MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Constrained Spacecraft Attitude Control on SO(3) Using Fast Nonlinear Model Predictive Control

Gupta, R.; Kalabić, U.; Di Cairano, S.; Kolmanovsky, I.V.; Bloch, A.

TR2015-066 July 01, 2015

Abstract

In this paper, a fast solver for the optimization problem arising in the nonlinear model predictive control of spacecraft attitude is developed and simulation results of its application to constrained spacecraft attitude control are presented. The solver exploits the numerical solution of the necessary conditions for optimality in a discrete-time optimal control problem defined over a prediction horizon, where the discrete-time dynamics are based on the Lie group variational integrator model. The inequality constraints (thrust constraint, inclusion/exclusion zone constraints, etc.) are handled using a penalty function approach. Our developments exploit the geometric mechanics and control formalism.

American Control Conference (ACC)

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Constrained Spacecraft Attitude Control on SO(3) Using Fast Nonlinear Model Predictive Control

Rohit Gupta[1], Uroš V. Kalabić[1], Stefano Di Cairano[2⋆], Anthony M. Bloch[3] and Ilya V. Kolmanovsky[1]

***Abstract*— Recently, a reference governor and a nonlinear model predictive control approach for constrained spacecraft attitude control on SO(3) were proposed in [9]. In this paper, we develop a fast solver for the nonlinear model predictive controller in [9]. The approach is based on a numerical solution of the necessary conditions for optimality resulting from a discrete optimal control problem over each prediction horizon, where the inequality constraints are incorporated as soft constraints through a penalty function. The formulation of the nonlinear model predictive control problem and the necessary conditions for optimality exploit the geometric mechanics formalism and Lie group variational integrator based spacecraft attitude dynamics model. Simulation results along with the computational time assessment are presented. Remarks on integrating the approach with continuation methods are made.**

I. Introduction

Constrained spacecraft attitude control is receiving broader attention to address stringent performance requirements, in particular, in small spacecraft with tight control and inclusion/exclusion zone constraints. Model predictive control of spacecraft attitude based on linearized dynamics has been studied in previous literature (e.g., see [7], [19], [20]). To deal more effectively with system nonlinearities and perform global maneuvers, nonlinear predictive control approaches have been proposed in [6], [9]. Unlike [6], [9] utilizes discrete-time spacecraft dynamics based on a Lie group variational integrator model. This model provides higher accuracy in prediction and unlike continuous-time integrators, preserves the conserved quantities of motion (momentum and energy) to machine precision in absence of external moments (see [11]). As SO(3) is closed under multiplication, the Lie group variational integrator updates the attitude by multiplying two matrices in SO(3) and hence ensures that the attitude always evolves on SO(3). For a detailed presentation on variational integrators, see [14] and for the discrete rigid body equations, see [15].

The nonlinear model predictive control approach described in [9] uses a standard constrained optimizer in `MATLAB` (`fmincon.m`), which is our baseline solver. Since the baseline solver calculates the derivatives numerically and does not explicitly exploit the knowledge of the underlying Lie group structure, one single optimization run can take a very long time, which makes it impractical for real-time implementation. In this paper, we develop a fast solver based on solving the necessary conditions for optimality resulting from a discrete optimal control problem over each prediction horizon, where the inequality constraints are incorporated as soft constraints through a penalty function. The receding horizon optimal control problem and necessary conditions for optimality are formulated exploiting geometric mechanics formalism. The idea of using a penalty function is similar to the one used in [18], where barrier functions are used for constrained motion planning for multiple vehicles on SE(3). The necessary conditions for optimality are solved using the *Indirect Single Shooting Method*, where Newton's method is used to determine the initial values of the Lagrange multipliers, using sensitivity derivatives obtained from the necessary conditions for optimality (see [12]).

The paper is organized as follows. In Section II, we derive the necessary conditions for optimality. In Section III, we give a description of the fast solver. In Section IV, we present numerical results along with the computational time assessment. Finally, in Section V, we give some concluding remarks and possible future work. Before proceeding further, we describe the notation used in the paper.

A. Notation

The notation used in the paper is mostly standard, except in a few places. For a given control sequence $\{u_{k+j}\}_{j=0}^{N-1}$, resulting in a state sequence $\{x_{k+j}\}_{j=0}^{N}$ starting from x_k, over a prediction horizon of N-steps, the control sequence is denoted by $\{u_{k+j|k}\}_{j=0}^{N-1}$ and the state sequence is denoted by $\{x_{k+j|k}\}_{j=0}^{N}$. Using the above notation, $x_{k|k} = x_k$. For a matrix $B \in \mathbb{R}^{3\times3}$, the antisymmetric part is denoted by $(B)_A$ and the Frobenius norm is denoted by $\|B\|_F$. For a differentiable scalar valued function f, the partial derivative of f with respect to one of its arguments as X, is denoted by $\mathbf{D}_X f$.

We also use the following maps: $(.)^\times : \mathbb{R}^3 \to \mathfrak{so}(3)$, $(.)^{-\times} : \mathfrak{so}(3) \to \mathbb{R}^3$ and $(.)^\diamond : \mathbb{R}^3 \to \mathfrak{so}(3)^*$. The standard inner product on $\mathbb{R}^3$ is denoted by $\langle .,. \rangle$ and the natural pairing between $\mathfrak{so}(3)$ and $\mathfrak{so}(3)^*$ is denoted by $\langle\langle .,. \rangle\rangle$.

II. Nonlinear Model Predictive Control on SO(3)

Consider the following nonlinear model predictive control problem

$$\min_{\{u_{k+j|k}\}_{j=0}^{N-1}} \mathcal{J}_d = K_d(R_{k+N|k}, \Pi^\times_{k+N|k}) +$$

[1]Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA.

[2]Mitsubishi Electric Research Laboratories, Cambridge, MA, 02139.

⋆This work was not sponsored by Mitsubishi Electric Co. or any of its subsidiaries.

[3]Department of Mathematics, University of Michigan, Ann Arbor, MI 48109, USA.

$$\sum_{j=0}^{N-1} C_d(R_{k+j|k}, \Pi^{\times}_{k+j|k}, u^{\times}_{k+j|k}), \quad (1)$$

subject to

$$\begin{aligned} h\Pi^{\times}_{k+j|k} &= F_{k+j|k}J_d - J_dF^T_{k+j|k}, & (2)\\ R_{k+1+j|k} &= R_{k+j|k}F_{k+j|k}, & (3)\\ \Pi_{k+1+j|k} &= F^T_{k+j|k}\Pi_{k+j|k} + hu_{k+j|k}, & (4)\\ H_i(R_{k+j|k}, \Pi^{\times}_{k+j|k}, u^{\times}_{k+j|k}) &\leq 0,\ i = 0, \ldots, m, & (5) \end{aligned}$$

