Variable sampling-time nonlinear model predictive control of satellites using magneto-torquers

Yi Cao[a] & Wen-Hua Chen[b]

[a] School of Energy, Environment and Agrifood, Cranfield University, Cranfield, Bedford MK43 0AL, UK

[b] Department of Aeronautical Automotive Engineering, Loughborough University, Leicestershire LE11 2SG, UK
Accepted author version posted online: 26 Aug 2014.Published online: 30 Sep 2014.

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis
Taylor & Francis Group

Variable sampling-time nonlinear model predictive control of satellites using magneto-torquers

Yi Cao[a*] and Wen-Hua Chen[b]

[a]School of Energy, Environment and Agrifood, Cranfield University, Cranfield, Bedford MK43 0AL, UK; [b]Department of Aeronautical Automotive Engineering, Loughborough University, Leicestershire LE11 2SG, UK

Satellite control using magneto-torquers represents a control challenge combined with strong nonlinearity, variable dynamics and partial controllability. An automatic differentiation-based nonlinear model predictive control (NMPC) algorithm is developed in this work to tackle these issues. Based on the previously developed formulation of NMPC, a novel variable sampling-time scheme is proposed to provide a better trade-off between transient control performance and closed-loop stability. More specifically, a small sampling time is adopted to improve the response speed when the satellite is far away from the desired position, and a large sampling time is employed for the closed-loop stability when the satellite is around its equilibrium position. This scheme also significantly reduces the online computational burden associated with fixed sampling-time NMPC where a large prediction horizon has to be adopted in order to the ensure closed-loop stability. The proposed approach is demonstrated through nonlinear simulation of a specific satellite case with satisfactory results obtained.

Keywords: predictive control; nonlinear systems; satellite; magneto-torquer; automatic differentiation; variable sampling time

1. Introduction

Using magnetic rods as the only actuators for spacecraft attitude control has been attracting a lot of attention in recent years due to its simplicity, low cost and power efficiency (Wood and Chen, 2013). This technique is especially suitable for low-Earth orbit satellites with modest control performance requirements. The control concept is based on that interaction between the magnetic moment generated within a spacecraft and the magnetic field of the Earth produces a torque which can be used to control the attitude of the spacecraft. For satellites whose attitude is controlled by chemicals or other thrusters, by exploring the usage of magnetic torquers, it could significantly reduce the consumption of chemicals so to extend the life of the satellites.

Satellite attitude control using magneto-torque actuators has a number of challenges. The dynamic system is strongly nonlinear, particularly when the satellite is in the stage of orbit installation. The magneto-torque actuator can only provide partial controllability to the satellite at any moment. The dynamics of the system are time varying and the response time can vary from seconds to several hours. Traditional proportional-integral-derivative control cannot provide satisfactory performance. In recent years, attempts have been made in applying model predictive control (MPC) to satellite attitude control, for example, Silani and Lovera (2005), Wood, Chen, and Fertin (2006), and

Wood and Chen (2008). However, the linear model-based MPC can only work for a specified attitude with relatively small actuations. On the other hand, although the minimum time control of satellites using magneto-torquers has been studied using optimal control principles, stability of the system when satellite approaching a desired attitude has not been analysed (Liang, Fullmer, and Chen, 2004). In this work, a novel varying sampling-time nonlinear model predictive control (NMPC) solution using automatic differentiation (AD) is proposed to tackle the challenging problem of the satellite attitude control using magneto-torquers for orbit installation or altering attitude.

AD is a technique to automatically generate derivatives through computer programming. It does not like numerical calculations, where derivatives are approximated through finite differences, hence are inaccurate and very inefficient. It also avoids code growth issues normally relating to symbolic computations. In the recent work (Cao, 2005), a new NMPC scheme has been developed using AD techniques. Using this approach, high-order Taylor coefficients of states and outputs can be automatically generated at each sampling instant, so that future trajectory can be predicted efficiently and accurately. Meanwhile, AD can also produce sensitivities of Taylor coefficients against control signals in a very efficient way. This makes the online optimization problem to be solved much more quickly. An improvement of one to two orders of mag-

---

nitude in computation speed has been observed in case studies (Al-Seyab and Cao, 2008a, 2008b; Cao, 2005).

In this work, the satellite attitude model with magneto-torque actuator has been converted into an iterative Taylor model based on AD principles. Using the iterative Taylor model, the satellite attitude system can be simulated much more efficiently and also much more accurately than traditional ordinary differential equation solvers. The Taylor model also enables a continuously variable sampling-time scheme to be implemented with the NMPC. That is, at each control interval, future behaviour of the system is predicted through iteratively calculating high-order Taylor coefficients. An error estimation approach is then adopted to calculate the maximum time interval, which makes the dynamic response prediction within a specified error tolerant range. Based on this, a fraction of the maximum time interval is used as the length of the next control interval. In this way, the NMPC can automatically adjust sampling time from the initial quick move to alter attitude to final slow move when the satellite approaches the desired attitude. This scheme works very well with the satellite attitude control system, where a fixed sampling rate may result in either a very slow and very poor response when the sampling time is too large, or an enormous computation load and potential stability problems when sampling time is too short.

The rest of this paper is organized as follows. In Section 2, a mathematical model describing the dynamics of a satellite equipped with magneto-torque actuators is presented. Then, in Section 3, algorithms of AD-based NMPC with variable sampling time are to be developed. Section 4 presents a case study with specific satellite parameters and control performance is evaluated through nonlinear simulation. This work is concluded in Section 5.

