Fixed ground-target tracking control of satellites using a nonlinear model predictive control

Abd El Hady M. Elbeltagy[1*], Ahmed M. Youssef[2], Amgad M. Bayoumy[3], Yehia Z. Elhalwagy[4]

[1] Aircraft Mechanics Dep., MTC, Cairo, Egypt
[2] Aircraft Electric Equipment Dep., MTC, Cairo, Egypt
[3] Mechanical Engineering Dep., MSA, Cairo, Egypt
[4] Navigation Dep., MTC, Cairo, Egypt

Corresponding Author Email: hady_beltagy@yahoo.com

ABSTRACT

This paper proposes a novel solution to the fixed-ground target tracking control problem of satellites utilizing a Nonlinear Model Predictive Control approach (NMPC). The Continuation / Generalized Minimal Residual (C/GMRES) algorithm is selected as a promising fast solver to an optimal control problem in real time. The algorithm could perfectly deal with the huge computational load of this approach, represented in solving Riccati differential equation, by simple and efficient approximations. A new control-oriented model converting the main tracking problem into a simple regulation problem is developed. This simple and easy traceable reformulated model has an advantage in dealing with modeling errors and unplanned external environmental disturbances. The update of the control input is obtained by integrating a deduced time-dependent inputs and Lagrange multipliers vector; representing the solution of a set of linear equations and corresponding to the optimality conditions. The proposed algorithm is simulated using real satellite parameters to track a fixed-ground target for reconnaissance purposes. The simulation results show that the algorithm of C/GMRES method can track a desired fixed ground target robustly, with precise tracking error and guaranteed safe stability limits for shooting activities throughout the overpass flight.

1. INTRODUCTION

The improvement of different strategic satellite attitude control systems by commissioning state-of-the-art intelligent control methods is an attractive arena of research recently. The extremely variant nature of satellite attitude control indicates that the elected effective intelligent control technique must be able to deal with a large Multi-Input Multi-Output (MIMO) system with nonlinear complex and constrained dynamics. Furthermore, in many conditions, it may also be necessary to avoid the different actuators from long operating near their thresholds to keep power consumption within some specified limits or to keep the rate of wear as low as possible.

Satellite Ground-Target Tracking (GTT) controllers must also guarantee a precise tracking error and fast response as well. NMPC, as a unique optimal model-based approach to intelligent control systems design, is a talented candidate that covers all of these features. The capability to handle Two-Point Boundary Value Problems (TPBVP) with such approach has made this technique superior. However, the main challenge faces NMPC controller developers is guaranteeing method implementation in a real-time in spite of the needed huge computational loads. Hence, fast Real-Time Optimization (RTO) methods are carefully searched to overcome this weakness. Optimization methods based on C/GMRES method are candidates of these RTO algorithms that have revealed great potential for real-time applications in general. Using the feature of the solution of a nonlinear receding horizon control problem that it varies smoothly related to the time at certain conditions is the main core idea in Ref. [1]. Thus, tracing this solution is possible without the need for iterative optimization methods. This is known by continuation method. Ohtsuka use this continuation method combined with a fast solver algorithm for linear equations instead of solving Riccati differential equation to develop his real-time algorithm of nonlinear receding horizon control problem. As he succeeded to manipulate the equations of nonlinear optimality conditions such that all the unknown variables which can't obtained recursively are put in one vector. Solving the nonlinear equations that are a function of this defined vector is replaced by searching this vector which can satisfy the features of any stable dynamic system. Hence, a set of linear equations are obtained and solved in real-time by employing the GMRES method [2].

Explicit MPC (eMPC) has been recently proposed as an alternative to online approaches in a trial to overcome the implementation challenges of a huge computational load for NMPC in each sampling time [3]. The nonlinear optimization problem in eMPC approach is completely solved offline to cancel the necessity of having an online optimization solver. However, the optimal solution is handled as a look-up table of linear gains that must be stored, the problem converted to storage challenges rather than computational ones to fulfill the limitations of the on-board processor.

The C/GMRES method is reapplied in a different application with using a developed modular code generation tool that can be tailored to be used in other applications [4]. However, some code changes should be employed to allow

covering the problems which comprise time-variant parameters that cannot considered as a system states.

The GMRES-based NMPC method has been used in designing ecological adaptive cruise controller (E-ACC) for plug-in Hybrid electric vehicles (HEV) [5]. In the developed E-ACC the upcoming trip data and an on-board vehicle radar reading are used to optimize energy cost of the trip and to preserve safety and comfort as well.

This paper investigates the potential of novel employing C/GMRES based RTO algorithms to design intelligent attitude control systems, in particular, intelligent fixed GTT controllers. The paper is organized to present a model predictive control and a control problem solution in section (2). In section (3) C/GMRES method is introduced in details. Section (4) is devoted to creating control-oriented model. Section (5) gives a brief about image quality requirements. In section (6) a case study with its simulation results is presented followed by the final conclusion statements in section (7).

2. CONTROL PROBLEM SOLUTION USING MPC

In the approach of MPC, or alternatively receding horizon control, the optimal solution of the control problem, called the control input is found by solving a constraint optimization problem at each time instant with a moving horizon. This results in a control law with a feedback nature. Actually, optimization problem of the constraint objective function can be converted into a root finding problem of a set of nonlinear equations. These nonlinear equations are obtained by means of applying necessary conditions for optimality [1].

A general nonlinear dynamic system can be represented as:

$$\left. \begin{array}{c} \dot{x}(t) = f\big(x(t), u(t), p(t)\big) \\ x(t_0) = x_o \end{array} \right\} \tag{1}$$

where, $x(t) \in R^n$ is a state vector with n states, $u(t) \in R^{m_u}$ is an input vector with m_u inputs and $p(t) \in R^{m_p}$ is a time-varying vector. The optimal control problem for this system where a cost function is minimized while satisfying n_g-dimensional equality constraints g and n_h-dimensional inequality constraints h are represented as [1]:

$$\underset{u}{\text{minimize}} : J = \varphi\left(x(t_f), p(t_f)\right)$$
$$+ \int_{t_0}^{t_f} L\big(x(\dot{t}), u(\dot{t}), p(\dot{t})\big) d\dot{t}$$
$$\text{subjected to} \begin{cases} \dot{x}(t) = f\big(x(t), u(t), p(t)\big) \\ x(t_0) = x_o \\ g\big(x(t), u(t), p(t)\big) = 0 \\ h\big(x(t), u(t), p(t)\big) \leq 0 \end{cases} \tag{2}$$

where, $T = t_f - t_0$ is the prediction horizon, $\varphi(.)$ represents the terminal cost at the end of the prediction horizon and $L(.)$ denotes the trajectory cost over the horizon. The actual state $x(t)$ is used as the initial state for the optimal control problem over the horizon $[t, t + T]$.

Although the control input is determined over this horizon, the actual submitted input as a feedback to the system is the only optimum value at the initial time of this current horizon [1,4,5].

