IEEE Transactions on Aerospace and Electronic Systems, available online 25 January 2018
DOI: 10.1109/TAES.2018.2798219

Optimal Tracking Guidance for Aeroassisted Spacecraft Reconnaissance Mission based on Receding Horizon Control

Runqi Chai, *Student Member, IEEE,* Al Savvaris, *Member, IEEE,* Antonios Tsourdos, *Member, IEEE,* Senchun Chai, and Yuanqing Xia, *Senior Member, IEEE,*

***Abstract*—This paper focuses on the application of model predictive control (MPC) for the spacecraft trajectory tracking problems. The motivation of the use of MPC, also known as receding horizon control, relies on its ability in dealing with control, state and path constraints that naturally arise in practical trajectory planning problems. Two different MPC schemes are constructed to solve the reconnaissance trajectory tracking problem. Since the MPC solves the online optimal control problems at each sampling instant, the computational cost associated with it can be high. In order to decrease the computational demand due to the optimization process, a newly proposed two-nested gradient method is used and embedded in the two MPC schemes. Simulation results are provided to illustrate the effectiveness and feasibility of the two MPC tracking algorithms combined with the improved optimization technique.**

***Index Terms*—Model predictive control, spacecraft trajectory tracking, receding horizon control, optimal control, two-nested gradient method**

I. Introduction

OVER the past couple of decades, aeroassisted spacecrafts have received considerable attention due to their extensive applications in space exploration [1], [2]. One important feature of using this type of vehicle is that it has the capability to apply the aerodynamic forces effectively [3]–[5]. Early works on developing the aeroassisted spacecraft mainly focus on the propulsion system and orbital transfer trajectory design [6]–[10]. For example, Rao et al. [11] formulated a multiple-pass aeroassisted orbital transfer problem and generated the optimal trajectory via numerical optimization techniques. In [12], the authors studied a small-scale aeroassisted orbital transfer problem using impulsive thrust. Meanwhile, many important research works focusing on the aeroassisted vehicle guidance system have been extensively investigated. For instance, Hull et al. [13] proposed an energy-optimal guidance strategy for the aeroassisted orbital plane change problem. Naidu et al. [14] designed a neighbouring optimal guidance scheme for the nonlinear aeroassisted vehicle dynamics.

While a large amount work has been carried out in the aeroassisted vehicle guidance system, due to the uncertainties in the flight environment and multiple constraints, it is still hard to design a robust online guidance algorithm such that the vehicle can fly along an optimum path and fulfill different mission requirements. In general, guidance methods for space vehicles can be divided into two categories: predictor-corrector based methods [15], and reference tracking based methods [16]. In a predictor-corrector based method, the control commands are obtained by a predesigned guidance law and a control reversal logic. However, the implementation of these techniques has some challenges. At each time instant, the algorithm needs to predict the flight path and adjust the design parameters to steer the final condition errors to zero. This process is usually time-consuming and cannot be computed in near real-time.

Alternatively, reference tracking methods are based on the reference trajectory, which can be carried out offline via well-developed trajectory optimization techniques [17], [18]. The aim of these methods is to seek the control command such that the actual trajectory can follow the references. Recent studies have shown the effectiveness of developing a reference tracking guidance method in real time. For example, Dai and Xia [19] applied a terminal sliding mode control to derive the guidance law and an extended state observer to handle the model errors. In [16], the authors designed a lateral path tracking control method to general the control command for the autonomous land vehicle.

The problem addressed in this research is a real-time MPC design for constrained spacecraft trajectory tracking problems, where the objective function is a combination of control efforts and tracking errors. These kind of problems are becoming popular in spacecraft navigation system. The motivation for the use of receding horizon control (RHC) or MPC relies on its ability in dealing with control, state and path constraints. The current control command is obtained by solving online, at each time instant, a finite horizon open-loop optimal control problem. Then an optimal control sequence can be calculated and the first control action in the sequence is implemented to the vehicle dynamics. Contributions made to apply MPC can be found in the literature. Specifically, in [20] the authors developed a specific numerical algorithm for nonlinear receding horizon control problems. Peng et al. [21] calculated the optimal guidance law for a spacecraft formation reconfiguration problem by applying a nonlinear model predictive control (NMPC). In addition, an indirect legendre pseudospectral method was proposed in [22] to calculate the

R. Chai, A. Savvaris and A. Tsourdos are with the School of Aerospace, Transport and Manufacturing, Cranfield University, UK, e-mail: (r.chai@cranfield.ac.uk), (a.savvaris@cranfield.ac.uk), and (a.tsourdos@cranfield.ac.uk).

S. Chai and Y. Xia are with the school of Automation, Beijing Institute of Technology, Beijing, China, e-mail: (chaisc97@163.com), (xia_yuanqing@bit.edu.cn).

©2018 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

optimal control command for reentry vehicles. Similarly, in [23] authors applied a modified linear model predictive control (LMPC) to compute the optimal guidance for a low-thrust orbital transfer problem. Motivated by relative works, in this paper, two different tracking MPC schemes are constructed to generate the optimal guidance command for the aeroassisted vehicle and this will be discussed in more detail in Section III of this paper.

One of the key components of the MPC is the optimization process [24], [25]. Since the online tracking MPC algorithm attempts to solve an open-loop optimal control problem recursively, the effectiveness and efficiency are largely affected by its optimization procedure. In order to enhance the online performance of the MPC schemes constructed in Section III, a newly proposed optimization technique is applied and embedded in the MPC framework. This algorithm contains a two nested structure, where the inner loop uses an interior point method and the outer loop is a standard sequential quadratic programming. A detailed description of this improved gradient-based method can be found in [26]. Applying this technique, the complicated quadratic programming solution-finding can be avoided by solving the subproblem inexactly. This indicates that the online optimization performance of the MPC can be improved.

The remainder of this paper is organized as follow: The overall guidance strategy and the aeroassisted spacecraft guidance problem are constructed in Section II. Two receding horizon control schemes for the online tracking problem are formulated in Section III. Following that, Section IV presents the two nested gradient optimization algorithm used in the MPC optimization process. Numerical simulations are provided in Section V to illustrate the effectiveness of the tracking guidance strategies investigated in this paper.

II. Guidance Strategy

For the reconnaissance trajectory tracking problem, the kinematics of the aeroassisted spacecraft used in this paper are given by:

$$\begin{aligned} \dot{r} &= V\sin\gamma \\ \dot{\theta} &= \frac{V\cos\gamma\sin\psi}{r\cos\phi} \\ \dot{\phi} &= \frac{V\cos\gamma\cos\psi}{r} \end{aligned} \tag{1}$$

where r represents the radial distance. θ and ϕ are the longitude and latitude, respectively.

Consider the angle of attack α and bank angle σ of the vehicle as control inputs, the dynamics of the spacecraft can then be formulated as:

$$\begin{aligned} \dot{V} &= \frac{T\cos\alpha - D}{m} - g\sin\gamma \\ \dot{\gamma} &= \frac{L\cos\sigma + T\sin\alpha}{mV} + \left(\frac{V^2 - gr}{rV}\right)\cos\gamma \\ \dot{\psi} &= \frac{L\sin\sigma}{mV\cos\gamma} + \frac{V}{r}\cos\gamma\sin\psi\tan\phi \end{aligned} \tag{2}$$

where V is the relative velocity, γ is the flight path angle. ψ stands for the heading angle. m is the vehicle's mass. It is assumed that the state variables r, θ, ϕ, V, γ and ψ can be measured or obtained (Assumption.1 [24], [27]). The control objective is to design a guidance system so as to steer the spacecraft from its initial conditions $r = r_0$, $\theta = \theta_0$, $\phi = \phi_0$, $V = V_0$, $\gamma = \gamma_0$ and $\psi = \psi_0$ to the mission-dependent final boundary conditions $r = r_f$, $\theta = \theta_f$, $\phi = \phi_f$, $V = V_f$ and $\gamma = \gamma_f$. Moreover, the time duration is minimized (for a time-optimal reconnaissance mission) in the presence of disturbances and initial entry perturbations, that naturally raises in most control application problems [28].

Assumption 1. Consider Eq.(1) and Eq.(2), the state variables r, θ, ϕ, V, γ and ψ can be measured or obtained [27].

Let us consider the state variable vector $x = [r, \theta, \phi, V, \gamma, \psi]^T$, control variable vector $u = [\alpha, \sigma]^T$ and rewrite Eq.(1) and Eq.(2) in the state space as follows:

$$\dot{x} = f(x(t), u(t)) \tag{3}$$

where $f \in \Re^6$ is the right-hand-side (RHS) of the equations of motion. A figure describing the vehicle reference frames is plotted in Fig.1, whereas aerodynamic forces acting on the spacecraft are shown in Fig.2.