where $R_{k+j|k}$, $F_{k+j|k} \in \mathrm{SO}(3)$, $\Pi_{k+j|k}$, $u_{k+j|k} \in \mathbb{R}^3$ and $h \in \mathbb{R}_+$ is the time step. The terminal cost K_d is a real-valued positive semi-definite function with respect to its arguments $R_{k+N|k}$ and $\Pi^{\times}_{k+N|k}$, with the property that $K_d(I_{3\times3}, 0_{3\times3}) = 0$. The incremental cost C_d is a real-valued positive semi-definite function with respect to its arguments $R_{k+j|k}$ and $\Pi^{\times}_{k+j|k}$ and a real-valued positive definite function with respect to its argument $u^{\times}_{k+j|k}$, with the property that $C_d(I_{3\times3}, 0_{3\times3}, 0_{3\times3}) = 0$. It is also assumed that the data are sufficiently regular, so that when the inequality constraints are incorporated as soft constraints through a penalty function, the *Implicit Function Theorem* can be used to uniquely determine the optimal control $u^{\times *}_{k+j|k}$ as a function of $R_{k+j|k}$, $\Pi^{\times}_{k+j|k}$ and $\lambda^i_{k+j|k}$, where $\lambda^i_{k+j|k}$, $i \in \{1, 2\}$ are the Lagrange multipliers, which are introduced shortly (e.g., see [2]). The inequality constraints are given by $H_i : \mathrm{SO}(3) \times \mathfrak{so}(3) \times \mathfrak{so}(3) \to \mathbb{R}$. In addition to all the above properties, the terminal cost K_d, the incremental cost C_d and the inequality constraints H_i also satisfy the appropriate smoothness assumptions. In what follows, we also assume that the constraint qualification condition holds at the optimal solution for each prediction horizon.

Note that $J_d \in \mathbb{R}^{3\times3}$ is the nonstandard moment of inertia matrix and is related to the standard moment of inertia matrix $J \in \mathbb{R}^{3\times3}$ as $J_d = \frac{1}{2}\operatorname{tr}(J)I_{3\times3} - J$. To obtain the necessary conditions for optimality, we will follow the same variational approach as given in [12]. However, here this approach is applied to a more general problem with inequality constraints (5) and an additional dependence in the cost (1) on R and Π. Since the fast solver is based on solving the necessary conditions for optimality resulting from a discrete optimal control problem over each prediction horizon, where the inequality constraints are incorporated as soft constraints through a penalty function, we consider the following discrete optimal control problem

$$\min_{\{u_k\}_{k=0}^{N-1}} \mathcal{J}_d = K_d(R_N, \Pi^{\times}_N) + \sum_{k=0}^{N-1} C_d(R_k, \Pi^{\times}_k, u^{\times}_k), \quad (6)$$

subject to

$$\begin{aligned} h\Pi^{\times}_k &= F_kJ_d - J_dF^T_k, & (7)\\ R_{k+1} &= R_kF_k, & (8)\\ \Pi_{k+1} &= F^T_k\Pi_k + hu_k, & (9)\\ H_i(R_k, \Pi^{\times}_k, u^{\times}_k) &\leq 0,\ i = 0, \ldots, m. & (10) \end{aligned}$$

Define the augmented cost functional as follows

$$\begin{aligned} \mathcal{J}_{d,aug} &= K_d(R_N, \Pi^{\times}_N) + \sum_{k=0}^{N-1} C_d(R_k, \Pi^{\times}_k, u^{\times}_k) + \\ &\sum_{k=0}^{N-1} \langle\langle \lambda^1_k, \operatorname{logm}(R_{k+1}) - \operatorname{logm}(R_kF_k)\rangle\rangle + \\ &\sum_{k=0}^{N-1} \langle\langle \lambda^2_k, (\Pi_{k+1} - F^T_k\Pi_k - hu_k)^{\diamond}\rangle\rangle + \\ &\sum_{k=0}^{N-1}\sum_{i=0}^{m} \mu_i\phi_i(H_i(R_k, \Pi^{\times}_k, u^{\times}_k)), \quad (11) \end{aligned}$$

where $\lambda^1_k \in \mathfrak{so}(3)^*$, $\lambda^2_k \in \mathfrak{so}(3)$, $\phi_i(.)$ is a penalty function and $\mu_i \in \mathbb{R}_+$.

A. Remark I

The special orthogonal group is a Lie group given by $\mathrm{SO}(3, \mathbb{R}) = \{A \in \mathrm{M}(3, \mathbb{R}) \mid A^T = A^{-1}, \det(A) = 1\}$, which has the Lie algebra given by $\mathfrak{so}(3, \mathbb{R}) = \{A \in \mathrm{M}(3, \mathbb{R}) \mid A^T = -A\}$. The map $(.)^{\times} : \mathbb{R}^3 \to \mathfrak{so}(3)$ is a Lie algebra isomorphism between $\mathbb{R}^3$ and $\mathfrak{so}(3)$ (e.g., see [10]), which shows $\mathbb{R}^3 \cong \mathfrak{so}(3)$. Under the above isomorphism, the Killing form $\kappa(.,.)$ on $\mathfrak{so}(3)$ gets identified with the standard inner product on $\mathbb{R}^3$ (e.g., see [10]). Specifically, if $\kappa(a^{\times}, b^{\times}) := \operatorname{tr}(\operatorname{ad}(a^{\times}) \circ \operatorname{ad}(b^{\times}))$, where ad is the adjoint representation of $\mathfrak{so}(3)$, then $\kappa(a^{\times}, b^{\times}) = \operatorname{tr}(a^{\times}b^{\times}) = -2\langle a, b\rangle_{\mathbb{R}^3\times\mathbb{R}^3}$. In fact, as SO(3) is compact and semi-simple, the negative of the Killing form $\kappa(.,.)$ on $\mathfrak{so}(3)$ gives a bi-invariant Riemannian metric on SO(3). Using the map $(.)^{\diamond} : \mathbb{R}^3 \to \mathfrak{so}(3)^*$ and by letting the natural pairing $\langle\langle a^{\diamond}, b^{\times}\rangle\rangle_{\mathfrak{so}(3)^*\times\mathfrak{so}(3)} := \langle a, b\rangle_{\mathbb{R}^3\times\mathbb{R}^3}$, it is easily seen that $\langle\langle a^{\diamond}, b^{\times}\rangle\rangle_{\mathfrak{so}(3)^*\times\mathfrak{so}(3)} = -\frac{1}{2}\kappa(a^{\times}, b^{\times}) = \frac{1}{2}\operatorname{tr}((a^{\times})^Tb^{\times})$, which also shows that $\mathfrak{so}(3) \cong \mathfrak{so}(3)^*$ (e.g., see [8]). In this way, the natural pairing between a covector and a vector gets identified to the Killing form on $\mathfrak{so}(3)$, which further gets identified with the standard inner product on $\mathbb{R}^3$. Using this, we can obtain the necessary conditions for optimality in $\mathbb{R}^3$.

The variations for the sequences $\{R_k\}_{k=0}^N$, $\{F_k\}_{k=0}^{N-1}$ and $\{\Pi_k\}_{k=0}^N$ are given as follows

$$\begin{aligned} R_{k,\epsilon} &= R_k\exp(\epsilon\eta^{\times}_k), & (12)\\ F_{k,\epsilon} &= F_k\exp(\epsilon\xi^{\times}_k), & (13)\\ \Pi_{k,\epsilon} &= \Pi_k + \epsilon\delta\Pi_k, & (14) \end{aligned}$$

where η_k, $\xi_k \in \mathbb{R}^3$ and exp is the exponential map. Note that $\eta_0 = 0$, $\xi_0 = 0$ and $\delta\Pi_0 = 0$.

