A New Real-Time Method for Nonlinear Model Predictive Control

Darryl DeHaan and Martin Guay

Department of Chemical Engineering, Queen's University, Kingston, Ontario, Canada, K7L 3N6
{dehaan,guaym}@chee.queensu.ca

Summary. A formulation of continuous-time nonlinear MPC is proposed in which input trajectories are described by general time-varying parameterizations. The approach entails a limiting case of suboptimal single-shooting, in which the dynamics of the associated NLP are allowed to evolve within the same timescale as the process dynamics, resulting in a unique type of continuous-time dynamic state feedback which is proven to preserve stability and feasibility.

1 Introduction

In this note we study the continuous-time evolution of nonlinear model predictive control in cases where the optimization must necessarily evolve in the same timescale as the process dynamics. This is particularly relevant for applications involving "fast" dynamics such as those found in aerospace, automotive, or robotics applications in which the computational lag associated with iterative optimization algorithms significantly limits the application of predictive control approaches.

In an attempt to reduce computational lag, interest has been focussed on the use of suboptimal solutions arrived at by early termination of the nonlinear program being solved online. Real-time computational algorithms such as [1] push this concept to evaluating only a single NLP iteration per discrete sampling interval. A similar concept of incremental improvement underlies realtime works such as [2, 3], where the input parameters are treated as evolving according to continuous-time differential equations driven by descent-based vector fields. In particular, [3] illustrates how this approach is effectively a type of adaptive feedback.

In this work, we present a form of real-time MPC which, in the spirit of [2] and [3], treats the evolving optimization as an adaptive control action. However, our results are more general in that we do not require global asymptotic stability of the unforced dynamics (unlike [3]), and our approach preserves stability without requiring "sufficiently many" parameters in the description of the input (unlike [2]). One important aspect of our approach is that the open-loop parameterization of the input is defined relative to a time partition that can potentially be

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 537–550, 2007.
springerlink.com

adapted online to make optimal use of the finite number of parameters used to describe the input. While the manner in which the input is parameterized has similarities to sampled-data approaches such as [4], a key difference is that our approach involves continuous measurement and control implementation throughout the intervals of the time partition, and as a result there are no intervals of open-loop behaviour introduced into the feedback path.

This paper is organized as follows. The basic problem is described in Section 2, with finite input parameterizations and local stabilizing controllers discussed in Sections 3 and 4, respectively. Section 5 discusses the realtime design approach, with an example in Section 6. Proofs are in the Appendix. In the following, we will use the notation $\mathring{\mathbb{S}}$ to denote the open interior of a closed set $\mathbb{S}$, and $\partial\mathbb{S}$ for the boundary $\mathbb{S} \setminus \mathring{\mathbb{S}}$. Furthermore, we denote by $\|z\|_{\mathbb{S}}$ the orthogonal distance of a point z to the set $\mathbb{S}$; i.e. $\|z\|_{\mathbb{S}} = \inf_{s\in\mathbb{S}} \|z - s\|$. A continuous function $\gamma : [0, \infty) \to \mathbb{R}_{\geq 0}$ is defined as class $\mathcal{K}$ if it is strictly increasing from $\gamma(0) = 0$, and class $\mathcal{K}_\infty$ if it is furthermore radially unbounded. Finally, a function will be described as C^{m+} if it is C^m, with all derivatives of order m yielding locally Lipschitz functions.

2 Problem Setup

Our control objective is the regulation of the dynamics

$$\dot{x} = f(x, u) \tag{1}$$

to the compact target set $\Sigma_{\mathbb{X}} \subset \mathbb{R}^n$, which is assumed to be weakly invariant for controls in some compact set $u \in \Sigma_{\mathbb{U}}(x) \subset \mathbb{R}^m$; i.e. there exists a static feedback rendering the set $\Sigma \triangleq \{(x, u) \in \Sigma_{\mathbb{X}} \times \mathbb{R}^m | u \in \Sigma_{\mathbb{U}}(x)\}$ forward invariant. Set stabilization allows for more general control problems than simple stabilization to a point, and in particular encompasses the notion of "practical-stabilization". We are interested in continuous-time model predictive control problems of the form

$$\min_{u(\cdot)} \left\{ \int_t^{t+T} L(x^p, u)\, d\tau + W(x^p(t+T)) \right\} \tag{2a}$$

$$s.t.\ \ \dot{x}^p = f(x^p, u), \qquad x^p(t) = x \tag{2b}$$

$$(x^p, u) \in \mathbb{X} \times \mathbb{U}, \quad \forall \tau \in [t,\, t+T] \tag{2c}$$

