


RESEARCH ARTICLE

Neighboring extremal nonlinear model predictive control of a rigid body on $SO(3)$

Shiva Bagherzadeh¹, Hossein Karimpour^{2,*}  and Mehdi Keshmiri¹

¹Department of Mechanical Engineering, Isfahan University of Technology, Isfahan, Iran and ²Department of Mechanical Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran

*Corresponding author. E-mail: h.karimpour@eng.ui.ac.ir

Received: 9 April 2022; **Revised:** 20 November 2022; **Accepted:** 22 November 2022; **First published online:** 6 January 2023

Keywords: nonlinear model predictive control, neighboring extremal, Lie group variational integrators, sensitivity derivatives, computation time

Abstract

The issue of implementing nonlinear model predictive control (NMPC) on mechanical systems evolving on special orthogonal group ($SO(3)$) is taken into consideration in the first place. Necessary conditions of optimality are extracted based on Lie group variational integrators, leading to a two-point boundary value problem (TPBVP) which is solved using sensitivity derivatives and indirect shooting methods. Fast Newton-like methods referred to as fast solvers which are commonly used to solve the TPBVP are established based on the repetition of a nonlinear process. The numerical schemes employed to alleviate the computation burden consist of eliminating some constraint-related but non-essential terms in the trend of sensitivity derivatives calculation and for solving the TPBVP equations. As another claim, assuming that a first attempt to resolve the NMPC problem is accessible, the problem subjected to some changes in its initial conditions (due to some re-planning schemes) can be resolved cost-effectively based on it. Instead of solving the whole optimization process from the scratch, the optimal control inputs and states of the system are updated based on the neighboring extremal (NE) method. For this purpose, two approaches are considered: applying NE method on the first solution that leads to a neighboring optimal solution, or assisting this latter by updating the NMPC-related optimization using exact TPBVP equations at some predefined intermediate steps. It is shown through an example that the first method is not accurate enough due to error accumulations. In contrast, the second method preserves the accuracy while reducing the computation time significantly.

1. Introduction

Nonlinear model predictive control (NMPC) of mechanical systems has been widely studied in the literature, including cases which evolve on Riemannian manifolds such as $SO(3)$ and $SE(3)$ [1–6]. Although hardware performance in terms of operational speed has reached a relative bounce nowadays, nonetheless the successive process of optimization required by this particular method is still very time-consuming. Therefore, alleviating the computation burden while preserving the accuracy of the results is a preliminary solution to the problem of real-time implementation. Linear matrix inequality (LMI) method is used to solve the optimizations related to the constrained model predictive control (MPC) control of a 3D pendulum on $SO(3)$ in ref. [7]. It is shown that using LMI for solving MPC problem on manifold needs less control efforts than standard MPC. However, the difference is not noticeable in terms of computation burden and time. In ref. [6], direct shooting method has been used to solve the optimization problem of implementing NMPC to flight control of a quadrotor on $SE(3)$. Direct methods are less accurate than indirect shooting methods and they may terminate with non-optimal solutions [8]. Generally, using fast solvers based on indirect shooting and Newton-like methods can increase the accuracy and speed of computations for optimal control problems. Another aspect to consider toward enhancing computational efficiency consists of deriving an appropriate discretized set of equations. One conventional approach is to apply the variational principle and then to discretize the resulting equations.

In contrast, another method known as the discretized Hamilton's principle consists of applying the variational principle to the discretized Lagrangian of the system. This leads to another set of equations for which the numerical integrators are termed variational integrators [9]. Lie group variational integrators (LGVI) which exploit the main aspects of Lie group methods are much more accurate than classical methods of integration such as Runge–Kutta for the same step size [10], mainly because LGVI are symplectic and momentum-preserving systematically and show good energy behavior [11, 12]. Furthermore, the discretization of equations and extraction of necessary conditions of optimality based on LGVI can preserve the geometric structure of the system which is of primary importance in tracking the evolution of the system configuration [13]. In terms of the optimal control problem on Lie group manifolds, the first-order discrete conditions for optimality are derived based on discrete equations of motion [14]. An exact and fast solver is then used in order to solve the corresponding two-point boundary value problem (TPBVP) based on sensitivity derivatives. This latter determines the sensitivity of the specified terminal boundary conditions to the change in the (unspecified) initial conditions of the Lagrange multipliers [14]. Forasmuch as the sensitivity derivatives are represented in the context of Lie algebra, singularities and other ambiguities that may manifest themselves in other contexts such as Euler angles and quaternions are avoided during the group actions; thus, sensitivity derivatives are able to provide a geometrically exact and efficient method for solving the TPBVPs [15]. The method expressed above is used in refs. [14, 16] for the optimal control of rigid bodies evolving on Lie groups $SO(3)$ and $SE(3)$, respectively. The method developed in ref. [14] is adapted in ref. [17] for discrete-time MPC of mechanical systems evolving on $SO(3)$, where using efficient fast solver is revealed to reduce the amount of optimization time in comparison with baseline solvers. However, since direct and indirect shooting methods are still considered as time-consuming procedures, it is necessary to find methods to reduce the computation time, especially when it comes to online implementation [8]. It is shown in ref. [18] that one can gain a considerable reduction in computational time in comparison with the procedure presented in refs. [14, 17], only by applying some subtle simplifications on the TPBVP equations and employing some techniques for calculating the sensitivity derivatives. The procedure of substituting non-essential nonlinear equations by simplified linear equations in the trend of solving the optimal control problem is being considered in the present work for mechanical systems evolving on $SO(3)$. Indeed, the terms which are considered non-essential for the solver process are nonetheless added to the system of equations during the last iteration, permitting to omit them in other iterations. These are some of the aspects that will be discussed in this work in the first place.

In addition, optimal control problems may face changes in their initial conditions or parameters, due to some reasons. In such cases, it is not necessary to perform the optimal control process from the scratch if an initial answer to the problem (corresponding to the first set of initial conditions) is already available. Using neighboring extremal (NE) methods can help to obtain the solution for the new set of initial conditions based on estimating the corresponding variations to the optimal control inputs [19, 20]. This method has been considered on Riemannian manifolds such as $SO(3)$ in a limited number of ref. [21]. Juxtaposing the NE method with NMPC could enhance the efficiency of the control system significantly. However, this issue has not been widely addressed in related sources, unless only on Euclidean spaces [22, 23]. In some cases, NE methods lead to a two-layer formulation [24, 25]. In the present work, for a system evolving on a Lie group $SO(3)$, the combination of NE method and NMPC is considered to estimate the optimal solution in the presence of some alteration in the initial conditions.

Therefore, two main goals are pursued in this article. First, it is shown that the potential of the present method to reduce the computation time is much more noticeable compared to what is offered in ref. [17], which is one of the great achievements of this work that provides a practical way toward real-time implementations of NMPC on $SO(3)$. In this manner, discrete-time NMPC is applied to the equations of a rigid body evolving on the Lie group $SO(3)$. Afterward, it is demonstrated that by reducing the nonlinearity of TPBVP equations and sensitivity derivatives, the NMPC-related calculations are lessened considerably compared with previous works. Secondly, NE method is applied to the governing equations in order to alter their optimal solution in case of change in the initial conditions, without having to perform the whole control planning procedure from the scratch. This method is highly valued

whenever some solution is already available, or if some inner delays occur in NMPC systems, or in case the initial conditions for a time step changed unpredictably. In such cases, the computations required for obtaining the updated solution based on the basic initial conditions can be performed using NE method in a much shorter time. The NE method can apply to such systems lonely or in combination with the NMPC process. It is shown that the NE method is not able to preserve the accuracy of the responses for long periods. In contrast, updating the NE method with NMPC computed at some predetermined steps through the main TPBVP equations can increase the efficiency of the method while reducing the computation time. The methodologies employed for aforementioned objectives are implemented on an example and the simulation results confirm their validity.

Accordingly, this article is organized as follows: The equations of motion of a rigid body on SO(3) are presented in Section 2. NMPC formulation and the proposed simplifications of TPBVP and sensitivity derivative equations are expressed in Section 3. The applications of NMPC with and without applying the proposed simplifications of Section 3 are compared in terms of accuracy and time via an example in Section 4. Section 5 is dedicated to introduce the NE method and its reformulation for the NMPC control of mechanical systems on Lie group SO(3). The results of Section 5 are implemented to an example in Section 6. Finally, Section 7 is the conclusion part.

2. Equations of a rigid body on SO(3)

The Lagrangian form of continuous equations of motion of a rigid body in the absence of potential, on Lie group SO(3), is obtained using Hamilton’s principle as follows [26]:

$$J\dot{\Omega} + \Omega \times J\Omega = Bu \tag{1}$$

$$\dot{R} = RS(\Omega) \tag{2}$$

By applying the Legendre transformation to the above equations, one can express the equations of motion in the Hamiltonian form as [27]:

$$\dot{\Pi} + \Omega \times \Pi = Bu \tag{3}$$

$$\dot{R} = RS(\Omega) \tag{4}$$

where $\Pi = J\Omega$ which belongs to \mathbb{R}^3 is the angular momentum of the body. The vector $\Omega \in \mathbb{R}^3$ is the angular velocity of the body expressed in the body-fixed frame. The matrix $J \in \mathbb{R}^{3 \times 3}$ is the standard moment of inertia matrix which is related to the nonstandard moment of inertia matrix, J_d , by $J_d = \frac{1}{2} \text{tr}(J) I_3 - J$. The vector of control inputs is denoted by $u \in \mathbb{R}^m$, where $B \in \mathbb{R}^{3 \times m}$ is its coefficient matrix. The Lie group $SO(3) = \{R \in \mathbb{R}^{3 \times 3} | RR^T = I_3, \det(R) = 1\}$ is the special orthogonal group where its Lie algebra $\mathfrak{so}(3)$ is the set of all skew-symmetric matrices on $\mathbb{R}^{3 \times 3}$. $S(\cdot) : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$ is an isomorphism, that is, a one-to-one invertible map between the vectors of \mathbb{R}^3 and the matrices of $\mathfrak{so}(3)$ and it is defined as follows: for all $x, y \in \mathbb{R}^3$, $S(x)y = x \times y$ [28, 29]. In order to extract the discrete format of the rigid body equations of motion on SO(3), referred to as LGVI, the discrete version of the Hamilton’s principle is used instead of discretizing the continuous equations of motion [14]. These two methods are not equivalent generally [30]. The discrete equations of motion extracted using variational approach based on discrete Hamilton’s principle are systematically symplectic and momentum preserving and show good energy behavior in long-time integration [15]. The discrete equations of motion in the Lagrangian form extracted using the aforementioned method are presented below [26]:

$$\frac{1}{h} (F_{k+1} J_d - J_d F_k - J_d F_{k+1}^T + F_k^T J_d) = hS(Bu_k) \tag{5}$$

$$R_{k+1} = R_k F_k \tag{6}$$

where the matrix F_k is an auxiliary orthogonal matrix [14]. Using the discrete format of Legendre transformation, the discrete-time format of equations of motion, referred to as LGVI, is given by:

$$hS(\Pi_k) = F_k J_d - J_d F_k^T \tag{7}$$

$$R_{k+1} = R_k F_k \tag{8}$$

$$\Pi_{k+1} = F_k^T \Pi_k + h B u_k \tag{9}$$

where parameter h is the time step of the problem. Also, since $F \in \text{SO}(3)$ and the group operation on $\text{SO}(3)$ is matrix multiplication, Eq. (8) ensures that the computed rotation matrix will always remain on $\text{SO}(3)$, that is, the geometric structure is preserved without the need to use any constraints or projections [13].