The infinitesimal variations of R_k, F_k and Π_k are given by

$$\begin{aligned} \delta R_k &= \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} R_{k,\epsilon}, \\ &= R_k\eta^{\times}_k, & (15) \end{aligned}$$

$$\begin{aligned}\delta F_k &= \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} F_{k,\epsilon},\\ &= F_k\xi_k^{\times}, \qquad (16)\\ \delta\Pi_k &= \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \Pi_{k,\epsilon},\\ &= \delta\Pi_k. \qquad (17)\end{aligned}$$

Before proceeding further, we will require a few facts. The variation of the discrete attitude update equation is used as a constrained variation instead of taking a variation of the matrix logarithm and yields the following fact.

Fact 1. ([12]) η_k and ξ_k in (12) and (13) satisfy, $\eta_{k+1} = F_k^T\eta_k + \xi_k$.

The variation of the implicit equation yields the following fact

Fact 2. ([12]) $\xi_k = \underbrace{hF_k^T(\mathrm{tr}(F_kJ_d)I_{3\times3} - F_kJ_d)^{-1}}_{M_k\in\mathbb{R}^{3\times3}}\delta\Pi_k$.

Also, $\forall a \in \mathbb{R}^3$, $\forall B \in \mathbb{R}^{3\times3}$, we have the following fact

Fact 3. ([8]) $\frac{1}{2}\,\mathrm{tr}(B^Ta^{\times}) = \langle((B)_A)^{-\times}, a\rangle$.

The above fact is used to obtain the following fact

Fact 4. ([8]) $\langle\langle \mathbf{D}_{R_k} f, R_k\eta_k^{\times}\rangle\rangle = \langle((R_k^T(\mathbf{D}_{R_k} f))_A)^{-\times}, \eta_k\rangle$,

where f is a differentiable scalar valued function, with one of its arguments as R_k. Using all of the above facts, the variation of the augmented cost functional can be written as follows

$$\begin{aligned}&\delta\mathcal{J}_{d,aug}\\ &= \langle((R_N^T(\mathbf{D}_{R_N}K_d))_A)^{-\times},\eta_N\rangle + \langle((\mathbf{D}_{\Pi_N^{\times}}K_d)_A)^{-\times},\\ &\quad \delta\Pi_N\rangle + \sum_{k=0}^{N-1}[\langle((R_k^T(\mathbf{D}_{R_k}C_d))_A)^{-\times},\eta_k\rangle +\\ &\quad \langle((\mathbf{D}_{\Pi_k^{\times}}C_d)_A)^{-\times},\delta\Pi_k\rangle + \langle((\mathbf{D}_{u_k^{\times}}C_d)_A)^{-\times},\delta u_k\rangle] +\\ &\quad \sum_{k=0}^{N-1}\langle\lambda_k^1,\eta_{k+1} - F_k^T\eta_k - \xi_k\rangle + \sum_{k=0}^{N-1}\langle\lambda_k^2,\delta\Pi_{k+1} -\\ &\quad (F_k\xi_k^{\times})^T\Pi_k - F_k^T\delta\Pi_k - h\delta u_k\rangle +\\ &\quad \sum_{k=0}^{N-1}\sum_{i=0}^{m}\mu_i[\langle((R_k^T(\mathbf{D}_{R_k}(\phi_i\circ H_i)))_A)^{-\times},\eta_k\rangle +\\ &\quad \langle((\mathbf{D}_{\Pi_k^{\times}}(\phi_i\circ H_i))_A)^{-\times},\delta\Pi_k\rangle +\\ &\quad \langle((\mathbf{D}_{u_k^{\times}}(\phi_i\circ H_i))_A)^{-\times},\delta u_k\rangle],\\ &= \langle((R_N^T(\mathbf{D}_{R_N}K_d))_A)^{-\times},\eta_N\rangle + \langle((\mathbf{D}_{\Pi_N^{\times}}K_d)_A)^{-\times},\\ &\quad \delta\Pi_N\rangle + \sum_{k=0}^{N-1}[\langle((R_k^T(\mathbf{D}_{R_k}C_d))_A)^{-\times},\eta_k\rangle +\\ &\quad \langle((\mathbf{D}_{\Pi_k^{\times}}C_d)_A)^{-\times},\delta\Pi_k\rangle + \langle((\mathbf{D}_{u_k^{\times}}C_d)_A)^{-\times},\delta u_k\rangle] +\\ &\quad \sum_{k=0}^{N-1}\langle\lambda_k^1,\eta_{k+1} - F_k^T\eta_k - \xi_k\rangle + \sum_{k=0}^{N-1}\langle\lambda_k^2,\delta\Pi_{k+1} +\\ &\quad ((F_k^T\Pi_k)^{\times})^T\xi_k - F_k^T\delta\Pi_k - h\delta u_k\rangle +\\ &\quad \sum_{k=0}^{N-1}\sum_{i=0}^{m}\mu_i[\langle((R_k^T(\mathbf{D}_{R_k}(\phi_i\circ H_i)))_A)^{-\times},\eta_k\rangle +\\ &\quad \langle((\mathbf{D}_{\Pi_k^{\times}}(\phi_i\circ H_i))_A)^{-\times},\delta\Pi_k\rangle +\\ &\quad \langle((\mathbf{D}_{u_k^{\times}}(\phi_i\circ H_i))_A)^{-\times},\delta u_k\rangle],\\ &= \langle((R_N^T(\mathbf{D}_{R_N}K_d))_A)^{-\times} + \lambda_{N-1}^1,\eta_N\rangle +\\ &\quad \sum_{k=1}^{N-1}[\langle -F_k\lambda_k^1 + \lambda_{k-1}^1 + ((R_k^T(\mathbf{D}_{R_k}C_d))_A)^{-\times} +\\ &\quad \sum_{i=0}^{m}\mu_i((R_k^T(\mathbf{D}_{R_k}(\phi_i\circ H_i)))_A)^{-\times},\eta_k\rangle] +\\ &\quad \langle((\mathbf{D}_{\Pi_N^{\times}}K_d)_A)^{-\times} + \lambda_{N-1}^2,\delta\Pi_N\rangle +\\ &\quad \sum_{k=1}^{N-1}[\langle -M_k^T\lambda_k^1 - (F_k - M_k^T(F_k^T\Pi_k)^{\times})\lambda_k^2 + \lambda_{k-1}^2 +\\ &\quad ((\mathbf{D}_{\Pi_k^{\times}}C_d)_A)^{-\times} + \sum_{i=0}^{m}\mu_i((\mathbf{D}_{\Pi_k^{\times}}(\phi_i\circ H_i))_A)^{-\times},\\ &\quad \delta\Pi_k\rangle] + \sum_{k=1}^{N-1}[\langle -h\lambda_k^2 + ((\mathbf{D}_{u_k^{\times}}C_d)_A)^{-\times} +\\ &\quad \sum_{i=0}^{m}\mu_i((\mathbf{D}_{u_k^{\times}}(\phi_i\circ H_i))_A)^{-\times},\delta u_k\rangle], \qquad (18)\end{aligned}$$

