IAA-AAS-DyCoSS2-04-09

PERFORMANCE EVALUATION OF THE INVERSE DYNAMICS METHOD FOR OPTIMAL SPACECRAFT REORIENTATION

Jacopo Ventura,[*] Marcello Romano,[†] and Ulrich Walter[‡]

In this paper we apply the inverse dynamics in the virtual domain method to Euler angles, quaternion, and modified Rodrigues parameters for rapid optimal attitude trajectory generation for spacecraft reorientation. 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. The virtual domain improves the solution, but at the cost of more computational time. The attitude representation also affects solution quality and computational speed. For minimum-fuel problems, the optimal solution can be obtained without the virtual domain.

INTRODUCTION

The reorientation a spacecraft is a common task in most space missions, and the more rapid the maneuvers the more effective and productive is the mission. For example, rapid retargeting allows more and longer observations of the target object per orbit. Therefore, optimal reorientation maneuvers with minimal maneuver time or fuel consumption are highly significant for Earth observation missions.

Optimal reorientation of a spacecraft has been studied theoretically for both minimum-time and minimum-energy problem.[1-3] 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 precessional component.[2]

Recently, the problem has been investigated numerically using two different direct optimization approaches: pseudospectral methods and inverse dynamics. Pseudospectral methods are based on the numerical integration of the differential equations of motion and provide accurate solutions, but may converge slowly.[4,5] 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. Louembert et al. first applied this method to the optimal spacecraft reorientation problem using B-Splines to represent the modified Rodrigues parameters.[6] Recently, Boyarko proposed a rapid attitude trajectory generation method based on the inverse dynamics in the virtual domain (IDVD).[7] Here the quaternion components are approximated by polynomials defined in an abstract argument (virtual domain).

[*] Ph.D. candidate, Institute of Astronautics, Technische Universität München, Boltzmannstr. 15, 85748, Garching, Germany.
[†] Associate Professor, Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, Monterey, California, 93943-5107.
[‡‡] Prof. Dr. h. c., Institute of Astronautics, Technische Universität München, Boltzmannstr. 15, 85748, Garching, Germany.

Finally, Yakimenko applied IDVD to Euler angles.[8] 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 presence of virtual domain on solution quality and computational speed. To evaluate the impact of the virtual domain, the attitude coordinates are 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.

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.

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 $\omega = [\omega_x, \omega_y, \omega_z]^T$ and the inertial tensor $I = \text{diag}([I_{xx}, I_{yy}, I_{zz}])$ referenced to the body-fixed principal axis can be expressed as[9]

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

In Eq. (1) $T = [T_x, T_y, T_z]^T$ 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.[10] In terms of quaternion $q = [q_1,q_2,q_3,q_4]^T$, the kinematic differential equation is given by

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

where the symbol $\otimes$ denotes the Hamilton product and $\Omega = [\omega_x, \omega_y, \omega_z, 0]^T$. In terms of Euler angles $\theta = [\theta_1, \theta_2, \theta_3]^T$ of the rotational sequence 1-2-3, the kinematic differential equation is

$$\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{bmatrix}=\frac{1}{\cos\theta_2}\begin{bmatrix}\cos\theta_3 & -\sin\theta_3 & 0\\ \sin\theta_3\cos\theta_2 & \cos\theta_3\cos\theta_2 & 0\\ -\cos\theta_3\sin\theta_2 & \sin\theta_3\sin\theta_2 & 1\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}. \tag{3}$$

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

$$\begin{bmatrix}\dot{\sigma}_1\\ \dot{\sigma}_2\\ \dot{\sigma}_3\end{bmatrix}=\frac{1}{4}\begin{bmatrix}1+\sigma_1^2-\sigma_2^2-\sigma_3^2 & 2(\sigma_1\sigma_2-\sigma_3) & 2(\sigma_1\sigma_3+\sigma_2)\\ 2(\sigma_1\sigma_2+\sigma_3) & 1-\sigma_1^2+\sigma_2^2-\sigma_3^2 & 2(\sigma_2\sigma_3-\sigma_1)\\ 2(\sigma_1\sigma_3-\sigma_2) & 2(\sigma_2\sigma_3+\sigma_1) & 1-\sigma_1^2-\sigma_2^2+\sigma_3^2\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}=\frac{1}{4}B(\sigma)\omega. \tag{4}$$

The optimal spacecraft reorientation problem consists of minimizing the cost function J by finding the optimal control T, subjected to the control constraints $T_{\min} \le T \le T_{\max}$, that brings the system described by Eq. (1) and Eqs. (2)-(4) from an initial state of angular velocity ω_0 and attitude q_0, θ_0 or σ_0 to a final state of angular velocity ω_F and attitude q_F, θ_F or σ_F. The cost function J is defined as

$$J=\int_{t_0}^{t_F} dt \tag{5}$$

for minimum-time maneuvers and

$$J=\frac{1}{2}\int_{t_0}^{t_F}\left(T_x^2+T_y^2+T_z^2\right)dt \tag{6}$$

for minimum-quadratic control (or energy) expenditure.

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 a nonlinear programming problem by describing the trajectory components with a set of polynomial functions.

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.[11] This permits to decouple space and time optimization. Thus 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 components z and a finite number of its derivatives. For the considered spacecraft reorientation problem it results in

$$\omega(\tau)=f_1(z,\dot{z}), \quad T(\tau)=f_2(z,\dot{z}). \tag{7}$$

Virtual domain and time domain are related to each other by

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

where

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

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 and the IDVD is reduced to the basic inverse dynamics method.

