Optimal reorientation of asymmetric underactuated spacecraft using differential flatness and receding horizon control

Wei-wei Cai*, Le-ping Yang, Yan-wei Zhu

College of Aerospace Science and Engineering, National University of Defense Technology, PR China

Received 24 June 2014; received in revised form 12 October 2014; accepted 14 October 2014
Available online 22 October 2014

Abstract

This paper presents a novel method integrating nominal trajectory optimization and tracking for the reorientation control of an underactuated spacecraft with only two available control torque inputs. By employing a pseudo input along the uncontrolled axis, the flatness property of a general underactuated spacecraft is extended explicitly, by which the reorientation trajectory optimization problem is formulated into the flat output space with all the differential constraints eliminated. Ultimately, the flat output optimization problem is transformed into a nonlinear programming problem via the Chebyshev pseudospectral method, which is improved by the conformal map and barycentric rational interpolation techniques to overcome the side effects of the differential matrix’s ill-conditions on numerical accuracy. Treating the trajectory tracking control as a state regulation problem, we develop a robust closed-loop tracking control law using the receding-horizon control method, and compute the feedback control at each control cycle rapidly via the differential transformation method. Numerical simulation results show that the proposed control scheme is feasible and effective for the reorientation maneuver.

© 2014 COSPAR. Published by Elsevier Ltd. All rights reserved.

Keywords: Differential flatness; Mapped Chebyshev pseudospectral method; Differential transformation; Receding-horizon control; Underactuated spacecraft

1. Introduction

Recent advantages in satellite technology have greatly promoted the attitude-control capabilities for small spacecraft, including higher agility, higher-precision pointing and/or slewing. Most of the existing attitude control schemes are developed for the actively controlled spacecraft that is equipped with sufficient actuators equal to, or more than, the number of degrees of freedom to be controlled (Alkhodari and Varatharajoo, 2009; Weiss et al., 2013; Wu et al., 2011). However, if actuators in one or two dimensions fail, the spacecraft, which is called underactuated, will be prevented from performing arbitrary maneuvers. In this case, developing suitable control strategies for the underactuated spacecraft is a cost-reducing alternative, compared with equipping the spacecraft with some redundant actuators, to improve the system’s reliability.

In the field of underactuated spacecraft control, many researches focusing on the controllability and stabilization problem have been carried out. Godard and Kumar (2011) investigated a robust sliding mode control scheme to stabilize the attitude of the spacecraft subject to actuator failures, external disturbances and physical parameter uncertainties. Based on the sequential Euler angle rotation strategy, Kim and Turner (2014) developed the necessary
conditions for coupling sub-maneuvers and addressed the near-minimum-time control problem of asymmetric underactuated spacecraft with bounded input magnitudes. For an axisymmetric spacecraft with only two control torques, the attitude stabilization and command tracking problems are addressed using a computationally efficient $H_{\infty}$ control design procedure (Zheng and Wu, 2009). Making use of the generalized dynamic inverse method and a novel saturated function, a full-state feedback control law is proposed for an underactuated spacecraft with bounded wheel speeds (Gui et al., 2013). Note that due to the inherent nonholonomic constraints, some trajectories in the configuration space are just infeasible, or cannot be steered, for an underactuated spacecraft, leaving it a challenge to reorientation controller design. It has been demonstrated that it is efficient to address the reorientation control problem for an underactuated spacecraft in two steps: nominal reorientation generation and trajectory tracking. Considering the obstacles along the angular path and constraints on admissible rotation axes, an attitude path planning strategy for the single-axis pointing of an underactuated spacecraft is studied (Angelis et al., in press). Taking the underactuated spacecraft and flywheels as a whole system, Zhang et al. (2014) planned the attitude trajectory which minimizes the angular velocity change of flywheels by Gauss pseudospectral method. Given a rigid spacecraft with only two control torques due to actuator failures, the reorientation trajectory with minimum control efforts are generated by using the sequential control concept, which aims to avoid the axis where the actuator failure has occurred (Kim et al., 2013a,b). Recently, Kim et al. (in press) has suggested a novel Davidenko-like homotopy algorithm to achieve the optimal reorientation trajectory of an underactuated spacecraft by starting from the associated trajectory where all three actuators are available. Aguilar (2005) investigated an asymptotic trajectory tracking control law to steer the spacecraft maneuver along a given attitude trajectory. Based on the latest development in nonlinear system and optimal control theories, this paper aims to develop a novel optimal attitude maneuver controller for an asymmetric spacecraft with only two control torque inputs available.

The nominal maneuver trajectory for an underactuated spacecraft is often designed by formulating a constrained nonlinear optimal control problem. Since the system is strongly coupled and highly nonlinear, the analytical solutions are seldom available or even possible, hence the superiority of numerical methods. However, the solution efficiency of numerical methods is often compromised by computational efficiency, which is mainly induced by the treatment of differential constraints. In recent years, the differential flatness based method has emerged as one of the promising numerical techniques for aerospace applications (Chamseddine et al., 2012; Morio et al., 2009). Based on the system’s flatness property, the states and inputs are formulated as functions of the flat output and its derivatives, thus transforming the original trajectory optimization problem into the flat output space, which means that all the differential constraints are eliminated. Tsintotas (2000) applied the differential flat theory to the problem of feasible trajectory generation for a spacecraft with two control torques, and the resulted trajectories could be used for closed-loop tracking controller design. In addition, the flatness of a general underactuated spacecraft was discussed in (Zhuang et al., 2012), and then the continuous flat system was discretized by a fixed count of control steps with the sampling periods as decision variables. Though the design procedure has been greatly simplified by using the differential flat theory, the resulted trajectory may only be feasible, not optimal. In this paper, we mainly focus on the reorientation trajectory optimization and tracking problem using differential flatness and receding horizon control techniques.