where the analogue of integration by parts in the discrete setting is used along with the fact that the variations η_k, $\delta\Pi_k$ vanish at $k = 0$ and the variation δu_k vanishes at $k = 0$, N. Since $\delta\mathcal{J}_{d,aug}$ should vanish for all η_k, $\delta\Pi_k$ and δu_k, the necessary conditions for optimality are as follows

$$\begin{aligned}h\Pi_k^{\times} &= F_kJ_d - J_dF_k^T, \qquad (19)\\ R_{k+1} &= R_kF_k, \qquad (20)\\ \Pi_{k+1} &= F_k^T\Pi_k + hu_k, \qquad (21)\\ \lambda_{k+1}^1 &= F_{k+1}^T[\lambda_k^1 + ((R_{k+1}^T(\mathbf{D}_{R_{k+1}}C_d))_A)^{-\times} +\\ &\quad \sum_{i=0}^{m}\mu_i((R_{k+1}^T(\mathbf{D}_{R_{k+1}}(\phi_i\circ H_i)))_A)^{-\times}], \qquad (22)\\ \lambda_{N-1}^1 &= -((R_N^T(\mathbf{D}_{R_N}K_d))_A)^{-\times}, \qquad (23)\\ \lambda_{k+1}^2 &= (F_{k+1} - M_{k+1}^T(F_{k+1}^T\Pi_{k+1})^{\times})^{-1}\\ &\quad [-M_{k+1}^T\lambda_{k+1}^1 + \lambda_k^2 + ((\mathbf{D}_{\Pi_{k+1}^{\times}}C_d)_A)^{-\times} +\\ &\quad \sum_{i=0}^{m}\mu_i((\mathbf{D}_{\Pi_{k+1}^{\times}}(\phi_i\circ H_i))_A)^{-\times}], \qquad (24)\\ \lambda_{N-1}^2 &= -((\mathbf{D}_{\Pi_N^{\times}}K_d)_A)^{-\times}, \qquad (25)\\ h\lambda_k^2 &= ((\mathbf{D}_{u_k^{\times}}C_d)_A)^{-\times} +\\ &\quad \sum_{i=0}^{m}\mu_i((\mathbf{D}_{u_k^{\times}}(\phi_i\circ H_i))_A)^{-\times}. \qquad (26)\end{aligned}$$

B. Cost and Inequality Constraints

We consider a quadratic-type cost, similar to the one used in [9] and [17]. Specifically, the terminal cost, the incremental cost and the inequality constraints are given as

follows

$$K_d = \frac{1}{2}\|P_1^{1/2}(R_N - I_{3\times 3})\|_F^2 + \frac{1}{2}\|P_2^{1/2}\Pi_N^{\times}\|_F^2, \quad (27)$$

$$C_d = \frac{h}{2}\|Q_1^{1/2}(R_k - I_{3\times 3})\|_F^2 + \frac{h}{2}\|Q_2^{1/2}\Pi_k^{\times}\|_F^2 + \frac{h}{2}\|Q_3^{1/2}u_k^{\times}\|_F^2, \quad (28)$$

$$H_0 = \frac{1}{2}\|u_k^{\times}\|_F^2 - \alpha, \quad (29)$$

$$H_i = \beta_i - v_i^T R_k^T w_i, \ \ i = 1, \ldots, m, \quad (30)$$

where P_1, P_2, Q_1, Q_2 are positive semi-definite symmetric matrices, Q_3 is a positive definite symmetric matrix, $\alpha \in \mathbb{R}_+$, $\beta_i \in \mathbb{R}$, v_i is the spacecraft body-fixed vector and w_i is the inertial direction vector. Note that for any $a \in \mathbb{R}^3$ and for any positive semi-definite/definite symmetric matrix B, $\frac{1}{2}\|B^{1/2}a^{\times}\|_F^2 = \frac{1}{2}a^T\tilde{B}a$, where $\tilde{B} = \mathrm{tr}(B)I_{3\times 3} - B$ (e.g., see [9]).

For the specific form of the terminal cost and the incremental cost defined above, the necessary conditions for optimality are as follows

$$h\Pi_k^{\times} = F_k J_d - J_d F_k^T, \quad (31)$$

$$R_{k+1} = R_k F_k, \quad (32)$$

$$\Pi_{k+1} = F_k^T \Pi_k + h u_k, \quad (33)$$

$$\lambda_{k+1}^1 = F_{k+1}^T[\lambda_k^1 - h((R_{k+1}^T Q_1)_A)^{-\times} - \sum_{i=1}^{m} \mu_i((R_{k+1}^T(\mathbf{D}_{H_i}(\phi_i \circ H_i)w_i v_i^T))_A)^{-\times}], \quad (34)$$

$$\lambda_{N-1}^1 = ((R_N^T P_1)_A)^{-\times}, \quad (35)$$

$$\lambda_{k+1}^2 = (F_{k+1} - M_{k+1}^T(F_{k+1}^T\Pi_{k+1})^{\times})^{-1} [-M_{k+1}^T\lambda_{k+1}^1 + \lambda_k^2 + h((Q_2\Pi_{k+1}^{\times})_A)^{-\times}], \quad (36)$$

$$\lambda_{N-1}^2 = -((P_2\Pi_N^{\times})_A)^{-\times}, \quad (37)$$

$$h\lambda_k^2 = h((Q_3 u_k^{\times})_A)^{-\times} + \mu_0((\mathbf{D}_{H_0}(\phi_0 \circ H_0)u_k^{\times})_A)^{-\times}, \quad (38)$$

where we have chosen the differentiable penalty function, $\phi_i \circ H_i = h\max\{0, H_i\}^2$.

C. Remark II

The trajectories for $(R_k, \Pi_k, \lambda_k^1, \lambda_k^2)$ (starting from a given $(R_0, \Pi_0, \lambda_0^1, \lambda_0^2)$), using the necessary conditions for optimality are computed in the same way, as presented in [12]. Also, for the specific form of the terminal cost and the incremental cost defined above, the necessary conditions for optimality are for the case of a *Regulator Problem*, which can be easily extended to the case of a *Tracking Problem*.

III. Description of the Fast Solver

The necessary conditions for optimality obtained in the previous section give a two-point boundary value problem and are solved using the *Indirect Single Shooting Method*, where Newton's method is used to determine the initial values of the Lagrange multipliers, using sensitivity derivatives obtained from the necessary conditions for optimality (see [12]). We will follow the same procedure as given in [12]. However, here we exploit Newton's method instead of a backtracking line search method.