INVERSE DYNAMICS WITH EULER ANGLES

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

Inverse Dynamics in the Virtual Domain

Following the main idea of IDVD, each Euler angle is represented in the virtual domain $\tau \in [0; 1]$ by an n-th order Bezier curve as follows

$$\theta_j(\tau) = \sum_{i=0}^{n} a_{j,i}\beta_{i,n}(\tau), \qquad j = 1,2,3 \quad (10)$$

where $a_{j,i}$ are the polynomial coefficients acting as control points and

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

is the Bernstein base. The map between the time domain and the virtual domain is defined by the speed factor $\lambda(\tau)$, which is also expressed through a polynomial of order p

$$\lambda(\tau) = \sum_{i=0}^{p} b_i\tau^i \quad . \quad (12)$$

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

$$\dot{\theta}_j(\tau) = \frac{d\theta_j(\tau)}{d\tau}\frac{d\tau}{dt} = \frac{d\theta_j(\tau)}{d\tau}\lambda, \qquad j = 1,2,3 \quad (13)$$

$$\ddot{\theta}_j(\tau) = \frac{d\dot{\theta}_j(\tau)}{d\tau}\frac{d\tau}{dt} = \frac{d^2\theta_j(\tau)}{d\tau^2}\lambda^2 + \frac{d\theta_j(\tau)}{d\tau}\frac{d\lambda}{dt}\lambda. \qquad j = 1,2,3 \quad (14)$$

Note that the attitude trajectory components can be also represented with high-order polynomials, leaving the virtual argument unbounded and its final value as variable of the problem. However, preliminary experiments showed that this approach is computationally more expensive that the proposed one and requires a good first guess on the final value of the virtual argument, which is not intuitive.

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

$$\begin{cases} a_{j,0} = \theta_{0j} \\ a_{j,1} = \dfrac{\dot{\theta}_{0j}}{n\lambda(0)} + a_{j,0} \\ a_{j,2} = \dfrac{1}{2}\dbinom{n}{2}^{-1} \dfrac{\ddot{\theta}_{0j} - \dot{\theta}_{0j}\lambda^{'}(0)}{\lambda^2(0)} + 2a_{j,1} - a_{j,0} \end{cases} \qquad j = 1,2,3 \tag{15}$$

for the initial conditions and

$$\begin{cases} a_{j,n} = \theta_{Fj} \\ a_{j,n-1} = -\dfrac{\dot{\theta}_{Fj}}{n\lambda(1)} + a_{j,n} \\ a_{j,n-2} = \dfrac{1}{2}\dbinom{n}{n-2}^{-1} \dfrac{\ddot{\theta}_{Fj} - \dot{\theta}_{Fj}\lambda^{'}(1)}{\lambda^2(1)} + 2a_{j,n-1} - a_{j,n} \end{cases} \qquad j = 1,2,3 \tag{16}$$

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 j-th 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[12]

$$n \geq N_B + 1\,. \tag{17}$$

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.

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 ω

$$\begin{cases} \omega_x = \cos\theta_2 \cos\theta_3 \dot{\theta}_1 + \sin\theta_3 \dot{\theta}_2 \\ \omega_y = -\cos\theta_2 \sin\theta_3 \dot{\theta}_1 + \cos\theta_3 \dot{\theta}_2 \\ \omega_z = \sin\theta_2 \dot{\theta}_1 + \dot{\theta}_3 \end{cases}. \tag{18}$$

In the previous expressions, the time derivatives of the angles are calculated using Eq. (13). The angular accelerations are obtained from the time derivative of Eq. (18). Finally, the inversion of Eq. (1) provides the control torques

$$\begin{cases} T_x = I_{xx}\dot{\omega}_x + (I_{zz} - I_{yy})\omega_z \omega_y \\ T_y = I_{yy}\dot{\omega}_y + (I_{xx} - I_{zz})\omega_z \omega_y \\ T_z = I_{zz}\dot{\omega}_z + (I_{yy} - I_{xx})\omega_x \omega_y \end{cases}. \tag{19}$$

Note that the obtained expressions of angular velocities and torques are functions of the virtual argument τ and the polynomial coefficients. The cost function J can be now transformed into a (nonlinear) function $J(a_{j,i},b_i)$ of the polynomials coefficients using a numerical integration.

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

$$\begin{cases} T_{\min} \le T(\tau_i) \le T_{\max} \\ t(\tau_i) \ge 0 \end{cases}, \tag{20}$$

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. For satisfying results the number of nodes should range between 25 and 100.[12] 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. (20) imposes Eq. (8) to map the points of the virtual domain into positive points of the time domain.

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. (10) and 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 a modified Eq. (10). Therefore, in this paper the Euler angles are expressed in the time domain $t \in [0; t_F]$ by n-th order polynomials

$$\theta_j(t) = \sum_{i=0}^{n} a_{j,i} t^i . \qquad j = 1,2,3 \tag{21}$$

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

$$\dot{\theta}_j(t) = \sum_{i=0}^{n} i a_{j,i} t^{i-1} , \quad \ddot{\theta}_j(t) = \sum_{i=0}^{n} i(i-1) a_{j,i} t^{i-2} . \qquad j = 1,2,3 \tag{22}$$

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

$$\begin{cases} a_{j,0} = \theta_{0j} \\ a_{j,1} = \dot{\theta}_{0j} \\ a_{j,0} = \dfrac{\ddot{\theta}_{0j}}{2} \end{cases}, \qquad j = 1,2,3 \tag{23}$$

$$
\begin{cases}
a_{j,n} = \dfrac{\theta_{Fj} - \sum_{i=0}^{n-1} a_{j,i} t_F^i}{t_F^n} \\
a_{j,n-1} = \dfrac{\dot{\theta}_{Fj} - \sum_{i=0}^{n-2} i a_{j,i} t_F^{i-1} - n a_{j,n} t_F^{i-1}}{(n-1) t_F^{n-2}} \\
a_{j,n-2} = \dfrac{\ddot{\theta}_{Fj} - \sum_{i=0}^{n-3} i(i-1) a_{j,i} t_F^{i-2} - \sum_{i=n-1}^{n} i(i-1) a_{j,i} t_F^{i-2}}{(n-2)(n-3) t_F^{n-4}}
\end{cases}
\quad . \qquad j = 1,2,3 \tag{24}
$$

The previous relationships are obtained by imposing the boundary conditions θ_0, $\dot{\theta}_0$, $\ddot{\theta}_0$ and θ_F, $\dot{\theta}_F$, $\ddot{\theta}_F$ to Eq. (21) and Eq. (22). Note that the order of the polynomials must satisfy Eq. (17). The expressions of angular velocity and controls are obtained from the angles and their derivatives with Eq. (18) and Eq. (19). Using a numerical integration, 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

$$
T_{\min} \le T(t_i) \le T_{\max} , \tag{25}
$$

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

INVERSE DYNAMICS WITH QUATERNIONS

The IDVD method applied to quaternions as proposed by Boyarko is outlined in the first paragraph of this section.[7] The particular case of IDVD in time domain is then described.

