Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Nonlinear model predictive missile control with a stabilising terminal constraint ⋆

V. Bachtiar * **T. Mühlpfordt** ** **W. H. Moase** *
T. Faulwasser *** **R. Findeisen** ** **C. Manzie** *

* *Department of Mechanical Engineering*
The University of Melbourne, Australia
{bachtiarv,moasew,manziec}@unimelb.edu.au
** *Otto-von-Guericke-Universität Magdeburg, Germany*
tillmann.muehlpfordt@st.ovgu.de, rolf.findeisen@ovgu.de
*** *École Polytechnique Fédérale de Lausanne, Switzerland*
timm.faulwasser@epfl.ch

Abstract: In this paper, an MPC scheme for a missile pitch axis autopilot is proposed. The scheme uses a nonlinear prediction model to give it an ability to push the controlled missile very close to its operating limits, and is stabilised through the use of an ellipsoidal terminal constraint. Tracking performance and computational load of the scheme are compared to that with a linear prediction model and other types of terminal constraint. Specifically, the choice of ellipsoidal, polytopic, or no terminal constraint is discussed. The terminally constrained nonlinear MPC scheme achieves comparable solution times to that with a linear prediction model, whilst being more aggressive to give a superior tracking performance.

Keywords: Predictive control, nonlinear control, missiles, stability

1. INTRODUCTION

Model Predictive Control (MPC) has a number of attractive properties for the control of a guided airframe. It explicitly handles constraints and has the ability to directly take into account plant nonlinearities. These are crucial as MPC would be able to push a missile to operate near its physical limits, at high angles of attack where aerodynamics are highly nonlinear (Gros et al., 2012). The other components of MPC such as objective cost and prediction horizon can be formulated to achieve the most desirable control behaviour. This makes MPC suited to address current challenges in missile autopilot design, which often revolve around the inability to account for nonlinearities, changes in missile behaviour during flight, and different missile configurations (Jackson, 2010).

Despite the aforementioned advantages, applications of MPC for guiding missiles are rare. This is primarily due to the high computational demand of MPC which can be problematic for applications in areas where plants posses fast, nonlinear dynamics, such as in the case of a missile (Hu and Chen, 2007). This makes the trade-off between theoretical advantages and implementability of MPC for missile control an important discussion.

To address the issue regarding computational burden, typical approaches include simplification of the problem formulation through model linearisation, relaxation of constraints, or bounding by uncertain linear models. Although reducing computational cost, these approximations can potentially have detrimental effects on the closed loop performance of the controller, such as a steady-state error due to model-plant mismatch or even instability.

The prediction model dictates the complexity of the pertaining optimisation problem in MPC. A nonlinear prediction model is associated with a higher computational load than a linear model due to the nonconvex nature of the resulting optimisation problem for most real-world applications. Early methods in solving MPC with a nonlinear prediction model utilised the direct multiple-shooting (Bock and Plitt, 1984) or collocation (Biegler, 1984). To achieve computational times needed in missile applications, a fast algorithm involving a sequential convex programming (Tran-Dinh and Diehl, 2010) to solve the nonlinear MPC scheme is followed in this paper. For benchmarking, the prediction model is simplified by a typical linearisation.

In this paper, a terminal constraint is used to guarantee stability of the MPC. In the early derivations of MPC stability, the terminal constraint was used to impose the terminal state to coincide at an invariant point, the origin (e.g. Mayne and Michalska, 1990). This is rather restrictive and in subsequent developments the notion of an invariant region (rather than an invariant point) was introduced (Michalska and Mayne, 1993). Chen et al. (1998; 2001) extended the theory by approximating the invariant region as an ellipsoid. An alternative approach in approximating such a region is to use a polytope (Cannon et al., 2003), which could better approximate the invariant region. These approaches derive closed-loop MPC stability with a linear differential inclusion (LDI) approximation of the nonlinear plant model, which simplifies calculation of the terminal region at the expense of being conservative. This paper compares the ellipsoid and polytopic terminal

⋆ This research was supported under the Australian's Research Council's Linkage Projects funding scheme (Project LP11020025).

Copyright © 2014 IFAC

region for missile autopilot. Further, these two approaches are compared to a computationally simpler approach with a relaxed constraint, i.e. the use of no terminal region.

The comparisons of the prediction models and terminal constraints are on the basis of how quickly the controller can drive the missile to track a given acceleration command and its computational load. This is to establish an understanding of the quality of nonlinear MPC with an ellipsoidal terminal constraint in terms of tracking performance and implementability of achieved solution times.