3. NMPC formulation

NMPC is a finite-horizon, iterative optimal control method. Therefore, the NMPC formulation for a system evolving on a Lie group is accomplished based on the principles of optimal control problem on Lie groups. The first step toward reaching that goal is to extract the LGVI [13]. In order to formulate the NMPC on $\text{SO}(3)$, the same method as what is employed in ref. [18] is pursued. The following general discrete-time cost functional for a finite control horizon based on minimizing error and energy is considered:

$$\min \tilde{J}_d(u) = \Phi_d(R_N, S(\Pi_N)) + \sum_{k=0}^{N-1} \mathcal{L}_d(R_k, S(\Pi_k), S(u_k)) \tag{10}$$

which is subjected to Eqs. (7), (8), and (9), given R_0 and Π_0 , and the inequality constraints of the form:

$$C_d(R_k, S(\Pi_k), S(u_k)) \leq 0 \tag{11}$$

where $\Phi_d : \text{SO}(3) \times \mathfrak{so}(3) \rightarrow \mathbb{R}_{\geq 0}$, $\mathcal{L}_d : \text{SO}(3) \times \mathfrak{so}(3) \times \mathfrak{so}(3) \rightarrow \mathbb{R}_+$ and $C_d : \text{SO}(3) \times \mathfrak{so}(3) \times \mathfrak{so}(3) \rightarrow \mathbb{R}$ are twice differentiable functions with respect to their arguments. The augmented cost functional to implicate the constraints is written as follows:

$$\begin{aligned} \bar{\tilde{J}}_d(u) = & \Phi_d(R_N, S(\Pi_N)) + \sum_{k=0}^{N-1} \left[\mathcal{L}_d(R_k, S(\Pi_k), S(u_k)) + \ll (\lambda_k^1)^\diamond, \log(F_k - R_k^T R_{k+1}) \gg \right. \\ & \left. + \ll S(\lambda_k^2), (F_k^T \Pi_k + h B u_k - \Pi_{k+1})^\diamond \gg \right] + \mu^T \psi_d(C_d(R_k, S(\Pi_k), S(u_k))) \end{aligned} \tag{12}$$

where $\lambda_k^1, \lambda_k^2 \in \mathbb{R}^3$ and $\mu \in \mathbb{R}^l$ are the Lagrange multipliers and $\psi_d : \mathbb{R} \rightarrow \mathbb{R}^l$ is the penalty function that incorporates inequality constraints into the cost functional as soft constraints [31]. Thus, Eq. (12) is the augmented form of Eq. (10) obtained by imposing the LGVI as dynamical constraints and the system state constraints to the cost functional using Lagrange multipliers. The $(\diamond) : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)^*$ represents an isomorphism defined as: $x^\diamond = \frac{1}{2} S(x)$ for any $x \in \mathbb{R}^3$, and $\mathfrak{so}(3)^*$ indicates the dual space of $\mathfrak{so}(3)$ [32]. As well, $\ll \cdot, \cdot \gg : \mathfrak{so}(3) \times \mathfrak{so}(3)^* \rightarrow \mathbb{R}$ denotes the natural pairing between the elements of $\mathfrak{so}(3)$ and $\mathfrak{so}(3)^*$ [28].

3.1. Extracting the necessary conditions for optimality

According to the above relations and the scheme presented in ref. [18], the first variation of the augmented cost functional is computed. The infinitesimal variations of R_k and F_k are defined through $\delta R_k = R_k S(\eta_k)$ and $\delta F_k = F_k S(\zeta_k)$ where $\eta_k, \zeta_k \in \mathbb{R}^3$. Consequently, the following relations can be obtained [14]:

$$\eta_{k+1} = F_k^T \eta_k + \zeta_k \tag{13}$$

$$\zeta_k = M_k \delta \Pi_k \tag{14}$$

where $M_k = hF_k^T[tr(F_k J_d)I_3 - F_k J_d]^{-1} \in \mathbb{R}^{3 \times 3}$. In addition, without loss of generality, the functions \mathcal{L}_d , Φ_d , C_d , and $\psi_d(C_d)$ are considered to be in the following formats:

$$\mathcal{L}_d = \frac{h}{2} \left\| P_1^{\frac{1}{2}}(R_k - I_3) \right\|_F^2 + \frac{h}{2} \left\| P_2^{\frac{1}{2}}S(\Pi_k) \right\|_F^2 + \frac{h}{2} \left\| P_3^{\frac{1}{2}}S(u_k) \right\|_F^2 \tag{15}$$

$$\Phi_d = \frac{1}{2} \left\| Q_1^{\frac{1}{2}}(R_N - I_3) \right\|_F^2 + \frac{1}{2} \left\| Q_2^{\frac{1}{2}}S(\Pi_N) \right\|_F^2 \tag{16}$$

$$C_d = \frac{1}{2} \|S(u_k)\|_F^2 - \alpha \tag{17}$$

$$\psi_d(C_d) = h \max\{0, C_d\}^2 \tag{18}$$

where P_1, P_2, P_3, Q_1 , and Q_2 are positive definite coefficient matrices. According to [33], the quadratic format of the above functionals can suffice to undertake the stability of the control system. However, since the matter of stability analysis for NMPC on Euclidean space and Riemannian manifolds has already been studied in the literature, it is out of the scope of this paper. The authors propose [33–35] for more information about the stability conditions. It should be mentioned that the imposed constraints, C_d , prevent saturation of the actuators. According to the above equations, the necessary conditions for optimality are extracted as below:

$$hS(\Pi_k) = F_k J_d - J_d F_k^T \tag{19}$$

$$R_{k+1} = R_k F_k \tag{20}$$

$$\Pi_{k+1} = F_k^T \Pi_k + hB u_k \tag{21}$$

$$\lambda_{N-1}^1 = -S^{-1}((R_N^T Q_1)_A) \tag{22}$$

$$\lambda_{k+1}^1 = F_{k+1}^T [\lambda_k^1 + hS^{-1}((R_{k+1}^T P_1)_A)] \tag{23}$$

$$\lambda_{N-1}^2 = S^{-1}((Q_2 S(\Pi_N))_A) \tag{24}$$

$$\lambda_{k+1}^2 = (F_{k+1} - M_{k+1}^T S(F_{k+1}^T \Pi_{k+1}))^{-1} [-M_{k+1}^T \lambda_{k+1}^1 + \lambda_k^2 - hS^{-1}((P_2 S(\Pi_{k+1}))_A)] \tag{25}$$

$$hB^T \lambda_k^2 = -hS^{-1}((P_3 S(u_k))_A) - \mu S^{-1}((D_{S(u_k)} \psi_d(C_d))_A) \tag{26}$$

where $(C)_A$ represents the antisymmetric part and $\|C\|_F$ represents the Frobenius norm for every matrix $C \in \mathbb{R}^{3 \times 3}$.

The above equations express a TPBVP that can be solved using the initial conditions of R_k, Π_k, λ_k^1 , and λ_k^2 . Given Π_0, F_0 is computed using Eq. (19). R_0 and F_0 permit to calculate R_1 through Eq. (20). By means of the initial values λ_0^1 and λ_0^2, u_0 will be computed using Eq. (26). Then λ_0^1 and λ_0^2 are computed via Eqs. (23) and (25). This process is repeated until $k = N - 2$. Since the initial values for the Lagrange multipliers are not available, an indirect shooting method should be used to estimate them. Accordingly, similar to ref. [18], the first variation of Eqs. (19) to (26) is derived to extract the sensitivity derivatives:

$$\eta_{k+1} = F_k^T \eta_k + M_k \delta \Pi_k \tag{27}$$

$$\delta \Pi_{k+1} = [F_k^T + S(F_k^T \Pi_k) M_k] \delta \Pi_k + hB \delta u_k \tag{28}$$

$$\delta \lambda_{N-1}^1 = S^{-1}((S(\eta_N) R_N^T Q_1)_A) \tag{29}$$

$$\delta\lambda_{k+1}^1 = (F_{k+1}S(M_{k+1}\delta\Pi_{k+1}))^T[\lambda_k^1 + hS^{-1}((R_{k+1}^T P_1)_A)] + F_{k+1}^T[\delta\lambda_k^1 + hS^{-1}(((R_{k+1}S(\eta_{k+1}))^T P_1)_A)] \tag{30}$$

$$\delta\lambda_{N-1}^2 = S^{-1}((Q_2S(\delta\Pi_N)_A)) \tag{31}$$

$$\delta\lambda_{k+1}^2 = (F_{k+1} - M_{k+1}^T S(F_{k+1}^T \Pi_{k+1}))^{-1} [(-\delta M_{k+1}^T \lambda_{k+1}^1 - M_{k+1}^T \delta\lambda_{k+1}^1 + \delta\lambda_k^2 - hS^{-1}((P_2S(\delta\Pi_{k+1}))_A)) - (\delta F_{k+1} - \delta M_{k+1}^T S(F_{k+1}^T \Pi_{k+1}) - M_{k+1}^T S(F_{k+1}^T \Pi_{k+1})) \lambda_{k+1}^2] \tag{32}$$

$$B^T \delta\lambda_k^2 = -S^{-1}((P_3S(\delta u_k) + 2\mu C_d S(\delta u_k) + 2\mu S(u_k)S(\delta u_k)S(u_k))_A) \tag{33}$$

where

$$\delta M_k = [h(F_k S(M_k \delta \Pi_k))^T - M_k(\text{tr}(F_k S(M_k \delta \Pi_k) J_d) I_3 - F_k S(M_k \delta \Pi_k) J_d)] (\text{tr}(F_k J_d) I_3 - F_k J_d)^{-1} \tag{34}$$

These equations can be rewritten in the following form considering Eqs. (26) and (33):

$$\begin{bmatrix} \eta_{k+1} \\ \delta\Pi_{k+1} \\ \delta\lambda_{k+1}^1 \\ \delta\lambda_{k+1}^2 \end{bmatrix} = \mathcal{K}_k \begin{bmatrix} \eta_k \\ \delta\Pi_k \\ \delta\lambda_k^1 \\ \delta\lambda_k^2 \end{bmatrix} \tag{35}$$

where \mathcal{K}_k is the transition matrix to the next step. Thus,

$$\begin{bmatrix} \eta_N \\ \delta\Pi_N \\ \delta\lambda_N^1 \\ \delta\lambda_N^2 \end{bmatrix} = (\Pi_{k=0}^{N-1} \mathcal{K}_k) \begin{bmatrix} \eta_0 \\ \delta\Pi_0 \\ \delta\lambda_0^1 \\ \delta\lambda_0^2 \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \eta_0 \\ \delta\Pi_0 \\ \delta\lambda_0^1 \\ \delta\lambda_0^2 \end{bmatrix} \tag{36}$$

Since for each step of the NMPC, the original values of the spacecraft attitude and the angular momentum are specified, and the amounts of η_0 and $\delta\Pi_0$ are set to zero. Consequently,

$$\begin{bmatrix} \eta_N \\ \delta\Pi_N \end{bmatrix} = \mathbf{K}_{12} \begin{bmatrix} \delta\lambda_0^1 \\ \delta\lambda_0^2 \end{bmatrix} \tag{37}$$

The submatrix \mathbf{K}_{12} , which is called the sensitivity derivative, determines the sensitivity of the specified terminal boundary conditions to the changes in the (unspecified) initial conditions of the Lagrange multipliers [15]. At this stage and in the context of NMPC formulation, based on given initial conditions R_0 and Π_0 and an initial guess for λ_0^1 and λ_0^2 , the final values for the rigid body attitude matrix and angular momentum, R_N and Π_N , are forward computed and associated with the final values of the Lagrange multipliers, λ_{N-1}^1 and λ_{N-1}^2 , through Eqs. (22) and (24). On the other hand, these latter are also predicted through Eqs. (23) and (25) for $k = N - 2$. Based on these two approaches, the following error function can be defined:

$$Er = \begin{bmatrix} \lambda_{N-1}^1 + S^{-1}((R_N^T Q_1)_A) \\ \lambda_{N-1}^2 - S^{-1}((Q_2S(\Pi_N)_A)) \end{bmatrix} \tag{38}$$

By applying an indirect shooting method (using a similar procedure used in refs. [14, 17]), the guessed initial values of the Lagrange multipliers are re-estimated as:

$$\begin{bmatrix} \lambda_0^1 \\ \lambda_0^2 \end{bmatrix}^{(l+1)} = \begin{bmatrix} \lambda_0^1 \\ \lambda_0^2 \end{bmatrix}^{(l)} - \gamma \left[\frac{\delta Er^{(l)}}{\begin{bmatrix} \delta\lambda_0^1 \\ \delta\lambda_0^2 \end{bmatrix}^{(l)}} \right]^{-1} Er^{(l)} \tag{39}$$

where γ is a real number that belongs to the interval $(0, 1]$, l is an index that specifies the number of iterations executed until $\|Er\| \leq \epsilon$, and ϵ is an allowed preset value. Therefore, the optimal control problem is solved with the inputs $(R_0)^i$, $(\Pi_0)^i$, and the final guess for $(\lambda_0^1)^i$ and $(\lambda_0^2)^i$ at each NMPC time step. The index (i) represents the step number with the maximum value $i_{max} = \frac{\mathcal{T}}{h}$ where \mathcal{T} is the total time of executing the defined maneuver. The process will be repeated from the scratch for the next steps until the entire time allotted for the problem ends.