Inverse Dynamics in the Virtual Domain

The polynomial functions used to express the Euler angles are not suitable for quaternion representation since the nonlinear unit norm condition must be preserved across the quaternion history. Consequently, the quaternion vector q is represented in the virtual domain $\tau \in [0; 1]$ with a product of n exponential functions[15]

$$
q(\tau) = q_0 \otimes \prod_{i=1}^{n} \exp(\omega_i^* \tilde{\beta}_{i,n}(\tau)) , \tag{26}
$$

where

$$
\tilde{\beta}_{i,n}(\tau) = \sum_{j=i}^{n} \binom{n}{j} (1-\tau)^{n-j} \tau^j \tag{27}
$$

and

$$
\omega_i^* = \log(\tilde{q}_{i-1}^{-1} \otimes \tilde{q}_i) . \tag{28}
$$

In Eq. (26) the symbol Π denotes the sequential quaternion product and $\tilde{q}_i$ are constant quaternions acting as control points, which satisfy the following conditions:

$$\tilde{q}_0 = q_0, \quad \tilde{q}_n = q_F \ .$$

The exponential and natural logarithmic maps are applied according to the following definition

$$q = \exp(v) = \begin{cases} \begin{bmatrix} \sin(|v|)\dfrac{v}{|v|} & \cos(|v|) \end{bmatrix}^T & |v| \neq 0 \\ \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T & |v| = 0 \end{cases}, \tag{29}$$

$$v = \log(q) = \begin{cases} \left(\dfrac{\arccos(q_4)}{\sqrt{q_1^2 + q_2^2 + q_3^2}} \right) \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T & \sqrt{q_1^2 + q_2^2 + q_3^2} \neq 0 \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T & \sqrt{q_1^2 + q_2^2 + q_3^2} = 0 \end{cases}. \tag{30}$$

In the previous expressions, the vector v is a vector whose norm is the Euler's rotation angle defined by the quaternion q and direction the Euler's axis.[7] Note that $|\exp(v)|=1 \ \forall \ v$ and therefore Eq. (26) automatically satisfies the unit norm constraint. The speed factor $\lambda(\tau)$ is defined with a polynomial as in Eq. (12). The virtual derivatives of Eq. (26) are calculated using the chain rule and their expressions can be found in Boyarko.[7] The time derivatives of the quaternion vector are calculated form its virtual derivatives using Eq. (13) and Eq. (14).

To allow the correct optimization using this approach, all the control points $\tilde{q}_i$ and the coefficients ω_i^* must be defined using the boundary conditions of the problem.[7] Consequently, the parameters to be optimized consist of the additional boundary conditions introduced to define $\tilde{q}_i$ and ω_i^* along with the polynomial coefficients of $\lambda(\tau)$. The number N_B of boundary conditions required to define all the parameters is

$$N_B = n + 1 \,. \tag{31}$$

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 coefficients ω_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 coefficients ω_i^* are defined through $\Omega_i^* = \begin{bmatrix} \omega_{ix}^* & \omega_{iy}^* & \omega_{iz}^* & 0 \end{bmatrix}^T$ with the following expressions:

$$\begin{cases} \Omega_1^* = \dfrac{q_0^{-1} \otimes \dot{q}_0}{5} \\ \Omega_2^* = \dfrac{q_0^{-1} \otimes \ddot{q}_0 - 2\left[\dbinom{5}{2} - 5(5-1) \right] \Omega_1^* - 5^2 \Omega_1^* \otimes \Omega_1^*}{2\dbinom{5}{2}} \end{cases}, \tag{32}$$

$$
\begin{cases}
\Omega_5^* = \dfrac{q_F^{-1} \otimes \dot{q}_F}{5} \\
\Omega_4^* = \dfrac{\tilde{q}_4^{-1} \otimes \left[\ddot{q}_F - 5^2 q_F \otimes \Omega_5^* \otimes \Omega_5^* - 2\begin{pmatrix} 5 \\ 5-2 \end{pmatrix} q_F \otimes \Omega_5^* \right] \otimes q_F^{-1} \otimes \tilde{q}_4}{2\left[\begin{pmatrix} 5 \\ 5-2 \end{pmatrix} - 5(5-1) \right]}
\end{cases}. \tag{33}
$$

All the control points $\tilde{q}_i$ and the coefficient Ω_3^* are then calculated using Eq. (28).[7] Note that Eq. (32) and Eq. (33) are written in general form and can be applied to any order *n* of the Bezier polynomial by substituting 5 with *n*. The expressions of angular velocity and accelerations are obtained with the inversion of Eq. (2) and its derivative

$$
\Omega = 2q^{-1} \otimes \dot{q}, \qquad \dot{\Omega} = 2q^{-1} \otimes \ddot{q} + \frac{1}{2} \Omega \otimes \Omega. \tag{34}
$$

The expressions of the torques are obtained with Eq. (23). 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. (26) along with the polynomial coefficients of $\lambda(\tau)$.