2. Satellite with magneto-torquer actuators

The model of a satellite equipped with magneto-torquers can be described in various reference frames (Wertz, 1978). In this work, the reference system described by Lovera and Astolfi (2004) is adopted. That is the Earth-centred inertial reference axes plus satellite body axes. The attitude dynamics can be represented by the well-know Euler's equations (Wertz, 1978), whilst the attitude kinematics are described by the Euler quaternions. Therefore, the complete dynamics model of the system is given as follows:

$$I_x \dot{\omega}_x = \omega_y \omega_z (I_y - I_z) + T_x, \tag{1}$$

$$I_y \dot{\omega}_y = \omega_z \omega_x (I_z - I_x) + T_y, \tag{2}$$

$$I_z \dot{\omega}_z = \omega_x \omega_y (I_x - I_y) + T_z, \tag{3}$$

$$\dot{q}_1 = \frac{(\omega_z q_2 - \omega_y q_3 + \omega_x q_4)}{2}, \tag{4}$$

$$\dot{q}_2 = \frac{(\omega_x q_3 - \omega_z q_1 + \omega_y q_4)}{2}, \tag{5}$$

$$\dot{q}_3 = \frac{(\omega_y q_1 - \omega_x q_2 + \omega_z q_4)}{2}, \tag{6}$$

$$\dot{q}_4 = \frac{(-\omega_x q_1 - \omega_y q_2 - \omega_z q_3)}{2}, \tag{7}$$

where ω_x, ω_y and ω_z are spacecraft angular rates expressed in the body frame, I_i and T_i, for $i = x, y, z$ are the inertial components and external toques in the satellite body axes, respectively, whilst $q_i, i = 1, 2, 3, 4$ are the Euler quaternions, which satisfy the constraint:

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1. \tag{8}$$

Note, the quaternion constraint (8) is implicitly represented by Equation (7). The magneto-torquers are modelled as follows:

$$T_x = b_3 u_2 - b_2 u_3, \tag{9}$$

$$T_y = b_1 u_3 - b_3 u_1, \tag{10}$$

$$T_z = b_2 u_1 - b_1 u_2, \tag{11}$$

where $u_i, i = 1, 2, 3$ are control inputs (magnetic rod dipole moments), whilst $b_i, i = 1, 2, 3$ are local magnetic field components in the body frame converted from the orbit frame:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = M_q \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}. \tag{12}$$

The conversion matrix, M_q is given as

$$M_q = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1 q_2 + q_3 q_4) & 2(q_1 q_3 - q_2 q_4) \\ 2(q_1 q_2 - q_3 q_4) & 1 - 2(q_1^2 + q_3^2) & 2(q_2 q_3 + q_1 q_4) \\ 2(q_1 q_3 + q_2 q_4) & 2(q_2 q_3 - q_1 q_4) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}. \tag{13}$$

The magnetic field components can be approximated as follows:

$$b_x = \mu_{fa} \cos(\omega_0 t) \sin(\alpha), \tag{14}$$

$$b_y = -\mu_{fa} \cos(\alpha), \tag{15}$$

$$b_z = 2\mu_{fa} \sin(\omega_0 t) \sin(\alpha), \tag{16}$$

where $\mu_{fa} = \mu_f / a_e$ with μ_f being the Earth dipole strength and a_e the orbit radius, ω_0 the orbit frequency and α the orbit inclination angle.

The above satellite system with magneto-torquers represents a difficult control challenge due the following reasons:

- The system is strongly nonlinear and time variant because of the varying local magnetic field strength along the orbit.
- The system is only partially controllable at any time instance. This is because at any time instance, the

torques generated by the magnetic actuators have only two independent effective directions.

- Depending on actuation strength and satellite angular rate, the system dynamic response can be from very fast (time constant in few milliseconds) to very slow (time constant in few hours).

Most previous works focused on the first two issues. Linear and nonlinear solutions to this problem have been summarized by Silani and Lovera (2005). However, the issue of variable response speed has not been considered in these studies so that all these solutions, even with nonlinear formulations (Lovera and Astolfi, 2004), resulted in very slow dynamic response. The only exception is the work reported by Liang et al. (2004), where a time optimal control was developed to drive the satellite from a given initial attitude to quickly reach the desired attitude. However, in that work, the stability of system at the desired attitude has not been considered so that the approach cannot be directly applied to a real satellite system. In the following section, an AD-based NMPC approach is to be developed to control the satellite attitude. The advantage of AD makes the nonlinear model predictive controller work at a variable sampling-time mode so that both response speed and stability issues can be handled satisfactorily.

3. NMPC with variable sampling time

An AD-based NMPC formulation was proposed by one of the authors (Cao, 2005). It is based on the recursive algorithms to automatically generate a high-order Taylor expansion of a wide class of nonlinear functions. Such recursive formulations of commonly used functions are given in the appendix. AD tools, such as ADOL-C (Griewank, Juedes, and Utke, 1996) available on public domain, can be directly used to generate Taylor coefficients and associated sensitivities. In this work, this formulation is expended to variable sampling time. Firstly, for tutorial purpose, analytic NMPC formulations are derived in Sections 3.1–3.3. Then, the variable sampling-time extension is developed in Section 3.4.

3.1. Integration using high-order Taylor expansion

Let $\omega_i(t)$, $i = x, y, z$ and $q_i(t)$, $i = 1, 2, 3, 4$ be approximated by the truncated Taylor series around t_0:

$$\omega_i(t) = \sum_{k=0}^{d} \omega_i^{[k]}(t - t_0)^k, \quad i = x, y, z, \tag{17}$$

$$q_i(t) = \sum_{k=0}^{d} q_i^{[k]}(t - t_0)^k, \quad i = 1, 2, 3, 4. \tag{18}$$

The first coefficients in these series can be determined from the initial values, i.e. $\omega_i^{[0]} = \omega_i(t_0)$, $i = x, y, z$ and

$q_i^{[0]} = q_i(t_0)$, $i = 1, 2, 3, 4$. According to the automatic differential theory (Rall, 1981), all high-order coefficients can be recursively determined. The recursive equations are derived from the differential equations (1)–(7) using the rules provided by Rall (1981)

$$\omega_x^{[k+1]} = \frac{I_y - I_z}{(k+1)I_x} \left(\sum_{j=0}^{k} \omega_y^{[j]}\omega_z^{[k-j]} + \frac{b_3^{[k]}u_2 - b_2^{[k]}u_3}{I_y - I_z} \right), \tag{19}$$

$$\omega_y^{[k+1]} = \frac{I_z - I_x}{(k+1)I_y} \left(\sum_{j=0}^{k} \omega_z^{[j]}\omega_x^{[k-j]} + \frac{b_1^{[k]}u_3 - b_3^{[k]}u_1}{I_z - I_x} \right), \tag{20}$$

