2015

Performance evaluation of the inverse dynamics method for optimal spacecraft reorientation

Ventura, Jacopo

Ventura, Jacopo, Marcello Romano, and Ulrich Walter. "Performance evaluation of the inverse dynamics method for optimal spacecraft reorientation." Acta Astronautica 110 (2015): 266-278.

Acta Astronautica 110 (2015) 266–278

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Performance evaluation of the inverse dynamics method for optimal spacecraft reorientation

Jacopo Ventura [a,*], Marcello Romano [b], Ulrich Walter [a]

[a] Institute of Astronautics, Technische Universität München, Boltzmannstr. 15, 85748 Garching, Germany
[b] Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, Monterey, 93943-5107 CA, United States

A R T I C L E I N F O

Article history:
Received 15 July 2014
Received in revised form
25 November 2014
Accepted 30 November 2014
Available online 10 December 2014

Keywords:
Attiude maneuver
Optimization
Nonlinear programming
Spacecraft reorientation

A B S T R A C T

This paper investigates the application of the inverse dynamics in the virtual domain method to Euler angles, quaternions, and modified Rodrigues parameters for rapid optimal attitude trajectory generation for spacecraft reorientation maneuvers. The impact of the virtual domain and attitude representation is numerically investigated for both minimum time and minimum energy problems. Owing to the nature of the inverse dynamics method, it yields sub-optimal solutions for minimum time problems. Furthermore, the virtual domain improves the optimality of the solution, but at the cost of more computational time. The attitude representation also affects solution quality and computational speed. For minimum energy problems, the optimal solution can be obtained without the virtual domain with any considered attitude representation.

1. Introduction

The reorientation of a spacecraft is a common task in most space missions. For instance, slew maneuvers are required for targeting imaging equipment and sensors, orienting antenna towards Earth or reorienting the spacecraft for solar power absorption. However, in particular applications the spacecraft is required to reorient while minimizing a certain mission parameter, such as maneuver duration or fuel expenditure.

Rapid retargeting maneuvers, also called time-optimal reorientation maneuvers, are required by Earth imaging satellite in order to increase mission effectiveness. Rapid retargeting capability increases the image collection capacity during a given observation window, which is a key aspect for commercial applications or climate and natural disaster monitoring [1]. Moreover, pointing the entire spacecraft rather than sweep the imaging system improves the resolution of the image [2]. Military space missions may also require agile reorientation capabilities. For instance, the TacSat-3 was developed to demonstrate responsive delivery of information to operational military users [3]. The key feature required by the satellite was the capability to rapidly change its attitude due to the limited operational window for mission acquisition, tasking and information delivery. Finally, agile attitude maneuvers are also required by tracking satellites for pointing moving ground targets [2].

Current onboard control systems execute the maneuver by steering about a spacecraft eigenaxis since it represents the shortest angular path between two orientations. However, Karpenko et al. demonstrated with the TRACE spacecraft that this procedure is not time-optimal, and onboard control systems which optimize the maneuver are required [4].

The optimal reorientation of a spacecraft has been studied theoretically for both minimum time and minimum energy problems [5–7]. In particular, Bilimoria and Wie investigated the time optimal rest to rest reorientation of a symmetric spacecraft, showing that bang–bang control is optimal and the resulting motion has a significant nutational component [6].

Recently, the problem has been investigated numerically using two different direct optimization approaches: pseudospectral methods and inverse dynamics. Pseudospectral

* Corresponding author. Tel.: +49 89 289 16016.
E-mail address: jacopo.ventura@tum.de (J. Ventura).

http://dx.doi.org/10.1016/j.actaastro.2014.11.041
0094-5765/

methods are based on the numerical integration of the differential equations of motion and provide accurate solutions, but may converge slowly [8,9]. On the contrary, in inverse dynamics the trajectory is approximated by analytical functions. In most cases the solution is sub-optimal, but computational speed is high due to the reduced number of variable parameters. Therefore, the inverse dynamics approach is suitable for onboard applications. Louembert et al. first applied the inverse dynamics method to the optimal spacecraft reorientation problem using B-Splines to represent the modified Rodrigues parameters [10]. Recently, Boyarko et al. proposed a rapid attitude trajectory generation method based on the inverse dynamics in the virtual domain (IDVD) [11]. Here the quaternion components are approximated by polynomials defined in an abstract argument (virtual domain). Finally, Yakimenko applied IDVD to Euler angles [12]. However, no performance evaluation of the IDVD method has been conducted.

The objective of the present paper is to investigate the performance of the inverse dynamics method for rapid optimal spacecraft attitude trajectory generation by evaluating the effects of the attitude representation and usage of virtual domain on solution quality and computational speed of the algorithm. To evaluate the impact of the virtual domain, the attitude trajectory is approximated in both time and virtual domains using analogous polynomial functions. To evaluate the impact of the attitude representation, the IDVD method is applied, both in time and virtual domains, to Euler angles, quaternions, and modified Rodrigues parameters (MRP). An ideal scenario taken from Bilimoria and Wie is analyzed for both minimum time and minimum energy problems. Additional scenarios generated with Monte Carlo simulation are investigated for the minimum time problem.

The present paper is organized as follows: the first section describes the optimal spacecraft reorientation problem and introduces the inverse dynamics optimization approach, including IDVD. The application of IDVD to the Euler angles, quaternion and MRP is then described. Numerical experiments and conclusions are reported in the last sections of the paper.

2. Problem formulation and optimization approach

This section introduces the optimal spacecraft reorientation problem and the IDVD method for solving optimal control problems. The spacecraft is assumed rigid body.

2.1. Rotational motion of the spacecraft and optimization problem formulation

The dynamics of the rotational motion of a spacecraft is governed by Euler's equations, which in scalar form with the angular velocity $\boldsymbol{\omega}=[\omega_x, \omega_y, \omega_z]^T$ and the inertial tensor $\boldsymbol{I}=\text{diag}([I_{xx}, I_{yy}, I_{zz}])$ referenced to the body-fixed principal axes can be expressed as [13]

$$\begin{cases} \dot{\omega}_x = \dfrac{(I_{yy}-I_{zz})\omega_z\omega_y + T_x}{I_{xx}} \\ \dot{\omega}_y = \dfrac{(I_{zz}-I_{xx})\omega_z\omega_x + T_y}{I_{yy}} \\ \dot{\omega}_z = \dfrac{(I_{xx}-I_{yy})\omega_y\omega_x + T_z}{I_{zz}} \end{cases} \tag{1}$$

In Eq. (1) $\mathbf{T}=[T_x, T_y, T_z]^T$ represents the component of the bounded external torque vector referenced to the body frame.

The kinematics of the rotational motion of a spacecraft can be described using different attitude representations [14]. In terms of quaternion $\boldsymbol{q}=[q_1,q_2,q_3,q_4]^T$, the kinematic differential equation is given by

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

where $\boldsymbol{\Omega}=[\omega_x, \omega_y, \omega_z, 0]^T$ and the symbol $\otimes$ denotes the (Hamiltonian) product between quaternions [14]. In terms of Euler angles $\boldsymbol{\theta}=[\theta_1, \theta_2, \theta_3]^T$ of the rotational sequence 1–2–3, the kinematic differential equation is

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \frac{1}{\cos\theta_2}\begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3\cos\theta_2 & \cos\theta_3\cos\theta_2 & 0 \\ -\cos\theta_3\sin\theta_2 & \sin\theta_3\sin\theta_2 & 1 \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}. \tag{3}$$

Finally, using the MRP $\boldsymbol{\sigma}=[\sigma_1, \sigma_2, \sigma_3]^T$, the kinematics of the rotational motion can be described by the following equation

$$\begin{aligned} \begin{bmatrix} \dot{\sigma}_1 \\ \dot{\sigma}_2 \\ \dot{\sigma}_3 \end{bmatrix} &= \frac{1}{4}\begin{bmatrix} 1+\sigma_1^2-\sigma_2^2-\sigma_3^2 & 2(\sigma_1\sigma_2-\sigma_3) & 2(\sigma_1\sigma_3+\sigma_2) \\ 2(\sigma_1\sigma_2+\sigma_3) & 1-\sigma_1^2+\sigma_2^2-\sigma_3^2 & 2(\sigma_2\sigma_3-\sigma_1) \\ 2(\sigma_1\sigma_3-\sigma_2) & 2(\sigma_2\sigma_3+\sigma_1) & 1-\sigma_1^2-\sigma_2^2+\sigma_3^2 \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \\ &= \frac{1}{4}\boldsymbol{B}(\boldsymbol{\sigma})\boldsymbol{\omega}. \end{aligned} \tag{4}$$

The optimal spacecraft reorientation problem consists of minimizing a (given) cost function J by finding the optimal control vector $\boldsymbol{T}$, subjected to the control constraints $\boldsymbol{T}_{min} \le \boldsymbol{T} \le \boldsymbol{T}_{max}$, that brings the system described by Eq. (1) and Eqs. (2)–(4) from an initial state of angular velocity $\boldsymbol{\omega}_0$ and attitude $\mathbf{q}_0$, $\boldsymbol{\theta}_0$ or $\boldsymbol{\sigma}_0$ to a final state of angular velocity $\boldsymbol{\omega}_F$ and attitude $\boldsymbol{q}_F$, $\boldsymbol{\theta}_F$ or $\boldsymbol{\sigma}_F$. The cost function J is defined as

$$J = \int_{t_0}^{t_F} \mathrm{d}t \tag{5}$$

for minimum time maneuvers and

$$J = \frac{1}{2}\int_{t_0}^{t_F} (T_x^2+T_y^2+T_z^2)\,\mathrm{d}t \tag{6}$$

for minimum quadratic control (or energy) expenditure.