2. PLANT MODEL

2.1 Missile pitch-axis dynamics

The missile autopilot control in the pitch-axis as depicted in Fig. 1 is governed by the equations

$$\dot{\alpha} = q + \cos(\alpha) F_z(\alpha, \delta)/(mV) \tag{1a}$$
$$\dot{q} = L(\alpha, \delta)/I_y \tag{1b}$$
$$a = F_z(\alpha, \delta)/(mg) \tag{1c}$$

where the aerodynamic lift force F_z and pitching moment L are modelled by (Nichols et al., 1993)

$$F_z = 0.7M^2 P_0 S \left[C_{Z\alpha,1}(2 - M/3)\alpha + C_{Z\alpha,2}\alpha|\alpha| + C_{Z\alpha,3}\alpha^3 + C_{Z\delta}\delta\right] \tag{2a}$$
$$L = 0.7M^2 P_0 S d \left[C_{L\alpha,1}(8M/3 - 7)\alpha + C_{L\alpha,2}\alpha|\alpha| + C_{L\alpha,3}\alpha^3 + C_{L\delta}\delta\right]. \tag{2b}$$

α is the angle of attack and $q = \dot{\varphi}$ is the pitch rate of the missile. The output of the system a is the normal acceleration of the missile in multiples of gravitational acceleration g. Missile speed $V = MV_s$, where V_s is the speed of sound, is treated as constant at Mach number $M = 2.5$. The actuation of the fin deflection δ is modelled as a second order system:

$$\ddot{\delta} = -\omega_a^2\delta - 2\zeta\omega_a^2\dot{\delta} + \omega_a^2\delta_c. \tag{3}$$

Aerodynamic coefficients $C_{Z\alpha,1}, C_{Z\alpha,2}$ etc., other flight condition and missile frame parameters, along with constants related to (3) used are the same as given in Nichols et al. (1993). Note that the symbols $C_{Z\alpha,1}, C_{Z\alpha,2}$ etc. follow the standard nomenclature, and are different to that used by Nichols et al. (1993).

The dynamics of the missile can be put in a concise form:

$$\dot{\tilde{x}} = \tilde{f}(\tilde{x}, u) \tag{4a}$$
$$y = h(\tilde{x}) \tag{4b}$$

with state variables $\tilde{x} = [\alpha,\, q,\, \delta,\, \dot{\delta},\, \delta_c]^\mathsf{T} \in \mathbb{R}^{n_x}$, control variables $u = \dot{\delta_c} \in \mathbb{R}^{n_u}$, and output $y = a$.

The missile operation is subject to a number of constraints. A constraint on α is associated with the fact that aerodynamic coefficients used are only valid for a range of α. Fin deflection δ and its rate $\dot{\delta}$ are subject to mechanical limits of the actuator. Although there is no physical restriction that limits q, a constraint is imposed (made large for it to never be active) to make compact constraint polytopes

$$X = \{\tilde{x} : -\bar{x} \le \tilde{x} \le \bar{x}\},\ U = \{u : -\bar{u} \le u \le \bar{u}\} \tag{5}$$

where $\le$ and $\ge$ denote element-wise inequalities. This describes a region within missile physical limitations and accuracy of aerodynamic coefficients. The states and input limits are $\bar{x} = [\bar{\alpha},\, \bar{q},\, \bar{\delta},\, \bar{\dot{\delta}},\, \bar{\delta}]^\mathsf{T}$, and $\bar{u} = \bar{\dot{\delta}}$ respectively. Due to these constraints, the missile is only capable of tracking a maximum acceleration of $a_{\max}$.

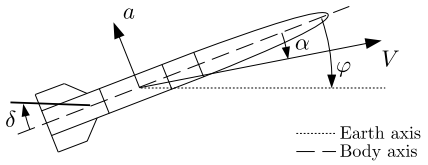

Fig. 1. Missile on the pitch-axis plane.

2.2 Tracking problem formulation

The autopilot control receives a commanded normal acceleration $a_\circ$ to track from the missile guidance law. Desired steady-state values of state variables $x_\circ$ which achieve $a_\circ$ are obtained from the solution of $0 = \tilde{f}(x_\circ, u_\circ)$ and $a_\circ = h(x_\circ)$. The system is injective therefore a commanded normal acceleration $a_\circ$ is associated with one unique steady state $x_\circ = [\alpha_\circ,\, q_\circ,\, \delta_\circ,\, 0,\, \delta_\circ]^\mathsf{T}$, $u_\circ = 0$.

If $a_\circ \neq 0$ then $x_\circ \neq 0$. The system can be formulated as an error system with an equilibrium at the origin $x = 0$:

$$x = \tilde{x} - x_\circ \tag{6a}$$
$$\dot{x} = \tilde{f}(\tilde{x}, u) = \tilde{f}(x + x_\circ, u) =: f(x, u) \tag{6b}$$
$$X = \{x : -\bar{x} \le x + x_\circ \le \bar{x}\},\ U = \{u : -\bar{u} \le u \le \bar{u}\} \tag{6c}$$

3. MODEL PREDICTIVE CONTROL

The missile in continuous time t is controlled at each sampling time t_i, for $i = 1, 2\ldots$ separated by a sampling period T_s, i.e. $t_{i+1} = t_i + T_s$. At each sampling instant, with current state $x(t_i)$, the MPC scheme considered in this paper is to solve the optimisation problem

$$\min_{\substack{x_k\ k=1\ldots N+1 \\ u_k\ k=1\ldots N}} \quad \sum_{k=1}^{N} \ell(x_k, u_k) + e(x_{N+1}) \tag{7a}$$
$$\text{s.t.} \quad x_1 = x(t_i) \tag{7b}$$
$$x_{k+1} = \Phi(x_k, u_k) \qquad \forall k = 1\ldots N \tag{7c}$$
$$x_k \in X,\ u_k \in U \qquad \forall k = 1\ldots N \tag{7d}$$
$$x_{N+1} \in \mathcal{X}_f \tag{7e}$$
$$\text{where} \quad \Phi(x_k, u_k) = x_k + \int_{t_{k+i-1}}^{t_{k+i}} f(x(\tau), u_k)\, d\tau. \tag{8}$$

The subscript k is used to discretise the continuous variables x and u into N discrete prediction variables to be solved by computational means. Here, N characterises the prediction horizon of the MPC scheme. The prediction model $\Phi(x_k, u_k)$ takes an initial state x_k and integrates the tracking error model (6b) over one sampling period with zero-order hold input u_k to obtain a predicted state x_{k+1}. The solution of (7a–e) are the optimal state x_k^* and control sequence u_k^*, $\forall k = 1\ldots N$, the first of which, u_1^*, is applied as feedback control to the plant.