Solving an optimization problem with inequality constraints is not an easy task because of the complexity of the optimality conditions known as Karush–Kuhn–Tucker (KKT) conditions. In this case, a particular interior point method known as barrier method is used to convert the inequality constraints to equality constraints. So, if $h_i \leq 0$ it can be written as $h_i + \alpha_i^2 = 0$ where α_i is a dummy input. If the problem is feasible, in the end, $h_i = -\alpha_i^2$ and for any given real value for α_i, $h_i \leq 0$. The integrant part in cost function presented in Equation (2) has been also modified by adding a term $-r^T\alpha$ that is to give incentive to the solver to stay away from the boundaries of the feasible set [1,4]. Thus, the optimal control problem can be modified as:

$$\underset{u,\alpha}{\text{minimize}} \ J = \varphi\left(x(t_f), p(t_f)\right)$$
$$+ \int_{t_0}^{t_f} \left(L\big(x(\dot{t}), u(\dot{t}), p(\dot{t})\big) - r^T\alpha(\dot{t}) \right) d\dot{t}$$
$$\text{subjected to} \begin{cases} \dot{x}(t) = f\big(x(t), u(t), p(t)\big) \\ x(t_0) = x_o \\ g\big(x(t), u(t), p(t)\big) = 0 \\ h\big(x(t), u(t), p(t)\big) + \alpha(t)^2 = 0 \end{cases} \tag{3}$$

So, if the entries of r are small positive numbers, the solution of this modified optimization problem is close to the solution of the original one with inequality constraints. Expanding the input vector $u(t)$ to contain not only the real inputs u but also the dummy inputs α and considering C is the final equality constraints with dimension $m_c = m_h + m_g$ including the original equality constraints and the converted inequality constraints to equality constraints, the optimal problem can be written in more useful and generalized form as following [1,4]:

$$\underset{u}{\text{minimize}} : J = \varphi\big(x(t + T), p(t + T)\big)$$
$$+ \int_t^{t+T} L\big(x(\dot{t}), u(\dot{t}), p(\dot{t})\big) d\dot{t}$$
$$\text{Subjected to:} \begin{cases} \dot{x} = f\big(x(t), u(t), p(t)\big) \\ x(t) = x_o \\ C\big(x(t), u(t), p(t)\big) = 0 \end{cases} \tag{4}$$

where, the optimal control $u^*(\dot{t}; t, x(t))$ that minimize J is calculated over $\dot{t} \in [t, t + T]$, and its initial value at initial time t, $u^*(t; t, x(t))$ is used as actual control input $u(t)$ to the system. The functions φ and L are commonly selected for such case as:

$$\varphi = \frac{1}{2}(x(t + T) - x_f)^T S_f(x(t + T) - x_f) \tag{5}$$

$$L = \frac{1}{2}\left((x - x_f)^T Q(x - x_f) + u^T R u - r^T\alpha \right) \tag{6}$$

where S_f, Q, and R are free parameters' weighting matrices.

The NMPC problem is basically a family of finite horizon optimal control problems (FHOCPs) along fictitious time τ as follows [1,4]:

$$\text{Minimize} : J = \varphi\big(x^*(T, t), p(t + T)\big)$$
$$+ \int_0^T L\big(x^*(\tau, t), u^*(\tau, t), p(t + \tau)\big) d\tau$$

Subjected to
$$\begin{cases} x_\tau^*(\tau,t) = f\big(x^*(\tau,t), u^*(\tau,t), p(t+\tau)\big) \\ \qquad x^*(0,t) = x(t) \\ C\big(x^*(\tau,t), u^*(\tau,t), p(t+\tau)\big) = 0 \end{cases} \tag{7}$$

where subscript τ represents partial differentiation with respect to τ. The new state vector, $x^*(\tau,t)$, denotes a trajectory along the τ axis starting from $x(t)$ at $\tau = 0$. The optimal control input, $u^*(\tau,t)$, is obtained on the τ axis as the solution to FHOCP for each t, and actual control input is set by $u(t) = u^*(0,t)$. Horizon T is commonly a function of time, $T = T(t)$.

To solve an optimization problem subject to equality constraints Lagrange multipliers method is used. The way it works is to write the Hamiltonian function as a sum of the integrant part of the cost function plus the dynamic equations multiplied by Lagrange multiplier $\lambda's$ known as costates plus the constraints equations multiplied by Lagrange multiplier $\boldsymbol{\mu's}$ as seen in the following equation [1].

$$\mathcal{H}(x,\lambda,u,\mu,p) = L(x,u,p) + \lambda^T f(x,u,p) + \mu^T C(x,u,p) \tag{8}$$

Once the Hamiltonian function is defined, the first-order conditions necessary for optimal control is getting by calculus of variation as Euler Lagrange Equations (ELEs) [4].

$$x_\tau^* = f(x^*, u^*, p) \tag{9}$$

$$x^*(0,t) = x(t) \tag{10}$$

$$\lambda_\tau^* = -\mathcal{H}_x^T(x^*, \lambda^*, u^*, \mu^*, p) \tag{11}$$

$$\lambda^*(T,t) = \varphi_x^T\big(x^*(T,t), p(t+T)\big) \tag{12}$$

$$\mathcal{H}_u(x^*, \lambda^*, u^*, \mu^*, p) = 0 \tag{13}$$

$$C(x^*, u^*, p) = 0 \tag{14}$$

where the argument of p is $t+\tau$ when absent. The ELEs describe a TPBVP, in which the value of initial state is known whereas the terminal costate is a function of the terminal state. Nonlinear TPBVP has to be solved within the sampling period for measured state $x(t)$ at each sampling time, which represents a hard-computational load and is one of the major problems in NMPC. Now with discretizing these conditions for optimality on τ-axis with forward difference by dividing the horizon into N time steps as follows [4].

$$x_{i+1}^*(t) = x_i^*(t) + f(x_i^*(t), u_i^*(t), p(t+i\Delta\tau))\Delta\tau \tag{15}$$

$$x_0^*(t) = x(t) \tag{16}$$

$$\lambda_i^* = \lambda_{i+1}^* + \mathcal{H}_x^T(x_i^*(t), \lambda_{i+1}^*(t), u_i^*(t), \mu_i^*(t), p(t+i\Delta\tau))\Delta\tau \tag{17}$$

$$\lambda_N^*(t) = \varphi_x^T\big(x_N^*(t), p(t+T)\big) \tag{18}$$

$$\mathcal{H}_u^T(x_i^*(t), \lambda_{i+1}^*(t), u_i^*(t), \mu_i^*(t), p(t+i\Delta\tau)) = 0 \tag{19}$$

$$C(x_i^*(t), u_i^*(t), p(t+i\Delta\tau)) = 0 \tag{20}$$

where time step $\Delta\tau = T/N$ and the real-time t is kept continuous while fictitious time τ is discretized. The computed sequences of the state, costate, input, and Lagrange multiplier on the discretized horizon are represented by $\{x_i^*(t)\}_{i=0}^N$, $\{\lambda_i^*(t)\}_{i=0}^N$, $\{u_i^*(t)\}_{i=0}^N$ and $\{\mu_i^*(t)\}_{i=0}^N$, respectively.