2.2. The inverse dynamics and inverse dynamics in the virtual domain methods

In the inverse dynamics approach for rapid trajectory generation the optimal control problem is converted into an equivalent nonlinear programming problem by describing the trajectory components with a set of polynomial functions defined in the time domain. The cost is then minimized through the optimization of the polynomial coefficients.

The inverse dynamics in the virtual domain method (IDVD) follows the inverse dynamics approach by defining the polynomials for trajectory representation in a virtual domain τ [15]. This permits to decouple space and time optimization and, consequently, a given trajectory can be followed using the different velocity profiles. By inverting the dynamics, the system state variables and controls can be expressed as function of the trajectory z, defined in the virtual domain, and a finite number of its derivatives. For the considered spacecraft reorientation problem it results in

$$\boldsymbol{\omega}(\tau)=f_1(z,z'), \quad \boldsymbol{T}(\tau)=f_2(z,z') \tag{7}$$

Virtual domain and time domain are related to each other by [15]

$$t(\tau_F)=\int_0^{\tau_F}\frac{\mathrm{d}\tau}{\lambda(\tau)}, \tag{8}$$

Where

$$\lambda(\tau)=\frac{\mathrm{d}\tau}{\mathrm{d}t} \tag{9}$$

is the so called speed factor, which is also expressed by a polynomial in the virtual domain. The inversion of the dynamics through Eq. (7) and the numerical integration of the cost function J reduce the problem into an equivalent nonlinear programming problem, whose variable parameters are the polynomial coefficients.

In the particular case of $\lambda(\tau)=1$, the time variable t is identical to the abstract argument τ and the problem is defined in the time domain. Therefore, the IDVD is reduced to the basic inverse dynamics method.

3. Inverse dynamics with Euler angles

This section describes the inverse dynamics method applied to the Euler angles of the rotational sequence 1–2–3.

3.1. Inverse dynamics in the virtual domain

In computer graphics and animation smooth curves are represented by parametric functions such as Splines or Bezier curves for their capability to approximate complex shapes through curve fitting and computational efficiency [16–18].

Given a set of control points $p_0, p_1, p_2, \ldots, p_n$ in $\mathbf{R}$, the Bezier curve $p(\tau)$ which interpolates these points is defined as [16]

$$p(\tau)=\sum_{i=0}^{n}p_i\beta_{i,n}(\tau), \tag{10}$$

where $\beta_{i,n}(\tau)$ is the nth order Bernstein polynomials

$$\beta_{i,n}(\tau)=\binom{n}{i}(1-\tau)^{n-i}\tau^i \tag{11}$$

acting as basis form of $p(\tau)$ and $\tau\in[0;\ 1]$ the virtual argument. This formulation yields to the following properties at the endpoints, which are employed by IDVD to define the trajectory with the Bezier curve [11,17]:

1) $p(0)=p_0$ and $p(1)=p_n$;
2) $\mathrm{d}p(0)/\mathrm{d}\tau=n(p_1-p_0)$ and $\mathrm{d}p(1)/\mathrm{d}\tau=n(p_n-p_{n-1})$.

Following this approach, each Euler angle is represented in the virtual domain $\tau\in[0;\ 1]$ by a Bezier curve as follows:

$$\theta_j(\tau)=\sum_{i=0}^{n}a_{j,i}\beta_{i,n}(\tau), \quad j=1,2,3 \tag{12}$$

where $a_{j,i}$ are the polynomial coefficients acting as control points and $\beta_{i,n}(\tau)$ the Bernstein base as in Eq. (11). The map between the time domain and the virtual domain is defined by the speed factor $\lambda(\tau)$, which is expressed through a polynomial of order m [11].

$$\lambda(\tau)=\sum_{i=0}^{m}b_i\tau^i. \tag{13}$$

The corresponding points in the time domain are calculated with Eq. (8). As a result of this mapping, the time derivatives of the angles, expressed as function of τ, are obtained from the virtual derivatives according to

$$\dot{\theta}_j(\tau)=\frac{\mathrm{d}\theta_j(\tau)}{\mathrm{d}\tau}\frac{\mathrm{d}\tau}{\mathrm{d}t}=\frac{\mathrm{d}\theta_j(\tau)}{\mathrm{d}\tau}\lambda, \quad j=1,2,3 \tag{14}$$

$$\ddot{\theta}_j(\tau)=\frac{\mathrm{d}\dot{\theta}_j(\tau)}{\mathrm{d}\tau}\frac{\mathrm{d}\tau}{\mathrm{d}t}=\frac{\mathrm{d}^2\theta_j(\tau)}{\mathrm{d}\tau^2}\lambda^2+\frac{\mathrm{d}\theta_j(\tau)}{\mathrm{d}\tau}\frac{\mathrm{d}\lambda}{\mathrm{d}\tau}\lambda, \quad j=1,2,3 \tag{15}$$

By imposing attitude, angular velocity and acceleration at the endpoints, the polynomial coefficients $a_{j,i}$ can be defined in such a way that these boundary conditions are automatically satisfied. By exploiting the properties of the Bezier curve at the endpoints, it results

$$\begin{cases} a_{j,0}=\theta_{0j} \\ a_{j,1}=\frac{\dot{\theta}_{0j}}{n\lambda(0)}+a_{j,0} \\ a_{j,2}=\frac{1}{2}\binom{n}{2}^{-1}\frac{\ddot{\theta}_{0j}-\dot{\theta}_{0j}\lambda'(0)}{\lambda^2(0)}+2a_{j,1}-a_{j,0} \end{cases} \quad j=1,2,3 \tag{16}$$

for the initial conditions and

$$\begin{cases} a_{j,n}=\theta_{Fj} \\ a_{j,n-1}=-\frac{\dot{\theta}_{Fj}}{n\lambda(1)}+a_{j,n} \\ a_{j,n-2}=\frac{1}{2}\binom{n}{n-2}^{-1}\frac{\ddot{\theta}_{Fj}-\dot{\theta}_{Fj}\lambda'(1)}{\lambda^2(1)}+2a_{j,n-1}-a_{j,n} \end{cases} \quad j=1,2,3 \tag{17}$$

for the final conditions. In the previous expressions λ' is the virtual derivative of the speed factor, while θ_{0j} and θ_{Fj} denote the jth Euler angle evaluated at the initial and final time respectively. The time derivatives of the angles are obtained with Eq. (3) and its time derivative, calculated at the endpoints. As a consequence, the degree n of the Bezier curves is related to the total number N_B of boundary conditions according to [19]

$$n\geq N_B+1. \tag{18}$$

In this way, at least one polynomial coefficient for each angle is left as variable parameter for the optimization, allowing flexibility of the shape of the trajectory.

At this point it is important to note that the singularity in Eq. (3) limits the application of the proposed algorithm: the trajectories having $\theta_2=\pm 90$ deg as final attitude cannot be represented by the parametric curve since the coefficients $a_{j,n\text{-}1}$ and $a_{j,n\text{-}2}$ defined in Eq. (17) cannot be determined.

The expressions of angular velocity and controls can be obtained from the angles by inverting the dynamics. Inversion of Eq. (3) provides the angular velocity $\boldsymbol{\omega}$

$$\begin{cases} \omega_x = \cos\theta_2 \cos\theta_3 \dot{\theta}_1 + \sin\theta_3 \dot{\theta}_2 \\ \omega_y = -\cos\theta_2 \sin\theta_3 \dot{\theta}_1 + \cos\theta_3 \dot{\theta}_2 \,, \\ \omega_z = \sin\theta_2 \dot{\theta}_1 + \dot{\theta}_3 \end{cases} \tag{19}$$

where the time derivatives of the angles are calculated using Eq. (14). The angular accelerations are obtained from the time derivative of Eq. (19) along with Eq. (15). Finally, the inversion of Eq. (1) provides the control torques

$$\begin{cases} T_x = I_{xx}\dot{\omega}_x + (I_{zz} - I_{yy})\omega_z\omega_y \\ T_y = I_{yy}\dot{\omega}_y + (I_{xx} - I_{zz})\omega_z\omega_x \,. \\ T_z = I_{zz}\dot{\omega}_z + (I_{yy} - I_{xx})\omega_x\omega_y \end{cases} \tag{20}$$

Note that the obtained expressions of angular velocities and torques are functions of the virtual argument τ and the polynomial coefficients.

The proposed procedure allows to transform the cost J of the optimization problem into a (nonlinear) function J $(a_{j,i}, b_i)$ of the polynomial coefficients: Eq. (5) is integrated numerically after substitution of Eqs. (9) and (13) for the minimum time problem, while Eq. (6) is integrated after substitution of the expression of the controls for the minimum energy problem.

The optimal control problem is consequently converted into the following equivalent nonlinear programming problem: Minimize the function $J(a_{j,i}, b_i)$ subjected to the constraints

$$\begin{cases} \boldsymbol{T}_{min} \le \boldsymbol{T}(\tau_i) \le \boldsymbol{T}_{max} \\ t(\tau_i) \ge 0 \\ t(1) \le t_{F\,max}, \end{cases} \tag{21}$$

and the variables of the problem being the polynomial coefficients $a_{j,i}$ and b_i. The constraints are evaluated at several nodes τ_i of the virtual domain. The higher this number, the more accurate the enforcement of the bounds, but at the cost of more computational time. The second constraint in Eq. (21) imposes Eq. (8) to map the points of the virtual domain into positive points of the time domain. The third condition is applied to Eq. (8) and limits the maximum maneuver duration to t_{Fmax}.

