

Optimal Attitude Motion Planner for Large Slew Maneuvers Using a Shape-Based Method

Albert Caubet¹, James D. Biggs²
University of Strathclyde, Glasgow, United Kingdom

Small satellites, in particular Cubesats, have limited torque and power available, thus optimization of slew maneuvers may be required. However, their on-board computational resources are usually low. In this paper a computationally light optimal path planning algorithm is presented, based on a shape-based method. This technique will enable 3-axis stabilized satellites to perform torque-optimal slew maneuvers with low computational overhead, thus especially suited to small satellites. The shape-based method presented relies on a quaternion representation of the trajectory, whose components are assumed to be polynomial functions. A rationale to select the degree of the polynomials is presented. The function's coefficients are then adjusted according to the maneuver boundary conditions, and by parametric optimization of the free coefficients, the torque is minimized. Once the trajectory is planned, a simple PD controller can track it. The results are validated using a pseudo-spectral optimal control solver, and a two-stage approach is discussed. This work also demonstrates the capacity of the shape-based method to deal with pointing constraints and multi-objective torque-time optimization.

I. Introduction

Nano-satellites and specifically Cubesats have very limited resources to perform their tasks. Regarding the attitude control system, Cubesats have low power, basic on-board processors and a small space inside to fit relatively large 3-axis stabilization devices. Most Cubesats are equipped with magnetic attitude control systems, simple and light but only allowing them to detumble and perform some coarse pointing stabilization. Recently, micro reaction wheels have been designed to fit in Cubesats, manufactured and flown. Such an attitude control system, if accompanied by high-performance sensors, may enable a whole new range of applications, historically placed in the realm of multi-ton spacecraft. Examples are celestial pointing, inter-spacecraft optical communication, Earth imaging or space debris monitoring, to name a few [1]. However, micro-reaction wheels may not provide enough torque, or they may be too power-consuming. Thus, repointing motions may have to be optimized in order to allow very small wheels to perform large maneuvers without draining all the power or saturating the system. In other words, if very small reaction wheels are capable of complying with the mission requirements, spacecraft designers can save precious weight and power budget. However, implementing a complex optimal motion planner on-board requires a certain amount of computational capability, which is limited as well in small satellites.

The method proposed in this paper tries to deal with this challenge using a shape-based approach, which enables fast optimization procedures. In essence, the shape-based method consists of approximating the trajectory with a function. In this case, a polynomial of a certain order has been selected as the most suitable for optimization purposes. Their coefficients are solved to meet the boundary conditions of the slew maneuver, and iterated to optimize the trajectory in terms of minimizing the accumulated torque. As opposed to applying pure numerical optimizers (as in [2]), the main point is to take advantage of analytical approaches to reduce computational cost. Similarly, in [3] a method based on trigonometric shapes of angular velocities is applied to a spin-stabilized spacecraft, but trajectories do not match arbitrary boundary conditions.

The shape-based method has been previously used to obtain low-thrust interplanetary trajectories [4-6], assisting numerical optimizers in broad searches over the design-space or providing them with good estimates. The method we present was inspired by this approach, and we applied it to the attitude control problem (extending the work of

¹ Ph.D. candidate, Advanced Space Concepts Laboratory, Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow, G1 1XJ, UK, albert.caubet@strath.ac.uk.

² Associate Director, Advanced Space Concepts Laboratory, Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow, G1 1XJ, UK, james.biggs@strath.ac.uk.

McInnes [7]). This attitude shape-based method can find a feasible and close-to-optimal trajectory on its own, without having to rely on more complex optimizers to refine the trajectory. However, a pseudo-spectral optimal control solver [8] has been used for validation purposes and to select the most appropriate family of function. Furthermore, some tests have been performed in order to assess the benefits of initializing the optimal control solver with a trajectory obtained by the shape-based method.

II. Rigid-body model

The attitude kinematics and dynamics of a rigid body are used to model the motion of the spacecraft [9]. The rotational kinematics is represented in quaternion form:

$$\dot{\mathbf{q}} = \frac{1}{2} \widehat{\Omega} \mathbf{q} \quad (1)$$

Where $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$ with q_4 as the scalar element. The skew-symmetric matrix $\widehat{\Omega}$ depends on the angular velocities and characterizes a 3-2-1 rotation sequence:

$$\widehat{\Omega} = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \quad (2)$$

