A MOTION PLANNING METHOD FOR SPACECRAFT ATTITUDE MANEUVERS USING SINGLE POLYNOMIALS

Conference Paper · August 2015

CITATIONS
7

READS
907

2 authors:

Albert Caubet
University of Strathclyde
8 PUBLICATIONS **31** CITATIONS
SEE PROFILE

James Biggs
Politecnico di Milano
117 PUBLICATIONS **853** CITATIONS
SEE PROFILE

AAS 15-627

A MOTION PLANNING METHOD FOR SPACECRAFT ATTITUDE MANEUVERS USING SINGLE POLYNOMIALS

Albert Caubet[*], **James D. Biggs**[†]

A motion planning technique for generating smooth attitude slew maneuvers is presented, which can generate suboptimal feasible trajectories with low computational cost in the presence of constraints. The attitude coordinates are shaped by time-dependent polynomials, whose coefficients are determined by matching prescribed arbitrary boundary conditions. Quaternions are used as the reference attitude parametrization for arbitrary maneuvers, which require normalization of the four independently shaped coordinates. In the case of spin-to-spin maneuvers, a particular combination of Euler Angles are used. The torque profile is evaluated using inverse dynamics, which allows the feasibility of the maneuver given the actuator constraints to be checked. With this approach, a root-finding method is used to select the minimum time for a certain path. By increasing the degree of the polynomial free coefficients are introduced, thus pointing constraints can be accommodated and time can be optimized amongst this class of motion. This motion planning method is applied to a flexible spacecraft model, demonstrating its effectiveness at reducing spillover vibrations.

INTRODUCTION

Motion planning refers to the problem of defining a feasible motion subject to constraints.[1] Additionally, the motion can be optimized according to a specified cost function. Polynomial motion planning is widely used in robotics and computer graphics, for their efficiency and ease of manipulation. Cubic splines are a popular method for connecting path points, however, the acceleration is not smooth.[2] Additionally, obstacle avoidance usually requires the computation of path points, which has been applied to spacecraft maneuvers in[3–5] with various degrees of computational expense. In spacecraft slew maneuvers, motion planning can be used to optimize a certain performance parameter and/or satisfy differential and path constraints, before the trajectory is executed. Path or pointing constraints are exclusion areas over the unit sphere which a certain body-fixed axis shall not enter, for instance the boresight of some sensitive instruments must be kept at a minimum angle from the Sun direction. Differential constraints include dynamic and kinematic constraints. They generally require an internal model to calculate variables such as torque, in what is known as an inverse dynamics approach. For instance, in a planned maneuver it must be ensured that torque is not larger than the actuators' limit at any point. Apart from torque, reaction wheels experience saturation given by the wheels speed limit, which can also be assessed. Sensors such as star trackers can have a maximum operational angular speed, and flexible structures may require limits on acceleration and jerk to avoid excessive deflection and

[*]PhD candidate, Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow G1 1XJ, UK
[†]Associate Director, Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow G1 1XJ, UK

vibration. In this paper, *motion* or *trajectory* planning is used to design the time evolution of the attitude coordinates, while *path* planning refers to the representation of the attitude in a time-independent space, such as the curves traced by the motion projected onto the unit sphere.

In previous work on spacecraft attitude motion planning, McInnes[6] first shows the potential of applying inverse dynamics to attitude maneuvers using polynomials and Euler angles, while Biggs[7] finds reference motions for reorienting a spinning satellite by solving analytically an optimal control problem, and Zhang[8] uses a 5^{th} degree polynomial to obtain a smooth eigenaxis rotation on a flexible spacecraft. In the field of computer graphics, Kim[9] proposes the use of exponential coordinates to satisfy the unit norm constraint of quaternions, with Tanygin[10] and Boyarko[11] applying this approach to spacecraft maneuvers using inverse dynamics to minimize time and/or energy amongst this class of curves. Following from the work by Boyarko,[11] Ventura[12] compares the performance (computational time and optimization cost) of different attitude parameterizations using splines for motion planning, and shows that the exponential functions in the quaternion parameterization require a larger computational expense. The motion planning method proposed in this paper uses quaternions individually shaped by smooth polynomials, which are then normalized. Quaternions are chosen since they are the preferred attitude coordinates for spacecraft applications due to their efficiency and lack of singularities. Additionally, a special axis-azimuth parameterization[13] is proposed for spin-to-spin maneuvers, in which the spinning axis is re-pointed, due to their suitability to this specific problem.

Regarding optimization, time minimization amongst the class of polynomial trajectories is discussed. Other performance metrics are possible, such as energy (based on the accumulated square root of torque) or fuel consumption. However, the planned trajectories are smooth, being more suited to reaction wheels than they are to a reaction control system. With reaction wheels, fuel or energy optimization is not relevant (provided that the spacecraft has enough power output to drive the wheels). Bilimoria and Wie[14] show that the time-optimal maneuver has a bang-bang torque profile, which is not achievable with the smooth torques provided by polynomial motion planning (although it can be close enough with high order polynomials). Therefore, with this method the minimum time will be selected within the set of feasible polynomial trajectories, which is limited by the maximum torque. Interestingly, Junkins[15] proves that for a single-axis rotational maneuver, the energy-optimal trajectory is a polynomial, which suggests that this function family may naturally provide energy-efficient maneuvers. In this paper, the time optimization strategy employs a combination of root-finding and unconstrained optimization, aiming to minimize computational cost and ensure convergence, as opposed to directly using nonlinear programming solvers such as in[11] and.[12] Pointing constraints are also addressed, in combination with time optimization. In some works such as Frazzoli[16] and Tanygin,[17] pointing constraints are satisfied using randomized path planning algorithms,[1] and then the time-independent path is followed with a controller. In contrast, trajectory planning (as shaping the attitude parameters with functions of time) addresses both path constraints and dynamic constraints by re-shaping the trajectory. Randomized path planning performs better at finding a feasible geometric path in highly constrained spaces, but a priori it does not consider the dynamic and kinematic aspects of the motion as trajectory planning does.

The smoothness of the resulting trajectories, along with the ability to monitor accelera-

tion and jerk, make this method particularly suited to spacecraft with flexible appendages. Discontinuities in the torque profile such as in bang-bang maneuvers, or the initial step input given by feedback controllers, result in infinite jerk leading to the excitation of flexible modes and spillover (post-maneuver vibrations).[18,19] Singh,[20] Kim,[21] and Byers[22] propose solutions by smoothing the discontinuous torque switches of bang-bang maneuvers with a variety of functions. The method in this paper aims to reduce vibrations with a guidance approach rather than with a feedback control approach, by providing continuously smooth torque along the trajectory—specially at the endpoints. In Singh[23] an input-shaped control is used to minimize the tip deflection of appendages and spillover vibrations. The torque profile obtained with the input-shaped method is made of discrete jumps (although not bang-bang), but an interpolation of that discrete torque profile would result in a smooth, sinusoidal-like shape. Interestingly, such a shape is similar to the torque profiles obtained with inverse dynamics in this polynomial trajectory planning.

