2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

A New Nonlinear Model Predictive Control Algorithm Using Differential Transformation with Application to Interplanetary Low-Thrust Trajectory Tracking

Rosemary Huang, Inseok Hwang, and Martin Corless

Abstract— **A nonlinear model predictive control algorithm (NMPC) is proposed that incorporates a novel differential transformation (DT) based algorithm for the repeated computation of the open-loop optimal control problem. This DT method converts the two-point boundary value problem from the indirect optimization approach into a set of algebraic equations, which is a less computationally expensive way of solving the open-loop problem. The new NMPC algorithm is applied to an interplanetary low-thrust trajectory tracking problem.**

I. INTRODUCTION

Ion engines have recently become popular as the primary means of propulsion for both earth-based satellites and interplanetary spacecraft [1], [2]. Their appeal is in their efficiency, which is ten times better than that of conventional chemical propulsion rocket engines. This results in decreased fuel cost, increased payload, longer spacecraft life, and/or feasibility of more types of interplanetary missions. Ion engines operate differently from conventional rocket engines. Chemical engines generate high thrust that only lasts a few minutes at most and is approximated as an impulse in trajectory design. In contrast, ion engines produce extremely small amounts of thrust, and operate for months. Consequently, ion engine thrust cannot be considered impulsive; thus new techniques are necessary for the design and control of low-thrust trajectories [3].

The bulk of the research into low-thrust trajectories so far has been on open-loop optimal trajectory design. It is a challenging problem due to the highly nonlinear dynamics with no closed form solution and long duration trajectories that could last for months or years. Thus, much effort has been devoted to developing new methods of finding low-thrust trajectories using both indirect and direct optimization approaches [4], [5].

Much less effort has been made in developing and applying nonlinear closed-loop control to ensure that a low-thrust spacecraft follows the designed path and reaches its destination [6]. This is important because there are many sources of disturbances on a spacecraft such as engine errors, unmodeled gravitational fields, atmospheric drag, and solar radiation pressure. Currently, NASA's ion propulsion spacecraft use linear controllers [7], which are easier to implement. However, they are not optimal nor robust for a nonlinear system. Nonlinear controllers are more challenging to design and implement, but they can be more optimal and robust than linear controllers.

There has been some effort applying nonlinear control theory to low-thrust spacecraft. Some have applied Lyapunov theory to the problem [8]. An approach using a combination of backstepping and forwarding has also been explored [9]. However, these approaches are not necessarily optimal.

A better technique to ensure a more optimal closed-loop controller is nonlinear model predictive control (NMPC). Linear model predictive control is considered a mature field, but NMPC is a relatively new area with serious attention only starting in the 1990s. The general concept behind NMPC is to solve an open-loop optimal control problem every time a new measurement is obtained for the state of the system. Then, the open-loop control input calculated is applied until the next measurement is available. Early NMPC techniques have already successfully been applied in other fields [10]. So far, there have been a few who have applied NMPC techniques to low-thrust transfer control. One applied NMPC with a direct transcription method for solving the open-loop problem [11]. Another combined NMPC with Rosenbrock's direct optimization method [12]. Both were for guidance on earth-centered low-thrust transfers.

One major disadvantage with NMPC implementation is the need to solve a nonlinear optimization problem online repeatedly. Optimization techniques currently used are computationally intensive. This limits the feasibility of applying NMPC. Recently, a new more efficient algorithm based on differential transformation (DT) has been developed by Hwang et al. [13] to solve nonlinear optimal control problems. It solves the two-point boundary value problem (TPBVP) from the indirect approach to solving an optimal control problem. The DT algorithm converts the TPBVP into a set of algebraic equations that require far less computational effort to solve. It has already been demonstrated to be effective in speeding up the process of finding optimal open-loop low-thrust trajectories [14]. This paper will introduce a new more efficient NMPC algorithm that uses the DT algorithm to repeatedly solve the nonlinear optimization problem, allowing NMPC to be more efficient. This algorithm is then applied to an interplanetary low-thrust trajectory tracking problem.

R. Huang is with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 47907, rhuang@purdue.edu

I. Hwang is with Faculty of School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 47907, ihwang@purdue.edu

M. Corless is with Faculty of School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 47907, corless@purdue.edu

978-1-4244-4524-0/09/$25.00 ©2009 AACC

This paper is organized as follows. Section II describes the dynamics for the low-thrust interplanetary transfer problem. Section III contains a description of the differential transformation algorithm. Section IV discusses the NMPC algorithm. In Section V, the NMPC algorithm is applied to a low-thrust interplanetary trajectory tracking problem. Section VI concludes the paper.

II. Low-Thrust Transfer Problem

A. Model

The interplanetary orbit transfer problem considered here is assumed to be between two coplanar orbits around the sun. Hence, the model will be two-dimensional. Additionally, Cartesian coordinates (x, y) were selected. Then the dynamic model for this problem is

$$\begin{array}{rcl} \dot{x_1} &=& x_3 \\ \dot{x_2} &=& x_4 \\ \dot{x_3} &=& -\frac{\mu x_1}{(x_1^2+x_2^2)^{\frac{3}{2}}} + a_x \\ \dot{x_4} &=& -\frac{\mu x_2}{(x_1^2+x_2^2)^{\frac{3}{2}}} + a_y \end{array} \tag{1}$$

where μ is the gravitational constant of the Sun; x_1, x_2, x_3, and x_4 are the states that correspond to x, y, $\dot{x}$, and $\dot{y}$, respectively; a_x is the x-component of the spacecraft acceleration, and a_y is the y-component.