Where $\omega_1, \omega_2, \omega_3$ are the angular velocities in body-fixed coordinates. The Euler equations with external torques $\mathbf{u} = [u_1, u_2, u_3]^T$ are used to model the attitude dynamics of a spacecraft, assuming continuous control with actuators aligned with the body's principal axes:

$$\left. \begin{aligned} \dot{\omega}_1 &= \delta_1 \omega_2 \omega_3 + \frac{u_1}{I_1} \\ \dot{\omega}_2 &= \delta_2 \omega_1 \omega_3 + \frac{u_2}{I_2} \\ \dot{\omega}_3 &= \delta_3 \omega_2 \omega_1 + \frac{u_3}{I_3} \end{aligned} \right\} \quad (3)$$

With $\delta_1 = \frac{I_2 - I_3}{I_1}$, $\delta_2 = \frac{I_3 - I_1}{I_2}$, $\delta_3 = \frac{I_1 - I_2}{I_3}$ where I_1, I_2, I_3 are the principal moments of inertia of the spacecraft. The angular velocities can be written in terms of quaternions [9]:

$$\begin{aligned} \omega_1 &= 2(\dot{q}_1 q_4 + \dot{q}_2 q_3 + \dot{q}_3 q_2 + \dot{q}_4 q_1) \\ \omega_2 &= 2(\dot{q}_2 q_4 + \dot{q}_3 q_1 + \dot{q}_1 q_3 + \dot{q}_4 q_2) \\ \omega_3 &= 2(\dot{q}_3 q_4 + \dot{q}_1 q_2 + \dot{q}_2 q_1 + \dot{q}_4 q_3) \end{aligned} \quad (4)$$

Additionally, it is straightforward to express the angular accelerations in terms of time derivatives of the quaternions.

III. Shape-based approach using polynomials

In order to explore which family of functions best represent the type of optimal trajectories of our problem, first, a number of maneuvers were obtained using an optimal control solver (q.v. section V.C). Second, those optimal trajectories were discretized and approximated using a data fitting tool [10], which uses nonlinear least squares to quantify how well a function fits a set of points. After considering a variety of functions, the polynomial family was chosen, as it ensures smooth trajectories that, according to our function fitting of extensive optimization tests, mimic well the optimal motions. Moreover, polynomials are easy to derive and manipulate, and are computationally efficient.

In the strategy presented here, each quaternion of the vector part is approximated by a time-dependent polynomial whose coefficients define the quaternion's trajectory path.

The criteria to define the minimum degree of the polynomials are given by the number of boundary conditions of the trajectory. For a given final maneuver time, if there are N boundary conditions the polynomials shall be at least of degree $N-1$. Thus, the N unknown coefficients of the polynomial can be easily found by solving a linear system of equations (q.v. section III.A). This linear system is exactly determined and has a unique solution.

In the case of choosing a polynomial of degree $M > N-1$, the system of equations of boundary conditions becomes underdetermined, since the number of unknown coefficients is larger than the number of equations. A priori the coefficients cannot be solved; therefore an optimization process is introduced (q.v. section III.B). We differentiate both cases as exact and optimized shape-based methods.

A. Exact shape-based method

The necessary boundary conditions (apart from final time) are the initial and final attitudes, and angular velocities (the slew maneuvers considered in this paper are rest-to-rest, so endpoint angular rates are zero). So, with those four boundary conditions we need, at least, a third-order polynomial with four coefficients:

$$q_i = a_i + b_i t + c_i t^2 + d_i t^3 \quad (5)$$

(For $i = 1,2,3$.) These polynomials represent the path for every quaternion (of the vector part), and they comply with the boundary conditions independently. For a given maneuver final time T_f , endpoint attitudes (q_{0i}, q_{fi}), and endpoint angular velocities ($\dot{q}_{0i}, \dot{q}_{fi}$), the boundary conditions form a simple linear system (Eq. (6)). By solving it, the polynomial's coefficients are obtained.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & T_f & T_f^2 & T_f^3 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2T_f & 3T_f^2 \end{bmatrix} \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \end{bmatrix} = \begin{bmatrix} q_{0i} \\ q_{fi} \\ \dot{q}_{0i} = 0 \\ \dot{q}_{fi} = 0 \end{bmatrix} \quad (6)$$

(For $i = 1,2,3$.) Finally, the quaternion scalar part q_4 is found using the quadratic condition of quaternions:

$$q_4 = \pm \sqrt{1 - (q_1^2 + q_2^2 + q_3^2)} \quad (7)$$

Note that the sign of q_4 along the trajectory shall be selected according to the signs of its initial and final value.