3.2. Simplification of the sensitivity derivatives and TPBVP equations

In this subsection, a procedure for improving numerical efficiency of the NMPC calculations is presented as the first main contribution of this paper, which consists of simplifying the expression for the sensitivity derivatives and TPBVP equations without affecting the accuracy of the whole process. By removing some nonlinear terms that are non-essential in the iterative shooting step of the algorithm, a reduction in the computation burden of the optimal control sub-problem related to the NMPC method is obtained [18]. Eq. (33) in Section 3.1 is an implicit nonlinear equation for the purpose of computing δu_k that has to be dealt with in the process of estimating $(\lambda_0^1)^l$ and $(\lambda_0^2)^l$. However, Eq. (33) has no direct effect on the overall trend of solving the necessary conditions for optimality and may only increase the number of Newton iterations in the indirect shooting method. In order to be more explicit, at each step of solving the NMPC problem, Eqs. (19)–(26) are solved based on initial guesses for $(\lambda_0^1)^l$ and $(\lambda_0^2)^l$. Then if $\|Er\| > \epsilon$, Eqs. (27)–(33) are solved to rectify those guessed values based on a Newton-like iterative method, and this process continues until $\|Er\| \leq \epsilon$. Therefore, Eqs. (27)–(33) are solved based on constraint-imposed control inputs. Hence, the constraint-related part of Eq. (33) can be removed from the solution procedure since the constraints are already applied to the system through Eqs. (19)–(26). Consequently, Eq. (33) is substituted with the following linear equation:

$$B^T \delta \lambda_k^2 = -S^{-1}((P_3 S(\delta u_k))_A) \tag{40}$$

In addition, the TPBVP equations may be prone to constraints that are activated only when their associated variables exceed some defined limits. In the problem under consideration, constraints on u_k only apply when the limits are exceeded and saturation occurs. Accordingly, assuming that the constraints are still inactive, Eq. (26) can be considered as the linear form below:

$$hB^T \lambda_k^2 = -hS^{-1}((P_3 S(u_k))_A) \tag{41}$$

Then u_k computed through Eq. (41) is checked for being within the allowed range of the imposed constraints. If the restrictions are not met, u_k is re-computed using Eq. (26).

These simplifications can reduce the time of computations to a considerable extent while maintaining the accuracy within acceptable limits compared to complete usage of nonlinear equations. In the next section, the results of applying simplifications proposed above are compared with the results of the methodology employed in refs. [14] and [17], where u_k and δu_k are computed without omitting the nonlinear terms, for the sake of comparison in terms of time saving, efficiency, and accuracy.

4. An example on SO(3)

As an example of this section, a fully actuated spacecraft evolving on SO(3) with Eqs. (7), (8), and (9) as the LGVI and numerical values tabulated in Table I is considered.

In the following figures, the results obtained by fully accounting for the nonlinear terms in both the constrained TPBVP and the sensitivity equations [17] are compared with those extracted by solely solving the TPBVP using the simplified cast of the sensitivity derivatives, and those obtained by simplifying both the TPBVP and the sensitivity derivatives, as discussed in Subsection 3.2. Diagrams of the control inputs u_1 , u_2 , u_3 , and $\|u\|$ are plotted in Fig. 1. The results show a relatively good match independent of the level of simplification employed. In fact, the noteworthy point about these simplifications is that they reduce the computation time of the problem by a considerable amount without affecting the accuracy of

Table I. Numerical values employed in the simulation of the spacecraft.

$R_0 = \exp(S([0.25, 0.5, 0.5]^T))$	$\Pi_0 = [0, 0, 0]^T$
$P_1 = P_2 = Q_1 = Q_2 = 0.01 I_3, \quad P_3 = I_3$	$J = \text{diag}(1, 0.8, 0.8) \text{ kg.m}^2, \quad B = I_3$
$t = 150 \text{ s}, \quad h = 0.4 \text{ s}, \quad N = 5$	$\alpha = 0.0001 \text{ N.m}, \quad \mu = 1e10$
$\lambda_0^1 = [0, 0, 0]^T$	$\lambda_0^2 = [0, 0, 0]^T$

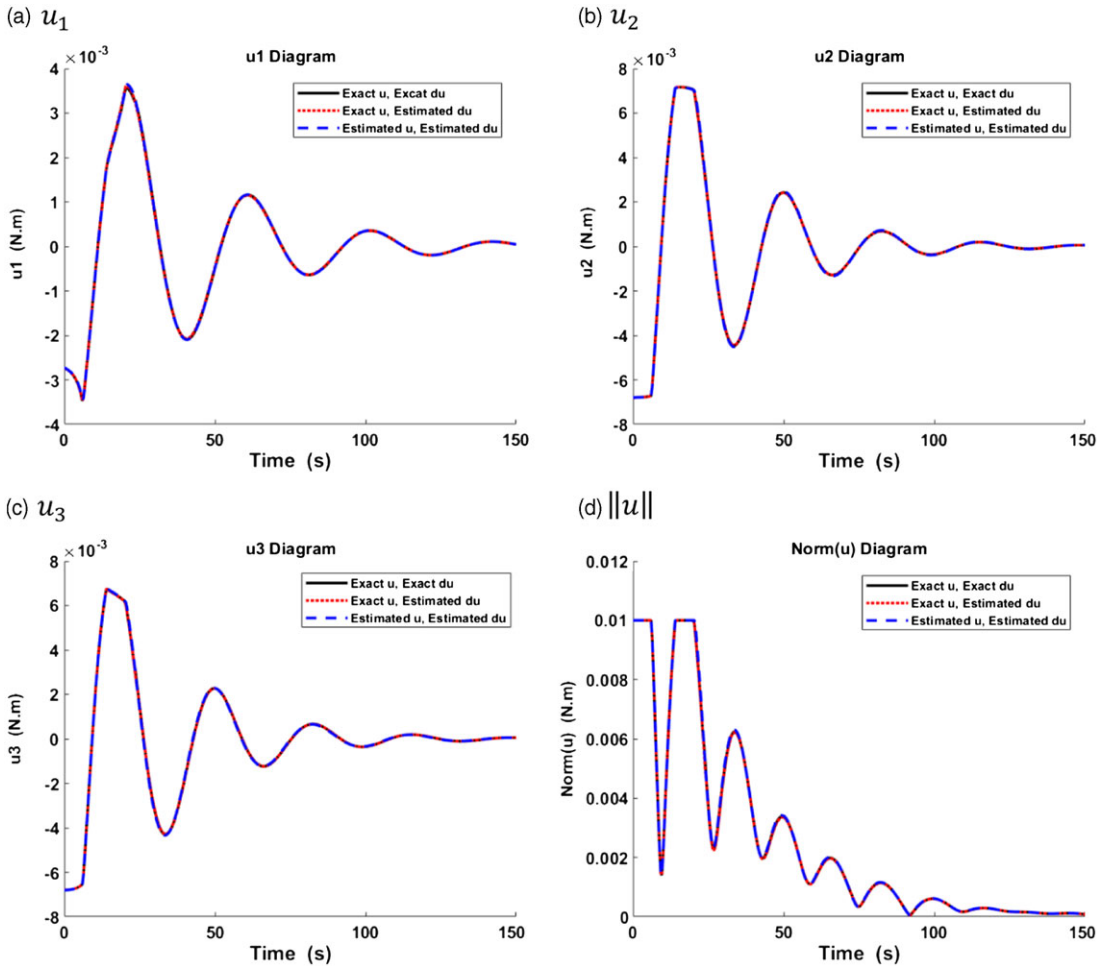


Figure 1. Diagrams of $u_1, u_2, u_3,$ and $\|u\|$, obtained using exact solution, estimated δu , and estimated u and δu .

the results. In order to show the effect of removing the nonlinear parts from Eqs. (26) and (33) on the optimal control action, diagrams of integral square error are plotted in Fig. 2. The black line shows the difference between the elements of exact input which is computed through Eqs. (26) and (33) and the input computed using the simplified sensitivity derivative (Eq. (40)). The red line represents the difference between the exact input and the input computed using both the simplified TPBVP (Eq. (41)) and the simplified sensitivity derivative (Eq. (40)). As described in this figure, using simplified sensitivity derivatives lead to less error in comparison with using both simplified TPBVP and sensitivity derivative. However, the amounts of error are in the order of about 10^{-7} which are definitely negligible. The body-fixed components of angular momentum are illustrated in Fig. 3. As predictable, there is a good agreement between the results achieved regardless of the approach that is used.

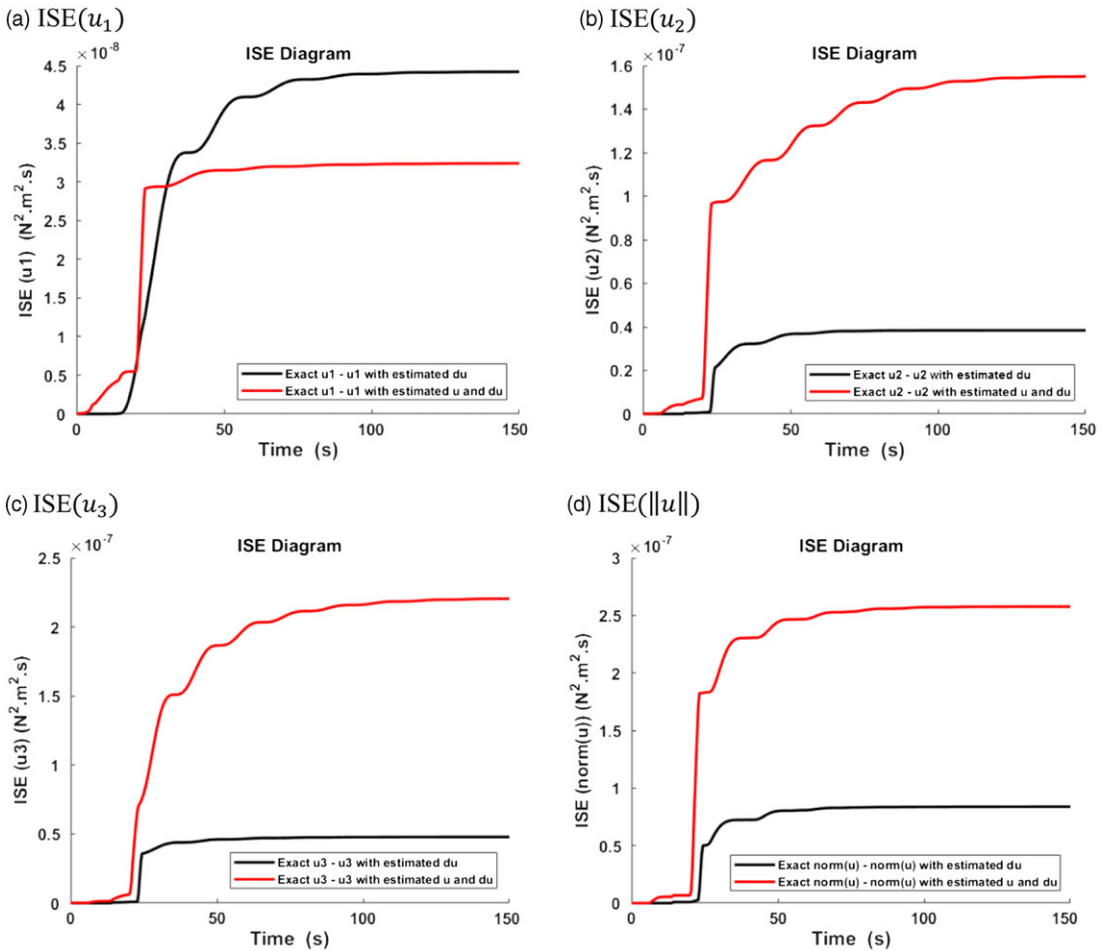


Figure 2. Diagrams of $ISE(u_1)$, $ISE(u_2)$, $ISE(u_3)$, and $ISE(\|u\|)$, obtained using exact solution minus solution extracted using estimated δu , and exact solution minus solution extracted using estimated u and δu .