In motion planning, there is a trade-off between the pursuit of global optimality and computational efficiency. The focus of this work is on computationally efficient motion planning strategies, providing suboptimal results while respecting the constraints imposed on the attitude motion. The first section describes the method, detailing quaternion normalization, the spin-to-spin case, and discussing the potential numerical issues associated with high-order polynomials. The second section explores strategies to generate feasible trajectories in terms of torque and path constraints, while finding the minimum possible maneuver time. The final section applies the method to simulations with a flexible spacecraft, focusing on the effects of smoothness regarding vibrations.

MOTION PLANNING METHOD OUTLINE

The proposed method represents the attitude of the body with a prescribed analytically defined function of time. In this paper polynomial functions were chosen since they are smooth, and easy to derive and manipulate. Polynomials are parameterized to match prescribed boundary conditions on attitude, velocity, and higher order derivatives. Once the desired attitude trajectory has been obtained, the torque profile can be obtained with inverse dynamics. Quaternions are used in this paper to parameterize attitude, as they are non-singular and computationally efficient.

The trajectory of each quaternion in the S^3 unit sphere is shaped by the rational polynomial function

$$q_i(t) = \frac{q_i^*(t)}{\|\mathbf{q}^*(t)\|} \tag{1}$$

for $i = 1, ..., 4$, where $q_i^*(t)$ is a polynomial:

$$q_i^*(t) = a_{i0} + a_{i1}t + a_{i2}t^2 + ... + a_{in}t^n = \sum_{j=0}^{n} a_{ij}t^j \tag{2}$$

Since these quaternions (depicted by the * supercript) are individually shaped, they form a vector in $\mathbb{R}^4$ whose norm is not constant, thus each component i is normalized in Eq. (1) using the quaternion unit norm

$$\|\mathbf{q}^*(t)\| = \sqrt{q_1^*(t)^2 + q_2^*(t)^2 + q_3^*(t)^2 + q_4^*(t)^2} \tag{3}$$

Arbitrary boundary conditions can be matched for any maneuver time t_f. As the attitude boundary conditions are normalized (i.e. $q_i(0) = q_i^*(0)$ and $q_i(t_f) = q_i^*(t_f)$) it is sufficient to consider Eq. (2) to match the boundary conditions. This results in solving a simple linear equation to find the value of the polynomial coefficients in Eq. (2). The $m = n+1$ boundary conditions of the maneuver determine the degree of the polynomial and provide a system of linear equations from which the coefficients a_{ij} can be obtained, given a final maneuver time. The minimum number of boundary conditions that define a slew maneuver are the initial and final attitude and velocity (requiring a 3^{rd} degree polynomial to define the quaternions' trajectories). Additionally, acceleration boundary conditions can be introduced, with the purpose of having zero torque at the trajectory endpoints. This may be required for flexible spacecraft, to avoid vibration-inducing discontinuities in angular acceleration. For the same reason, the boundary jerk (time derivative of acceleration) can be forced to zero so that it is continuous at the endpoints.

The degree of the polynomial can be increased beyond $n = m-1$ (i.e. the minimum needed for matching boundary conditions), which introduces degrees of freedom to the system in the form of the additional coefficients. The extended polynomial of degree $m - 1 + k$, for m boundary conditions and k additional terms, becomes

$$q_i^*(t) = a_{i0} + a_{i1}t + a_{i2}t^2 + ... + a_{i,m-1}t^{m-1} + ... + a_{i,m-1+k}t^{m-1+k} \tag{4}$$

In this paper, a scenario with $m = 8$ boundary conditions is considered, matching attitude, velocity, acceleration, and jerk. The 8 coefficients can thus be solved by a polynomial of degree 7 representing $q_i^*(t)$. However, we can increase the order of the polynomial with an additional term ($k = 1$ in Eq. (4)). Assuming that the additional free coefficient $a_{i,8}$ is guessed or known, the rest of the coefficients that make the trajectory match the boundary conditions are determined by the following linear system of equations:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 & t_f^6 & t_f^7 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 & 6t_f^5 & 7t_f^6 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 & 30t_f^4 & 42t_f^5 \\ 0 & 0 & 0 & 6 & 24t_f & 60t_f^2 & 120t_f^3 & 210t_f^4 \end{bmatrix} \begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \\ a_{i6} \\ a_{i7} \end{bmatrix} = \begin{bmatrix} q_i(0) \\ \dot{q}_i(0) \\ \ddot{q}_i(0) \\ \dddot{q}_i(0) \\ q_i(t_f) - a_{i,8}t_f^8 \\ \dot{q}_i(t_f) - 8a_{i,8}t_f^7 \\ \ddot{q}_i(t_f) - 56a_{i,8}t_f^6 \\ \dddot{q}_i(t_f) - 336a_{i,8}t_f^5 \end{bmatrix} \tag{5}$$

where the vector on the right-hand side contains the selected boundary conditions. The coefficients a_{ij} of the i-th quaternion are then a function of the maneuver final time t_f and the corresponding boundary conditions.

The values of the additional coefficients can be selected in an optimization process, and Eq. (5) is solved. In other words, the optimizer re-shapes the trajectory by adjusting the additional coefficients, while maintaining the endpoints at the specified boundary conditions. Alternatively, a more "deterministic" approach can be considered, where k trajectory waypoints (quaternions) are selected adding k equations to the system. In this case, besides matching the boundary conditions, the trajectory will pass through the specified attitudes at the specified times. At the waypoints, given the use of a single polynomial between endpoints, the curve is smooth (i.e. of differentiability class C^∞).

While Eq. (5) can be solved with linear algebra methods, it is more efficient to calculate the coefficients using closed-form expressions (which can be rapidly obtained with a symbolic mathematics software). Note that each polynomial degree has a different set of expressions.