3.2. Inverse dynamics in the time domain

In the particular case of $\lambda(\tau) = 1$, the time variable t is identical to the abstract argument τ and the problem is defined in the time domain. Thus, IDVD is reduced to the basic inverse dynamics method. The Euler angles can be represented in the time domain by Eq. (12) after imposing $\tau = t/t_F$, with t_F final maneuver time and variable of the problem. However, preliminary analysis showed that high-order polynomials are computationally more efficient than modified Eq. (12). Therefore, in this paper the Euler angles are expressed in the time domain $t \in [0;\ t_F]$ by nth order polynomials

$$\theta_j(t) = \sum_{i=0}^{n} a_{j,i} t^i \quad j = 1, 2, 3. \tag{22}$$

The time derivatives of the angles are given by the successive polynomial derivatives

$$\dot{\theta}_j(t) = \sum_{i=0}^{n} i a_{j,i} t^{i-1}, \quad \ddot{\theta}_j(t) = \sum_{i=0}^{n} i(i-1) a_{j,i} t^{i-2} \quad j = 1, 2, 3. \tag{23}$$

The boundary conditions are respected using the following expressions for the coefficients:

$$\begin{cases} a_{j,0} = \theta_{0j} \\ a_{j,1} = \dot{\theta}_{0j} \,, \quad j = 1, 2, 3 \\ a_{j,2} = \frac{\ddot{\theta}_{0j}}{2} \end{cases} \tag{24}$$

$$\begin{cases} a_{j,n} = \dfrac{\theta_{Fj} - \sum\limits_{i=0}^{n-1} a_{j,i} t_F^i}{t_F^n} \\ a_{j,n-1} = \dfrac{\dot{\theta}_{Fj} - \sum\limits_{i=0}^{n-2} i a_{j,i} t_F^{i-1} - n a_{j,n} t_F^{i-1}}{(n-1) t_F^{n-2}} \\ a_{j,n-2} = \dfrac{\ddot{\theta}_{Fj} - \sum\limits_{i=0}^{n-3} i(i-1) a_{j,i} t_F^{i-2} - \sum\limits_{i=n-1}^{n} i(i-1) a_{j,i} t_F^{i-2}}{(n-2)(n-3) t_F^{n-4}} \end{cases} , \quad j = 1, 2, 3 \tag{25}$$

The previous relationships are obtained by imposing the boundary conditions $\boldsymbol{\theta}_0$, $\dot{\boldsymbol{\theta}}_0$, $\ddot{\boldsymbol{\theta}}_0$ and $\boldsymbol{\theta}_F$, $\dot{\boldsymbol{\theta}}_F$, $\ddot{\boldsymbol{\theta}}_F$ to Eqs. (22) and (23). Therefore the order n of the polynomial in Eq. (22) must satisfy

$$n \ge N_B. \tag{26}$$

The expressions of angular velocity and controls are obtained from the angles and their derivatives with Eqs. (19) and (20) respectively. Using a numerical integration method, the cost function J is transformed into a (nonlinear) function $J(a_{j,i}, b_i, t_F)$ of the polynomials coefficients and final maneuver time.

The optimal control problem is therefore converted into the following equivalent nonlinear programming problem: minimize the function $J(a_{j,i}, b_i, t_F)$ subject to the constraints

$$\begin{cases} \boldsymbol{T}_{min} \le \boldsymbol{T}(t_i) \le \boldsymbol{T}_{max} \\ 0 \le t_F \le t_{F\,max}, \end{cases} \tag{27}$$

with the final time t_F and the polynomial coefficients $a_{j,i}$ and b_i being the variables of the problem.

4. Inverse dynamics with quaternions

The IDVD method applied to quaternions as proposed by Boyarko et al. is outlined in the first paragraph of this section [11]. From this formulation the particular case of IDVD in time domain is then derived.

4.1. Inverse dynamics in the virtual domain

Following the parameterization approach employed for the Euler angles, the attitude trajectory in terms of quaternions can be represented by a set of control points and basis functions. However, the parameterization must ensure the nonlinear unit norm condition along the trajectory and therefore Eq. (12) cannot be employed directly. For this reason, the parametric curve proposed by Kim et al. and successively applied by Boyarko et al. was chosen [11,16].

Given two unit quaternions $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, the quaternion curve $\boldsymbol{\gamma}(\tau)$ which connects $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ can be defined as

$$\boldsymbol{\gamma}(\tau) = \boldsymbol{q}_1 \otimes exp(\boldsymbol{\omega}^* \beta(\tau)), \tag{28}$$

where $\beta(\tau)$ is the basis function of the curve, $\tau \in [0;\ 1]$ the virtual argument and

$$\boldsymbol{\omega}^* = log(\mathbf{q}_1^{-1} \otimes \mathbf{q}_2) \tag{29}$$

a vector defined in $\mathbf{R}^3$ whose norm is the Euler's rotation angle defined by the quaternion $\boldsymbol{q}_1^{-1} \otimes \boldsymbol{q}_2$ and direction the Euler's axis. The exponential and natural logarithmic maps are applied according to the following definition:

$$\boldsymbol{q} = exp(\boldsymbol{\omega}^*) = \begin{cases} \begin{bmatrix} sin(\boldsymbol{\omega}^*)\frac{\boldsymbol{\omega}^*}{|\boldsymbol{\omega}^*|} & cos(|\boldsymbol{\omega}^*|) \end{bmatrix}^T & |\boldsymbol{\omega}^*| \neq 0 \\ \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T & |\boldsymbol{\omega}^*| = 0 \end{cases}, \tag{30}$$

$$\boldsymbol{\omega}^* = log(\boldsymbol{q}) = \begin{cases} \left(\frac{arccos(q_4)}{\sqrt{q_1^2+q_2^2+q_3^2}}\right) \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T & \sqrt{q_1^2+q_2^2+q_3^2} \neq 0 \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T & \sqrt{q_1^2+q_2^2+q_3^2} = 0 \end{cases}. \tag{31}$$

Note that $|exp(\boldsymbol{\omega}^*)| = 1 \,\forall \boldsymbol{\omega}^*$ and therefore Eq. (28) automatically satisfies the unit norm constraint.

Therefore, the quaternion trajectory curve $\boldsymbol{q}(\tau)$ can be represented by a set of parametric curves connecting a sequence of control points $\tilde{\mathbf{q}}_i$ according to [16]

$$\boldsymbol{q}(\tau) = \boldsymbol{q}_0 \otimes \prod_{i=1}^{n} \exp(\boldsymbol{\omega}_i^* \tilde{\beta}_{i,n}(\tau)), \tag{32}$$

where

$$\tilde{\beta}_{i,n}(\tau) = \sum_{j=i}^{n} \binom{n}{j} (1-\tau)^{n-j} \tau^j \tag{33}$$

and

$$\boldsymbol{\omega}_i^* = \log(\tilde{\boldsymbol{q}}_{i-1}^{-1} \otimes \tilde{\boldsymbol{q}}_i). \tag{34}$$

In Eq. (32) the symbol $\prod$ denotes the sequential quaternion product and the control points satisfy the following conditions at endpoints:

$$\tilde{\boldsymbol{q}}_0 = \boldsymbol{q}_0, \quad \tilde{\boldsymbol{q}}_n = \boldsymbol{q}_F\,.$$

The speed factor $\lambda(\tau)$ is defined with a polynomial as in Eq. (13). The virtual derivatives of Eq. (32) are calculated using the chain rule and their expressions can be found in [11]. The time derivatives of the quaternion vector are calculated from its virtual derivatives using Eqs. (14) and (15) after substituting the jth angle θ_j (and its derivatives) with the quaternion vector $\boldsymbol{q}$.

To allow the correct optimization using this approach, all the control points $\tilde{\mathbf{q}}_i$ and the vectors $\boldsymbol{\omega}^*$ must be defined using the boundary conditions of the problem [11]. Consequently, the parameters to be optimized consist of the additional boundary conditions introduced to define all the $\tilde{\mathbf{q}}_i$ and $\boldsymbol{\omega}_i^*$ along with the polynomial coefficients of $\lambda(\tau)$. The total number N_B of boundary conditions required to define all the parameters is therefore

$$N_B = n+1. \tag{35}$$

In this paper, the quaternion history is represented with a fifth-order Bezier function (and therefore a product of five exponential functions). Consequently, six boundary conditions are required to define all the control points and vectors $\boldsymbol{\omega}_i^*$. These conditions are initial and final attitude, angular velocity and angular acceleration.