$$\omega_z^{[k+1]} = \frac{I_x - I_y}{(k+1)I_z} \left(\sum_{j=0}^{k} \omega_x^{[j]}\omega_y^{[k-j]} + \frac{b_2^{[k]}u_1 - b_1^{[k]}u_2}{I_x - I_y} \right), \tag{21}$$

$$q_1^{[k+1]} = \frac{1}{2(k+1)}$$
$$\sum_{j=0}^{k} \left(\omega_z^{[j]}q_2^{[k-j]} - \omega_y^{[j]}q_3^{[k-j]} + \omega_x^{[j]}q_4^{[k-j]} \right), \tag{22}$$

$$q_2^{[k+1]} = \frac{1}{2(k+1)}$$
$$\sum_{j=0}^{k} \left(\omega_x^{[j]}q_3^{[k-j]} - \omega_z^{[j]}q_1^{[k-j]} + \omega_y^{[j]}q_4^{[k-j]} \right), \tag{23}$$

$$q_3^{[k+1]} = \frac{1}{2(k+1)}$$
$$\sum_{j=0}^{k} \left(\omega_y^{[j]}q_1^{[k-j]} - \omega_x^{[j]}q_2^{[k-j]} + \omega_z^{[j]}q_4^{[k-j]} \right), \tag{24}$$

$$q_4^{[k+1]} = \frac{-1}{2(k+1)}$$
$$\sum_{j=0}^{k} \left(\omega_x^{[j]}q_1^{[k-j]} + \omega_y^{[j]}q_2^{[k-j]} + \omega_z^{[j]}q_3^{[k-j]} \right). \tag{25}$$

The Taylor coefficients of b_i, $i = 1, 2, 3$ are derived from Equation (12)

$$\begin{bmatrix} b_1^{[k]} \\ b_2^{[k]} \\ b_3^{[k]} \end{bmatrix} = \sum_{j=0}^{k} \begin{bmatrix} m_{11}^{[j]} & m_{12}^{[j]} & m_{13}^{[j]} \\ m_{21}^{[j]} & m_{22}^{[j]} & m_{23}^{[j]} \\ m_{31}^{[j]} & m_{32}^{[j]} & m_{33}^{[j]} \end{bmatrix} \begin{bmatrix} b_x^{[k-j]} \\ b_y^{[k-j]} \\ b_z^{[k-j]} \end{bmatrix}, \tag{26}$$

where

$$m_{11}^{[k]} = \sum_{j=0}^{k} (q_1^{[j]}q_1^{[k-j]} - q_2^{[j]}q_2^{[k-j]} - q_3^{[j]}q_3^{[k-j]} + q_4^{[j]}q_4^{[k-j]}),$$

$$m_{12}^{[k]} = 2\sum_{j=0}^{k}(q_1^{[j]}q_2^{[k-j]} + q_3^{[j]}q_4^{[k-j]}),$$

$$m_{13}^{[k]} = 2\sum_{j=0}^{k}(q_1^{[j]}q_3^{[k-j]} - q_2^{[j]}q_4^{[k-j]}),$$

$$m_{21}^{[k]} = 2\sum_{j=0}^{k}(q_1^{[j]}q_2^{[k-j]} - q_3^{[j]}q_4^{[k-j]}),$$

$$m_{22}^{[k]} = \sum_{j=0}^{k}(-q_1^{[j]}q_1^{[k-j]} + q_2^{[j]}q_2^{[k-j]} \\ - q_3^{[j]}q_3^{[k-j]} + q_4^{[j]}q_4^{[k-j]}),$$

$$m_{23}^{[k]} = 2\sum_{j=0}^{k}(q_2^{[j]}q_3^{[k-j]} + q_1^{[j]}q_4^{[k-j]}),$$

$$m_{31}^{[k]} = 2\sum_{j=0}^{k}(q_1^{[j]}q_3^{[k-j]} + q_2^{[j]}q_4^{[k-j]}),$$

$$m_{32}^{[k]} = 2\sum_{j=0}^{k}(q_2^{[j]}q_3^{[k-j]} - q_1^{[j]}q_4^{[k-j]}),$$

$$m_{33}^{[k]} = \sum_{j=0}^{k}(-q_1^{[j]}q_1^{[k-j]} - q_2^{[j]}q_2^{[k-j]} \\ + q_3^{[j]}q_3^{[k-j]} + q_4^{[j]}q_4^{[k-j]}).$$

The Taylor coefficients of b_i, $i = x, y, z$ are obtained recursively according to Equations (14)–(16) as follows:

$$v_1^{[0]} = \cos(\omega_0 t_0), \tag{27}$$

$$v_2^{[0]} = \sin(\omega_0 t_0), \tag{28}$$

$$v_1^{[k+1]} = \frac{-\omega_0}{k+1}v_2^{[k]}, \quad k \geq 0, \tag{29}$$

$$v_2^{[k+1]} = \frac{\omega_0}{k+1}v_1^{[k]}, \quad k \geq 0, \tag{30}$$

$$b_y^{[0]} = -\mu_{\mathrm{fa}}\cos(\alpha), \tag{31}$$

$$b_y^{[k]} = 0, k > 0, \tag{32}$$

$$b_x^{[k]} = \mu_{\mathrm{fa}}\sin(\alpha)v_1^{[k]}, \tag{33}$$

$$b_z^{[k]} = 2\mu_{\mathrm{fa}}\sin(\alpha)v_2^{[k]}. \tag{34}$$

Equations (17)–(34) provide a complete solution to the initial value problem of the satellite differential equation system. Let $t_{i+1} = t_i + h_i$, $i = 0, \ldots, n$ and $t_f = t_{n+1}$. Then, initial values at t_i can be iteratively calculated using Equations (17) and (18) until the specified final time, t_f reaches. The order of Taylor series is determined by the integration step and the tolerance specified (Cao, 2005).

Note, in the above formulation, the actuating inputs are treated as constant. In MPC, these inputs are piecewise constant. Therefore, whenever an input changes its value, a new integration step should start.

The advantage of this integration approach against other numerical integration methods is its efficiency. It is able to provide a high-accuracy solution within a very short time. For the satellite differential equation system, one particular problem is that the algebraic constraint (8) cannot be satisfactorily agreed with solutions provided by traditional differential equation solvers. This problem is easily overcome by the high-order Taylor series-based approach as indicated in Section 4.

