2010

Time-Optimal Reorientation of a Spacecraft using a Direct Optimization Method Based on Inverse Dynamics

Boyarko, George A.; Romano, Marcello; Yakimenko, Oleg A.

Time-Optimal Reorientation of a Spacecraft using a Direct Optimization Method Based on Inverse Dynamics

George A. Boyarko, Marcello Romano and Oleg A. Yakimenko
Naval Postgraduate School
Department of Mechanical and Astronautical Engineering
Monterey, CA, 93943-5107
1-831-656-2885
gaboyark@nps.edu, mromano@nps.edu, oayakime@nps.edu

Abstract—This paper proposes a rapid attitude trajectory generation method for satellite reorientation that satisfies the spatial and temporal constraints imposed by the problem of docking with a tumbling object. [12]The problem is first formulated and solved using known academic software readily used for generating optimal guidance trajectories off-line. Then, the problem is reformulated using a polynomial structure that lends itself to satisfying special mathematical constraints imposed by using a unit quaternion for orientation description. The speed profile of the maneuver is varied in order to arrive at a quasi-optimal solution that is both feasible and exactly matches the endpoint conditions specified in the problem. The reduction in the number of varied parameters due to the predetermined structure of the trajectory leads to a faster computational speed as well as having a trajectory that satisfies the end constraints at each iteration. The paper ends with a discussion of solutions obtained for several cases.

TABLE OF CONTENTS

1. INTRODUCTION

The optimal satellite reorientation problem is of general interest to many in the field of aerospace engineering. The available literature on this subject is extensive (see for instance [1-3]). Many civilian and military space missions need to have agile attitude maneuver capability. For instance, TacSat-3 was intended to demonstrate responsive delivery of information to operational users [4]. Due to the satellite's Low Earth Orbit, the timeline for tasking, slewing and disseminating data is greatly reduced. Other challenges include the fact that the tasking can be modified at any moment up to a short period of time after the ground station starts uploading the tasking, as well as the idea that TacSat must autonomously slew to the target, collect and process the data, and then down link the data directly to the customer who is not collocated with the ground station. Finally, current real time feedback controls are not optimized for minimum time [2]. The preceding challenges lend themselves to the need for the ability to rapidly generate feasible trajectories that are optimized for minimum time.

The goal of the present paper is to provide a method to determine a feasible attitude trajectory solution that meets endpoint requirements and dynamic constraints while performing a good overall maneuver relative to a given performance index. Also, the method should work for any boundary conditions including non-rest to non-rest maneuvers. The major requirement is that the method must provide a feasible real-time solution as opposed to off-line computations even if it requires some sacrifices in optimality.

The existing techniques include using so-called pseudospectral methods. These methods can provide an incredibly accurate solution to an optimal control problem, but may require extended periods of time to converge to a solution or converge to a sub-optimal solution in case of a lesser number of nodes [5-7]. In addition, the optimal solution does not have an analytical representation, which may pose problems when trying to implement it using a feed-forward scheme of suggested control commands [5-7]. If the numerical solution is afterwards approximated with some analytical function and/or controls are smoothed, it may lose some optimality and disable the smoothed solution to arrive at the terminal conditions.

The authors pursue another approach exploiting the general idea of the direct optimization methods of calculus of variations together with an inverse dynamics approach. In particular, polynomials are used as basis functions to generate spatial trajectories that can be traversed according to a computed analytic speed profile. Instead of basing trajectories on time, an abstract argument is introduced that allows the trajectory to be formulated in such a way that the states and their derivatives are specified at the endpoints. This results in decoupling space and time, allowing the speed over the trajectory to be varied in order to satisfy

[1] 978-1-4244-3888-4/10/$25.00 ©2010 IEEE
[2] IEEEAC paper#1516, Version 2, Updated 2009:12:27

problem constraints. The resulting quasi-optimal trajectory solution can be generated (and updated) rapidly because of the reduction in the number of varied parameters due to the restriction on the trajectory structure by specifying a polynomial basis. Although this method lacks some flexibility due to the predefined structure, it provides a feasible solution that satisfies the endpoint constraints on the trajectory at every iteration, even when the initial conditions change due to disturbances or delays. Specific applications include scenarios where derivative conditions on beginning and ending states need to be met, such as tracking missions, docking missions and other missions where a simple eigenaxis slew may be unacceptable.

It should be noted that using inverse dynamics to optimize the rotational motion of a satellite has been already evaluated by other authors as well [8,9]. However, Euler angles were used, which suffer from well-known kinematic singularities, and no attempt to decouple the time and space domains was made.

The present paper proposes a novel approach resulting in a fairly robust computational technique as compared to all other aforementioned techniques and is organized as follows. Section 2 describes the problem to be solved as well as the dynamic and kinematic models. Section 3 demonstrates obtaining a solution using an academic solver employing a pseudospectral method. Section 4 introduces the Inverse Dynamics in the Virtual Domain (IDVD) approach and develops a complete computational scheme using Bezier curves to approximate attitude dynamics, followed by Section 5 presenting sample solutions. Finally, Section 6 compares the results of optimization obtained with both methods with conclusions drawn in Section 7.

2. SPACECRAFT MODEL AND ATTITUDE TRAJECTORY OPTIMIZATION PROBLEM

The rotational dynamics of the spacecraft can be described by Euler's rotational equations of motion. Written in the body-fixed principal axes this results in the vector equation [2,4]

$$\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} = \mathbf{T}, \tag{1}$$

which expands into the scalar equations:

$$\begin{aligned} \dot{\omega}_x \triangleq \alpha_x &= \frac{(I_{22} - I_{33})\omega_y \omega_z + T_x}{I_{11}}, \\ \dot{\omega}_y \triangleq \alpha_y &= \frac{(I_{33} - I_{11})\omega_x \omega_z + T_y}{I_{22}}, \\ \dot{\omega}_z \triangleq \alpha_z &= \frac{(I_{11} - I_{22})\omega_y \omega_x + T_z}{I_{33}}. \end{aligned} \tag{2}$$

In Eqs.(2), (3) $\mathbf{I} = diag([I_{11}, I_{22}, I_{33}])$ is the inertia matrix (along the principal axes), $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^{\mathrm{T}}$ is the vector of angular velocities, and $\mathbf{T} = [T_x, T_y, T_z]^{\mathrm{T}}$ is the vector of torques (bounded controls).