The sensitivity derivatives for the attitude and angular momentum equations are given as follows

$$\begin{bmatrix}\eta_{k+1}\\ \delta\Pi_{k+1}\end{bmatrix} = \begin{bmatrix}F_k^T & M_k\\ 0_{3\times 3} & F_k^T + (F_k^T\Pi_k)^{\times}M_k\end{bmatrix}\begin{bmatrix}\eta_k\\ \delta\Pi_k\end{bmatrix} + \begin{bmatrix}0_{3\times 3}\\ hI_{3\times 3}\end{bmatrix}\delta u_k. \quad (39)$$

The sensitivity derivatives for the Lagrange multiplier equations are given as follows

$$\begin{bmatrix}\delta\lambda_{k+1}^1\\ \delta\lambda_{k+1}^2\end{bmatrix} = S_k\begin{bmatrix}\eta_{k+1}\\ \delta\Pi_{k+1}\\ \delta\lambda_k^1\\ \delta\lambda_k^2\end{bmatrix}, \quad (40)$$

where $S_k \in \mathbb{R}^{6\times 12}$. Together (39) and (40) can be written as follows

$$\begin{bmatrix}\eta_{k+1}\\ \delta\Pi_{k+1}\\ \delta\lambda_{k+1}^1\\ \delta\lambda_{k+1}^2\end{bmatrix} = T_k\begin{bmatrix}\eta_k\\ \delta\Pi_k\\ \delta\lambda_k^1\\ \delta\lambda_k^2\end{bmatrix}, \quad (41)$$

where $T_k \in \mathbb{R}^{12\times 12}$. From (41)

$$\begin{bmatrix}\eta_N\\ \delta\Pi_N\\ \delta\lambda_N^1\\ \delta\lambda_N^2\end{bmatrix} = \left(\prod_{k=0}^{N-1} T_k\right)\begin{bmatrix}\eta_0\\ \delta\Pi_0\\ \delta\lambda_0^1\\ \delta\lambda_0^2\end{bmatrix}. \quad (42)$$

In the *Indirect Single Shooting Method*, the initial values of the Lagrange multipliers are taken as unknowns and the necessary conditions for optimality obtained in the previous section result in a root finding problem. We employ Newton's method to solve this root finding problem expecting quadratic rate of convergence (locally). Note that the baseline solver (`fsolve.m`) takes a long time, when used with *Indirect Single Shooting Method* as it does not utilize the knowledge of the underlying Lie group structure to calculate sensitivity derivatives and calculates the derivatives numerically. The updates have the following form

$$\lambda_0^{(i+1)} = \lambda_0^{(i)} - \gamma\left[\frac{\delta E^{(i)}}{\delta\lambda_0^{(i)}}\right]^{-1}E^{(i)}, \quad (43)$$

where the superscripts represent the iteration number, $\gamma \in (0, 1]$ is the step size and $E^{(i)}$ is given as follows

$$E^{(i)} = \begin{bmatrix}\lambda_{N-1}^{1(i)} - ((R_N^{T(i)}P_1)_A)^{-\times}\\ \lambda_{N-1}^{2(i)} + ((P_2\Pi_N^{\times(i)})_A)^{-\times}\end{bmatrix}. \quad (44)$$

Note that $E^{(i)}$ represents the error in satisfaction of the terminal boundary conditions at the *i-th* iteration. The sensitivity derivative for $E^{(i)}$ is given as follows

$$\delta E^{(i)} = \begin{bmatrix}\delta\lambda_{N-1}^{1(i)} + ((\eta_N^{\times(i)}R_N^{T(i)}P_1)_A)^{-\times}\\ \delta\lambda_{N-1}^{2(i)} + ((P_2\delta\Pi_N^{\times(i)})_A)^{-\times}\end{bmatrix}. \quad (45)$$

For a given $\delta\lambda_0^{(i)}$, $\delta E^{(i)}$ is obtained using (42). In this way, we obtain the *Jacobian Matrix* in Newton's method, which is used in the *Indirect Single Shooting Method*. Once the optimal initial values of the Lagrange multipliers are obtained, the optimal trajectories can be calculated using the necessary conditions for optimality obtained in the previous section.

A. Remark III

Continuation methods (e.g., see [1]) can be exploited to obtain additional time savings. There are two scenarios where continuation methods can be used. The first scenario occurs for the model predictive control problem over a fixed prediction horizon, when the weighting factor multiplying the penalty function is being increased. Generally, starting with a very high value of the weighting factor is not recommended as this might result in numerical ill-conditioning. Continuation with respect to the weighting factor can be used to obtain a desired solution quickly and avoids the problem of numerical ill-conditioning. The second scenario occurs when going from one prediction horizon to the next, wherein the initial state of the model predictive control problem undergoes a change. If the states change by a small amount, then this change can be thought of as a small perturbation. Continuation with respect to the states along with the solution computed in the previous prediction horizon can be used to predict a desired solution quickly. The idea of continuation presented here is similar to the one presented in [3], [4], [5], [16], [21]. While we do not formally take advantage of continuation methods in this paper, our subsequent numerical examples warm-start the fast solver with the previous solution.

IV. Numerical Results

We consider a spacecraft with moment of inertia matrix $J = \mathrm{diag}(1, 0.8, 0.8)\ kg\text{-}m^2$ and the time step $h = 0.4\ sec$. We take $P_1 = P_2 = Q_1 = Q_2 = 0.01 I_{3\times 3}$ and $Q_3 = I_{3\times 3}$, in (27) and (28).

In subsequent figures, the attitude maneuver is plotted on the 2-sphere, S^2, where the vector $[x\ y\ z]^T$ corresponding to the first column of R_0 is plotted in dashed-red, the second column of R_0 is plotted in dashed-green and the third column of R_0 is plotted in dashed-blue. Similarly, the vector $[x\ y\ z]^T$ corresponding to the first column of R_N is plotted in red, the second column of R_N is plotted in green and the third column of R_N is plotted in blue. For all other R_k, $k \neq 0$, N, only the corresponding co-ordinates are shown in the corresponding colors described above.

A. Simulation with Thrust Constraint

In this simulation, we only consider the thrust constraint, which is given by (29), with $\alpha = 0.0001\ N\text{-}m$. The simulation time is $150\ sec$, the prediction horizon is $2\ sec$, the weighting factor $\mu_0 = 10^{10}$ and the step size $\gamma = 1$. The initial condition for the attitude and angular momentum are given as follows

$$\begin{aligned} R_0 &= \exp(\zeta^\times), \\ \Pi_0 &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T, \end{aligned}$$

where $\zeta = \begin{bmatrix} 0.25 & 0.5 & 0.5 \end{bmatrix}^T$.

Figure 1 shows the trajectory of the angular momentum obtained using the fast solver. Figure 2 shows the trajectory of the control input obtained using the fast solver. Figure 3 shows the time history of the 2-norm of the control input obtained using the fast solver. Figure 4 shows the attitude maneuver obtained using the fast solver. Figure 5 shows the trajectory of the angular momentum obtained using the baseline solver. Figure 6 shows the trajectory of the control input obtained using the baseline solver. Figure 7 shows the time history of the 2-norm of the control input obtained using the baseline solver. Figure 8 shows the attitude maneuver obtained using the baseline solver. It can be seen from Figures 1-8 that the solution obtained by the fast solver is very close to the solution obtained by the baseline solver.

B. Simulation with Thrust and Exclusion Zone Constraints

In this simulation, we consider a thrust constraint and one exclusion zone constraint, which are given by (29) and (30), with $\alpha = 0.0001\ N\text{-}m$, $\beta_1 = -0.9962$, $v_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$ and $w_1 = -\begin{bmatrix} 0.9276 & 0.3736 & 0 \end{bmatrix}^T$. The simulation time is $300\ sec$, the prediction horizon is $2\ sec$, the weighting factors $\mu_0 = 10^{10}$, $\mu_1 = 10^3$ and a hybrid step size method is used. The initial condition for the attitude and angular momentum are given as follows

$$\begin{aligned} R_0 &= \exp(\zeta^\times), \\ \Pi_0 &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T, \end{aligned}$$

where $\zeta = \begin{bmatrix} 0 & 0 & 0.5 \end{bmatrix}^T$.