3.2. Sensitivity

The NMPC problem presented in Section 3.3 requires to solve a nonlinear least-square problem in real time, where the computation efficiency is a critical issue to make the control approach practically applicable. Experience shows that more than 90% of computation time is associated with derivative calculation in solving such an NMPC problem (Cao, 2005). Therefore, to accelerate the computation speed of the NMPC, an efficient algorithm to solve the sensitivity equations associated with the differential equations (1)–(7) is desirable. Based on the high-order Taylor coefficients derived above to solve the differential equations, the associated sensitivity problem can also be solved by deriving the corresponding Taylor coefficients of the sensitivity variables as follows.

Let $s = [\omega_x \ \omega_y \ \omega_z \ q_1 \ q_2 \ q_3 \ q_4]^{\mathrm{T}}$. For Equations (1)–(7), two sensitivity matrices are defined as follows:

$$A_s^{[k]} := \begin{bmatrix} A_{\omega\omega}^{[k]} & A_{\omega q}^{[k]} \\ A_{q\omega}^{[k]} & A_{qq}^{[k]} \end{bmatrix}, \quad A_u^{[k]} := \begin{bmatrix} A_{\omega u}^{[k]} \\ 0 \end{bmatrix}, \tag{35}$$

where

$$A_{\omega\omega}^{[k]} := \begin{bmatrix} 0 & \omega_z^{[k]}\dfrac{I_y - I_z}{I_x} & \omega_y^{[k]}\dfrac{I_y - I_z}{I_x} \\ \omega_z^{[k]}\dfrac{I_z - I_x}{I_y} & 0 & \omega_x^{[k]}\dfrac{I_z - I_x}{I_y} \\ \omega_y^{[k]}\dfrac{I_x - I_y}{I_z} & \omega_x^{[0]}\dfrac{I_x - I_y}{I_z} & 0 \end{bmatrix},$$

$$A_{\omega q}^{[k]} := \begin{bmatrix} 0 & \dfrac{-u_3}{I_x} & \dfrac{u_2}{I_x} \\ \dfrac{u_3}{I_y} & 0 & \dfrac{-u_1}{I_y} \\ \dfrac{-u_2}{I_z} & \dfrac{u_1}{I_y} & 0 \end{bmatrix} \begin{bmatrix} \beta_{11}^{[k]} & \beta_{12}^{[k]} & \beta_{13}^{[k]} & \beta_{14}^{[k]} \\ \beta_{21}^{[k]} & \beta_{22}^{[k]} & \beta_{23}^{[k]} & \beta_{24}^{[k]} \\ \beta_{31}^{[k]} & \beta_{32}^{[k]} & \beta_{33}^{[k]} & \beta_{34}^{[k]} \end{bmatrix},$$

$$A_{q\omega}^{[k]} := \frac{1}{2}\begin{bmatrix} q_4^{[k]} & -q_3^{[k]} & q_2^{[k]} \\ q_3^{[k]} & q_4^{[k]} & -q_1^{[k]} \\ -q_2^{[k]} & q_1^{[k]} & q_4^{[k]} \\ -q_1^{[k]} & -q_2^{[k]} & -q_3^{[k]} \end{bmatrix},$$

$$A_{qq}^{[k]} := \frac{1}{2}\begin{bmatrix} 0 & \omega_z^{[k]} & -\omega_y^{[k]} & \omega_x^{[k]} \\ -\omega_z^{[k]} & 0 & \omega_x^{[k]} & \omega_y^{[k]} \\ \omega_y^{[k]} & -\omega_x^{[k]} & 0 & \omega_z^{[k]} \\ -\omega_x^{[k]} & \omega_y^{[k]} & \omega_z^{[k]} & 0 \end{bmatrix},$$

$$A_{\omega u}^{[k]} := \begin{bmatrix} 0 & \dfrac{b_3^{[k]}}{I_x} & -\dfrac{b_2^{[k]}}{I_x} \\ -\dfrac{b_3^{[k]}}{I_y} & 0 & \dfrac{b_1^{[k]}}{I_y} \\ \dfrac{b_2^{[k]}}{I_z} & -\dfrac{b_1^{[k]}}{I_z} & 0 \end{bmatrix},$$

$$\beta_{11}^{[k]} := 2\left(q_2^{[k]}b_y + \sum_{j=0}^{k}(q_1^{[j]}b_x^{[k-j]} + q_3^{[j]}b_z^{[k-j]})\right),$$

$$\beta_{12}^{[k]} := 2\left(q_1^{[k]}b_y - \sum_{j=0}^{k}(q_2^{[j]}b_x^{[k-j]} + q_4^{[j]}b_z^{[k-j]})\right),$$

$$\beta_{13}^{[k]} := 2\left(q_4^{[k]}b_y - \sum_{j=0}^{k}(q_3^{[j]}b_x^{[k-j]} - q_1^{[j]}b_z^{[k-j]})\right),$$

$$\beta_{14}^{[k]} := 2\left(q_3^{[k]}b_y + \sum_{j=0}^{k}(q_4^{[j]}b_x^{[k-j]} - q_2^{[j]}b_z^{[k-j]})\right),$$

$$\beta_{21}^{[k]} := 2\left(-q_1^{[k]}b_y + \sum_{j=0}^{k}(q_2^{[j]}b_x^{[k-j]} + q_4^{[j]}b_z^{[k-j]})\right),$$

$$\beta_{22}^{[k]} := 2\left(q_2^{[k]}b_y + \sum_{j=0}^{k}(q_1^{[j]}b_x^{[k-j]} + q_3^{[j]}b_z^{[k-j]})\right),$$

$$\beta_{23}^{[k]} := 2\left(-q_3^{[k]}b_y - \sum_{j=0}^{k}(q_4^{[j]}b_x^{[k-j]} - q_2^{[j]}b_z^{[k-j]})\right)$$

$$\beta_{24}^{[k]} := 2\left(q_4^{[k]}b_y - \sum_{j=0}^{k}(q_3^{[j]}b_x^{[k-j]} - q_1^{[j]}b_z^{[k-j]})\right),$$