$$x^p(t+T) \in \mathcal{X}_f. \tag{2d}$$

Since the motivating problem of interest is assumed to involve an infinite horizon, the horizon length in (2a) is interpreted as designer-specifiable. The sets $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{U} \subset \mathbb{R}^m$ represent pointwise-in-time constraints, and are assumed to be compact, connected, of non-zero measure (i.e. $\mathring{\mathbb{X}}, \mathring{\mathbb{U}} \neq \emptyset$), and to satisfy the containment $\Sigma \subset \mathring{\mathbb{X}} \times \mathring{\mathbb{U}}$. The compact, connected terminal set $\mathcal{X}_f$ is typically designer-specified, and is assumed to strictly satisfy $\Sigma_{\mathbb{X}} \subset \mathcal{X}_f \subset \mathring{\mathbb{X}}$. The mapping

$L : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_{\geq 0}$ is assumed to satisfy $\gamma_L(\|x, u\|_\Sigma) \leq L(x, u) \leq \gamma_U(\|x, u\|_\Sigma)$ for some $\gamma_L, \gamma_H \in \mathcal{K}_\infty$, although this could be relaxed to an appropriate detectability condition. The mapping $W : \mathcal{X}_f \to \mathbb{R}_{\geq 0}$ is assumed to be positive semi-definite, and identically zero on the set $\Sigma_\mathbb{X} \subset \mathcal{X}_f$. For the purposes of this paper, the functions $L(\cdot, \cdot)$, $W(\cdot)$ and $f(\cdot, \cdot)$ are all assumed to be C^{1+} on their respective domains of definition, although this could be relaxed to locally Lipschitz with relative ease.

3 Finite-Dimensional Input Parameterizations

Increasing horizon length has definite benefits in terms of optimality and stability of the closed loop process. However, while a longer horizon obviously increases the computation time for model predictions, of significantly greater computational concern are the additional degrees of freedom introduced into the minimization in (2a). This implies that instead of enforcing a constant horizon length, it may be more beneficial to instead maintain a constant number of input parameters whose distribution across the prediction interval can be varied according to how "active" or "tame" the dynamics may be in different regions.

Towards this end, it is assumed that the prediction horizon is partitioned into N intervals of the form $[t^\theta_{i-1},\ t^\theta_i]$, $i = 1 \ldots N$, with $t \in [t^\theta_0,\ t^\theta_1]$. The input trajectory $u : [t^\theta_0,\ t^\theta_N] \to \mathbb{R}^m$ is then defined in the following piecewise manner

$$u(\tau) = u_\phi(\tau, t^\theta, \theta, \phi) \triangleq \begin{cases} \phi(\tau - t^\theta_0,\ \theta_1) & \tau \in [t^\theta_0,\ t^\theta_1] \\ \phi(\tau - t^\theta_{i-1},\ \theta_i) & \tau \in (t^\theta_{i-1},\ t^\theta_i],\ i \in \{2 \ldots N\} \end{cases} \tag{3}$$

with individual parameter vectors $\theta_i \in \Theta \subset \mathbb{R}^{n_\theta}$, $n_\theta \geq m$, for each interval, and $\theta = \{\theta_i \,|\, i \in \{1, \ldots N\}\} \in \Theta^N$. The function $\phi : \mathbb{R}_{\geq 0} \times \Theta \to \mathbb{R}^m$ may consist of any smoothly parameterized (vector-valued) basis in time, including such choices as constants, polynomials, exponentials, radial bases, etc. In the remainder, a (*control-* or *input-*) *parameterization* shall refer to a triple $\mathcal{P} \triangleq (\phi,\ \mathbb{R}^{N+1},\ \Theta^N)$ with specified N, although this definition may be abused at times to refer to the family of input trajectories spanned by this triple (i.e. the set-valued range of $\phi(\mathbb{R}^{N+1}, \Theta^N)$).

Assumption 1. *The C^{1+} mapping $\phi : \mathbb{R}_{\geq 0} \times \Theta \to \mathbb{R}^m$ and the set Θ are such that 1) Θ is compact and convex, and 2) the image of Θ under ϕ satisfies $\mathbb{U} \subseteq \phi(0, \Theta)$.*

Let $(t_0,\ x_0) \in \mathbb{R} \times \mathring{\mathbb{X}}$ represent an arbitrary initial condition for system (1), and let $(t^\theta,\ \theta)$ be an arbitrary choice of parameters corresponding to some parameterization $\mathcal{P}$. We denote the resulting solution to the prediction model in (2b), defined on some maximal subinterval of $[t_0,\ t^\theta_N]$, by $x^p(\cdot, t_0, x_0, t^\theta, \theta, \phi)$. At times we will condense this notation, and that of (3), to $x^p(\tau)$, $u_\phi(\tau)$.