Consequently, NMPC is formulated as TPBVP in Equation (15-20) for measured state $x(t)$ at real time t. The Equations (15,16) are the system dynamic equations with its initial conditions. The next two Equations (17,18) are dynamic equations corresponding to the Lagrange multiplier costates $\boldsymbol{\lambda}$ with its final conditions. The rest of equations are equality constraints on all the variables that include the partial derivative of the Hamiltonian with respect to vector u, the vector of real and dummy inputs, and also the derivative of the Hamiltonian with respect to vector μ which lead to total equality constraints equation $C = 0$.

MPC is to solve an optimal control problem in each time step of a discrete time controller. While the system is working the current states x_n are measured or estimated then the optimal control problem is solved by computing the input u_i for the next N steps and only the first step of the sequence is returned and that is the computed input which should be applied to the system then this loop is repeated again and again. Therefore, MPC is an implementation of an optimal controller in a real time.

3. C/GMRES METHOD

The C/GMRES method proposed by Ohtsuka in [1] allows for solving optimization problem in a real-time. The method consists of two steps. In the first step, the dynamic of the system Equations (15,16) are solved forward in time and then the dynamic equations of the costates Equations (17,18) are solved backward in time in order to get all the states x_i and the costates λ_i in terms of other variables u_i^* and μ_i^*. In the second step, all the remaining variables are put together in vector called U that includes the inputs $u's$, which contains the real inputs, dummy inputs and Lagrange multipliers $\mu's$ for the time step from 0 to $N-1$ as:

$$U = [u_0^T, \ldots, u_{N-1}^T, \mu_0^T, \ldots, \mu_{N-1}^T]^T \tag{21}$$

Then rewriting the remaining optimality conditions as:

$$F(U(t), x(t), t) = 0 \tag{22}$$
where

$$F(U(t), x(t), t) = \begin{bmatrix} \mathcal{H}_u^T(x_0^*, \lambda_1^*, u_0^*, \mu_0^*, p(t)) \\ C(x_0^*, u_0^*, p(t)) \\ \vdots \\ \vdots \\ \mathcal{H}_u^T(x_{N-1}^*, \lambda_N^*, u_{N-1}^*, \mu_{N-1}^*, p(t+(N-1)\Delta\tau)) \\ C(x_{N-1}^*, u_{N-1}^*, p(t+(N-1)\Delta\tau)) \end{bmatrix} \tag{23}$$

So, Equation (23) should be solved for U. The problem is that F is nonlinear function and this equation cannot be solved easily in general. C/GMRES method is used to solve this problem. The continuation method is simply instead of solving $F(U(t), x(t), t) = 0$, it selects $U(0)$ such that $F(U(0), x(0), t) = 0$ and then find U so that [1]:

$$\dot{F}(U(t), x(t), t) = -\xi\, F(U(t), x(t), t), \xi > 0 \qquad (24)$$

where $\dot{F}$ is the derivative of F with respect to time and the right-hand side is actually added to stabilize $F = 0$. If Equation (24) is satisfied, regardless of where F is started which can be nonzero, it will converge to zero because this is the dynamics of the stable linear system. The derivative of F with respect to time can be computed using partial derivative with respect to x and U as:

$$F_x \dot{x} + F_U \dot{U} + F_t = -\xi\, F(U, x, t) \qquad (25)$$

Equation (25) has a very interesting property which is its linearity in $\dot{U}$. Rewrite Equation (25) as:

$$\dot{U} = F_U^{-1}(-\xi F - F_x \dot{x} - F_t) \qquad (26)$$

Given the current states x, the current value of vector U and the vector field of the system $f(x, u, p)$ or in other words $\dot{x}$, everything can be computed except $\dot{U}$. Since Equation (26) is linear in $\dot{U}$, a linear solver to compute $\dot{U}$ is needed. Once $\dot{U}$ is computed $U(t)$, update can be done to compute the new U and in turn, compute the control signal.

Unfortunately, Equation (26) still needs high computational operations represented in Jacobians F_U, F_x and F_t in addition to linear equation accompanied with F_U^{-1}. To reduce these computational loads, two main devices are employed. First the forward difference approximation for products of Jacobians and vectors, and second the GMRES method for the linear equation. The approximation for the products of the Jacobian and some $W \in R^{mN}, w \in R^n$ and $\omega \in R$ as:

$$F_U(U, x, t)W + F_x(U, x, t)w + F_t(U, x, t) \simeq$$
$$D_h F(U, x, t: W, w, \omega) = \frac{F(U + h_{dir}W, x + h_{dir}w, t + h\omega) - F(U, x, t)}{h_{dir}} \quad (27)$$

where h_{dir} is a positive real number. So, Equation (25) can be approximated using forward difference approximation method as [1,4]:

$$D_h F(U, x, t: \dot{U}, \dot{x}, 1) = -\xi\, F(U, x, t) \qquad (28)$$

which can be written in an equivalent form as:
$$D_h F(U, x + h\dot{x}, t + h: \dot{U}, 0,0) = b(U, x, \dot{x}, t) \qquad (29)$$

where
$$b(U, x, \dot{x}, t) == -\xi\, F(U, x, t) - D_h F(U, x, t: 0, \dot{x}, 1) \quad (30)$$

It is clear that the forward difference approximation of the products of the Jacobians and vectors can be estimated with only an additional calculation of the function, which needs particularly less computational load than approximation of the Jacobians themselves.

As Equation (29) approximates a linear equation in $\dot{U}$, a linear solver algorithm GMRES is applied with slight modification to include a certain initial guess of $\widehat{U}$ [1]. The algorithm is employed with applying forward difference approximation so it is called FDGMRES [2] as follows:

Algorithm 1. $\dot{U} := $ FDGMRES $(U, x, \dot{x}, t, \widehat{U}, h, k_{max})$
(1) $\hat{r} := b(U, x, \dot{x}, t) - D_h F(U, x + h\dot{x}, t + h: \dot{U}, 0,0)$,
$v_1 := \hat{r}/\|\hat{r}\|$,
   $\rho := \|\hat{r}\|$,

$\beta := \rho$,
$k := 0$.
(2)   While $k < k_{max}$, do
a. $k = k + 1$
b. $v_{k+1} := D_h F(U, x + h\dot{x}, t + h: v_k, 0,0)$
c.   For $j = 1: k$ $h_{jk} := v_{k+1}^T v_j$
d. $v_{k+1} := v_{k+1} - h_{jk} v_j$
e. $h_{k+1,k} := \|v_{k+1}\|$
f. $v_{k+1} := v_{k+1}/\|v_{k+1}\|$
g.   For $e_1 = [1\,0\,\dots\,0]^T \in \mathbb{R}^{k+1}$ and $H_k = (h_{ij}) \in \mathbb{R}^{(k+1)\times k}$
$(h_{ij} = 0\ for\ i > j + 1)$,
Minimize $\|\beta e_1 - H_k y^k\|$
Determine $y^k \in \mathbb{R}^k$ $\rho := \|\beta e_1 - H_k y^k\|$.
(3) $\dot{U} := \widehat{U} + V_k y^k$ where $V_k = [v_1\,\dots\,v_k] \in \mathbb{R}^{mN\times k}$

For a linear equation $Ax = b$ with a nonsymmetric matrix A, the GMRES algorithm minimizes the residual $\rho := \|b - Ax\|$ at the kth iteration with $x \in x_0 + \mathcal{K}_k$, where x_0 vector is the initial guess and $\mathcal{K}_k$ denotes the Krylov subspace expressed by $\mathcal{K}_k := \text{span}\{r_0, Ar_0, \dots, A^{k-1}r_0\}$ and $r_0 := b - Ax_0$. GMRES creates an orthonormal basis $\{v_j\}_{j=1}^k$ for $\mathcal{K}_k$. Givens rotations is used for efficient minimization in step 2e shown in algorithm steps. Theoretically, GMRES has the ability to reduce the residual ρ for a large linear equation monotonically and converges rapidly to an accurate solution with specified error tolerance within number of iterations equal to the dimension of the equation as maximum [1].