$$\beta_{31}^{[k]} := 2\left(-q_4^{[k]}b_y + \sum_{j=0}^{k}(q_3^{[j]}b_x^{[k-j]} - q_1^{[j]}b_z^{[k-j]})\right),$$

$$\beta_{32}^{[k]} := 2\left(q_3^{[k]}b_y + \sum_{j=0}^{k}(q_4^{[j]}b_x^{[k-j]} - q_2^{[j]}b_z^{[k-j]})\right),$$

$$\beta_{33}^{[k]} := 2\left(q_2^{[k]}b_y + \sum_{j=0}^{k}(q_1^{[j]}b_x^{[k-j]} + q_3^{[j]}b_z^{[k-j]})\right),$$

$$\beta_{34}^{[k]} := 2\left(-q_1^{[k]}b_y + \sum_{j=0}^{k}(q_2^{[j]}b_x^{[k-j]} + q_4^{[j]}b_z^{[k-j]})\right).$$

The sensitivity Taylor coefficients are derived as follows:

$$B_s^{[k+1]} := \frac{ds^{[k+1]}}{ds^{[0]}} = \frac{1}{k+1}\sum_{j=0}^{k}A_s^{[j]}B_x^{[k-j]}, \tag{36}$$

$$B_u^{[k+1]} := \frac{ds^{[k+1]}}{du^{[0]}} = \frac{1}{k+1}\left(A_u^{[k]} + \sum_{j=0}^{k}A_s^{[j]}B_u^{[k-j]}\right). \tag{37}$$

Therefore,

$$B_s(t_0 + h_0) := \frac{ds(t_0 + h_0)}{ds(t_0)} = \sum_{k=0}^{d}B_s^{[k]}h_0^k, \tag{38}$$

$$B_u(t_0 + h_0) := \frac{ds(t_0 + h_0)}{du(t_0)} = \sum_{k=0}^{d}B_s^{[k]}h_0^k. \tag{39}$$

At the time, $t_i = t_{i-1} + h_{i-1}$, $i = 1, \ldots, n$, the sensitivity can be obtained using the chain-rule iteratively

$$B_{s_j}(t_i) := \frac{ds(t_i)}{ds(t_j)} = \prod_{k=0}^{i-(j+1)}B_s(t_{i-k}), \quad j < i, \tag{40}$$

$$B_{u_j}(t_i) := \frac{ds(t_i)}{du(t_j)} = \prod_{k=0}^{i-(j+2)}B_s(t_{i-k})B_u(t_{j+1}), \quad j < i. \tag{41}$$

3.3. Predictive control

The control performance of the satellite system is measured by a quadratic integration as follows:

$$\phi := \frac{1}{2}\sum_{k=0}^{n}\int_{t_k}^{t_k+h_k}((s - s_0)^{\mathrm{T}}Q(s - s_0) + u^{\mathrm{T}}Ru)\,dt, \tag{42}$$

where s_0 is the desired reference vector of s, Q and R are performance weights of states and inputs, respectively. Let ϕ_k represent the performance measure from t_k to $t_k + h_k$ and

$$E_k := \begin{bmatrix} F_k^{1/2}\begin{bmatrix} Q^{1/2}(s^{[0]} - s_0) \\ Q^{1/2}s^{[1]} \\ \vdots \\ Q^{1/2}s^{[d]} \\ R^{1/2}u(t_k) \end{bmatrix} \end{bmatrix}, \tag{43}$$

where the (i,j)th element of F_k is $h_k^{i+j-1}/(i+j-1)$. Then, it can be proven that $\phi_k = \frac{1}{2}E_k^{\mathrm{T}}E_k$. Stack all E_k together as $E = [E_0^{\mathrm{T}} \cdots E_n^{\mathrm{T}}]^{\mathrm{T}}$ then $\phi = \frac{1}{2}E^{\mathrm{T}}E$. This is a standard nonlinear least-square problem. To solve the problem, the Jacobian matrix, J of E against $u(t_k)$, $k = 0, \ldots, n$, is required. Partition J into J_{ij}, $i, j = 1, \ldots, n+$

1, where J_{ij} is the Jacobian matrix of E_{i-1} against u_{j-1}. Therefore,

$$
J_{ij} = \begin{cases}
0, & j > i \\[2mm]
\left[F_i^{1/2} \begin{bmatrix} 0 \\ Q^{1/2}B_u^{[1]} \\ \vdots \\ Q^{1/2}B_u^{[d]} \end{bmatrix} \\ R^{1/2} \end{bmatrix} \right], & j = i, \\[10mm]
\left[F_i^{1/2} \begin{bmatrix} Q^{1/2}B_{u_j}(t_i) \\ Q^{1/2}B_s^{[1]}B_{u_j}(t_i) \\ \vdots \\ Q^{1/2}B_s^{[d]}B_{u_j}(t_i) \end{bmatrix} \\ 0 \end{bmatrix} \right], & j < i.
\end{cases}
$$

Remark 1 Stability analysis for MPC of constrained nonlinear dynamics system has been established by adding an appropriate terminal weighting in the performance index for example see Chen and Cao (2012). However, the added terminal term complicates the control tuning as it is very hard to choose in particular for a nonlinear system and consequently understand its influence on the system performance. The satellite with magneto actuation system is partially controllable, where the uncontrollable subspace is periodically varying. In order to ensure stability, the prediction horizon has to be long enough to cover such uncontrollable subspace variation. Nevertheless, the time period of variation of controllable and uncontrollable subspaces depends on the velocity of the satellite. Therefore, it is particularly difficult to choose an appropriate terminal term for such a system. To focus on the variable sampling-time scheme, the cost function used in this paper, similar to classic MPC, has no terminal weights. Stability is ensured by tuning MPC parameters. It remains as an open problem to perform stability analysis for this kind of nonlinear MPC and only very limited results exist in the literature (Chen, 2010).