Thus, by selecting T_f , a smooth and close-to-optimal trajectory can be computed almost instantaneously. In order to autonomously choose the maneuver final time, an optimization algorithm can be used. This path planning method is computationally very efficient, as a smooth trajectory is generated analytically. However, torque shall be estimated in order to check the torque constraint (as described in section IV.). If, for a certain generated trajectory, the maximum capacity of the actuators was to be surpassed or we want to minimize torque, another trajectory with a longer maneuver time shall be computed. However, if the mission is constrained to really low computational overhead, one solution could be to build a look-up table or function

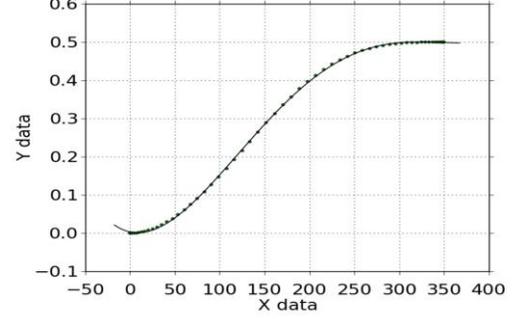


Figure 1. 3rd degree polynomial fits a discretized optimal trajectory. X-axis: time, Y-axis: attitude (q_1)

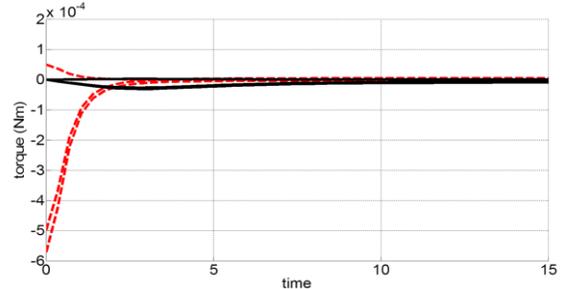


Figure 2. Torque comparison of a problematic maneuver (detail of first seconds). Red-dotted: 3rd degree poly., black: 5th degree poly.

assigning maneuver times to the angular distance of the desired maneuver.

A natural extension to the procedure previously described is to add two more boundary conditions: initial and final acceleration shall be zero. Thus, the polynomial for such a case is of 5th degree. This is motivated by the fact that some maneuvers, computed using the 3rd degree polynomial, induce bursts of torque near the endpoints that may surpass the maximum torque or power available. Those cases are rare, but there is a non-zero probability of them happening, even for long maneuver times, as shown in Fig. 2.

Therefore, by introducing end-point zero acceleration, smoother motions are enforced. This effectively reduces torque peaks, so it is useful not only for nano-spacecraft with limited resources but also for large flexible structures. The trajectory shape and the boundary conditions system is then as follows (with $i = 1,2,3$):

$$q_i = a_i + b_i t + c_i t^2 + d_i t^3 + f_i t^4 + g_i t^5 \quad (8)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & T_f & T_f^2 & T_f^3 & T_f^4 & T_f^5 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2T_f & 3T_f^2 & 4T_f^3 & 5T_f^4 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6T_f & 12T_f^2 & 20T_f^3 \end{bmatrix} \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \\ f_i \\ g_i \end{bmatrix} = \begin{bmatrix} q_{0i} \\ q_{fi} \\ \dot{q}_{0i} = 0 \\ \dot{q}_{fi} = 0 \\ \ddot{q}_{0i} = 0 \\ \ddot{q}_{fi} = 0 \end{bmatrix} \quad (9)$$

B. Optimized shape-based method

The 3rd or 5th degree polynomials are not versatile shapes, since their only degree of freedom is the maneuver time. In order to add more flexibility, so that torque can be further optimized, the degree of the polynomial is increased. So, considering a maneuver with four boundary conditions (initial and final attitudes and velocities), if the polynomial shaping the trajectory is of 4th degree the system becomes underdetermined. To solve it, the higher order coefficient of the polynomial, f_i , becomes an optimization free variable:

$$q_i = a_i + b_i t + c_i t^2 + d_i t^3 + f_i t^4 \quad (10)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & T_f & T_f^2 & T_f^3 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2T_f & 3T_f^2 \end{bmatrix} \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \end{bmatrix} = \begin{bmatrix} q_{0i} \\ q_{fi} - f_i T_f^4 \\ \dot{q}_{0i} = 0 \\ \dot{q}_{fi} = -4f_i T_f^3 \end{bmatrix} \quad (11)$$

(For $i = 1,2,3$.) The first coefficients are solved so that the trajectory matches the boundary conditions, while f_i is iterated at each step of the optimization process. In other words, it is like fixing the endpoints of the trajectory and keep reshaping it until an optimal solution is found. Similarly to the exact shape-based method, the maneuver time T_f can be either fixed or introduced as a variable into the optimizer.