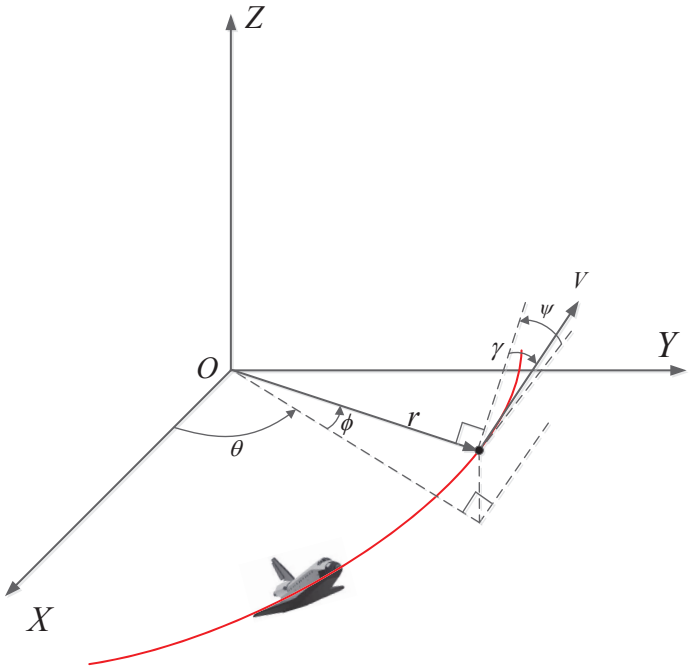


Fig. 1: Reference frame

Several path constraints are implemented to ensure a safe trajectory for the vehicle. During the mission, the state and control variables should vary in its tolerant regions and it can be written as follows:

$$x_{min} \le x(t) \le x_{max} \tag{4}$$

$$u_{min} \le u(t) \le u_{max} \tag{5}$$

where $x(t) \in \mathbf{X}$ and $u(t) \in \mathbf{U}$. x_{min} and x_{max} are the lower and upper bounds of the state, whereas u_{min} and u_{max} are the lower and upper bounds of the input.

In order to protect the vehicle's structure, the two path constraints taken into account in the guidance loop are the heating rate and normal acceleration, which can be formulated as:

$$\begin{aligned} \dot{Q}_d &= K_Q \rho^{0.5} V^{3.07} q_a < \dot{Q}_{dmax} \\ n_L &= \frac{\sqrt{L^2 + D^2}}{mg} < n_{Lmax} \end{aligned} \tag{6}$$

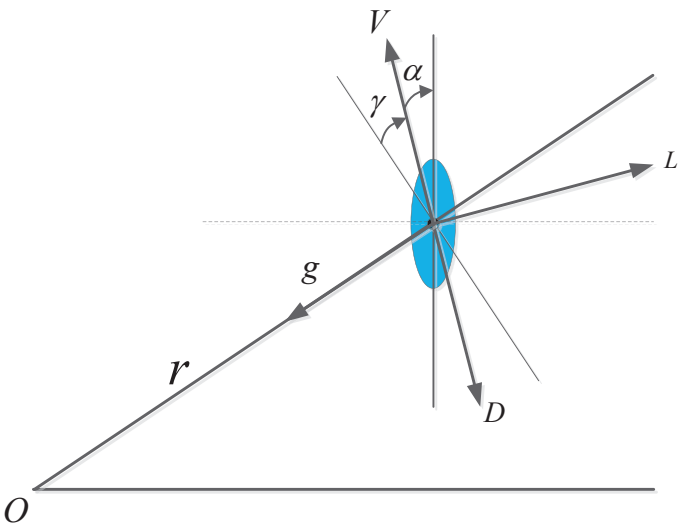


Fig. 2: Aerodynamic forces acting on the spacecraft

where $q_a = (c_0 + c_1\alpha + c_2\alpha^2 + c_3\alpha^3)$. Q_{dmax} represents the acceptable maximum heating rate.

A. Overall guidance framework

The overall architecture of the tracking guidance algorithm is shown in Fig.3.

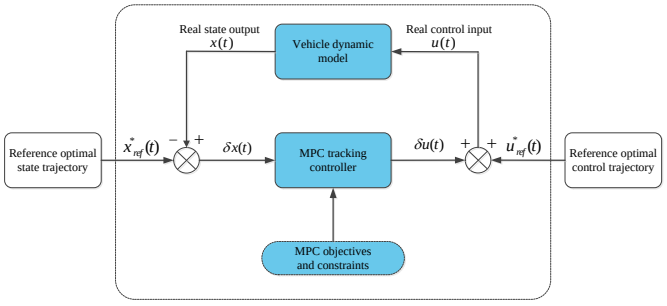


Fig. 3: Overall structure of the guidance system

As can be seen from Fig.3, the optimal state and control reference sequences (e.g. x^*_{ref} and u^*_{ref}) are calculated first. The close-loop guidance law is then achieved based on the inner MPC controller. It is worth noting that by applying the MPC, feedback can be achieved through real-time computation of the open-loop optimal control problem. The real control input $u(t)$ is calculated by combining the reference control u^*_{ref} and the feedback control variable $\delta u(t)$. The real state output $x(t)$ is then obtained by entering the real control input into the vehicle dynamics. This process will be further discussed in the next section of this paper.

III. Two Receding Horizon Control Schemes

A. Nonlinear Model Predictive Control

MPC [29] can be described as an iterative optimization process that generates control actions by applying a moving horizon trajectory optimization. The control is periodically recomputed with the current state as an initial condition, thereby providing a feedback action that can improve robustness to uncertainties and disturbances. In this section, a nonlinear formulation of the tracking MPC optimization problem is constructed. As illustrated in Fig.3, the open-loop solution is assumed to be known and used as reference trajectories (denoted as x^*_{ref} and u^*_{ref}). Then the trajectory tracking problem can be reduced to find a control law such that

$$x(t) - x^*_{ref}(t) \approx 0 \tag{7}$$

Since the MPC prediction model is used and the feedback control law is obtained in discrete-time, a discrete-time formulation of the equations of motion is needed [25]. Considering a sampling time step Δt and a sampling instant k, by applying the Euler's approximation to the vehicle kinematics and dynamics, the following discrete-time model can be constructed:

$$\begin{cases} r(k+1) = & r(k) + \Delta t(V(k)\sin\gamma(k)) \\ \theta(k+1) = & \theta(k) + \Delta t(\frac{V(k)\cos\gamma(k)\sin\psi(k)}{r(k)\cos\phi(k)}) \\ \phi(k+1) = & \phi(k) + \Delta t(\frac{V(k)\cos\gamma(k)\cos\psi(k)}{r(k)}) \\ V(k+1) = & V(k) + \Delta t(\frac{T(k)\cos\alpha(k) - D(k)}{m} - g\sin\gamma(k)) \\ \gamma(k+1) = & \gamma(k) + \Delta t(\frac{L(k)\cos\sigma(k) + T(k)\sin\alpha(k)}{mV(k)} \\ & + (\frac{V(k)^2 - gr(k)}{r(k)V(k)})\cos\gamma(k)) \\ \psi(k+1) = & \psi(k) + \Delta t(\frac{L(k)\sin\sigma(k)}{mV(k)\cos\gamma(k)} \\ & + \frac{V(k)}{r(k)}\cos\gamma(k)\sin\psi(k)\tan\phi(k)) \end{cases} \tag{8}$$

Eq.(8) can then be rewritten in a more compact form

$$x(k+1) = f(x(k), u(k)) \in \Re^6 \tag{9}$$

where $x(k) \in \mathbf{X}$, $u(k) \in \mathbf{U}$, and $k = 1, 2, ..., N$, with the prediction horizon N. N is satisfying $1 \leq N$. It is worth noting that since the reference trajectory can satisfy the equations of motion, it can also be written in a discrete-time formulation:

$$x^*_{ref}(k+1) = f(x^*_{ref}(k), u^*_{ref}(k)) \in \Re^6 \tag{10}$$

Based on Eq.(8), the prediction of the dynamic equations at kth time instant is calculated as follows:

$$x(k+j+1|k) = f(x(k+j|k), u(k+j|k)) \tag{11}$$

where $j \in [0, N-1]$. By introducing the error vectors $\delta x = x - x^*_{ref}$ and $\delta u = u - u^*_{ref}$, the control objective for Eq.(9) in MPC can be set to drive the state error vector to the origin. More precisely, the objective function of the trajectory tracking MPC can be defined as follows:

$$\begin{aligned} J_{NMPC}(\delta x, \delta u) = & \sum_{j=1}^{N} \delta x^T(k+j|k) Q \delta x(k+j|k) \\ & + \sum_{j=0}^{N-1} \delta u^T(k+j|k) R \delta u(k+j|k) \end{aligned} \tag{12}$$

where $Q \in \Re^{6\times 6}$ is a semi-definite matrix. $R \in \Re^{2\times 2}$ is a symmetric positive definite matrix. The discrete time horizon under which the stage costs is minimized is $k = 1, ..., N$. In Eq.(12), the first term on the RHS is to minimize the deviation between the nominal state and the reference state, whereas the second term is to minimize the control efforts.