A particular choice of control parameters (t^θ, θ) corresponding to some parameterization $\mathcal{P}$ will be called *feasible* with respect to $(t_0,\ x_0)$ if, for every

$\tau \in [t_0, t_N^\theta]$, the solution $x^p(\tau, t_0, x_0, t^\theta, \theta, \phi)$ exists and satisfies $x^p(\tau) \in \mathring{\mathbb{X}}$, $u_\phi(\tau) \in \mathring{\mathbb{U}}$, and $x^p(t_N^\theta) \in \mathring{\mathcal{X}}_f$. We let $\Phi(t_0, x_0, \mathcal{P}) \subseteq \mathbb{R}^{N+1} \times \Theta^N$ denote the set of all such feasible parameter values for a given (t_0, x_0) and parameterization $\mathcal{P}$. This leads to the following result, which is a straightforward extension of a similar result in [5].

Lemma 1. *Let $\mathbb{X}^0 \subseteq \mathring{\mathbb{X}}$ denote the set of initial states x_0 for which there exists open-loop pairs $(x(\cdot), u(\cdot))$ solving (1), defined on some interval $t \in [t_0, t_f]$ (on which $u(\cdot)$ has a finite number of discontinuities), and satisfying the constraints $x(t_f) \in \mathring{\mathcal{X}}_f$, and $(x, u)(t) \in \mathring{\mathbb{X}} \times \mathring{\mathbb{U}}$, $\forall t \in [t_0, t_f]$. Then, for every $(t_0, x_0) \in \mathbb{R} \times \mathbb{X}^0$ and every (ϕ, Θ) satisfying Assumption 1, there exists $N^* \equiv N^*(x_0, \phi, \Theta)$ such that $\Phi(t_0, x_0, \mathcal{P})$ has positive Lebesgue measure in $\mathbb{R}^{N+1} \times \Theta^N$ for all $N \geq N^*$.*

4 Requirements for a Local Stabilizing Control Law

Sufficient conditions for stability of NMPC presented in [6] require that $\mathcal{X}_f$ be a control-invariant set, and that the function $W(\cdot)$ be a control Lyapunov function on the domain $\mathcal{X}_f$. The following assumption represents a slight strengthening of those conditions - presented in integral rather than differential form - as applicable to the input parameterizations from the preceding section. In particular, a pair of feedbacks satisfying the assumption are required to be explicitly known, and the strict decrease in (4) is added to enable the use of interior-point methods for constraint handling.

Assumption 2. *The penalty $W : \mathcal{X}_f \to \mathbb{R}_{\geq 0}$, the sets $\mathcal{X}_f$ and Σ, the mapping ϕ, and a pair of known feedbacks $\delta : \mathcal{X}_f \to \mathbb{R}_{>0}$ and $\kappa : \mathcal{X}_f \to \Theta$ are all chosen s.t.*

1. *$\Sigma_{\mathbb{X}} \subset \mathring{\mathcal{X}}_f$, $\mathcal{X}_f \subset \mathring{\mathbb{X}}$, both $\mathcal{X}_f$ and Σ compact.*
2. *there exists a compact set $\mathbb{U}^0 \subset \mathring{\mathbb{U}}$ s.t. $\forall x \in \mathcal{X}_f$, $\sup_{\tau \in [0,\ \delta(x)]} \|\phi(\tau, \kappa(x))\|_{\mathbb{U}^0} = 0$.*
3. *Σ and $\mathcal{X}_f$ are both rendered positive invariant in the following sense:*
   - *there exists a constant $\varepsilon_\delta > 0$ such that $\delta(x_0) \geq \varepsilon_\delta$ for all $x_0 \in \mathcal{X}_f$.*
   - *for every $x_0 \in \Sigma_{\mathbb{X}}$, the (open-loop) solution to $\dot{x}_\kappa = f(x_\kappa, \phi(\tau_\kappa, \kappa(x_0)))$, $x_\kappa(0) = x_0$ exists and satisfies $(x_\kappa(\tau_\kappa), \phi(\tau_\kappa, \kappa(x_0))) \in \Sigma$ for $\tau_\kappa \in [0,\ \delta(x_0)]$.*
   - *$\exists \varepsilon^* > 0$ and a family of sets $\mathcal{X}_f^\varepsilon = \{\, x \in \mathcal{X}_f \ :\ \inf_{s \in \partial \mathcal{X}_f} \|s - x\| \geq \varepsilon \,\}$, $\varepsilon \in [0,\ \varepsilon^*]$, such that $x_0 \in \mathcal{X}_f^\varepsilon \implies x_\kappa(t) \in \mathcal{X}_f^\varepsilon, \forall t \in [0,\ \delta(x_0)]$, $\forall \varepsilon \in [0,\ \varepsilon^*]$*
4. *there exists $\gamma \in \mathcal{K}$ such that for all $x_0 \in \mathcal{X}_f$, (with $x_f \triangleq x_\kappa(\delta(x_0))$),*