B. Two-Point Boundary Value Problem

In this paper, we consider the cost function defined as follows from the given initial time t_0 to the given final time t_f

$$J = \int_{t_0}^{t_f} \frac{1}{2}(a_x^2 + a_y^2)dt \tag{2}$$

J will be minimized subject to the state equations (1), the given initial conditions $\mathbf{x}(t_0)$, and the given final conditions $\mathbf{x}(t_f)$. The state vector is $\mathbf{x} = [x_1\ x_2\ x_3\ x_4]^\top$.

Taking the indirect approach to solving this problem, the corresponding Hamiltonian is

$$H = \frac{1}{2}(a_x^2 + a_y^2) + \boldsymbol{\lambda}^\top \mathbf{f} \tag{3}$$

where $\boldsymbol{\lambda} = [\lambda_1\ \lambda_2\ \lambda_3\ \lambda_4]^\top$, $\mathbf{f} = [f_1\ f_2\ f_3\ f_4]^\top$, and $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$. The partial derivative of H with respect to $\mathbf{x}$ results in the following costate equations:

$$\dot{\boldsymbol{\lambda}} = -\left(\frac{\partial H}{\partial \mathbf{x}}\right)^\top \tag{4}$$

Let the control input vector be $\mathbf{a} = [a_x\ a_y]^\top$. In this paper, the input is assumed unconstrained. Then taking $\frac{\partial H}{\partial \mathbf{a}} = 0$ results in the following optimal control input

$$\mathbf{a}^* = \begin{bmatrix} a_x^* \\ a_y^* \end{bmatrix} = \begin{bmatrix} -\lambda_3 \\ -\lambda_4 \end{bmatrix} \tag{5}$$

The optimal inputs (5), the state (1) and costate (4) equations, and the boundary conditions together result in the following TPBVP:

$$\begin{array}{rcl} \dot{x_1} &=& x_3 \\ \dot{x_2} &=& x_4 \\ \dot{x_3} &=& -\frac{\mu x_1}{(x_1^2+x_2^2)^{\frac{3}{2}}} + a_x^* \\ \dot{x_4} &=& -\frac{\mu x_2}{(x_1^2+x_2^2)^{\frac{3}{2}}} + a_y^* \\ \dot{\lambda_1} &=& -\frac{\mu(2x_1^2-x_2^2)}{(x_1^2+x_2^2)^{\frac{5}{2}}}\lambda_3 - \frac{3\mu x_1 x_2}{(x_1^2+x_2^2)^{\frac{5}{2}}}\lambda_4 \\ \dot{\lambda_2} &=& -\frac{3\mu x_1 x_2}{(x_1^2+x_2^2)^{\frac{5}{2}}}\lambda_3 - \frac{\mu(-x_1^2+2x_2^2)}{(x_1^2+x_2^2)^{\frac{5}{2}}}\lambda_4 \\ \dot{\lambda_3} &=& -\lambda_1 \\ \dot{\lambda_4} &=& -\lambda_2 \\ a_x^* &=& -\lambda_3 \\ a_y^* &=& -\lambda_4 \end{array} \tag{6}$$

where $\mathbf{x}(t_0)$ and $\mathbf{x}(t_f)$ are the boundary conditions and the time span $[t_0\ t_f]$ is fixed. Solving this TPBVP will result in an optimal low-thrust trajectory.

C. Additional Equality Constraints from Numerical Algorithms

Many algorithms that solve the TPBVP discretize the time span $\Omega = [t_0\ t_f]$ into several smaller time intervals. Let $t_1, t_2, t_3, \cdots$ be time points in Ω. Then, the subintervals are defined as $\Omega_k = [t_{k-1}\ t_k]$, where $k = 1, 2, 3, \cdots$. The union of all the subintervals is the whole time span Ω, but they do not overlap. With this approach, additional equality constraints are necessary that relate the states and costates at the time point t_k to the states and costates immediately after the time point t_k^+. This will ensure that the algorithm will find a solution such that $\mathbf{x}(t)$ and $\boldsymbol{\lambda}(t)$ are continuous. For every $t_k \in \{t_1, t_2, t_3, \cdots\}$ where Ω is split, it is necessary to define the following set of equality constraints

$$\mathbf{x}(t_k) - \mathbf{x}(t_k^+) = 0 \tag{7}$$

$$\boldsymbol{\lambda}(t_k) - \boldsymbol{\lambda}(t_k^+) = 0 \tag{8}$$

Initially, these equality constraints will not be satisfied because $\mathbf{x}(t)$ and $\boldsymbol{\lambda}(t)$ at each t_k are unknown and must be guessed. A nonlinear program (NLP) will force these constraints to zero, where $\mathbf{x}(t_k)$ and $\boldsymbol{\lambda}(t_k)$ are NLP variables.

III. Differential Transformation Algorithm

The DT algorithm is discussed in this section. Initially, the definitions of DT and the inverse DT are presented. Then, the DT-based algorithm for solving the optimal control TPBVP is described.