It is worthy to notice that the similarity between the second and third components in both Figs 1 and 3 is justified as the second and the third moment of inertia of the spacecraft are equal, $J_2 = J_3$, so are the initial conditions R_0 and Π_0 according to Table I, leading to a symmetry in the optimal maneuver. Figure 4 represents the rotation matrices of the spacecraft, R_k , on S^2 . The columns of R_0 and R_N are indicated by dashed and solid colored lines, respectively. By comparing diagrams of Fig. 4, it is completely clear that regardless of which methods are chosen, the attitude maneuver of the spacecraft is the same.

As demonstrated, the results corresponding to the control efforts, angular momenta, and spacecraft attitude obtained by the foregoing methods matched precisely. However, what prioritizes these methods over each other is the time consumption for performing the computations. Table II provides the amounts of time needed to perform one step and the total computation time. In addition, the time of performing each step is plotted against time, for each of the three methods in Fig. 5.

As it is clear by comparing the numerical values of Table II and diagrams of Fig. 5, solving the exact TPBVP equations is very time-consuming. Using Eq. (40) to estimate δu via a linear equation in sensitivity derivatives can reduce the average time of performing each step and the total computation time about 73.4%. By considering both Eqs. (40) and (41) to estimate δu and u , the average time of performing each step and the total computation time would be reduced by about 84.8%.

Table II. Numerical values of time to perform each step and total computational time.

	Exact solution	Estimated δu	Estimated u and δu
Max. time for each step	1.5417 s	0.7834 s	0.7852 s
Min. time for each step	0.1228 s	0.1398 s	0.0591 s
Average time for each step	0.7689 s	0.2048 s	0.1166 s
Total computation time	289.287 s	77.091 s	44.006 s

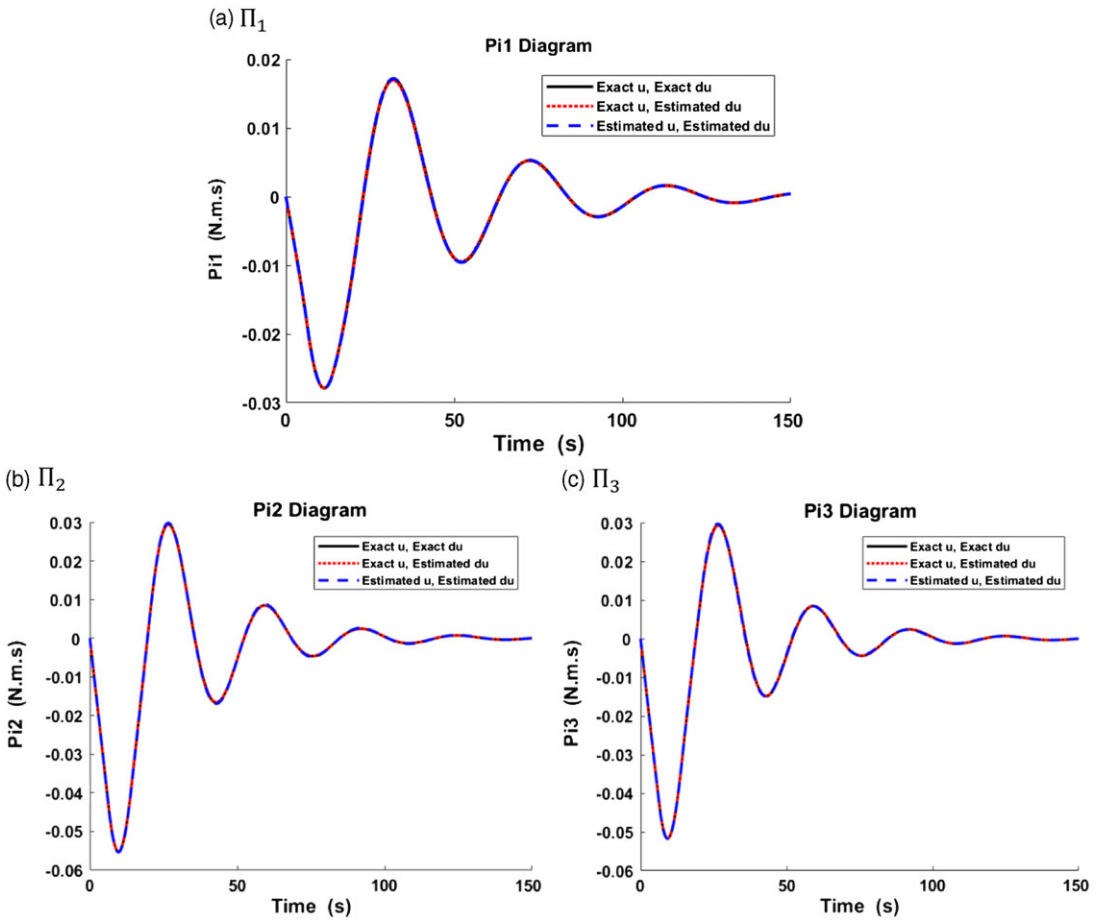


Figure 3. Diagrams of Π_1 , Π_2 , and Π_3 , obtained using exact solution, estimated δu , and estimated u and δu .

By comparing the results, one can come to the conclusion that both procedures presented in Subsection 3.2 have the ability of reducing the optimization process time while keeping the accuracy at a reasonable level. Thus, both methods could be used as a suitable alternative for what is presented in [17] where iterative computations to extract u and δu are performed based on solving nonlinear equations which is unnecessary according to the explanations given in Subsection 3.2. Nevertheless, approximating δu in the calculation of sensitivity derivatives has no direct effect on the general trend of solving the equations and the compliance with the constraints is also guaranteed. In addition, unlike referring to Eq. (41) for estimating u , the constraints do not need to be checked at each iteration. Thus, estimating δu is preferred over estimating both u and δu in the process of solving the NMPC optimization problem and that will be employed in the following sections.

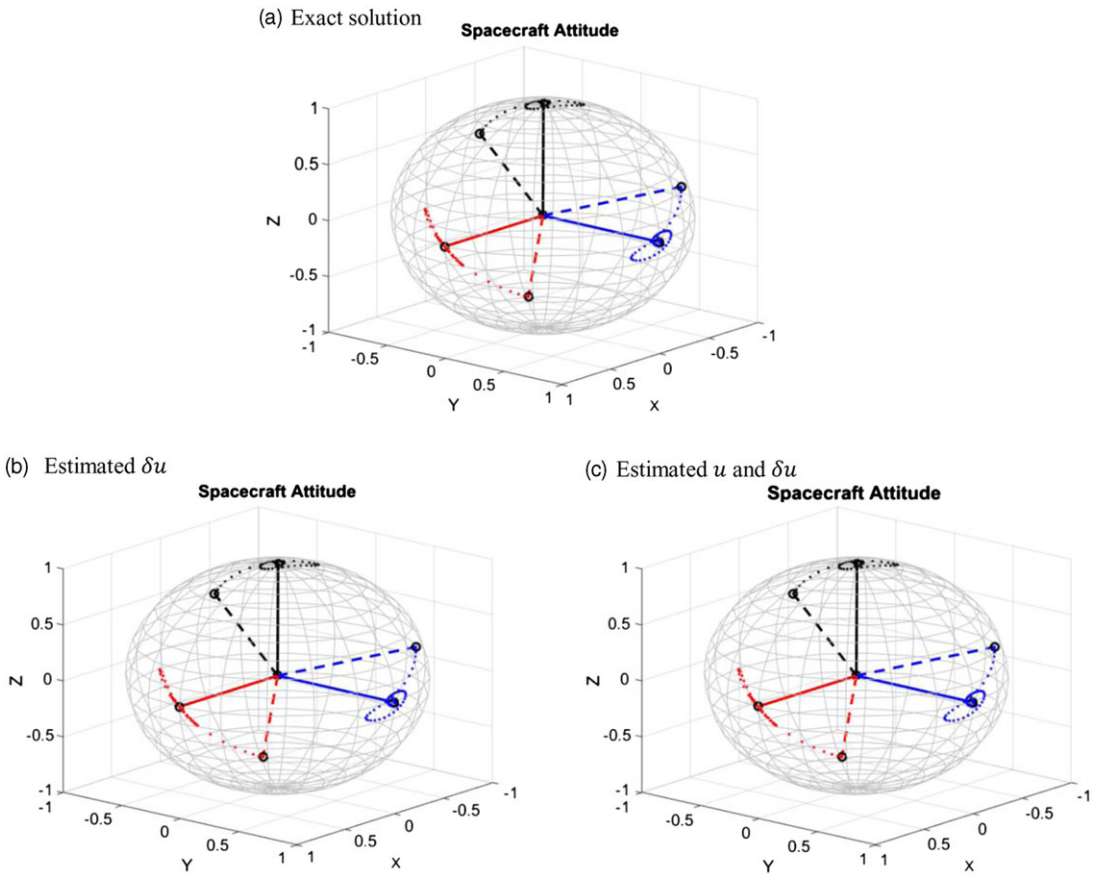


Figure 4. Diagrams of the columns of rotation matrices R_k , obtained using exact solution, estimated δu , and estimated u and δu .

5. Application of NE method in NMPC control of mechanical systems on SO(3)

In this section, it is supposed that the initial conditions of the system and consequently, the initial conditions for each interval may change due to some re-planning. The goal is to find a way to estimate the responses of the system to the new initial conditions without performing the optimization process from the scratch. Thus, the trend ahead is to use the NE method in the formulation of NMPC in order to find a relation between the new initial conditions and the variations induced in the control inputs and states of the system. The responses of the system are thus extracted based on the NE method, reducing the computation time to a considerable extent in comparison with repeating the NMPC process. Indeed, since the NMPC method is a repetitive time-consuming procedure, it makes perfect sense to reformulate the NMPC construction based on NE method to be used for situations where the initial conditions of a once-solved system are changed. In the following, first, a brief introduction to the NE method on Euclidean configuration is given and then, the equations of the reformulated NMPC based on NE method on SO(3) are extracted.