The initial and final values of the quaternions' time derivatives in Eq. (5) can be obtained, given the boundary angular velocities, via the kinematics equation:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{6}$$

Furthermore, the boundary values of $\ddot{q}_i$ and $\dddot{q}_i$ are obtained by differentiating Eq. (6) with respect to time. When selecting the endpoint attitudes, note that quaternions are not unique in the sense that the same attitude in the $SO(3)$ space can be represented both by $\mathbf{q}$ and $-\mathbf{q}$. However, a trajectory shaped between $q_i(0)$ and $q_i(t_f)$ (t_f being the maneuver final time) is different than one connecting $q_i(0)$ with $-q_i(t_f)$. This results in a winding trajectory, where the desired attitude is reached through a long path. To avoid this phenomenon, the sign of the quaternions should be selected according to a metric based on the difference between $\mathbf{q}(0)$ and $\pm\mathbf{q}(T_f)$. Specifically, if $\mathbf{q}_d$ is the difference between the endpoint attitudes, expressed in quaternion algebra as

$$\mathbf{q}_d = \mathbf{q}_f \cdot \mathbf{q}_0^{-1} \tag{7}$$

the corresponding rotation angle $\theta_d = 2\cos^{-1}(q_{d4})$ should be less than 180 deg in order to avoid a winding trajectory.

Spin-to-spin maneuvers

In the special case of spin-to-spin maneuvers, only the final pointing of a certain body axis is relevant. Since quaternions define the full attitude (i.e. the three body axes), the final spin phase angle must be determined, which adds another degree of freedom to the problem that must be chosen. In order to avoid this, the direction of the pointing axis can be parameterized with two spherical coordinates such as azimuth and declination angles (s_1 and s_2), which can be expressed independently as time polynomials. These two coordinates form a reduced attitude parameterization,[24] while the phase angle of the other two orthogonal vectors around the pointing axis is not specified. A third parameter (s_3), describing the rotation angle about the pointing axis, completes the full attitude in what is known as axis-azimuth parameterization.[13] The s_3 is also expressed as a polynomial, however, its final value s_{f3} is not included in the boundary conditions set since it is irrelevant. This strategy is particularly convenient for spin-to-spin maneuvers, i.e. transferring the spacecraft from one pointing direction and spinning state $\dot{\mathbf{s}}_0 = [0\,0\,\dot{s}_{03}]^T$ to another one with final $\dot{\mathbf{s}}_f = [0\,0\,\dot{s}_{f3}]^T$. In fact, the resulting attitude rotation formalism formed by s_1, s_2, and s_3 is a particular combination of *intrinsic Euler angles*.

For instance, assume a spacecraft with an instrument aligned with the body axis y, which is required to point in different directions. In this case, the attitude can be described as a $z-x'-y''$ Euler rotation, where the rotation matrix is $R = R_z(s_1)R_x(s_2)R_y(s_3)$. The order of the x and z rotations is not relevant, but the third rotation must be around the pointing

axis (y in this case). With 8 boundary conditions, the polynomials are of degree 7:

$$s_i(t) = a_{i0} + a_{i1}t + a_{i2}t^2 + ... + a_{i7}t^7 \tag{8}$$

For $i = 1,\ 2$. However, for $i = 3$, there are only 7 boundary conditions, thus $a_{37} = 0$. The coefficients are obtained by solving a linear system analogous to Eq. (5).

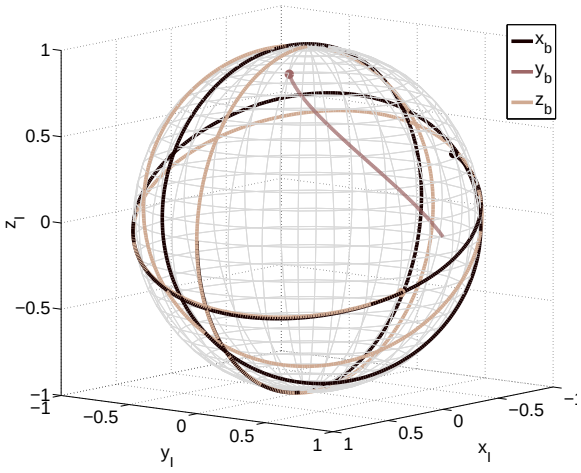

Figure 1. Path of the body axes on the unit sphere, in a spin-to-spin maneuver

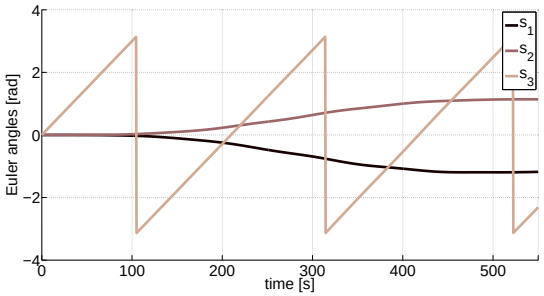

Figure 2. Trajectory of the attitude coordinates, in a spin-to-spin maneuver

Figure 1 shows the body axes paths of a spin-to-spin maneuver in an inertial frame, where the pointing axis y_b precesses towards the target direction (depicted by a point at the end of the path line) while the other two orthogonal axes keep rotating about it. In Figure 2, the two coordinates defining the direction of the pointing axis (s_1 and s_2) are driven to their final desired values, while s_3 follows a constant rate trajectory (since the prescribed initial and final spin rates are the same) where the final value of the angle is not relevant.

The singularity associated with Euler angles occurs when calculating their time derivatives with the kinematics equation at angles of 90 deg. However, in this case the kinematic equation is not used, since the time derivatives of the angles are obtained by differentiating the polynomial. Also, no singularities arise when evaluating the angular velocities and accelerations (needed to calculate the torque). Finally, the trajectory can be converted to

quaternions if required by the attitude control system of the spacecraft, at the cost of having to use trigonometric functions.

Numerical stability of high degree polynomials

High degree polynomials may have sensitivity issues, where small errors in the inputs cause relatively large errors in the outputs. In the attitude control scenario, the main source of error comes from sensor inaccuracies, namely the current attitude and velocity values that are inputs to the linear system. Mathematically speaking, the matrix of the linear system in Eq. (5), expressed in the form $Ax = b$, can be ill-conditioned for too high polynomial orders and t_f values. A matrix is ill-conditioned if it is close to being singular, therefore non-invertible. A metric of the conditioning of a matrix is its condition number C, or the ratio of the largest to smallest singular value in the singular value decomposition. The order of magnitude of C gives an estimate of the digits of accuracy lost in solving a linear system with that matrix.