A. Differential Transformation

The differential transformation method was originally introduced for the analysis of electrical circuits in the 1980's by Pukhov and Zhou [15], [16]. It has been applied in a few other areas since then, such as vibration analysis [17], [18], [19]. Motivations for using DT are a reduction in computational effort required for solving a set of differential algebraic equations (DAE's) and having a solution in an analytical form. By differential operations, DT transforms the DAE's from the time/space domain into a set of nonlinear algebraic equations in a transformed domain. Now the solution can

TABLE I

SOME DIFFERENTIAL TRANSFORMATION RELATIONSHIPS

$z(t)$	$Z_{t_e}(i)$ for $i = 0, 1, \cdots$ in the DT-domain
$cp(t) \pm q(t)$	$cP_{t_e}(i) \pm Q_{t_e}(i)$
$p(t)q(t)$	$P_{t_e}(i) * Q_{t_e}(i)$
$\frac{dp(t)}{dt}$	$(i+1)P_{t_e}(i+1)/S$
$\frac{1}{p(t)}$	$\frac{1}{P_{t_e}(0)}$ for $i = 0$ $\frac{-\left[\sum_{l=1}^{i} Z_{t_e}(i-l)P_{t_e}(l)\right]}{P_{t_e}}(0)$ for $i \geq 1$
$\sqrt{p(t)}$	$\sqrt{P_{t_e}(0)}$ for $i = 0$, $\frac{P_{t_e}(1)}{2Z_{t_e}(0)}$ for $i = 1$, $\frac{P_{t_e}(i) - \sum_{l=1}^{i-1} Z_{t_e}(i-l)Z_{t_e}(l)}{2Z_{t_e}(0)}$ for $i \geq 2$

be computed with a NLP. Recovery of the solution in the original time/space domain is done by applying the inverse DT to the transformed domain solution. The result will be in an analytical form, finite-term series approximations.

1) Definition of Differential Transformation: Let $z(t)$ be a (piecewise) smooth function on $[t_0\ t_f] \subset \Re$. Define an *expansion point* to be $t_e \in [t_0\ t_f]$. The images of $z(t)$ at t_e in the transformed domain (*DT-domain*) are

$$Z_{t_e}(i) = \frac{S^i}{i!}\left(\frac{d^i z(t)}{dt^i}\right)_{t=t_e} \qquad i = 0, 1, 2, \cdots \tag{9}$$

where $Z_{t_e}(i)$ is the i^{th} *order differential spectrum* of $z(t)$ at t_e, and S is the *spectrum scaling factor*, a positive constant.

The DT of $z(t)$ at t_e is a sequence of i^{th} order partial derivatives of the function, but explicitly taking the partial derivatives of $z(t)$ is unnecessary. Instead, one can use basic relationships, some are in Table I, to transform a function. A more complete list is in [14]. For Table I, $c \in \Re$ is a constant, $p(t)$ and $q(t)$ are (piecewise) smooth functions on $[t_0\ t_f]$, and $P_{t_e}(i) * Q_{t_e}(i) \equiv \sum_{l=0}^{i} P_{t_e}(l)Q_{t_e}(i-l)$.

2) Definition of Inverse Differential Transformation: The form of the inverse DT depends on the basis system selected. For this paper, $\{t^i\}_{i=0}^{\infty}$ is chosen as the basis system. Then, the inverse DT of the differential spectra $\{Z_{t_e}(i)\}_{i=0}^{\infty}$ is

$$z(t) = \sum_{i=0}^{\infty} Z_{t_e}(i)\left(\frac{t - t_e}{S}\right)^i \quad \forall t \in (t_0, t_f) \tag{10}$$

This definition allows $z(t)$ to be approximated as a finite-term Taylor series expansion about t_e. To observe this, substitute (9) into (10) to obtain

$$\begin{aligned} z(t) &= \textstyle\sum_{i=0}^{\infty} \left(\frac{d^i z}{dt^i}\right)_{t=t_e} \frac{(t-t_e)^i}{i!} \\ &= \textstyle\sum_{i=0}^{N} \left(\frac{d^i z}{dt^i}\right)_{t=t_e} \frac{(t-t_e)^i}{i!} + R_N \end{aligned} \tag{11}$$

Here, N is the number of terms to be kept in the Taylor series, and R_N is the remainder as derived from Taylor's theorem:

$$R_N = \left(\frac{d^{N+1} z}{dt^{N+1}}\right)_{t=\zeta} \frac{(t - t_e)^{N+1}}{N + 1!} \tag{12}$$

where ζ is a point between t and t_e.

B. DT-Based Algorithm for the Two-Point Boundary Value Problem

In this section we describe the DT algorithm for solving the TPBVP. It consists of an initialization step and three main steps. The reformulation of the TPBVP into its DT form will need to be done only once. It is complicated, but the payoffs are a reduction in the amount of calculations needed for analyzing a dynamical system and a solution in analytical form.

Initialization: The low-thrust transfer problem requires the application of DT over a large time span $\Omega \equiv [t_0\ t_f]$. Necessarily, Ω is discretized into subintervals Ω_k as described earlier. Choose $K-1$ expansion points, $t_{e,1}, t_{e,2}, \cdots, t_{e,K-1}$, such that $t_0 < t_{e,1} < t_{e,2} < \cdots < t_{e,K-1} < t_f$. These time points divide the time span Ω into K subintervals:

$$\begin{aligned} \Omega_1 &= [t_0\ t_{e,1}] \\ \Omega_k &= (t_{e,k-1}\ t_{e,k}] \text{ for } k = 2, 3, \cdots, K-1 \\ \Omega_K &= (t_{e,K-1}\ t_f] \end{aligned}$$

Select the order N of the DT spectrum, which also determines the order of Taylor series approximation of the solution. Additionally, the spectrum scaling factor S is set at this point, usually $S = 1$.

