

This item was submitted to [Loughborough's Research Repository](#) by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

Model predictive control for non-linear missiles

PLEASE CITE THE PUBLISHED VERSION

PUBLISHER

© IMechE / Professional Engineering Publishing

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Hu, Xiao-Bing, and Wen-Hua Chen. 2008. "Model Predictive Control for Non-linear Missiles". figshare.
<https://hdl.handle.net/2134/3757>.

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Model predictive control for non-linear missiles

X-B Hu^{1*} and W-H Chen²

¹Department of Informatics, University of Sussex, Brighton, UK

²Department of Aeronautical and Automotive Engineering, Loughborough University, Leicestershire, UK

The manuscript was received on 10 December 2006 and was accepted after revision for publication on 23 August 2007.

DOI: 10.1243/09596518JSCE394

Abstract: The present paper reports an attempt of applying model predictive control (MPC) to design an autopilot for a non-linear missile. The non-linear, fast dynamics of the missile raise three issues in the design of an MPC algorithm: the choice of the MPC performance index, in particular the terminal weighting term, to compromise the performance and the stability requirements; loss of the global minimum in the online optimization since it is a non-linear optimization; and the computational time limitation imposed by the fast sampling requirement. For the first issue, a procedure is developed to determine the terminal weighting term using a new representation of the control sequence in the moving horizon. For the other two issues, a new initial control profile and an associated control strategy are adopted in each optimization routine. It is shown that the new MPC algorithm can guarantee stability, even when a local minimum is attained in the online optimization or the optimization process has to stop owing to the limitation of the sampling time. Simulation results carried on the missile show that good performance and stability are achieved by the new MPC algorithm, whereas four other current MPC algorithms lose their stability.

Keywords: non-linear systems, model predictive control, missiles, stability

1 INTRODUCTION

Model predictive control (MPC), also referred to as receding horizon control (RHC), has been widely adopted in the process industry [1–4]. However, owing to various reasons, it has not been widely accepted by many other areas such as electrical, mechanical, and aeronautical engineering where most of the systems possess strong non-linearity and fast dynamics. The current paper reports an attempt at applying model predictive control to design a lateral autopilot for a missile. It is found that, in order successfully to apply MPC in a system such as the missile, several issues need to be addressed. The first issue is how to choose the performance index to be optimized, in particular the terminal weighing term, in MPC. For a system with non-linear dynamics, a non-linear optimization problem has to be solved online in each step. It is well known that there is no guarantee that the global minimum can be found for a non-linear optimization problem. The second issue addresses how to maintain stability of a non-

linear system under the MPC when a control profile generated by an optimizer stopping at a local minimum is implemented. The third issue is how to guarantee stability in the presence of limitation to online computational time. Allowable sampling interval for a system with fast dynamics is quite small and, therefore, a decision needs to be reached regarding what control strategy should be employed when the online optimization is not completed.

For the first issue, it is well known that the terminal weighting term in the performance index is introduced for the stability requirement of MPC [4]. Several methods to choose the terminal term for linear systems have been proposed [5–16], and recently these results have been extended to non-linear systems, e.g. see references [3], [5], and [10]. However, how to choose the terminal weighting term to compromise the performance and stability requirements remains largely unsolved, and has been identified in reference [17] as one of the main obstacles in the application of the MPC. This is mainly a result of the fact that it is quite difficult to estimate the stability region of an MPC algorithm, in particular for non-linear systems. In most of the current work, the stability region is estimated by the terminal region. Therefore, the terminal term is chosen based on the

* Corresponding author: Department of Informatics, University of Sussex, Brighton, UK. email: dr_xiaobinghu@hotmail.co.uk

terminal region rather than the stability region in the above-mentioned papers. For linear discrete time systems, reference [18] proposes a method to estimate the stability region directly by parameterizing the MPC sequence in the moving horizon. This idea will be extended to the non-linear missile control problem in the current paper. A new approach is then to determine the terminal weighting term that compromises the performance and the stability requirements.

In the implementation of MPC for non-linear systems, a non-linear optimization problem, which is quite time consuming, needs to be solved in real time. It is well known that only a local minimum might be attained. For a non-linear optimization problem, the implementation of such a control sequence implies that the MPC may lose not only its optimality but also its stability. It is even worse for a non-linear system with fast dynamics such as missiles. In this case the optimization process has to stop before the next sampling time arrives. One way to tackle this problem is to reduce the online computational burden by developing suboptimal but fast MPC algorithms [19].

Another way, the choice of the initial control profile, is adopted in the present paper. It is crucial that at each optimization routine, the optimization process starts from a 'good' initial control sequence. This is even more important for the MPC of non-linear systems. In the MPC of non-linear systems, it is very likely that a 'poorly' chosen initial sequence ends up with the loss of global optimum and then good performance even when there is no computational time limitation, while a good initial control sequence implies that stability and reasonable performance can be guaranteed even if only several iterations in the online optimization are allowed. Recently, Lyapunov-based model predictive controllers have been proposed that guarantee initial feasibility of the optimization problem from an explicitly characterized set of initial conditions, and also initialize the optimization problem with a feasible initial guess for non-linear systems without uncertainties [20] as well as in the presence of uncertainties [21]. In the present paper, based on a new parameterized control sequence in the moving horizon, a new initial control profile is suggested. It is then successfully applied to the control of missiles where many currently widely used methods for choosing initial control profiles fail.

The remainder of the current paper is organized as follows. Section 2 describes the non-linear missile problem. The choice of the performance index is discussed in section 3 and section 4 is devoted to the choice of initial control sequence and the design of

new MPC algorithm. Simulation results carried on the non-linear lateral dynamics of the missile are reported in section 5. Finally the paper ends with conclusions in section 6.

2 NON-LINEAR MISSILE CONTROL PROBLEM

The missile model to be considered in this paper has an extended medium range air-to-air technology (EMRAAT) airframe. The EMRAAT missile is a paper design used to explore a bank-to-turn (BTT) steering logic for the control of air-to-air missiles. Conventionally, such missiles are axisymmetric and use a skid-to-turn (STT) steering logic whereby motion is controlled by a set of cruciform fins at the tail. These are used to skid the missile through a turn in response to a sideslip demand while maintaining zero roll rate. Several factors have driven the desire to move from the conventional axisymmetric missile shape and the associated STT logic towards less conventional non-axisymmetric airframes. These factors include the desire to reduce drag, the need conformally to carry missiles on the aircraft, and the drive towards reducing the radar signature of the missile for low observable purposes. These missiles adopt configurations that are more aircraft-like and hence possess preferred orientation and preferred manoeuvre planes. This renders the STT logic redundant since non-axisymmetric airframes are difficult to skid and the aerodynamic coupling between sideslip and other degrees of freedom becomes very significant. As an alternative, a BTT steering logic is proposed and the EMRAAT may be used to help to design suitable controllers. As can be seen in Fig. 1, the EMRAAT airframe is longitudinally non-axisymmetric with four tail control fins and two wings approximately half way along the fuselage.

The equations of motion of this missile are fully derived in reference [22]. The motion is described by eight non-linear ordinary differential equations. They are given for the incidence time derivative, the sideslip time derivative, the rate time derivatives and the actuator deflection time derivatives. Figure 1 shows four tail control fins to generate moments about three axes. The actual fin deflections are

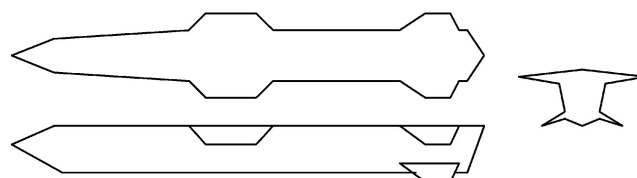


Fig. 1 EMRAAT airframe [20]

given by a mixing logic, which related the effective deflections about each of the axes to the actual deflections required. The missile dynamics in the present paper are given in terms of these effective deflections. Roll rate p , yaw rate r , sideslip angle β , rank angle ϕ , effective roll control input δ_p , and effective yaw control input δ_r , play an important role in the lateral dynamics of the missile.