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. (26) is modified as

$$
q(t) = q_0 \otimes \prod_{i=1}^{n} \exp(\omega_i^* \tilde{\beta}_{i,n}(t)), \tag{35}
$$

with

$$
\tilde{\beta}_{i,n}(t) = \sum_{j=i}^{n} \begin{pmatrix} n \\ j \end{pmatrix} \left(1 - \frac{t}{t_F}\right)^{n-j} \left(\frac{t}{t_F}\right)^{j}. \tag{36}
$$

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 in virtual domain. Due to Eq. (35) and Eq. (36), the expressions of the coefficients ω_i^* for a fifth-order Bezier curve are given by

$$
\begin{cases}
\Omega_1^* = \dfrac{t_F q_0^{-1} \otimes \dot{q}_0}{5} \\
\Omega_2^* = \dfrac{q_0^{-1} \otimes \ddot{q}_0 - \dfrac{2}{t_F^2}\left[\begin{pmatrix} 5 \\ 2 \end{pmatrix} - 5(5-1) \right] \Omega_1^* - \left(\dfrac{2}{t_F} \right)^2 \Omega_1^* \otimes \Omega_1^*}{\dfrac{2}{t_F^2} \begin{pmatrix} 5 \\ 2 \end{pmatrix}}
\end{cases}, \tag{37}
$$

$$\begin{cases} \Omega_5^* = \dfrac{t_F q_F^{-1} \otimes \dot{q}_F}{5} \\ \Omega_4^* = \dfrac{\tilde{q}_4^{-1} \otimes \left[\ddot{q}_F - \left(\dfrac{2}{t_F} \right)^2 q_F \otimes \Omega_5^* \otimes \Omega_5^* - \dfrac{2}{t_F^2} \begin{pmatrix} 5 \\ 5-2 \end{pmatrix} q_F \otimes \Omega_5^* \right] \otimes q_F^{-1} \otimes \tilde{q}_4}{\dfrac{2}{t_F^2} \left[\begin{pmatrix} 5 \\ 5-2 \end{pmatrix} - 5(5-1) \right]} \end{cases} . \tag{38}$$

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 n-th order Bezier curve:

$$\sigma_j(\tau) = \sum_{i=0}^{n} a_{j,i} \beta_{i,n}(\tau) . \qquad j = 1,2,3 \tag{39}$$

The speed factor $\lambda(\tau)$ is defined with Eq. (12). The boundary conditions are imposed with Eq. (15) and Eq. (16) after substituting the j-th angle θ_j (and its derivatives) with the j-th 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:[10]

$$\omega = 4 \frac{1}{\left(1 + |\sigma|^2\right)^2} B(\sigma)^T \dot{\sigma} , \tag{40}$$

$$\dot{\omega} = \frac{1}{\left(1 + |\sigma|^2\right)^2} B(\sigma)^T \left[4\ddot{\sigma} - \dot{B}(\sigma)\omega \right] \tag{41}$$

The time derivatives of the MRP are obtained from the virtual derivatives with Eq. (13) and Eq. (14) by substituting the angle θ_j (and its derivatives) with σ_j. The controls are obtained with Eq. (19). Similarly to IDVD with Euler angles, the optimal control problem is consequently converted into an equivalent nonlinear programming problem, with constraints given by Eq. (20). The application of the method in the time domain is analogous at the case with the Euler angles.

NUMERICAL EXPERIMENTS

Numerical experiments have been performed to investigate the performance of the IDVD method by evaluating the effects of the virtual domain and attitude representation on solution quality and computational speed. All the computations were carried out on a Windows 7 laptop computer with an Intel 2.27GHz i5 M430 processor and 4Gb of RAM.

Several test scenarios were examined using the proposed IDVD methods in a normalized form. All the maneuvers addressed in this paper are rest-to-rest, but the proposed methods can be extended to any angular velocity at endpoints.[7] To evaluate the impact of the presence of the virtual domain, a same problem was solved in the virtual domain and in the time domain separately, using an analogous polynomial to represent the attitude trajectory. A fifth-order Bezier function was used to represent the quaternion history, while fifth- and seventh-order polynomials were used for MRP and Euler angles histories. A fifth-order polynomial and a third-order polynomial were employed to represent $\lambda(\tau)$ for the minimum-time problem and minimum-energy problem respectively. Moreover, Eq. (8) was integrated using the Gauss-Legendre Quadrature method with 4 points. The nonlinear programming problem obtained from IDVD was solved using the Sequential Quadratic Solver method (SQP) of MATLAB fmincon.[13] Reference solutions were computed using the Gauss Pseudospectral Optimization Software GPOPS, since it is open source and freely available.[15] The following difference in percentage ΔJ is introduced to compare the solution J_{IDVD} obtained with IDVD with the reference optimal J_{GPOPS} computed with GPOPS

$$\Delta J = 100 \frac{| J_{IDVD} - J_{GPOPS} |}{J_{GPOPS}}. \tag{42}$$

The following performance index

$$MI = 0.6\Delta J + 0.4 CPUt \tag{43}$$

is also introduced to estimate the overall performance of each IDVD guidance by considering the computational time $CPUt$ and the solution quality ΔJ: the lower the performance index, the better the overall performance. Note that ΔJ and $CPUt$ have the same order of magnitude and the best possible performance is obtained when both values are close to zero. The coefficients have been chosen in such a way that the solution quality has higher impact on the performance than the computational time. All the solutions were propagated with MatLab ODE 45 using Eq. (1) and Eq. (2) with a time step of 0.01s. The errors of final angular velocity and attitude after propagation of the controls are denoted with $\Delta\omega$ and $\Delta\theta$ respectively.

Minimum-time maneuver: 180° 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 body and endpoint conditions[2]

$$\begin{array}{c} I = diag([1,1,1]) \\ -1 \le T \le 1 \\ \omega_0 = [0,0,0] \quad \theta_0 = [0,0,0]\deg \quad q_0 = [0,0,0,1] \quad \sigma_0 = [0,0,0] \\ \omega_F = [0,0,0] \quad \theta_F = [0,0,180]\deg \quad q_F = [0,0,1,0] \quad \sigma_F = [0,0,1] \end{array}. \tag{44}$$

Inertia, torques and angular velocities are normalized. The maneuver time is constrained in the interval [2 10] s. 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. (15)-(16)-(23)-(24)-(32)-(33); 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. Initial guesses for all the other polynomial coefficients are zero. Finally, 50 nodes are considered for imposing the constraints.