If the condition of zero acceleration at endpoints is considered, then the polynomial becomes of 6th degree. However, while a 3rd degree polynomial trajectory may present torque peaks in certain problematic maneuvers, the 4th degree polynomial avoids those peaks since the optimizer already ensures that the maximum torque is never surpassed. Therefore, using a 6th degree polynomial is not necessary in terms of robustness, but it would still help to obtain smoother maneuvers at the endpoints.

IV. Optimization and algorithm implementation

The cost functional of the optimization process is a measure based on the accumulated torque (squared) during the maneuver:

$$J_u = \int_0^{T_f} \langle \mathbf{u}, \mathbf{u} \rangle dt = \int_0^{T_f} (u_1^2 + u_2^2 + u_3^2) dt \quad (12)$$

Where $\mathbf{u} = [u_1, u_2, u_3]^T$ is the external torque vector. Although it is mathematically possible to obtain an analytical expression for this functional in terms of the polynomial coefficients, it is not computationally efficient – the expression requires a very large amount of additional code. Comparing the analytical evaluation to the numerical evaluation of J , it was found that the numerical evaluation was less computationally expensive. Therefore, J_u was evaluated numerically. In this procedure, the polynomials representing attitude were derived with respect to time to

find expressions for \dot{q}_i and \ddot{q}_i , and the trajectory obtained with the shape-based method was discretized. Then the dynamical model described in Section II was used to evaluate angular velocities, angular accelerations and finally the torques u_i at the discretization nodes. Calculating angular rates is useful since they can be fed into a PD tracking controller along with the attitudes. Also, by directly predicting torque values, a maximum-torque constraint can be included in the optimization.

The optimization method used was Nelder-Mead [11], a common algorithm for nonlinear optimization based on a downhill simplex. In order to avoid getting stuck in local minima, the optimizer can be run several times with random initial guesses.

The maneuver time can be included in a multi-objective optimization problem if desired. Regarding time and torque minimization, on one hand time-optimal trajectories obtained may be rather power-consuming; on the other hand, slow maneuvers are low-torque but they may not be operationally practical. Thus, time and torque weighting parameters shall be adjusted accordingly to the mission requirements and constraints.

Introducing maneuver time into the optimization is at the expense of enlarging the search space. Even so, in the exact case of the shape-based method, time is the only optimization variable, thus it is still computationally very efficient.

V. Simulations and results

A. The small satellite model

Since this study was motivated by the need for nano-satellites to perform slew maneuvers, the model used in the simulations corresponds to the 3U Cubesat UKube-1. It weighs 4 kg and has dimensions 10x10x30 cm, with moments of inertia $I_1 = 0.0109 \text{ kgm}^2$, $I_2 = 0.0506 \text{ kgm}^2$, $I_3 = 0.0509 \text{ kgm}^2$.

The torque and storing constraints of very small reaction wheels used for Cubesats, which provide a torque of 23 μNm and are capable of storing 0.58 mNms of angular momentum, were included in the simulations [12]. The real UKube-1 is not equipped with reaction wheels; nevertheless, this type of actuator (3-axis stabilization) was used in the simulations. Those reaction wheels are probably too small to be practical in a Cubesat such as UKube-1, since it would take a long time to perform a maneuver and they would get saturated quickly. Nevertheless, for the purpose of this paper it is interesting to explore highly constrained designs, in order to put the algorithm to test. In addition, the method is also valid for a spacecraft equipped with a reaction control system or any other continuous control actuator for attitude control.

B. Results using the exact and optimized shape-based method

The baseline maneuver presented is a rest-to-rest motion from $q_0 = [0 \ 0 \ 0 \ 1]$ to $q_f = [0.5 \ 0.5 \ 0.5 \ 0.5]$. In terms of Euler Angles, it corresponds to a 90 degree turn of two angles. To simplify, a maneuver time was fixed at $T_f = 350 \text{ s}$.