In turn, rotational kinematics can be described using quaternion $\mathbf{q} = [q_1, q_2, q_3, q_4]^{\mathrm{T}}$ as: [10,11]

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}. \tag{3}$$

The problem in question is to find the slew trajectory (quaternion time history) for a satellite subject to specific constraints that minimizes the time to complete the maneuver, t_f. This is expressed as minimizing the performance index:

$$J = \int_0^{t_f} dt, \tag{4}$$

while reorienting a satellite from the initial conditions $\boldsymbol{\omega}_0$, $\mathbf{q}_0$ to final conditions $\boldsymbol{\omega}_f$, $\mathbf{q}_f$ for a system (2)-(3), subject to constraints on controls

$$\mathbf{T}_{\min} \le \mathbf{T} \le \mathbf{T}_{\max}. \tag{5}$$

Bilamoria and Wie formulated this problem for a rest-to-rest maneuver and presented the solution using an indirect method [3]. They showed that, in general, for the case of a symmetric body with bounds on each torque component, it results in a non-eigenaxis maneuver. In addition to that, the following section presents a more general solution obtained off-line to be used along with that of [3] as a reference for the proposed on-line solution obtained using a direct method exploiting the inverse dynamics of Eqs. (2)-(3).

To this end, Table 1 shows two different test cases examined in this paper. Test Case 1, representing an idealized rather than real spacecraft, was taken directly from [3], while another was chosen to illustrate a more general scenario, when a spacecraft is not necessarily symmetric.

Table 1. Description of the test cases.

Case	*Normalized Inertia Matrix*
Case 1	$\mathbf{I} = diag([1,1,1])$
Case 2	$\mathbf{I} = diag([3,1,2])$

In terms of the endpoint conditions the paper considers two basic scenarios assuming ϕ =90° and ϕ=180° slew maneuvers about the *z*-axis (so that $\mathbf{q}_0 = [0,0,0,1]^T$ and $\mathbf{q}_f = [0,0,\sin\frac{1}{2}\phi, \cos\frac{1}{2}\phi]^T$) with zero and non-zero normalized body rates at the endpoints ($\boldsymbol{\omega}_0 = \boldsymbol{\omega}_f = \mathbf{0}_{3\times1}$ and

$\boldsymbol{\omega}_0 = -\boldsymbol{\omega}_f \neq \mathbf{0}_{3\times1}$. Finally, for the normalized states, the constraints (5) take the form $-\mathbf{1}_{3\times1} \leq \mathbf{T} \leq \mathbf{1}_{3\times1}$.

All computations were carried out on a 2.33GHz Dell Precision M90 desktop computer with an Intel T7600 processor and 1Gb of RAM. As the optimization engine, SNOPT (GPOPS) and MATLAB *fmincon* function (IDVD) were used. For the sake of completeness and repeatability it should also be noted that while the IDVD solution was obtained in the purely interpretative environment of MATLAB, SNOPT used a library of optimized executable files and, therefore, was much more computationally effective.

3. SOLVING THE PROBLEM USING THE GAUSS PSEUDOSPECTRAL METHOD

Before we proceed with the proposed on-line solution, let us address the problem formulated in the previous section using one of the prominent pseudospectral (collocation) methods. The goal is to have some reference solutions and also to see if the solutions obtained using this approach can be reliably used for on-line optimization.

It should be noted that the Bilimoria and Wie solution has been matched using one quasioptimal method already (see the work by Fleming [12], who used the commercial software package DIDO [13]). The authors of this study chose to explore the Gauss Pseudospectral Optimization Software (GPOPS) package based on its open source nature and free availability [14].

Figures 1-8 show the results of applying GPOPS to obtain (reference) minimum-time solutions for a 180° slew of a satellite. Specifically, Figs.1 and 2 present time histories of all states and controls for the solution that involves 100 nodes, which results in (100-2)×10=980 variable parameters. Figure 3 depicts the three-dimensional (3D) representation of the solution in inertial space, clearly showing that it is not an eigenaxis maneuver, with an inclination of the z_b axis during rotation in the $x_b y_b$ direction.

This solution compares with the solution presented in [3] fairly well. The final calculated maneuver time, t_f, was found to be 3.243 seconds. However, it took almost two hours of CPU time to obtain this solution. Another observation is that because of the nature of the system (2), the optimal control has a bang-bang structure (Fig.2). That results in the maximum magnitude of the angular acceleration at the boundary points (for Case 1 angular accelerations are simply equal to the corresponding controls). It means arriving at the terminal conditions with the maximum angular acceleration. Also, if we are to update a trajectory while the satellite performs this rotation (to accommodate possible disturbances and unmodeled dynamics), it would cause discontinuities of angular acceleration (sudden jumps in controls). Increasing the order of the system to account for the boundary conditions on angular accelerations will obviously cause a slight degradation of the performance index and a further increase of the required CPU time to obtain a solution. Hence, although in this case GPOPS does produce a valid solution, it is not practical and cannot be used for online computations.

Figure 1. Time histories of the states for Case 1.

Figure 2. Time history of the controlling torques for Case 1.

Figure 3. The 3D representation of the solution for Case 1.

As pointed out in [14], reducing the number of nodes may lead to a more robust (in terms of computational time) result, therefore an attempt was made to obtain a solution of the same problem using a lesser number of nodes. These GPOPS solutions are shown in Figs.4-6.

It turns out that for a lesser number of nodes the GPOPS converges to different solutions. To this end, Fig.4 shows time histories of the angular velocity components for the 25-

and 50-node solution (involving 230 and 480 varied parameters, respectively). Obviously, they are different from that of the 100-node solution in Fig.2. While a 25-node solution is simply symmetrical with respect to the 100-node solution, as can be seen by comparing Fig.3 and Fig.5 showing an inclination of the z_b axis during rotation in the $-x_b y_b$ direction, and represents another equally optimal solution out of possible four solutions [12], a 50-node solution appears not to be valid (optimal) at all (Fig.6).

As expected, decreasing the number of nodes leads to a substantial decrease in the computational time, but as shown above the method could produce a non-valid solution. Also, even if it produces a valid solution the time histories for control torques may not be trackable by the inner-loop controllers. These give two more reasons why the solutions obtained using pseudospectral methods may not be used in a real-time feedforward control scheme.