$$W(x_f) - W(x_0) + \int_0^{\delta(x_0)} L(x_\kappa, \phi(\tau, \kappa(x_0)))\, d\tau \leq - \int_0^{\delta(x_0)} \gamma(\|x_\kappa\|_{\Sigma_{\mathbb{X}}}) d\tau \qquad (4)$$

4.1 Design Considerations

For the purposes of this work, any locally stabilizing pair (κ, δ) satisfying Assumption 2 can be used. For the case where ϕ is a piecewise-constant parameterization, several different approaches exist in the literature for the design of such feedbacks (see [7, 8] and references therein). Below we present one possible extension of these approaches for finding κ and δ in the case of more general parameterizations.

1. Assume that a known feedback $u = k_f(x)$ and associated CLF $W(x)$ satisfy
$$\frac{\partial W}{\partial x} f(x, k_f(x)) + L(x, k_f(x)) \leq -\gamma_k(\|x\|_{\Sigma_{\mathbb{X}}}) \qquad \forall x \in \mathcal{X}_f \tag{5}$$
for some $\gamma_k \in \mathcal{K}$, with $\mathring{\Sigma} \neq \emptyset$ (if necessary, take Σ as a small neighbourhood of the true target). Let Σ^ε denote a family of nested inner approximations of Σ. For some $\varepsilon^* > 0$, the sets $\mathcal{X}_f^\varepsilon$ and Σ^ε are assumed forward-invariant with respect to $\dot{x} = f(x, k_f(x))$, and $k_f(x) \in \mathbb{U}^0$ for all $x \in \mathcal{X}_f$, $\varepsilon \in [0, \varepsilon^*]$.
2. Without loss of generality, assume a number $r \in \{0, 1, \ldots, \mathrm{floor}(n_\theta/m) - 1\}$ is known such that $k_f \in C^{r+}$, and
$$\mathrm{span}_{\theta_i \in \Theta} \begin{bmatrix} \phi(0, \theta_i) \\ \vdots \\ \frac{\partial^r \phi}{\partial \tau^r}(0, \theta_i) \end{bmatrix} = \mathbb{U} \oplus \mathbb{R}^{rm}. \tag{6}$$
Select any C^{1+} mapping $\kappa(x) : \mathcal{X}_f \to \{ \varpi \in \Theta \ : \ \varpi \text{ satisfies (7) for } x \}$, whose range is nonempty by (6) and Assumption 1. (i.e. invert the function $\phi(0, \cdot)$)
$$\begin{bmatrix} k_f(x) \\ \frac{\partial k_f}{\partial x} f(x, k_f(x)) \\ \vdots \\ L_f^r k_f \end{bmatrix} = \begin{bmatrix} \phi(0, \varpi) \\ \frac{\partial \phi}{\partial \tau}(0, \varpi) \\ \vdots \\ \frac{\partial^r \phi}{\partial \tau^r}(0, \varpi) \end{bmatrix} \tag{7}$$
3. Specify $\gamma = \frac{1}{2}\gamma_k$, and simulate the dynamics forward from $x_\kappa(0) = x$ under control $u = \phi(\tau_\kappa, \omega)$ until one of the conditions in Assumption 2 fails, at a time $\tau_\kappa = \delta^*$. Set $\delta(x) = c_\delta \delta^*$, for any $c_\delta \in (0, 1)$.

This approach effectively assigns $\kappa(x)$ by fitting a series approximation of order r to the input trajectory generated by $u = k_f(x)$. By the invariance (and compactness) of the inner approximations $\mathcal{X}_f^\varepsilon$ and Σ^ε for some $\varepsilon^* > 0$, a lower bound $\varepsilon_\delta \equiv \varepsilon_\delta(\varepsilon^*) > 0$ exists such that $\delta(x) \geq c_\delta \varepsilon_\delta$, $\forall x \in \mathcal{X}_f$. In contrast, a similar problem of initializing input trajectories is solved in [4] by using forward simulation of the dynamics $\dot{x} = f(x, k_f(x))$ to generate $u(t)$. Within our framework, however, it could be difficult to ensure that these generated trajectories lie within the span of $\mathcal{P}$.