3.1 Cost function

The cost function (7a) (with stage cost $\ell(\cdot)$ and terminal cost e$(\cdot)$) is a performance measure of the missile within the prediction horizon, indicating how far the states are to the desired values. A typical quadratic cost function:

$$\ell(x, u) = \|x\|_Q^2 + \|u\|_R^2\,, \quad e(x) = \|x\|_P^2 \tag{9}$$

is used. Q and R are positive definite stage cost weighting matrices for the state and control variables respectively, and P is the terminal cost weighting matrix. Minimising (9) drives the error x to zero and the system states $\tilde{x}$ to the steady state $x_\circ$ corresponding to the command $a_\circ$.

To only penalise errors in states relevant to the acceleration, the following formulation is used for Q and R:

$$Q = C^\mathsf{T} C \text{ where } C = \left.\frac{\partial h}{\partial \tilde{x}}\right|_{x_\circ, u_\circ}, \quad R = \epsilon_R \tag{10}$$

noting that $\|x\|^2_{C^\mathsf{T}C} \approx (a - a_\circ)^2$. This eliminates the need for tuning of Q and R, and simplifies the design process of the controller. Because the input u is constrained, there is no need to penalise its deviations from zero. R is set to ϵ_R where $0 < \epsilon_R \ll 1$ to ensure that (7a–e) is locally convex with respect to u, and avoid a chattering control behaviour. The value of P is chosen to ensure stability and will be found using different approaches in Section 3.3.

3.2 Prediction model

To solve (7a–e) with the nonlinear prediction model, equations (7b–c) can be formulated as a dynamic system representing the difference between the states x_k and an estimate of their values $\hat{x}_k$, $\forall k = 1 \dots N$, represented by a 1st-order Taylor expansion along the estimated trajectories:

$$x_{k+1} = \Phi(\hat{x}_k, \hat{u}_k) + \left.\frac{\partial \Phi}{\partial x}\right|_{\hat{x}_k, \hat{u}_k} (x_k - \hat{x}_k) + \left.\frac{\partial \Phi}{\partial u}\right|_{\hat{x}_k, \hat{u}_k} (u_k - \hat{u}_k). \tag{11}$$

Defining the Jacobians and a new vector F_k at each time step in the horizon as

$$A_k = \left.\frac{\partial \Phi}{\partial x}\right|_{\hat{x}_k, \hat{u}_k}, \quad B_k = \left.\frac{\partial \Phi}{\partial u}\right|_{\hat{x}_k, \hat{u}_k}, \tag{12a}$$

$$\text{and} \quad F_k = -\Phi(\hat{x}_k, \hat{u}_k) + A_k \hat{x}_k + B_k \hat{u}_k \tag{12b}$$

the differences between the true and estimated states and controls can now be captured in the following equality constraint, replacing (7b–c) in the optimisation problem:

$$\begin{bmatrix} B_1 & -I & & & \\ & A_2 & B_2 & -I & \\ & & & \ddots & \\ & & & A_N & B_N & -I \end{bmatrix} \begin{bmatrix} u_1 \\ x_2 \\ u_2 \\ x_3 \\ \vdots \\ u_N \\ x_{N+1} \end{bmatrix} = \begin{bmatrix} F_1 - A_1 x_1 \\ F_2 \\ \vdots \\ F_N \end{bmatrix}. \tag{13}$$

Following Tran-Dinh and Diehl (2010), the overall nonlinear MPC (NMPC) algorithm to solve (7a–e) is:

Algorithm 1. [NMPC] ($k = 1 \dots N+1$ for x and $k = 1 \dots N$ for u)

0 Obtain the current state $x(t_i)$. Initialise the estimate: $\hat{x}_k = x(t_i)$, $\hat{u}_k = 0$. Set $j = 0$.

1 Get an optimal (x_k, u_k) by solving (7a) s.t. (7b,7d–e,11). Denote this optimal solution $(\hat{x}^*_k, \hat{u}^*_k)$. $j = j + 1$.

2 Compute normalised mean squared errors: $\Delta x = \frac{1}{N+1}\sum_{k=1}^{N+1}((\hat{x}^*_k - \hat{x}_k)/\bar{x})^2$, $\Delta u = \frac{1}{N}\sum_{k=1}^{N}((\hat{u}^*_k - \hat{u}_k)/\bar{u})^2$.

3 If $\Delta x < \epsilon_x$ and $\Delta u < \epsilon_u$, or $j > j_{\max}$ for some thresholds $\epsilon_x \ll 1$, $\epsilon_u \ll 1$, and $j_{\max}$, then go to 4. Otherwise re-initialise: $\hat{x}_k = \hat{x}^*_k$, $\hat{u}_k = \hat{u}^*_k$ and go to 1.