Figure 4. Case 1 comparison of time histories of angular velocity components obtained for 25 nodes (top) and 50 nodes (bottom) for GPOPS solution.

Figure 5. The 3D representation of the 25-node GPOPS solution for Case 1.

For Case 2, the nonsymmetrical inertia with the bounds on individual control torques, the solution is slightly different (Figs.7 and 8). The overall characteristic of this and other solutions involving different sets of the boundary conditions will be presented in Section V, but the general tendency is the same – it requires at least a hundred nodes to produce a valid and feasible off-line solution. Yet, GPOPS presents a good and easy to-use tool to produce reference trajectories that can be used for comparison with solutions obtained using other approaches. One of them is introduced next.

Figure 6. Case 1 comparison of time histories of torques, obtained for 25 nodes (top) and 50 nodes (bottom) for GPOPS solution.

Figure 7. Time histories of the states for Case 2.

Figure 8. Time history of the controlling torques for Case 2.

4. ESSENCE OF INVERSE DYNAMICS IN THE VIRTUAL DOMAIN APPROACH

One of the two main ideas of the Inverse Dynamics in the Virtual Domain (IDVD) method is exploiting the differential flatness property of the equations of motion [15-17]. In the above problem, this relates to the fact that all the

state and control variables can be expressed as functions of the output variable or time derivatives of the output variable, which in this case is the quaternion itself:

$$\boldsymbol{\omega} = f_1(\mathbf{q}, \dot{\mathbf{q}}), \quad \mathbf{T} = f_2(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}). \tag{6}$$

Another aspect of IDVD involves handling computations in the virtual domain allowing space and time decoupling. By doing so, a trajectory can be computed while also manipulating the speed at which that trajectory is followed.

The following presents a novel parameterization for the output variable, components of the quaternion, and develops a step-by-step computational routine.

Quaternion Parameterization

In order to parameterize the problem, the output trajectory is approximated using some combination of basis functions. The standard approach would be to choose some combination of polynomials or trigonometric functions for the output variables [15-17]. While this may be straightforward when dealing with state variables in translational space, it may become more challenging when dealing with expressions for orientation.

While a quaternion may be the preferred method to express attitude because of the lack of singularities, choosing basis functions becomes more challenging because a nonlinear unit norm condition needs to be preserved across the quaternion history [18]. For this reason, a specific polynomial expression for the quaternion was chosen inspired by the work of Kim, Kim and Shin [19]. This consists of expressing the quaternion time history as an exponential function containing a constant parameter multiplied by a Bezier polynomial of a degree *n*:

$$\mathbf{q}(\tau) = \mathbf{q}_0 \prod_{i=1}^{n} \exp(\tilde{\boldsymbol{\omega}}_i \tilde{\beta}_{i,n}(\tau)), \tag{7}$$

where

$$\tilde{\beta}_{i,n}(\tau) = \sum_{j=i}^{n} \beta_{j,n}(\tau) \tag{8}$$

and

$$\beta_{i,n}(\tau) = \binom{n}{i} (1-\tau)^{n-i} \tau^i. \tag{9}$$

Note that in Eqs. (7)-(9) $\tau \in [0;1]$ is an abstract argument that is used instead of time. This allows us to exploit certain attributes of the Bezier polynomials and define properties at the beginning and endpoints.

For example, the analytic expressions of the 5th-order Bezier polynomial, $\tilde{\beta}_{i,5}(\tau)$, are as follows:

$$\begin{aligned}
\tilde{\beta}_{1,5}(\tau) &= \tau^5 - 5\tau^4 + 10\tau^3 - 10\tau^2 + 5\tau, \\
\tilde{\beta}_{2,5}(\tau) &= -4\tau^5 + 15\tau^4 - 20\tau^3 + 10\tau^2, \\
\tilde{\beta}_{3,5}(\tau) &= 6\tau^5 - 15\tau^4 + 10\tau^3, \\
\tilde{\beta}_{4,5}(\tau) &= -4\tau^5 + 5\tau^4, \\
\tilde{\beta}_{5,5}(\tau) &= \tau^5.
\end{aligned} \tag{10}$$

These expressions have the favorable properties:

$$\tilde{\beta}_{i,5}(0) = 0 \text{ and } \tilde{\beta}_{i,5}(1) = 0, \text{ for } i = 1,...,5; \tag{11}$$

$$\begin{aligned}
&\tilde{\beta}'_{1,5}(0) = \tilde{\beta}'_{5,5}(1) = 5, \ \tilde{\beta}'_{1,5}(1) = \tilde{\beta}'_{5,5}(0) = 0, \\
&\tilde{\beta}'_{i,5}(0) = \tilde{\beta}'_{i,5}(1) = 0, \text{ for } i = 2,3,4;
\end{aligned} \tag{12}$$

$$\begin{aligned}
&\tilde{\beta}''_{1,5}(0) = \tilde{\beta}''_{5,5}(1) = 20, \quad \tilde{\beta}''_{1,5}(1) = \tilde{\beta}''_{5,5}(0) = 0, \\
&\tilde{\beta}''_{2,5}(0) = \tilde{\beta}''_{4,5}(1) = -20, \ \tilde{\beta}''_{2,5}(1) = \tilde{\beta}''_{4,5}(0) = 0, \\
&\tilde{\beta}''_{3,5}(0) = \tilde{\beta}''_{3,5}(1) = 0,
\end{aligned} \tag{13}$$

which fix the value of the polynomial and it derivatives at the endpoints specified by values of $\tau \in [0; \tau_f]$ where we set τ_f=1 to exploit the above properties in Eqs. (11)-(13). The values of $\tilde{\boldsymbol{\omega}}_i$ are coefficients defined by the following relation:

$$\tilde{\boldsymbol{\omega}}_i = \log(\tilde{\mathbf{q}}_{i-1}^{-1} \tilde{\mathbf{q}}_i), \text{ for } i = 1,\ldots,5. \tag{14}$$

Here $\tilde{\mathbf{q}}_i$ are the constant column vectors that act as control points and $\tilde{\boldsymbol{\omega}}_i$ represents a constant augmented angular velocity vector based on Eq.(14).