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.[2] The optimal controls have a bang-bang nature (see Figure 3) and the consequent optimal attitude trajectory has a precessional component and is represented in Figure 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 showed in Figure 3. This is due to the nature of the method: the controls are calculated as combination of the trajectory coordinates and their derivatives through the inversion of the dynamics. Since the trajectory is represented with polynomials, the controls result nonlinear continuous function of 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. 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.

Table 1. Optimization of a minimum-time rest-to-rest 180 deg reorientation of a symmetric spacecraft using IDVD and GPOPS. Trajectories and controls obtained with the marked methods are represented in Figure 2 and Figure 3.

Attitude	Method	J (s)	CPUt (s)	$\Delta\omega$ (deg/s)	$\Delta\theta$ (deg)	ΔJ (%)	MI
Modified Rodrigues Parameters	GPOPS*	3.2431	68.7	1.8e-1	6.9e-3	0	27.5
	IDVD 5th - time	3.8153	0.3	3.0e-1	7.4e-3	17.6	10.7
	IDVD 5th - virtual	3.4351	2.2	5.2e-2	3.7e-2	5.9	4.4
	IDVD 7th – time*	3.4853	0.7	2.6e-1	4.7e-3	7.5	4.7
	IDVD 7th – virtual*	3.3179	7.7	1.3e-1	3.9e-2	2.3	4.5
Euler Angles	GPOPS	3.2431	93.2	1.8e-1	6.9e-3	0	37.3
	IDVD 5th - time	3.6871	0.3	1.6e-1	6.3e-3	13.7	8.3
	IDVD 5th - virtual	3.4443	1.6	5.6e-1	3.4e-1	6.2	4.4
	IDVD 7th – time	3.5277	1.1	1.2e-1	5.3e-3	8.8	5.7
	IDVD 7th – virtual	3.3302	13.5	2.8e-1	1.4e-4	2.7	7.0
Quaternion	GPOPS	3.2431	100.3	1.8e-1	6.9e-3	0	40.1
	IDVD 5th – time	3.5418	2.0	9.7e-2	5.0e-3	10.4	7.1
	IDVD 5th - virtual	3.3971	7.8	2.4e-2	5.4e-2	4.7	5.9

The presence of the virtual argument improves the sub-optimal solution with respect to the same polynomial formulation in the time domain, for every attitude coordinate 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. As a result, the sub-optimal solution is improved. Figure 3 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 (see also Eq. (13) and Eq. (14)), which result closer to the bang-bang optimal shape. The other effect of the virtual domain is the increment of computational time required to the solver due to the additional variable parameters (see Eq. (20)) and constraints and the integration of Eq. (8).

The attitude parameter affects the computational time and solution quality. The IDVD applied to quaternions is slower than IDVD applied to MRP and Euler angles due to the usage of the exponential and logarithmic maps to maintain the unit norm along the quaternion trajectory. The overall best optimal solution is obtained with the seventh-order IDVD applied to the MRP in the virtual domain. The 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.

Figure 1 summarizes the performance of IDVD based on the performance index *MI*. The overall best performances are obtained with IDVD applied to MRP with a fifth-order polynomial in the virtual domain and a seventh-order polynomial in the time domain. The worst performance is obtained with fifth-order IDVD in time domain, due to the high ΔJ.

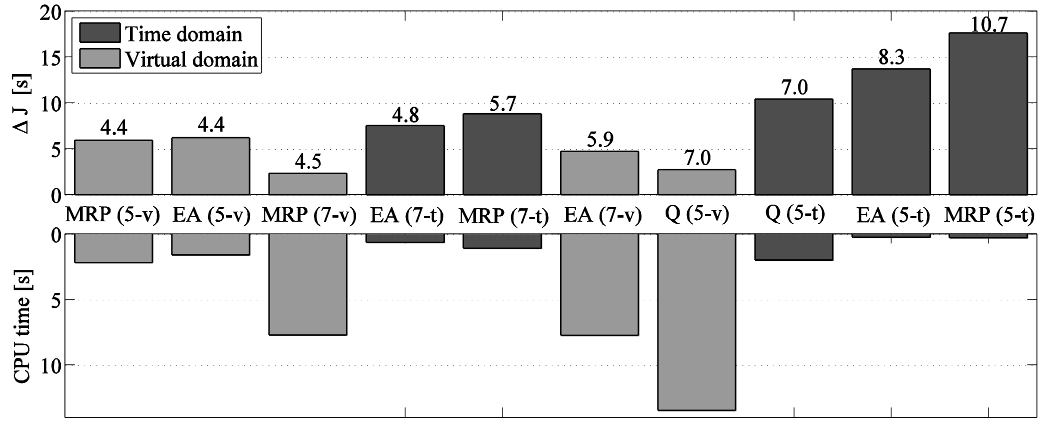

Figure 1. Performance of IDVD for the minimum-time 180 deg reorientation. The symbol Q denotes the quaternion, MRP denotes the modified Rodrigues parameters and EA indicates the Euler angles. The order of IDVD and domain are indicated in parenthesis. The graph is ordered according to the performance index, which is reported on the top of each bar.