4 Apply control $\hat{u}^*_1$. Move forward in time: $i = i + 1$. Note that at this step, $\hat{u}^*_1 \simeq u^*_1$ (as per Remark 1 below).

5 Obtain $x(t_i)$. To hot-start Steps 1–3, re-initialise the estimate: $\hat{x}_k = \hat{x}^*_k$, $\hat{u}_k = \hat{u}^*_k$. Then shift: $\hat{x}_{k+1} = \hat{x}_k$, $\hat{u}_{k+1} = \hat{u}_k$. Set $j = 0$. Go to 1.

In Step 1, (7a) s.t (7b,7d–e,11) is solved as a quadratic program (QP). Then Steps 2–3 effectively impose the condition $x_{k+1} = \Phi(x_k, u_k)$ for (11), ultimately satisfying (7c). In Step 3, the condition $j < j_{\max}$ can be used to ensure that the optimisation is solved under a certain time.

Remark 1. [Convergence of Algorithm 1 (Theorem 1 in Tran-Dinh and Diehl (2010))] $(\hat{x}_k, \hat{u}_k)$ in Steps 1–3 of Algorithm 1 converges to (x^*_k, u^*_k) if $(\hat{x}_k, \hat{u}_k)$ is initialised sufficiently close to (x^*_k, u^*_k).

Linear prediction model

Algorithm 1 yields an exact solution of (7a–e). As a benchmark, a computationally simpler alternative which yields an approximate solution of (7a–e) can be achieved by a time-invariant linearisation of (7c) about the reference $(\tilde{x} = x_\circ)$, i.e. origin of the error system $(x = 0)$, yielding a linear MPC scheme (LMPC) with

$$x_{k+1} = \left.\frac{\partial f}{\partial x}\right|_0 x_k + \left.\frac{\partial f}{\partial u}\right|_0 u_k = A x_k + B u_k \tag{14}$$

in place of (11). Now in (13), $A_k = A$, $B_k = B$, and $F_k = 0$, $\forall k = 1 \dots N$ in place of (12a–b). Consequently, (7a–e) can be solved as a QP in one step, simplifying the computation of u^* at each sampling instant, although introducing a greater model error in approximating (7c).

3.3 Terminal constraint

The terminal constraint $x_{N+1} \in \mathcal{X}_f$, along with the stage cost $\ell(\cdot)$, terminal cost $e(\cdot)$, and prediction horizon N are responsible for the stability of the MPC scheme. Specifically, to follow Mayne and Rawlings (2009) closely:

Theorem 1. [MPC Stability (Mayne and Rawlings, 2009)] For continuous $f(\cdot)$, $\ell(\cdot)$, $e(\cdot)$, and compact X and U, the closed-loop error system with (7a–e) is asymptotically stable about the origin if $\mathcal{X}_f$ is a subset of $\mathcal{X}_s$ and X, where $\mathcal{X}_s$ is defined as a control invariant region where once the system state x enters, there is a terminal controller $u = \kappa(x) \in U$ such that $\Phi(x, \kappa(x))$ stays within $\mathcal{X}_s$.

The methods of computing $\mathcal{X}_f$ followed in this paper are derived on a linear differential inclusion (LDI) approximation of the nonlinear missile model, which is more conservative and easier to analyse. The terminal controller considered is a linear controller $\kappa(x) = Kx$ for some gain matrix K. An associated control Lyapunov function (CLF) with $u = \kappa(x)$ can be determined once x enters $\mathcal{X}_s$.

No terminal constraint The conditions in Theorem 1 are sufficient but not necessary for closed-loop stability. Guarantee of stability without a terminal constraint has been shown by approximating a stable combination of P and N (Parisini and Zoppoli, 1995) for example. Grüne and Pannek (2011) use controllability assumptions which are in general very challenging to verify for a given nonlinear system. Thus, for a given N, trial-and-error is employed for P when no terminal constraint is used.

Ellipsoidal terminal constraint In this paper, a terminal constraint (7e) is used to guarantee stability through The-

orem 1. To keep (7e) simple for computational purposes, $\mathcal{X}_f$ will be restricted to either a polytope or ellipsoid.

With a finite prediction horizon, a stable MPC scheme under Theorem 1 might be infeasible in driving the states into the terminal region. In an effort to make the scheme as practicable as possible, a maximal volume terminal region is desired, such that the optimisation problem is feasible for the largest set of initial states. An ellipsoid

$$\mathcal{X}_f = \mathcal{E} = \{x : x^\mathsf{T} W x \leq \gamma\} \tag{15}$$

for some W and γ is a common choice, with a volume of $\mathrm{vol}(\mathcal{E}) = 4/3\pi \det(W/\gamma)$. This paper follows Chen et al.'s (2001) proposed stable NMPC with $\mathcal{X}_f = \mathcal{E}$, with a terminal cost weighting $P = W$. The associated CLF is

$$\|\Phi(x, Kx))\|_P^2 - \|x\|_P^2 \leq -\ell(x, Kx). \tag{16}$$

For volume maximisation of $\mathcal{E}$ under state and input constraints, and such that $\mathcal{X}_f \subseteq \mathcal{X}_s$, Chen et al.'s approach through linear matrix inequalities is used in this paper.

