Figure 3.

VI. Simulation Results

With the problem formulation established, we now revisit the motivational example introduced in Section I involving a hypothetical DEFIANT bus mission. In this scenario, a controlled formation of RF-geolocation-capable DEFIANT spacecraft orbit the Earth in a 500 km Sun-synchronous orbit, with a 12:30 local time of descending node. A final DEFIANT spacecraft equipped with an optical payload trails this formation by 1000 km.

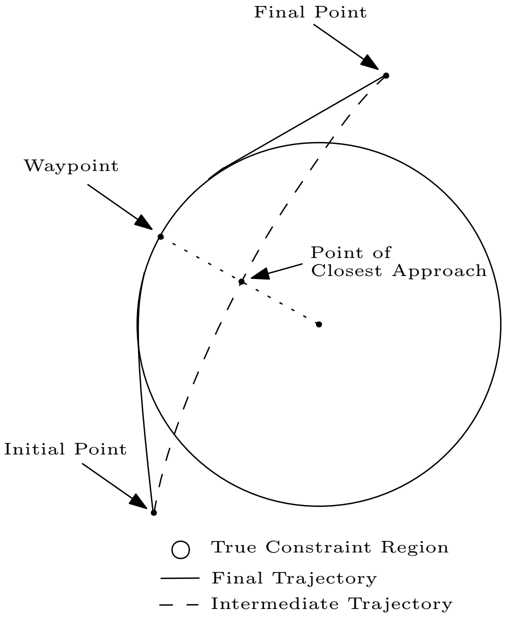


Fig. 3: Planar representation of attitude constraint application for the iterative closed-form minimum-time maneuver.

The goal of the mission is for the leading formation to geolocate the source of RF emissions and cue follow-up observation of targets of interest. When targets are cued for observation, their geographic coordinates are sent to the trailing spacecraft via an intersatellite communication system, with no ground contact. The trailing satellite must then reorient itself and capture a burst of images around the target area. The number of images in this scenario is set to four, in a rectangular pattern, as shown in Figure 4. In order to limit image smear, and to ensure the proper target area is captured, each image must be taken with the imager boresight (which is aligned with the +Y axis) fixed to the stationary ground target with angular rate error below 0.1°/s, and an angular error below 0.5°. Moreover, for operation of secondary payloads, the spacecraft -Z axis is constrained to the velocity direction when imaging.

The simulation starts with the spacecraft in a maximum power generation attitude with its -X axis nearly aligned with the Sun vector. The spacecraft also has a light-sensitive payload aligned with the body vector $\mathbf{v}^p_{b_0} = \begin{bmatrix} -0.59 & -0.64 & 0.49 \end{bmatrix}$. This payload vector must point at least 45° away from the Sun vector, and 90° from the nadir vector (for Earth limb avoidance). At $t = t_0$, a target is cued for imaging, and the spacecraft must slew from its initial Sun-pointing attitude to its imaging attitude in under two minutes, which is the time it takes for the target area to pass at nadir. This initial scenario is shown in Figure 4.

Due to the timing requirement for the slew and the lack of ground contact, the path-planning calculation in this case must be done on board with no operator intervention. Moreover, the slew must be done while avoiding the Earth and Sun avoidance cones, and respecting the angular velocity and angular acceleration limits set to 2.25°/s and $0.14°/\text{s}^2$ respectively.

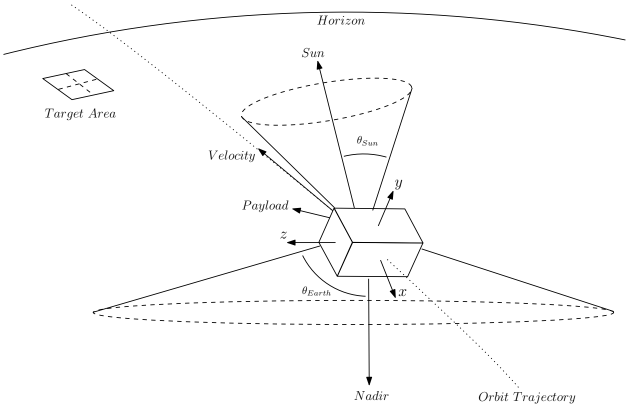


Fig. 4: Initial spacecraft attitude with Sun and Earth avoidance cones shown. The spacecraft must reorient itself such that the +Y axis points toward the target area with the -Z axis constrained to spacecraft velocity.