The product of the Jacobian matrix and a vector, $F_U(U, x, t)v_j$ in FDGMRES algorithm is computed using forward difference approximation, $D_h F(U, x + h\dot{x}, t + h: v_k, 0,0)$. Since the orthonormal basis $\{v_j\}_{j=1}^k$, k vectors in $\mathbb{R}^{mN}$, must be saved during the implementation of FDGMRES, a massive amount of data storage may be required for a large-scale problem. Additionally, several iterations may need larger execution time which cannot be available in a real-time implementation. Therefore, the maximum number of iterations k_{max} should be carefully selected as small as possible to avoid excessive processing. Fortunately, both warm start at which the solution $\dot{U}$ at the previous sampling time is used as a consecutive initial guess for FDGMRES and selection of a small k_{max} often guarantee an accurate performance in a real-time application. On the other hand, in case of the need for a large number of iterations k_{max} while a limited amount of data storage is available, restart FDGMRES from the current iterate can efficiently handle this case. In restart technique, the orthonormal basis is reset with the latest obtained basis.

Once $\dot{U}$ is computed using FDGMRES algorithm, the update of the solution curve $U(t)$ of $F(U, x, t) = 0$ is traced simply without iterative optimization methods by integrating this $\dot{U}$ with the Euler method in real time [6]. This approach is typical continuation method. Although the computational load of updating $U(t)$ for the explicit Euler technique related to only one iteration in Newton's method, it realizes higher accuracy by taking the time dependence of the equation into account [4]. As the continuation method is combined with GMRES, this algorithm is named C /GMRES. The Ohtsuka's suggestion of warm start in [1] which considers the solution for $\dot{U}$ in previous sampling time as the initial guess for current timestep's first inner iteration is used in this paper. The vector $\dot{x}$ in FDGMRES algorithm is computed through vector field

equations $f(x(t), u(t), p(t))$ or approximated by finite difference in practical implementation as:

$$\dot{x}(t) \approx (x(t + \Delta t) - x(t))/\Delta t \tag{31}$$

For designed sampling period Δt and using the suffix (l) to express different variables at time $l\Delta t$, the C/GMRES algorithm for receding horizon control can be concluded as follows [1].

Algorithm 2. C /GMRES
(1)   Initialization
$t := 0$,
$l := 0$,
Measure the initial state and set $x_{(0)} := x(0)$
find $U_{(0)}$ analytically such that $\|F(U_{(0)}, x_{(0)}, 0)\| \le \delta$ for some positive δ
(2)   For $\dot{t} \in [t, t + \Delta t]$, set $U(\dot{t}) := P_0(U_{(l)})$.
(3)   At time $t + \Delta t$,
set the state $x_{(l+1)} := x(t + \Delta t)$
set $\Delta x_{(l)} := x_{(l+1)} - x_{(l)}$.
(4)   compute $\dot{U}_{(l)} :=$ FDGMRES $(U_{(l)}, x_{(l)}, \Delta x_{(l)}/ \Delta t, t, \widehat{U}_{(l)}, h, k_{max})$ where the initial guess of $\widehat{U}_{(l)}$ is appropriately chosen; e.g., $\widehat{U}_{(l)} := 0$ or $\widehat{U}_{(l)} := \dot{U}_{(l-1)}$ with $\widehat{U}_{(-1)} := 0$.
(5)   Set $\Delta U_{(l)} := \dot{U}_{(l)} \Delta t$
(6)   Set $U_{(l+1)} := U_{(l)} + \Delta U_{(l)}$.
(7)   Update
$t := t + \Delta t$
$l := l + 1$
Go back to step 2.

It is clear from the C/GMRES algorithm that the usage of this iterative methodology is limited only in solving the linear Equation (26) with respect to $\dot{U}$. Then using this solution, the solution of the original nonlinear equation, $F(U, x, t) = 0$, is traced easily with no line search or further Newton iteration. It means that the C/GMRES solves the linear Equation (26) only once per sampling time and, consequently, needs much less computational load than other standard iterative methods such as Newton's method which needs to solve the linear equation many times to find search directions. Additionally, the C/GMRES algorithm doesn't need line search, which is also a major difference from other regular optimization methods.

A simple way of initializing $U(0)$ is to select a time-dependent horizon, $T(t)$, as a smooth function satisfying $(T(0) = 0)$ and $(T(t) \to T_f)$ at $(t = \infty)$ as [1]:

$$T(t) = T_f(1 - e^{-\alpha t}) \tag{32}$$

where T_f and α are positive numbers. Since the horizon has a zero length at $(t = 0)$, this leads to the following initial optimal parameters:

$$
\begin{aligned}
u_i^*(0) &= u(0) & (i = 0{:}N-1) \\
\mu_i^*(0) &= \mu(0) & (i = 0{:}N-1) \\
x_i^*(0) &= x(0) & (i = 0{:}N) \\
\lambda_i^*(0) &= \varphi_x^T(x(0)) & (i = 0{:}N)
\end{aligned}
\tag{33}
$$

The $U(0)$ initialization reduces to finding only $m_u + m_c$ unknown quantities, $u(0)$ and $\mu(0)$, such that [1]:

$$\left\| \begin{bmatrix} \mathcal{H}_u^T(x(0), \varphi_x^T(x(0)), u(0), \mu(0), p(0)) \\ C(x(0)), u(0), p(0) \end{bmatrix} \right\| \le \frac{\delta}{\sqrt{N}} \tag{34}$$

where δ is a positive number. Equation (34) has a much-reduced size than the original Equation (23) and can be solved with Newton's method off line or within the sampling period directly after initial state $x(0)$ has been estimated. The error in optimality, $\|F\|$, is guaranteed to be limited under several assumptions if the parameter ξ is selected such that $(0 < \xi\Delta t \le \bar{\xi}\Delta t)$ for assumed sampling period Δt and some $\bar{\xi}$ such that $(1 < \bar{\bar{\xi}}\Delta t \le 2)$ [1].

Even if the problem of real-time optimization becomes a realistic option in implementing NMPC, another vital problem for practical application remains how to properly tune an NMPC performance index. Since it is problematic to have an explicit solution for a nonlinear optimal control problem, it is really difficult to relate the closed-loop response clearly to design parameters setting in a performance index such as weighting matrices and horizon length. Thus, tuning the performance index to realize a desired closed-loop performance is very effortful, not systematic and is considered as one of the major drawbacks of this approach.