The computational framework facilitated by the pseudospectral (PS) methods applies quite naturally and easily to the flat output optimization problem. Note that the approximation accuracy of the high order derivatives should be considered since they are required in the framework of flatness based trajectory optimization. The PS method employs globally orthogonal polynomial approximations for the flat output with its values at suitably chosen discretization points as the expansion coefficients. Compared with other discretizing schemes, the PS approximations offer a higher degree of accuracy with much fewer nodes. And the differentiation matrix based computation approach for the high order derivatives of flat output at the selected nodes is more accurate and effective. Among various PS methods, the Chebyshev PS method (CPM) is somewhat more attractive since the Chebyshev expansion is very close to the best polynomial approximation of a given function in the infinity norm (Trefethen, 2000). Moreover, the Chebyshev–Gauss–Lobatto (CGL) nodes and Chebyshev differentiation matrix can be evaluated in closed form, providing prominent computational advantages. Thus the CPM is selected to transform the flat outputs optimization problem here. However, a direct application of the CPM may result in great degradation of the accuracy for high order derivatives of flat output due to the ill-condition of the Chebyshev differentiation matrix (Mead and Renaut, 2002). A mapped Chebyshev pseudospectral method (MCPM) based on the conformal map and barycentric rational interpolation techniques is proposed in this paper to improve the ill condition so as to enhance the numerical computational performance.

In practice, an underactuated spacecraft would deviate from the nominal trajectory due to inevitable external disturbances and/or model uncertainties, calling for the closed-loop trajectory tracking control. The receding horizon control, with which closed-loop stability can always be achieved, has been used as an efficient tracking method for various systems (Lu, 1999; Peng et al., 2013). The implementation of this tracking control law requires on-line solution to the resulted two-point boundary value problem (TPBVP) over a shorter moving horizon. However, the
conventional solving methods, such as backward sweep method and transformation matrices method, are potentially unstable, numerically intensive and time-consuming. Thus if highly efficient numerical algorithm can be developed to reduce the computation burden in solving the TPBVP, the implementation of receding horizon control method should be quite appealing. In a recent work, the TPBVP is solved efficiently with the differential transformation method, avoiding integrating the matrix Riccati equation or inverting the ill-conditioned transition matrices (Saberi Nik et al., 2013). Compared to the conventional methods, the differential transformation can greatly reduce computational complexity, suitable for on-line implementation.

The remainder of this paper is organized as follows. Firstly, a general asymmetric underactuated rigid spacecraft model with only two available control torque inputs is presented. Secondly, the reorientation trajectory optimization problem is formulated, and transformed into a non-linear programming problem by the differential flatness and mapped Chebyshev pseudospectral method. Thirdly, an optimal feedback tracking controller based on differential transformation is presented in detail. Finally, numerical simulation is carried out to validate the performance of the proposed nominal trajectory optimization approach and closed-loop tracking control scheme. Some usefully conclusions are put forward in the end.

2. Underactuated spacecraft model

The reorientation motion of a rigid spacecraft can be described by Euler’s rotational equations of motion. Written in the body-fixed reference frame whose origin is located at the spacecraft mass center, and coordinates are aligned with the principal axes of inertia, the vector equation of governing dynamics is

\[ I \ddot{\omega} + [\omega \times I \omega] = u \]  

(1)

Since we only focus on the two-input asymmetric underactuated spacecraft, it is, without loss of generality, reasonable to assume that there is no control authority along the 3-axis, then the aforementioned dynamics expands the scalar equations as

\[
\begin{align*}
I_1 \ddot{\omega}_1 &= (I_2 - I_3) \omega_2 \omega_3 + u_1 \\
I_2 \ddot{\omega}_2 &= (I_3 - I_1) \omega_1 \omega_3 + u_2 \\
I_3 \ddot{\omega}_3 &= (I_1 - I_2) \omega_1 \omega_2
\end{align*}
\]  

(2)

where \( I = \text{diag}(I_1, I_2, I_3) \), \( \omega = [\omega_1, \omega_2, \omega_3]^T \) and \( u = [u_1, u_2]^T \) denote the inertia matrix, the angular velocity vector and the external torques along the principle axes, respectively.

In turn, using the body 3-2-1 sequenced rotation, the rotational kinematics can be described by Euler angles:

\[
\begin{align*}
\dot{\phi} &= \omega_1 + (\omega_2 \sin \phi + \omega_3 \cos \phi) \tan \theta \\
\dot{\theta} &= \omega_2 \cos \phi - \omega_3 \sin \phi \\
\dot{\psi} &= (\omega_2 \sin \phi + \omega_3 \cos \phi) \sec \theta
\end{align*}
\]  

(3)

where \( \phi, \theta \) and \( \psi \) are roll, pitch and yaw angles, respectively.