5.1. A brief introduction to NE method on Euclidean Space

NE method is a rapid method for finding the change in the optimal solution according to the changes in the initial conditions and parameters of the system [19, 21]. Suppose that the discrete equations of motion of a system are expressed in the following form on Euclidean space:

$$x_{k+1} = f(x_k, u_k), \quad x_k \in \mathbb{R}^n, \quad u_k \in \mathbb{R}^m \tag{42}$$

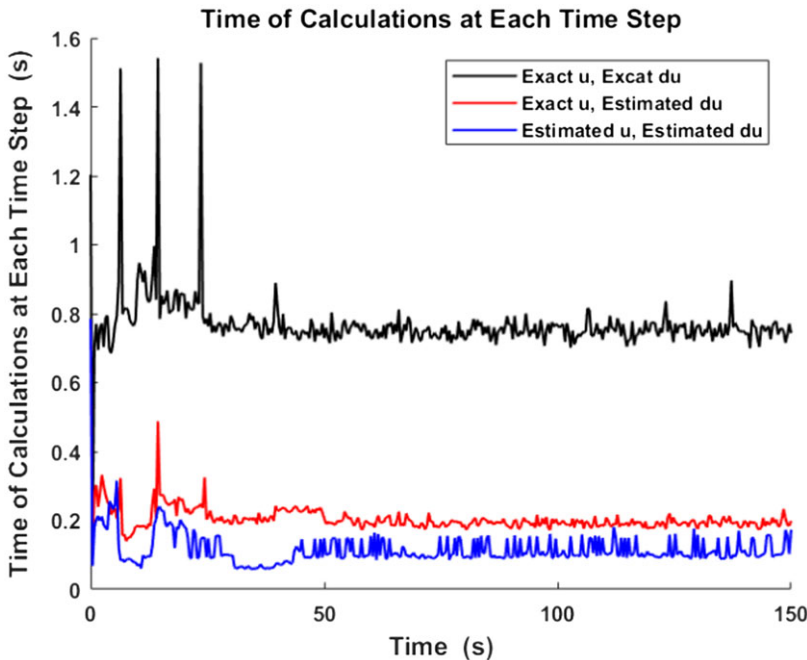


Figure 5. Diagram of time to perform each step, obtained using exact solution, estimated δu , and estimated u and δu .

with the initial conditions and constraints as:

$$x_{k=0} = x_0 \tag{43}$$

$$C(x_k, u_k) \leq 0 \tag{44}$$

The optimal control is obtained through minimizing the following cost functional:

$$\min J_u = \Phi(x_N) + \sum_{k=0}^{N-1} L(x_k, u_k) \tag{45}$$

subjected to Eqs. (42), (43), and (44). The functions f , C , Φ , and L are twice differentiable. By defining the function H as:

$$H(x_k, u_k) = L(x_k, u_k) + \lambda_k^T f(x_k, u_k) + \mu_k^T C(x_k, u_k) \tag{46}$$

the augmented cost functional is obtained by adding the constraints to Eq. (45):

$$\min \bar{J}_u = \Phi(x_N) + \sum_{k=0}^{N-1} (H(x_k, u_k) - \lambda_k^T x_{k+1}) \tag{47}$$

where λ_k and μ_k are the Lagrange multipliers. As well, x_{k+1} is the predicted state vector based on the initial condition x_0 and the input vector u_k at instant k for $k = 0, \dots, N$. Now suppose that there is a small change in the initial conditions of the system. The variation of the initial conditions is defined by $\delta x_0 = \bar{x}_0 - x_0$, where \bar{x}_0 represents the new set of initial conditions. The NE method is established to find δu_k based on minimizing the second-order variation of the system cost functional, since the first-order necessary conditions of optimality vanish at nominal solutions. Therefore, the new optimal control problem is formulated as follows [19, 36]:

$$\min \delta^2 \bar{J}_u = \frac{1}{2} \delta x_N^T \Phi_{x_k x_k}(x_N) \delta x_N + \frac{1}{2} \sum_{k=0}^{N-1} \begin{bmatrix} \delta x_k \\ \delta u_k \end{bmatrix}^T \begin{bmatrix} H_{x_k x_k}(x_k, u_k) & H_{x_k u_k}(x_k, u_k) \\ H_{u_k x_k}(x_k, u_k) & H_{u_k u_k}(x_k, u_k) \end{bmatrix} \begin{bmatrix} \delta x_k \\ \delta u_k \end{bmatrix} \tag{48}$$

subjected to the variation of the dynamics of the system and the variation of the constraints as given below:

$$\delta x_{k+1} = f_{x_k}(x_k, u_k) \delta x_k + f_{u_k}(x_k, u_k) \delta u_k \tag{49}$$

$$\delta x_{k=0} = \delta x_0 \tag{50}$$

$$C_{x_k}(x_k, u_k) \delta x_k + C_{u_k}(x_k, u_k) \delta u_k = 0 \tag{51}$$

The NE solution based on variations of the initial conditions is obtained through solving the above new optimal control problem [19, 36, 37]:

$$\delta u_k = -[I_m \quad 0_{\aleph}] \begin{bmatrix} Z_{u_k u_k}(x_k, u_k) & C_{u_k}^T(x_k, u_k) \\ C_{u_k}(x_k, u_k) & 0_{\Upsilon} \end{bmatrix}^{-1} \begin{bmatrix} Z_{u_k x_k}(x_k, u_k) \\ C_{x_k}(x_k, u_k) \end{bmatrix} x_k \tag{52}$$

where \aleph and Υ depend on whether the constraints are active or not. Moreover,

$$Z_{u_k u_k}(x_k, u_k) = H_{u_k u_k}(x_k, u_k) + f_{u_k}(x_k, u_k)^T G(x_{k+1}, u_{k+1}) f_{u_k}(x_k, u_k) \tag{53}$$

$$Z_{u_k x_k}(x_k, u_k) = Z_{x_k u_k}(x_k, u_k)^T = H_{u_k x_k}(x_k, u_k) + f_{u_k}(x_k, u_k)^T G(x_{k+1}, u_{k+1}) f_{x_k}(x_k, u_k) \tag{54}$$

$$Z_{x_k x_k}(x_k, u_k) = H_{x_k x_k}(x_k, u_k) + f_{x_k}(x_k, u_k)^T G(x_{k+1}, u_{k+1}) f_{x_k}(x_k, u_k) \tag{55}$$

where

$$G(x_k, u_k) = Z_{x_k x_k}(x_k, u_k) - [Z_{x_k u_k}(x_k, u_k) \quad C_{x_k}^T(x_k, u_k)] \begin{bmatrix} Z_{u_k u_k}(x_k, u_k) & C_{u_k}^T(x_k, u_k) \\ C_{u_k}(x_k, u_k) & 0_{\Upsilon} \end{bmatrix}^{-1} \begin{bmatrix} Z_{u_k x_k}(x_k, u_k) \\ C_{x_k}(x_k, u_k) \end{bmatrix} \tag{56}$$

$k = 0, \dots, N - 1$

$$G(x_N, u_N) = \Phi_{x_N x_N}(x_N, u_N) \tag{57}$$

Consequently, the NE method which is established based on the following equations [38]:

$$\bar{x}_k = x_k + \delta x_k \tag{58}$$

$$\bar{u}_k = u_k + \delta u_k \tag{59}$$

provides a method to find the updated optimal control for the new set of initial conditions using the above-mentioned recursive method.

5.2. Equations extraction for application of NE method into NMPC on SO(3)

The equations presented in Subsection 3.1 should be converted into a suitable form for implementing the NE method in the NMPC on SO(3). Hence, Eq. (47) is rewritten in the following form:

$$\begin{aligned} \bar{J}_d &= \Phi_d(R_N, S(\Pi_N)) + \sum_{k=0}^{N-1} H_d(R_k, S(\Pi_k), S(u_k)) - \ll (\lambda_k^1)^\diamond, \log(R_k^T R_{k+1}) \gg - \ll S(\lambda_k^2), (\Pi_{k+1})^\diamond \gg \\ &+ \mu^T \psi_d(C_d(R_k, S(\Pi_k), S(u_k))) \end{aligned} \tag{60}$$

where

$$H_d(R_k, S(\Pi_k), S(u_k)) = \mathcal{L}_d(R_k, S(\Pi_k), S(u_k)) + \ll (\lambda_k^1)^\diamond, \log(F_k) \gg + \ll S(\lambda_k^2), (F_k^T \Pi_k + hBu_k)^\diamond \gg \tag{61}$$

By denoting $v(k) = [\eta_k, \delta \Pi_k]^T$, the second variation of \bar{J}_d would be

$$\delta^2 \bar{J}_d = \frac{1}{2} v(N)^T \Phi_{dvv}(N) v(N) + \frac{1}{2} \sum_{k=0}^{N-1} \begin{bmatrix} v(k) \\ \delta u_k \end{bmatrix}^T \begin{bmatrix} H_{dvv}(k) & H_{dvv}(k) \\ H_{dvv}(k) & H_{dvv}(k) \end{bmatrix} \begin{bmatrix} v(k) \\ \delta u_k \end{bmatrix} \tag{62}$$

constrained to the first variation of the necessary conditions for optimality as follows:

$$\delta R_{k+1} = \delta(R_k F_k) = [F_k^T \quad M_k] v(k) \tag{63}$$

$$\delta \Pi_{k+1} = \delta(F_k^T \Pi_k + h B u_k) = [0_{3 \times 3} \quad (F_k^T + S(F_k^T \Pi_k) M_k)] v(k) + h B \delta u_k \tag{64}$$

$$C_{dv(k)}^T(k) v(k) + C_{du_k}^T(k) \delta u_k = 0 \tag{65}$$

Similar to what has been explained in Subsection 3.1 and in order to find the relationship between the changes in the initial conditions and the associated variations in the control inputs, the following equations for mechanical systems evolving on the manifold SO(3) are extracted:

$$\delta u_k = -[I_3 \quad 0_{\mathbb{R}}] \begin{bmatrix} Z_{uu}(k) & C_{du_k}^T(k) \\ C_{du_k}(k) & 0_{\mathbb{T}} \end{bmatrix}^{-1} \begin{bmatrix} Z_{uv}(k) \\ C_{dv(k)}(k) \end{bmatrix} v(k) \tag{66}$$

Also,

$$Z_{uu}(k) = H_{dvv}(k) + \begin{bmatrix} \frac{\delta R_{k+1}}{\delta u_k} & \frac{\delta \Pi_{k+1}}{\delta u_k} \end{bmatrix} G(k+1) \begin{bmatrix} \frac{\delta R_{k+1}}{\delta u_k} & \frac{\delta \Pi_{k+1}}{\delta u_k} \end{bmatrix}^T \tag{67}$$

$$Z_{uv}(k) = Z_{vu}(k)^T = H_{dvv}(k) + \begin{bmatrix} \frac{\delta R_{k+1}}{\delta u_k} & \frac{\delta \Pi_{k+1}}{\delta u_k} \end{bmatrix} G(k+1) \begin{bmatrix} \frac{\delta R_{k+1}}{\delta \eta_k} & \frac{\delta R_{k+1}}{\delta \Pi_k} & \frac{\delta \Pi_{k+1}}{\delta \eta_k} & \frac{\delta \Pi_{k+1}}{\delta \Pi_k} \end{bmatrix}^T \tag{68}$$

$$Z_{vv}(k) = H_{dvv}(k) + \begin{bmatrix} \frac{\delta R_{k+1}}{\delta \eta_k} & \frac{\delta R_{k+1}}{\delta \Pi_k} & \frac{\delta \Pi_{k+1}}{\delta \eta_k} & \frac{\delta \Pi_{k+1}}{\delta \Pi_k} \end{bmatrix} G(k+1) \begin{bmatrix} \frac{\delta R_{k+1}}{\delta \eta_k} & \frac{\delta R_{k+1}}{\delta \Pi_k} & \frac{\delta \Pi_{k+1}}{\delta \eta_k} & \frac{\delta \Pi_{k+1}}{\delta \Pi_k} \end{bmatrix}^T \tag{69}$$

where $G(k), k = 0, \dots, N$, are obtained by substituting v instead of $x(k)$ in Eqs. (56) and (57).

The elements in Eqs. (60)–(69) are extracted considering Eqs. (15)–(18). Should any changes occur in the initial conditions of the system, above equations can be used instead of employing the normal trend, resulting to a considerable reduction in the amount of computations needed for the successive optimizations related to the NMPC process on SO(3). The remarkable superiority of this method over solving the optimization problem from the scratch in reducing the computation burden and hence increasing the speed of calculations is shown in the next section through an example.