The prevailing idea is that at $\mathbf{q}(\tau = 0) = \tilde{\mathbf{q}}_0$ and $\mathbf{q}(\tau = 1) = \tilde{\mathbf{q}}_5$, where $\tilde{\mathbf{q}}_0$ and $\tilde{\mathbf{q}}_5$ can be fixed such that $\tilde{\mathbf{q}}_0 \triangleq \mathbf{q}(t_0)$ and $\tilde{\mathbf{q}}_5 \triangleq \mathbf{q}(t_f)$. This also results in a straightforward calculation of higher order derivatives of the quaternion curve with respect to the virtual domain argument τ.

The results for the first derivative for a 3rd-order Bezier polynomial were presented in [19] already. In our case, for the 5th-order Bezier polynomial (10), the first-order derivative with respect to the argument τ is given by:

$$\mathbf{q}'(\tau) = \frac{d\mathbf{q}}{d\tau}(\tau) = \mathbf{q}_0 \sum_{j=1}^{n} \left(\tilde{\boldsymbol{\omega}}_j \tilde{\beta}'_{j,n}(\tau) \right) \prod_{i=1}^{n} \exp(\tilde{\boldsymbol{\omega}}_j \tilde{\beta}_{j,n}(\tau)). \tag{15}$$

Similarly, the 2nd-order derivative of a 5th-order Bezier polynomial-based quaternion is calculated by proper application of the chain rule as follows:

$$\mathbf{q}''(\tau)=\mathbf{q}_0\left(\sum_{j=1}^{n}\left(\tilde{\boldsymbol{\omega}}_j\tilde{\beta}''_{j,n}(\tau)\right)+\sum_{k=1}^{n}\left(\tilde{\boldsymbol{\omega}}_k\tilde{\beta}'_{k,n}(\tau)\right)\sum_{j=1}^{n}\left(\tilde{\boldsymbol{\omega}}_j\tilde{\beta}'_{j,n}(\tau)\right)\right)\prod_{i=1}^{n}\exp(\tilde{\boldsymbol{\omega}}_j\tilde{\beta}_{j,n}(\tau)). \tag{16}$$

Now, the reason that the polynomial was expanded from a 3rd- (as shown in Kim [19]) to a 5th-order is that we need to fix the 1st- and 2nd-order derivatives of the quaternion function at the endpoints. By applying Eqs. (9)-(13), therefore exploiting the property that certain terms $\tilde{\beta}_{i,5}$, $\tilde{\beta}'_{i,5}$ and $\tilde{\beta}''_{i,5}$ are equal to zero, derivative values at the endpoints can be calculated by the simple expression:

$$\left.\frac{d\mathbf{q}}{d\tau}\right|_{\tau=0}=5\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_1,\ \left.\frac{d\mathbf{q}}{d\tau}\right|_{\tau=1}=5\tilde{\mathbf{q}}_5\tilde{\boldsymbol{\omega}}_5. \tag{17}$$

Note in Eq. (17), the first-order derivative of the quaternion describes how the parameter $\tilde{\boldsymbol{\omega}}_i$ is related to the angular velocity at the endpoints. In similar fashion,

$$\begin{aligned}\left.\frac{d^2\mathbf{q}}{d\tau^2}\right|_{\tau=0}&=-20\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_1+25\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_1{}^2+20\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_2,\\ \left.\frac{d^2\mathbf{q}}{d\tau^2}\right|_{\tau=1}&=-20\tilde{\mathbf{q}}_4\tilde{\boldsymbol{\omega}}_4\tilde{\mathbf{q}}_4{}^{-1}\tilde{\mathbf{q}}_5+20\tilde{\mathbf{q}}_5\tilde{\boldsymbol{\omega}}_5+25\tilde{\mathbf{q}}_5\tilde{\boldsymbol{\omega}}_5{}^2.\end{aligned} \tag{18}$$

Mapping from the Virtual to Time Domain

Now that the trajectory is set using a virtual domain, a mapping must be employed to convert this trajectory into a time dependent one. To do this, a speed factor, λ, is defined that maps the points on the trajectory from the virtual domain to the time domain, therefore defining the final time of the maneuver:

$$\lambda=\frac{d\tau}{dt},\ t_f=\int_0^{t_f}\frac{d\tau}{\lambda}. \tag{19}$$

Obviously, more complex structures of $\lambda(\tau)$ will provide more flexibility in the trajectory. However, for this application, we restrict $\lambda(\tau)$ to a function that contains a reduced number of varied parameters as follows:

$$\lambda(\tau)=\lambda_0+a\tau^2+b(1-\tau)^2+c\left(1-(1-\tau)^2\right)+d(1-\tau^2) \tag{20}$$

(the idea is to keep it positive). The analytical integral to Eq.(20) not only provides computational efficiency and an accurate integration to the minimum-time performance index, but also provides a continuous mapping from the virtual domain to the time domain. Alternatively stated, a continuous control history is available, whose resolution does not suffer from a limited number of node points.

Although a speed factor of the form (18) does not allow matching the optimal minimum-time solutions precisely, varying the parameters contained within $\lambda(\tau)$ (λ_0, *a*, *b*, *c*, and *d*) still allows sufficient variation of the speed along the trajectory defined by Eq. (7) to produce feasible and easy to track solutions.

Inverting the Dynamics

As a result of the mapping from virtual to time domain, the expression for the differential of $\mathbf{q}$ with respect to time is:

$$\dot{\mathbf{q}}=\frac{d\mathbf{q}}{dt}=\frac{d\mathbf{q}}{d\tau}\lambda,\ \ddot{\mathbf{q}}=\frac{d^2\mathbf{q}}{dt^2}=\frac{d^2\mathbf{q}}{d\tau^2}\lambda^2+\frac{d\mathbf{q}}{d\tau}\frac{d\lambda}{d\tau}\lambda. \tag{21}$$

Now if the trajectory in the virtual domain, $\mathbf{q}(\tau)$, is specified, along with the speed trajectory, $\lambda(\tau)$, the resulting trajectory of $\mathbf{q}(t)$ as well as its higher order derivatives can be analytically expressed and mapped to the time domain.