In order to reduce the condition number of the matrix, the time domain can be scaled so that the final time is 1, to prevent some elements in A from being too large. For instance, with the new variable $\tau \in [0, 1]$, where $\tau = t/t_f$, the condition number is reduced from 10^{18} (for $m = 8$ and $t_f = 300$ sec) to 10^4. The scaled coefficients can be calculated by solving the corresponding linear system, but the vector of boundary values changes due to the differentiation with respect to a scaled variable. The differential operator with respect to time can be expressed as

$$\frac{d}{dt} = \frac{d}{dt}(\tau)\frac{d}{d\tau} \tag{9}$$

by replacing $\tau = t/t_f$, Eq. (9) becomes

$$\frac{d}{dt} = \frac{1}{t_f}\frac{d}{d\tau} \tag{10}$$

which can be raised to the k-th derivative. Thus, the corresponding k-th time derivatives of the scaled quaternions are calculated as

$$\frac{d^k q_i(\tau)}{d\tau^k} = t_f^k \cdot \frac{d^k q_i(t)}{dt^k} \tag{11}$$

and the boundary values in the right-hand side vector of Eq. (5) must be adjusted accordingly (while in the matrix, $t_f = 1$). The scaled coefficients $\overline{a}_{in}$ of the polynomial in τ are related to the original ones by

$$a_{in} = \frac{\overline{a}_{in}}{t_f^n} \tag{12}$$

MINIMUM TIME MANEUVERS IN A CONSTRAINED SPACE

The maneuver time t_f and any free coefficients of the polynomial must be selected in order to calculate the rest of the coefficients, defining a trajectory matching the boundary conditions. Those variables can be selected in an optimization process, so that the minimum t_f is chosen amongst the set of feasible polynomial trajectories. In this section, the constraints considered are torque, which is evaluated from the quaternions and their derivatives with an internal model (inverse dynamics), and pointing keep-out areas. While

this is a constrained optimization problem, the use of sequential quadratic programming (SQP) algorithms has been avoided. Instead, in a more efficient approach, a combination root-finding and unconstrained optimization has been used.

Inverse dynamics

A simple model of a fully actuated rigid body has been used to obtain the torque profile. The Euler's equation of rigid-body dynamics relates the torque u_i (along the body i-th axis) to the angular velocity ω_i and acceleration $\dot{\omega}_i$ and principal moments of inertia I_i, as

$$\begin{aligned} u_1 &= I_1\dot{\omega}_1 - (I_2 - I_3)\omega_2\omega_3 \\ u_2 &= I_2\dot{\omega}_2 - (I_3 - I_1)\omega_1\omega_3 \\ u_3 &= I_3\dot{\omega}_3 - (I_1 - I_2)\omega_1\omega_2 \end{aligned} \tag{13}$$

The angular velocities and accelerations are related to quaternions and their time derivatives through the rotational kinematics,[25] as

$$\begin{aligned} \omega_1 &= 2(\dot{q}_1 q_4 + \dot{q}_2 q_3 - \dot{q}_3 q_2 - \dot{q}_4 q_1) \\ \omega_2 &= 2(\dot{q}_2 q_4 + \dot{q}_3 q_1 - \dot{q}_1 q_3 - \dot{q}_4 q_2) \\ \omega_3 &= 2(\dot{q}_3 q_4 + \dot{q}_1 q_2 - \dot{q}_2 q_1 - \dot{q}_4 q_3) \end{aligned} \tag{14}$$

$$\begin{aligned} \dot{\omega}_1 &= 2(\ddot{q}_1 q_4 + \ddot{q}_2 q_3 - \ddot{q}_3 q_2 - \ddot{q}_4 q_1) \\ \dot{\omega}_2 &= 2(\ddot{q}_2 q_4 + \ddot{q}_3 q_1 - \ddot{q}_1 q_3 - \ddot{q}_4 q_2) \\ \dot{\omega}_3 &= 2(\ddot{q}_3 q_4 + \ddot{q}_1 q_2 - \ddot{q}_2 q_1 - \ddot{q}_4 q_3) \end{aligned} \tag{15}$$

Analytical expressions for the quaternion derivatives are obtained by differentiating Eq.(1) with respect to time:

$$\dot{q}_i(t) = \frac{\dot{q}_i^*(t)}{\|\mathbf{q}^*(t)\|} - \frac{q_i^*(t)}{\|\mathbf{q}^*(t)\|^3}\left(\sum_{i=1}^{4} q_i^*(t)\dot{q}_i^*(t)\right) \tag{16a}$$

$$\begin{aligned} \ddot{q}_i(t) = \frac{\ddot{q}_i^*(t)}{\|\mathbf{q}^*(t)\|} - \frac{\dot{q}_i^*(t)}{\|\mathbf{q}^*(t)\|^3}\left(\sum_{i=1}^{4} q_i^*(t)\dot{q}_i^*(t)\right) + \\ q_i^*(t)\left[\frac{3}{\|\mathbf{q}^*(t)\|^5}\left(\sum_{i=1}^{4} q_i^*(t)\dot{q}_i^*(t)\right)^2 - \frac{1}{\|\mathbf{q}^*(t)\|^3}\left(\sum_{i=1}^{4}(\dot{q}_i^*(t)^2 + q_i^*(t)\ddot{q}_i^*(t))\right)\right] \end{aligned} \tag{16b}$$

Since the quaternions in Eqs. (14) and (15) can be replaced by their corresponding time-dependent polynomials (Eq. (1) and their derivatives), whose coefficients are a function of the maneuver time t_f, ultimately the torque is a function of t and t_f (given a set of boundary conditions for a particular maneuver). Evaluating the torque along the trajectory is essential to ensure that the actuators always remain within their operative limits. Similarly, an additional time differentiation of Eq. (15) allows for the evaluation of jerk along the maneuver.

If the actuators are reaction wheels, it can be useful to assess the speed buildup during the maneuver, to ensure that they will not become saturated. Assuming that the wheels are aligned with the body axes, the planned torque can be related to the derivative of their angular momentum. The body angular velocity is considered negligible compared to the

magnitude of typical wheels' speeds. Therefore the wheel's acceleration and moment of inertia can be related to the torque provided along its axis by

$$\mathbf{u} \approx -\mathbf{I}_W \dot{\boldsymbol{\omega}}_W \tag{17}$$

Where the vector $\dot{\boldsymbol{\omega}}_W$ contains the wheels' angular acceleration and $\mathbf{I}_W = \text{diag}(I_{W1},\, I_{W2},\, I_{W3})$ is their inertia matrix. The wheel speeds are obtained by replacing Eq. (13) into Eq. (17) and integrating:

$$\boldsymbol{\omega}_W(t) \approx -\frac{1}{\mathbf{I}_W}\left(\mathbf{I}(\boldsymbol{\omega}(t) - \boldsymbol{\omega}(0)) + \int_0^t \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega})dt\right) + \boldsymbol{\omega}_W(0) \tag{18}$$

where the angular velocity of the body $\boldsymbol{\omega}$ can in turn be replaced by Eq. (14). While the integral in Eq. (18) has a closed-form solution as a function of time and the polynomial coefficients, it is so complex that it is computationally more efficient to evaluate it numerically.