6. NMPC and NE control of a spacecraft on SO(3) with changes in the initial conditions

In this section, the extracted relationships of Subsection 5.2 are implemented on the example of Section 4. It is supposed that the initial conditions of the system are altered. The example is solved using three methods. First, it is solved by the method provided in Subsection 3.2. Second, the NE method is used to estimate the necessary adaptation of the control inputs and its effect on the response of the system, and finally, the combination of these two methods is employed to get the solution. The combination of the methods is established based on solving the problem with altered initial conditions using NE method. However, at some predetermined steps the optimization problem is solved from the scratch

Table III. First and second set of initial conditions.

$\underline{\Pi}_0 = [0 \ 0 \ 0]^T$	$\underline{R}_0 = \exp(S ([0.25 \ 0.5 \ 0.5]^T))$
$\overline{\Pi}_0 = [0.0025 \ 0.0025 \ 0.0025]^T$	$\overline{R}_0 = \exp(S ([0.3 \ 0.5 \ 0.5]^T))$

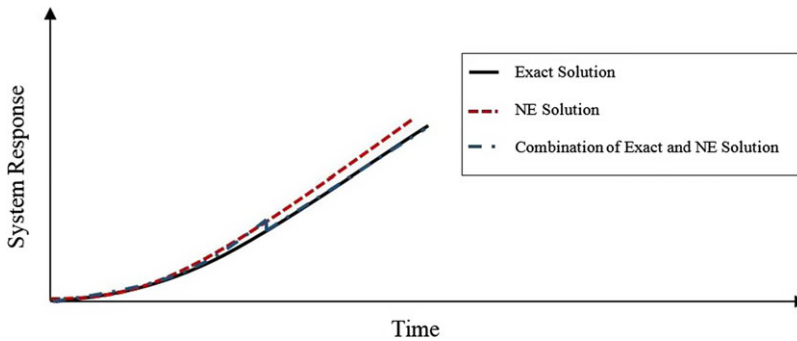


Figure 6. The basic schematic diagram which shows how the three methods work for solving the NMPC problem.

in order to compensate for error accumulation. In fact, whenever the control inputs and system variables deviate inexorably from their actual value, a round of running exact NMPC-related optimization at intermediate steps takes them back to the expected actual path. Figure 6 shows the logic behind this proposed method in a simplified schematic form. The diagrams of the control inputs and the evolution of the system states as well as the diagrams of the time needed to perform each step are plotted and the results are compared in order to determine the more appropriate method.

6.1. The spacecraft problem: Exact solution of the NMPC versus NE solution

Consider the first and second set of initial conditions of the system given in Table III. The (–) symbol above the variables indicates the new initial conditions.

The example has been solved for the second set of initial conditions in two ways: the first approach is according to the method offered in Subsection 3.2, part A, and the second one consists of using the NE method presented in Subsection 5.2. Figure 7 shows the control inputs obtained by using these two methods. Based on these diagrams, the control inputs do not match for the initial steps but the differences vanish as time goes ahead. However, based on what can be seen in Fig. 8, the resulted angular momenta do not match and the difference even accentuates with time. The same trend can also be seen in the attitude of the spacecraft, plotted in Fig. 9. In this figure, black, green, and pink lines correspond to the exact solution of TPBVP, whereas blue, red, and cyan lines indicate the NE solution. As it is clear, the two methods represent considerable differences, and this happens due to the error accumulated at each step. Therefore, one can conclude that the NE method accompanied with exact TPBVP solutions calculated at some intermediate steps may provide better answers.

6.2. The spacecraft problem: Exact solution of the NMPC versus the combination of NE and exact TPBVP solution

In this subsection, the combination of NE and the exact solution of NMPC-related TPBVP is used in order to prevent the accumulation of errors during the whole run time of the simulation. The procedure to follow is explained next: As a first step, the “exact” solution to the new initial conditions is obtained through the TPBVP equations using the method explained in Subsection 3.2, part A. Then, for a number of steps which depends on the size of the problem, the NE method is used to find the neighborhood

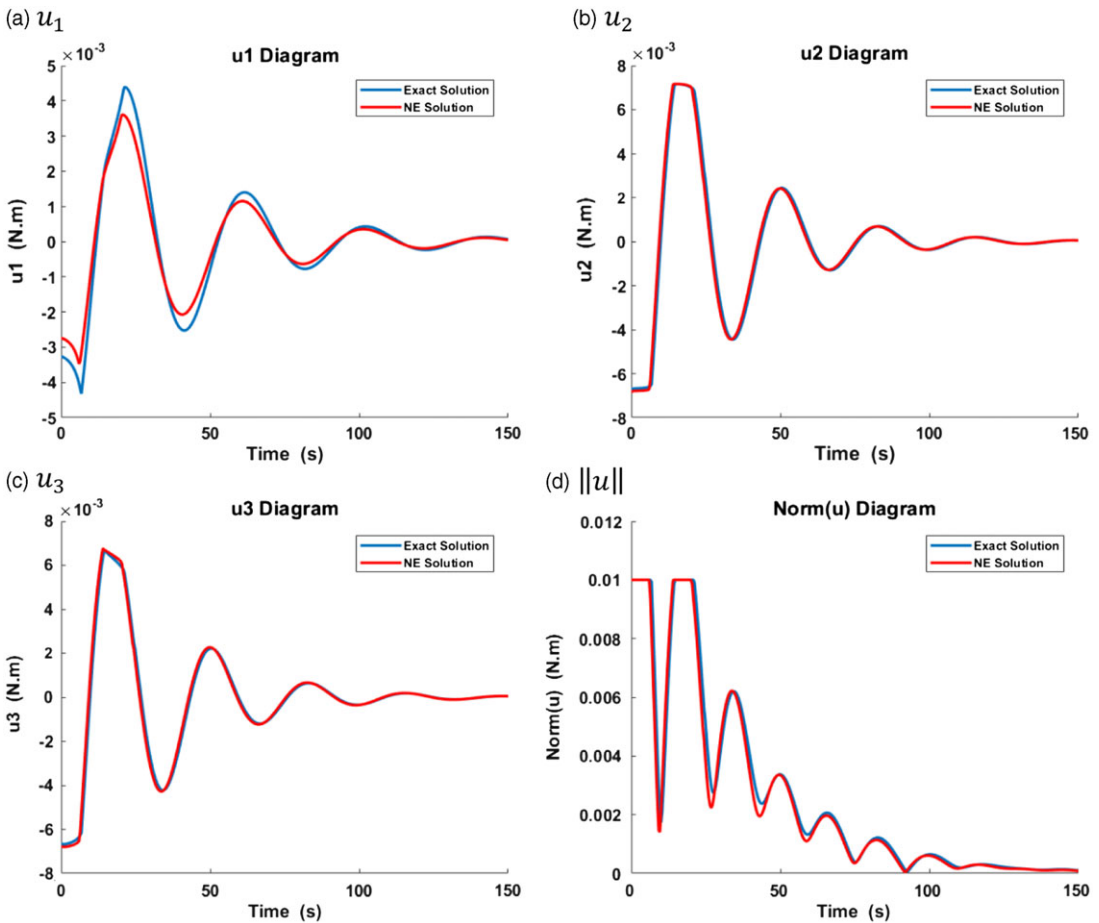


Figure 7. Diagrams of u_1 , u_2 , u_3 , and $\|u\|$, obtained using exact solution for the NMPC-related TPBVP and the NE solution.

solution. Next, the exact solution is computed for another step via NMPC-related TPBVP and the whole process continues until the end. The strategy of our proposed procedure relies on employing the direct solution of NMPC-related TPBVP to return to the actual path whenever the neighborhood solution is deemed to deviate too much. By this way, a compromise between accuracy and the amount of computational operations can be reached. The diagrams for this example seem to confirm this claim. It turns out from Figs. 10, 11, and 12 that this method is able to regulate the error and to extract the relevant solution. In this example, the number of steps to reiterate the exact solution is considered equal to 10. Needless to say, the lower the number, the more accurate the result, but the heavier the calculations.

6.3. Time comparison of the introduced methods

The methods provided in Subsections 6.2 and 6.3 should also be compared in terms of computational time. The maximum, minimum, and the average values of time consumed to perform one step and the total computation duration are given in Table IV for the aforementioned approaches, namely i. solving the exact NMPC-related TPBVP directly according to Subsection 3.2, ii. using the NE method to estimate the control inputs and system states similar to what is offered in Subsection 6.1, and iii. using the combination of these two methods according to what is presented in Subsection 6.2. The time consumed for performing each step is plotted against time for each of the three methods in Fig. 13 as well.

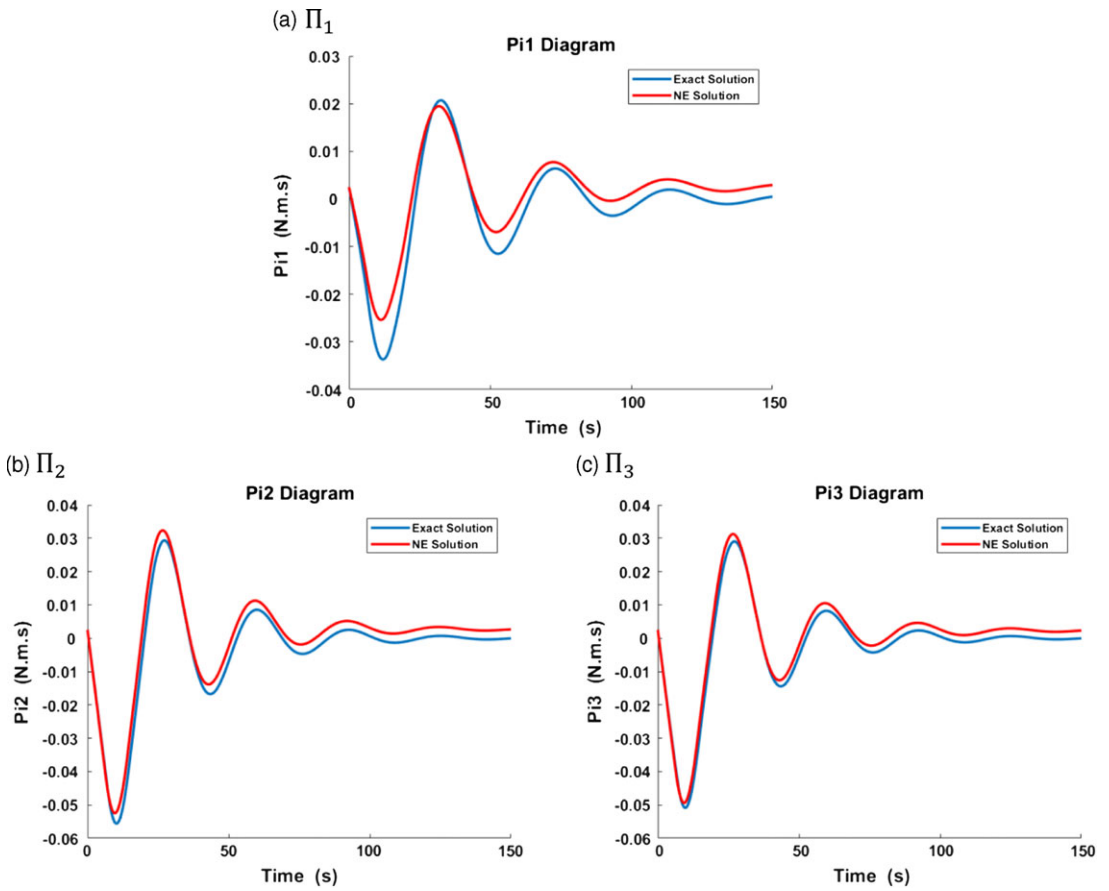


Figure 8. Diagrams of Π_1 , Π_2 , and Π_3 , obtained using exact solution for the NMPC-related TPBVP and the NE solution.