By exploiting the properties of the fifth-order Bezier polynomial and imposing the boundary conditions, the vectorial terms $\boldsymbol{\omega}_i^*$ are defined through

$\boldsymbol{\Omega}_i^* = \begin{bmatrix} \omega_{ix}^* & \omega_{iy}^* & \omega_{iz}^* & 0 \end{bmatrix}^T$ with the following expressions:

$$\begin{cases} \boldsymbol{\Omega}_1^* = \frac{\boldsymbol{q}_0^{-1} \otimes \dot{\boldsymbol{q}}_0}{5} \\ \boldsymbol{\Omega}_2^* = \frac{\boldsymbol{q}_0^{-1} \otimes \ddot{\boldsymbol{q}}_0 - 2\left[\binom{5}{2} - 5(5-1)\right]\boldsymbol{\Omega}_1^* - 5^2 \boldsymbol{\Omega}_1^* \otimes \boldsymbol{\Omega}_1^*}{2\binom{5}{2}} \end{cases}, \tag{36}$$

$$\begin{cases} \boldsymbol{\Omega}_5^* = \frac{\boldsymbol{q}_F^{-1} \otimes \dot{\boldsymbol{q}}_F}{5} \\ \boldsymbol{\Omega}_4^* = \frac{\ddot{\boldsymbol{q}}_4^{-1} \otimes \left[\ddot{\boldsymbol{q}}_F - 5^2 \boldsymbol{q}_F \otimes \boldsymbol{\Omega}_5^* \otimes \boldsymbol{\Omega}_5^* - 2\binom{5}{5-2}\boldsymbol{q}_F \otimes \boldsymbol{\Omega}_5^*\right] \otimes \boldsymbol{q}_F^{-1} \otimes \tilde{\boldsymbol{q}}_4}{2\left[\binom{5}{5-2} - 5(5-1)\right]} \end{cases}. \tag{37}$$

Note that Eqs. (36) and (37) are written in general form and can be applied to any order n of the Bezier polynomial by substituting 5 with n. All the control points $\tilde{\mathbf{q}}_i$ and the coefficient $\boldsymbol{\Omega}_3^*$ are then calculated using Eq. (34). The expressions of angular velocity and accelerations are obtained with the inversion of Eq. (2) and its derivative

$$\boldsymbol{\Omega} = 2\boldsymbol{q}^{-1} \otimes \dot{\boldsymbol{q}}, \quad \dot{\boldsymbol{\Omega}} = 2\boldsymbol{q}^{-1} \otimes \ddot{\boldsymbol{q}} + \frac{1}{2}\boldsymbol{\Omega} \otimes \boldsymbol{\Omega}. \tag{38}$$

The expressions of the torques are obtained with Eq. (20).

By numerically integrating the cost function J, the optimal control problem is converted into an equivalent nonlinear programming problem, whose variables are the additional boundary conditions introduced to define all the parameter of Eq. (32) along with the polynomial coefficients of Eq. (13).

4.2. Inverse dynamics in the time domain

By imposing $\lambda(\tau) = 1$, the problem is formulated in the time domain $t \in [0;\ t_F]$, where t_F is set as variable parameter of the problem. Consequently, Eq. (32) is modified as

$$\boldsymbol{q}(t) = \boldsymbol{q}_0 \otimes \prod_{i=1}^{n} exp(\boldsymbol{\omega}_i^* \tilde{\beta}_{i,n}(t)), \tag{39}$$

with

$$\tilde{\beta}_{i,n}(t) = \sum_{j=i}^{n} \binom{n}{j} \left(1 - \frac{t}{t_F}\right)^{n-j} \left(\frac{t}{t_F}\right)^j. \tag{40}$$

Note that the unit norm constraint and the property of the Bezier polynomial at endpoints are still preserved. The transformation of the problem into the equivalent nonlinear programming problem is analogous to IDVD with virtual domain. Due to Eqs. (39) and (40), the (general) expressions of the coefficients $\boldsymbol{\omega}_i^*$ for a fifth-order Bezier curve are given by

$$\begin{cases} \boldsymbol{\Omega}_1^* = \frac{t_F \boldsymbol{q}_0^{-1} \otimes \dot{\boldsymbol{q}}_0}{5} \\ \boldsymbol{\Omega}_2^* = \dfrac{\boldsymbol{q}_0^{-1} \otimes \ddot{\boldsymbol{q}}_0 - \frac{2}{t_F^2}\left[\binom{5}{2} - 5(5-1)\right]\boldsymbol{\Omega}_1^* - \left(\frac{2}{t_F}\right)^2 \boldsymbol{\Omega}_1^* \otimes \boldsymbol{\Omega}_1^*}{\frac{2}{t_F^2}\binom{5}{2}}, \end{cases} \tag{41}$$

$$\begin{cases} \boldsymbol{\Omega}_5^* = \frac{t_F \boldsymbol{q}_F^{-1} \otimes \dot{\boldsymbol{q}}_F}{5} \\ \boldsymbol{\Omega}_4^* = \dfrac{\tilde{\boldsymbol{q}}_4^{-1} \otimes \left[\ddot{\boldsymbol{q}}_F - \left(\frac{2}{t_F}\right)^2 \boldsymbol{q}_F \otimes \boldsymbol{\Omega}_5^* \otimes \boldsymbol{\Omega}_5^* - \frac{2}{t_F^2}\binom{5}{5-2}\boldsymbol{q}_F \otimes \boldsymbol{\Omega}_5^*\right] \otimes \boldsymbol{q}_F^{-1} \otimes \tilde{\boldsymbol{q}}_4}{\frac{2}{t_F^2}\left[\binom{5}{5-2} - 5(5-1)\right]}. \end{cases} \tag{42}$$

5. Inverse dynamics with modified Rodrigues parameters

The IDVD method is applied to the MRP similarly to the Euler angles. Each MRP is represented in the virtual domain $\tau \in [0;\ 1]$ with an nth order Bezier curve:

$$\sigma_j(\tau) = \sum_{i=0}^{n} a_{j,i}\beta_{i,n}(\tau) \quad j = 1, 2, 3. \tag{43}$$

The speed factor $\lambda(\tau)$ is defined with Eq. (13). The boundary conditions are imposed with Eqs. (16) and (17) after substituting the jth angle θ_j (and its derivatives) with the jth MRP σ_j. The required initial and final time derivatives of the MRP are obtained with Eq. (4) and its derivative calculated at the endpoints.

By inverting the kinematic Eq. (4) and differentiating the obtained expression, results in the following compact expression of the angular velocity and acceleration [14]:

$$\boldsymbol{\omega} = 4\frac{1}{\left(1+|\boldsymbol{\sigma}|^2\right)^2}\boldsymbol{B}(\boldsymbol{\sigma})^T\dot{\boldsymbol{\sigma}}, \tag{44}$$

$$\dot{\boldsymbol{\omega}} = \frac{1}{\left(1+|\boldsymbol{\sigma}|^2\right)^2}\boldsymbol{B}(\boldsymbol{\sigma})^T\left[4\ddot{\boldsymbol{\sigma}} - \dot{\boldsymbol{B}}(\boldsymbol{\sigma})\boldsymbol{\omega}\right]. \tag{45}$$

The time derivatives of the MRP are obtained from the virtual derivatives with Eqs. (14) and (15) by substituting the angle θ_j (and its derivatives) with σ_j. The control torques are obtained with Eq. (20).

Similarly to IDVD with Euler angles, the optimal control problem is consequently converted into an equivalent nonlinear programming problem, with constraints given by Eq. (21). The application of the method in the time domain is analogous to the case with the Euler angles.

6. Numerical experiments

The numerical results of comparison among the IDVD algorithms for rapid optimal spacecraft reorientation are presented in this section.

Several test scenarios have been considered in non-dimensional form as in Bilimoria and Wie [6]. The duration of each maneuver is limited in the interval [2 10] s. All the scenarios addressed in this paper are rest to rest, but IDVD can be extended to any angular velocity at endpoints [11].

The proposed algorithms have been implemented in MatLab on a Windows 7 laptop computer with an Intel 2.27 GHz i5 M430 processor and 4 Gb of RAM. The nonlinear programming problem generated by IDVD is solved using the Sequential Quadratic Solver method (SQP) of MatLab *fmincon* [20]. A total number of 50 nodes is considered for imposing the constraints of the optimization problem. The IDVD applied to virtual domain employs the Gauss–Legendre Quadrature method with 4 points to integrate Eq. (8). Reference solutions are computed with the Gauss Pseudospectral Optimization Software GPOPS [21]. The obtained controls are propagated with MatLab ODE 45 using Eqs. (1) and (2) and a time step of 0.01 s. The errors in magnitude of final angular velocity and attitude after propagation of the controls are denoted with $\Delta\omega$ and $\Delta\theta$ respectively.

The performance of the IDVD method are evaluated in terms of effects of the virtual domain, attitude representation and order of polynomials on solution quality and computational speed required by the algorithm to converge to an optimal solution. In particular, effects of the virtual domain are evaluated by solving a single optimization problem in the virtual domain and in the time domain separately, using an analogous polynomial to represent the attitude trajectory.

To compare the results of the considered algorithms in terms of solution optimality, the difference in percentage ΔJ is defined as follows:

$$\Delta J = 100\frac{|J_{IDVD} - J_{GPOPS}|}{J_{GPOPS}}, \tag{46}$$

where J_{IDVD} is the optimal cost function computed by IDVD and J_{GPOPS} the optimal cost computed with GPOPS taken as reference solution.

The overall performance of IDVD is determined in terms of computational time *CPUt* required by the algorithm to converge and solution quality ΔJ according to the following performance index:

$$MI = K_J \Delta J + K_{CPU} CPUt, \tag{47}$$

where K_J and K_{CPU} are weighting coefficients for solution quality and computational time respectively. Since ΔJ and *CPUt* have the same order of magnitude, the weights are defined in the range [0; 1]. Different combinations of weighting values are considered to analyze the performance of IDVD. Note that the best possible performance of the algorithm is obtained when ΔJ and *CPUt* are both close to zero. Therefore, the lower the performance index, the better the overall performance.