An extensive search involving performance impact of numerous trials and modification of the free parameters is an alternative solution to overcome this drawback. This search is used here by randomly select the tuned parameters and hence their effect on the performance of the proposed NMPC controller is studied. This effect is used as a guide for optimum parameters selection.

4. CONTROL ORINTED MODEL

In fact, the efficient control performance of the NMPC scheme is always obstructed by the limited model precision. As for the complicated satellite dynamics with time-varying parameters and nonlinear characteristics, the used model may have some errors in addition to unexpected disturbance. Moreover, for the optimal tracking problem, the error dynamic model is crucial for the robust control design. Therefore, a new control-oriented model based on the attitude and rate errors is designed in this paper. Doing that allows handling the tracking task as a simple regulation task.

In this section, the main tracking problem is converted into a well-formulated global and simple regulation problem. The simple regulation problem is a more traceable problem and has a great potential dealing with disturbance and uncertainty. The attitude kinematic equation using quaternion error representation is described in [7] considering the inertial frame as a reference frame. The satellite is assumed to start GTT task from nadir pointing attitude and hence the orbital frame is used as a reference frame instead of the inertial frame for simplicity purposes. Thus, the attitude kinematic equation using quaternion error representation can be reformulated as:

$$\delta \dot{\bar{q}} = 0.5 \mathbf{\Omega}(\delta\omega) \delta\bar{q} \tag{35}$$

where

$$\delta\omega = \omega_{SBC}^{ORC} - A_d^{SBC} \omega_d^{ORC} \tag{36}$$

$$\delta\bar{q} = \begin{bmatrix} \boldsymbol{q_e} \\ q_{4e} \end{bmatrix} = \begin{bmatrix} q_{4d} & q_{3d} & -q_{2d} & -q_{1d} \\ -q_{3d} & q_{4d} & q_{1d} & -q_{2d} \\ q_{2d} & -q_{1d} & q_{4d} & -q_{3d} \\ q_{1d} & q_{2d} & q_{3d} & q_{4d} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (37)$$

The subscript d represents the desired frame, $\boldsymbol{\Omega}(\delta\omega)$ is 4×4 skew matrix function, the $\delta\bar{q}$ represents the quaternion error from the desired to the body frame, q_d represents desired quaternion, q represents the current attitude, $\delta\omega$ is the error in angular rate represented in the body frame, ω_{SBC}^{ORC} is the orbital referenced body angular rate in body frame, ω_d^{ORC} is the orbital referenced desired angular rate in the desired frame and A_d^{SBC} is the transformation matrix from desired to body frame.

The dynamic model of an Earth-pointing satellite using reaction wheels (RWs) only as active actuators is given by:

$$J\dot{\omega}_{SBC}^{ORC} = N_{\omega o} + N_{GG} - \omega_{SBC}^{EIC} \times (J\omega_{SBC}^{EIC} + h_w) - \dot{h}_w \quad (38)$$

Using Equation (36) to express the body angular rate ω_{SBC}^{ORC} as:

$$\omega_{SBC}^{ORC} = \delta\omega + A_d^{SBC}\omega_d^{ORC} \quad (39)$$

Differentiating Equation (39) to get

$$\dot{\omega}_B^{ORC} = \delta\dot{\omega} - \Omega(\delta\omega)A_d^{SBC}\omega_d^{ORC} + A_d^{SBC}\dot{\omega}_d^{ORC} \quad (40)$$

where $\Omega(\delta\omega)$ is 3×3 skew matrix. Multiplying Equation (40) by J leads to

$$J\dot{\omega}_{SBC}^{ORC} = J\delta\dot{\omega} - J[\Omega(\delta\omega)A_d^{SBC}\omega_d^{ORC} - A_d^B\dot{\omega}_d^{ORC}] \quad (41)$$

Substituting $J\dot{\omega}_B^{ORC}$ from Equation (41) in Equation (38) gives:

$$J\delta\dot{\omega} = N_{\omega o} + N_{GG} + J \quad [\quad \Omega(\delta\omega)\, A_d^{SBC}\omega_d^{ORC} - A_d^{SBC}\,\dot{\omega}_d^{ORC}] - \omega_{SBC}^{EIC} \times (J\omega_{SBC}^{EIC} + h_w) - \dot{h}_w \quad (42)$$

Generally, the terms N_{GG} and $N_{\omega o}$ and $\omega_{SBC}^{EIC} \times (J\omega_{SBC}^{EIC} + h_w)$, are not significant for many practical rotation maneuvers [8]. The term $\dot{\omega}_d^{ORC}$ can also be neglected as it is lower several orders of magnitude than other terms for nearly circular orbit. All the omitted terms can be considered as some types of uncertainty which can be handled effectively using such presented model. Robustness check in the simulation is also used to confirm the validity of NMPC with the presented model in general. Rewriting Equation (42) in more simple and efficient form after neglecting the smaller terms yields:

$$J\delta\dot{\omega} = J\Omega(\delta\omega)A_d^{SBC}\omega_d^{ORC} + u_w$$
$$\dot{h}_w = -u_w \quad (43)$$

where u_w represents the RWs torque. Equation (35) and Equation (43) are the control-oriented model or in other words, the tracking error dynamic model used in this paper. This model has 10 states represented in attitude quaternion error, rate error, and RWs momentum. The model also includes time-varying parameter represented in $A_d^{SBC}\omega_d^{ORC}$ which add a little bit difficulty in implementation. The desired rate ω_d^{ORC} is [8]:

$$\omega_d^{ORC} = u_{S/T}^{ORC} \times \dot{u}_{S/T}^{ORC} \quad (44)$$

where $u_{S/T}^{ORC}$ is unit direction vector of the direction from satellite to the target represented in orbit frame, $X_{S/T}^{ORC}$ computed by [8]:

$$X_{S/T}^{ORC} = A_{EIC}^{ORC}(A_{EFC}^{EIC}X_T^{EFC} - r_{sat}) \quad (45)$$

as r_{sat} is the satellite position in Earth-centered inertial frame, X_T^{EFC} is the target location in Earth-centered fixed frame and the transformation matrix A_{EFC}^{EIC} from the fixed frame to the inertial is time-dependent matrix can be estimated considering constant Earth angular rate around its rotation axis, ω_E in the tracking maneuver period and knowing the initial phase , α_G between the x-axes of both EFC and EIC frames at $t = 0$ as:

$$A_{EFC}^{EIC} = \begin{bmatrix} cos(\omega_E t + \alpha_G) & -sin(\omega_E t + \alpha_G) & 0 \\ sin(\omega_E t + \alpha_G) & cos(\omega_E t + \alpha_G) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (46)$$

The transformation matrix A_{EIC}^{ORC} from the inertial frame to orbital frame can be deduced by using the satellite position r_{sat} and velocity v_{sat} in the inertial frame as [8]:

$$A_{EIC}^{ORC} = [\hat{u} \quad \hat{v} \quad \hat{w}]^T \quad (47)$$

where the unit vectors $\hat{u}$, $\hat{v}$ and $\hat{w}$ are:

$$\left. \begin{aligned} \hat{w} &= -\frac{r_{sat}}{\|r_{sat}\|} \\ \hat{v} &= -\frac{r_{sat} \times v_{sat}}{\|r_{sat} \times v_{sat}\|} \\ \hat{u} &= \hat{v} \times \hat{w} \end{aligned} \right\} \quad (48)$$