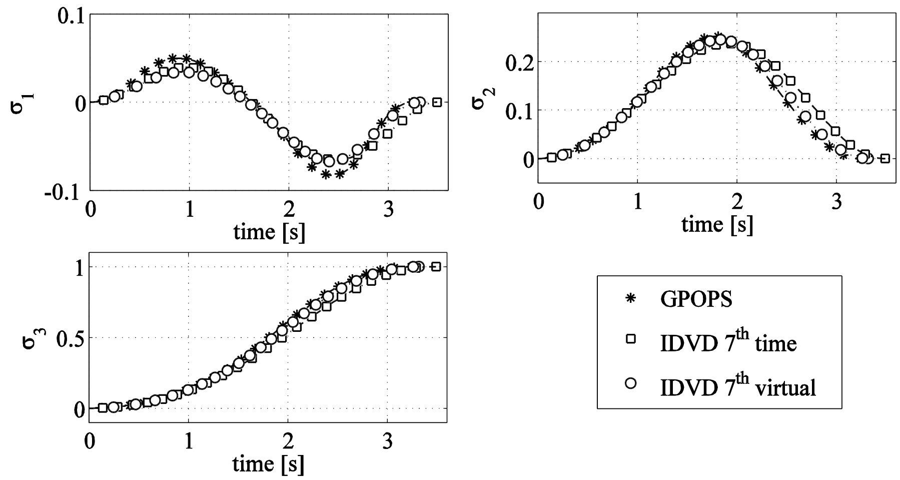

Figure 2. Optimal attitude trajectory in terms of modified Rodrigues parameters, generated with GPOPS, seventh-order IDVD in virtual domain and seventh-order IDVD in time domain, applied to the Modified Rodrigues Parameters.

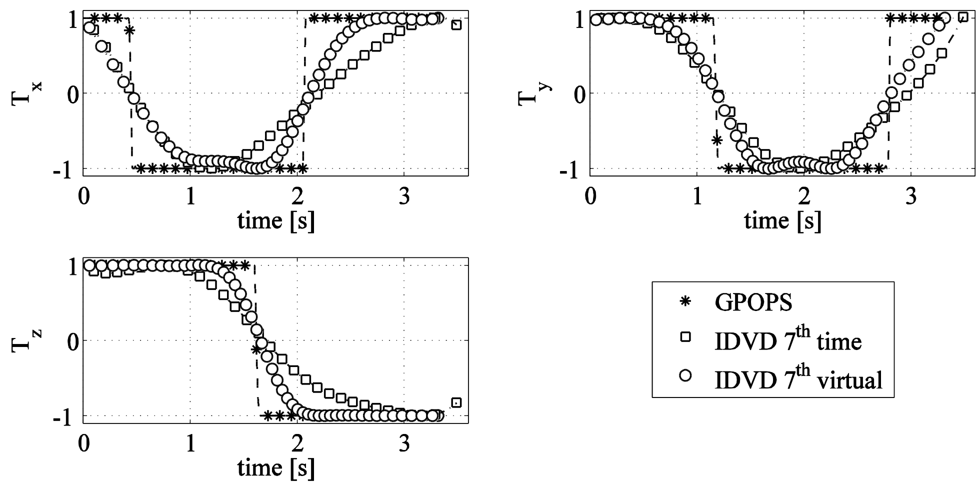

Figure 3. Optimal control time history generated with GPOPS and seventh-order IDVD applied to the modified Rodrigues parameters in virtual domain and time domain.

Minimum-time maneuver: Monte Carlo analysis for inertia and maneuver

Additional test scenarios have been generated by varying inertia, control bounds and reorientation angles with the Monte Carlo method. Data of these scenarios are summarizes in Table 2. The symbol $x_{A,B}$ denotes a random value of x in the range $[A,B]$. Similarly to the 180 deg reorientation, the test cases have been analyzed using IDVD. 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.

Table 2. Reference scenarios generated with Monte Carlo method.

Scenario	Inertia	T_{MAX}	θ_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]	[$\theta_{-180,180}$, $\theta_{-180,180}$, $\theta_{-180,100}$]	1000

Table 3 summarizes the performance of IDVD. The symbol σ_x denotes the standard deviation of the data x. The virtual domain improves the solution quality, but at the cost of more computational time, for any polynomial order and attitude parameter considered. 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 case of IDVD in time domain, the convergence percentage is 100%. In case of IDVD in virtual domain, the convergence of the solver is lower. In fact, the first guesses on the polynomial coefficients of $\lambda(\tau)$ were not accurate since these parameters are not related to any state variable or physical behavior of the spacecraft. As a consequence, the optimization solver failed in some cases due to the bad first guesses. The attitude representation affects the solution quality and computational time. As in the ideal 180 deg reorientation scenario, the overall best average optimal is obtained with a seventh-order IDVD applied to MRP in the virtual domain. However, all the tested methods have a large standard deviation on ΔJ and *CPUt*.

Table 4 summarizes the errors in final angular velocity and attitude after propagation of the controls obtained from IDVD. The IDVD in time domain is slightly more accurate that IDVD in virtual domain: the maximum errors on angular velocity and attitude in case of IDVD in time domains are 0.01 deg/s and 0.25 deg, while the maximum errors in case of IDVD in virtual domain are 0.82 deg/s and 0.37 deg.

Figure 4 shows the performance of IDVD based on the performance index. Similarly to the 180 deg re-orientation of a symmetric spacecraft, the best performance are obtained with fifth-order IDVD applied to MRP in the virtual domain and seventh-order IDVD applied to MRP in time domain, since they compute sub-optimal solutions in a short amount of time. Notably, IDVD applied to the Euler angles has the highest values of performance index.

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

Attitude	Method	Conv. (%)	ΔJ (%)	$\sigma_{\Delta J}$ (%)	CPUt (s)	σ_{CPUt} (s)	MI
Modified Rodrigues Parameters	IDVD 5th - time	100	19.0	21.8	0.4	0.2	11.5
	IDVD 5th - virtual	99.9	8.7	15.1	3.2	1.5	6.5
	IDVD 7th – time	100	10.0	16.2	1.7	1.1	6.7
	IDVD 7th – virtual	97.4	6.4	13.8	15.0	5.5	9.8
Euler angles	IDVD 5th - time	100	27.2	31.2	0.4	0.7	16.5
	IDVD 5th - virtual	98.1	18.7	26.7	4.0	15.6	12.9
	IDVD 7th – time	100	20.7	27.8	1.9	1.8	13.2
	IDVD 7th – virtual	95.7	14.9	22.6	15.6	34.8	15.2
Quaternions	IDVD 5th - time	100	14.7	17.2	2.8	1.5	9.9
	IDVD 5th - virtual	99.7	12.0	20.1	13.3	8.2	12.5