To summarize, the problem is to compute trajectories that take the spacecraft from its initial state to its four imaging states (four total slew calculations), while respecting constraints on its attitude, angular velocity and angular acceleration. The calculation involves tracking of targets that are stationary in the ECEF frame, and avoidance of constraints defined in the ECI and LVLH frames with a payload vector defined in the body frame. All of this must be taken into account in the path-planning algorithm.

In addition, this maneuvers must be performed such that at the terminal state, the control error is below the aforementioned requirement, so that the target can be imaged immediately with no settling time. The simulation was performed in UTIAS-SFL's high-fidelity simulation environment, which runs in parallel with the on-board flight code to ensure that results are representative of on-orbit performance. Numerical optimization was done using YALMIP and MOSEK in MATLAB.

Note that the complexity of this maneuver is such that it cannot be performed using the current "rest-to-rest" assumption. The formulation does not allow for inclusion of attitude constraints, and the additional settling time prohibits imaging the four targets in a single pass. Therefore, use of the dynamic-endpoint spline or RCR maneuver can enable new mission concepts with demanding operational requirements, such as this one.

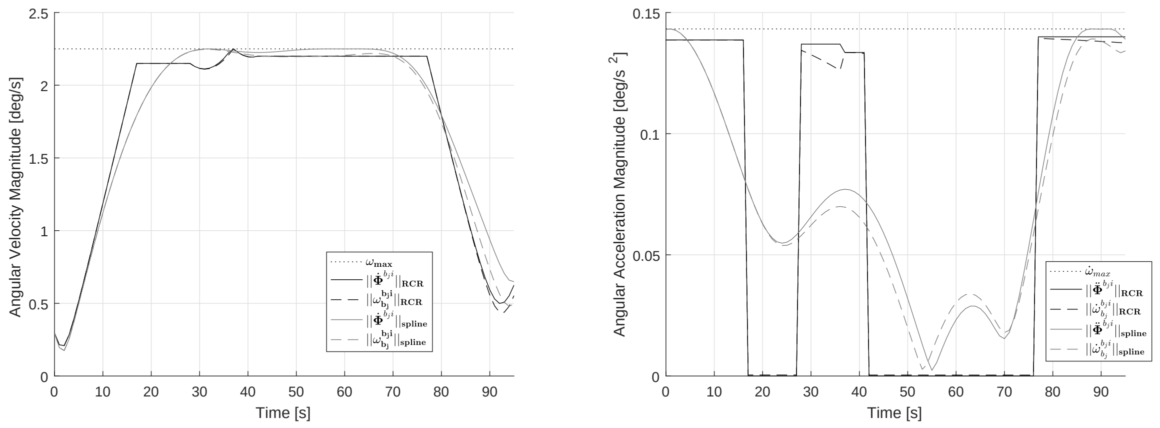
In the simulation, when using the minimum-time formulation, it was found that the initial slew could be done in 96 s, well below the two-minute requirement. The optimal spline trajectory with $n = 20$ was also computed for the 96 s slew using the minimum control effort objective function. The angular velocity and angular acceleration profiles for both the minimum-time RCR maneuver and the optimal spline maneuver are shown in Figure 5.

As can be seen in Figure 5, angular velocity and acceleration constraints are respected throughout both maneuvers, even with the approximations inherent in constraints (11) and (12). Moreover, the magnitude of the angular velocity and acceleration is very closely approximated by the magnitude of the derivatives of the rotation vector. At worst, the approximation represents a slightly conservative upper bound when values approach the constraint boundary. Note that one waypoint was computed for the RCR maneuver (reached at 37 s), and though the angular velocity profile is piecewise linear in each axis, it's magnitude is in general not linear.

As can also be seen, both the RCR maneuver and the optimal spline maneuver have similar angular velocity magnitude profiles, but the spline maneuver has a more smooth torque profile. This is due to the nature of the spline interpolation, as well as the use of the minimum control effort objective function.

Figure 6 shows the orientation of the payload vector in the ECI frame throughout all four slew maneuvers for both the RCR and optimal spline maneuvers. The payload vector trajectory is also shown for a slew where the attitude constraint is not applied. The initial orientation is at 82° azimuth and -30° elevation, where azimuth is defined as the rotation about the ECI +Z axis from the +X axis, and elevation is defined as the angle from the X-Y plane toward the +Z axis. When the attitude constraint is not applied, the payload trajectory passes through the Sun avoidance cone. However, application of the constraint yields a trajectory that avoids the Sun cone throughout the entire maneuver. Note that the Earth limb avoidance cone is not shown, as no violations were observed for this constraint.