The vector $\dot{u}_{S/T}^{ORC}$ can be calculated as following [8]:

$$\dot{u}_{S/T}^{ORC} = \frac{1}{\|X_{S/T}^{ORC}\|}[I_3 - u_{S/T}^{ORC}(u_{S/T}^{ORC})^T]\,\dot{X}_{S/T}^{ORC} \quad (49)$$

where the time derivative $\dot{X}_{S/T}^{ORC}$ is calculated by differentiating Equation (45) to get the following [8]:

$$\dot{X}_{S/T}^{ORC} = \dot{A}_{EIC}^{ORC}(A_{EFC}^{EIC}X_T^{EFC} - r_{sat}) + A_{EIC}^{ORC}(\dot{A}_{EFC}^{EIC}X_T^{EFC} - v_{sat}) \quad (50)$$

The time derivative of transformation matrices A_{EIC}^{ORC} and A_{EFC}^{EIC} in Equation (50) are calculated as following [8]:

$$\dot{A}_{EIC}^{ORC} = [\hat{\dot{u}} \quad \hat{\dot{v}} \quad \hat{\dot{w}}]^T \quad (51)$$

where

$$\left. \begin{aligned} \hat{\dot{w}} &= -\frac{1}{\|r_{sat}\|}(I_3 - \hat{w}\hat{w}^T)v_{sat} \\ \hat{\dot{v}} &= 0 \\ \hat{\dot{u}} &= \frac{1}{\|r_{sat}\|}\hat{v} \times [(I_3 - \hat{w}\hat{w}^T)v_{sat}] \end{aligned} \right\} \quad (52)$$

$$\dot{A}_{EFC}^{EIC} = \omega_E \begin{bmatrix} -sin(\omega_E t + \alpha_G) & -cos(\omega_E t + \alpha_G) & 0 \\ cos(\omega_E t + \alpha_G) & -sin(\omega_E t + \alpha_G) & 0 \\ 0 & 0 & 0 \end{bmatrix} (53)$$

The transformation matrix A_d^{SBC} in Equation (43) can be extracted easily from the quaternion error $\delta\bar{q}$ given in

Equation (37). The desired quaternion q_d can be estimated as following [8]:

$$q_d = \begin{bmatrix} u_c sin\frac{\delta}{2} \\ cos\frac{\delta}{2} \end{bmatrix} \qquad (54)$$

where δ is the angle between the time-dependent vector $u_{S/T}^{ORC}$ and commissioned axis or payload axis represented in body frame u_{com}^{SBC} while u_c define the axis of rotation needed for constructing q_d given by [8]:

$$u_c = \frac{u_{com}^{SBC} \times u_{S/T}^{ORC}}{\|u_{com}^{SBC} \times u_{S/T}^{ORC}\|} \qquad (55)$$

The tracking error can be examined by calculating θ_{err} to define the angle between $u_{S/T}^{SBC}$ and $u_{S/T}^{ORC}$ vectors as [8]:

$$\theta_{err} = cos^{-1}(u_{S/T}^{SBC} \cdot u_{S/T}^{ORC}) \qquad (56)$$

where the unit vector $u_{S/T}^{SBC}$ defined for the current attitude matrix A_{ORC}^{SBC} as:

$$u_{S/T}^{SBC} = A_{ORC}^{SBC} \cdot u_{S/T}^{ORC} \qquad (57)$$

5. IMAGE QUALITY REQUIREMENTS

The image quality enhancement is directly linked to accurate and low distortion images. To obtain low distortion images during tracking mode, the shooting conditions should guarantee stability requirements. Moreover, to ensure accurate shooting of definitely targeted area, minimum permissible geo-location error should be kept. For a given GSD (i.e. one pixel) capability of a certain payload its corresponding angular distance θ_{pixel} should be obtained as [9,10].

$$\theta_{pixel} = tan^{-1}(\frac{GSD}{h}) \qquad (58)$$

where h is orbit altitude. The exposure time t_{exp} of a single pixel (or a single row of pixels) is a function of the effective ground speed v_{geff} as follows [9]:

$$t_{exp} = \frac{GSD}{v_{geff}} \qquad (59)$$

$$v_{geff} = \frac{v_g}{n_{fmc}} \qquad (60)$$

as n_{fmc} is the forward motion compensation (FMC) factor; reducing the effective ground speed of the image sensor boresight, v_g is the satellite ground speed defined, μ_E is the Earth gravitational constant and R_E is the Earth radius.

The permissible safe variation in angular rate around the image sensor's boresight, ω_{sz}, during shooting can be determined [9]:

$$\omega_{sz} \approx \frac{tan^{-1}\left(\frac{max\_smear \times GSD}{0.5 \times L_{swath}}\right)}{t_{exp}} \qquad (61)$$

where L_{swath} is the image swath width while max_smear is the maximum pixel smear in percentage. The permissible safe

variation in angular rate across the image sensor's boresight, ω_{sxy}, can similarly be computed via [9]:

$$\omega_{sxy} \approx \frac{max\_smear \times GSD}{t_{exp}} \qquad (62)$$

The stability requirement is selected to be the minimum of the calculated angular rate variation around the image sensor's boresight, ω_{sz}, and the calculated angular rate variation across the image sensor's boresight, ω_{sxy}. On the other hand, the payload integration time during which data is collected by the image sensor is another issue that directly affects the image quality. The higher the integration time, the more data can be acquired and consequently the more enhancement in the image quality.

6. CASE STUDY

In order to evaluate the performance of using algorithm C/GMRES in solving a ground target tracking problem, the configuration UoSAT-12 satellite with higher RWs capabilities is used in the simulation. The inertia matrix J of the satellite in kgm^2 is $diag(40,40,32)$.

Table 1. Controller and simulation parameters

Setting parameters		
Symbol	Description	Setting
$dimx$	Dimension of state vector	10
u	Number of real input (actuators)	3
$dimc$	Number of constraints	6
$dimu$	Dimension of control input vector	15
$dimp$	Dimension of Time-Variant Parameters	3
T_f	final horizon length (sec.)	10
α	For defining variable prediction horizon	0.8
ξ	For Stabilization of Continuation Method	5
h_{dir}	Step in forward difference approximation	0.0001
r_{tol}	Error tolerance in control input vector	1e-06
k_{max}	Number of iteration in GMRES	60
d_v	Number of grids on the horizon	10
r_1 to r_6	For keeping boundary limit of feasible set	0.0028
t_{sim}	Simulation time (sec.)	800
h_t	Simulation step (sec.)	0.2
d_{step}	time step for saving data (sec.)	5
Weighting matrices and initial input vector		
Symbol	Description and setting	
$U(0)$	Initial guess for initial control input vector $0.0001 * diag(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)$	
S_f	Weighting values for terminal state $10^3 * diag(1,1,1,1,1,1,1,0.00001,0.00001,0.00001)$	
Q	Weighting values for trajectory impact $diag(50,50,50,50,50,50,50,0.01,0.01,0.01)$	
R	Weighting values for control input $diag(77,77,77,0.01,0.01,0.01,0.01,0.01,0.01,0.01)$	

Three reaction wheels installed in a three-axis configuration to allow full control of the satellite attitude orientation with maximum torque $\pm0.2\,Nm$ and momentum capacity $\pm6\,Nmsec$. The working orbit is almost circular with altitude $650\,km$, inclination angle $64.5\,deg$, right ascension of ascending node $10\,deg$, initial mean anomaly $0\,deg$ and initial mean phase $0\,deg$. The tracked fixed ground-target in EFC frame is $[4021.9\ \ -35.1\ \ 4933.6]^T$. The satellite

commissioned axis is $[1 \quad -1 \quad 9]^T$ and the satellite attitude is assumed initially aligned with orbit frame.