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)	Δθ (deg)	$\sigma_{\Delta\theta}$ (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

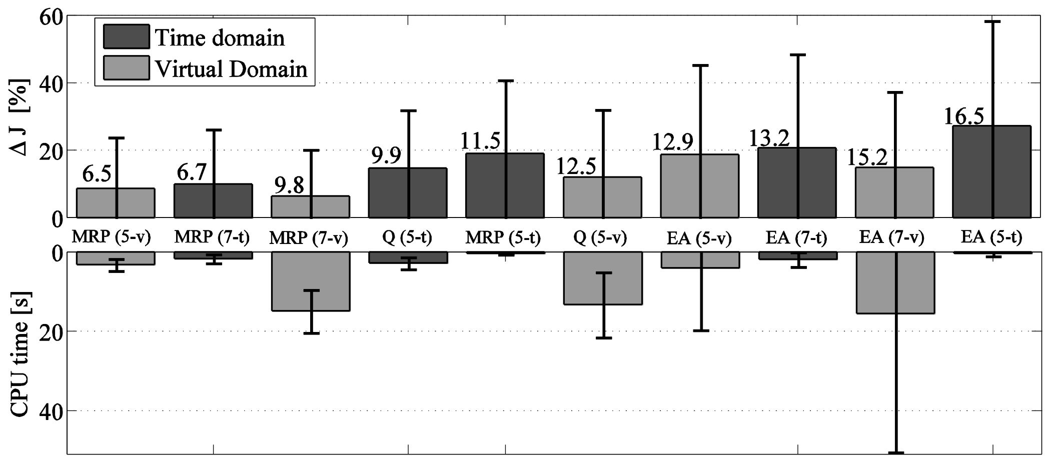

Figure 4. Performance of IDVD for several pseudorandom spacecraft geometry and maneuvers. The symbol Q denotes the quaternion, MRP denotes the modified Rodrigues parameters and EA indicates the Euler angles. Order of IDVD and domain are indicated in parenthesis. The graph is ordered according to the performance index, which is reported on the top of each bar. The vertical lines indicate the standard deviations.

Minimum-energy maneuver: 180° reorientation of a symmetric spacecraft

Similarly to the minimum-time analysis, an 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. (44) and the cost function is given by Eq. (6). The maneuver time is constrained in the interval [2 10] s. The first guesses for the normalized 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 given with Eqs. (15)-(16)-(23)-(24)-(32)-(33). The guess on the maneuver duration is 10 s. The trapezoidal rule is used to numerically integrate the cost function in Eq. (9). The *x* values used for integration are the instants of time corresponding to 100 equally spaced nodes in the virtual domain. The *y* values are the integrand function evaluated at the virtual nodes. To ensure convergence, a third-order polynomial is employed to represent the speed factor.

As summarized in Table 5, the optimal solution computed by GPOPS can be obtained with IDVD applied to Euler angles and quaternion in time domain with difference percentile of 0.02 %. In fact, the optimal controls which minimize the maneuver energy do not saturate and their continuous shape can be represented by polynomial functions. This behavior is graphically showed in Figure 6, 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.

Figure 5 graphically summarizes the performance of IDVD based on the performance index defined in Eq. (43). The overall best performances are obtained with IDVD in time domain due to the solution quality and the low computational time required by the solver. The presence of the virtual domain increases the computational time only and therefore the performances are lower.

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 Figure 6.

Attitude	Method	J	CPUt (s)	$\Delta\omega$ (deg/s)	$\Delta\theta$ (deg)	ΔJ (%)	MI
Quaternion	GPOPS*	5.922e-2	198.8	2.8e-3	2.9e-2	0	55.9
Euler angles	IDVD 5^{th} – time*	5.923e-2	0.7	4.9e-11	1.8e-4	0.02	0.3
	IDVD 5^{th} – virtual*	5.923e-2	1.7	4.4e-6	3.8e-4	0.02	0.7
Modified Rodrigues parameters	IDVD 5^{th} - time	5.935e-2	1.0	5.3e-6	8.0e-5	0.22	0.5
	IDVD 5^{th} - virtual	5.919e-2	5.8	3.3e-2	2.9e-2	0.05	2.4
Quaternion	IDVD 5^{th} - time	5.923e-2	1.1	3.3e-12	1.8e-4	0.02	0.5
	IDVD 5^{th} - virtual	5.923e-2	16.8	1.9e-5	3.3e-4	0.02	6.7

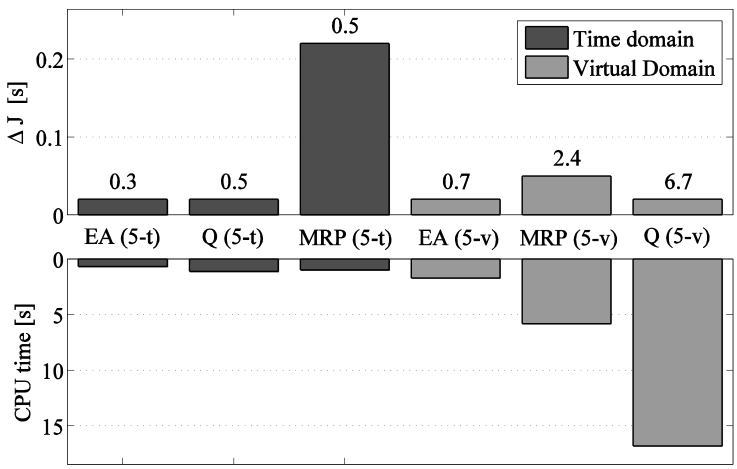

Figure 5. Performance of IDVD for the minimum-energy 180 deg reorientation. The symbol Q denotes the quaternion, MRP denotes the modified Rodrigues parameters and EA indicates the Euler angles. Order of IDVD and domain are indicated in parenthesis. The graph is ordered according to the performance index.