It is desirable to avoid exciting coupling modes with the flight control system and as such the controller designed here comprises separate pitch and roll autopilots/controller to achieve attack angle and bank angle demands from the guidance laws of the missile. Clearly this approach will not avoid all of the coupling, but in particular the coupling between the product of attack angle and roll rate with sideslip may be minimized. Since a pitch controller and a roll controller are required, a key assumption must be made. Those full equations of motion may be decoupled into longitudinal and lateral-directional modes. Clearly this will introduce some error into the controller since the models of motion are closely coupled. However, the assumption simplifies the analysis and design in the final stage.

Finally, the flight case of the missile must be discussed. The flight condition considered in the present paper is that the missile travels at Mach 2 and at an altitude of 30 000 ft. This flight condition is representative of a missile as it nears its target: the engine has ceased thrusting and the missile carries out violent manoeuvres in order to intercept the target. The flight condition data include dynamic pressure, density and velocity, and – importantly – a full set of aerodynamic derivatives, which are crucial for describing the dynamics of the missile. A full listing of all the flight condition data may be found in references [19] and [22]. Several non-linear control methods have been designed and tried on this missile [19], [22]. The current paper discusses how to design an autopilot for the lateral dynamics of the missile using the MPC technique.

There are some requirements in the design of a roll controller [22]. The primary requirement of the roll controller is to roll the missile to a commanded bank angle up to a maximum of 180° in either direction. These commands must be followed as quickly and precisely as possible. Furthermore, the real-world hardware restrictions must be enforced. The actuator deflection cannot exceed 45° in either direction. The propulsion system of the missile is air-breathing. As such excessive sideslip will cause a disruption of flow into the air intake causing the engine to flame out – an undesirable situation. For this reason, the sideslip angle must remain below 5° .

At this stage, for the sake of simplicity, the maximum deflection rate, actuator dynamics, disturbances, and system uncertainties are not considered.

After the EMRAAT missile specific terms, the flight condition specific terms, and $\alpha = 0^\circ$ (angle of attack), $\theta = 0^\circ$ (pitch angle), $q = 0^\circ/\text{s}$ (pitch rate), $\cos(\beta\pi/180) \approx 1$ (because $|\beta| \leq 5^\circ$) have been substituted in the full equations of motion, the equations of motion for the lateral dynamics of the missile are obtained as the following

$$\dot{p} = f_1(p, r, \beta, \phi, \delta_p, \delta_r) = 0.000\,179p^2 - 0.0184r^2 - 0.002\,32pr - 2.177p + 0.8055r + 1001.167\beta - 1243.48\delta_p - 959.523\delta_r \quad (1)$$

$$\dot{r} = f_2(p, r, \beta, \phi, \delta_p, \delta_r) = 0.003\,88p^2 + 0.000\,181r^2 + 0.000\,5024pr - 0.003\,54p - 0.605\,26r + 96.093\beta + 17.524\delta_p - 75.995\delta_r \quad (2)$$

$$\dot{\beta} = f_3(p, r, \beta, \phi, \delta_p, \delta_r) = 1.000\,398r - 0.368\beta - 0.000\,018\,03p + 0.0166 \sin \bar{\phi} \quad (3)$$

$$\dot{\phi} = f_4(p, r, \beta, \phi, \delta_p, \delta_r) = p \quad (4)$$

$$\bar{\phi} = \phi\pi/180 \quad (5)$$

Thus, the state vector and input vector are give by

$$\mathbf{x} = [p \quad r \quad \beta \quad \phi]^T, \quad \mathbf{u} = [\delta_p \quad \delta_r]^T \quad (6)$$

Owing to the actuator limits mentioned earlier, the control input is subject to the constraint

$$\mathbf{u}(t, \mathbf{x}(t)) \in U = \{\mathbf{u} = [\delta_p, \delta_r]^T : |\delta_p| \leq 45^\circ, |\delta_r| \leq 45^\circ\} \quad (7)$$

3 CHOICE OF THE PERFORMANCE INDEX

To implement an MPC algorithm on the lateral dynamics of the missile, a performance index to be optimized online should be determined. In general the MPC performance index is chosen as

$$J(t) = g(\hat{\mathbf{x}}(t + T_H)) + \int_0^{T_H} \hat{\mathbf{x}}(t + \tau)^T \mathbf{Q} \hat{\mathbf{x}}(t + \tau) + \hat{\mathbf{u}}(t + \tau; \mathbf{x}(t))^T \mathbf{R} \hat{\mathbf{u}}(t + \tau; \mathbf{x}(t)) d\tau \quad (8)$$

where T_H is the length of the moving horizon, $g(\cdot)$ is the terminal weighting term, $\mathbf{Q} \geq 0$ and $\mathbf{R} \geq 0$ are the state weighting matrix and the control weighting matrix, respectively. In order to distinguish the real variables from the variables in the moving horizon time frame, hatted variables are used in the moving horizon time frame. $\hat{\mathbf{u}}(\cdot; \mathbf{x}(t))$ explicitly indicates that the control profile $\hat{\mathbf{u}}$ depends on the state measurement $\mathbf{x}(t)$ at time t . It is required that the terminal

weighting term $g(x)$ is a continuous, differentiable function of x , $g(0) = 0$, and $g(x) > 0$ for all $0 \neq x \in R^4$. A typical choice of $g(x)$, as used in this paper, is given by

$$g(x(t)) = x(t)^T P x(t) \quad (9)$$

where $0 < P \in R^{4 \times 4}$ is called the terminal weighting matrix.

The integral part in the performance index represents the performance requirement and it can be chosen by the linear quadratic regulator (LQR) method. The terminal weighting is imposed owing to the stability requirement. Unfortunately, few methods are available to determine the terminal term, or P , for non-linear systems. Even for linear MPC, it is often chosen based on the terminal region rather than the stability region. In the present paper, for the missile control problem, a procedure to choose the terminal region is proposed based on the stability region.

3.1 LDI representation of missile lateral dynamics

The non-linear lateral dynamics given by equation (1) to equation (5) can be represented by its linear differential inclusion (LDI)

$$\begin{bmatrix} \dot{p} \\ \dot{r} \\ \dot{\beta} \\ \dot{\phi} \end{bmatrix} = A(p, r, \beta, \phi, \delta_p, \delta_r) \begin{bmatrix} p \\ r \\ \beta \\ \phi \end{bmatrix} + B(p, r, \beta, \phi, \delta_p, \delta_r) \begin{bmatrix} \delta_p \\ \delta_r \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial f_1}{\partial p} & \frac{\partial f_1}{\partial r} & \frac{\partial f_1}{\partial \beta} & \frac{\partial f_1}{\partial \phi} \\ \frac{\partial f_2}{\partial p} & \frac{\partial f_2}{\partial r} & \frac{\partial f_2}{\partial \beta} & \frac{\partial f_2}{\partial \phi} \\ \frac{\partial f_3}{\partial p} & \frac{\partial f_3}{\partial r} & \frac{\partial f_3}{\partial \beta} & \frac{\partial f_3}{\partial \phi} \\ \frac{\partial f_4}{\partial p} & \frac{\partial f_4}{\partial r} & \frac{\partial f_4}{\partial \beta} & \frac{\partial f_4}{\partial \phi} \end{bmatrix} \begin{bmatrix} p \\ r \\ \beta \\ \phi \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial \delta_p} & \frac{\partial f_1}{\partial \delta_r} \\ \frac{\partial f_2}{\partial \delta_p} & \frac{\partial f_2}{\partial \delta_r} \\ \frac{\partial f_3}{\partial \delta_p} & \frac{\partial f_3}{\partial \delta_r} \\ \frac{\partial f_4}{\partial \delta_p} & \frac{\partial f_4}{\partial \delta_r} \end{bmatrix} \begin{bmatrix} \delta_p \\ \delta_r \end{bmatrix}$$