Maneuver time minimization

The final time t_f is required to evaluate the coefficients in Eq. (5). In this section the goal of finding a minimum t_f is addressed, which requires the evaluation of the torque. The maneuver duration affects the torque profile, with shorter final times resulting in higher torques. A criterion for choosing t_f is to find its minimum value such that the calculated maximum torque in the maneuver (of any axis, in absolute value) is equal to the actuator's torque limit u_{lim}. In an analogous way, other differential constraints can be considered, such as a limit on velocity, acceleration, jerk, or the reaction wheels rate.

It is possible to obtain an expression of the planned torque as a function of time and t_f by combining the dynamic equations and the polynomials representing the attitude parameters, as mentioned above. However, due to the high non-linearity of this expression, finding the minimum t_f with a purely analytical approach is not practical. A more efficient strategy consists in discretizing the trajectory and evaluating the torques at each node ($u_i(t_k)$ for the i-th axis and k-th node). The difference with the torque limit u_{lim} is calculated for each node and the maximum value of the set is obtained:

$$J_i = \max_k(|u_i(t_k)| - u_{lim}) \tag{19}$$

where J_i is the largest difference amongst all nodes of the i-th torque profile. The three axes can be combined in $J = \max\{J_1,\, J_2,\, J_3\}$.

As shown in Figure 3, the optimum point corresponds to $J = 0$, whereas if $J > 0$, the maximum torque is above the limit. While gradient-based optimization algorithms would use J^2 to find the optima, this performance index allows to (a) know if a trajectory is feasible (in terms of torque) simply by checking the sign of J, and (b) use a root-finding algorithm, which are more efficient than numerical gradient-based ones. For general maneuvers with arbitrary boundary angular velocities, accelerations, and jerks, the shape of J as a function of t_f may feature multiple local optima—some of which could be unfeasible. Thus, an algorithm capable of finding the global optimum would be required to compute the minimum feasible t_f. However, in the cases of rest-to-rest and spin-to-spin maneuvers, the evolution

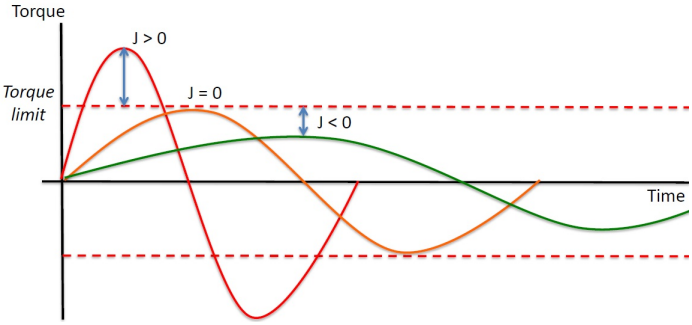

Figure 3. Illustration of torque profiles with different final times, with J being a metric of the peak height

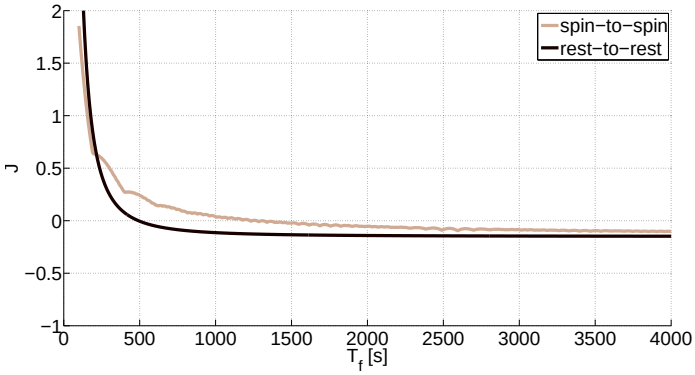

Figure 4. Evolution of J (maximum torque over the limit) with maneuver time

of J with t_f is monotonically decreasing, as shown in the example of Figure 4. In this case, finding the root of this curve, corresponding to the minimum t_f of that particular path, is performed with very few iterations.

However, other paths (still matching the prescribed boundary values) may have a lower final time, which can be explored by polynomials with additional terms. In this case, free variables are added to the system, whose value can be selected by an optimizer. Best results are obtained with a two-layer optimization approach. The outer layer explores the space of $4\times k$ free coefficients (considering 4 quaternions, shaped with polynomials of degree $m+k$), using an unconstrained optimization method with maneuver time as the performance index. At every iteration of the optimizer, the root-finding algorithm runs as the inner loop, giving the minimum feasible time for that specific trajectory. This approach is robust in the sense that, even if the optimizer converges to a local optimum, the trajectory will satisfy torque constraints. While there exists a global minimum time within the set of trajectories given by polynomials of a certain degree, this value can be reduced if higher degrees are used. The absolute global optimum of the problem could be that given by an optimal control computational method (although optimality is not guaranteed either).

Obstacle avoidance

With the additional degrees of freedom provided by free variables in the polynomial, the path of the body axes on the unit sphere can be diverted in order to avoid pointing constraints. The static path constraints[3] or keep-out areas are represented by cones intersecting the unit sphere. The resulting circle should not be trespassed by the path of the corresponding body axis i, in other words, the angle between the body axis $\mathbf{v}_i^I(t)$ (resolved in the inertial frame) and the cone axis $\mathbf{w}_c$ should not be lower than the cone angle γ_c:

$$\mathbf{v}_i^I(t) \cdot \mathbf{w}_c \leq \cos(\gamma_c) \tag{20}$$

The pointing body axis can be drawn from quaternions with

$$\mathbf{v}_i^I = (q_4^2 - \|\vec{q}\|^2)\mathbf{v}_i^B + 2(\vec{q}^T\mathbf{v}_i^B)\vec{q} + 2q_4(\vec{q} \times \mathbf{v}_i^B) \tag{21}$$

where $\vec{q} = [q_1,\, q_2,\, q_3]^T$ and $\mathbf{v}_i^B$ is the pointing axis resolved in the body frame.