Step 1: Conversion of a TPBVP into an Algebraic Problem: This step converts the TPBVP from a system of differential equations and boundary conditions into a set of algebraic difference equations. It is done in three substeps.

a. *Recursive Equations*: Apply DT (9) to the TPBVP differential equations (6) to transform them into the set of recursive difference equations in the DT-domain as shown in the Appendix. This allows the $i+1^{\text{th}}$ spectral terms at $t_{e,k}$ to be a function of $\mathbf{X}_{t_{e,k}}(m)$,$\mathbf{\Lambda}_{t_{e,k}}(m)$, and $\mathbf{A}^*{}_{t_{e,k}}(m)$ for $m = 0, 1, 2, \cdots, i$. This allows for rapid calculation of all spectral terms of orders $1, \cdots, N$ from only the 0^{th} order terms, enabling N to be extremely high.

b. *Boundary Condition Equations*: The equations derived in this substep enforce the boundary conditions. First, apply the inverse DT (10) to obtain N-term Taylor series expansions of $\mathbf{x}(t)$ at $t_{e,1}$ and $t_{e,K-1}$. Then, equate these expansions to the initial and final conditions, respectively, to obtain

$$\mathbf{x}(t_0) \approx \sum_{i=0}^{N} \mathbf{X}_{t_{e,1}}(i)(t_0 - t_{e,1})^i \tag{13}$$

$$\mathbf{x}(t_f) \approx \sum_{i=0}^{N} \mathbf{X}_{t_{e,K-1}}(i)(t_f - t_{e,K-1})^i \tag{14}$$

c. *Continuity Constraints*: This substep uses the equality constraints (7)-(8) to derive a set of algebraic relations between the differential spectra at neighboring expansion points. First, note that $X_{t_{e,k}}(0) = x(t_{e,k})$ for all expansion points. Application of DT and inverse DT to the equality constraints results in

$$\mathbf{X}_{t_{e,k}}(0) = \sum_{i=0}^{N} \mathbf{X}_{t_{e,k+1}}(i)(t_{e,k} - t_{e,k+1})^i \quad (15)$$

$$\mathbf{\Lambda}_{t_{e,k}}(0) = \sum_{i=0}^{N} \mathbf{\Lambda}_{t_{e,k+1}}(i)(t_{e,k} - t_{e,k+1})^i \quad (16)$$

Step 2: Solution of the Nonlinear Algebraic Equations: The solution to the TPBVP is obtained in terms of the differential spectra in the DT-domain in this step. The nonlinear algebraic problem to be solved is defined by the recursive equations, the boundary condition equations, and the continuity equations. $\mathbf{X}_{t_{e,k}}(0)$, $\mathbf{\Lambda}_{t_{e,k}}(0)$, and $\mathbf{A}_{t_{e,k}}(0)$ at each $t_{e,k}$, $k = 1, 2, \cdots, K-1$ are the variables to solve for. Then, actual solution values $\mathbf{X^*}_{t_{e,k}}(0)$, $\mathbf{\Lambda^*}_{t_{e,k}}(0)$, and $\mathbf{A^*}_{t_{e,k}}(0)$ are used to find the values of the higher-order spectral terms for the solution $\{\mathbf{X^*}_{t_{e,k}}(i), \mathbf{\Lambda^*}_{t_{e,k}}(i), \mathbf{A^*}_{t_{e,k}}(i)\}_{i=1}^{N}$ by reusing the recursive equations.

Step 3: Conversion to Time Domain Solution: After calculating the solution of the TPBVP in the DT-domain, now $\mathbf{x}^*(t)$, $\boldsymbol{\lambda}^*(t)$, and $\mathbf{a}^*(t)$ need to be recovered in the original time/space domain. Applying the inverse DT, the solution is in the form of an N^{th}-order Taylor series expansion of each variable for each of the K time subintervals. For the states $\mathbf{x}^*(t)$, this results in

$$\begin{aligned}
\mathbf{x}^*(t) &= \sum_{i=0}^{N} \mathbf{X^*}_{t_{e,k}}(i)(t - t_{e,k})^i, t \in \Omega_{k=1,\cdots,K-1} \\
\mathbf{x}^*(t) &= \sum_{i=0}^{N} \mathbf{X^*}_{t_{e,K-1}}(i)(t - t_{e,K-1})^i, t \in \Omega_K
\end{aligned} \quad (17)$$

The expressions for $\boldsymbol{\lambda}^*(t)$ and $\mathbf{a}^*(t)$ are computed similarly.

IV. Nonlinear Model Predictive Control Algorithm

This section presents an NMPC algorithm using the DT method. First, a brief introduction into important NMPC concepts is given. Then the algorithm is described.