Polytopic terminal constraint As an alternative to an ellipsoid, a polytope

$$\mathcal{X}_f = \Pi = \{x : \|Vx\|_\infty \leq \lambda\} \tag{17}$$

for some V and λ can be chosen as the terminal region, with a volume of $\mathrm{vol}(\Pi) = (2\lambda)^{n_x} / \det(V)$. Cannon et al.'s (2003) work for stability under such a terminal region and its volume optimisation is followed in this paper, with $P = Q$ and an associated CLF of

$$\|V\Phi(x, Kx))\|_\infty - \|Vx\|_\infty \leq 0. \tag{18}$$

Cannon et al.'s first method (Theorem 3 and Remark 5) is used for the volume optimisation of Π under state and input constraints, and $\mathcal{X}_f \subseteq \mathcal{X}_s$. The motivation for $\mathcal{X}_f = \Pi$ is that it may lead to a volumetrically larger $\mathcal{X}_f$ than with $\mathcal{X}_f = \mathcal{E}$ depending on the exact shape of $\mathcal{X}_s$. This comes at a cost; the two methods for volume maximisation of Π by Cannon et al. are nonconvex. They are associated with multiple local minima, unlike the convex, globally-optimum method for $\mathcal{E}$.

Tracking of non-constant references

As the reference $a_\circ$ changes, the constraint polytope X in (6c) changes, and so does the region $\mathcal{X}_s$. As a result the terminal constraint must be recalculated. Recalculation of the terminal region $\mathcal{X}_f$ is computationally expensive and is not feasible to do within each sampling period. A simple approach is to precompute a set of $\mathcal{X}_f$ for a finite number of acceleration commands $a_L = \{a_1 \dots a_{n_l}\}$:

$$\mathcal{E}_l = \{x : x^\mathsf{T} W_l x \leq \gamma_l\} \quad l = 1 \dots n_l. \tag{19}$$

In this subsection, $\mathcal{E}$ is interchangeable with Π. n_l is the number of precomputed terminal regions. $l = 1, 2 \dots n_l$ corresponds to $a_\circ = a_1, a_2 \dots a_{n_l}$, in increasing order from $a_1 = 0$ to $a_{n_l} \approx a_{\max}$. Note that $\mathcal{X}_f$ for $a_\circ = a_{\max}$ contains only the origin. The terminal region can be recalculated with (the system is symmetric about the origin thus an extension for a negative acceleration command is trivial):

Assumption 1. [Strictly shrinking $\mathcal{X}_f$] $\mathcal{E}_{l_2} \subseteq \mathcal{E}_{l_1}\ \forall l_2 > l_1$.

Proposition 1. [Stability for set-point changes] Consider the following method for determining the terminal constraint for a given $a_\circ \notin a_L$: Find a_s and a_l, the smaller and larger immediate neighbours of $a_\circ$ in a_L respectively, with the associated terminal regions $\mathcal{E}_s$ and $\mathcal{E}_l$. Set the terminal constraint with $\mathcal{X}_f = \mathcal{E}_l$. For continuous $f(\cdot)$, $\ell(\cdot)$, $e(\cdot)$, compact X and U, and satisfaction of Assumption 1, the method (extended accordingly for $a_\circ < 0$) leads to asymptotic stability of the closed-loop error system in tracking set-point reference changes to $a_\circ \notin a_L$ for $-a_{\max} \leq a_\circ \leq a_{\max}$.

Remark 2. Assumption 1 is made on the basis that as $a_\circ$ increases, the constraint polytope X shifts away more from the origin. Thus, since $\mathcal{E}$ is symmetric about the origin, a smaller terminal region is required to keep $\mathcal{E}_l \subseteq X$. If Assumption 1 is valid, $\mathcal{E}_l \subseteq \mathcal{E}_s$ in Proposition 1. The method given in the proposition is then conservative in guaranteeing stability with a more restrictive constraint.

Remark 3. [Stability for time-varying references] Theorem 1 and Proposition 1 are valid for set-point references. For time-varying references i.e. trajectory tracking, the error system has been shown to converge to a bounded region under certain assumptions (Falugi and Mayne, 2012). For an asymptotically constant reference, Faulwasser and Findeisen (2011) presented an MPC approach where convergence to zero of the tracking error is guaranteed.

4. SIMULATION RESULTS

MatLab simulations were conducted to demonstrate the controller performance in tracking an acceleration command. Parameters used are typical in missile autopilot: $T_s = 10\,\mathrm{ms}$, $\bar{\alpha} = 20°$, $\bar{q} = 1000°\,\mathrm{s}^{-1}$, $\bar{\delta} = 45°$, and $\dot{\bar{\delta}} = 500°\,\mathrm{s}^{-1}$. For NMPC $\epsilon_x = \epsilon_u = 10^{-4}$ and $j_{\max} = 10$. Prediction horizon N and simulation time T are indicated for each individual result presented. Gurobi (Gurobi Optimization, 2013) was used to solve the QP pertaining to (7a–e). Acado (Houska et al., 2011) was used to calculate $\Phi(\cdot)$ using a 4^{th} order Gauss-Legendre method in 16 steps, and the Jacobians in (11). In what follows, "rise time" is the time taken for the controlled missile to reach and stay within $\pm 10\,\%$ of the desired acceleration command.