$$= \begin{bmatrix} 3.5802 \times 10^{-5}p - 2.3000 \times 10^{-3}r & -3.6800 \times 10^{-2}r - 2.3000 \times 10^{-3}p & 1.0012 \times 10^3 & 0 \\ -2.117 & +8.0560 \times 10^{-1} & 0 & 0 \\ 7.8000 \times 10^{-3}p + 5.0240 \times 10^{-4}r & 3.6226 \times 10^{-4}r + 5.0240 \times 10^{-4}p & 9.6093 \times 10^1 & 0 \\ +3.5000 \times 10^{-3} & -6.0530 \times 10^{-1} & 0 & 0 \\ -1.8030 \times 10^{-5} & -9.9960 \times 10^{-1} & -3.6800 \times 10^{-1} & 2.8972 \times 10^{-4} \times \cos \bar{\phi} \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ r \\ \beta \\ \phi \end{bmatrix} + \begin{bmatrix} -1243.5 & -959.5304 \\ 17.5241 & -75.9936 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_p \\ \delta_r \end{bmatrix} \quad (10)$$

Let

$$A(\cdot) = A(p, r, \beta, \phi, \delta_p, \delta_r), B(\cdot) = B(p, r, \beta, \phi, \delta_p, \delta_r) \quad (11)$$

and $a(i, j)$ denotes the element in i th row and j th column.

Let Θ denote the set of $[A(\cdot), B(\cdot)]$ when the missile operates within a given range. Let $Co\Theta$ denote the minimum convex hull, which covers the set Θ , and the corresponding system is called the relaxed LDI. It is known that every trajectory of the non-linear system is also a trajectory of the relaxed LDI. As a result a MPC that can stabilize the relaxed LDI can also stabilize the original non-linear system [23].

At first glance, there are five variable elements in $A(\cdot)$, and $B(\cdot)$ is a constant matrix. However, it is found that within the operation range, compared with the constant 2.177, the variation caused by p and r in the element $a(1, 1)$ is very small and thus can be ignored. Similarly, $a(2, 2)$ can also be considered as a constant. Therefore $Co\Theta$ has eight vertices depending on $a(1, 2)$, $a(2, 1)$, and $a(3, 4)$. These vertices can be calculated according to the range of p , r , and ϕ . Denote these eight vertices as

$$V(l) = [A_l \ B_l], \quad l = 1, \dots, 8 \quad (12)$$

and the corresponding eight linear vertex systems are given by

$$\dot{x}(t) = A_l x(t) + B_l u(t), \quad u(t) \in U, \quad l = 1, \dots, 8 \quad (13)$$

When the missile is digitally controlled with a sampling time T_{sampling} , the corresponding discrete time vertex systems are

$$x(k+1) = \bar{A}_l x(k) + \bar{B}_l u(k), \quad u(k) \in U, \quad l = 1, \dots, 8 \quad (14)$$

with the discrete performance index

$$J(k) = \mathbf{x}(k+N|k)^T \mathbf{P} \mathbf{x}(k+N|k) + \sum_{i=1}^{N-1} (\mathbf{x}(k+i|k)^T \mathbf{Q} \mathbf{x}(k+i|k) + (\mathbf{u}(k+i|k)^T \mathbf{R} \mathbf{u}(k+i|k)) \quad (15)$$

where \mathbf{P} is the terminal weighting matrix, N is the number of sampling intervals included in a moving horizon, and $(\cdot|k)$ indicates the associated variable is used in the moving horizon at the time instant k .

Here, it should be noted that the computational delay caused by the online optimization is taken into account in the performance index. For details, please refer to reference [10].

Before presenting further results, the definitions of terminal region and stability region are necessary.

Definition 1. The terminal region ν is defined as a region where, once the state $\mathbf{x}(k+N|k)$, under the control $\mathbf{u}^*(k+i|k)$, $i=0, \dots, N-1$ yielded by the solution to the MPC optimization problem, arrives, there exists a control $\mathbf{u}(k+N|k)$ which can steer the state to the origin.

Definition 2. The stability region M refers to a set of initial state points \mathbf{x}_0 from which there exists an open-loop control profile $\mathbf{u}^*(k+i|k)$, $i=0, \dots, N-1$, such that the state trajectory $\mathbf{x}(k+i|k)$, $i=1, \dots, N$, is steered into the terminal region ν at the terminal time $t+T_H$.

3.2 Determining the terminal term

The method outlined in reference [18], which is proposed to choose the terminal term to maximize the stability region for linear systems, will be extended in this section to the case where the LDI of a non-linear system is covered by a convex hull defined by several vertex systems as in equation (14). The underlying idea in reference [18] is to use a new representation of the control sequence in the moving horizon and by such means it is possible to determine the terminal term directly based on the stability region rather than on the terminal region. Another significant advantage of this approach is that, as will be discussed later, it provides a new initial control profile for online optimization, which enables MPC to work in a larger range.

Suppose that there exist $0 < \mathbf{W} \in R^{4 \times 4}$, $\bar{\mathbf{W}} \in R^{4 \times 2}$, $1 \leq \mu \in R$, $0 < \mathbf{S} \in R^{4 \times 4}$, and $\mathbf{S}_i \in R^{4 \times 2}$ such that the following conditions hold

$$\begin{bmatrix} \mathbf{S} & (\bar{\mathbf{A}}_l^N \mathbf{S} + \bar{\mathbf{I}}_l \bar{\mathbf{S}})^T \\ \bar{\mathbf{A}}_l^N \mathbf{S} + \bar{\mathbf{I}}_l \bar{\mathbf{S}} & \mathbf{W} \end{bmatrix} \geq 0, l=1, \dots, 8 \quad (16)$$

$$\begin{bmatrix} \mathbf{Y} & \mathbf{S}_i \\ (\mathbf{S}_i)^T & \mathbf{S} \end{bmatrix} \geq 0, Y_{jj} \leq 45^2, i=0, \dots, N-1; j=1, 2 \quad (17)$$

$$\begin{bmatrix} \mathbf{W} & (\bar{\mathbf{A}}_l \mathbf{W} + \bar{\mathbf{B}}_l \bar{\mathbf{W}})^T & (\mathbf{Q}^{1/2} \mathbf{W})^T & \bar{\mathbf{W}}^T \\ \bar{\mathbf{A}}_l \mathbf{W} + \bar{\mathbf{B}}_l \bar{\mathbf{W}} & \mathbf{W} & 0 & 0 \\ \mathbf{Q}^{1/2} \mathbf{W} & 0 & \mu \mathbf{I} & 0 \\ \bar{\mathbf{W}} & 0 & 0 & \mu \mathbf{R}^{-1} \end{bmatrix} \geq 0 \quad l=1, \dots, 8 \quad (18)$$

$$\begin{bmatrix} \mathbf{Y} & \bar{\mathbf{W}} \\ \bar{\mathbf{W}}^T & \mathbf{W} \end{bmatrix} \geq 0, Y_{jj} \leq 45^2, j=1, 2 \quad (19)$$

where

$$\bar{\mathbf{S}} = [\mathbf{S}_0^T \dots \mathbf{S}_{N-1}^T]^T \quad (20)$$

$$\bar{\mathbf{I}}_l = [\bar{\mathbf{A}}_l^{N-1} \bar{\mathbf{B}}_l \dots \bar{\mathbf{A}}_l^0 \bar{\mathbf{B}}_l] \quad (21)$$