Obstacle avoidance is carried out using unconstrained optimization of the free parameters. Constraints can be considered in unconstrained optimization with a performance index known as penalty function. The penalty function is applied to the keep-out area:

$$J_{OA} = \begin{cases} \max_k(\mathbf{v}_i^I(t_k) \cdot \mathbf{w}_c - \cos(\gamma_c)) & \text{if } \mathbf{v}_i^I(t) \cdot \mathbf{w}_c > \cos(\gamma_c); \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

where the trajectory is discretized, with $\mathbf{v}_i^I(t_k)$ depicting the k-th node. Essentially, the penalty function in Eq. (22) is related to the closest distance to the center of the cone (as far as the resolution of the discretization allows) when the trajectory path crosses it. During the optimization, the path "slides down" out of the keep-out area. When the path of the axis is tangent or anywhere outside the keep-out area, the penalty is constant at zero—deactivating the obstacle avoidance process.

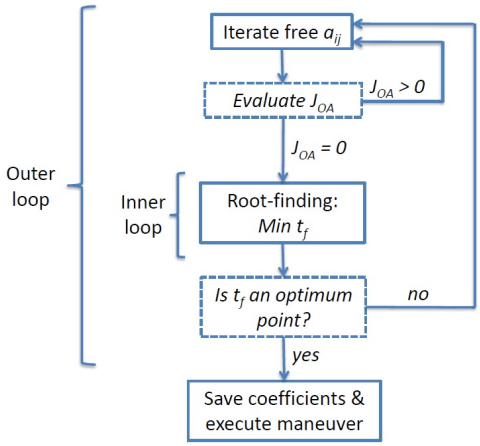

Figure 5. Two-layer time optimization algorithm

The obstacle avoidance penalty function can be included in the two-layer optimization for finding the minimum time described in the previous section. As soon as the path crosses

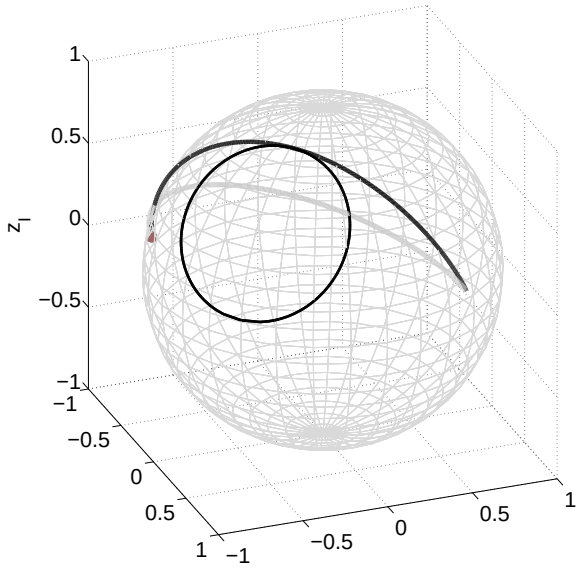

Figure 6. Paths of the pointing body axis on the unit sphere and keep-out circle, for the nominal trajectory (grey line) and the diverted one (black line)

a keep-out area, the obstacle avoidance algorithm is activated and time minimization is not considered, until the penalty index J_{OA} becomes zero. The flow chart of the algorithm of time optimization with pointing constraints is shown in Figure 5. The torque constraints are embedded in the inner loop (the root-finding process described in the previous section), which are used to find the minimum t_f for every iterated combination of free coefficients. This t_f becomes the performance index if obstacle avoidance is not activated ($J_{OA} = 0$). An interesting property of rest-to-rest maneuvers, for some given values of free coefficients, is that the axes' paths remain unchanged when varying the maneuver time t_f. Therefore in the obstacle avoidance part of the algorithm, the final time can be set to $t_f = 1$.

Figure 6 shows the path of the pointing axis (black line), using a single additional term on the polynomials ($k = 1$ in Eq. (4))—therefore, four free variables are included. The grey line shows an initial-guess trajectory (minimum polynomial degree for matching boundary conditions) which does not satisfy the path constraint. While the obstacle avoidance algorithm forces the path out of the cone, in this case the one featuring the minimum time happens to cross the keep-out area, making the optimizer converge to a path tangent to the circle (i.e. the minimum-time trajectory satisfying the pointing constraint).

Alternatively, obstacle avoidance can be achieved in a deterministic way using a path point (thus adding an additional equation to the linear system in Eq. (5)). The point is selected by calculating the nearest point of the nominal path to the cone center and moving it to the closest point on the circle. This implies rotating the attitude about the axis $\mathbf{w}_c \times \mathbf{v}_i^I(\tau_k)$ by an angle $\gamma_c - \gamma$. Since time is adimensional, the τ_k associated with the new point is the same as the original one. The resulting path of the pointing axis is tangent to the keep-out cone. This approach is suboptimal, but as it is deterministic it avoids the unconstrained optimization process and a feasible motion is obtained.

Note that on maneuvers with arbitrary endpoint velocities, the variable t_f affects the path, thus the time-minimisation and the obstacle avoidance problems are coupled. The two-layer approach can still be used, but in this case the actual t_f must be considered in the outer layer (where path constraints are checked). However, in the case of spin-to-spin using the previously described parameterization, the path of the pointing axis is not dependent on t_f, and an analogous approach to the rest-to-rest scenario can be used.

SIMULATIONS ON A FLEXIBLE SPACECRAFT

So-called flexible spacecraft are non-rigid bodies, which may have flexible appendages and/or multiple parts with non-rigid links. Flexibility introduces additional challenges to attitude maneuvering due to oscillations of the appendages and a varying inertia matrix. To study how the proposed method performs in this scenario, simulations on a flexible spacecraft with reaction wheels were realized. The spacecraft has been modeled as a multi-body object, formed by a central cuboid with two solar panels, joint together by hinges (torsional spring + damper). The model used in the simulations is detailed in Shahriari.[26] As opposed to modeling a spacecraft with Bernoulli beams,[15] which require assumed modes of vibration, a multi-body only needs the stiffness and damping rate of the joints to be specified. Also, as shown in Shahriari,[26] it is shown to be more representative of the three-dimensional case when compared to a finite elements model.