6.1. Minimum time maneuver: 180 deg reorientation of a symmetric spacecraft

The maneuver addressed in this section is a minimum time 180 deg reorientation about the z-axis of a symmetric spacecraft with the following conditions [6]:

$$\boldsymbol{I} = \mathrm{diag}([1, 1, 1])$$
$$-1 \le \boldsymbol{T} \le 1$$

$$\boldsymbol{\omega}_0 = [0,0,0] \quad \boldsymbol{\theta}_0 = [0,0,0]deg \quad \boldsymbol{q}_0 = [0,0,0,1] \quad \boldsymbol{\sigma}_0 = [0,0,0]$$

$$\boldsymbol{\omega}_F = [0,0,0] \quad \boldsymbol{\theta}_F = [0,0,180]deg \quad \boldsymbol{q}_F = [0,0,1,0] \quad \boldsymbol{\sigma}_F = [0,0,1]. \tag{48}$$

Initial guesses for the controls at endpoints are taken to be equal to one, which is the maximum torque. Accordingly, the guesses on the polynomial coefficients related to angular accelerations are given with Eqs. (16), (17), (24), (25), (36), (37), (41), (42); initial guess on the optimal maneuver time is 2 s, which is the minimum allowable maneuver time. The first guess for b_1 is the inverse of the guess of the maneuver time, while initial guesses for all the other polynomial coefficients are zero as in Ref. [11].

Table 1 summarizes the results of the optimizations. The GPOPS method computes the optimal solution found by Bilimoria and Wie with all the considered attitude representations. The optimal controls have a bang–bang nature (see Fig. 1) and the consequent rotation has a significant nutational component, evident from the large values of the angle θ_1 and θ_2 represented in Fig. 2.

The solutions obtained with IDVD are sub-optimal, for every attitude parameter, polynomial order and domain considered, and the resulting optimal controls have always a smooth time history instead of bang–bang, as shown in Fig. 1. This is due to the nature of the method: the controls are calculated as combination of the trajectory components and their derivatives through the inversion of the dynamics (see Eqs. (19), (20), (38), (44), (45)). Since the trajectory is represented with polynomials, the controls result nonlinear continuous function defined in the virtual argument or time. Notably, the controls obtained with this approach are always continuous since the trajectory components must be represented at least with class C^2 functions (see the conditions in Eqs. (18) and (26)). Therefore, a bang–bang shape of the controls cannot be achieved and only sub-optimal solutions can be obtained. Better solutions can be computed by increasing the order of the polynomials for trajectory representation, but at the cost of higher computational time.

The definition of the trajectory in the virtual domain improves the sub-optimal solution with respect to the same polynomial formulation in the time domain, for every attitude representation and polynomial order considered. In fact, the speed factor introduces additional variables to the problem, which allow optimization of the velocity profile along the trajectory (see also Eqs. (14) and (15)). Fig. 1 graphically represents the effect of the virtual domain on the control time histories. The introduction of the speed factor modifies the shape of the controls, which result closer to the bang–bang optimal shape. The other effect of the virtual domain is the increment of computational time required by the solver due to the additional variable parameters, additional constraints (see Eq. (21)) and integration of Eq. (8) at every iteration of the SQP algorithm. The effect of the parameterization in the virtual domain can be observed also in Fig. 2: the attitude trajectory computed with IDVD in virtual domain is closer to the solution of GPOPS.

The attitude parameter affects the computational time and solution quality. The IDVD applied to quaternions requires more time to converge than the analogous IDVD applied to MRP and Euler angles due to the usage of the exponential and logarithmic. Moreover, the solutions obtained with quaternions are closer to the reference optimal solution provided by GPOPS than MRP and Euler angles represented by fifth-order polynomials. The overall best optimal solution is obtained with the seventh-order IDVD applied to the MRP in the virtual domain. The

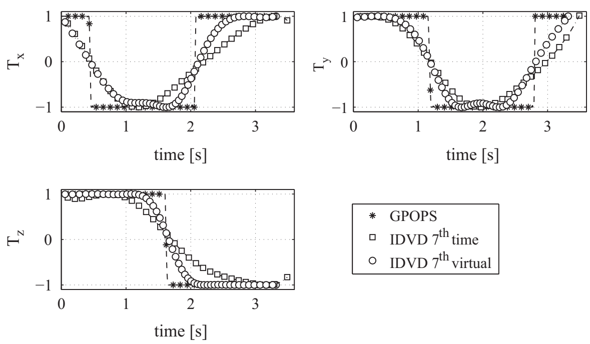

Fig. 1. Optimal control time history generated with GPOPS and seventh-order IDVD applied to the Euler angles in virtual domain and time domain.

Table 1
Optimization of a minimum time rest to rest 180 deg reorientation of a symmetric spacecraft using IDVD and GPOPS. Controls and trajectories which are obtained with the marked methods are represented in Figs. 1 and 2 respectively.

Attitude	Method	J (s)	ΔJ (%)	$CPUt$ (s)	$\Delta\omega$ (deg/s)	$\Delta\theta$ (deg)
Modified Rodrigues Parameters	GPOPS	3.2431	–	68.7	1.8e – 1	6.9e – 3
	IDVD 5th – time	3.8153	17.6	0.3	2.7e – 1	3.7e – 3
	IDVD 5th – virtual	3.4351	5.9	2.2	2.5e – 1	1.7e – 3
	IDVD 7th – time	3.4853	7.5	0.7	1.6e – 1	2.0e – 3
	IDVD 7th – virtual	3.3179	2.3	7.7	1.9e – 1	2.5e – 2
Euler angles	GPOPS*	3.2431	–	93.2	1.8e – 1	6.9e – 3
	IDVD 5th – time	3.6871	13.7	0.3	1.6e – 1	3.4e – 3
	IDVD 5th – virtual	3.4443	6.2	1.6	5.6e – 1	1.3e – 1
	IDVD 7th – time*	3.5277	8.8	1.1	1.2e – 1	2.8e – 3
	IDVD 7th – virtual*	3.3302	2.7	7.1	2.8e – 1	6.1e – 2
Quaternion	GPOPS	3.2431	–	100.3	1.8e – 1	6.9e – 3
	IDVD 5th – time	3.5418	9.2	2.0	9.7e – 2	2.3e – 3
	IDVD 5th – virtual	3.3971	4.7	7.8	2.4e – 2	1.8e – 3

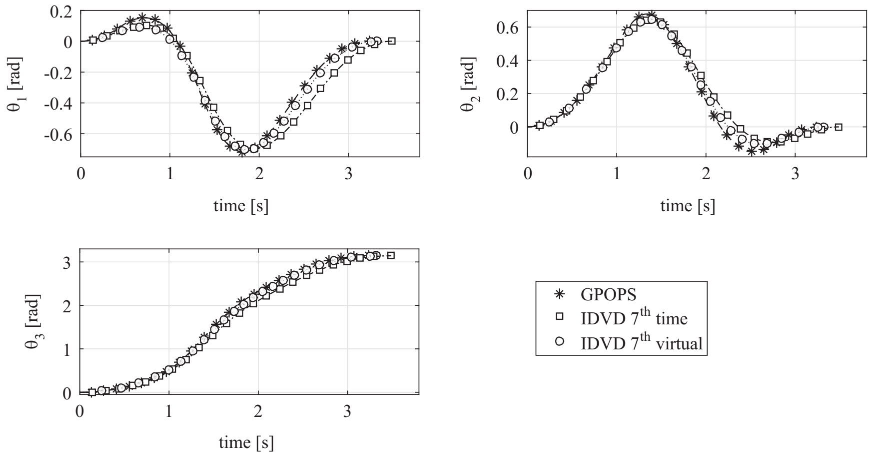

Fig. 2. Optimal attitude trajectory in terms of Euler angles, generated with GPOPS, seventh-order IDVD in virtual domain and seventh-order IDVD in time domain, applied to the Euler angles.

polynomial order influences the solution quality and computational time required by the solver. Higher polynomial orders provide better optimal solution, but at the cost of more computational time due the higher number of variable parameters.

Fig. 3 summarizes the performance of IDVD based on the performance index *MI*, for different combinations of weighting coefficients. Notably, in Fig. 3(a) the performance bars are ordered according to the weights $K_J=0$ and $K_{CPU}=1$. In this case, the performance criteria is the computational speed required by IDVD to converge to a sub-optimal solution. The figure shows that higher computational speed is achieved when IDVD is applied to fifth-order polynomials (lowest order considered) in the time domain. The reason of this behavior is that by increasing the order of polynomials representing the trajectory, more variables are introduced in the optimization problem and the SQP solver requires more time to converge. Furthermore, the definition of the parametric curves in the virtual domain introduces additional variables, being the polynomial coefficients for the speed factor (see Eq. (13)), which further increment the computational time. The lowest values of *MI* are obtained with seventh-order IDVD applied to the virtual domain.

Fig. 3(b) shows the performance of IDVD ordered according to the weights $K_J=0.5$ and $K_{CPU}=0.5$. In this case, the performance criteria is the computational speed and the optimality of the solution equally balanced. The best performance is obtained with fifth-order IDVD applied to MRP and Euler angles defined in the virtual domain. The low polynomial order ensures high computational speed, while the usage of the virtual domain improves the optimality of the solution. Lowest performance is obtained with fifth-order IDVD applied to Euler angles and MRP in the time domain, due to the low optimality of the solution.