Based on the Levenberg–Marquardt algorithm (Marquardt, 1963), using the Jacobian, J, and the performance vector, E, the inputs, $U := [u^{\mathrm{T}}(t_0)\ u^{\mathrm{T}}(t_1)\ \cdots\ u^{\mathrm{T}}(t_n)]^{\mathrm{T}}$ is iteratively updated as follows:

$$
U_{k+1} = U_k - (J^{\mathrm{T}}J + \lambda I)^{-1}J^{\mathrm{T}}E, \tag{44}
$$

where λ is determined by the algorithm to make sure the performance measure is reduced at each iteration step. The optimal solution, U^* is obtained when the algorithm is converged, whilst only the first block of U^*, i.e. $u^*(t_0)$ is applied to the system until the next sampling instance.

Note that in the above configuration, it is assumed that there is no any state constraint. Standard algorithm is readily available to deal with input constraints, e.g. by using the `lsqnonlin` function in MATLAB Optimization Toolbox (MathWorks, 2007).

3.4. *Variable sampling time*

Performance of discrete control systems is limited by the sampling time. Fast response requires a short sampling time. However, on the other hand, to ensure closed-loop stability of MPC, the predictive period should be long enough. Combining both requirements may result in an impractical large prediction horizon (number of predictive steps) so that computation load of NMPC is not tractable. The situation is even worse in the satellite control system, where the dynamic response speed depends on the magnitude of actuating inputs. The rotating speed under the maximum actuating force can be three to four orders of magnitude higher than that under the minimum force. On the other hand, the partially controllable nature and the time-varying magnetic field along the earth orbit make the stabilization of the system only achievable over sufficient long periods (Lovera and Astolfi, 2004). Therefore, a novel variable sampling-time scheme is proposed in this work to tackle this problem.

Naturally, when the satellite is far away from its desired equilibrium position, a relatively large actuating force is required to drive the satellite to the desired position as quickly as possible. In this situation, control performance, i.e. response speed is the main concern. Hence, a relatively small sampling time should be adopted. As the satellite gradually approaches the desired position, the required magnitude of actuating force is reduced and the stability issue becomes more and more important. In general, for a performance index as given in Equation (42), the stability of the MPC algorithms can be achieved by employing a sufficiently large horizon. Therefore, the sampling time should increase so that within the tractable computation time closed-loop stability can be ensured. Both requirements are satisfied by adjusting the sampling time to match the integration steps, which is determined based on the error tolerance control algorithm described as follows.

Assume that the state of the satellite at the next sampling time is estimated by the Taylor series as $s(h) = \sum_{k=0}^{d} s^{[k]} + \varepsilon(h, d)$, which has the radius of convergence equal to r. Then,

$$
\varepsilon(h, d) \approx C \left(\frac{h}{r} \right)^{d+1}, \tag{45}
$$

where C is constant. For sufficient large d,

$$
r \approx r_d := \frac{\|s^{[d-1]}\|_\infty}{\|s^{[d]}\|_\infty}. \tag{46}
$$

Since, $\varepsilon(h, d-1) \approx \varepsilon(h, d)(r_d/h) \approx \varepsilon(h, d) + \|s^{[d]}\|_\infty h^k$, it leads to the following estimation of the truncation error:

$$
\varepsilon(h, d) = \frac{h^{d+1}\|s^{[d]}\|_\infty^2}{\|s^{[d-1]}\|_\infty - h\|s^{[d]}\|_\infty}. \tag{47}
$$

Therefore, for a given d and a specified error tolerance, ε_0, the integration step (sampling time) can be estimated from

$\varepsilon(h, d) \leq \varepsilon_0$. For $h > 1$, it leads to

$$h \leq \left(\frac{\varepsilon_0 \|s^{[d-1]}\|_\infty}{\|s^{[d]}\|_\infty^2} \right)^{1/(d+1)}. \qquad (48)$$

When the satellite position is far away from its desired attitude and a large actuating force is imposed, a small sampling time is determined by applying the above estimation. Therefore, the control system has fast response speed. As the satellite approaches the desired attitude, actuation from the MPC is small. The above estimation will result in a large sampling time and hence a long predictive horizon, which then ensures the closed-loop stability.

4. Case study and simulation

The developed AD-based NMPC with variable sampling-time algorithm is implemented in MATLAB and tested with a specific satellite case, which has the following parameters: $I_x = 128$, $I_y = 600$ and $I_z = 500$ all in [kg m²]; $\alpha = 50°$, $\omega_0 = 2\pi/5400$ (rad/s) and $\mu_f = 0.0632$ based on orbit height 500 (km). The maximum dipole moment of each magnetic rod is 400 (A m²).

The NMPC is configured with two sets of parameters in Equation (42) depending on the angle velocity: for fast rotating speed, $\|\omega\| > 0.001$ (rad/s), $n = 4$ and

$$Q = \text{diag}(30 \ 2 \ 30 \ 0.2 \ 0.0001 \ 0.3 \ 0.0001)^2,$$

$$R = 0.008^2 I.$$

For slow rotating speed, $\|\omega\| < 0.001$ (rad/s), $n = 6$, and

$$Q = \text{diag}(120^2 I_x \ 120^2 I_y \ 120^2 I_z \ 3 \ 3 \ 3 \ 3),$$

$$R = \text{diag}(303 \ 3.4 \ 253)^2.$$

For variable sampling time, the error tolerance is given as $\varepsilon_0 = 10^{-16}$ and the Taylor series order $d = 20$.

To test the performance of the NMPC, a set of initial conditions are randomly generated. The NMPC is able to achieve satisfactory control performance and ensure the closed-loop stability as indicated in Figure 1.