During simulation perfect attitude knowledge and measurements of the vectors r_{sat} and v_{sat} are assumed. The developed code of C/GMRES method is outputted as an ANSI/C standard source code for simulation purposes in this paper. In Table 1 a list of all the controller variable and simulation parameters used during simulation for GTT task.

Since the clear desired states for reformulated regulation problem represented in the attitude quaternion and angular rate, it is planned to lightly punish the deviation between the desired zero angular momentum and realized momentum during optimum solution searching.

Actually, the restrict requirements concern the trajectory of angular momentum change is considered in the limits put on the input vector real elements (the first three elements in the input vector). All parameters which have no real physical limits are allowed to be changed freely that is why their corresponding weighting values are relatively too low.

UoSAT -12 conveys two redundant $32\ m$ ground sampling distance (GSD) multi-spectral cameras to provide $64\ Km$ wide swath width [11]. Furthermore, UoSAT-12 also carries a high-resolution narrow-angle panchromatic camera with $10\ m$ GSD and $10\ Km$ swath [11]. The stability requirement during imaging to obtain acceptable image quality with relaxed maximum pixel smear in percentage 17% as explained in section (5) is computed and found to be better than $0.1\ deg/sec$. It can be realized from Figure 1 and Figure 2 that the satellite needs only (≈ 200 sec) to be kept in a stable tracking mode with a very small tracking error defined in Equation (56), $\theta_{err} \approx 0.0045\ deg$.

The supreme advantage here is that the proposed controller succeeded to maintain this precise tracking error during the overpass flight simulation time. Normally this accuracy is deteriorated whenever the satellite position becomes closer to the target. The resultant bell-shaped figure with maximum satellite angular slew rate at a closer position to the target violate the stability conditions needed for shooting with accepted image quality. The proposed tracking controller here can react as fast as the kinematic of the target. Keeping the tracking error within this precise limit during the overpass flight of the target allows shooting this target at any time. Hence, an additional shooting mode covering stereo triplets imaging and video surveillance can be carried out.
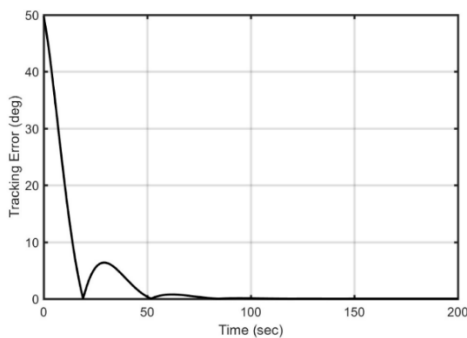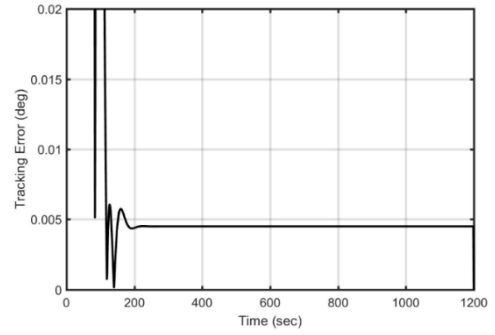

Figure 1. Tracking error

Figure 2. Tracking error zoom in

The Euler angles representing the satellite attitude change related to the desired frame is shown in Figure 3. The initial attitude mismatch is diminished fast.
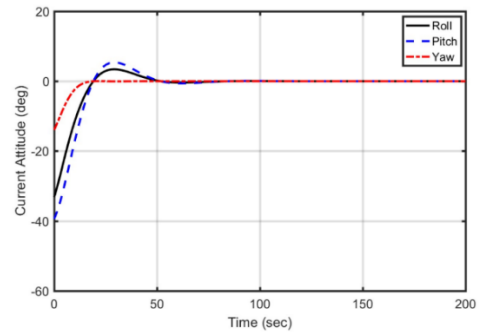

Figure 3. Current attitude in Euler angles

The angular rate error of the satellite body is shown in Figure 4 during the tracking control. It is seen that the that the orbit-referenced angular velocity of the satellite body ω_B^{ORC} follows the desired angular rate command ω_d^{ORC} without a noticeable error during the tracking period.

The stability requirement (better than $0.1\ deg/sec$) during imaging to obtain acceptable image quality is initially guaranteed after only $57\ sec$ as cleared from Figure 5. Fortunately, this stability guarantee is kept through over the target overpass flight simulation time. From Figure 2 and Figure 5, the shooting conditions with minimum geo-location error is guaranteed after $200\ sec$ of starting tracking process with tracking error 0.0045 deg.
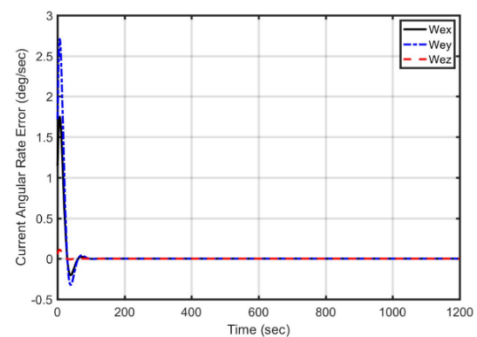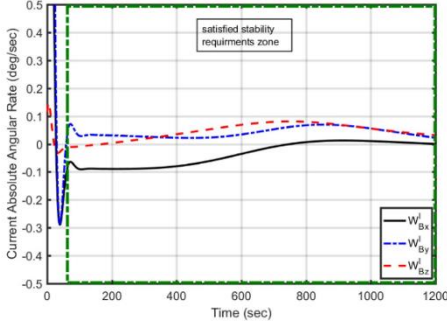

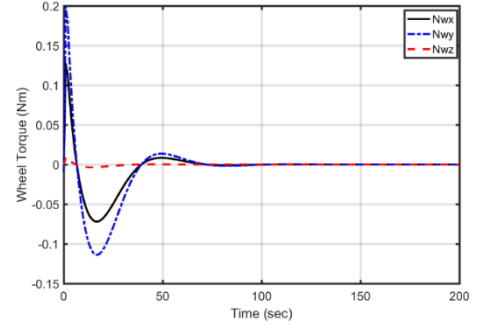**Figure 4.** Satellite angular rate error