Figure 9 shows the trajectory of the angular momentum obtained using the fast solver. Figure 10 shows the trajectory of the control input obtained using the fast solver. Figure 11 shows the time history of the 2-norm of the control input obtained using the fast solver. Figure 12 shows the time history of the exclusion zone constraint obtained using the fast solver. Figure 13 shows the attitude maneuver obtained using the fast solver.

Table I compares the total time for the fast solver and the baseline solver, on a 3.6 GHz Intel Xeon desktop computer with 16 GB of RAM. In Table I, under the heading Case, I refers to the simulation with thrust constraint and II refers to the simulation with thrust and exclusion zone constraints. This comparison demonstrates the time savings with the fast solver versus the baseline solver. For Case I, the maximum time taken by the fast solver to obtain the optimal solution for one prediction horizon is approximately $0.44\ sec$. For Case II, the maximum time taken by the fast solver to obtain the optimal solution for one prediction horizon is approximately $4.19\ sec$. Figures showing the results for Case II for the baseline solver have not been provided because of space limitation. Also, note that the code has been implemented using

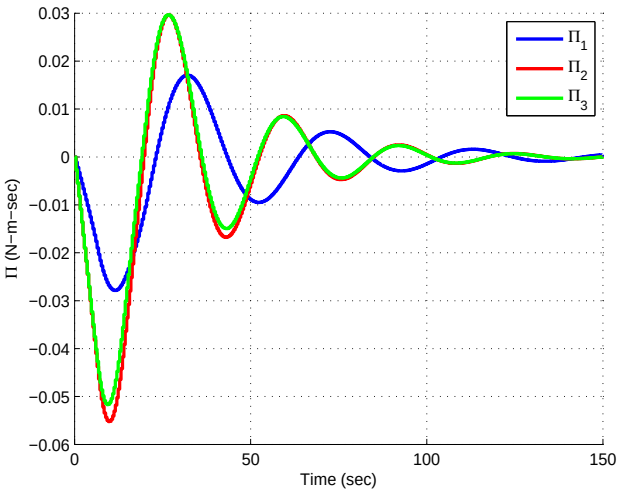

Fig. 1. Angular Momentum (Fast Solver).

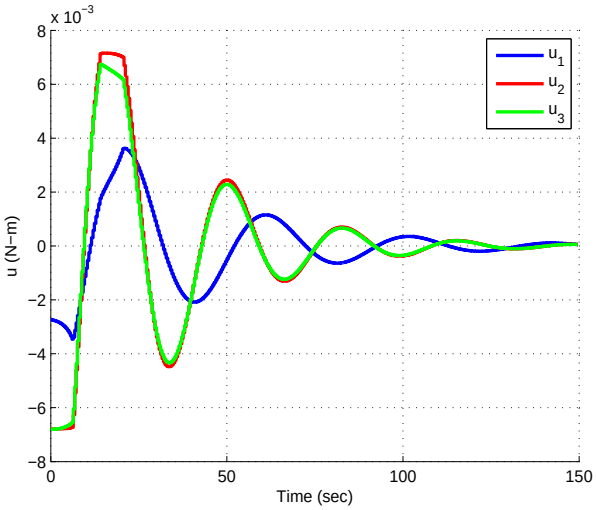

Fig. 2. Control Input (Fast Solver).

a `MATLAB` `m-file` and the computational time assessment has been done using the `tic-toc` function in `MATLAB`.

TABLE I
TOTAL TIME FOR BOTH THE SOLVERS

Case	Fast Solver	Baseline Solver
I	38.22 sec (approx.)	271.01 sec (approx.)
II	134.66 sec (approx.)	767.97 sec (approx.)

V. CONCLUSIONS AND FUTURE WORK

In this paper, we developed a fast solver for constrained spacecraft attitude control on $\mathrm{SO}(3)$ using the nonlinear model predictive control approach. Simulation results along with the computational time assessment were presented. Comparison with other solvers will be pursued in the future work. However, these solvers, with possible exception of [6], do not directly apply to the $\mathrm{SO}(3)$ setting. Extending the nonlinear model predictive control approach to mechanical systems evolving on other types of Lie groups, e.g., $\mathrm{SE}(3) \cong \mathrm{SO}(3) \ltimes \mathbb{R}^3$ (e.g., see [8]), etc. and the integration with continuation methods will be pursued in future work.

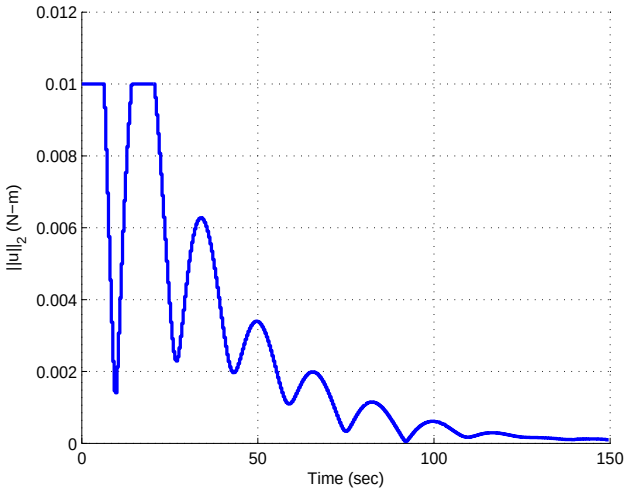

Fig. 3. 2-Norm of the Control Input (Fast Solver).

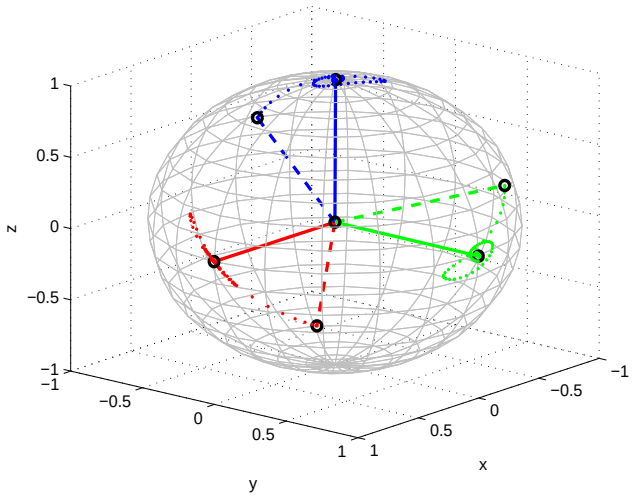

Fig. 4. Attitude Maneuver (Fast Solver).

ACKNOWLEDGMENTS

The research of Anthony M. Bloch was supported by NSF grants DMS-1207693 and INSPIRE-1343720. The research of Rohit Gupta, Uroš V. Kalabić and Ilya V. Kolmanovsky was supported by NSF grants CMMI-1130160 and ECCS-1404814.

REFERENCES

[1] E.L. Allgower and K. Georg, *Numerical Continuation Methods: An Introduction*, Berlin: Springer Series in Computational Mathematics, Springer-Verlag, 1990.