3. Reorientation trajectory optimization

3.1. Problem statement

The problem in question is to find the state-control profile \([x(t), u(t)]\) which steers the underactuated spacecraft from the initial orientation \(x_0\) to the desired one \(x_f\), minimizing the performance index \(J(x, u)\) and observing some constraints \(C(x, u)\). Here \(x = [\phi, \theta, \psi, \omega_1, \omega_2, \omega_3]^T\) denotes the state variable vector. Besides the governing equations in Eqs. (2) and (3), the constraints \(C(x, u)\) also include the limitations on allowable control inputs:

\[ u_{\min} \leq u_i \leq u_{\max}, \quad (i = 1, 2) \]  

(4)

Generally the performance criterion could be written in Bolza form as

\[ J = \Phi[x(t_0), x(t_f), t_0, t_f] + \int_{t_0}^{t_f} A[x(t), u(t), t] dt \]  

(5)

where \(\Phi(\cdot)\) denotes the Mayer cost and \(A(\cdot)\) the Lagrangian cost.

For the spacecraft reorientation problem, considering the agile attitude maneuver capability required by many civilian and military space missions, the minimum maneuver time in Eq. (6) is an important performance criterion. On the other hand, the minimum control effort in Eq. (7) constitutes another important index since it indicates the engineering feasibility of space missions, especially for those realized by propellant-based actuators.

\[ J_1 = t_f \]  

(6)

\[ J_2 = \int_{t_0}^{t_f} (u_1^2 + u_2^2) dt \]  

(7)

Using the classical physical programming method, both of these two performance criteria are optimized simultaneously in our research. A detailed description of the physical programming method is given in (Messac, 1996). Here the fundamental concepts are presented for better understanding. In the physical programming procedure, the designer’s preference with respect to each performance criterion is categorized into eight different subclasses. Specifically, the criteria in Eqs. (6) and (7) that are expected to be minimized belong to the so-called class 1-S. Moreover, the class functions are defined to quantitatively describe the spectrum of preference for a given performance criterion. The mathematical nature of the class function for class 1-S is illustrated in Fig. 1, where \(J_i\) denotes the \(i\)th criterion, and \(p_i\) the class function. The horizontal axis is divided into six intervals by the user-specified parameters \([J_0, \ldots, J_3]\), and the degrees of desirability for each interval are defined as ideal, desirable, tolerable, undesirable, highly undesirable and unacceptable, respectively.
For the case of class 1-S in our research, the piecewise
and second-order continuous function proposed by
(Messac, 1996) is applied to the cost function representa-
tion. Obviously, the class functions help transform the ori-
ginal design metrics with different units and physical
meaning into non-dimensional, strictly positive real
cumbers. Given the class functions for each performance
criterion, an aggregate objective function is proposed by
\[
\min P = \log_{10} \left( \frac{1}{n_c} \sum_{i=1}^{n_c} p_i(J_i(x, u)) \right) \tag{8}
\]
where \(n_c\) is the number of original performance criteria.
Thus the original multi-objective optimization problem is
converted and can be solved using single-objective optimi-
ization algorithms.

3.2. Flatness property of the rotational dynamics

Differential flatness reveals a structural property of
general nonlinear systems, denoting that all states and
inputs can be expressed in terms of a set of differentially
independent variables and their time derivatives (Lévine,
2011). More precisely, consider the nonlinear system in
the general form:
\[
\dot{x}(t) = f(x(t), u(t), t)
\]
where \(f(\cdot)\) is a smooth nonlinear function, \(x \in \mathbb{R}^n\) is the
state vector and \(u \in \mathbb{R}^m\) the control input vector with
\(m \leq n\). The system is differential flat if and only if there is a
vector \(\xi \in \mathbb{R}^n\) with differentially independent compo-
nents, so-called flat outputs, such that
\[
\begin{align*}
\begin{cases}
x &= \Gamma_x(\xi, \xi_1, \ldots, \xi_{(n-1)}) \\
u &= \Gamma_u(\xi, \xi_1, \ldots, \xi_{(n)})
\end{cases}
\end{align*}
\tag{10}
\]
where \(\Gamma_x, \Gamma_u\) are smooth functions, \(\xi^{(k)}\) is the \(k\)th order time
derivative of the \(i\)th component of the vector \(\xi\), and
\(\eta = [\eta_1, \ldots, \eta_m]^T\) the relative degree of \(\xi\):
\[
\eta_i = \min \left\{ k \in \mathbb{N}^+ \mid \exists j \in \{1, \ldots, m\} \mid \frac{\partial \xi^{(k)}}{\partial u_j} \neq 0 \right\},
\tag{11}
(i = 1, \ldots, m)
\]
For the general asymmetric underactuated spacecraft
studied, the concept of pseudo control inputs and the cor-
responding constraints are introduced in our research to
make the underactuated spacecraft explicitly flat. This idea
is based on the fact that more control inputs a dynamic sys-
}
and path constraints
\[ I_w(\tilde{z}(t), t_0, t_f) \leq u_{\text{max}}, \quad (k = 1, 2) \]  
\[ I_T(\tilde{z}(t), t_0, t_f) = 0 \]  

Since the flat output represents a minimal description of the system’s behavior, the number of design variables for the orientation trajectory optimization problem has been greatly reduced after converting into the flat space. Moreover, the flatness transformation has entirely eliminated the original differential constraints, generating an integration-free geometric programming problem, which is computationally tractable and quickly solvable. Although the feasible region in the flat output space is usually non-convex, which may affect the search for the global optimal solution, the side effects could be improved by approximating the feasible region with polytopes or superquadric surfaces (Louemiet et al., 2009).