In the simulation, initially, the controller adopts the sampling time as small as 500 (ms) to facilitate the maximum control action, as shown in sub-figure (c) so that the rotating speed of the satellite decreases quickly as indicated in sub-figure (a). Overall, the system is able to reach the desired attitude within 2 h as seen from sub-figure (b). When the system is close to its desired equilibrium position, the sampling time is gradually increased to 50 (s), which is about 100 time larger than the minimum sampling time such that the stability is observed over a long time period (6 h) as shown in sub-figure (b). Finally, sub-figure (d) shows the residual of the algebraic constraint, which is very difficult to be maintained at a small tolerance by using traditional differential equation solvers. By using the high-order Taylor series-based formulation developed
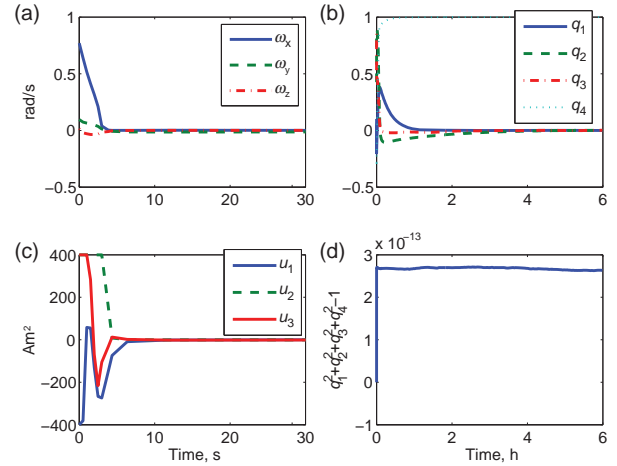

Figure 1. Simulation results of satellite NMPC (a) angular velocities, (b) Euler parameters (quaternion), (c) actuating forces and (d) residual of the algebraic constraint.
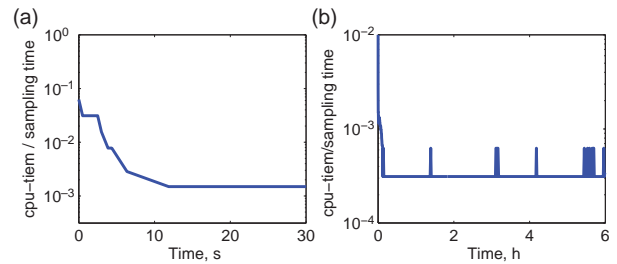

Figure 2. Computation time against sampling time (a) in short time scope, 0–30 s and (b) in long time scope, 0–6 h.

in this work, the residual is comfortably maintained at a very small level.

The computation efficiency is examined by checking the ratio of computation time against the sampling time, which is shown in Figure 2. The result was obtained by using an Intel® Core™ Duo Process T2500 (2.0 GHz, 2 MB L2 Cache, 667 MHz FSB) running MATLAB© R2009b. From Figure 2, it can be seen that in most time, the computation time spent is less than one-tenth of the sampling time. Therefore, the proposed scheme can comfortably handle the NMPC problem studied.

In contrast with the variable sampling-time scheme, if the traditional fixed sampling-time scheme were adopted, to cope with the large initial offset in ω_x (≈ 0.9 rad/s), the sampling time had to be fixed at 500 (ms), whilst to ensure the stability using the similar configuration adopted in the variable sampling-time scheme, the prediction horizon had to be 300 (s), i.e. the 600 sampling intervals. To fully implement the NMPC scheme proposed in Section 3.3, the Jacobian matrix would be in size of $120,000 \times 1800$, which requires at least 2 GB memory, hence is difficult to be implemented in a normal personal computer. Even with a simplified formulation given in Cao (2005), the error vector, E in Equation (44) would still have $(7 + 3) \times 600 = 6000$ elements and the Jacobian matrix, J would be in size
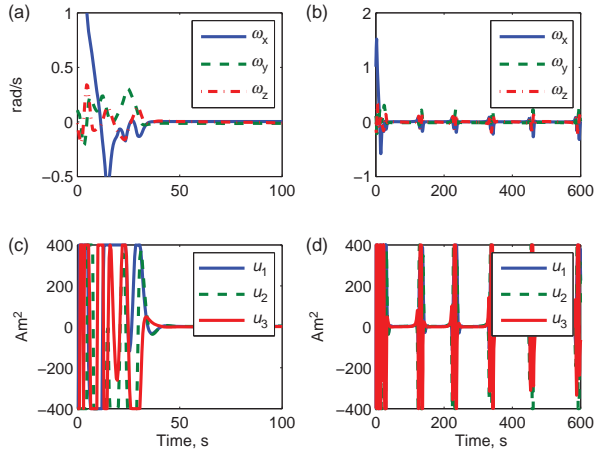
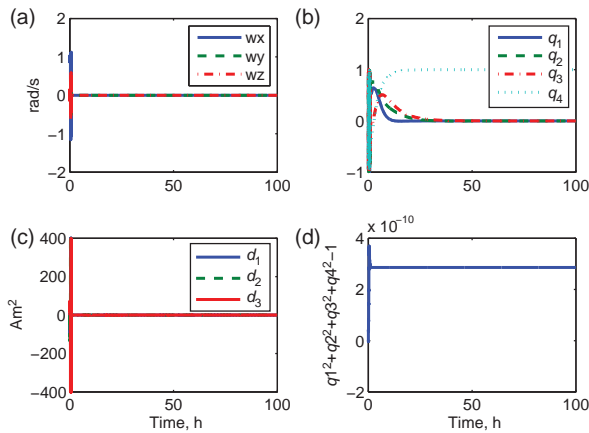Figure 3.   Simulation result of fixed sampling rate of 0.5 (s).

Figure 4.   Simulation result of the fixed sampling rate of 10 (s).

of 6000 × 1800. To solve such a large size NMPC problem within 500 (ms) demands a huge computation power making the approach computationally forbidden. For example, on the same PC described above, it requires more than 60 (s) to calculate a single update shown in Equation (44). Therefore, to solve the NMPC problem within a sampling interval of 0.5 (s) requires at least a 128-core supercomputer, which is almost impossible to be available on a small satellite like the one under study.

To highlight the advantage of the varying sampling-time scheme proposed, two fixed sampling-time schemes with sampling time at 0.5 and 10 (s) are simulated and the corresponding results are illustrated in Figures 3 and 4, respectively. In the 0.5 (s) sampling time, the satellite can be quickly converged to the desired attitude. However, due to the partial controllability, with such small sampling time, the desired attitude cannot be maintained. It can be observed that the satellite will deviate from, then converge back to the desired attitude periodically. In the second case when the sampling time is 10 (s), the stability around the desired attitude can be maintained. However, due to the large sampling time, it takes a very long time, longer than

20 h, for the satellite to converge from the initial state to the desired attitude.