A. Concepts

Nonlinear model predictive control is a strategy for closed-loop control where at every state measurement, a nonlinear open-loop optimal control problem is solved. Figure 1 illustrates general NMPC concepts. At time t measurements of the state $\mathbf{x}(t)$ are obtained. Based on these states and a nonlinear dynamic model of the system, the behavior of the system $\hat{\mathbf{x}}$ and open-loop optimal control $\hat{\mathbf{u}}$ is predicted over the prediction horizon T_p. The input $\hat{\mathbf{u}}$ is then applied until the next measurement is available after a sampling interval δ. With the new measurement at $t + \delta$, the whole procedure repeats.

An essential part of any NMPC approach is the assurance of stability. There are several NMPC strategies that guarantee stability [10]. One of these is the zero terminal equality constraint strategy, which this paper uses. This method involves enforcing a hard constraint on the states at $t + T_p$, achievable by imposing boundary conditions at $t + T_p$ [11]. In this paper, this strategy is applied by forcing the boundary condition of the state at $t+T_p$ to be on the nominal trajectory.

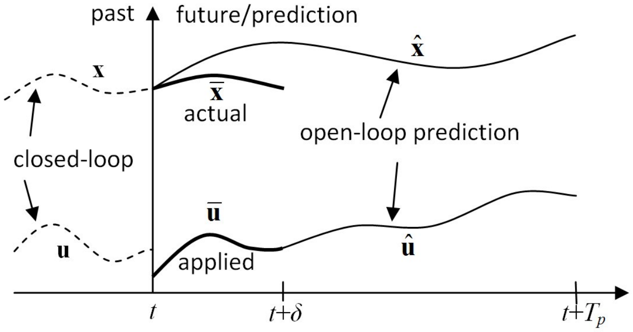

Fig. 1. Schematic of model predictive control [10].

B. DT-Based NMPC Algorithm

This section presents an NMPC algorithm for tracking a nominal low-thrust trajectory. It takes advantage of the differential transformation in two ways. First, the nominal trajectory is itself a solution to an open-loop optimal control problem, and DT can be applied to find an analytical representation of it in the form of equation (17). Second, the DT algorithm is used to quickly solve the open-loop optimal control problem within each NMPC iteration.

Initialization: Obtain the DT realization of the nominal trajectory. This includes the expansion points $t_{e,k}$ and the spectral terms of the states and costates, $\mathbf{X}_{t_{e,k}}^{nom}(k)$ and $\mathbf{\Lambda}_{t_{e,k}}^{nom}(k)$ for $k = 0, 1, 2, \cdots$, at all the expansion points. Additionally, choose an initial prediction horizon T_p.

Step 1: Measure $\mathbf{x}$ at the current time t.

Step 2: Find the boundary condition for the zero terminal constraint. Determine the target state $\mathbf{x}_{nom}(t + T_p)$ on the nominal trajectory at the end of the prediction horizon T_p. This can be done by substituting the time $t + T_p$ into the inverse DT of the nominal trajectory (17). Set the boundary condition to $\hat{\mathbf{x}}(t + T_p) = \mathbf{x}_{nom}(t + T_p)$. Note that if $T_f < t+T_p$, then T_p is reduced to $T_p = T_f - t$ and the target state $\hat{\mathbf{x}}(t+T_p)$ will then be the endpoint of the nominal trajectory $\mathbf{x}_{nom}(T_f)$.

Step 3: With the current measured state $\mathbf{x}(t)$ as the initial condition, and the target state $\hat{\mathbf{x}}(t + T_p)$, found in Step 2, as the final condition, solve the low-thrust TPBVP (6) using the DT algorithm. The initial guesses for the 0^{th}-order spectral terms are the states and costates of the nominal trajectory at the times corresponding to the expansion points, $\mathbf{X}_{t_{e,k}}(0) = \mathbf{x}_{nom}(t_{e,k})$ and $\mathbf{\Lambda}_{t_{e,k}}(0) = \boldsymbol{\lambda}_{nom}(t_{e,k})$. These can be quickly obtained using the DT representation of the nominal trajectory and equation (17). The solution to the TPBVP will result in an open-loop optimal control input $\hat{\mathbf{u}}(t)$.

Step 4: Apply the control input $\hat{\mathbf{u}}(t)$ until the next sampling time $t + \delta$ at the end of the sampling interval δ or until the end time of the transfer T_f, whichever is sooner.

Step 5: Return to Step 1

V. Application to a Low-Thrust Trajectory Tracking Problem

Here, the NMPC with DT algorithm is applied to a low-thrust spacecraft trajectory tracking problem. The nominal trajectory is a 420-day continuous thrust transfer from Earth to Mars optimized according to the cost defined in (2). The orbits of Earth and Mars are assumed to be circular and coplanar. The nominal trajectory begins at true anomaly $\nu = 0°$ at Earth orbit and ends at $\nu = 300°$ at Mars orbit. True anomaly is the angle of the spacecraft position vector in the xy-plane. The nominal trajectory is shown in Fig. 2 along with its acceleration profile. Within the NMPC implementation, this nominal trajectory was represented in DT form by 45th-order expansions at 10 evenly spaced expansion points.

Notice that the acceleration magnitude remains extremely small throughout the trajectory. Ion engines produce extremely small accelerations. For example, NEXT ion engines will be capable of a maximum thrust of only 236 mN [20]. Deep Space 1 (DS1), NASA's first interplanetary ion engine spacecraft, had an initial mass of 486.3 kg and a mass of 404.8 kg without the xenon fuel [21]. If DS1 had used the NEXT engines, its maximum acceleration magnitude would have ranged from 4.85x10^{-7} km/s^2 to 5.83x10^{-7} km/s^2.