[2] A.M. Bloch, *Nonholonomic Mechanics and Control*, New York: Interdisciplinary Applied Mathematics, Springer-Verlag, 2003.

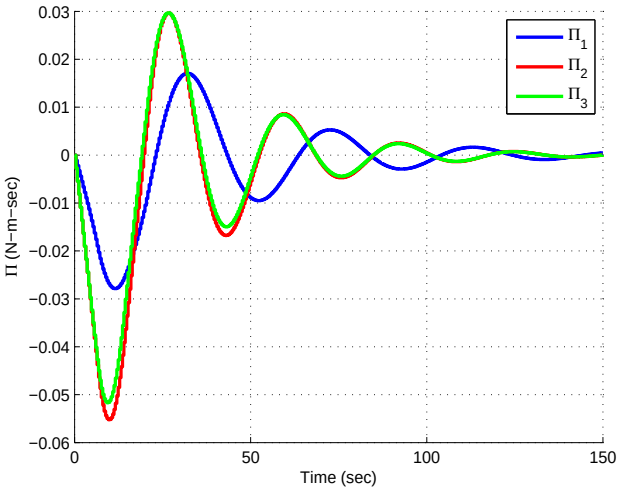

Fig. 5. Angular Momentum (Baseline Solver).

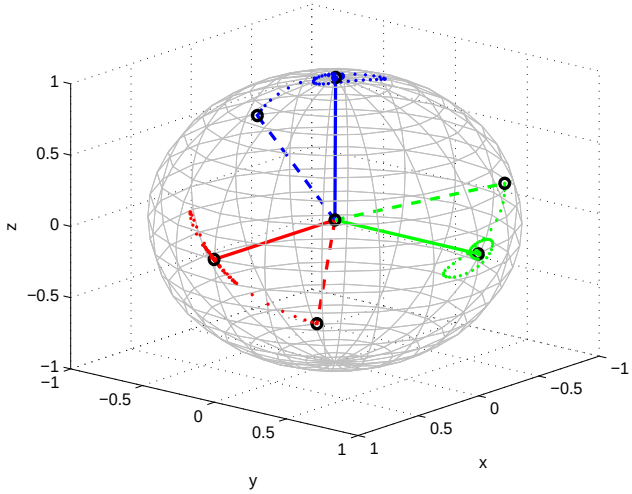

Fig. 8. Attitude Maneuver (Baseline Solver).

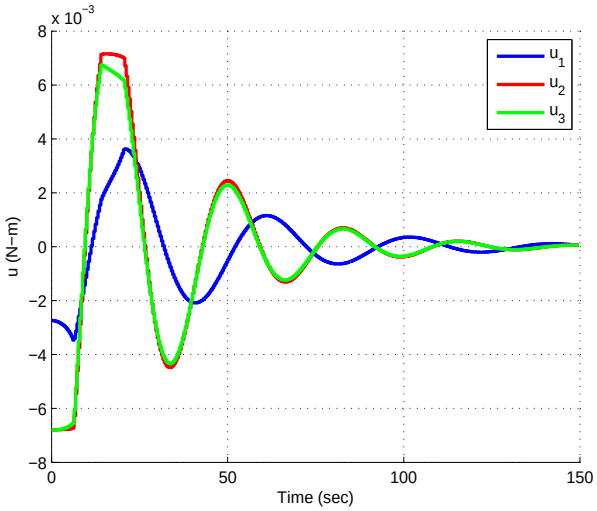

Fig. 6. Control Input (Baseline Solver).

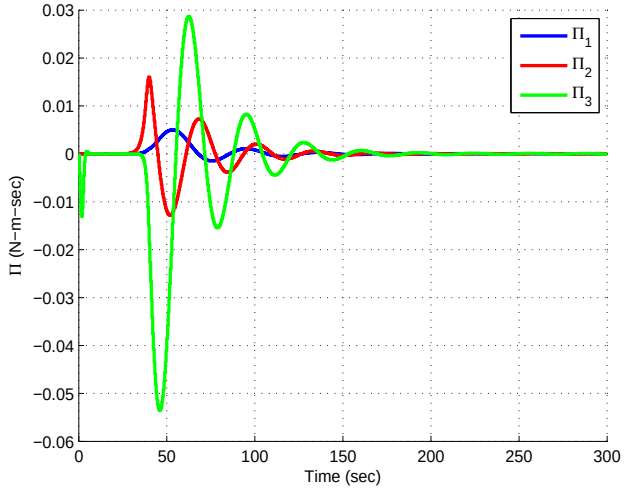

Fig. 9. Angular Momentum (Fast Solver).

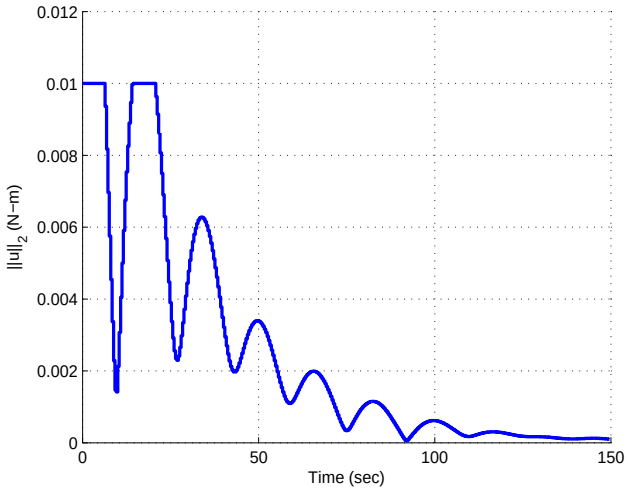

Fig. 7. 2-Norm of the Control Input (Baseline Solver).

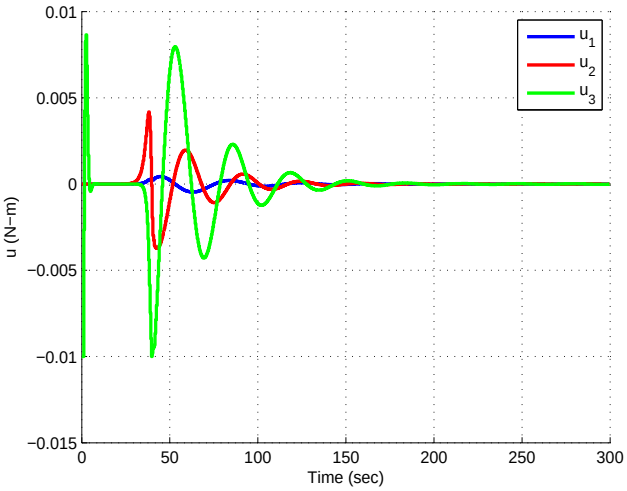

Fig. 10. Control Input (Fast Solver).

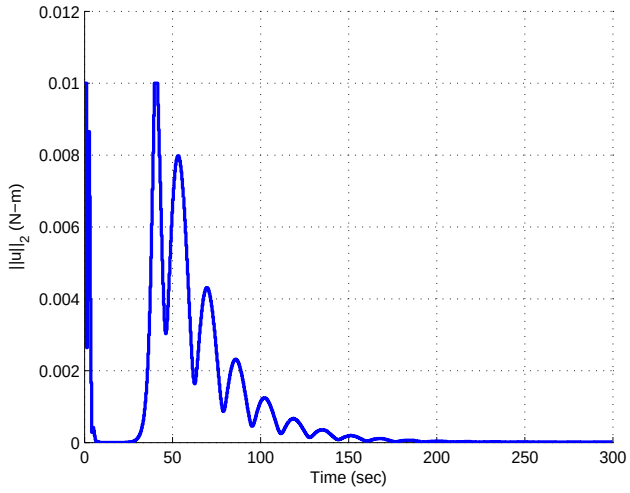

Fig. 11. 2-Norm of the Control Input (Fast Solver).