Figure 3 shows a trajectory planned using the 5th degree polynomial, i.e. the exact shape-based method. Due to the rather symmetric character of this maneuver and being shaped by polynomials, the attitudes and velocities of different axis appear to overlap. Since the time was fixed, no optimization was performed. The maneuver is very smooth: the angular velocities at endpoints are flat (Fig. 3b) and torque values start and end at zero (Fig. 3c; where dotted yellow lines indicate torque constraint). If the maneuver time is modified, the general shape of the trajectory would be the same, but the torque requirements would change. For this particular maneuver, the trajectory shape generated by a 3rd degree polynomial would be rather similar to the 5th degree one; however, as previously

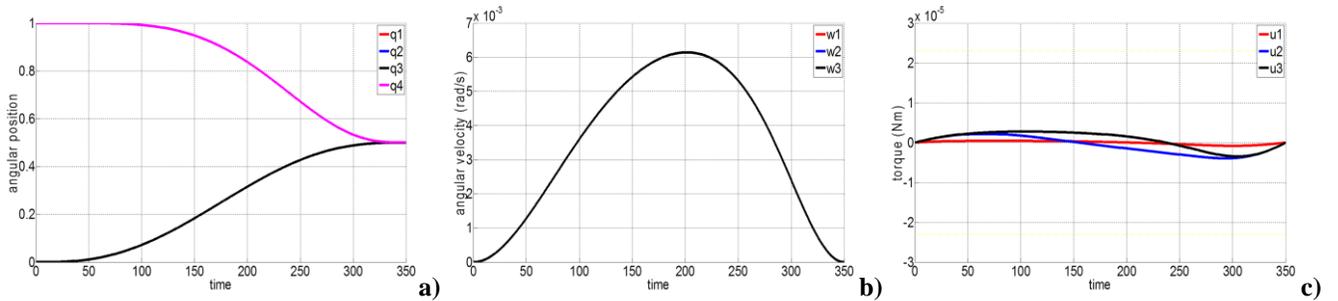


Figure 3. 5th degree polynomial trajectory (exact)

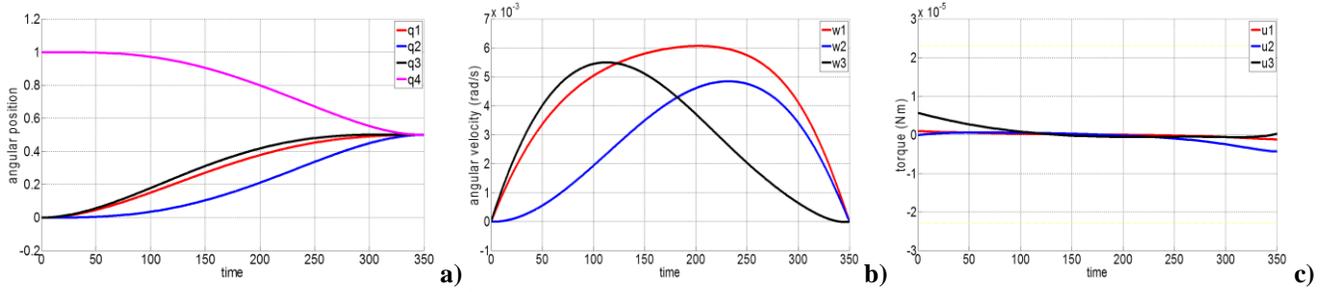


Figure 4. 4th degree polynomial trajectory (optimized)

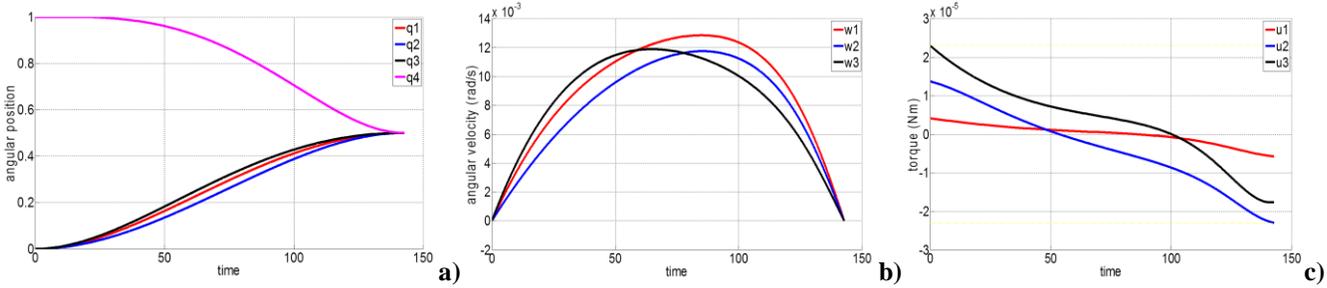


Figure 5. Time-optimal trajectory (4th degree polynomial)

discussed in section III.A, the 5th degree polynomial is more suitable for the sake of robustness (at almost no computational expense) when using the determined shape-based method.