Based on the discretized dynamic equations, path constraints and objective function, the NMPC optimization model can then be constructed. The aim of the NMPC trajectory

tracking algorithm is to minimize the objective function subject to the dynamic constraints and path constraints repeatedly over the prediction horizon $k = 1, 2, ..., N$. The optimization formulation can be summarized as:

$$\begin{aligned}
\text{minimize} \quad & J_{NMPC}(\delta x, \delta u) = \sum_{j=1}^{N} \delta x^T(k+j|k) Q \delta x(k+j|k) \\
& \quad + \sum_{j=0}^{N-1} \delta u^T(k+j|k) R \delta u(k+j|k) \\
\text{subject to} \quad & \forall j \in [1, 2, ..., N] \\
& x(k+j+1|k) = f(x(k+j|k), u(k+j|k)) \\
& x(k|k) = x_k \\
& x_{min} \leq x(k+j+1|k) \leq x_{max} \\
& u_{min} \leq u(k+j+1|k) \leq u_{max} \\
& \delta u_{min} \leq \delta u(k+j+1|k) \leq \delta u_{max}
\end{aligned} \tag{13}$$

where x_k is the initial condition corresponding to the values of the states measured at the current sampling time point. δu_{min} and δu_{max} stand for the lower and upper bounds of the input vectors.

In most real-world applications, the state variables will have a constraint at the final time (e.g. terminal conditions). In this case, the terminal penalty term might be added in the objective so as to ensure that the algorithm will seek to reduce the terminal state error in the process of optimizing the cost function J. The optimization problem given by Eq.(13) should be solved at each time instant k, thereby generating a sequence of optimal states $(\delta x^*(k+1|k), \delta x^*(k+2|k), ..., \delta x^*(k+N|k))$ and controls $(\delta u^*(k|k), \delta u^*(k+1|k), ..., \delta u^*(k+N-1|k))$. Subsequently, the first control action in this sequence is applied to the plant and the remaining portion of this sequence is discarded. Specifically, the overall NMPC algorithm is constructed in Algorithm 1.

Algorithm 1 The main framework of the NMPC

1: **Offline:** Perform trajectory optimization algorithm in order to generate the reference state and control sequences x^*_{ref} and u^*_{ref};
2: Initialize Q, R, and the prediction horizon N;
3: /*Main Loop*/
4: **Online:** At each time instant $k := 0, 1, ...$;
5: (a). Calculate the current state variable $x(k)$ of the plant;
6: (b). Discretize the continuous problem so as to obtain the static NLP model shown in Eq.(13);
7: (c). Solve the optimization problem (13):
8: $\delta u^* = \arg\min J_{NMPC}(\delta x, \delta u)$
9: subject to constraints given by Eq.(13).
10: (d). Calculate $u_k = u^*_{ref} + \delta u^*$ and implement the control law to the plant until the next sampling instant;
11: (e). Set $k = k + 1$;
12: (f). Repeat the procedure a)-e) for the next sampling time point;

B. Linear Model Predictive Control

Although many well-developed NMPC schemes have been proposed and applied in the literature [30], it should be noted that usually, the computational complexity for NMPC schemes is much higher than the linear schemes. Moreover, the NMPC method tends to generate a large scale nonconvex nonlinear programming (NLP) problem. Consequently, the global convergence property for the optimization algorithm can hardly be achieved. This indicates that the NLP solver may fail to converge or spend a large amount of root-finding iterations. Therefore, in order to reduce the computational burden, a LMPC scheme is constructed as an alternative to the nonlinear version. A linear model is obtained by constructing an error model with respect to the reference state and control sequences. Based on Eq.(3), the reference trajectory (x^*_{ref}, u^*_{ref}) should satisfy $\dot{x}^*_{ref} = f(x^*_{ref}, u^*_{ref})$. By expanding the RHS of the dynamics around the reference (x^*_{ref}, u^*_{ref}) and discarding the high order terms, the following equation can be obtained:

$$\dot{x} = f(x^*_{ref}, u^*_{ref}) + f_A(x - x^*_{ref}) + f_B(u - u^*_{ref}) \tag{14}$$

where f_A and f_B are partial derivatives of $f(x, u)$ with respect to x and u at the reference points, respectively. By replacing $x - x^*_{ref}$ and $u - u^*_{ref}$ as δx and δu, Eq.(14) is rewritten as:

$$\delta \dot{x} = f_A \delta x + f_B \delta u \tag{15}$$

subject to the initial conditions $\delta x(t_0) = \delta x_0$. The approximation of $\delta \dot{x}$ by applying discretization gives the following discrete-time system model:

$$\delta x(k+1) = A(k)\delta x(k) + B(k)\delta u(k) \tag{16}$$

The two matrices $A(k)$ and $B(k)$ are obtained analytically. The values of $A(k)$ and $B(k)$ can be calculated based on the state and control histories of the reference trajectories and their analytical form can be found in the Appendix.

Remark 1. It is worth noting that the approximation of system (3) will result in approximation errors. Therefore, the discretized dynamics should be formulated as a linear time-variant stochastic system (e.g. $\delta x(k+1) = A(k)\delta x(k) + B(k)\delta u(k) + w_k$, where w_k can be treated as a random variable). Since the aim of this paper is to construct a new optimization method under the MPC framework, this stochastic term is omitted in the current design. Future work will extend the LMPC-based guidance design to handle stochastic errors by applying, for example, constraint-tightening techniques in stochastic model predictive control [31].

The main advantage of using LMPC is that it can transform the control problem (Eq.(13)) to a standard quadratic optimization problem, and the optimal solution can be found via well-developed gradient-based methods.

Define the following vectors:

$$\bar{x}(k+1) = [\delta x(k+1|k), ..., \delta x(k+N|k)]^T \in \Re^{6N} \tag{17}$$

$$\bar{u}(k) = [\delta u(k|k), \delta u(k+1|k), ..., \delta u(k+N-1|k)]^T \in \Re^{2N} \tag{18}$$

Thus, by introducing $\bar{Q} = diag(Q, ..., Q) \in \Re^{6N \times 6N}$ and $\bar{R} = diag(R, ..., R) \in \Re^{2N \times 2N}$, the cost function (Eq.(12)) can be rewritten as:

$$\bar{J}_{LMPC}(k) = \bar{x}^T(k+1)\bar{Q}\bar{x}(k+1) + \bar{u}^T(k)\bar{R}\bar{u}(k) \tag{19}$$

Based on Eq.(16)-(18), the predicted system can then be

transcribed to a more compact form,

$$\bar{x}(k+1) = \bar{A}(k)\bar{x}(k|k) + \bar{B}(k)\bar{u}(k) \tag{20}$$

where $\bar{A}(k)$ and $\bar{B}(k)$ can be calculated by using Eq.(21) and Eq.(20), respectively.

$$\bar{A}(k) = \begin{pmatrix} A(k|k) \\ A(k+1|k)A(k|k) \\ \vdots \\ \prod_{i=N-2}^{0} A(k+i|k) \\ \prod_{i=N-1}^{0} A(k+i|k) \end{pmatrix} \tag{21}$$

Let us define the following terms:

$$H(k) = 2(\bar{B}^T(k)\bar{Q}\bar{B}(k) + \bar{R}) \tag{21}$$

$$F(k) = 2\bar{B}^T(k)\bar{Q}\bar{A}(k)\delta x(k|k) \tag{22}$$

$$c(k) = \delta x^T(k|k)\bar{A}^T(k)\bar{Q}\bar{A}(k)\delta x(k|k) \tag{23}$$

Therefore, according to the definition of $H(k)$, $F(k)$ and $c(k)$, the optimization objective can be transcribed to a standard quadratic form:

$$\bar{J}_{LMPC}(k) = \frac{1}{2}\bar{u}^T(k)H(k)\bar{u}(k) + F^T(k)\bar{u}(k) + c(k) \tag{24}$$

The matrix $H(k)$ can be simply treated as the Hessian matrix, and it is positive definite. $H(k)$ describes the quadratic part in the objective function (Eq.(12)), whereas the term $F(k)$ describes the linear part. Based on all the definitions stated above, the LMPC optimization model can be given by the following formulation:

$$\begin{aligned} \text{minimize} \quad & \bar{J}_{LMPC}(k) = \tfrac{1}{2}\bar{u}^T(k)H(k)\bar{u}(k) + F^T(k)\bar{u}(k) + c(k) \\ \text{subject to} \quad & \forall j \in [1,2,...,N] \\ & u_{min} \le u(k+j+1|k) \le u_{max} \\ & \delta u_{min} \le \delta u(k+j+1|k) \le \delta u_{max} \end{aligned} \tag{25}$$

It is worth noting that the dynamic constraints (e.g. equations of motion) are no longer necessary. This is because the linearized equations of motion are implicitly embedded in the cost function. To better show the structure of the constructed LMPC algorithm, the overall procedure is illustrated in the Pseudocode (see Algorithm 2).