Fig. 3(c) shows the performance of IDVD ordered according to the weights $K_J=1$ and $K_{CPU}=0$. In this case, the performance criteria is the optimality of the solution. The best performance is obtained with high-order polynomials defined in the virtual domain, due to the high number of variables to be optimized.

6.2. Minimum time maneuver: Monte Carlo analysis for inertia and maneuver

Additional test scenarios have been generated by varying inertia, control bounds and reorientation angles using the Monte Carlo method, as summarized in Table 2. The symbol $x_{A,B}$ denotes a random value of the parameter x in the range $[A,B]$. The generated scenarios have been analyzed using IDVD as in the previous section.

The first guesses on the controls at endpoints are taken as the endpoints controls computed by GPOPS. The first guesses for the maneuver time of Scenario 1, Scenario 2 and Scenario 3 are 6 s, 1 s and 1 s respectively.

Tables 3 and 4 summarize the performance of IDVD. The symbol σ_x denotes the standard deviation of the data x, while *Conv* indicates the converge percentage. Similarly to the analysis of the minimum time 180 deg reorientation of a symmetric spacecraft of the previous section, the virtual domain improves the solution quality, but at the cost of more computational time, for any polynomial order and attitude parameter considered. Moreover, all the solutions obtained with IDVD are sub-optimal, since the optimal bang–bang nature of the controls cannot be achieved due to the nature of the method.

The virtual domain also affects the convergence percentage. In the case of IDVD applied to time domain, all the algorithms converge to a sub-optimal solution (100% of convergence percentage). In the case of IDVD applied to virtual domain, the convergence percentage is always below 100%. The reason is that the polynomial coefficients of $\lambda(\tau)$ are not related to any state variable or physical behavior of the spacecraft and therefore the guesses of these parameters may not be accurate enough to ensure convergence of the optimization solver. Lowest convergence rates are observed with seventh-order IDVD applied to virtual domain, due to the higher number of variable parameters, which require an accurate first guess to ensure convergence of the solver.

The attitude representation affects the solution quality and computational time. As in the ideal 180 deg reorientation scenario, the average optimal cost closest to the one computed from GPOPS is obtained with a seventh-order IDVD applied to MRP in the virtual domain.

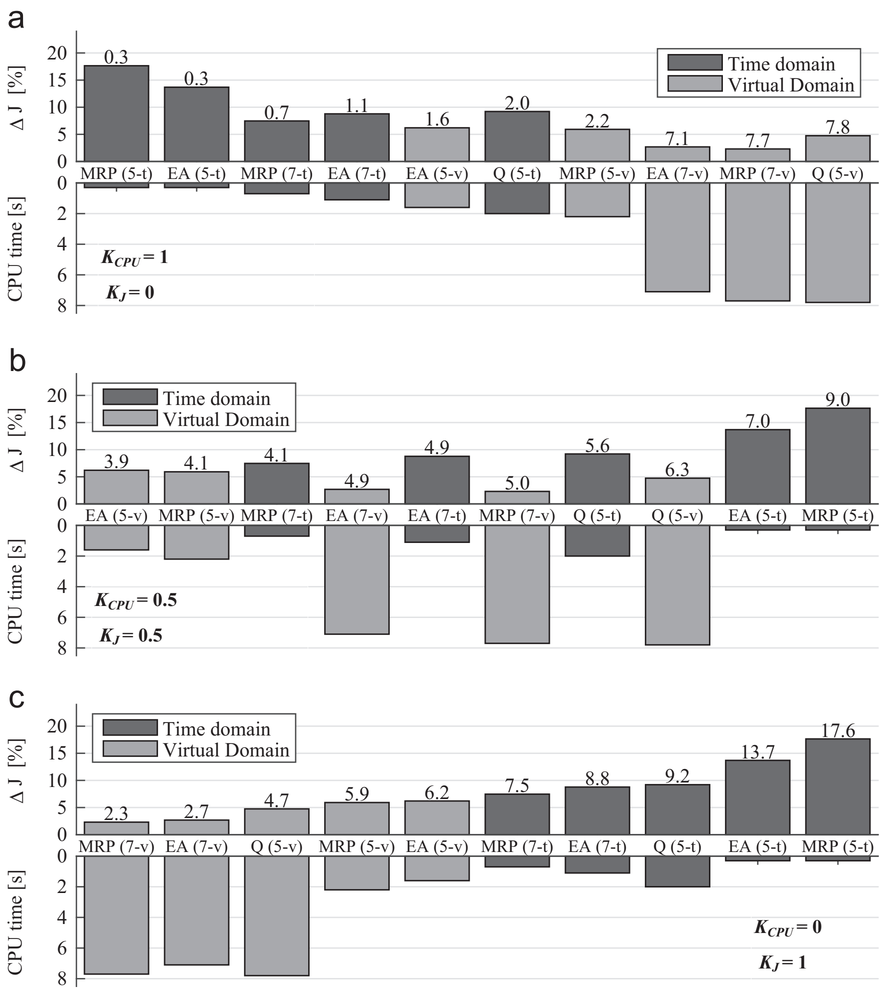

Fig. 3. Performance of IDVD for the minimum time 180 deg reorientation. The symbol Q denotes the quaternion, MRP denotes the modified Rodrigues parameters and EA indicates the Euler angles. The order of IDVD and domain are indicated in parenthesis. The graphs are ordered according to the performance index (see Eq. (47)), for several weighting coefficients. The performance index is indicated above each bar.

Table 2
Reference scenarios generated with the Monte Carlo method.

Scenario	Inertia	$\boldsymbol{T}_{max}$	$\boldsymbol{\theta}_F$ (deg)	Number of cases
1	diag($[I_{1,10}, I_{1,10}, I_{1,10}]$)	[1,1,1]	[0,0,180]	500
2	diag([1,1,1])	$[T_{1,10}, T_{1,10}, T_{1,10}]$	[0,0,180]	500
3	diag([1,1,1])	[1,1,1]	$[\boldsymbol{\theta}_{-180,180}, \boldsymbol{\theta}_{-180,180}, \boldsymbol{\theta}_{-180,180}]$	1000

All the tested methods present large standard deviation for ΔJ. In fact, *fmincon* is a local optimization solver and therefore an inaccurate first guess on the variable of the problem may cause convergence to a solution not even close to the one obtained with GPOPS. The first guess effects also the computational time to obtain a solution, since a bad first guess increases the number of iterations required by the optimization solver to converge.

Fig. 4 shows the performance of IDVD based on the performance index *MI*, for different combinations of weighting coefficients. Similarly to the minimum time 180 deg reorientation, IDVD applied to time domain presents the best performance when the criterion is the computational speed (Fig. 4(a), $K_J=0$ and $K_{CPU}=1$). As observed, this behavior is due to the lower number of variables to be optimized than IDVD applied to virtual domain, which results in fewer iterations of the solver to converge.

Fig. 4(b) shows the performance of IDVD ordered when the performance criteria is the computational speed and the optimality of the solution is equally balanced ($K_J=0.5$ and $K_{CPU}=0.5$). The best performance is obtained with seventh-order IDVD applied to MRP in time domain, due to the

Table 3
Performance of IDVD in case of pseudorandom reorientation test scenarios.

Attitude	Method	Conv. (%)	ΔJ (%)	$\sigma_{\Delta J}$ (%)	$CPUt$ (s)	σ_{CPUt} (s)
Modified Rodrigues Parameters	IDVD 5th – time	100	19.0	21.8	0.4	0.2
	IDVD 5th – virtual	99.9	8.7	15.1	3.1	1.5
	IDVD 7th – time	100	10.0	16.2	1.7	1.1
	IDVD 7th–virtual	97.4	6.4	13.8	15.0	5.5
Euler angles	IDVD 5th – time	100	27.2	31.2	0.4	0.7
	IDVD 5th – virtual	98.1	18.7	26.7	4.0	15.6
	IDVD 7th – time	100	20.7	27.8	1.9	1.8
	IDVD 7th – virtual	95.7	14.9	22.6	15.6	21.2
Quaternions	IDVD 5th – time	100	14.7	17.2	2.8	1.5
	IDVD 5th – virtual	99.7	12.0	20.1	13.3	8.2

Table 4
Final errors on angular velocity and attitude after propagation of the controls obtained from IDVD in case of pseudorandom reorientation test scenarios.

Attitude	Method	$\Delta\omega$ (deg/s)	$\sigma_{\Delta\omega}$ (deg/s)	$\Delta\boldsymbol{\theta}$ (deg)	$\sigma_{\Delta}\boldsymbol{\theta}$(deg)
Modified Rodrigues Parameters	IDVD 5th – time	1.4e−2	1.7e−2	2.5e−1	3.6e−1
	IDVD 5th – virtual	9.9e−2	2.7e−2	1.2e−1	2.2e−1
	IDVD 7th – time	9.4e−3	1.1e−2	2.5e−1	3.6e−1
	IDVD 7th – virtual	1.5e−1	2.9e−1	2.3e−1	4.9e−1
Euler angles	IDVD 5th – time	1.0e−2	1.3e−2	2.2e−1	3.5e−1
	IDVD 5th – virtual	3.7e−1	7.2e−1	1.9e−1	3.7e−1
	IDVD 7th – time	1.0e−2	1.1e−2	2.4e−1	3.5e−1
	IDVD 7th – virtual	8.2e−1	1.6	3.7e−1	7.4e−1
Quaternion	IDVD 5th – time	1.1e−2	1.1e−2	2.2e−1	3.4e−1
	IDVD 5th – virtual	3.2e−1	3.2e−1	9.0e−2	2.7e−1

solution quality and the high computational speed. Furthermore, IDVD applied to virtual domain does not present favorable performance as in Fig. 3(b) due to the high computational time required to converge in some scenarios.