The presented NMPC scheme with an ellipsoidal terminal constraint is used to control a simulated missile model in tracking a given command. The missile model is initialised at the origin, and the controller is used to drive the model to track a $20, -5, 5\,g$ acceleration step for $0.5\,\mathrm{s}$ each. The response exhibits zero steady-state error and no overshoot, with a rise time of about $150\,\mathrm{ms}$ (Fig. 2). This confirms stability in tracking set-point changes as posed in Proposition 1. The controller is also tested to track a non-constant reference, applying the method in Proposition 1 in choosing a terminal constraint at every sampling time. The controller achieves a lag of about $100\,\mathrm{ms}$ in tracking a $15\,g$, $2\,\mathrm{Hz}$ sinusoidal reference (Fig. 3). This result shows that the controller exhibits a bounded, although non-zero error, as per Remark 3.

To understand the quality of the presented NMPC as a controller on the basis of tracking performance and computational time, it is compared with the more typical LMPC approach outlined in Section 3.2. It is then benchmarked against the use of polytopic and no terminal regions.

4.1 Linear vs. nonlinear prediction model

Comparing NMPC and LMPC, both schemes achieve zero steady-state error and no overshoot for a step change

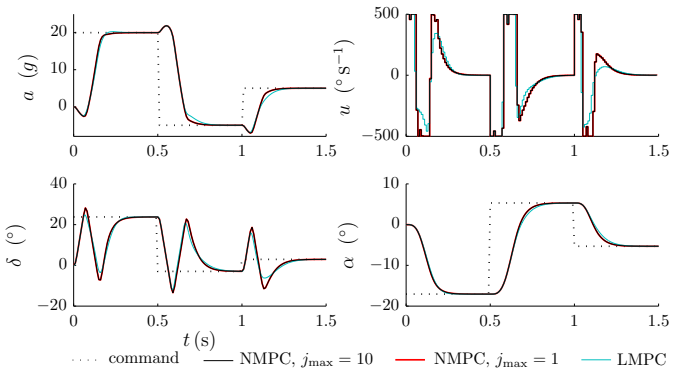

Fig. 2. Response to step changes in acceleration command. An ellipsoidal terminal constraint is imposed. $N = 25$.

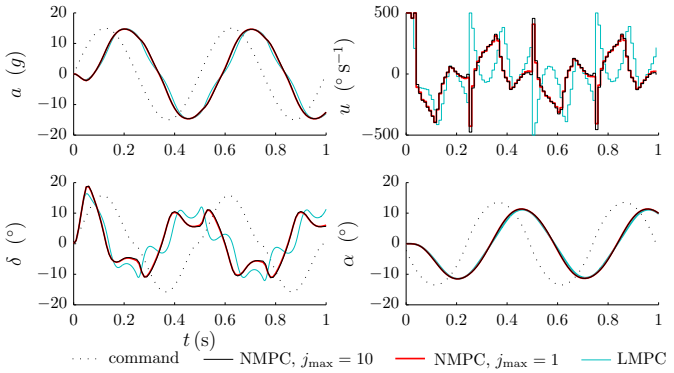

Fig. 3. Response to a sinusoidal acceleration command with $15\,g$ amplitude and 0.5 s period. An ellipsoidal terminal constraint is imposed. $N = 25$.

(Fig. 2), and a similar phase lag in tracking a sinusoidal reference (Fig. 3). In Fig. 2, NMPC exhibits an almost bang-bang response immediately after each step change, pushing the input u to its upper and lower limits for 0.1 s or so. This is less evident in LMPC, demonstrating that NMPC is more aggressive than LMPC, producing control commands with greater deviations from 0 (Table 1).

NMPC's aggressiveness leads to faster rise times by about 10-20 ms (Table 2), and an overall better performance as measured by the cumulated difference between the actual and desired acceleration (Table 3). This highlights the capability of NMPC to push the missile nearer to its limits to better track a given command than LMPC.

The superior performance of NMPC for tracking comes at a cost of computational load. NMPC takes around 10-15 ms longer to solve in each sampling time (Table 4) than LMPC. Nonetheless, computational times achieved by NMPC have the same order of magnitude as a typical missile sampling time of 10 ms. This is promising for practical implementation, where a dedicated processor and compiled code can be used to reduce computational time.

Furthermore, in NMPC simulations, the iteration limit is set at $j_{\max} = 10$ to let $\Delta x < \epsilon_x$ and $\Delta u < \epsilon_u$ in each sampling instant. For practical implementation, $j_{\max}$ can be set sufficiently small to guarantee that a solution is obtained within each sampling period. The first input $\hat{u}_1^*$ does not change significantly between iterations of Steps 1–3 in Algorithm 1. Changes mostly occur in the latter part of the sequence – which is not taken as feedback control – allowing $j_{\max} \sim 1$ without much performance degradation.

Table 1. $\int_0^T u^2 dt$ in tracking a step change to various $a_\circ$ from $0\,g$. $T = 0.5$ s, $N = 25$.