3.3. Flat output optimization via MCPM

In the MCPM, the function to be determined is approximated by barycentric rational interpolation with the values of this function at mapped CGL nodes as the expansion coefficients. The derivatives of the function at the mapped CGL nodes can be obtained directly with the barycentric weight:
\[ o_k^\text{bary}(k = 0, 1, \ldots, N - 1) \]

Utilizing the famous Kosloff and Tal-Eaer conformal map, the mapped CGL nodes are given as (Kosloff and Tal-Ezzer, 1993)
\[ \lambda_k = g(\tau_k, x) = \frac{\sin(\pi \tau_k)}{\sin \pi} \in [-1, 1], \quad (k = 0, \ldots, N) \]

where 0 < x < 1 is the map parameter that determines the degree to which the nodes are shifted and the elements of the Chebyshev differentiation matrix are scaled. The conformal map improves the interpolation approximation of a smooth function by enlarging the function’s analytical region. And compared with other maps, the one-to-one and sufficiently smooth properties of conformal maps preserve the CPM’s spectral accuracy.

In addition, to compensate for the round-off error induced by the conformal map, the parameter \(x\) is preferred to be valued as:
\[ x = \text{sech}(\ln \varepsilon / N) \]
\[ \varepsilon > 0 \] is the numerical computation accuracy.

Since the mapper CGL nodes are shifted toward equidistant spacing, the Lagrange interpolation formulation in the standard CPM may induce the well-known Runge phenomenon, thus the barycentric rational interpolation technique is selected to approximate the flat output (Berrut and Trefethen, 2004):
\[ \tilde{z}(\tau) \approx \tilde{z}_N(\tau) = \sum_{k=0}^{N} o_k^\text{bary} \tilde{z}(\lambda_k) / \lambda - \lambda_k \]  
\[ o_0^\text{bary} = \frac{1}{2}, \quad o_k^\text{bary} = \frac{(-1)^N}{2}, \quad o_k^\text{bary} = (-1)^k, \quad (k = 1, \ldots, N - 1) \]  

Evaluating the derivatives of flat output \(\tilde{z}(\tau)\) at the mapped CGL nodes \(\lambda_k\) gives a matrix multiplication of the following form:
\[ \begin{align*}
\tilde{z}'(\lambda_k) &\approx \hat{\tilde{z}}_N'(\lambda_k) = \sum_{j=0}^{N} \mathcal{D}_{kj} \hat{\tilde{z}}(\lambda_j) \\
\tilde{z}''(\lambda_k) &\approx \hat{\tilde{z}}_N''(\lambda_k) = \sum_{j=0}^{N} \mathcal{D}^{(2)}_{kj} \hat{\tilde{z}}(\lambda_j)
\end{align*} \]  

where \(\mathcal{D}_{kj}\) are entries of the \((N + 1) \times (N + 1)\) first order mapped Chebyshev differentiation matrix, while \(\mathcal{D}^{(2)}_{kj}\) the second order. The matrices are given by:
\[ \begin{cases}
\mathcal{D}_{ij} = \text{diag}(o_j^\text{bary}) / \lambda_j - \lambda_j \\
\mathcal{D}^{(2)}_{ij} = -2 o_j^\text{bary} \sum_{k \neq j} \frac{o_k^\text{bary}}{\lambda_j - \lambda_k} - \frac{1}{\lambda_j - \lambda_j}
\end{cases} \]  

Recall that the original objective of introducing the conformal map and barycentric rational interpolation techniques is to improve the ill-condition of the standard Chebyshev differentiation matrix. Here both the distributions of the standard and mapped Chebyshev differentiation matrices for \(N = 32, x = 0.99\) are presented in Fig. 2. Compared with the standard differentiation matrix, the magnitude of the elements near the diagonal endpoints greatly decreases, validating the feasibility of our modification in improving the Chebyshev differentiation matrix’s ill-condition.

In the MCPM, the Bolza performance criterion in Eq. (5) can be reformulated as
\[ J = \Phi[\tilde{z}(t_0), \tilde{z}(t_f), t_0, t_f] + 0.5(t_f - t_0) \times \int_{-1}^{1} A(\tilde{z}(\lambda))g'(\tau)d\tau \]
Applying the Clenshaw-Curtis quadrature scheme to discretizing the integral part into a finite sum yields (Fahroo and Ross, 2002)

\[ J \approx J^N = \Phi[\tilde{\varphi}(-1), \tilde{\varphi}(1), t_0, t_f] + \sum_{k=0}^{N} \omega_{C}^k g'(t_k)A[\tilde{\varphi}(\lambda_k)] \]  

(30)

where \( g'(t) \) is the first order derivative of the conformal map in Eq. (23), and \( \omega_{C}^k \) ( \( k = 0, \ldots, N \) ) are Clenshaw-Curtis weights given by (Fahroo and Ross, 2002):

for \( N \) even,

\[
\begin{aligned}
\omega_{C}^0 &= \omega_{C}^N = \frac{1}{N^2} \\
\omega_{C}^s &= \frac{4}{N} \sum_{j=0}^{(N/2)^n} \frac{1}{1-4j^2} \cos \frac{2\mu j}{N}, \quad s = 1, \ldots, \frac{N}{2}
\end{aligned}
\]

(31)

for \( N \) odd,

\[
\begin{aligned}
\omega_{C}^0 &= \omega_{C}^N = \frac{1}{N^2} \\
\omega_{C}^s &= \frac{4}{N} \sum_{j=0}^{(N-1)/2} \frac{1}{1-4j^2} \cos \frac{2\mu j}{N}, s = 1, \ldots, \frac{N-1}{2}
\end{aligned}
\]

(32)

where the double prime in the summation operations indicates that the first and last elements have to be halved.