Then when the terminal weighting matrix \mathbf{P} , terminal control gain \mathbf{K}_{term} , stability region matrix \mathbf{Z} and the associated control gain \mathbf{K}_{stab} are chosen as

$$\mathbf{P} = \mathbf{W}^{-1} \mu, \mathbf{K}_{\text{term}} = \bar{\mathbf{W}} \mathbf{P} \mu^{-1}, \mathbf{Z} = \mathbf{S}^{-1}, \mathbf{K}_{\text{stab}} = \bar{\mathbf{S}} \mathbf{Z} \quad (22)$$

according to reference [18], the corresponding terminal region is given by

$$\nu = \{\mathbf{x} \in R^4 | \mathbf{x}^T \mathbf{P} \mathbf{x} < \mu\} \quad (23)$$

This is because the conditions (18) and (19) are the same as the conditions in reference [13] to determine the terminal region except that the extra parameter μ is introduced. It has been proved that the introduction of μ can significantly increase the terminal region [11].

Actually, μ is also a tuning knob for trade-off between performance and stability. That is, a large μ gives a large stability region, but results in less influence from the integral part. By fixing μ to a

proper value, the trade-off between stability and performance can be achieved [11].

The following theorem will state that the stability region for the set of discrete time systems (14) can be estimated by

$$M = \{x \in R^4 | x^T Z x < 1\} \quad (24)$$

Theorem. For a set of discrete-time systems (14), suppose that there exist $0 < W \in R^{4 \times 4}$, $\bar{W} \in R^{4 \times 2}$, $1 \leq \mu \in R$, $0 < S \in R^{4 \times 4}$, and $S_i \in R^{4 \times 2}$ such that the conditions (16) to (19) hold. Then, under an MPC with the terminal region ν , the closed-loop system is asymptotically stable about the origin for all $k > 0$ and all initial states within the stability region M subject to input constraint

$$|u_i| \leq \bar{u}_i, \quad i = 1, 2 \quad (25)$$

Proof. Set the initial control sequence as

$$\begin{aligned} u(k+i|k) &= K(k+i)x(k), \quad K(k+i) = S_i Z \\ i &= 1, \dots, N-1 \end{aligned} \quad (26)$$

Substituting the above control law to the system (14) gives

$$x(k+N|k) = (\bar{A}_l^N + \bar{\Gamma}_l K_N(k))x(k), \quad l = 1, \dots, 8 \quad (27)$$

where

$$K_N(k) = \begin{bmatrix} K(k) \\ \vdots \\ K(k+N-1) \end{bmatrix} \quad (28)$$

For all $x(k)$ within the set M , if

$$Z \geq (\bar{A}_l^N + \bar{\Gamma}_l K_N(k))^T P \mu^{-1} (\bar{A}_l^N + \bar{\Gamma}_l K_N(k)) \quad (29)$$

then

$$\begin{aligned} 1 &\geq x(k)^T (\bar{A}_l^N + \bar{\Gamma}_l K_N(k))^T P \mu^{-1} (\bar{A}_l^N + \bar{\Gamma}_l K_N(k)) x(k) \\ &= x(k+N|k)^T P \mu^{-1} x(k+N|k) \end{aligned}$$

This implies that if the condition (29) is satisfied, there exists a control sequence to steer any initial state within the set into the terminal set ν .

Using the transforms (22) and (26), the condition (29) can be expressed in the same LMI form as the condition (16). It can be shown that the condition (17) guarantees that the input constraint (25) is satisfied.

In other words, the conditions (16) and (17) guarantee that for any initial state x_0 within the set M , there exists a control sequence such that the state arrives in the terminal set ν at the time instant $k+N$. The conditions (18) and (19) ensure that there exists a terminal control satisfying the control constraints which can steer every state within the terminal set ν to the origin. Hence the results.

QED

The Theorem gives the estimation of the stability region M for the initial state instead of the terminal region ν for the terminal state. The stability region can be maximized by solving the optimization problem

$$\min_{S, \bar{S}} \log(\det(S^{-1})) \quad (30)$$

subject to the conditions (16) to (19).

Then the terminal term in the performance (15) for the missile control is determined by equation (22), and the associated terminal region and stability region are given by equations (23) and (24) respectively.

4 INITIAL CONTROL PROFILE AND MPC ALGORITHM

After the performance index of an MPC algorithm is determined, the online implementation issues need to be addressed. The lateral dynamics of the missile are continuous and are digitally controlled with sampling time T_{sampling} . At each sampling time, also referred to as time instant, after the system state is measured, the online optimization problem to be solved is formulated as

$$\begin{aligned} &\min_{u(k+1|k), \dots, u(k+N-1|k)} J(k) \\ &\begin{cases} \hat{u}(t) = u(k+i|k), & (k+i)T_{\text{sampling}} \leq t < (k+i+1)T_{\text{sampling}}, & i = 1, \dots, N-1; k \geq 0 \\ \hat{u}(t) = u^*(k|k-1), & kT_{\text{sampling}} \leq t < (k+1)T_{\text{sampling}}, & k > 0 \\ \hat{u}(t) = K_{\text{stab}}(1:2, :)x_0, & 0 \leq t < T_{\text{sampling}}, & k = 0 \end{cases} \end{aligned} \quad (31)$$

$$\begin{aligned} &\begin{cases} \hat{u}(t) = u(k+i|k), & (k+i)T_{\text{sampling}} \leq t < (k+i+1)T_{\text{sampling}}, & i = 1, \dots, N-1; k \geq 0 \\ \hat{u}(t) = u^*(k|k-1), & kT_{\text{sampling}} \leq t < (k+1)T_{\text{sampling}}, & k > 0 \\ \hat{u}(t) = K_{\text{stab}}(1:2, :)x_0, & 0 \leq t < T_{\text{sampling}}, & k = 0 \end{cases} \end{aligned} \quad (32)$$

subject to the input constraint (7), the lateral dynamics (1) to (5) and (32), where $\mathbf{u}^*(k|k-1)$ is given by the optimizer at the past sampling instant. Owing to the computational delay, it takes up to one sampling interval to calculate the control sequence, i.e. the maximum computational time available for the optimizer is T_{sampling} .

The following is the new MPC algorithm

Step 1. Measure the state $\mathbf{x}(t_0)$. Start the MPC algorithm by implementation of the control $\mathbf{u}(t_0) = \mathbf{K}_{\text{stab}}\mathbf{x}(t_0)$. Let $k = 0$.

Step 2. Check whether the state $\mathbf{x}(k)$ is within the set ν . If it is, go to step 4; otherwise, go to step 3.

Step 3. Determine an initial feasible control profile according to equation (32) and

$$\mathbf{u}(k+i|k) = \mathbf{K}_{\text{stab}}(2i-1:2i, :)\mathbf{x}(k), i = 1, \dots, N-1 \quad (33)$$

Here a feasible control profile means it satisfies the input constraint and is able to steer the system state to the origin or the terminal region. Start the online optimization process with the initial control sequence to solve the optimization problem (equation (31)). If the optimization is completed before the next sampling time arrives, execute the first element of the yielded control sequence. When the online optimization cannot be completed within the specified sampling time, the latest control sequence yielded by the optimizer is tested to see whether it can drive the state into the terminal region. If it can, it will be implemented. Otherwise, execute the first element of the initial feasible control sequence.