Algorithm 2 The main framework of the LMPC

Offline: Perform trajectory planning algorithm to generate the reference state and control sequences x^*_{ref} and u^*_{ref};
Initialize Q, R, and the prediction horizon N;
/*Main Loop*/
Online: At each time instant $k := 0, 1, ...$;
    (a). Calculate the current state variable $x(k)$ of the plant;
    (b). Linearize the nonlinear dynamics given by Eq.(1) and Eq.(2);
    (c). Calculate $A(k)$ and $B(k)$ with respect to the reference trajectory;
    (d). Construct the quadratic optimization problem (25) based on Eq.(17)-(24);
    (e). Solve the quadratic optimization problem (25):
        $\delta u^* = \arg\min J_{LMPC}(\delta x, \delta u)$
        subject to constraints shown in Eq.(25).
    (f). Set $u_k = u^*_{ref} + \delta u^*$ and implement the control law to the plant until the next sampling instant;
    (g). Set $k = k + 1$;
    (h). Repeat the procedure a)-g) for the next sampling time point;

IV. Optimization algorithm

The performance of MPC guidance algorithms mainly depends on the ability for solving NLP problems. Therefore, a highly efficient online optimization algorithm is needed to be developed. In this paper, a globally convergent gradient-based algorithm is applied to solve the resulting NLP [26]. Note that the global convergence does not mean global optimality [32]. This method combines the advantages of the interior point method (IP) [33] and the sequential quadratic programming method (SQP). Also, it can reduce the computational effort by using its two nested structure. For completeness, a brief description of this gradient-based method is introduced in this Section.

In terms of the optimization problem given in Eq.(13), one can define a new decision vector as $e = [\delta x, \delta u]^T$. For simplicity reasons, the optimization formulation (e.g. Eq.(13) and Eq.(25)) can then be rewritten in a general form

$$\begin{aligned} \text{minimize} \quad & J(e) \\ \text{subject to} \quad & h(e) = 0 \quad \text{(equality constraints)} \\ & g(e) \le 0 \quad \text{(inequality constraints)} \end{aligned} \tag{26}$$

where $h(e) = (h_1(e), h_2(e), ..., h_l(e))^T$ and $g(e)$= $(g_1(e), g_2(e), ..., g_m(e))^T$, respectively. The basic idea of the improved gradient-based algorithm is to divide the optimization process into two loops. The outer loop is a normal SQP iteration. For example, it constructs a sequence of quadratic programming subproblems by approximating the augmented Lagrangian function $L(e, \lambda, u) = J(e) + \lambda^T h(e) + u^T g(e)$ quadratically, where λ and u are the Lagrange multipliers associated with the equality and inequality constraints, respectively.

Using quadratic model to approximate the augmented Lagrangian function, the QP subproblem is:

$$\begin{aligned} \min \tfrac{1}{2} de^T H(e_k, \lambda_k, u_k) de + \nabla J(e_k)^T de \\ h(e_k) + \nabla h(e_k) de = 0 \\ g(e_k) + \nabla g(e_k) de \le 0 \end{aligned} \tag{27}$$

where de represents the directional derivative whereas $H(e_k, \lambda_k, u_k)$ stands for the Hessian matrix. Commonly, the Hessian is calculated using a suitable approximation defined by the user (e.g. BFGS approximation).

Following that, in the inner loop, the resulting quadratic programming subproblem is solved by applying the IP method. In order to distinguish the inner and outer loops, the internal iteration index is defined as l while the external iteration

$$\bar{B}(k)=\begin{pmatrix} B(k|k) & 0 & \cdots & 0 \\ A(k+1|k)B(k|k) & B(k+1|k) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \prod_{i=N-2}^{1} A(k+i|k)B(k|k) & \prod_{i=N-2}^{2} A(k+i|k)B(k+1|k) & \cdots & 0 \\ \prod_{i=N-1}^{1} A(k+i|k)B(k|k) & \prod_{i=N-1}^{2} A(k+i|k)B(k+1|k) & \cdots & B(k+N-1|k) \end{pmatrix} \tag{20}$$

number is defined as k. The outer iterates of the primal, slack and dual variables are denoted as e_k, s_k, λ_k, u_k, respectively. Correspondingly, $de_{k,l}$, $ds_{k,l}$, $d\lambda_{k,l}$ and $du_{k,l}$ are the inner iterates. After continuing the internal loop until termination or reaching the maximum number of l_{max} given by the user, a SQP solution at the next searching point $(k+1)$ can be achieved.

The main advantage of this two nested gradient-based approach is that the user can control the inner loop by setting the termination conditions or l_{max} at any time. Specifically, since the H_k is fixed at the internal circle, it is not required to solve the QP subproblem exactly, which means the time-consuming QP solution finding can be avoided.

Similar to traditional IP and SQP methods, the Karush-Kuhn-Tucker (KKT) system of the improved gradient-based algorithm is given by Eq.(28).

In Eq.(28), $\Delta d = [\Delta de_{k,l}, \Delta ds_{k,l}, \Delta d\lambda_{k,l}, \Delta du_{k,l}]^T$. $Ds_{k,l}$ and $Du_{k,l}$ are positive diagonal matrices corresponding to the slack variables and multipliers while λ and μ are Lagrangian multipliers and penalty factors related to equality constraints and inequality constraints, respectively.

Solving the KKT system described in Eq.(28), the new iteration can be calculated by:

$$\begin{aligned} dx_{k,l+1} &= de_{k,l} + \alpha_{k,l}\Delta de_{k,l} \\ du_{k,l+1} &= du_{k,l} + \alpha_{k,l}\Delta du_{k,l} \\ ds_{k,l+1} &= ds_{k,l} + \alpha_{k,l}\Delta ds_{k,l} \\ d\lambda_{k,l+1} &= d\lambda_{k,l} + \alpha_{k,l}\Delta d\lambda_{k,l} \end{aligned} \tag{29}$$

where the step length parameter $\alpha_{k,l} \in (0,1]$ should be chosen to ensure that the merit function achieves sufficient decrease but is not too short. In order to measure the progress of each iterate k, the merit function M should be designed. The merit function used in this paper is the same with [26].

The overall structure of this two nested gradient optimization algorithm is illustrated in the Pseudocode (see Algorithm.1).

Since the algorithm can be controlled by the maximum iteration number of the inner loop, the user can have more flexibility with respect to the optimization process. This algorithm is then combined with the two MPC control algorithms investigated in this paper to solve the online tracking optimization problem.

Algorithm 3 Pseudocode for the improved gradient method

1: **procedure** (Two nested structure)
2: Choose starting values $z_0 = (e_0, u_0, \lambda_0, s_0)$
3: **for** $k := 0, 1, 2, ...$ **do**
4: (a). Check stopping criteria for the outer loop
5: (b). Choose $de_{k,0}$, $du_{k,0}$, $d\lambda_{k,0}$ and $ds_{k,0}$
6: **for** $l := 0, 1, 2, ..., l_{max}$ **do**
7: i. Determine $Du_{k,l}$, $Ds_{k,l}$ and $\mu_{k,l}$.
8: ii. Solve the KKT system described in Eq.(28).
9: iii. Apply the line search algorithm shown in
10: Eq.(29).
11: iv. If the inner loop solution can satisfy the
12: stopping condition of QP, break for-loop;
13: **end for**
14: (c). Find step length for the outer loop such that
15: the merit function can have a proper
16: improvement.
17: (d). Update the current searching point and go back
18: to line 3.
19: **end for**
20: Output the optimal solution
21: **end procedure**

V. Simulation Studies

A. Reference trajectory generation

In this section, the numerical simulation for the aeroassisted spacecraft optimal guidance problem is presented. The mission scenario investigated in this study is different from the classic reentry problem [33]. The aeroassisted spacecraft re-enters the atmosphere to a predetermined position for observation and gathering of information of inaccessible areas. Therefore, some of the state variables at the final time should be specified to achieve the mission requirements. According to Fig.3, the offline trajectory and real-time optimal feedback guidance are designed based on the translational equations in Eq.(1) and Eq.(2). The offline solution provides the reference trajectory and it should be pre-designed. Then the inner loop facilitates tracking of the reference trajectory in real time.

The offline spacecraft trajectory optimization problem is formulated as a multi-objective problem [17] and can be stated as follows: given the initial state $x_0 = [6450451.9m, 0deg, 0deg, 7802.9m/s, -1deg, 90deg]^T$ and final state $[r(t_f), \theta(t_f), \phi(t_f), V(t_f), \gamma(t_f)]^T = [6421201.2m, 38.57deg, 10.41deg, 4767.2m/s, 0deg]^T$, find the optimal control sequences $u^* = [\alpha^*, \sigma^*]$, which optimizes the cost function without violating the path constraints. Two objective functions are selected in the offline trajectory design. The first objective is to minimize the final time so as to complete the observation mission in the shortest possible time interval (e.g. $J_1 = t_f$). In addition, minimizing the total

$$\begin{pmatrix} H_k & 0 & \nabla h(e_k)^T & \nabla g(e_k)^T \\ 0 & Du_{k,l} & 0 & Ds_{k,l} \\ \nabla h(e_k) & 0 & 0 & 0 \\ \nabla g(e_k) & I & 0 & 0 \end{pmatrix} \Delta d = - \begin{pmatrix} H_k de_{k,l} + \nabla J(e_k) + \nabla h(e_k)^T d\lambda_{k,l} + \nabla g(e_k)^T du_{k,l} \\ Ds_{k,l} du_{k,l} - \mu_{k,l} \\ h(e_k) + \nabla h(e_k) de_{k,l} \\ g(e_k) + \nabla g(e_k) de_{k,l} + s_k + e^T ds_{k,l} \end{pmatrix} \tag{28}$$

aerodynamic heating is also chosen as the second objective since the vehicle structure integrity is largely affected by the aerodynamic heating (e.g. $J_2 = \int_{t_0}^{t_f} \dot{Q}(t)dt$). The algorithms used in this stage are the Radau pseudospectral method (RPM) and fuzzy physical programming method (FPP). A detailed description in terms of these two algorithms can be found in [17]. It is worth noting that in [17], the authors generated an optimal multi-objective trajectory for the aeroassisted spacecraft. In this research, apart from all the mission requirements stated in [17], an additional observation requirement is taken into account. That means one of the aims of the guidance algorithm is to guide the vehicle to the target region.