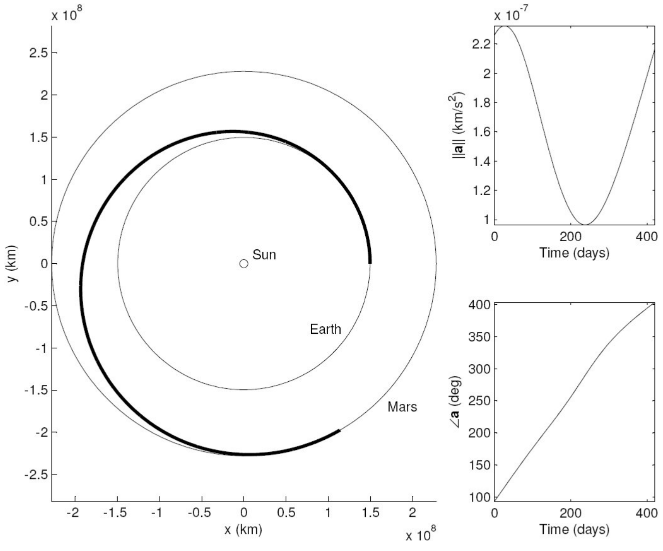

Fig. 2. Nominal Trajectory and Acceleration Profile

While following the nominal trajectory, the spacecraft is subjected to 2-day total engine failure from day 100 to day 102, where there is no control input available. The goal of the NMPC algorithm is to track the nominal trajectory in spite of this disturbance. In this implementation, the prediction horizon T_p is initially 70 days. Near the end of the trajectory, this is gradually reduced as described in Step 2 of the NMPC algorithm. The sampling interval is 2 days. For solving the open-loop TPBVP at each NMPC iteration, the number of expansion points used in the DT algorithm is initially 12 with 45th-order expansions. Towards the end of the nominal trajectory, this is gradually reduced to 1.

The results from simulating the above scenario are shown in Fig. 3. In this figure, the error is defined as $||\mathbf{q}_A - \mathbf{q}_N||$, where $\mathbf{q}_A$ is the actual spacecraft position and $\mathbf{q}_N$ the desired spacecraft position, with $\mathbf{q} = [x_1\ x_2]^\top$. The top left plot is the error over the entire 420-day span of the simulation. On the bottom left is the closed-loop input magnitude over the same time span. The plots on the right are close-ups of the error and input magnitude response to the engine failure. Also included in the control input close-up plot is the input magnitude for the nominal trajectory (dotted line).

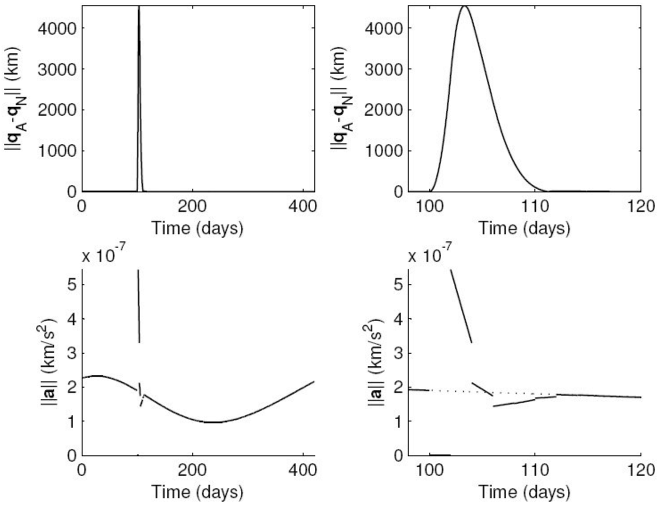

Fig. 3. NMPC Trajectory Tracking Results (solid lines: NMPC with DT, dotted line: nominal trajectory)

The NMPC algorithm succeeds in rejecting the disturbance and in tracking the nominal trajectory. Observe that there is an increase in the error starting at 100 days when the engine fails and control input is not available. Then, with the engine operational on day 102, the NMPC algorithm is able to reduce the error and steer the spacecraft back onto the nominal trajectory. The transient behavior shows a maximum deviation from the nominal trajectory of 4548 km and an overshoot of only 15 km. Additionally, 9 days from when the engine is operational, the error settles to below 20 km. The magnitude of the input spikes to 5.445x10^{-7} km/s^2 when the NMPC corrects for the large deviation from the nominal trajectory due to the engine failure, but it remains within the maximum acceleration range discussed earlier. Note that the NMPC input plot is segmented because at every sampling time the control input is recalculated and a discontinuity in the control input can occur at the sampling time.

VI. Conclusion

A nonlinear model predictive control algorithm has been proposed in this paper that utilizes a new differential transformation based algorithm to solve the on-line open-loop optimization problem within each iteration of the controller. Typical methods of solving the open-loop problem are computationally intensive and take a significant amount time to find a solution. The DT algorithm proposed transforms the TPBVP from the indirect approach into a system of nonlinear algebraic equations where the solution to the optimal control problem is less computationally expensive. This makes NMPC more efficient for on-line application of closed-loop trajectory tracking in spacecraft and in other applications. The algorithm was demonstrated to be effective at tracking a low-thrust interplanetary trajectory.