Inverting kinematic equations (3) and differentiating the result yields analytical expressions for the angular velocity and angular acceleration:

$$\begin{aligned}\boldsymbol{\omega}(t)&=2\mathbf{q}^{-1}(t)\dot{\mathbf{q}}(t),\\ \boldsymbol{\alpha}(t)&=\dot{\mathbf{q}}^{-1}(t)2\ddot{\mathbf{q}}(t)-\dot{\mathbf{q}}(t)\boldsymbol{\omega}(t).\end{aligned} \tag{22}$$

The torque history needed to follow such a trajectory is calculated by inverting Eq.(2):

$$\begin{aligned}T_x(t)&=\alpha_x(t)+(I_{33}-I_{22})\omega_y(t)\omega_z(t),\\ T_y(t)&=\alpha_y(t)+(I_{11}-I_{33})\omega_1(t)\omega_z(t),\\ T_z(t)&=\alpha_z(t)+(I_{22}-I_{11})\omega_y(t)\omega_x(t).\end{aligned} \tag{23}$$

Matching Initial Conditions

From the preceding equations, a quaternion history can be developed based on the Bezier polynomial that satisfies predefined beginning and ending quaternion values as well as setting the angular velocity and angular acceleration at the endpoints. The desired angular velocity and acceleration dictate the quaternion derivatives at the endpoints to be as follows:

$$\begin{aligned}\dot{\mathbf{q}}(t_0)&=\mathbf{q}(t_0)\boldsymbol{\omega}(t_0),\\ \dot{\mathbf{q}}(t_f)&=\mathbf{q}(t_f)\boldsymbol{\omega}(t_f),\\ \ddot{\mathbf{q}}(t_0)&=\dot{\mathbf{q}}(t_0)2\boldsymbol{\alpha}(t_0)+\dot{\mathbf{q}}(t_0)\boldsymbol{\omega}(t_0),\\ \ddot{\mathbf{q}}(t_f)&=\dot{\mathbf{q}}(t_f)2\boldsymbol{\alpha}(t_f)+\dot{\mathbf{q}}(t_f)\boldsymbol{\omega}(t_f).\end{aligned} \tag{24}$$

Based on the properties of the 5th-order Bezier polynomial, the coefficients of the quaternion expression are then calculated by:

$$\tilde{\boldsymbol{\omega}}_1 = \frac{1}{10}\boldsymbol{\omega}(t_0), \tag{25}$$

$$\tilde{\boldsymbol{\omega}}_5 = \frac{1}{10}\boldsymbol{\omega}(t_f), \tag{26}$$

$$\tilde{\boldsymbol{\omega}}_2 = \frac{\tilde{\mathbf{q}}_0^{-1}\ddot{\mathbf{q}}(t_0) + 20\tilde{\boldsymbol{\omega}}_1 - 25\tilde{\boldsymbol{\omega}}_1^{\,2}}{20}, \tag{27}$$

$$\tilde{\boldsymbol{\omega}}_4 = \frac{\tilde{\mathbf{q}}_4^{\,-1}\left(\ddot{\mathbf{q}}(t_f) + 20\tilde{\mathbf{q}}_5\tilde{\boldsymbol{\omega}}_1 - 25\mathbf{q}_5\tilde{\boldsymbol{\omega}}_1^{\,2}\right)\tilde{\mathbf{q}}_5^{\,-1}\tilde{\mathbf{q}}_4}{-20}. \tag{28}$$

Here $\ddot{\mathbf{q}}$ is computed using the second equation in Eq.(20) and the complementary $\tilde{\mathbf{q}}_i$ parameters defined as:

$$\begin{aligned} &\tilde{\mathbf{q}}_0 = \mathbf{q}(t_0) = \mathbf{q}(\tau)\big|_{\tau=0}, \quad \tilde{\mathbf{q}}_1 = \tilde{\mathbf{q}}_0\exp(\tilde{\boldsymbol{\omega}}_1), \\ &\tilde{\mathbf{q}}_5 = \mathbf{q}(t_f) = \mathbf{q}(\tau)\big|_{\tau=1}, \quad \tilde{\mathbf{q}}_4 = \tilde{\mathbf{q}}_5\exp(\tilde{\boldsymbol{\omega}}_5)^{-1}. \end{aligned} \tag{29}$$

and

$$\begin{aligned} &\tilde{\mathbf{q}}_2 = \mathbf{q}_1\exp(\tilde{\boldsymbol{\omega}}_2), \\ &\tilde{\mathbf{q}}_3 = \mathbf{q}_5\left(\exp(\tilde{\boldsymbol{\omega}}_4)\exp(\tilde{\boldsymbol{\omega}}_5)\right)^{-1}. \end{aligned} \tag{30}$$

Finally,

$$\tilde{\boldsymbol{\omega}}_3 = \log(\tilde{\mathbf{q}}_2^{\,-1}\tilde{\mathbf{q}}_3). \tag{31}$$

The resulting benefit of this laborious formulation is that the attitude trajectory history of a 4x1 $\mathbf{q}$ vector that satisfies the constraints of a unit quaternion can be specified by a reduced set of parameters. These parameters are the initial and final conditions on the quaternion itself as well as values of angular velocity and angular acceleration at those endpoints.

Increasing the Polynomial Order

More flexibility in the trajectory can be obtained by increasing the order of the Bezier polynomial used in the basis function. For the case of a 7th-order polynomial, the same structure as Eq. (7) is employed, but now with n=7. This leads to the introduction of $\tilde{\mathbf{q}}_6$, $\tilde{\mathbf{q}}_7$, $\tilde{\boldsymbol{\omega}}_6$ and $\tilde{\boldsymbol{\omega}}_7$ which are defined to be consistent with previous definitions from Eqs. (7) and (14). If the values of orientation and angular velocity at the endpoints are set, endpoint conditions of angular jerk as well as angular acceleration can be used as varied parameters. Setting a specified (low) value for the initial and final jerk can be critical for slewing maneuvers of flexible spacecraft, in order to avoid excitation of structural modes.