Step 4. Determine the initial control profile using the terminal control. Start the online optimization process with the initial control sequence to solve the optimization problem (equation (31)). Execute the first element of the yielded control sequence.

Step 5. Measure the state $\mathbf{x}(t)$. Let $\mathbf{x}(k) = \mathbf{x}(t)$. Let $k = k + 1$, i.e. $t = t + T_{\text{sampling}}$ and go to step 2.

When the system state arrives in the terminal region ν , the stability is guaranteed by using the terminal control. For the state outside the terminal region ν , the first part of the new MPC is to steer the system state from M into ν . This is a dual-mode control in

the sense that the different initial control profiles are used for the online optimizer and different control strategies are used when the online optimization is not completed for a state within or outside the terminal region. However, it should be noticed that the same online optimization problem with the same performance index needs to be solved for both cases.

5 MPC FOR THE LATERAL DYNAMICS

Before applying MPC for the missile control problem, there are several parameters to be determined, i.e. T_{sampling} , T_H , N , \mathbf{Q} , and \mathbf{R} .

A typical flight control system has the sampling frequency of 80 Hz. Therefore $T_{\text{sampling}} = 0.0125$ s is chosen. T_H is determined based on the trade-off between optimality and online computational burden. Here $T_H = 0.05$ s is chosen, which gives $N = 4$. \mathbf{Q} and \mathbf{R} can be tuned and determined according to the performance of an LQR. In the simulation, \mathbf{Q} and \mathbf{R} are chosen as

$$\mathbf{Q} = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 250 & 0 \\ 0 & 0 & 0 & 5000 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (34)$$

5.1 Choice of the terminal weighting and the stability region

Following the procedure developed in section 3, the terminal matrix \mathbf{P} , μ , and the associated terminal control gain \mathbf{K}_{term} are obtained as

$$\mathbf{P} = \mu \begin{bmatrix} 0.4478 & -0.0202 & 0.0825 & 1.7776 \\ -0.0202 & 0.4530 & -0.2167 & -0.0999 \\ 0.0825 & -0.2167 & 22.3971 & 2.4211 \\ 1.7776 & -0.0999 & 2.4211 & 57.2101 \end{bmatrix} \quad (35)$$

$$\mu = 1.3119 \times 10^3 \quad (36)$$

$$\mathbf{K}_{\text{term}} = \begin{bmatrix} 0.3715 & -0.5788 & 0.1744 & 1.4145 \\ -0.4547 & 0.7932 & 0.7984 & -1.6632 \end{bmatrix} \quad (37)$$

Then \mathbf{Z} and the associated feasible initial control gain sequence \mathbf{K}_{stab}

$$\mathbf{Z} = \begin{bmatrix} 1.9909 \times 10^{-5} & -8.5828 \times 10^{-6} & 5.9945 \times 10^{-4} & 4.3341 \times 10^{-4} \\ -8.5828 \times 10^{-6} & 3.5914 \times 10^{-4} & -0.0154 & -7.1358 \times 10^{-4} \\ 5.9945 \times 10^{-4} & -0.0154 & 1.1417 & 0.0772 \\ 4.3341 \times 10^{-4} & -7.1358 \times 10^{-4} & 0.0772 & 0.0844 \end{bmatrix} \quad (38)$$

$$\mathbf{K}_{\text{stab}} = \begin{bmatrix} 0.0552 & -0.6973 & 0.4358 & 0.4036 \\ 0.0132 & 0.8998 & 0.4901 & -0.1258 \\ 0.0941 & -0.7457 & 21.6451 & 4.4088 \\ 0.0008 & 0.9631 & -26.9118 & 0.8435 \\ 0.0730 & -0.6960 & 33.9518 & 3.5038 \\ 0.0240 & 0.9013 & -43.2202 & 1.8471 \\ 0.0679 & -0.5136 & 15.8675 & 1.9790 \\ 0.0316 & 0.6190 & -18.2080 & 0.9526 \end{bmatrix} \quad (39)$$

Since the state space is four-dimensional, it is difficult to plot either the terminal region or the stability region. However, the value of $\log(\det(\mathbf{P}\mu^{-1}))$ or $\log(\det(\mathbf{Z}))$ can be used to assess approximately the size of terminal region or stability region. It was found that $\log(\det(\mathbf{P}\mu^{-1})) = 5.4178$, while $\log(\det(\mathbf{Z})) = -22.1544$. This indicates that the stability region determined by the new MPC is much larger than the terminal region.

However, as is well known, the LDI technique is quite conservative. Simulation has shown that the actual stability region of the MPC algorithm under this performance index is larger than what is estimated. Nevertheless, this gives a practical way to choose the terminal weighting term for the missile control problem. How to reduce the conservativeness caused by the LDI representation of a non-linear system is an interesting problem worthy of further investigation.

Another way to enlarge the stability region further is to use a long predictive horizon. For example, when the predictive length is chosen as $N=7$, a new \mathbf{Z} is yielded with $\log(\det(\mathbf{Z})) = -27.3951$. Compared with the predictive length of $N=4$, a much larger stability region is obtained. However, if the terminal region is used to estimate the stability region, the length of

predictive horizon has no influence on the size of stability region, which indicates the deficiency of the method of using the terminal region to estimate the stability region.

It should be noted that the following closed-loop simulation is based on the original missile model given by equations (1) to (5), while the approximated LDI model in equations (12) to (14) is only used for designing MPC controllers. To solve the offline optimization problem (30) and the online optimization problem (31), the LMI toolbox and optimization toolbox in MATLAB are employed.

5.2 MPC performance without computational time limitation

In following simulation tests, the new MPC algorithm developed in the current paper, referred to as NMPC, is compared with four other methods to choose initial control profiles. These four methods are denoted as OMPC1, OMPC2, OMPC3, and OMPC4, respectively, in Table 1 where 'inheriting' means that at the time instant $k+1$, the initial control sequence for optimization is chosen as

$$\mathbf{u}(k+i|k+1)_{\text{initial}} = \mathbf{u}^*(k+i|k), i = 1, \dots, N \quad (40)$$

where $\mathbf{u}^*(k+i|k)$ denotes the final sequence yielded by the optimizer at the time instant k . For example, OMPC1 means that at the time instant 0, the initial control profile is chosen as zero and after that the initial control sequence for the online optimizer consists of two parts: the first $N-1$ components are carried from the sequence yielded by the optimizer in the past time instant, as in equation (40), and the last component is chosen as $\mathbf{K}_{\text{term}}\mathbf{x}(k)$.

First, suppose that there is no time limitation for the online optimization process. The purpose of this test is to find the average computational time

Table 1 Methods to set initial control profiles

Time	Initial control profile $\hat{\mathbf{u}}_{\text{initial}}(\cdot; \mathbf{x}(t))$				
	OMPC 1	OMPC 2	OMPC 3	OMPC 4	NMPC
$t = t_0$	$[0, \dots, 0]$	$[0, \dots, 0]$	$[0, \dots, 0]$	$\hat{\mathbf{u}}^*(\cdot; \mathbf{x}(t_0))$	$\mathbf{K}_{\text{stab}}(\cdot)\mathbf{x}(t_0)$
$t > t_0$	[Inheriting, $\mathbf{K}_{\text{term}}\mathbf{x}(t)$]	[Inheriting, 0]	$[0, \dots, 0]$	[Inheriting, $\mathbf{K}_{\text{term}}\mathbf{x}(t)$]	See the algorithm

required to perform the online optimization for different MPC algorithms with different initial control profiles. Figures 2 and 3 illustrate the online control performances of OMPCs and NMPC. For the lateral autopilot of the missile, the same control performances are achieved by the MPC algorithms with different choices of initial control sequence. It should be noticed that the initial state in the simulation shown in Figs 2 and 3 is outside the set ν . This also

implies that using a terminal region to estimate the stability region is quite conservative.