The optimized method generates trajectories using a 4th degree polynomial (no boundary conditions on acceleration), and uses a model of the spacecraft to find the one that minimizes the accumulated torque. Results are shown in Fig. 4. Regarding computation time of the optimized method, running on a desktop computer using MatlabTM it took of the order of 10^{-2} seconds. Here, the attitude and velocity profiles (Fig. 4a, 4b) seek to minimize accumulated torque (Fig. 4c).

Figure 5 shows a maneuver where time was minimized and torque was constrained to its maximum value (but not optimized, in other words, its optimization weight was zero). The shape of the attitude is very similar to the torque-optimal trajectory, but in much less time, therefore the torque profile is more aggressive (Fig. 5 looks as if it was compressed, when compared with Fig. 4). This is the fastest trajectory for this maneuver obtained using the polynomial shape-based method; however, it is not the fastest possible. When only maneuver time is optimized, faster trajectories can be obtained with the optimal control approach: it has the advantage that the torque curves may not be smooth, so that they stick to the limits like a bang-bang control law.

C. The pseudo-spectral optimal control solver

A pseudo-spectral (PS) method for solving optimal control problems [13-15] was used in order to assist the process of selecting a suitable function for the shape-based method and validate the results. Essentially, the pseudo-spectral method discretizes the optimal control problem using special collocation nodes. The differential equations are approximated by a system of algebraic equations and the cost functional is evaluated using a Gauss quadrature. By doing so, the problem is reformulated as a Nonlinear Programming (NLP) problem which is solved using a numerical solver (e.g. a Quasi-Newton method). The software used for the simulations presented here is PSOPT [8]. This method has an exponential or spectral rate of convergence, thus it is able to obtain accurate results with few nodes – therefore a potential candidate for on-board implementation.

In the simulations, the performance index

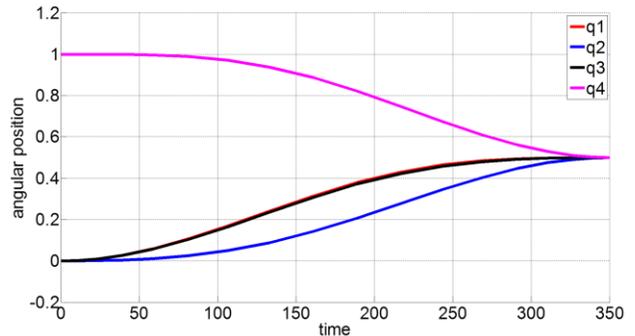


Figure 6. Trajectory obtained using Optimal Control

considered was also Eq. (12). PSOPT has been run with 60 nodes and low tolerance (down to 10^{-6}) in order to seek global optimality. The trajectory obtained with PSOPT (Fig. 6) is quite similar to that obtained with the optimized shape-based method (Fig. 4a), for the same boundary conditions of the maneuver (the performance index is the same).

The solver requires an initial guess. As the reorientation problem formulated here is not highly constrained, convergence was achieved even with a suboptimal and unfeasible initial guess. However, the performance of the pseudo-spectral method may be improved with a good initial guess.

PSOPT was initialized using the output trajectory of the shape-based method, forming a two-stage path planning procedure. Results in Table 1 show that computational time of the solver is indeed improved when fed with a good initial guess, while the solution remains identical. With respect to the suboptimal solution, convergence to the same trajectory was achieved with 30% less iterations. Other maneuvers tested showed a similar behavior.

D. Performance comparison

In this section, the optimization cost (as in Eq. (12)) of a trajectory obtained with several methods is compared. The maneuver at test is the same described in section V.B. The comparison is relative to the best trajectory, obtained by PSOPT (so its relative cost is one). The shape-based method costs (exact and optimized) were tested, as well as a trajectory obtained with a quaternion feedback controller. In order to be comparable, the feedback controller was tuned so that the settling time is near 350 s (the fixed maneuver time for the other methods).

As shown in Table 2, it is important to notice the performance similarity between the optimized shape-based method and the PS method. This quantifies the similarity found between Fig. 6 and Fig. 4a, and validates the polynomial function as a good candidate to obtain minimum accumulated torque maneuvers.

The computation time is of significant importance in nano-spacecraft applications when implementing the algorithm on its microcontroller. Although computation times may vary significantly when optimization parameters such as tolerances or number of nodes are modified, or even when programming languages are different, in this study, the optimized shape-based method was between 10 and 100 times faster than PSOPT.