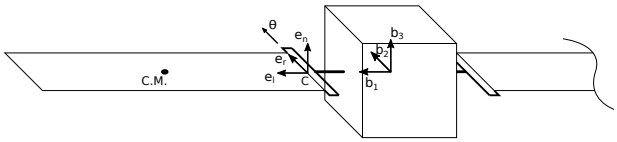

Figure 7. Spacecraft with hinged symmetric panels

Once the polynomial coefficients are obtained for a particular maneuver in the motion planning process, the desired state and predicted torque are computed at each time step. Due to the torque being planned with a simplified, rigid-body internal model, if inputted in open loop there will be an error in the attitude profile if flexible structures or other major disturbances are involved. Therefore, a proportional-derivative (PD) quaternion feedback controller[25] was implemented, which tracks the desired attitude and angular velocity. Since the planned torque profile is available, the controller is augmented by including it as a feed-forward command. The control law is

$$\mathbf{u}_{cmd} = -K_p \mathbf{q}_e - K_d \boldsymbol{\omega}_e + \mathbf{u}_{pln} \tag{23}$$

where K_p and K_d are the proportional and derivative coefficients, q_e is the error quaternion, $\boldsymbol{\omega}_e$ is the error angular velocity, and $\mathbf{u}_{pln}$ is the torque as obtained in the planning.

The simulations have been realized considering a 2000-kg spacecraft, with a body inertia of $\mathbf{I}_b = \text{diag}(310, 310, 310)$ kgm^2 and 3×1 m solar panels with a mass of 40 kg. The hinges linking the panels with the body have a stiffness of 12 N m rad^{-1} and a damping of 0.01 kg m^2 s^{-1} rad^{-1}. The simulated maneuvers have been performed considering reaction wheels limited to 0.2 Nm.

Figure 8 shows the quaternions of a rest-to-rest maneuver, while Figure 9 shows the torque profile. Solid lines are the actual states, whereas dashed lines represent the planned

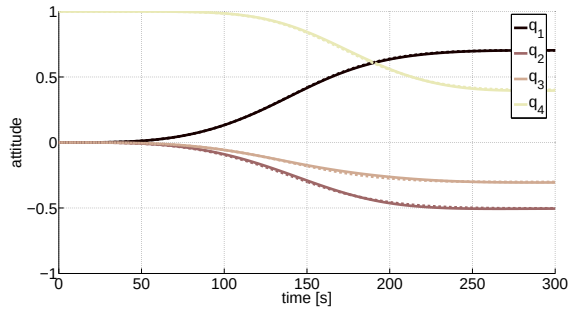

Figure 8. Quaternions trajectory

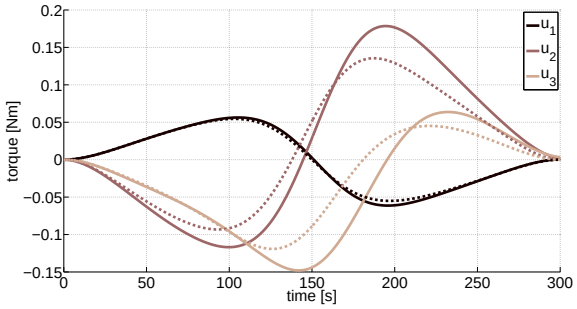

Figure 9. Torque profile

trajectory. Note that whereas quaternions are tracked with precision, the actual torque (as given by the controller) is higher than the planned profile due to disturbance rejection caused by large flexibility. This result emphasizes the need for applying an appropriate safety margin during the motion planning phase, to avoid the actual torque from reaching the actuators' limit.

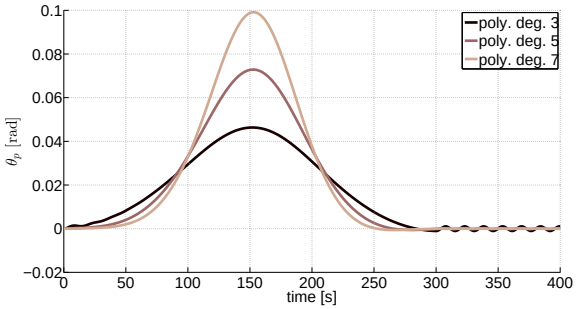

Figure 10. Solar panel deflection angle

Figures 10 and 11 show, for the rest-to-rest case, the evolution of the deflection angle of the solar panels at the hinges θ_p, which relates to the panels' tip displacement, and its derivative $\dot{\theta}_p$. Polynomials of different degrees were tested. Zero jerk at the boundaries or endpoints of the trajectory can be achieved by a trajectory shaped with a polynomial of degree 7, zero acceleration can be achieved by a polynomial of degree 5, and a polynomial

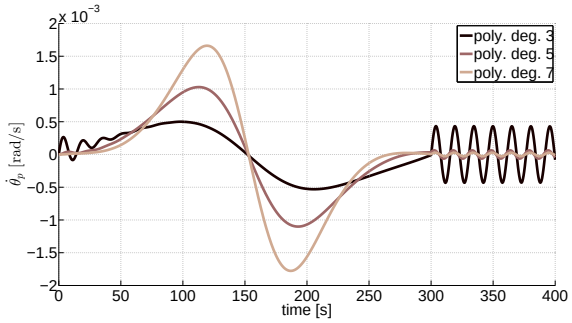

Figure 11. Time derivative of the panel deflection angle

of degree 3 suffices to enforce zero velocity. The initial and final torques planned with a 3^{rd} degree polynomial are not zero, thus the jerk is infinite causing short-period vibration modes (which is better appreciated in $\dot{\theta}_p$). However, the feedback control tracking the trajectory acts as a damper of those initial oscillations, that would remain if open-loop control was used. When the torque is cut off the spacecraft is left with residual vibrations on the spacecraft, an issue known as *spillover*. This situation is significantly improved with a polynomial of degree 5. The initial and final jerk, while not zero, are limited, significantly reducing the initial vibration and the spillover. With this level of stiffness at the hinges, the initial vibrations are barely noticeable—and quickly damped by the feedback control. With a 7^{th} degree polynomial enforcing zero jerk, virtually no initial short-period vibrations are generated. A small yet significantly reduced spillover can still be observed in the simulations, caused by the offset between the planned torque and the actual control torque.

Taking the aforementioned into consideration, choosing the right polynomial for shaping the trajectory will depend mainly on the specific mission requirements and constraints. Since the maneuver time is fixed, lower degree polynomials plan trajectories with lower acceleration and torque peaks, leading to smaller maximum tip displacement. Thus, given the same torque limit, lower degree polynomials can plan the same maneuver with a smaller final time, at the expense of being more aggressive at the endpoints. The smoothest yet longest maneuvers are achieved using a polynomial of degree 7, whereas time can be slightly reduced by using a trajectory allowing non-zero initial jerk as represented by a 6^{th} degree polynomial (the closed-loop feedback controller will damp the initial vibrations). It is advised to use a very smooth ending torque to prevent spillover, as well as to implement trajectory tracking rather than open-loop control—to both minimize the attitude error and damping vibrations. If there is a constraint on some appendage's maximum tip displacement, a limit on maximum acceleration can be considered so that a higher t_f may be chosen, even if the torque is well below the actuators' limit. Finally, with this method, the design of the spacecraft structure can be allowed to be less stiff, and still prevent large oscillations.