All the simulation results were carried out using Matlab under Windows 7 and Intel (R) i7-3520M CPU, 2.90GHZ, with 4.00 GB RAM. It should be noted that both the offline and online optimization processes are carried out by using the proposed two nested gradient method. After generating the nominal time history with respect to the state and control variables, the solutions are used as the references and provided to the constructed MPC tracking algorithms.

B. Optimal tracking solutions

The effectiveness of the constructed guidance method is analyzed in this section. The simulation results were carried out under the following initial condition uncertainty:

$$\begin{array}{ll} |\delta r(t_0)| \leq 1000(m) & |\delta\theta(t_0)| \leq 0.1(deg) \\ |\delta\phi(t_0)| \leq 0.1(deg) & |\delta V(t_0)| \leq 50(m/s) \\ |\delta\gamma(t_0)| \leq 0.05(deg) & |\delta\psi(t_0)| \leq 0.05(deg) \end{array} \tag{30}$$

The predictive horizon is set as: $N = 20s$. In the LMPC case, the values of Q and R used to generate the optimal guidance law are obtained according to the Bryson's rule [22]. A 1000-run Monto-Carlo study was performed to evaluate the effectiveness and robustness of the two MPC schemes in the presence of the dispersions in entry states and model errors. The aerodynamic coefficients and atmospheric density were perturbed normally up to 10%. The vehicle mass was perturbed uniformly up to 5% with the nominal mass of 6309.43slug, this gives a range of value of 5993.96slug to 6624.90slug. The drag and lift coefficients were modeled as random Gaussian distributions.

Figs.4-6 show the results of the trajectory tracking via the NMPC guidance algorithm. Fig.4 and Fig.5 indicate the trajectory tracking results between the nominal state trajectories and the reference state trajectories. It can be observed from Fig.4 and Fig.5 that the deviation between the obtained results and the reference is relatively small. Furthermore, it can be seen from Fig.5 that all the flight trajectories can satisfy the heating and normal acceleration path constraints.

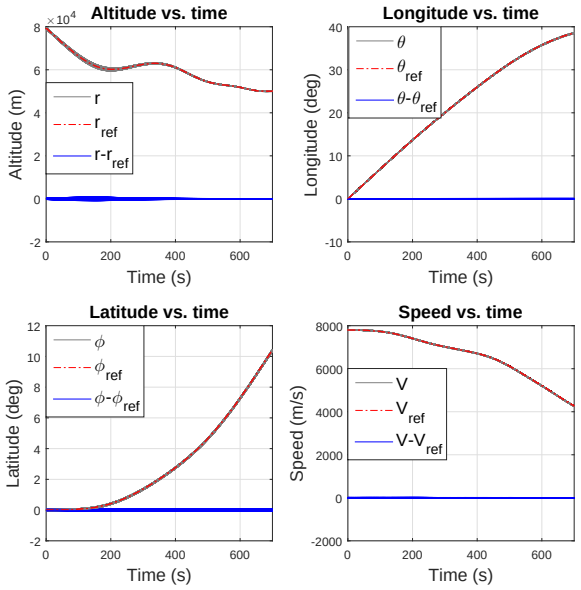


Fig. 4: NMPC state tracking results

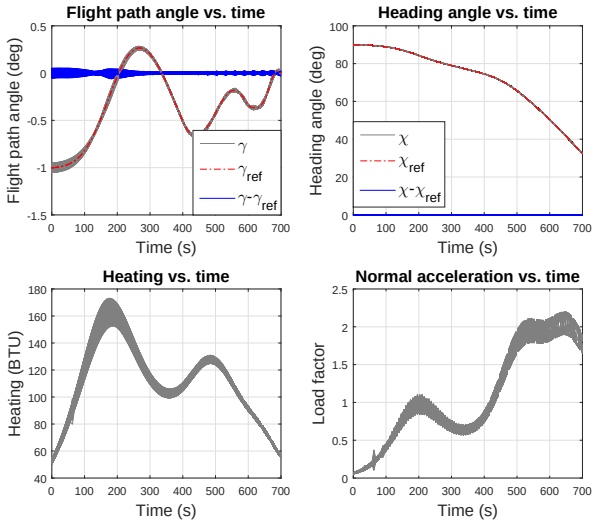


Fig. 5: NMPC state tracking and path constraints results

Fig.6 shows the results of the final longitude and latitude error for 1000 Monto-Carlo simulations. It is worth noting that for the observation mission considered in this paper, it is desirable to use the online guidance law to guide the spacecraft to the acceptable region. From the results shown in Fig.6, it can be calculated that around 72.7% of the runs can guide the vehicle to the desirable region (e.g. the inner

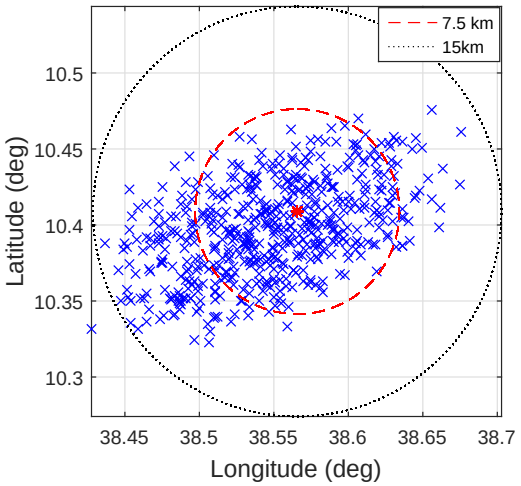


Fig. 6: NMPC final entry longitude and latitude

circular area), while almost all the runs (e.g. 99.4%) can guide the vehicle to be within the acceptable region (e.g. the outer circular area). All the results indicate that the NMPC is able to track the desired reference trajectory in the presence of model uncertainty without violating path constraints.

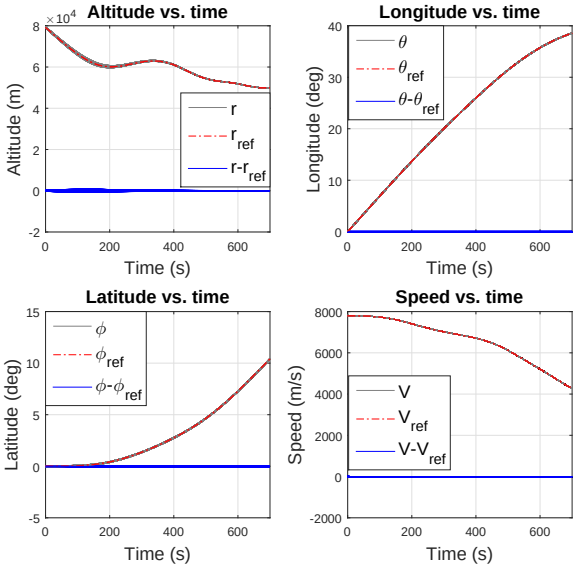


Fig. 7: LMPC state tracking errors

As for the LMPC case, Figs.7-11 show the results of the trajectory tracking under the LMPC control with entry dispersions and model uncertainties. The deviations between the nominal trajectory and the reference trajectory are illustrated in Fig.7 and Fig.8, where the error is again small and the difference can be omitted. Fig.9 illustrates the dispersion values with respect to the final longitude and latitude. In all of the Monto-Carlo runs, the vehicle can reach its desirable region without violating the path constraints (see Fig.8). These results confirm that the constructed LMPC scheme is feasible and effective in solving the online aeroassisted spacecraft

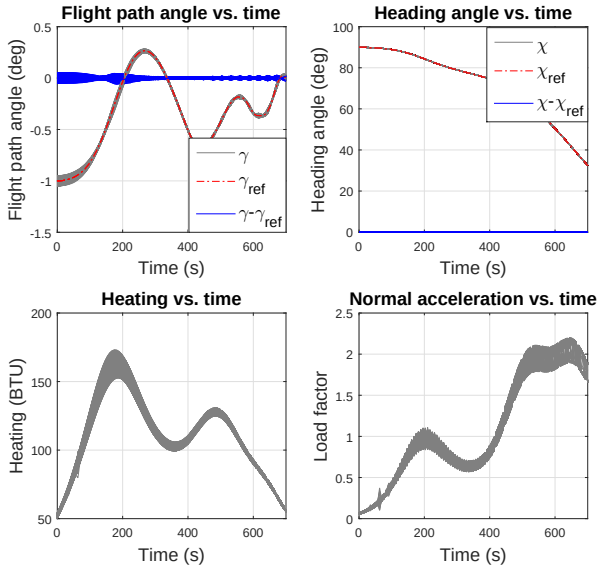


Fig. 8: LMPC state tracking and path constraints results

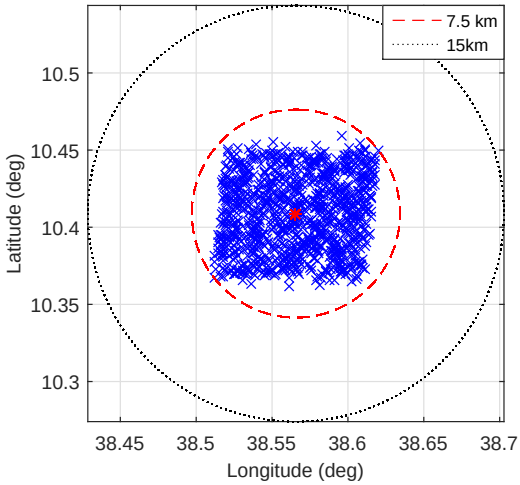


Fig. 9: LPMC final entry longitude and latitude

guidance problems in the presence of entry state perturbations and model uncertainties.