To do this, the third-order derivative of the output vector is calculated as:

$$\begin{aligned} \dddot{\mathbf{q}} = &\frac{d^3\mathbf{q}}{d\tau^3}\lambda^3 + \frac{d^2\mathbf{q}}{d\tau^2}2\lambda\frac{d\lambda}{d\tau}\lambda + \frac{d^2\mathbf{q}}{d\tau^2}\frac{d\lambda}{d\tau}\lambda^2 \\ &+ \frac{d\mathbf{q}}{d\tau}\frac{d^2\lambda}{d\tau^2}\lambda^2 + \frac{d\mathbf{q}}{d\tau}\left(\frac{d\lambda}{d\tau}\right)^2\lambda. \end{aligned} \tag{32}$$

The new expressions for the virtual derivatives (taken with respect to the virtual argument τ) are also recalculated and are analogous to Eqs. (17) and (18), but with the addition of a third derivative to accommodate the change between a 5th- and 7th-order polynomial:

$$\begin{aligned} &\mathbf{q}'\big|_{\tau=0} = 7\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_1, \\ &\mathbf{q}'\big|_{\tau=1} = 7\tilde{\mathbf{q}}_7\tilde{\boldsymbol{\omega}}_7; \end{aligned} \tag{33}$$

$$\begin{aligned} &\mathbf{q}''\big|_{\tau=0} = 49\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_1^2 + 42\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_2 - 42\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_1, \\ &\mathbf{q}''\big|_{\tau=1} = 42\tilde{\mathbf{q}}_7\tilde{\boldsymbol{\omega}}_7 + 49\tilde{\mathbf{q}}_7\tilde{\boldsymbol{\omega}}_7^2 - 42\tilde{\boldsymbol{\omega}}_6\tilde{\mathbf{q}}_7; \end{aligned} \tag{34}$$

$$\begin{aligned} \mathbf{q}'''\big|_{\tau=0} = &343\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_1^3 - 882\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_1^2 - 420\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_2 \\ &+210\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_1 + 210\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_3 + 882\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_1\tilde{\boldsymbol{\omega}}_2, \\ \mathbf{q}'''\big|_{\tau=1} = &343\tilde{\mathbf{q}}_7\tilde{\boldsymbol{\omega}}_7^3 - 882\tilde{\mathbf{q}}_6\tilde{\boldsymbol{\omega}}_6\tilde{\boldsymbol{\omega}}_7\tilde{\mathbf{q}}_6^{-1}\tilde{\mathbf{q}}_7 + 210\tilde{\boldsymbol{\omega}}_5\tilde{\mathbf{q}}_7 \\ &+210\tilde{\mathbf{q}}_7\tilde{\boldsymbol{\omega}}_7 + 882\tilde{\mathbf{q}}_7\tilde{\boldsymbol{\omega}}_7^2 + 882\tilde{\mathbf{q}}_0\tilde{\boldsymbol{\omega}}_1\tilde{\boldsymbol{\omega}}_2 - 420\tilde{\boldsymbol{\omega}}_6\tilde{\mathbf{q}}_7. \end{aligned} \tag{35}$$

New values for the constants that fix the initial conditions of the quaternion trajectory can be calculated similarly to the 5th-order polynomial, except an extra step needs to be taken to accommodate the third-order derivative of $\mathbf{q}$.

5. SOLVING THE PROBLEM USING IDVD METHOD

This section presents the results of using the IDVD method with two different parameterizations to obtain the minimum-time solutions of the problem posed in Section 2. As discussed in the previous section, as opposed to hundreds of varied parameters as in the case with the pseudospectral methods the list of parameters to be optimized using IDVD includes as few as 11 variables – boundary values of all three components of the angular acceleration plus five coefficients defining the chosen approximation of the virtual speed profile (18). Increasing the order of quaternion approximation polynomials from 5 to 7 allows assigning the boundary conditions for angular acceleration and varying the boundary values of angular jerks. (Note, with IDVD to satisfy the higher-order derivatives at the boundary points there is no need to introduce new equations of motion.) However, in what follows, for the 7th-order polynomial we will be varying both the angular jerk and angular acceleration. We'll do it with the only purpose of showing the improvement of the performance index to match that of the GPOPS solution.

During optimization the constraints are that the resulting control must obey Eq. (5) and λ_0 cannot be negative at any instant. Initial guesses for the angular acceleration and jerk

were taken to be equal to zero and the initial guess of 10^{-4} is used for all parameters contained in $\lambda(\tau)$.

IDVD Solutions using 5th-order Bezier Polynomial

Figures 9-11 present the results obtained when applying the IDVD with a quaternion based on a 5th-order Bezier polynomial (compare it with the GPOPS solution presented in Figs.1-3). The solution was run using 100 points (although as opposed to GPOPS it would make no difference running it for a larger or smaller number of points), and resulted in a slightly higher value of t_f, but took significantly less time to compute (to be discussed further, in Section 5). Figure 12 shows the virtual speed factor, the key element in matching the virtual and time domains.

The major difference compared to the GPOPS solution is that the controls do not have a bang-bang nature. Again, this was done intentionally by the choice of the quaternion parameterization. When implemented in the real-time controller, these controls may be easier to track. Also, having different controlling torques at the endpoints means having different angular accelerations. While for the GPOPS solution the initial and terminal angular accelerations are at the mercy of the optimization routine, using IDVD allows matching them with the current accelerations, making the control algorithm more robust.

Figure 9. Time histories of the states for Case 1.

Figure 10. Time history of controlling torques for Case 1.

The resulting solution for the same scenario using a lesser number of nodes, say 25 nodes, is shown in Figs.13-15. As in the case of GPOPS, due to symmetry it also converges to another equally optimal solution.

Figure 11. Principal axis outline of 180° slew for Case 1.

Figure 12. Mapping the virtual and time domains for the Case 1 solution.

Figure 13. Time histories of the states for Case 1 with 25 nodes.

Figure 14. Time history of controlling torques for Case 1 using 25 nodes.

Figure 15. Mapping the virtual and time domains for the Case 1 solution using 25 nodes.

With any number of nodes the IDVD solution results in a smooth control history, readily available to be fed forward to the tracking.

As in the case of the GPOPS solution for Case 2, the nonsymmetrical inertia matrix causes certain changes as compared to the symmetric matrix solution. The 100-node IDVD solution in this case results in 4.767s maneuver and requires about a minute to compute (see Figs.16-18).

Figure 16. Time histories of the states for Case 2.

Figure 17. Time history of controlling torques for Case 2.

IDVD Solutions using 7th-order Bezier Polynomial

For the sake of comparison, Figs. 19-21 present the solution of the same problem using a quaternion based on a 7th-order Bezier polynomial with angular acceleration and jerk varied at both ends at the trajectory. As seen, this brings a solution closer to that of GPOPS but doubles the computational time required to converge. The following section addresses this issue in more detail.