Finally, shown in Figure 7 is the spacecraft control error throughout all four slews and imaging bursts. The control error and stability requirements were met throughout the entire simulation without exception, demonstrating that these rapid slew trajectories can be tracked with current attitude control capabilities.



(a) **(b)**

Fig. 5: (a) Magnitude of spacecraft angular velocity throughout the first slew maneuver. (b) Magnitude of spacecraft angular acceleration throughout the first slew maneuver.

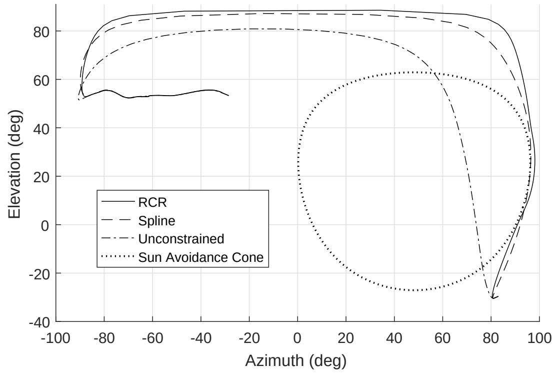


Fig. 6: Payload Trajectory.

Given that the proposed approach is for real-time implementation, a discussion of run time is appropriate. As mentioned, LMI-constrained SDP optimization problems can be solved extremely efficiently in practice. Moreover, they scale very well as the number of constraints and optimization variables increases. In fact, [18] discusses how, in practice, the computational intensity of the problem scales as $O(m^\alpha L^\beta)$, where $\alpha \approx 2.1$, $\beta \approx 1.2$, m is the dimension of the optimization variable, and L is the number of LMI constraints on that variable. Thus, the problem scales approximately quadratically with the variable size, and approximately linearly with the number of constraints. With the spline interpolation formulation, the number of optimization variables is set for a given spline order. Therefore, as the maneuver duration increases, the computation time tends to increase linearly. This is far more efficient for long-duration slews than the techniques proposed in [6], [7], [8], where both the number of variables and the number of constraints must increase. In addition, it allows engineers to limit the spline to an order that is appropriate for the computational capabilities of the spacecraft in question.

In the author's experience, a spline order of $n = 20$ or less was sufficient for most application, which, for slew durations of approximately 100 s, resulted in run times on the order of a few seconds for implementation in MATLAB using YALMIP and MOSEK on a typical desktop computer. This run time is shorter (on the order of a second or less) if the attitude constraint does not need to be enforced, as the run time increases linearly with the number of iterations required, and implementing this constraint introduces iteration. It is difficult to directly compare this run time with on-board implementation, as the computational power of small satellites may be more limited. However, utilization of a custom interior-point solver in a more-efficient programming language could offset the increased run time due to limited computational capability. The viability of on-board implementation is therefore dependent on the specific application, but the desktop computer implementation demonstrates that the run times are not infeasible for real-time application.

In cases where numerical optimization is not possible in practice, the minimum-time RCR solution can be used, as it still allows for constraint satisfaction with a iterative closed-form solution that can be more-quickly computed in real time. Run times for this formulation tend to be on the order of milliseconds as opposed to seconds.

The implications of this work are twofold. Firstly, it demonstrates that large-angle slews with non-rest endpoints that incorporate a variety of dynamic and kinematic constraints can be planned and tracked in practice, enabling a wider range of autonomous operation. Secondly, it demonstrates that rapid slews between subsequent targets are possible with no settling time, allowing for immediate imaging after slew completion. This increases the number of images possible in a given pass, giving a larger observation area, and a greater data return for the mission.

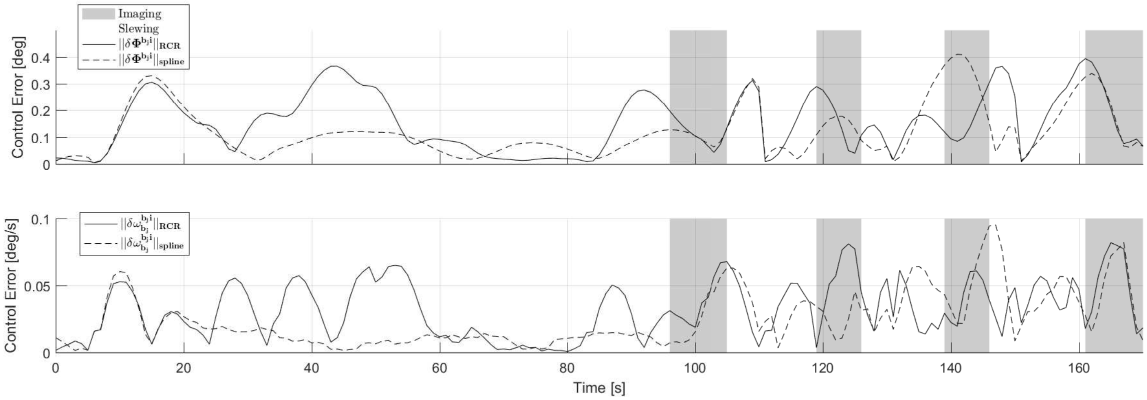


Fig. 7: Simulated control error throughout planned slew maneuvers and imaging bursts.