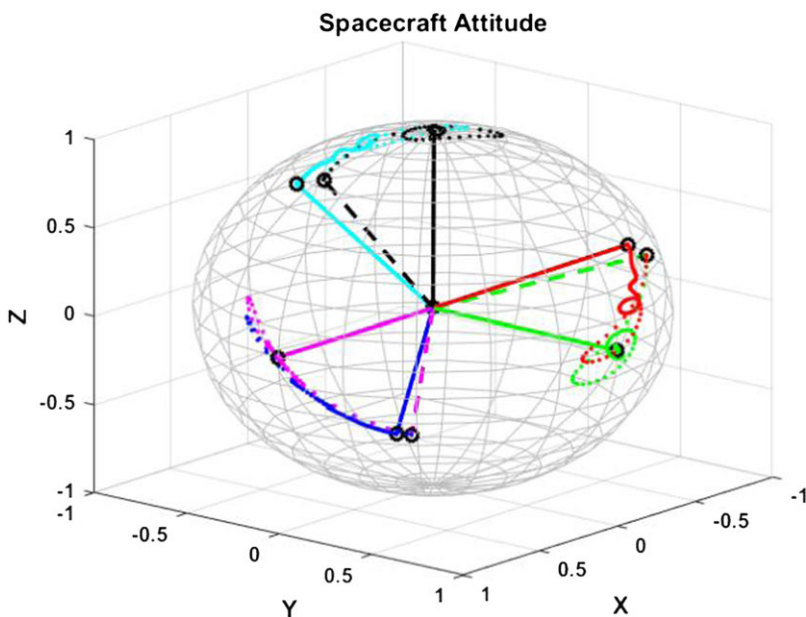


Figure 9. Diagrams of the columns of the rotation matrices R_k , obtained using exact solution for the NMPC-related TPBVP, and the NE solution.

Table IV. Numerical values of time to perform each step and total computational time.

	TPBVP solution	NE solution	TPBVP and NE solution
Max. time for each step	0.7955 s	0.1185 s	0.7422 s
Min. time for each step	0.1417 s	0.0040 s	0.0102 s
Average time for each step	0.2978 s	0.0069 s	0.0431 s
Total computation time	112.079 s	2.653 s	16.267 s

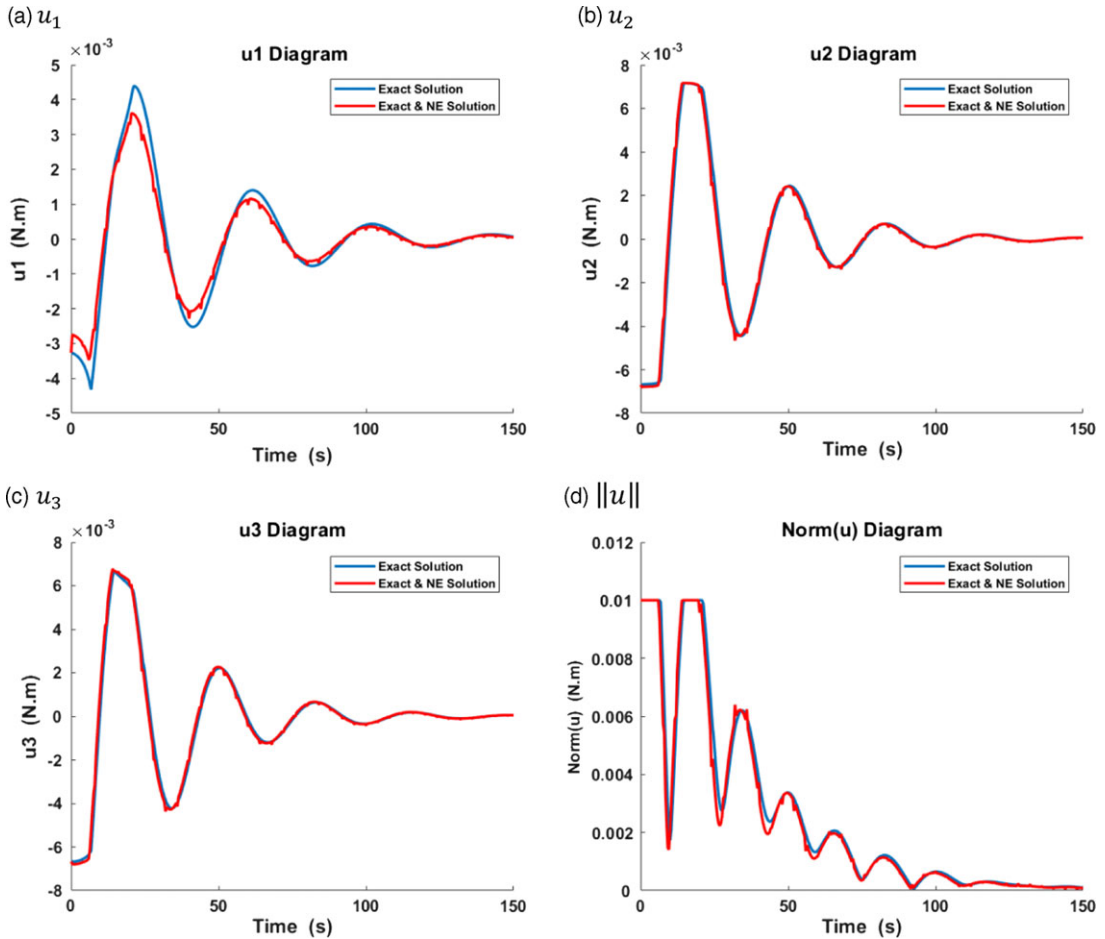


Figure 10. Diagrams of u_1 , u_2 , u_3 , and $\|u\|$, obtained using exact solution for the NMPC-related TPBVP, and the combination of the NE and the exact TPBVP solution.

Considering the numerical values of Table IV, it is completely clear that using the NE method alone is absolutely superior to the other two methods in terms of computation time reduction. By using this method, the average time of performing each step and the total computation time would reduce by about 97.6%. However, the results of Subsection 6.1 indicate that using the NE method alone is not sufficient to estimate the expected responses of the system properly. Consequently, the combination of using the NE method and solving the NMPC optimal control problem using the exact TPBVP equations at some predefined intermediate steps, as presented in Subsection 6.2, is the suggested method of this article to meet the dual objectives. In addition, through achieving an accurately enough estimate of the control inputs and system states, this method can reduce the average time of performing each step by 85.5% and the total computation time by 86.1%. So, if some changes occur in the initial conditions of the system,

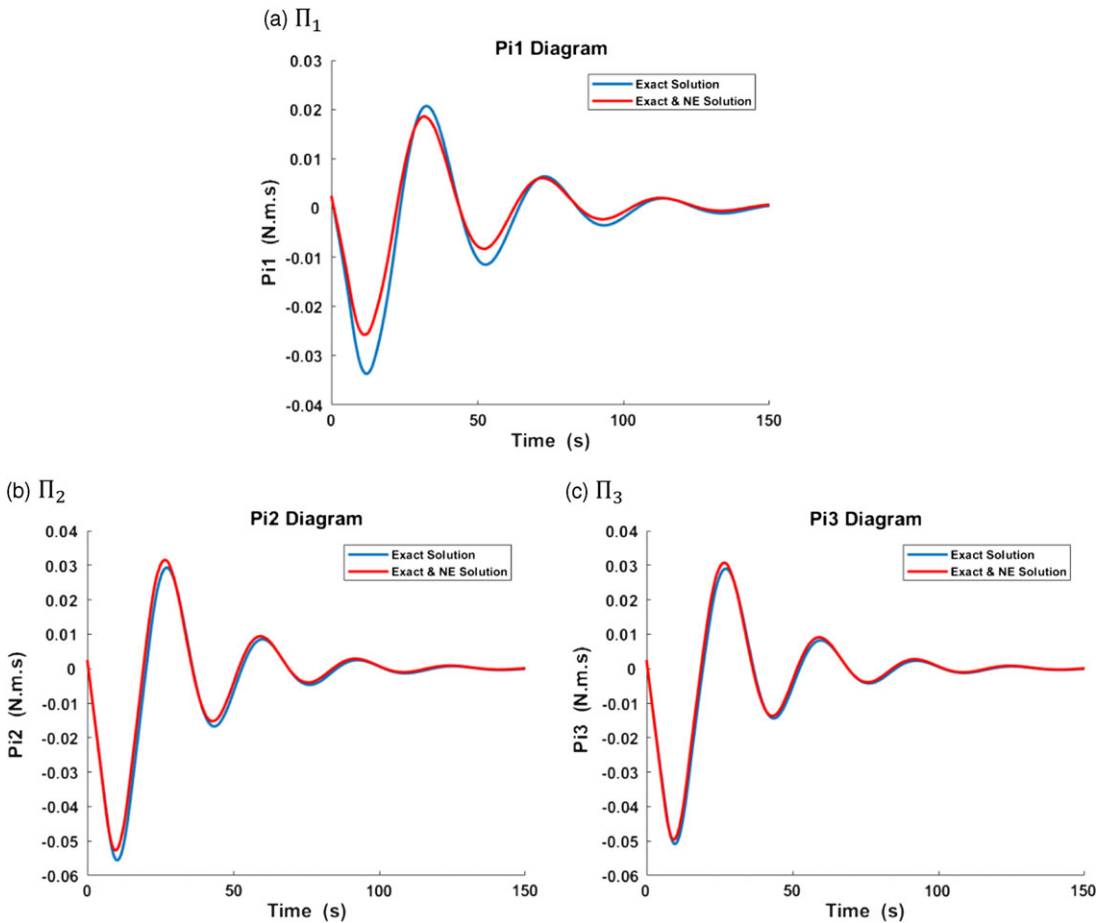


Figure 11. Diagrams of Π_1 , Π_2 , and Π_3 , obtained using exact solution for the NMPC-related TPBVP, and the combination of the NE and the exact TPBVP solution.

the method as presented in Subsection 6.2 can prove efficient in updating the optimal control inputs and responses of the system without solving the whole optimization process from the scratch.

7. Conclusion

In this article, the NMPC of mechanical systems on SO(3) is considered. LGVI are used to extract the necessary conditions for optimality. The extracted TPBVP is solved using an indirect shooting method based on iterative Newton-like schemes. The proposed method in this article offers two computation tricks based on reducing the nonlinearity of the equations in order to alleviate the computation burden of solving TPBVP and sensitivity derivatives. These simplifications are explicitly applied to the equations and their effects on reducing the computation time while keeping the accuracy of the results are investigated through an example. Since the simplification applied on the sensitivity derivatives reduces the optimization time significantly while it has no effect on the overall process of the optimization (in contrast to both TPBVP and sensitivity derivatives simplification), owing to the fact that the constraints of the equations are ultimately applied in the last round of calculations, it is designated as the preferred procedure. The obtained numerical values for the integral square error and the percentage of reducing the computation time express the efficiency of the proposed method in a better way.

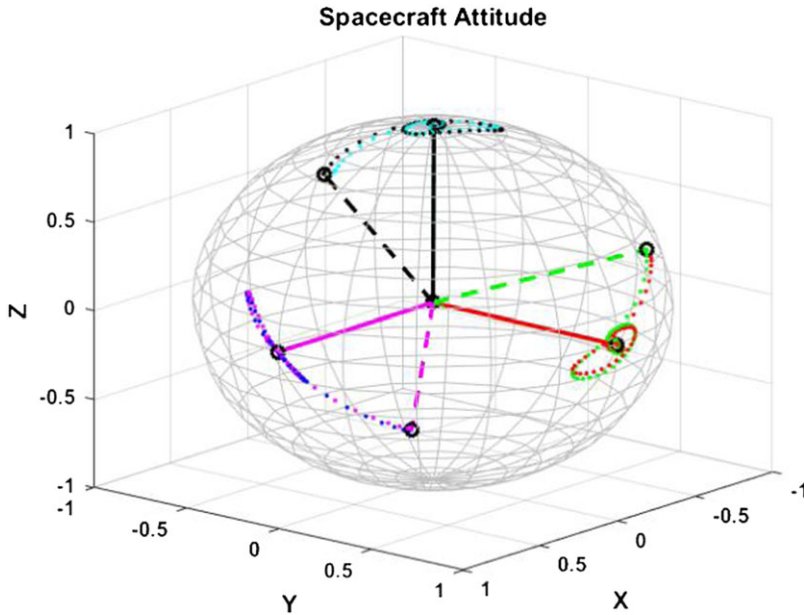


Figure 12. Diagrams of the columns of rotation matrices R_k , obtained using exact solution for the NMPC-related TPBVP, and the combination of the NE and the exact TPBVP solution.