Table 1 PSOPT's initial guess effect

Initial guess	Nb. of iterations	Comp. time	Optimization cost
Suboptimal	65	10 s	1.867e-9
Shape-based	52	7 s	1.867e-9

Table 2 Performance index comparison

Method	Relative cost
Optimal Control (PSOPT)	1
Exact shape-based (5 th order poly.)	1.84
Optimized shape-based (4 th order poly.)	1.05
Quaternion feedback	11.9

VI. The constrained attitude control problem

Some missions may have pointing constraints, e.g. a sensitive instrument or sensor mounted on the spacecraft that should not point at the Sun. This is a so-called static hard constraint [16], and the shape-based algorithm can be enhanced to take them into account. Such a constraint is formalized as follows:

$$\mathbf{v}(t) \cdot \mathbf{w} \leq \cos \theta_c \quad (13)$$

Where $\mathbf{v}(t)$ is the pointing vector of the sensor and \mathbf{w} is the vector constraint (directed to the Sun), both in the inertial frame. θ_c is the angle of a half-cone whose height is parallel to \mathbf{w} . In other words, the cone defines a keep-out zone in the attitude unit sphere to be avoided by $\mathbf{v}(t)$. In terms of quaternions, the pointing vector is expressed as:

$$\mathbf{v}(t) = \mathbf{v}_B - 2(\mathbf{q}_0^T \mathbf{q}_0) \mathbf{v}_B + 2(\mathbf{q}_0^T \mathbf{v}_B) \mathbf{q}_0 + 2q_4(\mathbf{v}_B \times \mathbf{q}_0) \quad (14)$$

Where \mathbf{v}_B is the pointing vector in the body frame and $\mathbf{q}_0 = [q_1 \ q_2 \ q_3]^T$.

Simulations were undertaken where a certain trajectory crossed the keep-out cone of a pointing constraint. Using a penalty method [11], a proportional cost was applied in the optimization if the trajectory violated the condition stated in Eq. (13). Thus, the optimizer respects the pointing constraint while minimizing torque (Fig. 7). However, if in the constrained attitude problem the maneuver time is fixed, the optimizer may not be able to find a feasible trajectory. Adding the maneuver time as a free optimization variable or as a performance index increases the search

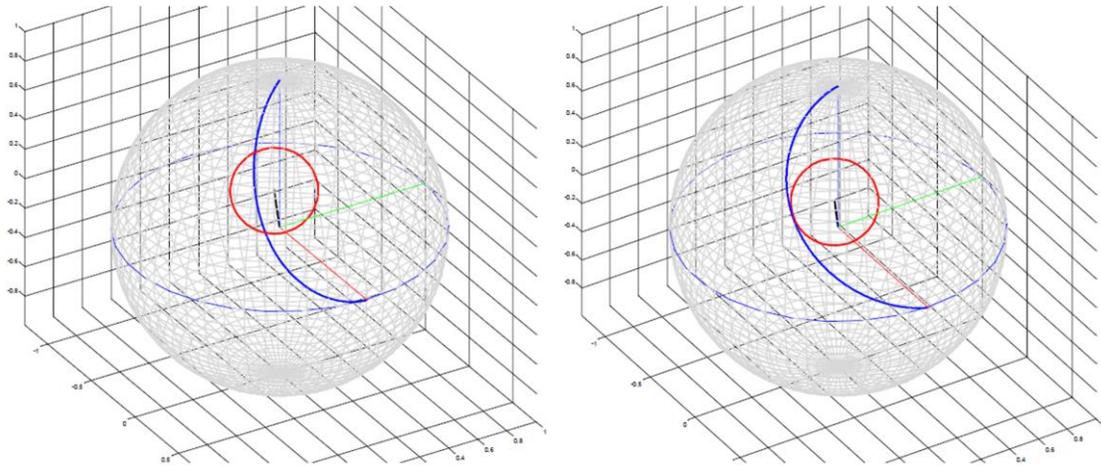


Figure 7. Attitude trajectory avoiding a constraint (red) in the unit sphere

space, improving the versatility of the planner. For similar reasons, the exact shape-based method is not suitable for the constrained attitude problem. If the problem is highly constrained and feasible solutions are not easily found, higher order terms shall be added to the polynomials and their coefficients included in the optimization. By doing so more complex trajectories would be enabled, so that they could deal with heavily constrained spaces.