Figure 5. Current angular rate

The geo-location error due to this error will be $0.05\ km$. Therefore, the high-resolution narrow-angle panchromatic camera which offers $10\ km$ swath (i.e. image size 10×10) can be used efficiently while tracking with the proposed intelligent tracking controller. The steady-state quaternion error $q_e = [0\ 0\ 0\ 1]^T$ represented in Figure 6 means that body attitude is totally aligned with the desired attitude.
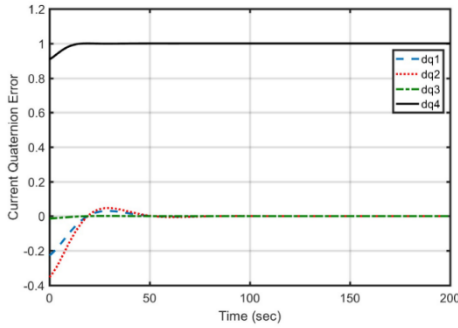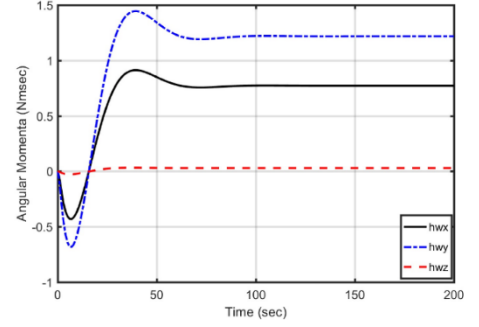

Figure 6. Quaternion error

Since the UoSAT-12 satellite is equipped with body-fixed imager, aligning the body attitude with the desired attitude means that this imager (payload) pointing at the target.

The activity of the three-axis RWs is shown in Figure 7 and Figure 8. Figure 7 confirms that the torques of the three-axis RWs are generally far from saturation limits ($0.2\ Nm$) during the duration needed for tracking the ground target. A wheel torque transient, in general, happens at the beginning of the tracking maneuver due to initial attitude error between the satellite payload axis and the desired target direction. The max torque provided by the RW of Y-axis and it is almost ($0.19\ Nm$) with very limited duration time. Once the controller starts to respond to the desired command, this attitude error starts to be reduced and hence the torques needed from RWs are decreased.

The angular momentums of the three-axis RWs in Figure 8 are far from angular momentum capacity h_{w_max} ($6.0\ Nm\ sec$) during tracking task with the proposed controller. This safe working momentum region allows the RWs to further compensate for any disturbance easily and increase the robustness of the system as whole. It is also noticed that for this specific target and selected commissioned axis the effort needed from RW of Z-axis is minor however the other RWs provide the main effort for carrying out the required tracking maneuver. Since the disturbance is not considered during simulation, it is logic to note that the angular momentums of the RWs are kept unchangeable after achieving the tracking task.
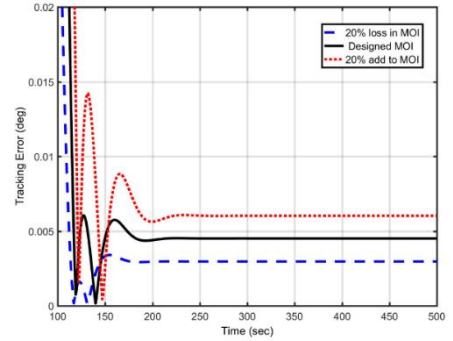

Figure 7. Wheel torque

Figure 8. Wheel momenta

Figure 9. Robustness check against uncertainty in MOI

Because of the feedback nature of the proposed C/GMRES controller, the robustness of this controller against model uncertainties is guaranteed. Simulations were also done to examine the robust performance of the proposed tracking controller against a ±20% error in the moment of inertia of the satellite. These simulations demonstrate that the precise tracking error can still be preserved to designed values ± 0.003 degree as shown in Figure 9. However, in practice, the TE will ultimately be impacted by the attitude determination accuracy as well as the orbital measurements of the position and velocity vectors from the GPS receiver which have been not considered in this study.

7. CONCLUSION

In this paper, a feasible solution to enhance the UoSAT-12 satellite shooting capabilities in GTT task is proposed. Employing the C/GMRES, as fast RTO methods of NMPC approach, along with a proposed control-oriented model, that converts the main tracking problem into a well-formulated simple regulation problem, is presented in details. The

simulation results show that proposed algorithm succeeds to track a fixed-ground target robustly through a large angle maneuver with a precise tracking error of 0.0045 deg, while maintaining the stability requirements within 0.1 deg/sec throughout the overpass flight. Therefore, the shooting mode capabilities are significantly enhanced.

REFERENCES

[1] Ohtsukan T. (2004). A continuation/GMRES method for fast computation of nonlinear receding horizon control. Journal of Automatica 40 (4): 563–574. https://doi.org/ 10.1016/j.automatica.2003.11.005

[2] Kelley CT. (1995). GMRES Iteration, Iterative methods for linear and nonlinear equations, (16), Philadelphia, Pa.: Society for Industrial and Applied Mathematics, North Carolina, pp. 33 – 60.

[3] Kouramas KI, Panos C, Faisca NP, Pistikopoulos EN. (2013). An algorithm for robust explicit/multi-parametric model predictive control. Journal of Automatica 49 (2): 381-389. https://doi.org/ 10.1016/j.automatica.2012.11.035

[4] Ohtsuka T. (2015). A tutorial on C/GMRES and automatic code generation for nonlinear model. European Control Conference, Linz, Austria, pp. 73-86. https://doi.org/ 10.1109/ECC.2015.7330528

[5] Tajeddin S, Vajedi M, Azad NL. (2016). A Newton/GMRES approach to predictive ecological adaptive cruise control of a plug-in hybrid electric vehicle in car-following scenarios, International Federation of Automatic Control 49 (21): 059–065.

[6] Richter SL, Decarlo RA. (1983). Continuation methods: Theory and applications, IEE. Transactions on Circuits and Systems 28(6): 660–665. https://doi.org/10.1109/TAC.1983.1103294

[7] Weiss H. (1993). Quaternion-based rate/attitude tracking system with application to gimbal attitude control. Journal of Guidance, Control, and Dynamics 16 (4) : 609-616. https://doi.org/ 10.2514/3.21057

[8] Chen XJ, Willem HS, Hashida Y. (2000). Ground-target tracking control of earth-pointing satellites. AIAA Guidance, Navigation And Control Conference, Denver, USA. https://doi.org/ 10.2514/6.2000-4547

[9] Vuuren GH, Willem SH. (2015). The design and simulation analysis of an attitude determination and control system for a small earth satellite. M.S thesis, Department of Electrical and Electronic Engineering, Stellenbosch University, Stellenbosch.

[10] Wahballah W, Bazan TM, El-tohamy F, Fathy M. (2016). Analysis of smear in high-resolution remote sensing satellites. SPIE, Edinburgh. UK, 1-11.

[11] Sun W. (2000). In-Orbit Results From UOSAT-12 Earth Observation Minisatellite Mission, IAA-B3-0305P, Guildford, Surrey, 1-4.