In order to better show the state errors of the monto-carlo simulation, Fig.10 and Fig.11 present the altitude, velocity and flight path angle error histories obtained using the NMPC and LMPC. It can be seen from the numerical results that compared with the NMPC, LMPC can have a better performance in terms on achieving smaller final error values. Besides, by applying the LMPC control method, the tracking errors with respect to the state variables are ultimately bounded. It should be noted that according to Fig.11, the flight path angle error history has some oscillations. This can be explained that the tracking performance might be affected at the time period where the nonlinearity of the reference trajectory is high.

Remark 2. For the LMPC control scheme, by selecting N, Q and R, it always exists a finite horizon length such that the

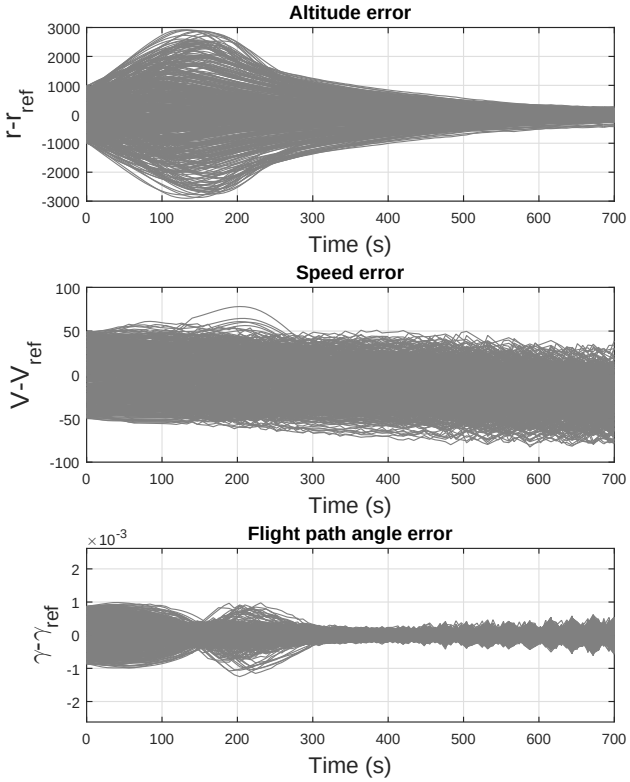


Fig. 10: NMPC tracking error history

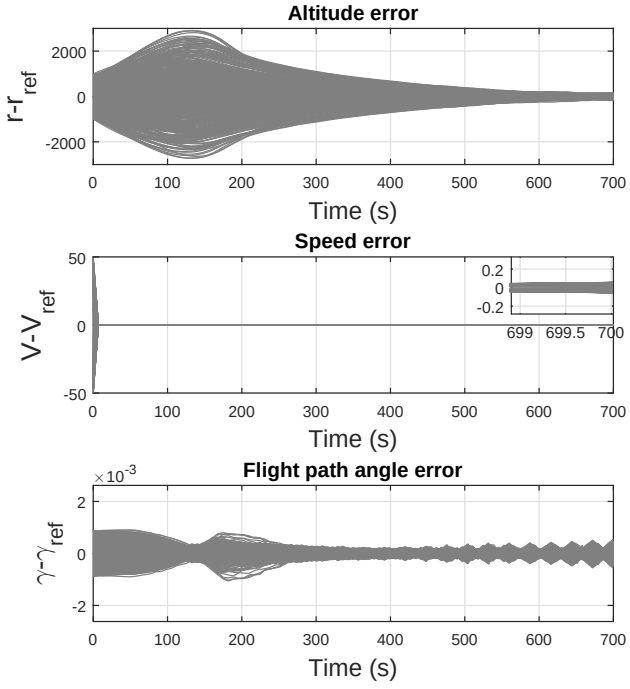


Fig. 11: LMPC tracking error history

trajectory tracking error can be ultimately bounded, which can be found in [34].

From the tracking results shown in Fig.4 to Fig.11, it can be concluded that both the LMPC and NMPC schemes constructed in Section III of this paper are able to generate the optimal guidance command for the spacecraft observation mission. When the perturbations and uncertainties are modeled into the problem, the guidance law calculated via the two MPC schemes can lead the tracking errors to a small value. Moreover, since the path constraints are embedded in the MPC optimization formulation, the obtained flight trajectories will not violate the path constraints during the entire flight mission.

C. Comparative analysis

To further compare the solutions obtained via LMPC and NMPC schemes, attention is given to the optimization process of these two MPC schemes. By applying the newly proposed two nested gradient method and standard IP approach, the convergence results (e.g. the number of maximum, minimum, average solution finding iterations and the total CPU time), for each control loop, are tabulated in Table I. In order to preserve the online performance, for each control loop, the maximum number of solution finding iteration for the two optimization algorithms is set as 50. As can be seen from Table I, the average computational time to solve one optimal control problem over the finite prediction horizon is less than $0.5s$ for the LMPC and NMPC cases. This could potentially allow the real-time application of these two schemes. The two nested gradient method investigated in this paper has generally better performance in terms of convergence ability when compared with the IP method for both LMPC and NMPC cases. Moreover, regarding the CPU time, the newly proposed NLP solver can also have positive influences in terms of reducing the processing time, which is important especially in the online guidance law design. Therefore, it is advantageous to use the investigated gradient NLP solver in solving the online MPC-based spacecraft guidance problems.

From Table I, it can be observed that there is a significant difference between the NMPC case and LMPC case in terms of the performance of optimization process. For each MPC loop, the NMPC case takes more solution-finding iterations than the linear case. This is because, from the linear MPC optimization formulation (Eq.(25)), the original problem can be convexified to some extent, which means the problem becomes much smoother. Therefore, it becomes easier for the optimization methods to achieve the global convergence [32]. However, in the nonlinear case, the resulting NLP problem to be solved online is usually nonconvex and has a large number of optimization parameters. This will affect the number of solution-finding iterations significantly. Consequently, the optimization problem at each time instant may not be solvable efficiently or reliably.

Moreover, comparative studies were also performed to compare the guidance performance achieved by applying the proposed two MPC solvers and another promising strategy. For example, a neural-network optimization based NMPC design reported in [24]. This guidance strategy applies a specific

TABLE I: Average optimization performance of the LMPC and NMPC

NMPC	Average iteration	Max iteration	Min iteration	Average CPU time (s)	Max CPU time (s)	Min CPU time
IP method	22	50	11	0.441	0.907	0.220
Two nested method	18	50	10	0.324	0.843	0.233
LMPC	Average iteration	Max iteration	Min iteration	Average CPU time (s)	Max CPU time (s)	Min CPU time
IP method	16	41	14	0.342	0.743	0.285
Two nested method	11	28	8	0.228	0.487	0.178

TABLE II: Statistics of terminal state dispersions

State	Mean			Standard deviation		
	LMPC	NMPC	Controller [24]	LMPC	NMPC	Controller [24]
Altitude, m	-0.0108	-35.7233	-15.7599	32.3619	128.8344	100.6163
Longitude, deg	5.03×10^{-4}	-0.0174	-0.0197	0.0223	0.0499	0.0591
Latitude, deg	3.96×10^{-5}	-0.0118	-0.0112	0.0176	0.0307	0.0303
Velocity, m/s	-3.52×10^{-4}	-25.2434	-28.7414	0.0317	22.7690	28.9327
Path angle, deg	9.93×10^{-5}	5.89×10^{-4}	1.05×10^{-4}	0.0122	0.0167	0.0169
Heading angle, deg	-4.83×10^{-4}	3.62×10^{-3}	-1.51×10^{-3}	0.0118	0.0244	0.0258

primal-dual neural-network optimization technique to calculate the optimal control command. Table II presents the statistical analysis of different guidance schemes in terms of the terminal state dispersions. From Table II, it can be seen that the guidance accuracy of the NMPC strategy developed in this paper and the controller reported in [24] is comparable. By applying the improved gradient-based optimization method, the LMPC scheme investigated in this study can perform better in terms of reducing the terminal state dispersions.