Till now, Problem B has been converted to the following mathematical programming problem.

**Problem B**. Determine the flat output values at the mapped CGL nodes \( [\tilde{\varphi}(\lambda_k)] \) \( (i = 1, 2, 3; k = 0, \ldots, N) \) and possibly the time \( t_f \) that minimize

\[ J^N = \Phi[\tilde{\varphi}(-1), \tilde{\varphi}(1), t_0, t_f] + \sum_{k=0}^{N} \omega_{C}^k g'(t_k)A[\tilde{\varphi}(\lambda_k)] \]  

(33)

and subject to

\[ F(\tilde{\varphi}(\lambda_k), t_0, t_f) \leq 0 \]  

(34)

where \( F(\cdot) \) represents the transformed constraints in the flat output space. In general, this is a nonlinear programming problem which can be solved by suitable algorithms or commercial packages. After obtaining the flat output solutions, the reorientation trajectory and the corresponding control inputs can be easily generated by mapping back into the original state space.

### 4. Trajectory tracking controller design

The objective of the closed-loop tracking controller is to eliminate deviations from the nominal trajectory, which are induced by external disturbances or model uncertainties. In this section, the trajectory tacking controller is developed based on the receding horizon control method. And a high-efficiency numerical method based on differential transformation is utilized to solve the resulted TPBVP in every moving horizon.

The linearized time-varying error dynamics about the nominal trajectory is derived as

\[ \Delta \dot{x}(t) = A(t) \Delta x(t) + B(t) \Delta u(t) \]  

(35)

where \( \Delta \dot{x}(t) \) and \( \Delta u(t) \) are the state deviation and feedback control law respectively. The non-zero elements of the Jacobian matrices \( A(t) = [A_{ij}] \in \mathbb{R}^{6 \times 6} \) and \( B(t) = [B_{ij}] \in \mathbb{R}^{6 \times 2} \) are given by:

\[ A_{12} = (I_2 - I_3)\omega_3/I_1, \quad A_{13} = (I_2 - I_3)\omega_2/I_1, \]
\[ A_{21} = (I_1 - I_3)\omega_3/I_2, \quad A_{22} = (I_3 - I_1)\omega_3/I_2, \]
\[ A_{31} = (I_1 - I_2)\omega_3/I_3, \quad A_{32} = (I_3 - I_1)\omega_3/I_3, \]
\[ A_{41} = 1, \quad A_{42} = \sin \phi \tan \theta, \quad A_{43} = \cos \phi \tan \theta, \]
\[ A_{44} = \tan \theta(\omega_2 \cos \phi - \omega_3 \sin \phi) \]
\[ A_{45} = \sec^2 \theta(\omega_2 \sin \phi + \omega_3 \cos \phi), \quad A_{52} = \cos \phi, \]
\[ A_{53} = - \sin \phi, \quad A_{54} = - \omega_2 \sin \phi - \omega_3 \cos \phi \]
\[ A_{61} = \sin \phi \sec \theta, \quad A_{62} = \cos \phi \sec \theta, \]
\[ A_{64} = \sec \theta(\omega_2 \cos \phi - \omega_3 \sin \phi) \]
\[ A_{65} = \sin \theta \sec^2 \theta(\omega_2 \sin \phi + \omega_3 \cos \phi), \quad B_{11} = 1/I_1, \]
\[ B_{22} = 1/I_2 \]

The values of these Jacobian matrices at any given time \( t \geq t_0 \) are dependent on the state and control histories of the nominal trajectory. The receding horizon control problem at time \( t \) is stated as an optimal control problem with the following performance criterion:
\[ S = \frac{1}{2} \Delta x^T(t_{\text{end}}) P_s \Delta x^T(t_{\text{end}}) + \frac{1}{2} \int_{t_{\text{end}}}^{t_\ast} |\Delta x^T(\tau) Q \Delta x^T(\tau) + \Delta u^T(\tau) R \Delta u^T(\tau)| \, d\tau \]  

(36)

where \( t_{\text{end}} = t + T \). Note that \( \delta_c \leq T < \infty \) and \( \delta_c \) is a positive constant used in the definition of uniform controllability of the aforementioned error dynamics (Lu, 1999). The matrices \( P_s, Q \in \mathbb{R}^{6 \times 6} \) are symmetric positive semi-definite matrices, and \( R \in \mathbb{R}^{2 \times 2} \) symmetric positive define. The main idea of receding horizon control is to obtain the optimal control \( u_{\text{opt}}(t) \) for the finite moving horizon \([t, t_{\text{end}}]\) with the current state \( \Delta x(t) \) as the initial condition. Note that only the first data \( \Delta u_{\text{opt}}(t) \) is selected as the applied feedback input for the current control cycle, while the rest of \( \Delta u_{\text{opt}}(t) \) is discarded. At the beginning of the next control cycle, the solution process is repeated, and the control input is recomputed.