Figure 18. Mapping the virtual and time domains for the Case 2 solution.

6. IDVD VS. GPOPS RESULTS COMPARISON

This section presents a comparison of the results obtained using the IDVD method with those of the GPOPS method. It disregards the fact that the results obtained with GPOPS for low number of nodes are infeasible, but rather concentrates on the computational advantages the IDVD approach has for any number of intermediate points (nodes in the case of GPOPS). To start with, Tables 2 and 3 summarize the 180º rest-to-rest slew maneuver solutions for symmetric and asymmetric inertia matrix, obtained using GPOPS and IDVD as discussed in Sections 3 and 5.

In these tables all results are compared against the eigenaxis maneuver solution. First, it is shown that the true optimal solution, obtained offline in [3], which is not an eigenaxis rotation, provides about 8.5% and 11% improvement of the performance index, time of maneuver t_f, for Case 1 and Case 2, respectively. It would be great to exploit this economy solution onboard, but unfortunately it cannot be produced in real time and hence the need for other methods arise.

As seen from Tables 2 and 3 the GPOPS solution converging to one of the equally optimal solutions (if at all), assures about the same gain in the performance index as the truly optimal one. But again it takes too much computational time to be implemented onboard. Specifically, the 100-node solution that does converge and assures a smooth controls history, takes about on the order of an hour to converge. (It should be noted that the performance index for all solutions in this paper relates to the normalized states and controls, so that in reality, if we can afford several hours to perform a 180º slew maneuver, than GPOPS can be used onboard.)

a) b)

Figure 19. Time histories of the states for a quaternion based on a 7^{th}-order Bezier polynomial for Case 1 (a) and Case 2 (b) with 100 nodes.

a) b)

Figure 20. Time history of controlling torques for Case 1 (a) and Case 2 (b) with 100 nodes.

a)

b)

Figure 21. Mapping the virtual and time domains for Case 1 (a) and Case 2 (b) solutions with 100 nodes..

Table 2. The 180º rest-to-rest slew maneuver about the z-axis, symmetric inertia (Case 1).

Table 3. The 180º rest-to-rest slew maneuver about the z-axis, asymmetric inertia (Case 2).

As discussed in the previous section the IDVD solution has a much more robust performance, allowing computing the same type of maneuvers just in a few seconds as opposed to hours (using an executable optimization library the IDVD method produces solutions in fractions of a second [17]). Of course some of the optimality (performance index value) has to be sacrificed. On the positive side, the solution is always feasible and smooth for any number of computational points, and can be brought closer to the GPOPS solution (in terms of the value of a performance index) by increasing the number of varied parameters (the order of the quaternion approximation polynomial). Furthermore, IDVD has an analytic representation of the solution, which allows the number of nodes generated, possible for better tracking performance, to be increased without complex interpolation schemes or recalculating the entire solution.

To make it short and clear, the GPOPS candidate solution should use no less than 100 nodes, but about 10% gain in the performance index "costs" an order of an hour of CPU time. As discussed in Section 3, this solution features a bang-bang control, i.e. does not account for controllers' dynamics, and therefore can still not be used onboard as is. On the other hand, the always-feasible and ready-to-go IDVD solution (employing as low as say 25 computational points) can be produced much faster, but surrenders up to ⅔ of its gain as compared to that of the GPOPS solution (about ½ for the 7^{th}-order approximation).

Tables 4 and 5 present similar data for the 90º rest-to-rest slew maneuver. While GPOPS provides about 3% gain compared to a simple eigenaxis slew solution, the IDVD method has almost no advantage or may be even worse if using a 5^{th}-order quaternion approximation. All major conclusions, however, remain the same.

Table 4. The 90º rest-to-rest slew maneuver about the z-axis, symmetric inertia (Case 1).

*Solved recursively using previous 50 node solution as initial guess.

Table 6 compares GPOPS and IDVD solutions for one of such cases, when $\boldsymbol{\omega}_0 = -\boldsymbol{\omega}_f = \frac{1}{10}\mathbf{1}_{3\times 1}$ (other sets of non-zero boundary conditions were explored as well, and proved to maintain the same trends). In this table all results are compared against a valid 100-node GPOPS solution. As seen, the GPOPS solutions with a lesser number of nodes produce somewhat infeasible solutions, meaning that they cannot be implemented in the control scheme explicitly. The IDVD solutions may yield to GPOPS as much as about 4% with respect to the performance index, but again are produced much faster. Furthermore, while the 90º and 180 º rest-to-rest slew maneuvers with zero boundary rates feature multiple equally optimal solutions, so that both GPOPS and IDVD solutions converge to different solutions, when changing the number of nodes (GPOPS) / computational points (IDVD), in the case of non-zero boundary rates they all converge to the same solution as illustrated in Fig. 22.

Table 5. The 90º rest-to-rest slew maneuver about the z-axis, asymmetric inertia (Case 2).

It should be noted that in practice, the direct methods would likely be used in situations where the end-conditions (angular rates, accelerations) of the slew are specified and not equal to zero (to meet mission requirements of matching attitude rates of a tumbling vehicle, for example). For this case no simple eigenaxis slew solution exists and therefore any solution produced on-line would be good.

Table 6. The 90º maneuver for symmetric inertia (Case 1) and non-zero boundary rates.

a) b) c)

Figure 22. Projections of the 90º rotation maneuver for GPOPS (a), 5th-order (b), and 7th-order approximation (c) solutions.

The final observation is that apparently there is no need to use as complex of an approximation for the speed factor $\lambda(\tau)$ as that of Eq.(18). By looking at Figs. 12,15,18,21 it appears that the number of coefficients (varied parameters) can be easily reduced to 3 (which would result in even faster convergence, but at the cost of a slightly larger performance index):

$$\lambda(\tau) = \lambda_0 + a\tau + b\tau^2 . \tag{36}$$

In this case the optimization routine should assure the following inequalities hold:

$$\begin{aligned} &\lambda_0 > 0, \\ &\lambda_0 + a + b > 0, \\ &4b\lambda_0 - 2a(1+a) > 0, \text{ if } b > 0 \text{ and } -2b < a < 0. \end{aligned} \tag{37}$$

7. CONCLUSIONS

The IDVD method with the novel quaternion approximation functions proposed in this paper allows computing feasible solutions fast enough to be used onboard satellites for on-line computation of slew maneuvers. Moreover, because of the smooth controls' histories it can be implemented in the control schemes involving a feedforwad loop. Compared to the true time-optimal solutions the IDVD trajectories do not have a bang-bang control, which results in a slightly worse performance index. However, smooth controls benefit other mission preferences of having desired rates at the endpoints. In addition, it is a definite advantage in rapidly changing acquisition or tracking scenarios and when the slewing spacecraft possesses low frequency flexible modes. The formulation presented in this paper is currently being applied to situations where the attitude is coupled with other dynamics such as translational motion in rendezvous and docking applications. In this case, a simple eigenaxis slew would not meet mission criteria such as matching rotational motion (work in progress).