VII. Conclusion

A motion planner for the generation of close-to-optimal slew maneuvers using a shape-based method has been presented. The method is computationally efficient since trajectories are analytically obtained using polynomials. Using a PS method, the polynomial shape has been validated as an emulator of torque-optimal trajectories. Two different approaches have been studied: the exact and the optimized shape-based method. The first approach has a single free variable, maneuver time. Because of this, the algorithm is very efficient but there is only one possible motion once the boundary conditions (including time) have been set. The former approach uses higher order polynomials to optimize torque as well as time. The search space is larger but results are much closer to the optimal solution found by a PS solver, while computation overhead remains low. Multi-objective torque-time optimizations require appropriate tuning, which is mission dependent. Moreover, the optimized shape-based method can deal with pointing constraints, while it finds a minimum-torque trajectory.

The possibility of using a two-stage planning approach has been studied, where a trajectory generated by the shape-based method is used as a good initial guess for a PS solver. Results show an improvement in computation time, but not by a significant amount. Nevertheless, this approach could be considered for implementation on a nano-spacecraft when facing a highly constrained problem.

Future work may include improving the optimization method (for robustness and speed), implementation and development to highly constrained cases, implementing the algorithm into a microprocessor or a satellite test-bed to demonstrate on-board applications, and extend the method to translational motions.

The shape-based method generates torque- and time- optimal trajectories while ensuring smoothness, is able to deal with constrained spaces and requires low computational overhead. Also, the shape-based algorithm is relatively simple, making software validation cost-effective. Thus, the method has potential applications for on-board path planning on nano-spacecraft, spacecraft with flexible structures, and missions with pointing requirements and constraints.

Acknowledgments

This work has been supported by the Marie Curie fellowship PITN-GA-2011-289240 AstroNet-II

References

- ¹Kashif, G., "Camera Design for Pico and Nano Satellite Applications", Master's Thesis, Luleå University of Technology, Sep-2009, ISSN 1653-0187

- ²Kang, W., Bedrossian, N., “Pseudospectral Optimal Control Theory Makes Debut Flight, Saves NASA \$1M in Under Three Hours”, *SIAM News*, Vol. 40, No. 7, Sep-2007
- ³Biggs, J. D., Horri, N., “Optimal geometric motion planning for spin-stabilized spacecraft”, *Systems and Control Letters*, vol. 61, issue.4. pp. 609-616, 2012.
- ⁴Petropoulos, A. E., Longuski, J. M., “Shape-Based Algorithm for Automated Design of Low-Thrust, Gravity-Assist Trajectories”, *Journal of Spacecraft and Rockets*, Vol. 41, No. 5, Sep-Oct 2004, pp. 787-796.
- ⁵Wall, B. J., Conway, B. A., “Shape-Based Approach to Low-Thrust Rendezvous Trajectory Design”, *Journal of Guidance, Control and Dynamics*, Vol. 32, No. 1, Jan-Feb 2009, pp. 95-101.
- ⁶Vasile, M., De Pascale, P., and Casotto, S., “On the optimality of a shape-based approach based on pseudo-equinoctial elements”. *Acta Astronautica*, 61 (1-6). pp. 286-297, 2007.
- ⁷McInnes, C., “Satellite Attitude Slew Manoeuvres Using Inverse Control”, *The Aeronautical Journal*, 1998, pp. 259-265.
- ⁸Becerra, M. V., “PSOPT Optimal Control Solver User Manual”, Release 3 2011, URL: <http://www.psopt.org>
- ⁹Wie, B., *Space Vehicle Dynamics and Control*, 2nd ed., AIAA Education Series, 2008.
- ¹⁰Phillips, J. R., “ZunZun.com Online Curve Fitting”, URL: <http://zunzun.com>
- ¹¹Nocedal, J., Wright, J.S., “Numerical Optimization”, 2nd ed., Springer
- ¹²Stoltz, S., Courtois, K., Raschke, C., Baumann, F., “The World’s smallest reaction wheel – the development, fields of operation and flight results of the RW 1”, *33rd Annual AAS Guidance and Control Conference*, Feb-2010
- ¹³Elnagar, G., Kazemi, M. A., and Razzaghi, M., “The Pseudospectral Legendre Method for Discretizing Optimal Control Problems”, *IEEE Transactions on Automatic Control*, 40:1793–1796, 1995
- ¹⁴Ross, M. I., Karpenko, M., “A review of pseudospectral optimal control: From theory to flight”, *Annual Reviews in Control*, 2012.
- ¹⁵Fahroo, F., Ross, M. I., “Advances in Pseudospectral Methods for Optimal Control”, *AIAA Guidance, Navigation and Control Conference*, Aug-2008.
- ¹⁶Kim, Y., Mesbahi, M., “On the constrained attitude control problem”, *AIAA Guidance, Navigation and Control Conference*, Aug-2004.