APPENDIX

Here we present the recursive equations for the coplanar low-thrust transfer problem being considered in this paper. They represent the TPBVP's differential equations in the DT-domain.

$$\begin{bmatrix} X_{1,t_{e,k}}(i+1) \\ X_{2,t_{e,k}}(i+1) \\ X_{3,t_{e,k}}(i+1) \\ X_{4,t_{e,k}}(i+1) \\ \Lambda_{1,t_{e,k}}(i+1) \\ \Lambda_{2,t_{e,k}}(i+1) \\ \Lambda_{3,t_{e,k}}(i+1) \\ \Lambda_{4,t_{e,k}}(i+1) \\ A_{x,t_{e,k}}(i) \\ A_{y,t_{e,k}}(i) \end{bmatrix} = \frac{S}{i+1} \begin{bmatrix} X_{3,t_{e,k}}(i) \\ X_{4,t_{e,k}}(i) \\ F_{31,t_{e,k}}(i) + A_{x,t_{e,k}}(i) \\ F_{41,t_{e,k}}(i) + A_{y,t_{e,k}}(i) \\ F_{51,t_{e,k}}(i) + F_{52,t_{e,k}}(i) \\ F_{61,t_{e,k}}(i) + F_{62,t_{e,k}}(i) \\ -\Lambda_{1,t_{e,k}}(i) \\ -\Lambda_{2,t_{e,k}}(i) \\ -\Lambda_{3,t_{e,k}}(i) \\ -\Lambda_{4,t_{e,k}}(i) \end{bmatrix}$$

$$\begin{aligned} F_{31,t_{e,k}}(i) &= -\mu X_{1,t_{e,k}}(i) * Y_{r3,t_{e,k}}(i) \\ F_{41,t_{e,k}}(i) &= -\mu X_{2,t_{e,k}}(i) * Y_{r3,t_{e,k}}(i) \\ F_{51,t_{e,k}}(i) &= -\mu Y_{d1r5,t_{e,k}}(i) * \Lambda_{3,t_{e,k}}(i) \\ F_{52,t_{e,k}}(i) &= -3\mu Y_{12r5,t_{e,k}}(i) * \Lambda_{4,t_{e,k}}(i) \\ F_{61,t_{e,k}}(i) &= -3\mu Y_{12r5,t_{e,k}}(i) * \Lambda_{3,t_{e,k}}(i) \\ F_{62,t_{e,k}}(i) &= -\mu Y_{d2r5,t_{e,k}}(i) * \Lambda_{4,t_{e,k}}(i) \end{aligned}$$

$$\begin{aligned} Y_{d1r5,t_{e,k}}(i) &= D_{1,t_{e,k}}(i) * Y_{r5,t_{e,k}}(i) \\ Y_{d2r5,t_{e,k}}(i) &= D_{2,t_{e,k}}(i) * Y_{r5,t_{e,k}}(i) \\ Y_{12r5,t_{e,k}}(i) &= X_{1t2,t_{e,k}}(i) * Y_{r5,t_{e,k}}(i) \end{aligned}$$

$$\begin{aligned} Y_{r3,t_{e,k}}(0) &= \tfrac{1}{R_{cu,t_{e,k}}(0)} \text{ for } i = 0 \\ Y_{r3,t_{e,k}}(i) &= \tfrac{[-\sum_{l=1}^{i} Y_{r3,t_{e,k}}(i-l) R_{cu,t_{e,k}}(l)]}{R_{cu,t_{e,k}}(0)} \text{ for } i \geq 1 \\ Y_{r5,t_{e,k}}(0) &= \tfrac{[-\sum_{l=1}^{i} Y_{r5,t_{e,k}}(i-l) R_{5th,t_{e,k}}(l)]}{R_{5th,t_{e,k}}(0)} \text{ for } i = 0 \\ Y_{r5,t_{e,k}}(i) &= \tfrac{1}{R_{5th,t_{e,k}}(0)} \text{ for } i \geq 1 \end{aligned}$$

$$\begin{aligned} R_{cu,t_{e,k}}(i) &= R_{t_{e,k}}(i) * R_{sq,t_{e,k}}(i) \\ R_{5th,t_{e,k}}(i) &= R_{sq,t_{e,k}}(i) * R_{cu,t_{e,k}}(i) \end{aligned}$$

$$\begin{aligned} R_{t_{e,k}}(0) &= \sqrt{R_{sq,t_{e,k}}(0)} \text{ for } i = 0, \\ R_{t_{e,k}}(1) &= \tfrac{R_{sq,t_{e,k}}(1)}{2R_{t_{e,k}}(0)} \text{ for } i = 1 \\ R_{t_{e,k}}(i) &= \tfrac{[R_{sq,t_{e,k}}(i) - \sum_{l=1}^{i-1} R_{t_{e,k}}(i-l) R_{t_{e,k}}(l)]}{2R_{t_{e,k}}(0)} \text{ for } i \geq 2 \end{aligned}$$

$$\begin{aligned} D_{1,t_{e,k}}(i) &= 2X_{1sq,t_{e,k}}(i) - X_{2sq,t_{e,k}}(i) \\ D_{2,t_{e,k}}(i) &= 2X_{2sq,t_{e,k}}(i) - X_{1sq,t_{e,k}}(i) \\ R_{sq,t_{e,k}}(i) &= X_{1sq,t_{e,k}}(i) + X_{2sq,t_{e,k}}(i) \end{aligned}$$

$$\begin{aligned} X_{1sq,t_{e,k}}(i) &= X_{1,t_{e,k}}(i) * X_{1,t_{e,k}}(i) \\ X_{2sq,t_{e,k}}(i) &= X_{2,t_{e,k}}(i) * X_{2,t_{e,k}}(i) \\ X_{1t2,t_{e,k}}(i) &= X_{1,t_{e,k}}(i) * X_{2,t_{e,k}}(i) \end{aligned}$$