Based on the Pontryagin’s principle, the optimal control problem is transformed into the following TPBVP:

\[
\begin{bmatrix}
\dot{A}x(t) \\
\dot{\Delta}l(t)
\end{bmatrix} =
\begin{bmatrix}
A & -BR^{-1}B^T \\
-Q & -A^T
\end{bmatrix}
\begin{bmatrix}
x(t) \\
l(t)
\end{bmatrix}
\]

\(x(t_0) = \Delta x_0, \Delta l(t_{\text{end}}) = P_s \Delta x(t_{\text{end}})\)

And the optimal control law is given by

\[
\Delta u(t) = -R^{-1}B^T \Delta l(t)
\]

(38)

where \( \Delta l(t) \) is the costate vector.

In this paper, the differential transformation method is utilized to solve the TPBVP, since it overcomes the potential instability and time-consuming computation issues of the conventional methods. Before discussing the solution process, a brief review of differential transformation is simply presented first.

4.1. Differential transformation method

As a basis-function based numerical method for optimal control, the differential transformation method converts the differential equations from the original time and/or space domain into a series of algebraic equations in a transformed domain; and the solution, computed by solving the resulted algebraic equations, is then mapped back to the original domain in the form of a finite-term Taylor series using the inverse differential transformation (Hwang et al., 2009).

The images of a smooth function \( x(t), t \in (a, b) \) at the expansion point \( t_e \in (a, b) \) in the transformed domain are defined as:

\[
X_{i_e}(i) = \frac{H^i}{i!} \left( \frac{\partial x(t)}{\partial t} \right)_{t_e}
\]

(39)

where \( X_{i_e}(i) \) is the \( i \)th order differential spectrum of \( x(t) \) at \( t_e \), and \( H > 0 \) the spectrum scaling factor.

Selecting \( \{ t_e \}^\infty_{e=0} \) as the basis system, the inverse differential transformation of \( \{ X_{i_e}(i) \}^\infty_{i=0} \) is written as

\[
\Xi_i(t) = \sum_{i=0}^{\infty} X_{i_e}(i) \left( \frac{t-t_e}{H} \right)^i
\]

(40)

Substituting Eq. (39) into Eq. (40) yields

\[
\Xi_i(t) = \sum_{i=0}^{N} \left( \frac{\partial^i x}{\partial t^i} \right)_{t_e} \left( \frac{t-t_e}{H} \right)^i + R_N
\]

(41)

\[
R_N = \left( \frac{\partial^{N+1} x}{\partial t^{N+1}} \right)_{t_e} \left( \frac{t-t_e}{H} \right)^{N+1} \frac{(N+1)!}{(N+1)!}
\]

(42)

where \( R_N \) denotes the remainder for Taylor’s theorem, and \( \xi \) is a point in the interval \([t, t_e]\). In general, the inverse differential transformation \( \Xi_i(t) \) would convergence to \( x(t) \) for any \( t \in (a, b) \) when the function \( x(t) \) is smooth.

Three steps should be exploited to apply the differential transformation method to solving the general differential equation \( x(t) = f(x(t)) \). First, convert the differential equation into a set of recursive difference equations of \( X_{i_e}(i) \) at the expansion point \( t_e \) using the differential transformation. Second, given the initial values of \( X_{i_e}(0) \), compute recursively the values of \( X_{i_e}(i) \), \( i = 1, \ldots, N \). Note that the initial values in our research are developed based on the boundary and trans-versatility conditions of TPBVP. Third, give the analytical solution to the smooth function \( x(t) \) as an \( N \)-term Taylor expansion polynomial using the inverse differential transformation. It is worth noting that the differential spectrum is critical for the differential transformation method, thus some basic operations of differential spectrum that are utilized in our research are presented in Table 1 where \( c \in \mathbb{R} \) is a constant.

4.2. Apply to TPBVP solution

For the linear system in Eq. (37), its solution could be written as:

\[
\begin{bmatrix}
\Delta x(t) \\
\Delta l(t)
\end{bmatrix} =
\begin{bmatrix}
F(t, t_{\text{end}}) & G(t, t_{\text{end}}) \\
L(t, t_{\text{end}}) & M(t, t_{\text{end}})
\end{bmatrix}
\begin{bmatrix}
\Delta x(t_{\text{end}}) \\
\Delta l(t_{\text{end}})
\end{bmatrix}
\]

(43)

where the \( 6 \times 6 \) matrices \( F, G, L, M \) satisfy:

\[
F(t_{\text{end}}, t_{\text{end}}) = I, G(t_{\text{end}}, t_{\text{end}}) = 0, L(t_{\text{end}}, t_{\text{end}}) = 0, M(t_{\text{end}}, t_{\text{end}}) = I
\]

(44)

Substituting the trans-versatility condition \( \Delta l(t_{\text{end}}) = P_s \Delta x(t_{\text{end}}) \) into Eq. (43) yields:

\[
\begin{cases}
\Delta x(t) = (F + GP_s) \Delta x(t_{\text{end}}) \\
\Delta l(t) = (L + MP_s) \Delta x(t_{\text{end}})
\end{cases}
\]