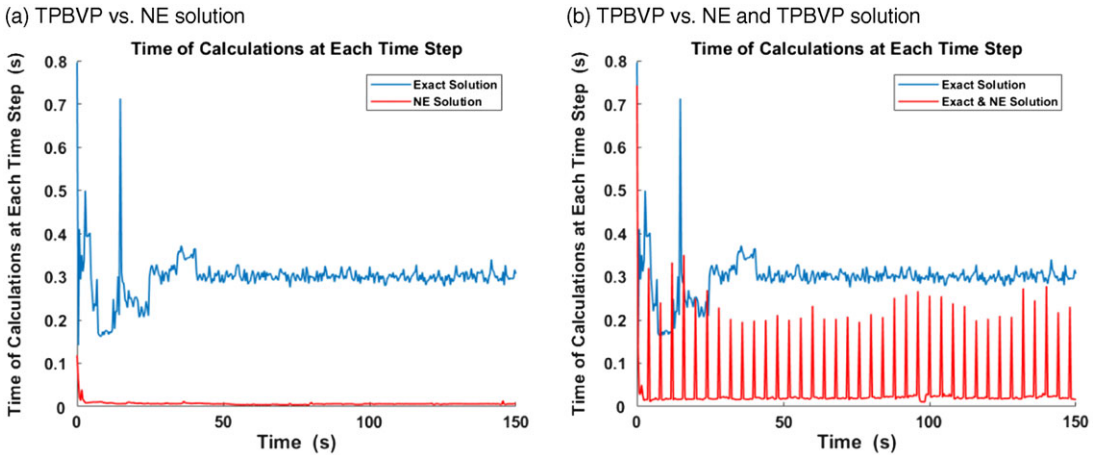


Figure 13. Diagram of time to perform each step, obtained using i. exact solution for the NMPC-related TPBVP, ii. the NE solution, and iii. the combination of the NE and the exact TPBVP solution.

In the following, it is supposed that the initial conditions of the system may be prone to some changes. Instead of solving the issue from the scratch, the NE method is used to estimate the corresponding variations in the control inputs and system states based on the variations in the initial conditions. The NE method is employed alone or in intermittent combination with the exact solution of the NMPC-related optimal control problem at some predefined intermediate steps, in order to remediate from error accumulation. The results show that although using the NE method alone can reduce the computation time of the problem extensively, nonetheless it does not provide the required accuracy. On the other hand, using the combination of NE method together with intermittent exact solutions of TPBVP at some limited steps can effectively estimate the control inputs and system states related to the NMPC problem with altered initial conditions while still reducing the computation time to a considerable extent.

The methods proposed in this article could also be implemented on underactuated systems on $SO(3)$, and systems evolving on $SE(3)$. Moreover, experiments can be considered for taking implementation issues into account. All these issues will hopefully be addressed in our future works.

Authors contributions. SB, HK, and MK conceived and designed the model; SB and HK proposed the computational framework; SB performed simulations; SB, HK, and MK analyzed and interpreted the results; and SB and HK wrote the paper.

Financial support. There was no financial support.

Conflict of interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] U. V. Kalabic, R. Gupta, S. Di Cairano, A. M. Bloch and I. V. Kolmanovsky, “MPC on manifolds with an application to the control of spacecraft attitude on $SO(3)$,” *Automatica* **76**, 293–300 (2017).
- [2] U. V. Kalabic, R. Gupta, S. Di Cairano, A. M. Bloch and I. V. Kolmanovsky, “Constrained Spacecraft Attitude Control on $SO(3)$ Using Reference Governors and Nonlinear Model Predictive Control,” *In: American Control Conference*, Portland, Oregon (2014) pp. 5586–5593.
- [3] M. Kamel, K. Alexis, M. Achtelik and R. Siegwart, “Fast Nonlinear Model Predictive Control for Multicopter Attitude Tracking on $SO(3)$,” *In: IEEE Conference on Control Applications*, Sydney, Australia (2015) pp. 1160–1166.
- [4] D. E. Chang, K. S. Phogat and J. Choi, “Model predictive tracking control for invariant systems on matrix lie groups via stable embedding into euclidean spaces,” *IEEE Trans. Automat. Contr.* **65**(7), 3191–3198 (2020).
- [5] S. Hong, J. H. Kim and H. W. Park, “Real-Time Constrained Nonlinear Model Predictive Control on $SO(3)$ for Dynamic Legged Locomotion,” *In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, USA (2020) pp. 3982–3989.
- [6] J. C. Pereira, V. J. S. Leite and G. V. Raffo, “Nonlinear model predictive control on $SE(3)$ for quadrotor aggressive maneuvers,” *J. Intell. Robot. Syst.* **101**(62) (2021). doi: [10.1007/s10846-021-01310-8](https://doi.org/10.1007/s10846-021-01310-8).
- [7] S. Mansourinasab, M. Sojoodi and S. R. Moghadas, “Model predictive control for a 3D pendulum on $SO(3)$ manifold using convex optimization,” *Control Optim. Appl. Math.* **4**(2), 69–80 (2019).
- [8] Z. Sabeh, M. Shamsi and I. M. Novon, “An indirect shooting method based on the POD/DEIM technique for distributed optimal control of the wave equation,” *Int. J. Numer. Meth. Fl.* **86**, 1–27 (2017).
- [9] E. Hairer, C. Lubich and G. Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. 2nd edition (Springer-Verlag Berlin Heidelberg, 2006).
- [10] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett and A. Zanna, “Lie-Group methods,” *Acta Numer.* **9**, 215–265 (2000).
- [11] M. Leok and T. Shingel, “General techniques for constructing variational integrators,” *Front. Math. China* **7**(2), 273–303 (2012).
- [12] J. E. Marsden and M. West, “Discrete mechanics and variational integrators,” *Acta Numer.* **10**, 357–514 (2001).
- [13] M. Leok, “An Overview of Lie Group Variational Integrators and Their Applications to Optimal Control,” *In: International Conference on Scientific Computation and Differential Equations*, Saint-Malo, France (2007).
- [14] T. Lee, M. Leok and N. H. McClamroch, “Optimal attitude control of a rigid body using geometrically exact computations on $SO(3)$,” *J. Dyn. Control Syst.* **14**(4), 465–487 (2008).
- [15] T. Lee, N. H. McClamroch and M. Leok, “Attitude Maneuvers of a Rigid Spacecraft in a Circular Orbit,” *In: Proceedings of the American Control Conference*, Minnesota, USA (2006) pp. 1742–1747.
- [16] T. Lee, N. H. McClamroch and M. Leok, “Optimal Control of a Rigid Body using Geometrically Exact Computations on $SE(3)$,” *In: 45th IEEE Conference on Decision and Control*, San Diego, CA, USA (2006) pp. 2710–2715.
- [17] R. Gupta, U. V. Kalabic, S. Di Cairano, A. M. Bloch and I. V. Kolmanovsky, “Constrained Spacecraft Attitude Control on $SO(3)$ Using Fast Nonlinear Model Predictive Control,” *In: American Control Conference*, Chicago, IL, USA (2015) pp. 2980–2986.
- [18] S. Bagherzadeh, H. Karimpour and M. Keshmiri, “Efficient Numerical Trends for Nonlinear Model Predictive Control of a Rigid Body Spacecraft on $SE(3)$,” *In: 9th RSI International Conference on Robotics and Mechatronics (ICRoM)*, Tehran, Iran (2021).
- [19] A. E. Bryson and Y. C. Ho. *Applied Optimal Control* (Taylor & Francis, New York, 1975).
- [20] H. J. Pesch, “Real-Time computation of feedback controls for constrained optimal control problems. Part1: Neighbouring extremals,” *Optim. Contr. Appl. Met.* **10**, 129–145 (1989).
- [21] A. M. Bloch, R. Gupta and I. V. Kolmanovsky, “Neighboring extremal optimal control for mechanical systems on Riemannian Manifolds,” *J. Geom. Mech.* **8**(3), 257–272 (2016).
- [22] R. Ghaemi, J. Sun and I. V. Kolmanovsky, “A Neighboring Extremal Approach to Nonlinear Model Predictive Control,” *In: 8th IFAC Symposium on Nonlinear Control Systems*, Bologna, Italy (2010) pp. 747–752.
- [23] L. Würth, R. Hannemann and W. Marquardt, “Neighboring-Extremal updates for nonlinear model-Predictive control and dynamic real-time optimization,” *J. Process. Contr.* **19**(8), 1277–1288 (2009).

- [24] E. Esche and J. U. Repke, “Dynamic process operation under demand response – a review of methods and tools,” *Chem. Ing. Tech.* **92**(12), 1898–1909 (2020).
- [25] L. Würth, R. Hannemann and W. Marquardt, “A Two-layer architecture for economically optimal process control and operation,” *J. Process. Contr.* **21**(3), 311–321 (2011).
- [26] T. Lee, N. H. McClamroch and M. Leok, “A Lie Group Variational Integrator for the Attitude Dynamics of a Rigid Body with Applications to the 3D Pendulum,” **In: Proceedings of the IEEE Conference on Control Applications**, Toronto, Canada (2005) pp. 962–967.
- [27] T. Lee, M. Leok and N. H. McClamroch, “Lie group variational integrators for the full body problem,” *Comput. Method Appl. Mech. Eng.* **196**(29), 2907–2924 (2007).
- [28] D. D. Holm. *Geometric Mechanics, Part II: Rotating, Translating and Rolling*. 2nd edition (Imperial College Press, London, 2011).
- [29] F. Bullo and A. D. Lewis. *Geometric Control of Mechanical Systems* (Springer, New York, 2005).
- [30] I. I. Hussein, M. Leok, A. K. Sanyal and A. M. Bloch, “A Discrete Variational Integrator for Optimal Control Problems on $SO(3)$,” **In: Proceedings of the 45th IEEE Conference on Decision and Control**, CA, USA (2006).
- [31] D. L. Poole and A. K. Mackworth. *Artificial Intelligence: Foundations of Computational Agents*. 2nd edition (Cambridge University Press, Cambridge, 2017).
- [32] D. D. Holm, T. Schmah and C. Stoica. *Geometric Mechanics and Symmetry: From Finite to Infinite Dimensions* (Oxford University Press, Oxford, 2009).
- [33] U. V. Kalabic, R. Gupta, S. D. Cairano, A. M. Bloch and I. V. Kolmanovsky, “MPC on manifolds with applications to the control of systems on matrix lie groups, 2015-09-29, arXiv: 1509.08567v1 [math.OC].
- [34] M. R. Azizi and J. Keighobadi, “Point stabilization of nonholonomic spherical mobile robot using nonlinear model predictive control,” *Robot. Auton. Syst.* **98**, 347–359 (2017).
- [35] R. Findeisen, L. Imsland, F. Allgower and B. A. Foss, “State and output feedback nonlinear model predictive control: an overview,” *Eur. J. Control* **9**(2-3), 179–195 (2003).
- [36] R. Ghaemi, J. Sun and I. V. Kolmanovsky, “An integrated perturbation analysis and sequential quadratic programming approach for model predictive control,” *Automatica* **45**(10), 2412–2418 (2009).
- [37] S. R. McReynolds, “The successive sweep method and dynamic programming,” *J. Math. Anal. Appl.* **19**(3), 565–598 (1967).
- [38] H. Schättler and U. Ledzewicz. *Geometric Optimal Control: Theory, Methods and Examples* (Springer, New York, (2012).