$a_\circ$	$\mathcal{X}_f = \mathcal{E}$ NMPC	$\mathcal{X}_f = \mathcal{E}$ LMPC	$\mathcal{X}_f = \Pi$ NMPC	$\mathcal{X}_f = \Pi$ LMPC	No $\mathcal{X}_f$ NMPC	No $\mathcal{X}_f$ LMPC
$5g$	1.00	0.86	1.00	0.86	0.99	0.85
$10g$	1.00	0.69	1.00	0.69	1.00	0.69
$15g$	1.00	0.64	1.00	0.64	1.00	0.63
$20g$	1.00	0.66	1.00	0.66	1.00	0.66

Values are normalised on results for NMPC with Ellipsoidal $\mathcal{X}_f$.

Table 2. Rise time in milliseconds. $N = 25$.

$a_\circ$	$\mathcal{X}_f = \mathcal{E}$ NMPC	$\mathcal{X}_f = \mathcal{E}$ LMPC	$\mathcal{X}_f = \Pi$ NMPC	$\mathcal{X}_f = \Pi$ LMPC	No $\mathcal{X}_f$ NMPC	No $\mathcal{X}_f$ LMPC
$5g$	150	164	150	165	150	165
$10g$	152	171	151	171	153	171
$15g$	152	171	152	171	153	171
$20g$	160	170	160	170	160	170

Table 3. $\int_0^T (a - a_\circ)^2 dt$ in tracking a step change to various $a_\circ$ from $0\,g$. $T = 0.5$ s, $N = 25$.

$a_\circ$	$\mathcal{X}_f = \mathcal{E}$ NMPC	$\mathcal{X}_f = \mathcal{E}$ LMPC	$\mathcal{X}_f = \Pi$ NMPC	$\mathcal{X}_f = \Pi$ LMPC	No $\mathcal{X}_f$ NMPC	No $\mathcal{X}_f$ LMPC
$5g$	1.00	1.00	1.00	1.00	1.00	1.00
$10g$	1.00	1.01	1.00	1.01	1.00	1.01
$15g$	1.00	1.01	1.00	1.01	1.00	1.01
$20g$	1.00	1.01	1.00	1.01	1.00	1.01

Values are normalised on results for NMPC with Ellipsoidal $\mathcal{X}_f$.

Table 4. Solution time in milliseconds, averaged for tracking constant and 2 Hz sinusoidal references with $5, 10, 15, 20\,g$ steps and amplitudes respectively. $T = 0.5$ s, $N = 25$.

$\mathcal{X}_f = \mathcal{E}$ NMPC	$\mathcal{X}_f = \mathcal{E}$ LMPC	$\mathcal{X}_f = \Pi$ NMPC	$\mathcal{X}_f = \Pi$ LMPC	No $\mathcal{X}_f$ NMPC	No $\mathcal{X}_f$ LMPC
25.7 (1.97)	10.1	18.8 (1.92)	6.7	17.5 (1.95)	5.4

Average iterations taken in NMPC are given in brackets. Simulations were run on a 3.4GHz, 8.0GB, 64-bit desktop.

As shown in Figs. 2 and 3, the performance with $j_{\max} = 1$ is barely distinguishable from that with $j_{\max} = 10$.

4.2 Ellipsoidal vs. polytopic vs. no terminal region

To further gauge the quality of the presented NMPC scheme, the use of an ellipsoidal terminal constraint ($\mathcal{X}_f = \mathcal{E}$) is compared with a polytopic ($\mathcal{X}_f = \Pi$) or no terminal constraint (no $\mathcal{X}_f$). The choice of terminal constraint has an effect on the quality of the controller, both in tracking performance and solution time. In terms of tracking performance, the use of no $\mathcal{X}_f$ is shown to be able to achieve a performance similar to that with $\mathcal{X}_f = \mathcal{E}$ (Fig. 4), although there is a potential performance degradation (e.g. a steady-state error) when the terminal cost is poorly chosen.

With regards to solution time, although an appropriately chosen terminal constraint guarantees closed-loop stability, it inevitably has an associated computational burden, adding 1-5 ms of solution time for both NMPC and LMPC (Table 4). The constraint $\mathcal{X}_f = \Pi$ is linear and less computationally burdensome to solve than $\mathcal{X}_f = \mathcal{E}$ which quadratically constrains the pertaining optimisation problem. The results shown in Table 4 are consistent with this notion; NMPC and LMPC with $\mathcal{X}_f = \mathcal{E}$ take about 4 ms longer in solution time than those with $\mathcal{X}_f = \Pi$.

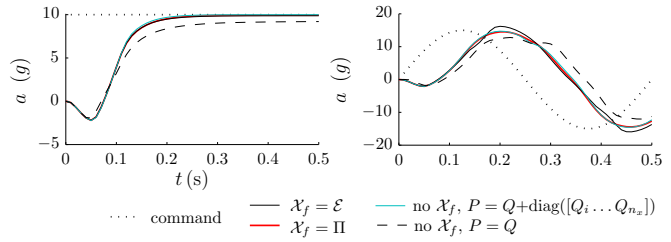

Fig. 4. NMPC response to a 10 g step and 15 g, 2 Hz acceleration commands. Q for the no terminal constraint case is as indicated in the legend (Q_i denotes the i^{th} diagonal of Q). $N = 12$.