The simulation has been conducted for the missile starting from different initial states. Table 2 compares the average time and the maximum time consumed by the online optimization process for a same initial state. It indicates that as the time passes, the average time for all MPC algorithms reduces. This is mainly because, after 0.5 s (see Figs 2 and 3), the control

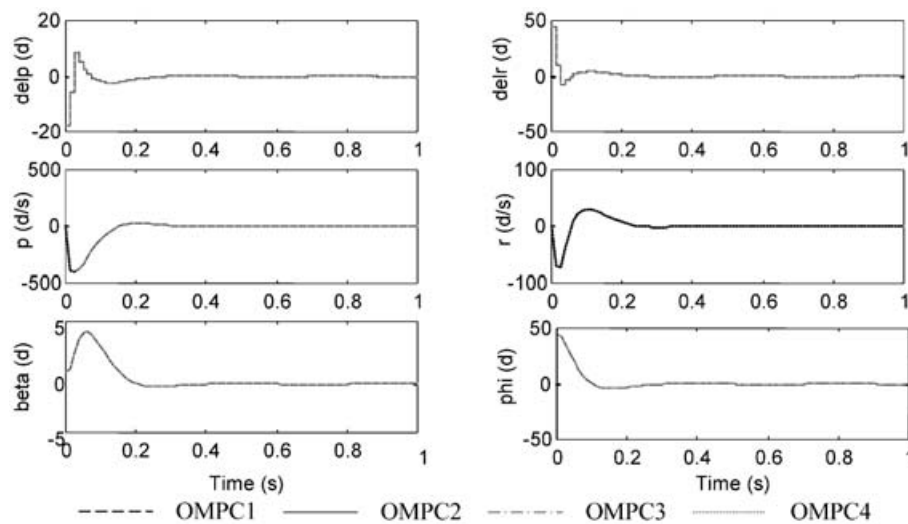


Fig. 2 Performances of OMPC without online computational time limitation ($x_0 = [0, 0, 1, 45]'$)

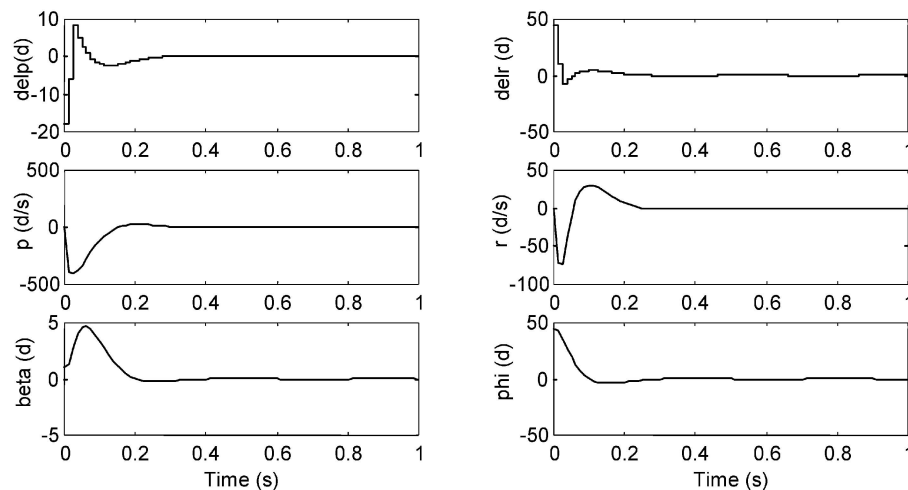


Fig. 3 Performances of NMPC without online computational time limitation ($x_0 = [0, 0, 1, 45]'$)

Table 2 Comparison of computational time ($x_0 = [0, 0, 1, 45]'$)

Simulation time (s)		0.25	0.5	1.0	Maximum computational time (s)
Average computational time of a run of online OP solver (s)	OMPC 1	1.0680	0.7675	0.6089	2.2000
	OMPC 2	0.5575	0.4447	0.4406	1.0500
	OMPC 3	0.5655	0.4640	0.4278	2.5800
	OMPC 4	0.8790	0.6303	0.4908	2.1900
	NMPC	1.1590	0.7965	0.6036	1.8800

effort for all MPC algorithms is close to zero and therefore it takes a short time to achieve the minimum. Table 2 shows that (a) OPMC1, OPMC2, and OPMC4 take less computational time than NMPC, and this is because they inherit the 'optimal' solution yielded by optimization at the past time instant; (b) the maximum computational time of a non-linear optimization routine depends on many factors, e.g. the method to set initial control sequence, the plant chosen for simulation tests, and the state. Although it might make little sense to compare the maximum computational times of these MPC algorithms, the maximum computational time can help to understand the influence of computational time limitation imposed by fast sampling.

5.3 MPC performance with computational time limitation

For a non-linear system with fast dynamics such as the missile, it is unlikely that all of the online optimization will be performed within a sampling interval. If the online optimization is not completed when time runs out, the intuitive way is to execute the latest solution yielded by the optimizer. This idea is adopted by all OMPs. NMPC uses the latest solution if and only if this solution is feasible, i.e. it can drive the state trajectory into the terminal region at the end of the predictive horizon. Otherwise, the associated initial control law, which has been worked out offline as in equation (39), is implemented to avoid losing stability.

In the simulation, the computational time limitation is set as 0.5 s. In reality, the computational time limitation must not exceed the sampling time, i.e. $T_{\text{sampling}} = 0.0125$ s. The choice of 0.5 s is based on

the justification that, in a practical implementation, special purpose-oriented computer and assemble code rather than personal computer and general MATLAB code as in the simulation will be employed, which can lead to a much higher online computing speed than in the simulation tests. As shown in Figs 4 and 5, OMPs cannot stabilize the system when the initial state is outside the terminal region, while in Figs 6 and 7, NMPC still achieves good control performances.

Another option for MPC to handle the computational time limitation is to use the terminal control as a back-up control law. That is, when the online optimization is not completed, the terminal control, rather than the latest control sequence yielded by the optimizer as in the simulation of Figs 4 and 5, is executed. However, as shown in Figs 8 and 9, for the same initial states as in the previous tests, all OMPs with K_{term} as a back-up control law still cannot stabilize the missile. This is because the terminal region ν is too small and both the initial states are outside the terminal region. However, Figs 6 and 7 show that NMPC works well for both cases. The feasible initial control sequence generated by the procedure in section 3 is very useful in the implementation of MPC for the lateral dynamics of the missile.