REFERENCES

[1] S. R. Vadali, and J. L. Junkins, "Optimal Open-Loop and Stable Feedback Control of Rigid Spacecraft Attitude Maneuvers," The Journal of the Astronautical Sciences, Vol. 32, No. 2, April-June 1984, pp.105-122.

[2] J. L. Junkins, and J. D. Turner, Optimal Spacecraft Rotational Maneuvers, Elsevier, 1986.

[3] 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.

[4] T. Davis, and S. Straight, "Development of the Tactical Satellite 3 for Responsive Space Missions," Proceedings of the 4th Responsive Space Conference, April 24-27, Los Angeles California, 2006.

[5] O. Yakimenko, Y. Xu, and G. Basset, "Computing Short-Time Aircraft Maneuvers Using Direct Methods," Proceedings of AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, HI, August 18-21, 2008.

[6] G. Boyarko, O. Yakimenko, and M. Romano "Formulation and Analysis of Matching Points of Interest in Two-Spacecraft for Optimal Rendezvous," Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Chicago, IL, August 10-13, 2009.

[7] G. Boyarko, O. Yakimenko, and M. Romano, "Modeling and Optimization of a Spacecraft Maneuvering with Respect to a Tumbling Object," Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, Pittsburgh, PA, August 9-13, 2009.

[8] C. R. McInnes, "Satellite Attitude Slew Maneuver using Inverse Control," The Aeronautical Journal, May 1998, pp.259-265.

[9] C. Louembet, F. Cazaurang, A. Zolghardi, C. Charbonnel, and C. Pittet, "Design of Algorithms for Satellite Slew Manoeuver by Flatness and Collocation," Proceedings of the American Control Conference, New York, NY, July 11-13, 2007.

[10] B. Wie, Space Vehicle Dynamics and Control, AIAA Education Series, 1998.

[11] D. T. Greenwood, Principles of Dynamics, Prentice Hall, 1987.

[12] A. Fleming, Real-time Optimal Slew Maneuver Design and Control. Astronautical Engineer's Thesis, US Naval Postgraduate School, 2004.

[13] I.M. Ross, User's Manual for DIDO: A MATLAB Application Package for Solving Optimal Control Problems, Tomlab Optimization, Sweden, Feb 2004.

[14] A. V. Rao, D. A. Benson, C. L. Darby, C. Francolin, M. A. Patterson, I. Sanders, and G. T. Huntington, "Algorithm: GPOPS, A Matlab Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method," ACM Transactions on Mathematical Software, Accepted for Publication, June 2009.

[15] O. Yakimenko, "Direct Method for Rapid Prototyping of Near Optimal Aircraft Trajectories," AIAA Journal of Guidance, Control, and Dynamics, 23(5), 2000, pp.865-875.

[16] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne and A. K. Cooke, "A Prototype of an Autonomous Controller for a Quadrotor UAV," European Control Conference 2007 Kos, 2-5 July, Kos, Greece, 2007.

[17] O. Yakimenko, and N. Slegers, "Using Direct Methods for Terminal Guidance of Autonomous Aerial Delivery Systems," Proceedings of the European Control Conference, Budapest, Hungary, August 23-26, 2009.

[18] M. Milam, "Real-Time Optimal Trajectory Generation for Constrained Dynamical Systems," Doctoral Thesis, California Institute of Technology, 2003.

[19] M. Kim, M., Kim, and S. 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, p. 369-376, 1995.

BIOGRAPHY

George Boyarko *is a research engineer with the Space Vehicles Directorate in the Simulation and Technology Assessment Branch and the Astrodynamics, GNC Team of the Air Force Research Laboratory, Kirtland AFB, NM. He is involved in modeling and simulation of vehicle subsystems, analysis supporting future concepts, and various aspects of guidance and control. Mr. Boyarko holds a B.S. in Chemical Engineering from the University of Cincinnati, Cincinnati, OH and a M.S. in Aerospace Engineering from Case Western Reserve University, Cleveland, OH. He is an Astronautical Engineering Ph.D. candidate at the Naval Postgraduate School focusing on spacecraft robotics, dynamics and control.*

Marcello Romano *received his M.S. and Ph.D. degrees in Astronautical Engineering from Politecnico di Milano University, Milan, Italy, in 1997 and 2001, respectively. Since 2004 he has been Assistant Professor at the Naval Postgraduate School, in the Department of Mechanical and Astronautical Engineering and in the Space Systems Academic Group. His research interests are in Spacecraft Dynamics, Guidance and Control, Robotics, and Nanosatellites Systems. He is PI and director of the Spacecraft Robotics Laboratory, and the Nanosatellites Advanced Concepts Laboratory. He is an author of more than 20 journal publications and two US patents. In 2006, he received the Menneken award for excellence in scientific research from the NPS foundation. He is an Associate Fellow of AIAA, and a Senior Member of IEEE.*

Oleg Yakimenko *received his M.S. degree in Aeronautical and Astronautical Engineering from the Moscow Institute of Physics and Technology, Moscow, Russia, in 1986. In 1988 he received another M.S. degree in Operations Research from the Air Force Engineering Academy, Moscow, Russia (AFEA). In the same academy he received the degree of the Candidate of Technical Sciences (Ph.D.) (1991) and Doctor of Technical Sciences (1996). He had progressed trough all professorial ranks at the AFEA and since late 1998 he has been a research professor at NPS. His research interests include Flight Mechanics; Optimal Control; Integrated Guidance, Navigation and Control with applications to Unmanned Systems. He is an author of more than 200 publications and an Associate Fellow of AIAA.*