Fig. 4(c) shows the performance of IDVD ordered according to the weights $K_J=1$ and $K_{CPU}=0$. In this case, the performance criteria is the optimality of the solution. The best performance is obtained with seventh-order and fifth-order IDVD applied MRP in virtual domain. However, IDVD applied to virtual domain does not always offer better performance than IDVD in time domain due to the sensitivity to the first guess (compare Fig. 4(c) with Fig. 3(b)). As already observed, the first guess on the polynomial coefficients of $\lambda(\tau)$ is not straightforward and an inaccurate guess may cause convergence to a non-optimal solution.

6.3. Minimum energy maneuver: 180 deg reorientation of a symmetric spacecraft

The ideal 180 deg reorientation about the z-axis of a symmetric spacecraft is analyzed for the minimum energy problem. The endpoint conditions are defined in Eq. (48).

The first guesses for the dimensionless initial and final controls are $[0\ \ 0\ \ 0.25]$ and $[0\ \ 0\ \ -0.25]$ respectively. Accordingly, the guesses on the polynomial coefficients related to angular accelerations are calculated with Eqs. (16), (17), (24), (25), -(36), (37), (41), (42). The guess on the maneuver duration is 10 s. The trapezoidal rule is employed to integrate the cost function in Eq. (6) using 100 equally spaced nodes in the virtual and time domains. To ensure convergence of the optimization algorithm, the speed factor is represented by a third-order polynomial.

As summarized in Table 5, the IDVD method closely matches the optimal solution computed by GPOPS in both time and virtual domains and for any considered attitude representation, with a maximum difference percentile of $\Delta J=0.22\%$ obtained with MRP parameterized in time domain. In fact, the optimal controls which minimize the maneuver energy do not saturate and their continuous shapes can be represented by polynomial functions. This behavior is showed in Fig. 5, where the controls obtained with GPOPS and IDVD applied to the Euler angles in time and virtual domains are represented. As a consequence, the usage of the virtual domain is not necessary since it increases the computational time without improving the solution quality. However, the virtual domain slightly improves the solution quality in the case of IDVD applied to MRP.

Fig. 5 also provides significant information about the maneuver: only the torque about the z-axis of the spacecraft is actuated and the consequent energy-optimal maneuver is an eigenaxis rotation about the same axis.

Fig. 6 summarizes the performance of IDVD based on the performance index MI, for different combinations of weighting coefficients. Independently of the choice of the weights, the IDVD method applied to the time domain presents the best performance, since the optimal solution can be achieved without the virtual domain. The presence of the virtual domain increases the computational time only and therefore the performances are lower.

7. Conclusions

The application of the inverse dynamics approach to rapid optimal attitude trajectory generation for spacecraft

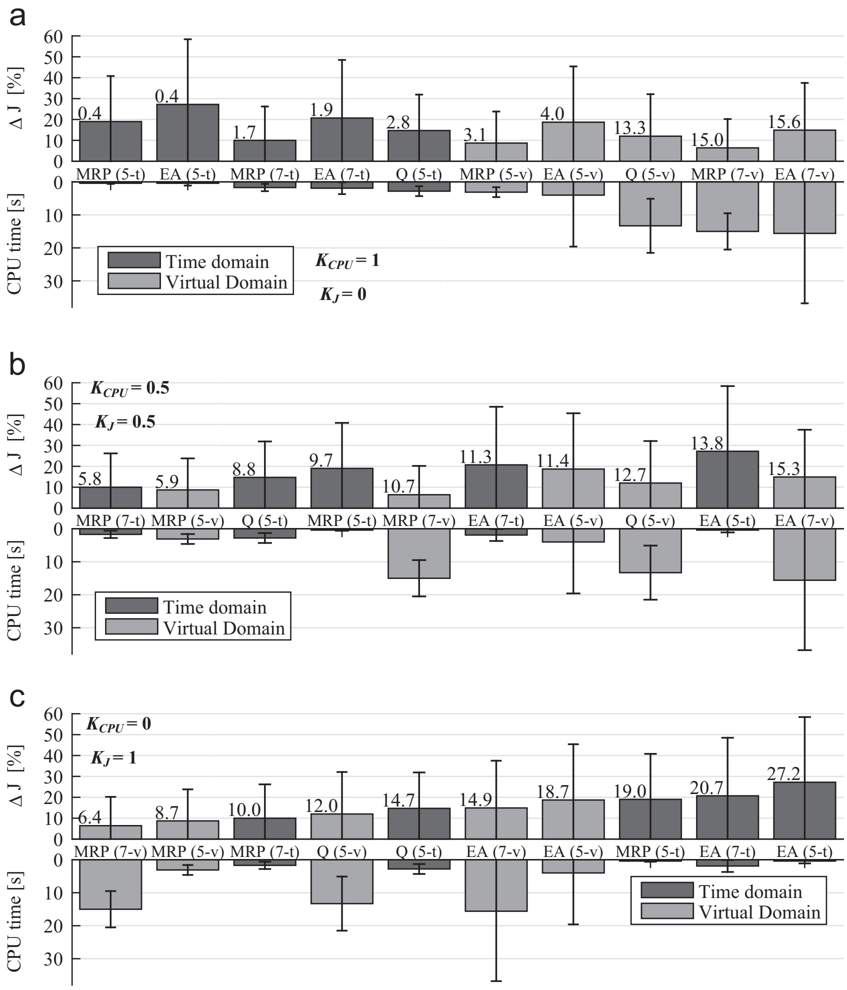

Fig. 4. Performance of IDVD for several pseudorandom spacecraft geometry and maneuvers. The symbol Q denotes the quaternion, MRP denotes the modified Rodrigues parameters and EA indicates the Euler angles. Order of IDVD and domain are indicated in parenthesis. The graphs are ordered according to the performance index (see Eq. (47)), for several weighting coefficients. The performance index is indicated above each bar. The vertical lines indicate the standard deviations.

Table 5
Optimization of a minimum energy rest to rest 180 deg reorientation of a symmetric spacecraft using IDVD and GPOPS. Controls obtained with the marked methods are represented in Fig. 5.

Attitude	Method	J	$CPUt$ (s)	$\Delta\omega$ (deg/s)	$\Delta\boldsymbol{\theta}$ (deg)	ΔJ (%)
Quaternion	GPOPS*	5.922e−2	198.8	2.8e−4	2.9e−4	0
Modified Rodrigues parameters	IDVD 5th – time	5.935e−2	1.0	5.3e−6	8.0e−5	0.22
	IDVD 5th – virtual	5.923e−2	5.8	3.3e−4	2.9e−4	0.02
Euler angles	IDVD 5th – time*	5.923e−2	0.7	4.9e−11	1.8e−4	0.02
	IDVD 5th – virtual*	5.923e−2	1.7	4.4e−6	3.8e−4	0.02
Quaternion	IDVD 5th – time	5.923e−2	1.1	3.3e−12	1.8e−4	0.02
	IDVD 5th – virtual	5.923e−2	16.8	1.9e−5	3.3e−4	0.02

reorientation maneuvers is investigated. The method is applied to the modified Rodrigues parameters, Euler angles and quaternions with different polynomial orders defined in the time domain and in the virtual domain.

In the case of minimum time maneuvers, the optimal bang-bang shape of the controls cannot be achieved with IDVD due to the nature of the method and only sub-optimal solutions are computed. The definition of the attitude trajectory in the virtual domain improves the solution quality with respect to the analogous formulation in the time domain since the speed factor increases the flexibility of the velocity along the trajectory. However, the computational time is increased due to the higher number of variables and constraints. The attitude representation also influences the computational time and solution quality: quaternions require high computational time due to the logarithmic and

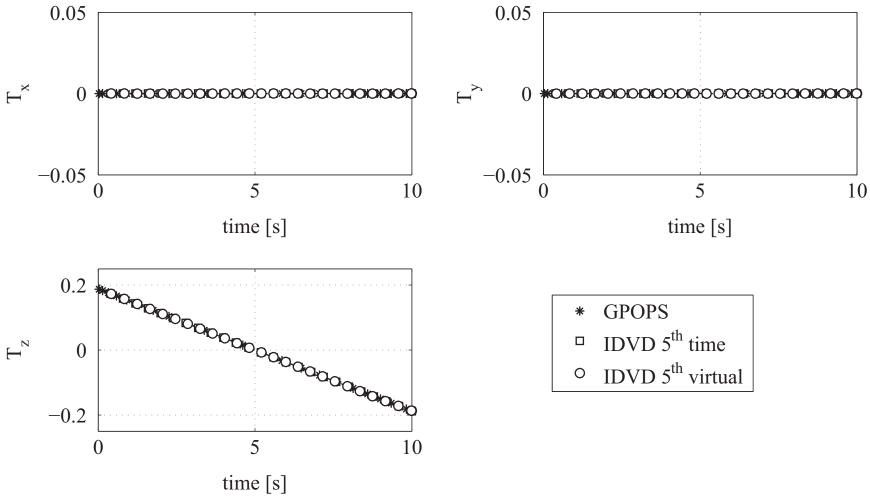

Fig. 5. Optimal control time history generated with GPOPS and fifth-order IDVD applied to the Euler angles in virtual domain and time domain.