5.   Conclusions

An AD-based NMPC formulation is developed for the satellite attitude control problem using magneto-torquers. A novel variable sampling-time mechanism has been proposed to tackle the problem where due to the coupling between the local magnetic field and the satellite attitude, the satellite dynamics change dramatically from initial orbit installation to the end of the manoeuvre so that both control performance and closed-loop stability can be maintained satisfactorily. Nonlinear simulation on a particular satellite case demonstrates the superior remits of the proposed control scheme.

References

Al-Seyab, R. K., & Cao, Y. (2008a). Differential recurrent neural network based predictive control. *Computers and Chemical Engineering*, *32*(7), 1533–1545.

Al-Seyab, R. K., & Cao, Y. (2008b). Nonlinear system identification for predictive control using continuous time recurrent neural networks and automatic differentiation. *Journal of Process Control*, *18*(6), 568–581.

Cao, Y. (2005). A formulation of nonlinear model predictive control using automatic differentiation. *Journal of Process Control*, *15*, 851–858.

Chen, W.-H. (2010). Stability of classic finite horizon model predictive control. *International Journal of Control, Automation and Systems*, *8*(2), 187–197.

Chen, W.-H., & Cao, Y. (2012). Stability analysis of constrained nonlinear model predictive control with terminal weighting. *Asian Journal of Control*, *14*(6), 1374–1381.

Griewank, A., Juedes, D., & Jean Utke. (1996). ADOL-C: A package for the automatic differentiation of algorithms written in C/C++. *ACM Transactions on Mathematical Software*, *22*, 131–167.

Liang, J., Fullmer, R., & Chen Y. (2004). Time-optimal magnetic attitude control for small spacecraft. In *IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas.

Lovera, M., & Astolfi, A. (2004). Spacecraft attitude control using magnetic actuators. *Automatica*, *40*, 1405–1414.

Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, *11*, 431–441.

Rall, L. B. (1981). *Automatic Differentiation: Techniques and Applications*. Lecture Notes in Computer Science (Vol. 120). Berlin: Springer Verlag.

Silani, E., & Lovera, M. (2005). Magnetic spacecraft attitude control: A survey and some new results. *Control Engineering Practice*, *13*, 357–371.

The MathWorks. (2007). *MATLAB Optimization Toolbox User's Guide*. The MathWork, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, 3 edition.

Wertz, J. (1978). *Spacecraft Attitude Determination and Control*. Dordrecht: D. Reidel Publishing Company.

Wood, M., & Chen, W.-H. (2008). Model predictive control of low earth orbiting satellites using magnetic actuation. *IMechE Part I: Journal of Systems and Control Engineering*, *222*(I6), 619–631.

Wood, M., & Chen, W.-H. (2013). Attitude control of magnetically actuated satellites with an uneven inertia distribution. *Aerospace Science and Technology*, 25(1), 29–39.

Wood, M., Chen, W.-H., & Fertin, D. (2006). Model predictive control of low earth orbiting spacecraft with magnetotorquers. In *IEEE Conefrence on Control Applications*. German.

Appendix. Recursive calculation of Taylor coefficients

Let $x^{[k]}$ represent the kth Taylor coefficient of function $x(t)$, i.e.

$$x(t) = x^{[0]} + x^{[1]}t + \cdots + x^{[d]}t^d. \qquad (A1)$$

The Taylor coefficients of arithmetic and simple functions can be directly derived in Tables A1 and A2.

Taylor coefficients of most other elementary functions, $z = f(x)$ can be derived through differentiation, $\dot{z} = f'(x)\dot{x}$.

Table A1.   Taylor coefficients.

Function	Taylor coefficients
$z = x \pm y$	$z^{[k]} = x^{[k]} \pm y^{[k]}$
$z = xy$	$z^{[k]} = \sum_{i=0}^{k} x^{[i]}y^{[k-i]}$
$z = x/y$	$z^{[0]} = x^{[0]}/y^{[0]}, z^{[k]} = (x^{[k]} - \sum_{i=0}^{k-1} z^{[i]}y^{[k-i]})/y^{[0]}$
$z = x^2$	$z^{[k]} = \sum_{i=0}^{k} x^{[i]}x^{[k-i]}$
$z = \sqrt{x}$	$z^{[0]} = \sqrt{x^{[0]}}, z^{[k]} = \left(x^{[k]} - \sum_{i=1}^{k-1} z^{[i]}z^{[k-i]}\right)/(2z^{[0]})$
$z = dx/dt$	$z^{[k]} = (k+1)x^{[k+1]}$

Table A2.   Taylor coefficients through differentiation.

Function	Derivative	Taylor coefficients
$z = x^a$	$\dot{z}x = az\dot{x}$	$z^{[0]} = (x^{[0]})^a, z^{[k+1]} = \dfrac{z_1^{[k]} - z_2^{[k]}}{(k+1)x^{[0]}}$
		$z_1^{[k]} = a \sum_{i=0}^{k} (k-i+1)z^{[i]}x^{[k-i+1]}$
		$z_2^{[k]} = \sum_{i=0}^{k-1} (i+1)z^{[i+1]}x^{[k-i]}$
$z = e^x$	$\dot{z} = zx$	$z^{[0]} = e^{x^{[0]}},$
		$z^{[k+1]} = \dfrac{1}{k+1} \sum_{i=0}^{k} z^{[i]}x^{[k-i]}$
$z = \ln x$	$zx = \dot{x}$	$z^{[0]} = \ln x^{[0]},$
		$z^{[k]} = ((k+1)x^{[k+1]}$
		$- \sum_{i=0}^{k-1} z^{[i]}x^{[k-i]})/x^{[0]}$
$u = \sin(x)$	$\dot{u} = v\dot{x}$	$u^{[0]} = \sin(x^{[0]}), v^{[0]} = \cos(x^{[0]})$
$v = \cos(x)$	$\dot{v} = -u\dot{x}$	$u^{[k+1]} = \dfrac{1}{k+1}$
		$\sum_{i=0}^{k} (k-i+1)v^{[i]}x^{[k-i+1]}$
		$v^{[k+1]} = \dfrac{-1}{k+1}$
		$\sum_{i=0}^{k} (k-i+1)u^{[i]}x^{[k-i+1]}$