REFERENCES

[1] M. Patterson, "Ion Propulsion," NASA Factsheet, FS-2004-11-021-GRC, Glenn Research Center, Nov. 2004, http://www.nasa.gov/centers/glenn/about/fs21grc.html

[2] D.A. Lichtin, "An Overview of Electric Propulsion Activities in US Industry - 2005," *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, AIAA Paper 2005-3532, Tucson, AZ, 2005, pp. 1-10.

[3] G.D. Racca, "New Challenges to Trajectory Design by the Use of Electric Propulsion and Other New Means of Wandering in the Solar System," *Celestial Mechanics and Dynamical Astronomy*, Vol. 85, 2003, pp. 1-24.

[4] A.L. Herman and D.B. Spencer, "Optimal, Low-Thrust Earth-Orbit Transfers Using Higher-Order Collocation Methods," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 40-47.

[5] S.Lee, P. von Allmen, W. Fink, A.E. Petropoulos, and R.J. Terrile, "Design and Optimization of Low-thrust Orbit Transfers," *Aerospace Conference, 2005 IEEE*, pp. 855 - 869.

[6] P. Gurfil, "Control-Theoretic Analysis of Low-Thrust Orbital Transfer Using Orbital Elements," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 6, 2003, pp. 979-983.

[7] S.D. Desai, S. Bhaskeran, W.E. Bollman, C.A. Halsell, J.E. Riedel, and S.P. Synnott, "The DS-1 Autonomous Navigation System: Autonomous Control of Low Thrust Propulsion System," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 1997-3819, 1997, pp. 639-649.

[8] D.E. Chang, D.F. Chichka, and J.E. Marsden, "Lyapunov-Based Transfer Between Elliptic Keplerian Orbits," *Discrete and Continuous Dynamical Systems - Series B*, Vol. 2, No. 1, 2002, pp. 57-67.

[9] C.M. Kellett and L. Praly, "Nonlinear Control Tools for Low Thrust Orbital Tranfer," *6th IFAC Symposium on Nonlinear Control Systems*, Institut fur Systemtheorie, 2004, pp. 81-88.

[10] R. Findeisen and F. Allgower, "An Introduction to Nonlinear Model Predictive Control," http://www.wfw.wtb.tue.nl/benelux2002/nmpc.pdf

[11] J.J. Arrieta-Camacho and L.T. Biegler, "Real Time Optimal Guidance of Low-Thrust Spacecraft: An Application of Nonlinear Model Predictive Control," *Annals of the New York Academy of Sciences*, Vol. 1065, 2005, pp. 174-188.

[12] Y. Gao, "Low-Thrust Nonlinear Guidance by Tracking Mean Orbital Elements," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, 2008, pp. 1103-1110.

[13] I. Hwang, J. Li, and D. Du, "A Numerical Algorithm for Optimal Control of a Class of Hybrid Systems: Differential Transformation Based Approach," *International Journal of Control*, Vol. 8, No. 2, 2008, pp. 277-293.

[14] R. Huang, I. Hwang, and M. Corless, "Application of Differential Transformation Based Algorithm to Find Optimal Interplanetary Low-Thrust Transfers," *AIAA/AAS Astrodynamics Specialist Conference*, AIAA-2008-7364, 2008.

[15] G.E. Pukhov, *Differential Transforms of Functions and Equations*, Naukova Dumka (in Russia), Kiev, 1980.

[16] J. Zhou, *Differential Transformation and Its Application for Electrical Circuits*, Huazhong University Press (in Chinese), 1986.

[17] C. Chen, and S. Ho, "Transverse Vibration of a Rotation Twisted Timoshenko Beam Under Axial Loading Using Differential Transformation," *International Journal of Mechanical Sciences*, Vol. 41, 1999, pp. 1339-1356.

[18] C. Bert, and H. Zeng, "Analysis of Axial Vibration of Compound Bars by Differential Transformation Method," *Journal of Sound and Vibration*, Vol. 275, 2004, pp. 641-647.

[19] C. Mei, "Differential Transformation Approach for Free Vibration Analysis of a Centrifugally Stiffened Timoshenko Beam," *Journal of Vibration and Acoustics*, Vol. 128, 2006, pp. 170-175.

[20] "Advanced Electric Propulsion," NASA Factsheet, ISPT-2007-12-05-GRC, http://www.grc.nasa.gov/WWW/InSpace

[21] M.D. Rayman and S.N. Williams, "Design of the First Interplanetary Solar Electric Propulsion Mission," *Journal of Spacecraft and Rockets*, Vol. 39, No. 4, 2002, pp. 589-595.