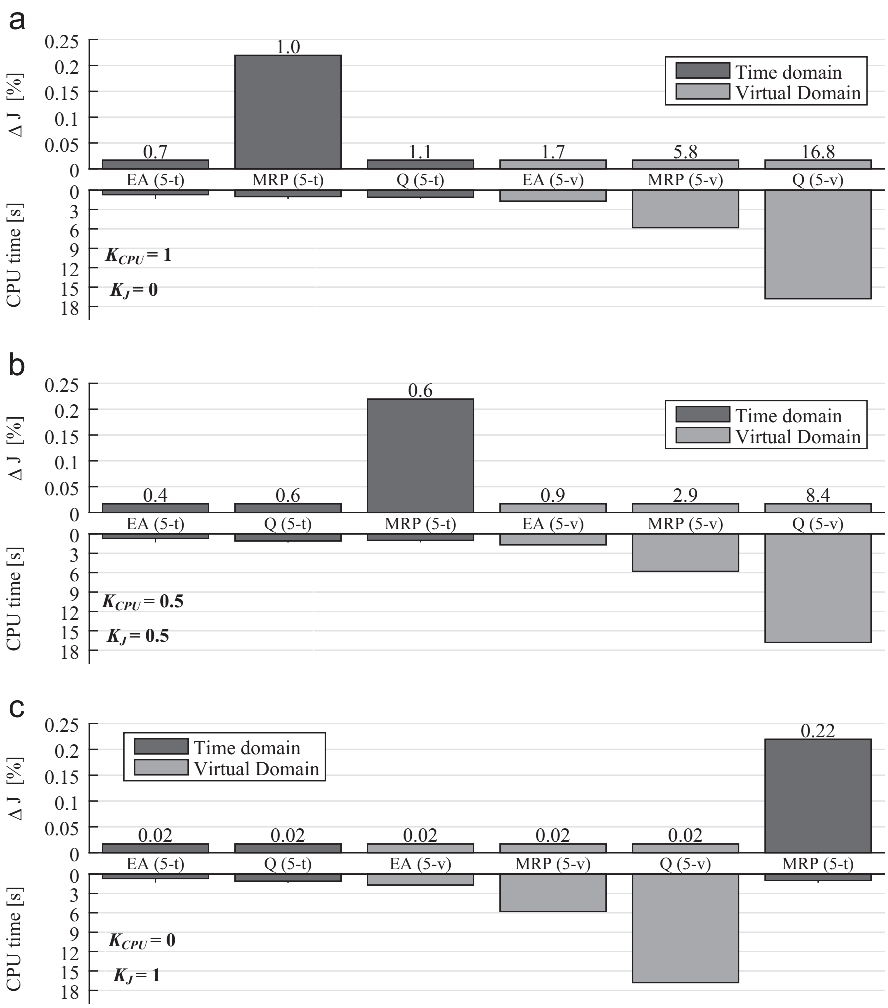

Fig. 6. Performance of IDVD for the minimum energy 180 deg reorientation. The symbol Q denotes the quaternion, MRP denotes the modified Rodrigues parameters and EA indicates the Euler angles. Order of IDVD and domain are indicated in parenthesis. The order of IDVD and domain are indicated in parenthesis. The graphs are ordered according to the performance index (see Eq. (46)), for several weighting coefficients. The performance index is indicated above each bar.

exponential maps; modified Rodrigues parameters and Euler angles are faster since no constraints on the attitude history must be respected.

The choice of the trajectory parameterization for minimum time maneuvers depends on the requested performance. Looking for a quick computation of a sub-optimal solution, IDVD applied to time domain represents the most reasonable choice, since the limited number of variable parameters ensure rapid convergence to a solution, but at the cost of low optimality. Looking for the optimality of the solution, high-order polynomials defined in the virtual domain provide solutions closer to the ones obtained with a Gauss Pseudospectral method, but require accurate first guess to ensure 100% convergence. If the performance criteria take into account both convergence speed and optimality of the solution, then high order IDVD used in the time domain or low-order IDVD used in the virtual domain should be considered and applied to the modified Rodrigues parameters.

In the case of the considered minimum energy maneuver, IDVD applied to the time domain computes the optimal solution found by the pseudospectral method GPOPS, since the optimal controls do not saturate and can be represented with continuous functions of the time.

The inverse dynamics approach is applicable as sub-optimal guidance for spacecraft reorientation maneuvers. The modified Rodrigues parameters may represent the most suitable parameter due to the absence of singularities in typical reorientation ranges and the favorable compromise between computational speed and solution quality. Notably, high order polynomials in time domain or low-order polynomials in virtual domain should be considered for attitude representation in case of minimum time maneuvers; low order polynomials in time domain should be considered when the optimal controls do not saturate, for example in minimum energy maneuvers.

Acknowledgments

The financial support from Munich Aerospace e.V. is gratefully acknowledged. The first author thanks Dr. Marco Ciarcia, NRC research associate at Spacecraft Robotics Laboratory of Naval Postgraduate School, Monterey for his precious advice on IDVD.

References

[1] I. Bayir , IKONOS collection capability with the direct tasking from a regional operation center, in: Proceeding of the 2003 International Conference on Recent Advances in Space Technologies, Istanbul, Turkey, 2003, pp. 30–33.

[2] B. Wie, D. Bailey, C. Heiberg, Rapid multitarget acquisition and pointing control of agile spacecraft, J. Guidance Control Dyn. 25 (1) (2002) 96–104, http://dx.doi.org/10.2514/2.4854.

[3] T. Davis, S. Straight, Development of the tactical satellite 3 for responsive space missions, in: Proceedings of the 4th Responsive Space Conference, AIAA 2006-4003, April, 2006.

[4] M. Karpenko, S. Bhatt, N. Bedrossian, A. Fleming, I.M. Ross, First flight results on time-optimal spacecraft slews, J. Guidance Control Dyn. 35 (2) (2012) 367–376, http://dx.doi.org/10.2514/1.54937.

[5] X. Bai, J.L. Junkins, New results for time-optimal three axis reorientation of a rigid spacecraft, J. Guidance Control Dyn. 32 (4) (2009) 1071–1076.

[6] K.D. Bilimoria, B. Wie, Time-optimal three axis reorientation of a rigid spacecraft, J. Guidance Control Dyn. 16 (3) (1993) 446–452.

[7] M.V. Dixon, T.N. Edelbaum, J.E. Potter, W.E. Vandervelde, Fuel optimal reorientation of axisymmetric spacecraft, J. Spacecr. Rockets 7 (11) (1970) 1345–1351.

[8] B.M. Yutko, R.G. Melton, Optimizing spacecraft reorientation maneuvers using a pseudospectral method, J. Aerosp. Eng. Sci. Appl. 2 (1) (2010).

[9] R.G. Melton, Hybrid methods for determining time-optimal, constrained spacecraft reorientation maneuvers, Acta Astronaut. 94 (2014) 294–301.

[10] C. Louembet, F. Cazaurang, A. Zolghadri, C. Charbonnel and C. Pittet, Design of algorithms for satellite slew manoeuver by flatness and collocation, in: Proceedings of the 2007 American Control Conference, New York, USA, July 11–13, 2007.

[11] G.A. Boyarko, M. Romano, O.A. Yakimenko, Time-optimal reorientation of a spacecraft using an inverse dynamics optimization method, J. Guidance Control Dyn. 34 (4) (2011) 1197–1208.

[12] O.A. Yakimenko, Optimization of Holonomic Attitude Dynamics using IDVD Method, in: Proceedings of the 3rd International Symposium on Systems and Controls in Aeronautics and Astronautics (ISSCAA), Harbin, China, June 8–10, 2010.

[13] B. Wie (Education Series), Space Vehicle Dynamics and Control, AIAA, Reston, VA, 1998, 403–407.

[14] M.D. Shuster, A survey of attitude representations, J. Astronaut. Sci. 41 (4) (1993) 439–517.

[15] O.A. Yakimenko, Direct method for rapid prototyping of near optimal aircraft trajectories, J. Guidance Control Dyn. 23 (5) (2000) 865–875.

[16] M. Kim, M. Kim, S. Shin, A general construction scheme for unit quaternion curves with simple high order derivatives, in: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, ACM, New York, 1995, pp. 369–376.

[17] W. Boehm, G. Farin, J. Kahmann, A survey of curve and surface methods in CAGD (July), Comput. Aided Geom. Des. 1 (1) (1984) 1–60, http://dx.doi.org/10.1016/0167-8396(84)90003-7.

[18] G.E. Farin, Curves and Surfaces for Computer-Aided Geometric Design: A Practical Code, 4th ed. Academic Press, Inc., Orlando, FL, USA, 1996.

[19] M. Ciarcia, M. Romano, Suboptimal guidance for orbital proximity maneuver with path constraints capability, in: Proceedings of the AIAA Guidance, Navigation and Control Conference, Minneapolis, Minnesota, 13–16 August, 2012, http://dx.doi.org/10.2514/6.2012-4920.

[20] MatLab Version 2013b User Manual, The Mathworks, Natick, MA, 2013.

[21] A.V. Rao, D.A. Benson, C.L. Darby, M.A. Patterson, C. Francolin, I. Sanders, G.T. Huntington, GPOPS: A MATLAB software for solving multiple-phase optimal control problems using the gauss pseudospectral method, ACM Trans. Math. Softw. 37 (2) (2010). Paper 22, pp. 1–39.