(45)

then \( \Delta l(t) \) could be written as:

\[
\Delta l(t) = (L + MP_s)(F + GP_s)^{-1} \Delta x(t)
\]

(46)

Table 1

<table>
<thead>
<tr>
<th>Basic operations for differential transformation.</th>
<th>Differential Spectrum ( i = 0, 1, \ldots, N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x(t) \pm y(t) )</td>
<td>( X_{i_e}(i) \pm Y_{i_e}(i) )</td>
</tr>
<tr>
<td>( c \cdot x(t) )</td>
<td>( c \cdot X_{i_e}(i) )</td>
</tr>
<tr>
<td>( dx(t)/dt )</td>
<td>( (i+1)X_{i_e}(i+1)/H )</td>
</tr>
</tbody>
</table>
Let the variables $V(t)$ and $W(t)$ be

$$
\begin{align*}
V(t) &= F(t, t_{\text{end}}) + G(t, t_{\text{end}})P_f \\
W(t) &= L(t, t_{\text{end}}) + M(t, t_{\text{end}})P_f
\end{align*}
$$

(47)

Substituting Eqs. (45) and (47) into the system Eq. (37) gives:

$$
\begin{align*}
\dot{V}(t) &= AV(t) - BR^{-1}B^TW(t) \\
\dot{W}(t) &= -QV(t) - A^TW(t)
\end{align*}
$$

(48)

with the boundary conditions written as

$$
V(t_{\text{end}}) = I, W(t_{\text{end}}) = P_f
$$

(49)

Selecting the time $t_{\text{end}}$ as the expansion point and applying the differential transformation to Eqs. (48) and (49), we obtain a system of recursive equations as follows:

$$
\begin{align*}
(k + 1)\dot{V}_{\text{tend}}(k + 1) &= AV_{\text{tend}}(k) - BR^{-1}B^T\dot{W}_{\text{tend}}(k) \\
(k + 1)\dot{W}_{\text{tend}}(k + 1) &= -QV_{\text{tend}}(k) - A^T\dot{W}_{\text{tend}}(k)
\end{align*}
$$

(50)

$$
\dot{V}_{\text{tend}}(0) = I, \dot{W}_{\text{tend}}(0) = P_f
$$

(51)

Consequently, the elements of the matrices $\dot{V}_{\text{tend}}(i)$, $\dot{W}_{\text{tend}}(i)$, $(i = 0, 1, \ldots N)$ can be computed recursively by substituting the initial values into Eq. (50). Then the analytical solution to $V(t)$, $W(t)$ can be formulated by the following N-term Taylor series:

$$
\begin{align*}
V(t) &= \sum_{k=0}^{N} \dot{V}_{\text{tend}}(k)(t - t_{\text{end}})^k \\
W(t) &= \sum_{k=0}^{N} \dot{W}_{\text{tend}}(k)(t - t_{\text{end}})^k
\end{align*}
$$

(52)

Substituting Eq. (52) into Eq. (46), and the resulting one into Eq. (38), the optimal feedback control law is given as:

$$
\Delta u(i) = -R^{-1}B^TW(i)V^{-1}(i)\Delta x(i)
$$

(53)

If the system states at current time $t$ are obtained, the optimal feedback control law $\Delta u(i)$ can then be computed. The actual trajectory is then steered by $u(t) = u(i(t) + \Delta u(i)$ for the current control cycle with the nonlinear governing dynamics in Eqs. (2) and (3). At the next cycle, the state derivation and the feedback control input are generated in the same way. The process iterates until the trajectory satisfies the terminal state requirements.

5. Numerical simulations

To evaluate the performance of the proposed reorientation trajectory optimization and tracking approaches, the numerical simulation for a generic asymmetric underactuated spacecraft with an inertia matrix of $I = [66.36, 61.80, 50.16] \text{kg m}^2$ is carried out in this section. The upper limit of the control inputs is set as 1 N m. Note that for the 3-2-1 attitude sequence, computational singularity occurs when the pitch angle is $\theta = \pm \pi/2$. Therefore in the simulations, the initial and desired terminal angular velocities are assumed zero, and the Euler angles are set to be $[-\pi/2, -\pi/3, \pi/2]^T$ and $[\pi/2, \pi/3, -\pi/2]^T$, respectively.

In the MPCM, the order of the Chebyshev polynomial $N$ and the numerical computation accuracy $\varepsilon$ are set to be $32$ and $1.0 \times 10^{-16}$ respectively. Thus the conformal mapping parameter $\varepsilon$ is valued as $0.574960$. The trajectory optimization is required to minimize the maneuver time and control effort simultaneously. The problem is converted into a single-objective optimization problem by applying the aforementioned physical programming method. The ranges of desirability for each performance criterion are presented in Table 2, where the parameters $J_{i1} - J_{i3}$ are physically meaningful values of $J_i (i = 1, 2)$ that are prescribed by the designer to quantify the preferences associated within the criterion.

The numerical simulation is conducted in the MATLAB environment, and the fmincon function from the Optimization Toolbox is utilized to solve the transformed Problem BN with the parameters ‘TolX’, ‘TolFun’ and ‘TolCon’ set as $1.0 \times 10^{-8}$, $1.0 \times 10^{-8}$ and $1.0 \times 10^{-12}$, respectively. The time histories of system states and control inputs are presented in Figs. 3–5, where the circles denote the corresponding variables at the mapped CGL nodes. The reorientation trajectory is smooth, and the control inputs observe the saturation constraints. As can be seen in Fig. 5, the maximum magnitude of pseudo input at the mapped CGL nodes is much smaller than $1.0 \times 10^{-12}$, including the rationality of the solution to an underactuated systems.