6 CONCLUSIONS

Motivated by the attempt to apply MPC to control missiles, several practical issues in the implementation of MPC for non-linear systems with fast dynamics have been identified in the current paper. Several

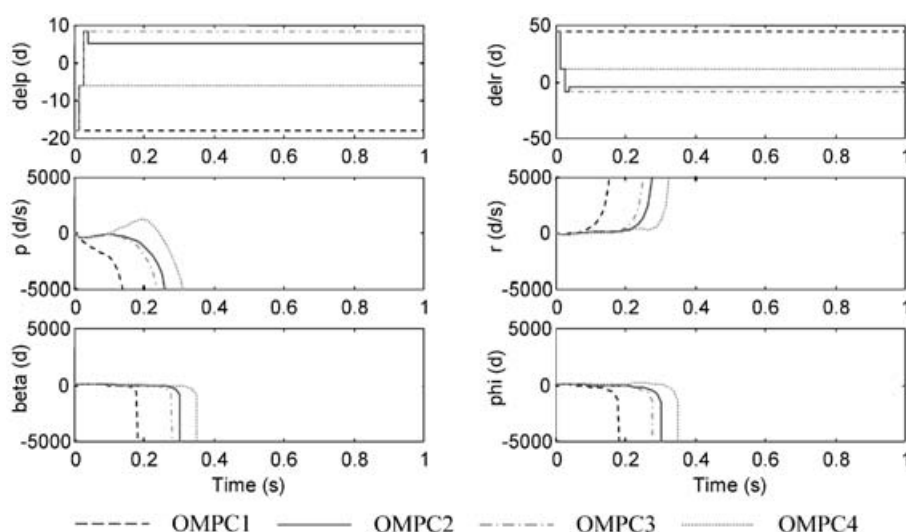


Fig. 4 Performances of all the OMPs with online computational time limitation ($x_0 = [0, 0, 1, 45]^T$)

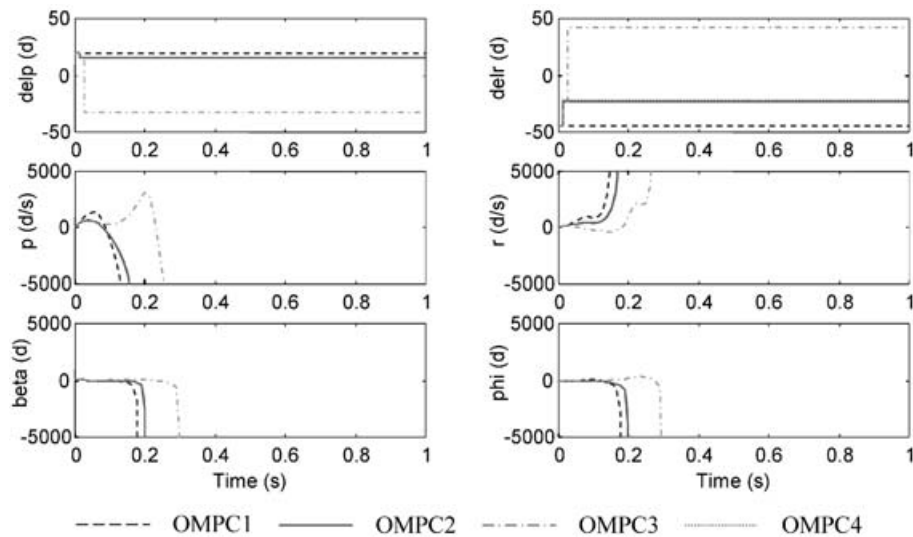


Fig. 5 Performances of OMPC with online computational time limitation ($x_0 = [0, 0, 5, -90]'$)

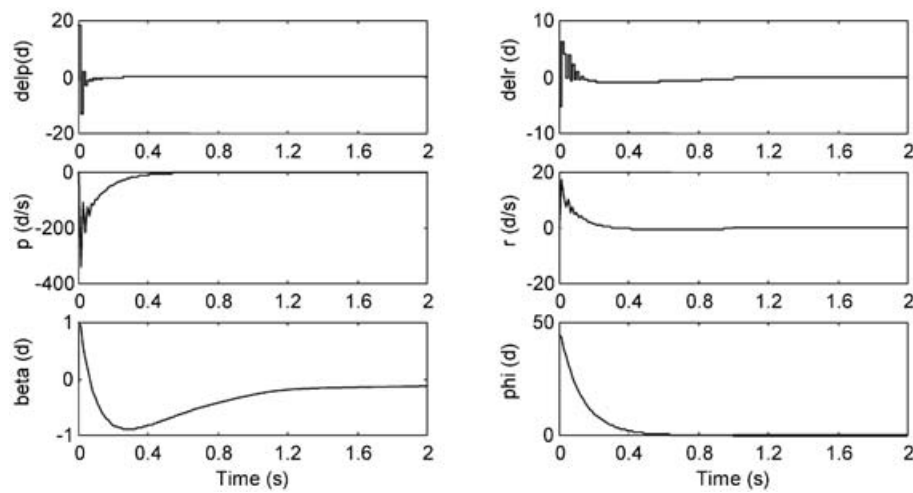


Fig. 6 Performance of NMPC with online computational time limitation ($x_0 = [0, 0, 1, 45]'$)

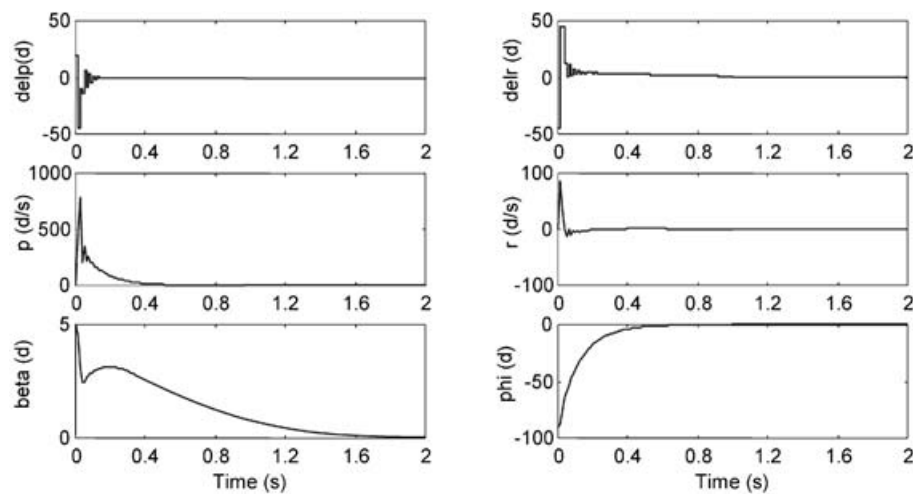


Fig. 7 Performance of NMPC with online computational time limitation ($x_0 = [0, 0, 5, -90]'$)

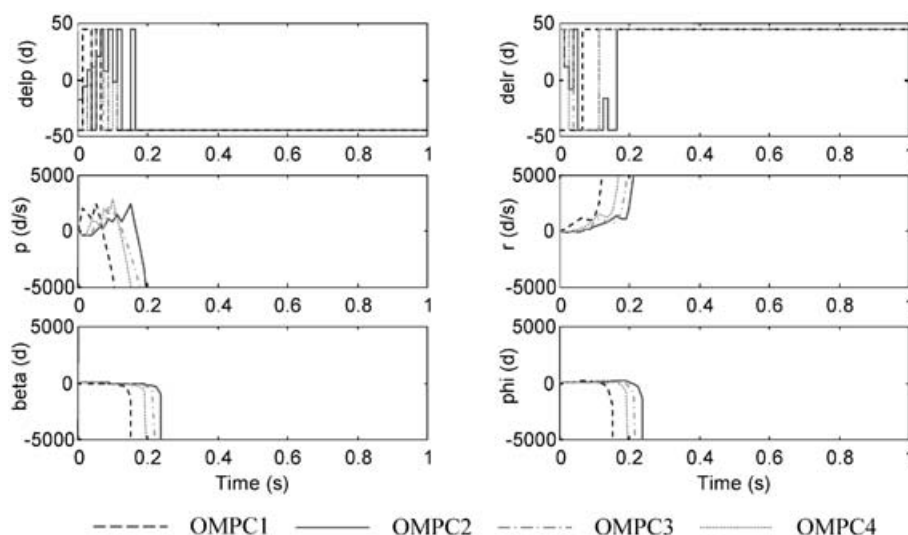


Fig. 8 Performances of all the OMPCs with online computational time limitation and K_{term} as back-up control gain ($x_0 = [0, 0, 1, 45]'$)