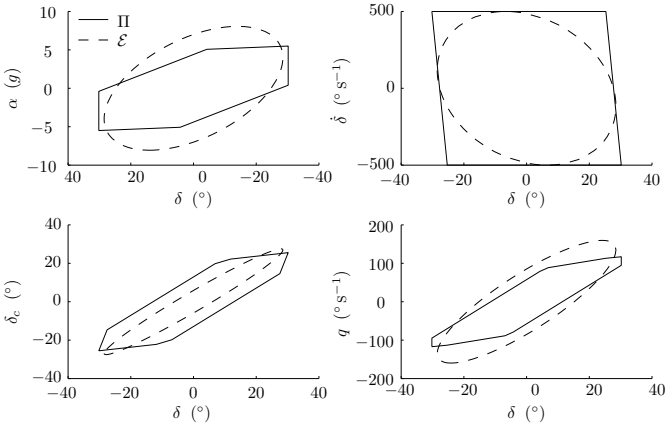

Fig. 5. 2-dimensional projections of controller terminal region with NMPC to track a 10 g acceleration.

In Fig. 5, a volume-optimal ellipsoidal terminal region $\mathcal{E}$ is compared a polytopic one Π for tracking a 10 g acceleration. The associated volumes are $\text{vol}(\mathcal{E}) = 6.8 \times 10^7$, and $\text{vol}(\Pi) = 8.8 \times 10^7$. The larger volume of the polytope is consistent with results presented in Cannon et al. (2003), allowing for shorter prediction horizons, and ultimately faster solution times. Further shortened prediction horizons can be used when no terminal constraint is imposed.

It is important to consider the effort required in calculating the terminal constraint for the controller. In optimising the volume of Π in Fig. 5, a number of initial conditions for the optimiser were used until a volume larger than $\mathcal{E}$ was obtained. As previously stated, global optimality is not guaranteed for the result since the optimisation of Π is nonconvex. There is more effort required to calculate Π compared to $\mathcal{E}$, which is more costly to design for the controller and is often undesirable in real-life practice.

5. CONCLUSIONS

An MPC scheme with a nonlinear prediction model and an ellipsoidal terminal constraint for missile autopilot is presented. It is shown that the use of a nonlinear prediction model pushes the missile more to its limits to produce better tracking performance than that with a linearly approximated prediction model, and the terminal constraint guarantees stability in constant reference tracking. Although the presented scheme is associated with a higher computational load, it maintains comparable solution times to schemes with a linearised approximation and/or polytopic or no terminal constraint in each sampling instance for a typical missile autopilot applications.

ACKNOWLEDGEMENTS

The authors would like to thank BAE Systems Australia for their support of this research.

REFERENCES

Biegler, L.T. (1984). Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Comput Chem Eng*, 8(3), 243–247.

Bock, H.G. and Plitt, K.J. (1984). A multiple shooting algorithm for direct solution of optimal control problems. In *IFAC World Congress P*, 243–247.

Cannon, M., Deshmukh, V., and Kouvaritakis, B. (2003). Nonlinear model predictive control with polytopic invariant sets. *Automatica*, 39(8), 1487–1494.

Chen, H. and Allgöwer, F. (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10), 1205–1217.

Chen, W.H., Ballance, D., and O'Reilly, J. (2001). Optimisation of attraction domains of nonlinear MPC via LMI methods. In *P Amer Contr Conf*, volume 4, 3067–3072.

Falugi, P. and Mayne, D.Q. (2012). Tracking performance of model predictive control. In *IEEE Decis Contr P*, 2631–2636.

Faulwasser, T. and Findeisen, R. (2011). A model predictive control approach to trajectory tracking problems via time-varying level sets of Lyapunov functions. In *IEEE Decis Contr and Euro Contr Conf P*, 3381–3386.

Gros, S., Quirynen, R., and Diehl, M. (2012). Aircraft control based on fast non-linear MPC & multiple-shooting. In *IEEE Decis Contr P*, 1142–1147.

Grüne, L. and Pannek, J. (2011). *Nonlinear Model Predictive Control*. Springer, London.

Gurobi Optimization (2013). Gurobi optimizer reference manual. URL `http://www.gurobi.com`.

Houska, B., Ferreau, H., and Diehl, M. (2011). ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optim Contr Appl Met*, 32(3), 298–312.

Hu, X.B. and Chen, W.H. (2007). Model predictive control for non-linear missiles. *P I Mech Eng I-J Sys*, 221(8), 1077–1089.

Jackson, P.B. (2010). Overview of missile flight control systems. *J Hopkins APL Tech D*, 29(1).

Mayne, D.Q. and Michalska, H. (1990). Receding horizon control of nonlinear systems. *IEEE T Automat Contr*, 35(7), 814–824.

Mayne, D.Q. and Rawlings, B. (2009). *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison.

Michalska, H. and Mayne, D.Q. (1993). Robust receding horizon control of constrained nonlinear systems. *IEEE T Automat Contr*, 38(11), 1623–1633.

Nichols, R.A., Reichert, R.T., and Rugh, W.J. (1993). Gain scheduling for H_∞ controllers: a flight control example. *IEEE T Contr Syst T*, 1(2), 69–79.

Parisini, T. and Zoppoli, R. (1995). A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica*, 31(10), 1443–1451.

Tran-Dinh, Q. and Diehl, M. (2010). Local convergence of sequential convex programming for nonconvex optimization. In *Recent Advances in Optimization and its Applications in Engineering*, 93–102. Springer.