CONCLUSIONS

An inverse dynamics trajectory planning method using polynomials has been presented. The polynomials shape the quaternions of an attitude maneuver by matching arbitrary boundary conditions on attitude, velocity, and higher derivatives (acceleration, jerk) if required. The torque profile of the actuators can be assessed, allowing torque limits to be

satisfied. By increasing the degree of the polynomial there are more coefficients than those required to match boundary conditions. Those extra free coefficients can be used as optimization variables to further minimize time (or any other cost function) and deal with pointing constraints. A combination of root-finding and unconstrained optimization using few variables (four free coefficients plus the final time, in the examples of this study) has been proposed, avoiding the more computationally expensive nonlinear programming solvers. With this approach, suboptimal but feasible trajectories are found in a constrained space at low computational cost. The limits of this method arise in highly constrained spaces, where there may not be enough degrees of freedom to find a feasible trajectory. Apart from trajectory constraints such as torque limits and keep-out areas, the ability to generate smooth motions and enforce boundary conditions on acceleration and jerk has proven relevant to flexible spacecraft, specially for the reduction of spillover vibrations.

REFERENCES

[1] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.

[2] Y. Guan, K. Yokoi, and O. Stasse, "On Robotic Trajectory Planning Using Polynomial Interpolations," *IEEE International Conference on Robotics and Biomimetics*, 2005, pp. 111–116, 10.1109.

[3] Y. Kim, M. Mesbahi, G. Singh, and F. Y. Hadaegh, "On the constrained attitude control problem," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004, 10.2514/6.2004-5129.

[4] X. Cheng, H. Cui, P. Cui, and R. Xu, "Large angular autonomous attitude maneuver of deep spacecraft using pseudospectral method," *ISSCAA2010 - 3rd International Symposium on Systems and Control in Aeronautics and Astronautics*, 2010, pp. 1510–1514, 10.1109/ISSCAA.2010.5632498.

[5] H. C. Kjellberg and E. G. Lightsey, "A Constrained Attitude Control Module for Small Satellites," *Proceedings of the AIAA/USU Conference on Small Satellites*, SSC12-XII-1, 2014.

[6] C. R. McInnes, "Satellite attitude slew manoeuvres using inverse control," *The Aeronautical Journal*, 1998, pp. 259–265.

[7] J. D. Biggs and N. Horri, "Optimal geometric motion planning for a spin-stabilized spacecraft," *Systems and Control Letters*, Vol. 61, No. 4, 2012, pp. 609–616, 10.1016/j.sysconle.2012.02.002.

[8] Y. Zhang and J.-R. Zhang, "Combined control of fast attitude maneuver and stabilization for large complex spacecraft," *Acta Mechanica Sinica*, Vol. 29, No. 6, 2013, pp. 875–882, 10.1007/s10409-013-0080-8.

[9] M.-J. Kim, M.-S. Kim, and S. Y. Shin, "A general construction scheme for unit quaternion curves with simple high order derivatives," *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 369–376, 10.1145/218380.218486.

[10] S. Tanygin, "Parametric optimization of closed-loop slew control using interpolation polynomials," *AAS/AIAA Space Flight Mechanics Meeting*, 2006, pp. 1–21.

[11] G. Boyarko, M. Romano, and O. Yakimenko, "Time-Optimal Reorientation of a Spacecraft Using an Inverse Dynamics Optimization Method," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 4, 2011, pp. 1197–1208, 10.2514/1.49449.

[12] J. Ventura, M. Romano, and U. Walter, "Performance evaluation of the inverse dynamics method for optimal spacecraft reorientation," *Acta Astronautica*, Vol. 110, 2015, pp. 266–278, 10.1016/j.actaastro.2014.11.041.

[13] M. D. Shuster, "A Survey of Attitude Representations," *The Journal of the Astronautical Sciences*, Vol. 41, No. 4, 1993, pp. 439–517, 10.2514/6.2012-4422.

[14] K. D. Bilimoria and B. Wie, "Time-optimal three-axis reorientation of a rigid spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 3, 1993, pp. 446–452, 10.2514/3.21030.

[15] J. L. Junkins and J. D. Turner, *Optimal spacecraft rotational maneuvers*. New York: Elsevier Scientific, 1985.

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

[17] S. Tanygin, "Fast Three-Axis Constrained Attitude Pathfinding and Visualization Using Minimum Distortion Parameterizations," *Journal of Guidance, Control, and Dynamics*, 2015, pp. 1–13, 10.2514/1.G000974.

[18] G. Singh, P. T. Kabamba, and N. H. Mcclamrochj, "Bang-bang control of flexible spacecraft slewing maneuvers: Guaranteed terminal pointing accuracy," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 2, 1989, pp. 376–379, 10.2514/3.56512.

[19] S. B. Skaar and L. Tang, "On-Off attitude control of flexible satellites," *Journal of Guidance, Control, and Dynamics*, Vol. 9, No. 4, 1986, pp. 507–510, 10.2514/3.20140.

[20] T. Singh, "Jerk limited input shapers," *Proceedings of the American Control Conference*, Vol. 5, No. March, 2004, pp. 4825–4830, 10.1109/ACC.2004.182716.

[21] J.-J. Kim and B. N. Agrawal, "Experiments on Jerk-Limited Slew Maneuvers of a Flexible Spacecraft," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, pp. 1–20.

[22] R. M. Byers, S. R. Vadali, and J. L. Junkins, "Near-minimum time, closed-loop slewing of flexible spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 1, 1990, pp. 57–65, 10.2514/3.20517.

[23] T. Singh and S. R. Vadali, "Input-shaped control of three-dimensional maneuvers of flexible spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 6, 1993, pp. 1061–1068, 10.2514/3.21128.

[24] N. Chaturvedi, A. Sanyal, and N. McClamroch, "Rigid-Body Attitude Control," *IEEE Control Systems*, Vol. 31, No. 3, 2011, pp. 30–51, 10.1109/MCS.2011.940459.

[25] B. Wie, *Space Vehicle Dynamics and Control, Second Edition*. Reston, Virginia: AIAA, 2008.

[26] S. Shahriari, S. Azadi, and M. M. Moghaddam, "An accurate and simple model for flexible satellites for three-dimensional studies," *Journal of Mechanical Science and Technology*, Vol. 24, No. 6, 2010, pp. 1319–1327, 10.1007/s12206-010-0329-0.