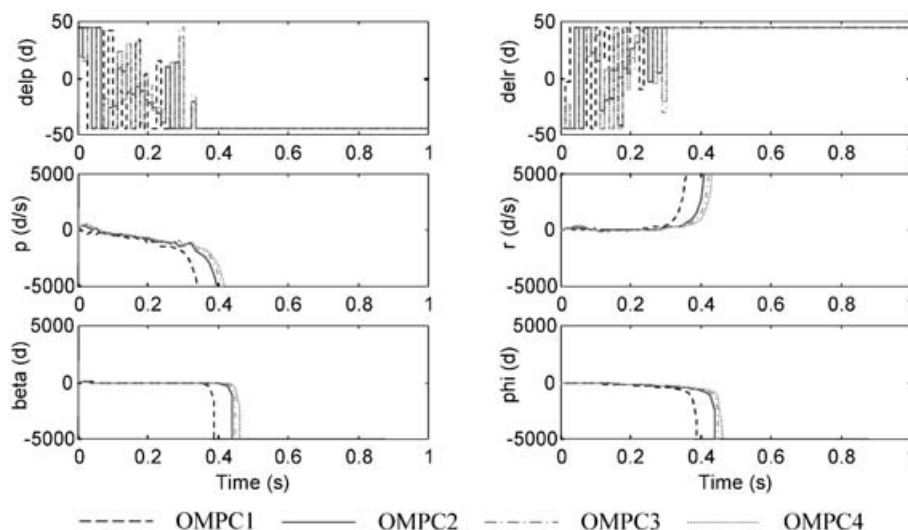


Fig. 9 Performance of all the OMPCs with online computational time limitation and K_{term} as back-up control gain ($x_0 = [0, 0, 5, -90]'$)

contributions have then been made. The first contribution is a new way to choose the terminal term in the performance index in order to enlarge the stability region, which is different from most existing MPC methods where the terminal term is chosen based on the terminal region. The second contribution is that a feasible initial control profile for the online optimization is suggested to guarantee stability, which can be determined offline. The third contribution is that a new control strategy is suggested when the online optimization is not completed in the sampling interval. The simulation carried out on the non-linear model of the missile shows that if there is no computational time limitation, the existing MPC algorithms and the new MPC scheme

can guarantee stability. However, in the presence of the computational time limitation, the existing MPC algorithms might fail to stabilize the missile, while the new MPC still works well.

REFERENCES

- 1 Clarke, D. W. *Advances in model-based predictive control*, 1994 (Oxford University Press).
- 2 Maciejowski, J. M. *Predictive control with constraints*, 2002 (Pearson Education under Prentice Hall).
- 3 Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Sokaert, P. O. M. Survey paper – constrained model predictive control: stability and optimality. *Automatica*, 2000, **36**, 789–814.

- 4 **Garcia, C. E., Prett, D. M., and Morar, M.** Model predictive control: theory and practice – a survey. *Automatica*, 1989, **25**(3), 335–348.
- 5 **Chen, H. and Allgower, F.** A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 1998, **34**(10), 335–348.
- 6 **Magni, L. and Sepulchre, L.** Stability margins of nonlinear receding-horizon control via inverse optimality. *System Control Lett.*, 1997, **32**, 241–245.
- 7 **Qin, S. J. and Badgwell, T. A.** An overview of nonlinear model predictive control applications. Proceedings of International Symposium on *Nonlinear model predictive control: assessment and future directions* (Eds F. Allgower and A. Zheng), Ascona, Switzerland, 1998, pp. 128–145.
- 8 **Bitmead, R. R., Gevers, M., and Wertz, V.** *Adaptive optimal control: the thinking man's GPC*, 1990 (Prentice-Hall, New York).
- 9 **Mayne, D. Q. and Michalska, H.** Receding horizon control of nonlinear systems. *IEEE Trans. Autom. Control*, 1990, **35**(7), 814–824.
- 10 **Chen, W. H., Balance, D. J., and O'Reilly, J.** Model predictive control of nonlinear systems: computational burden and stability. *IEE Proc.-Control Theory Applic.*, 2000, **147**(4), 387–394.
- 11 **Chen, W. H., Balance, D. J., and O'Reilly, J.** On attraction domain of model predictive control of nonlinear systems with input/state constraints. In *Proceedings of American control conference*, Chicago, Illinois, USA, 2001.
- 12 **Lee, Y. I. and Kouvaritakis, B.** Stabilizable regions of receding horizon predictive control with input constraints. *System Control Lett.*, 1999, **38**, 13–20.
- 13 **Lee, J. W., Kwon, W. H., and Choi, J.** On stability of constrained receding horizon control with finite terminal weighting matrix. *Automatica*, 1998, **34**(12), 1607–1612.
- 14 **Lee, J. W.** Exponential stability of constrained receding horizon control with terminal ellipsoid constraints. *IEEE Trans. Autom. Control*, 2000, **45**, 83–88.
- 15 **Gencell, H. and Nikolaon, M.** Robust stability analysis of constrained L_1 -norm model predictive control. *AIChE J.*, 1993, **39**(12), 1954–1965.
- 16 **Rawlings, J. B. and Muske, K. R.** The stability of constrained receding horizon control. *IEEE Trans. Autom. Control*, 1993, **38**(10), 1512–1516.
- 17 **De O. Kothare, S. L. and Morari, M.** Contractive model predictive control for constrained nonlinear systems. *IEEE Trans. Autom. Control*, 2000, **45**(6), 1053–1071.
- 18 **Chen, W. H.** Maximisation of stability/feasibility region of model predictive control for constrained linear systems. *IEE Proc.-Control Theory Applic.*, 2002, **149**(3), 243–246.
- 19 **Kouvaritakis, B., Cannon, M., and Rossiter, J. A.** Non-linear model-based predictive control. *Int. J. Control*, 1999, **72**(10), 919–928.
- 20 **Mhaskar, P., El-Farra, N. H., and Christofides, P. D.** Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. *System Control Lett.*, 2006, **55**, 650–659.
- 21 **Mhaskar, P.** Robust model predictive control design for fault-tolerant control of process systems. *Ind. Engng Chem. Res.*, 2006, **45**, 8565–8574.
- 22 **Schumacher, D.** *Tactical missile autopilot design using non-linear control*. PhD Thesis, Department of Aerospace Engineering, University of Michigan, 1994.
- 23 **Boyd, S., Ghaoul, L. E., Feron, E., and Balakrishnan, V.** *Linear matrix inequalities in system and control theory*, 1994 (The Society for Industry and Applied Mathematics, Philadelphia).

APPENDIX

Notation

A	system state matrix
B	input matrix
g	terminal weighting term
J	performance index
k	discrete-time index
K_{stab}	feasible control gain in stability region
K_{term}	terminal control gain
M	stability region
N	length of discrete-time moving horizon
p	roll rate (deg/s)
P	terminal weighting matrix
q	pitch rate (deg/s)
Q	state weighting matrix
r	yaw rate (deg/s)
R	input weighting matrix
t	continuous time (s)
T_H	length of continuous-time moving horizon
T_{sampling}	sampling time (s)
u	input vector
U	set of allowable input values
v	terminal region
x	system state vector
α	angle of attack (deg)
β	sideslip angle (deg)
θ	pitch angle (deg)
δ_p	effective roll control input (deg)
δ_r	effective yaw control input (deg)
ϕ	rank angle (deg)