VI. Conclusion

In this paper, the linear and nonlinear MPC models were constructed and applied to solve the online aeroassisted vehicle optimal guidance problems. The two MPC schemes solve the tracking optimal control problem in a receding manner and allow flight path limitations that ensure safety of the flight vehicle to be taken into account. The application of MPC for the online guidance problems requires a high-efficiency optimization solver that can satisfy the real-time requirement. The optimization method applied in this paper is adjustable in terms of computational complexity such that it can reduce the computational time and match the real-time requirements. Simulation studies indicate that both the linear and nonlinear MPC schemes are able to track the pre-designed reference trajectories for the aeroassisted space vehicle. In particular, the Monto-Carlo simulations further confirm the effectiveness of the MPC algorithm since it can guide the vehicle to the acceptable region without violating path constraints. Moreover, according to the convergence analysis, the two nested gradient optimization method is well suited to the two MPC formulations. Therefore, it is effective and efficient to use the developed guidance algorithm to solve the online spacecraft trajectory tracking problems.

Appendix

The non-zero components of the discrete-time system matrix $A(k) = [A_{ij}(k)] \in \Re^{6\times 6}$ are defined in Eq.(A.1) and Eq.(A.2), where $k = \rho S_{ref}/2m$, C_L and C_D are lift and drag coefficients, respectively.

Similarly, $B(k) = [B_{ij}(k)] \in \Re^{6\times 3}$ is given by:

$$
\begin{aligned}
B_{41}(k) &= -kC_{D_\alpha}V_{ref}^2(k)\Delta t \\
B_{51}(k) &= kC_{L_\alpha}V_{ref}(k)\cos\sigma_{ref}(k)\Delta t \\
B_{52}(k) &= -kC_L V_{ref}(k)\sin\sigma_{ref}(k)\Delta t \\
B_{61}(k) &= kC_{L_\alpha}V_{ref}(k)\sin\sigma_{ref}(k)\Delta t \\
B_{62}(k) &= kC_L V_{ref}(k)\cos\sigma_{ref}(k)\Delta t
\end{aligned}
$$

Acknowledgment

The authors would like to thank all the anonymous reviewers for their constructive comments and helpful suggestions with regards to this paper.

References

[1] G. D. Walberg, "A survey of aeroassisted orbit transfer," *Journal of Spacecraft and Rockets*, vol. 22, no. 1, pp. 3–18, 1985.

[2] K. D. Mease, "Optimization of aeroassisted orbital transfer - current status," *Journal of the Astronautical Sciences*, vol. 36, no. 1-2, pp. 7–33, 1988.

[3] H. S. London, "Change of satellite orbit plane by aerodynamic maneuvering," *Journal of the Aerospace Sciences*, vol. 29, no. 3, pp. 323–332, 1962.

[4] H. Baumann, "Thrust limited coplanar aeroassisted orbital transfer," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 4, pp. 732–738, 2001.

[5] H. Seywald, "Variational solutions for the heat-rate-limited aeroassisted orbital transfer problem," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 3, pp. 686–692, 1996.

[6] H. Baumann and H. J. Oberle, "Numerical computation of optimal trajectories for coplanar, aeroassisted orbital transfer," *Journal of Optimization Theory and Applications*, vol. 107, no. 3, pp. 457–479, 2000.

[7] N. Berend, S. Bertrand, and C. Jolly, "Optimization method for mission analysis of aeroassisted orbital transfer vehicles," *Aerospace Science and Technology*, vol. 11, no. 5, pp. 432–441, 2007.

[8] A. J. Trask and V. L. Coverstone, "Optimal low-thrust trajectories combined with an aeroassist maneuver," *Journal of Spacecraft and Rockets*, vol. 41, no. 4, pp. 629–634, 2004.

[9] N. X. Vinh and Y.-W. Shih, "Optimum multiple-skip trajectories," *Acta Astronautica*, vol. 41, no. 2, pp. 103–112, 1997.

[10] F. Zimmermann and A. J. Calise, "Numerical optimization study of aeroassisted orbital transfer," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 127–133, 1998.

[11] A. V. Rao, S. Tang, and W. P. Hallman, "Numerical optimization study of multiple-pass aeroassisted orbital transfer," *Optimal Control Applications and Methods*, vol. 23, no. 4, pp. 215–238, 2002.

[12] C. L. Darby and A. V. Rao, "Minimum-fuel low-earth orbit aeroassisted orbital transfer of small spacecraft," *Journal of Spacecraft and Rockets*, vol. 48, no. 4, pp. 618–628, 2011.

$$
\begin{aligned}
A_{14}(k) &= \sin\gamma_{ref}(k)\Delta t & A_{15}(k) &= V_{ref}(k)\cos\gamma_{ref}(k)\Delta t \\
A_{21}(k) &= -\frac{V_{ref}(k)\cos\gamma_{ref}(k)\sin\chi_{ref}(k)}{r_{ref}(k)^2\cos\phi_{ref}(k)}\Delta t & A_{23}(k) &= \frac{V_{ref}(k)\cos\gamma_{ref}(k)\sin\chi_{ref}(k)\tan\phi_{ref}(k)}{r_{ref}(k)\cos\phi_{ref}(k)}\Delta t \\
A_{24}(k) &= \frac{\cos\gamma_{ref}(k)\sin\chi_{ref}(k)}{r\cos\phi_{ref}(k)}\Delta t & A_{25}(k) &= -\frac{V\cos\gamma_{ref}(k)\sin\chi_{ref}(k)}{r_{ref}(k)\cos\phi_{ref}(k)}\Delta t \\
A_{26}(k) &= \frac{V_{ref}(k)\cos\gamma_{ref}(k)\cos\chi_{ref}(k)}{r_{ref}(k)\cos\phi_{ref}(k)}\Delta t & A_{31}(k) &= -\frac{V_{ref}(k)\cos\gamma_{ref}(k)\cos\chi_{ref}(k)}{r_{ref}(k)^2}\Delta t \\
A_{34}(k) &= \frac{\cos\gamma_{ref}(k)\cos\chi_{ref}(k)}{r_{ref}(k)}\Delta t & A_{35}(k) &= -\frac{V_{ref}(k)\cos\gamma_{ref}(k)\cos\chi_{ref}(k)}{r_{ref}(k)}\Delta t \\
A_{36}(k) &= -\frac{V_{ref}(k)\cos\gamma_{ref}(k)\sin\chi_{ref}(k)}{r_{ref}(k)}\Delta t & A_{41}(k) &= (\frac{kC_DV_{ref}(k)^2}{H} + \frac{2g\sin\gamma_{ref}(k)}{r_{ref}(k)})\Delta t \\
A_{44}(k) &= 1 - 2kC_DV_{ref}(k)\Delta t & A_{45}(k) &= -g\cos\gamma_{ref}(k)\Delta t
\end{aligned}
\tag{A.1}
$$

$$
\begin{aligned}
A_{51}(k) &= -V_{ref}(k)(\frac{kC_L\cos\sigma_{ref}(k)}{H} + \frac{\cos\gamma_{ref}(k)}{r_{ref}(k)^2})\Delta t + \frac{2g\cos\gamma_{ref}(k)}{V_{ref}(k)r_{ref}(k)}\Delta t \\
A_{54}(k) &= kC_L\cos\sigma_{ref}(k)\Delta t + \frac{gr_{ref}(k)+V_{ref}(k)^2}{V_{ref}(k)^2r_{ref}(k)}\cos\gamma_{ref}(k)\Delta t \\
A_{55}(k) &= 1 + \sin\gamma_{ref}(k)\frac{gr_{ref}(k)-V_{ref}(k)^2}{V_{ref}(k)r_{ref}(k)}\Delta t \\
A_{61}(k) &= -V_{ref}(k)(\frac{kC_L\sin\sigma_{ref}(k)}{H\cos\gamma_{ref}(k)} + \frac{\cos\gamma_{ref}(k)\sin\chi_{ref}(k)\tan\theta_{ref}(k)}{r_{ref}(k)^2})\Delta t \\
A_{63}(k) &= \frac{V_{ref}(k)\cos\gamma_{ref}(k)\sin\chi_{ref}(k)}{r_{ref}(k)}(1+\tan^2\theta_{ref}(k))\Delta t \\
A_{64}(k) &= (\frac{kC_L\sin\sigma_{ref}(k)}{\cos\gamma_{ref}(k)} + \frac{\cos\gamma_{ref}(k)\sin\chi_{ref}(k)\tan\theta_{ref}(k)}{r_{ref}(k)})\Delta t \\
A_{65}(k) &= (\frac{V_{ref}(k)\tan\gamma_{ref}(k)kC_L\sin\sigma_{ref}(k)}{\cos\gamma_{ref}(k)} - \frac{V_{ref}(k)\sin\gamma_{ref}(k)\sin\chi_{ref}(k)\tan\theta_{ref}(k)}{r_{ref}(k)})\Delta t \\
A_{66}(k) &= 1 + \frac{V_{ref}(k)\cos\gamma_{ref}(k)\tan\theta_{ref}(k)\cos\chi_{ref}(k)}{r_{ref}(k)}\Delta t
\end{aligned}
\tag{A.2}
$$

[13] D. G. Hull, J. M. Giltner, J. L. Speyer, and J. Mapar, "Minimum energy-loss guidance for aeroassisted orbital plane change," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 4, pp. 487–493, 1985.