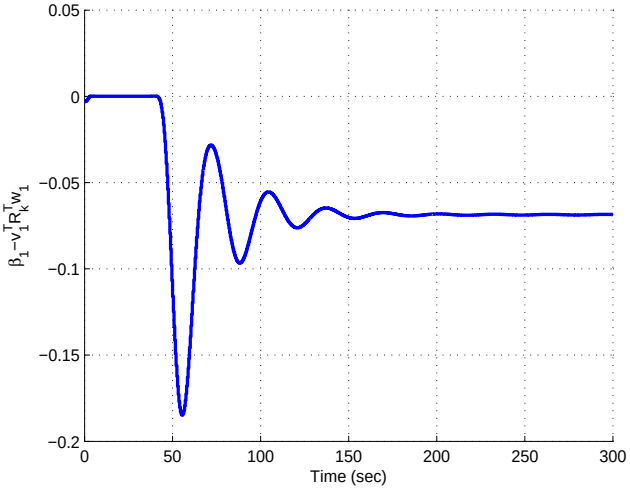

Fig. 12. Exclusion Zone Constraint (Fast Solver).

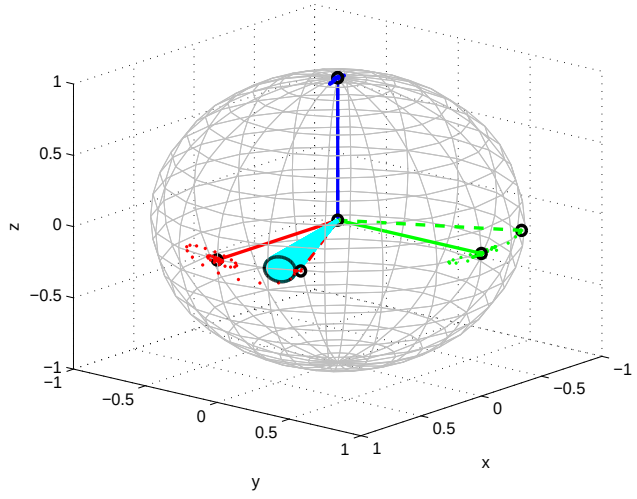

Fig. 13. Attitude Maneuver (Fast Solver).

[3] M. Diehl, H.G. Bock and J.P. Schlöder, "A Real-Time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control", *SIAM Journal on Control and Optimization*, Vol. 43, No. 5, pp. 1714-1736, March 2005.

[4] M. Diehl, R. Findeisen, F. Allgöwer, H.G. Bock and J.P. Schlöder, "Nominal Stability of Real-Time Iteration Scheme for Nonlinear Model Predictive Control", *IEE Proceedings-Control Theory and Applications*, Vol. 152, No. 3, pp. 296-308, May 2005.

[5] R. Ghaemi, J. Sun and I.V. Kolmanovsky, "An Integrated Perturbation Analysis and Sequential Quadratic Programming Approach for Model Predictive Control", *Automatica*, Vol. 45, No. 10, pp. 2412-2418, October 2009.

[6] S. Gros, M. Zanon, M. Vukov and M. Diehl, "Nonlinear MPC and MHE for Mechanical Multi-Body Systems with Application to Fast Tethered Airplanes", *Proceedings of Nonlinear Model Predictive Control Conference*, pp. 86-93, Noordwijkerhout, Netherlands, 2012.

[7] Ø. Hegrenæs, J.T. Gravdahl and P. Tøndel, "Spacecraft Attitude Control Using Explicit Model Predictive Control", *Automatica*, Vol. 41, No. 12, pp. 2107-2114, December 2005.

[8] D.D. Holm, T. Schmah and C. Stoica, *Geometric Mechanics and Symmetry: From Finite to Infinite Dimensions*, Oxford Texts in Applied and Engineering Mathematics, 2009.

[9] U.V. Kalabić, R. Gupta, A.M. Bloch, S. Di Cairano and I.V. Kolmanovsky, "Constrained Spacecraft Attitude Control on SO(3) Using Reference Governors and Nonlinear Model Predictive Control", *Proceedings of 2014 American Control Conference*, pp. 5586-5593, Portland, OR, June 2014.

[10] A.W. Knapp, *Lie Groups: Beyond an Introduction (Second Edition)*, Birkhäuser, 2002.

[11] 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", *Proceedings of 2005 IEEE Conference on Control Applications*, pp. 962-967, Toronto, Canada, August 2005.

[12] T. Lee, M. Leok and N.H. McClamroch, "Optimal Attitude Control of a Rigid Body Using Geometrically Exact Computations on SO(3)", *Journal of Dynamical and Control Systems*, Vol. 14, No. 4, pp. 465-487, October 2008.

[13] T. Lee, "Computational Geometric Mechanics and Control of Rigid Bodies", *PhD Dissertation*, Department of Aerospace Engineering, University of Michigan, Ann Arbor, 2008.

[14] J.E. Marsden and M. West, "Discrete Mechanics and Variational Integrators", *Acta Numerica*, Vol. 10, pp. 357-514, May 2001.

[15] J. Moser and A.P. Veselov, "Discrete Versions of Some Classical Integrable Systems and Factorization of Matrix Polynomials", *Communications in Mathematical Physics*, Vol. 139, No. 2, pp. 217-243, August 1991.

[16] T. Ohtsuka, "A Continuation/GMRES Method for Fast Computation of Nonlinear Receding Horizon Control", *Automatica*, Vol. 40, No. 4, pp. 563-574, April 2004.

[17] A. Saccon, J. Hauser and A.P. Aguiar, "Exploration of Kinematic Optimal Control on the Lie Group SO(3)", *8th IFAC Symposium on Nonlinear Control Systems*, pp. 1302-1307, Bologna, Italy, September 2010.

[18] A. Saccon, A.P. Aguiar, A.J. Häusler, J. Hauser and A.M. Pascoal, "Constrained motion planning for multiple vehicles on SE(3)", *Proceedings of 51st IEEE Conference on Decision and Control*, pp. 5637-5642, Maui, Hawaii, December 2012.

[19] E. Silani and M. Lovera, "Magnetic Spacecraft Attitude Control: A Survey and Some New Results", *Control Engineering Practice*, Vol. 13, No. 3, pp. 357-371, March 2005.

[20] J. Vandersteen, M. Diehl, C. Aerts and J. Swevers, "Spacecraft Attitude Estimation and Sensor Calibration Using Moving Horizon Estimation", *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 3, pp. 734-741, May 2013.

[21] V.M. Zavala and L.T. Biegler, "The Advanced-Step NMPC Controller: Optimality, Stability and Robustness", *Automatica*, Vol. 45, No. 1, pp. 86-93, January 2009.