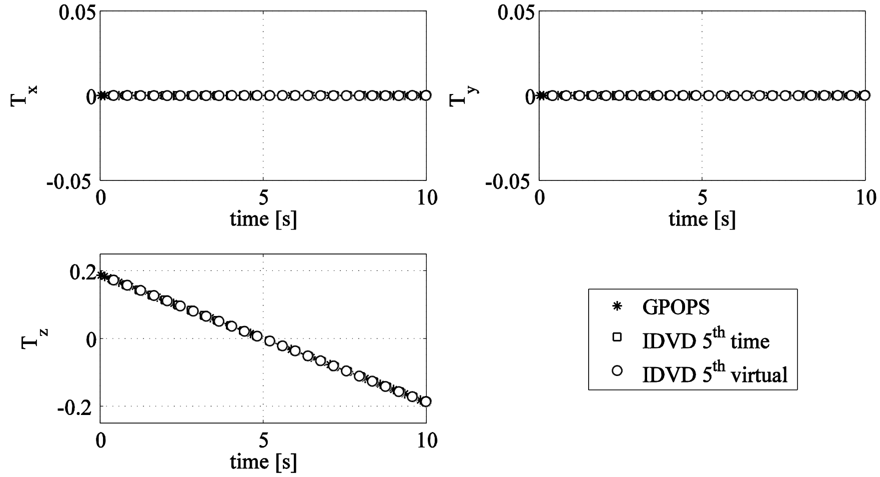

Figure 6. Optimal control time history generated with GPOPS and fifth-order IDVD applied to the Euler angles in virtual domain and time domain.

Note about singularities

In a typical reorientation range such as [-180; 180] deg, quaternions and MRP do not present any singularity. Problems may occur using the Euler angles since the values of $\dot{\theta}$ at endpoints are calculated using the kinematic Eq. (3). Clearly, singularities occur when $\theta_2 = \pm 90$ deg.

CONCLUSION

The application of the inverse dynamics approach to the rapid optimal attitude trajectory generation for spacecraft reorientation has been investigated. The IDVD method was applied to the modified Rodrigues parameters, Euler angles and quaternion with different polynomial orders defined in both time and virtual domains.

In the case of the considered minimum-energy maneuver, IDVD applied in 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.

In the case of minimum-time maneuvers, the optimal bang-bang shape of the controls cannot be achieved with the considered approach, which computes sub-optimal solutions only due to the nature of the method. Representation of the attitude coordinates in the virtual domain improves the solution quality with respect to the same polynomial representation 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 exponential maps; modified Rodrigues parameters and Euler angles are faster since no constraints on the attitude history must be respected. Better results have been obtained using the modified Rodrigues parameters, since each attitude coordinate is more suitable for high-order polynomial representation. The overall best performance in terms of compromise between computational time and solution quality have been obtained with fifth-order IDVD applied to modified Rodrigues parameters in virtual domain or seventh-order IDVD applied to the modified Rodrigues parameters in time domain, since these methods produce sub-optimal solutions in a short amount of time. However, the use of the virtual domain requires accurate first guesses on the polynomial coefficients of the speed factor in order to ensure 100% convergence.

The inverse dynamics approach can be used as optimal guidance for spacecraft reorientation. 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 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] X. Bai and J.L. Junkins, "New Results for Time-Optimal Three Axis Reorientation of a Rigid Spacecraft." *Journal of Guidance, Control, and Dynamics*. Vol. 32, No. 4, 2009, pp. 1071–1076.

[2] 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.

[3] M.V. Dixon, T.N. Edelbaum, J.E: Potter and W.E. Vandervelde, "Fuel Optimal Reorientation of Axisymmetric Spacecraft." *Journal of Spacecraft and Rockets*. Vol. 7, Issue 11, 1970, pp. 1345–1351.

[4] B.M. Yutko and R.G. Melton, "Optimizing spacecraft reorientation maneuvers using a pseudospectral method." *Journal of Aerospace Engineering, Sciences and Applications*. Vol. 2, No. 1, 2010.

[5] R.G. Melton, "Hybrid methods for determining time-optimal, constrained spacecraft reorientation maneuvers." *Acta Astronautica*. Vol. 94, 2014, pp. 294–301.

[6] C. Louembet, F. Cazaurang, A. Zolghadri, C. Charbonnel and C. Pittet, "Design of Algorithms for Satellite Slew manoeuver by Flatness and Collocation." *Proceedings of the 2007 American Control Conference*. New York, USA, July 11-13, 2007.

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

[8] O.A. Yakimenko, "Optimization of Holonomic Attitude Dynamics using IDVD Method." *3rd Internationa Symposium on Systems and Controls in Aeronautics and Astronautics (ISSCAA)*. Harbin, China, June 8-10, 2010.

[9] B. Wie, "Space Vehicle Dynamics and Control." *AIAA Education Series*. 1998, pp. 403-407.

[10] M.D. Shuster, "A survey of Attitude Representations." *The Journal of Astronautical Sciences*. Vol. 41, No. 4, 1993, pp.439-517.

[11] O.A. Yakimenko, "Direct Method for Rapid Prototyping of Near Optimal Aircraft Trajectories." *Journal of Guidance, Control, and Dynamics*. Vol. 23, No. 5, 2000, pp. 865-875.

[12] M. Ciarcia, M. Romano, "Suboptimal Guidance for Orbital Proximity Maneuver with Path Constraints Capability." *Proceedings of the AIAAA Guidance, Navigation and Control Conference*. 2012.

[13] MatLab Version 2013b User manual, The Mathworks, Natick, MA, 2013.

[14] A.V. Rao, D.A. Benson, C.L. Darby, M.A. Patterson, C. Francolin, I. Sanders and G.T. Huntington, "GPOPS: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems using the Gauss Pseudospectral Method." *ACM Transaction on Mathematical Software*. Vol. 37, No. 2, 2010.

[15] 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*. ACM, New York, 1995, pp. 369-376.