Neither of these capabilities are possible with the current "rest-to-rest" slew assumption. The proposed trajectory generation algorithm will therefore be implemented in upcoming UTIAS-SFL missions to extend the capabilities of the DEFIANT platform, and enable new mission concepts for UTIAS-SFL spacecraft in general.

VII. Conclusion

This paper proposes a slew maneuver path-planning algorithm suitable for real-time implementation on board small spacecraft, that can include a variety of dynamic and kinematic constraints. The algorithm uses high-order spline interpolation solved as a convex SDP optimization problem using interior-point methods. For applications that are more computationally limited, or require minimum-time slews, an iterative closed-form solution is presented that incorporates common constraints into its formulation. Both formulations are detailed, and their utility is explored in a hypothetical mission involving UTIAS-SFL's DEFIANT platform. Simulation results show that the the computed trajectories can be tracked in practice, and timing results are encouraging for real-time on-board implementation.

VIII. Acknowledgments

The author would like to thank Dr. Robert E. Zee for his incisive feedback, and for providing the opportunity to study at a the Space Flight Laboratory. It has been a truly exceptional experience that would not be possible without his support. The author would also like to acknowledge his colleagues at UTIAS-SFL. In particular, Niels Roth whose guidance was crucial to the development of this work, as well as Robert Magner for his advice and comments.

References

[1] E. Frazzoli, M. A. Dahleh, E. Feron, R. P. Kornfeld, and R. P. Kornfeld, "A randomized attitude slew planning algorithm for autonomous spacecraft," in *In AIAA Guidance, Navigation, and Control Conference*, 2001.

[2] C. R. McInnes, "Large angle slew maneuvers with autonomous sun vector avoidance," *Journal of guidance, control, and dynamics*, vol. 17, no. 4, pp. 875–877, 1994.

[3] R. Kornfeld, "On-board autonomous attitude maneuver planning for planetary spacecraft using genetic algorithms," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, p. 5784, 2003.

[4] Y. Somov, "Guidance, navigation and control of information satellites: Methods for modeling, synthesis and nonlinear analysis," *Journal— MESA*, vol. 7, no. 2, pp. 223–248, 2016.

[5] S. Tanygin, "Attitude interpolation," *Advances in the Astronautical Sciences*, vol. 114, pp. 1353–1369, 2003.

[6] Y. Kim and M. Mesbahi, "Quadratically constrained attitude control via semidefinite programming," *IEEE Transactions on Automatic Control*, vol. 49, no. 5, pp. 731–735, 2004.

[7] I. Okoloko and A. Basson, "Consensus with collision avoidance: An LMI approach," in *Proc. 5th IEEE International Conference on Automation, Robotics and Applications*, pp. 84–89, 2011.

[8] C. Sun and R. Dai, "Spacecraft attitude control under constrained zones via quadratically constrained quadratic programming," in *AIAA Guidance, Navigation, and Control Conference*, p. 2010, 2015.

[9] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.

[10] "SFL Satellite Platforms." https://www.utias-sfl.net/?page_id=89. Accessed: 2019-04-16.

[11] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[12] P. C. Hughes, *Spacecraft attitude dynamics*. Dover, 1986.

[13] A. H. De Ruiter, C. Damaren, and J. R. Forbes, *Spacecraft dynamics and control: an introduction*. John Wiley & Sons, 2012.

[14] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).*, 2015.

[15] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, (Taipei, Taiwan), pp. 284–289, IEEE, 2-4 Sept. 2004.

[16] A. Walsh, J. R. Forbes, S. A. Chee, and J. J. Ryan, "Kalman-filter-based unconstrained and constrained extremum-seeking guidance on SO(3)," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 9, pp. 2260–2271, 2017.

[17] J. G. VanAntwerp and R. D. Braatz, "A tutorial on linear and bilinear matrix inequalities," *Journal of Process Control*, vol. 10, pp. 363–385, Aug. 2000.

[18] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1994.