To evaluate the results of our research, Radau pseudospectral method (RPM) is also applied to this problem by utilizing a modified version of the open source GPOPS package (Rao et al., 2010). The field of ‘autoscale’ in GPOPS is set ‘on’ to invoke the automatic scaling routine, while the ‘mesh refinement’ option is set by default to accurately distribute the collocation nodes. The RPM trajectories are also illustrated in Figs. 3–5 by dashed lines for comparison. Moreover, Table 3 presents the values of the aggregate objective functions in (8) for each method, as well as the associated maneuver time and control effort. Some slight differences exist between the trajectories, and the aggregate objective function of RPM is a litter better than the flatness solution. The differences are mainly induced by three reasons. First, different optimization engines are used for each approach, the commercial package SNOPT (Gill et al., 2005) for GPOPS, while the fmincon function for our approach. Second, only the flat outputs are optimized in our approach. In contrast, both of the state and input vectors are parameterized and optimized simultaneously in RPM. Third, the collocation

<table>
<thead>
<tr>
<th>Performance criterion</th>
<th>Design metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J_1 (s)$</td>
<td>80 100 130 160 200</td>
</tr>
<tr>
<td>$J_2 (N^2 m^2 s)$</td>
<td>0.7 0.9 1.3 1.6 1.9</td>
</tr>
</tbody>
</table>
nodes are refined in GPOPS, while unchanged for the flat outputs optimization. Thus the differences between the flat output trajectories in Fig. 4 mainly exist in the middle of the time interval where the MCGL nodes are more sparsely spaced.

In addition, the nonlinear reorientation dynamics are propagated using the MATLAB ode45 integrator with the control inputs of our approach. The profiles of the trajectory error with respect to the flatness solution are presented in Figs. 6 and 7. Obviously the propagation errors

<table>
<thead>
<tr>
<th>Methods</th>
<th>Aggregate objective</th>
<th>Time (s)</th>
<th>Control effort (N² m² s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPM</td>
<td>1.232509</td>
<td>137.671123</td>
<td>1.292686</td>
</tr>
<tr>
<td>Flatness</td>
<td>1.248954</td>
<td>137.728169</td>
<td>1.314972</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dispersions</th>
<th>$\Delta \theta_1$</th>
<th>$\Delta \theta_2$</th>
<th>$\Delta \theta_3$</th>
<th>$\Delta \phi$</th>
<th>$\Delta \Theta$</th>
<th>$\Delta \psi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3\sigma$</td>
<td>0.01 rad/s</td>
<td>0.01 rad/s</td>
<td>0.0001 rad/s</td>
<td>$3^\circ$</td>
<td>$2^\circ$</td>
<td>$2^\circ$</td>
</tr>
</tbody>
</table>

Table 3
Values of performance criteria for each method.

Table 4
$3\sigma$ Standard Deviations of Dispersions.
for angular velocity and Euler angles are comparatively small, validating the feasibility of the flatness solution.

The performance of the proposed closed-loop tracking controller is accessed by considering the dispersions presented in Table 4.

In the receding horizon control method, the LQR matrices are valued as

\[ Q = \text{diag}([2 \ 5 \ 6 \ 5 \ 6 \ 8]) \times 10^3, \quad R = \text{diag}([3.5]), \quad P_f = 0 \]

The distribution of the terminal state dispersions for a 200-run Monte Carlo simulation is presented in Fig. 8. Clearly the closed-loop tracking controller can steer the terminal state dispersions to a small interval, which is very close to zero. The real control inputs which satisfy the limitations are illustrated in Fig. 9. Thus based on the Monte Carlo results, we can see that the proposed control scheme is effective and robust to enable the reorientation maneuver with constraints observed.

Fig. 8. Monte Carlo simulation results for terminal state dispersion.

Fig. 9. Time histories of control inputs for Monte Carlo Simulation.
6. Conclusions

In this paper, an integrated reorientation control algorithm, including nominal trajectory optimization and tracking methods suitable for on-line implementation, is developed for a general asymmetric underactuated spacecraft. Some useful conclusions are drawn as follows. First, the trajectory optimization problem of a differential flat system can be formulated as an integration-free geometric programming problem that is more computationally tractable. Second, the CPM applies easily to flat output discretization, and the numerical accuracy can be improved via the conformal mapping and barycentric rational interpolation techniques. Third, the differential transformation method for the receding horizon control problem has an advantage of computational efficiency over the conventional methods.

This paper is mainly concerned with the two-input underactuated spacecraft, but the approaches studied are also applicable to the full-actuated cases and other underactuated systems. Note that the Euler angles described kinematics may induce singularity issues when applying the suggested method for arbitrary reorientation missions. Considering that the quaternions and modified Rodrigues parameters have eliminated the mentioned singularity issues, we would investigate the feasibility of incorporating the associated attitude dynamics with the numerical framework suggested here in the future. In addition, although the pseudospectral method is known to be numerically convenient, yet the constraints are only guaranteed at the collocation nodes. Thus further studies may focus on developing a trajectory planning methodology satisfying the constraints continuously in time.

References