[14] D. S. Naidu, J. L. Hibey, and C. D. Charalambous, "Neighboring optimal guidance for aeroassisted orbital transfer," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 3, pp. 656–665, 1993.

[15] C. W. Brunner and P. Lu, "Skip entry trajectory planning and guidance," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 5, pp. 1210–1219, 2008.

[16] Y. Xia, F. Pu, S. Li, and Y. Gao, "Lateral path tracking control of autonomous land vehicle based on adrc and differential flatness," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 5, pp. 3091–3099, 2016.

[17] R. Chai, A. Savvaris, and A. Tsourdos, "Fuzzy physical programming for space manoeuvre vehicles trajectory optimization based on hp-adaptive pseudospectral method," *Acta Astronautica*, vol. 123, pp. 62–70, 2016.

[18] ——, "Violation learning differential evolution-based hp-adaptive pseudospectral method for trajectory optimization of space maneuver vehicle," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 4, pp. 2031–2044, 2017.

[19] J. Dai and Y. Xia, "Mars atmospheric entry guidance for reference trajectory tracking," *Aerospace Science and Technology*, vol. 45, pp. 335–345, 2015.

[20] T. Ohtsuka, "A continuation/gmres method for fast computation of nonlinear receding horizon control," *Automatica*, vol. 40, no. 4, pp. 563–574, 2004.

[21] H. Peng and X. Jiang, "Nonlinear receding horizon guidance for spacecraft formation reconfiguration on libration point orbits using a symplectic numerical method," *ISA Transactions*, vol. 60, pp. 38–52, 2016.

[22] B. Tian and Q. Zong, "Optimal guidance for reentry vehicles based on indirect legendre pseudospectral method," *Acta Astronautica*, vol. 68, no. 7-8, pp. 1176–1184, 2011.

[23] H. Peng, Q. Gao, Z. Wu, and W. Zhong, "Optimal guidance based on receding horizon control for low-thrust transfer to libration point orbits," *Advances in Space Research*, vol. 51, no. 11, pp. 2093–2111, 2013.

[24] Z. Li, Y. Xia, C. Y. Su, J. Deng, J. Fu, and W. He, "Missile guidance law based on robust model predictive control using neural-network optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 8, pp. 1803–1809, 2015.

[25] G. P. Liu, "Design and analysis of networked non-linear predictive control systems," *IET Control Theory & Applications*, vol. 9, no. 11, pp. 1740–1745, 2015.

[26] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "Improved gradient-based algorithm for solving aeroassisted vehicle trajectory optimization problems," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 8, pp. 2093–2101, 2017.

[27] Y. B. Shtessel, I. A. Shkolnikov, and A. Levant, "Guidance and control of missile interceptor using second-order sliding modes," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 1, pp. 110–124, 2009.

[28] A. Gopalan, A. Ratnoo, and D. Ghose, "Time-optimal guidance for lateral interception of moving targets," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 3, pp. 510–525, 2015.

[29] R. Genest and J. V. Ringwood, "Receding horizon pseudospectral control for energy maximization with application to wave energy devices," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 29–38, 2017.

[30] H. Li and Y. Shi, "Robust distributed model predictive control of constrained continuous-time nonlinear systems: A robustness constraint approach," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1673–1678, 2014.

[31] M. Lorenzen, F. Dabbene, R. Tempo, and F. Allgower, "Constraint-tightening and stability in stochastic model predictive control," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3165–3177, 2017.

[32] F. Fahroo and I. M. Ross, "Pseudospectral methods for infinite-horizon nonlinear optimal control problems," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 4, pp. 927–936, 2008.

[33] J. Laurent-Varin, F. Bonnans, N. Berend, M. Haddou, and C. Talbot, "Interior-point approach to trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1228–1238, 2007.

[34] B. Zhu and X. Xia, "Adaptive model predictive control for unconstrained discrete-time linear systems with parametric uncertainties," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 3171–3176, 2016.



Runqi Chai (SM'15) was born in Beijing, China, in 1993. He is currently pursuing the Ph.D. degree in Aerospace Engineering at Cranfield University. His research interests include trajectory optimization, guidance and control.



Al Savvaris (M'08) received the M.Eng. degree in aerospace systems engineering from the University of Hertfordshire, Hertfordshire, U.K., in 1998 and the Ph.D. degree in radiowave propagation and system design from the University of South Wales, Pontypridd, U.K., in 2004.

He is a Reader with the Centre for Cyber-Physical Systems, Cranfield University, Cranfield, U.K. He established the Autonomous Vehicle Dynamics and Control M.Sc. course and the COMAC training programme at Cranfield. His current research interests include systems integration, hybrid energy management, communication systems, embedded systems, guidance, and control. He is currently researching on Innovate U.K. Funded AirStart and USMOOTH Projects. In the past, he researched on the FLAVIIR and ASTRAEA UAS Projects, developing new technologies for unmanned systems, researching on hardware and system integration. He has published over 100 peer-reviewed journal and conference papers.

Dr. Savvaris participated in FP6, in the scope of the FLYSAFE Project, researching on next generation integrated safety systems. He was a member of the Autonomous Systems National Technical Committee, EPSRC College Review Member, and a Reviewer on several international publications, including IMechE and IEEE.



Antonios Tsourdos (M'99) obtained an M.Eng. on electronic, control and systems engineering from the University of Sheffield, in 1995, an M.Sc. on systems engineering from Cardiff University, in 1996, and a Ph.D. on nonlinear robust missile autopilot design and analysis from Cranfield University, in 1999. He is currently a professor of control systems at Cranfield University. Appointed Head of the Autonomous Systems Group in 2007. Professor Tsourdos was member of the Team Stellar, the winning team for the UK MoD Grand Challenge (2008) and the IET Innovation Award (Category Team, 2009). Antonios is an editorial board member on several international publications including IMechE and IEEE. He is member if the IFAC Technical Committee on Intelligent Autonomous Vehicles, the IET Executive Team on Robotics and Automation and the ATI Autonomous Systems National Technical Committee. He was also involved in the SEAS DTC on Autonomous Systems Verifications. Professor Tsourdos has also been engaged in research on guidance & control for single and multiple vehicles as well as verifiable autonomy of autonomous systems and lately dealing with the newly-important subjects of integrated system health management and cyber-physical systems. He has published over 100 peer-reviewed journal and conference papers.



Senchun Chai was born in Beijing, China. He received his Ph.D. degree and then held a Postdoc fellowship with School of Electronics, University of Glamorgan, UK, in 2007 and 2009, respectively. He joined the School of Automation in September 2010, where he is currently an associate Professor in control. He was a researcher at Cranfield University, UK, from 2009 to 2010, and was a visiting scholar at University of Illinois at Urbana-Champaign Urbana, USA, from January 2010 to May 2010. His current research interests include design of unmanned aerial vehicles (UAV), wireless sensor network control, networked control systems and multi-agent control systems.



Yuanqing Xia (M'15-SM'16) was born in Anhui Province, China, in 1971. He received the B.S. degree from the Department of Mathematics, Chuzhou University, Chuzhou, China, in 1991, the M.S. degree in fundamental mathematics from Anhui University, Wuhu, China, in 1998, and the Ph.D. degree in control theory and control engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2001. His current research interests are in the fields of networked control systems, robust control and signal processing, active disturbance rejection control and flight control. He has published 8 monographs with Springer and Wiley, and more than 200 papers in journals. He has obtained Second Award of the Beijing Municipal Science and Technology (No. 1) in 2010, Second National Award for Science and Technology (No. 2) in 2011, and Second Natural Science Award of The Ministry of Education (No. 1) in 2012. He is a Deputy Editor of the Journal of the Beijing Institute of Technology, Associate Editor of Acta Automatica Sinica, Control Theory and Applications, the International Journal of Innovative Computing, Information and Control, and the International Journal of Automation and Computing.



https://dspace.org/

Optimal tracking guidance for aeroassisted spacecraft reconnaissance mission based on receding horizon control

Chai, Runqi; Savvaris, Al; Tsourdos, Antonios; Chai, Senchun; Xia, Yuanqing

2018-01-25
IEEE
http://dx.doi.org/10.1109/TAES.2018.2798219
Runqi Chai, Al Savvaris, Antonios Tsourdos et. al. Optimal tracking guidance for aeroassisted spacecraft reconnaissance mission based on receding horizon control. IEEE Transactions on Aerospace and Electronic Systems, Volume 54, Issue 4, August 2018, pp1575-1588

Downloaded from a DSpace repository. https://dspace.org/