5 Real-Time Design Approach

5.1 Constraint Handling

While both active-set and interior-point approaches have been successfully used to handle constraints in NMPC problems, one limitation of using active sets within the context of our realtime framework is that constraint violation can only be tested at discrete, pre-defined points in time along the prediction interval. In contrast, interior point approaches such as [9] preserve constraint feasibility all points along the prediction trajectory, which is advantageous when the time support t^θ is nonuniform and potentially involves large intervals. A second benefit of using interior-point methods is that nominal robustness in the presence of state constraints is guaranteed automatically, whereas it is shown in [10] that active set approaches must be modified to use interior approximations of the constraint in order to guarantee nominal robustness. To this end, the constraints are incorporated defining

$$L^a(x,u) = L(x,u) + \mu\left(B_x(x) + B_u(u)\right), \qquad W^a(x_f) = W(x_f) + \mu\, B_{x_f}(x_f) \quad (8)$$

where $\mu > 0$ is a design constant, and B_x, B_u, B_{x_f} are barrier functions on the respective domains $\mathbb{X}$, $\mathbb{U}$ and $\mathcal{X}_f$. For the purposes of this work, it is assumed that the barrier functions are selected a-priori to satisfy the following minimum criteria, where the pair $(s, \mathbb{S})$ is understood to represent $\{(x,\mathbb{X}),\ (u,\mathbb{U}),\ (x_f, \mathcal{X}_f)\}$.[-1mm]

Criterion 1. *The individual barrier functions each satisfy*

1. $B_s : \mathbb{S} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$, *and* B_s *is* C^{1+} *on the open set* $\mathring{\mathbb{S}}$.
2. $s \to \partial\mathbb{S}$ *(from within) implies* $B_s(s) \to \infty$.
3. $B_s \equiv 0$ *on* $s \in \Sigma_{\mathbb{S}}$, *and* $B_s \geq 0$ *on* $s \in \mathbb{S} \setminus \Sigma_{\mathbb{S}}$.

The assumed differentiability of B_s is for convenience, and could be relaxed to locally Lipschitz. We note that additional properties such as convexity of $\mathbb{S}$ and B_s or self-concordance of B_s (see [9, 11]) are not technically required, although in practice they are highly advantageous. The third criterion implies that the B_s is "centered" around the target set Σ. For basic regulation problems ($\Sigma = \{(0,0)\}$) with convex constraints a self concordance-preserving recentering technique is given in [9], which could be extended to more general Σ, but likely at the expense of self-concordance. For nonconvex constraints, a barrier function satisfying Criterion 1 must be designed directly. In addition to the above criteria, it must be ensured that substituting (8) does not compromise the stability condition (4). Thus we require:

Criterion 2. *For a given local stabilizer satisfying Assumption 2, the barrier functions* B_x, B_u, B_{x_f} *and multiplier* μ *are chosen to satisfy, for all* $x \in \mathcal{X}_f$,

$$\sup_{(\tau,\, x_0)\in\mathcal{I}(x)} \left\{\nabla B_{x_f}(x)^T f(x, \phi(\tau, \kappa(x_0))) + B_x(x) + B_u(\phi(\tau,\kappa(x_0)))\right\} \leq \frac{1}{\mu}\gamma(\|x\|_{\Sigma_{\mathbb{X}}}) \quad (9)$$

$$\mathcal{I}(x) \triangleq \{\ (\tau,\ x_0) \in [0,\ \delta(x_0)] \times \mathcal{X}_f\ :\ \dot{x}_\kappa = f(x_\kappa, \phi(t, x_0)),\ x_\kappa(0) = x_0 \text{ and } x_\kappa(\tau) = x\ \}$$

In general, Criterion 2 can be readily satisfied if 1) level curves of $B_{\mathcal{X}_f}$ are invariant; i.e. they align with level curves of W, 2) μ is chosen sufficiently small, and 3) the growth rates of B_x and $\phi \circ B_u$ are less than that of γ in an open neighbourhood of Σ. When using the design approach for κ and δ in Section 4.1, one can treat Criterion 2 as a constraint on the interval length $\delta(x)$ by designing the barriers to satisfy

$$\nabla B_{x_f}(x)^T f(x, k_f(x)) + B_x(x) + B_u(k_f(x)) < \frac{1}{\mu}\gamma(\|x\|_{\Sigma_{\mathbb{X}}}) \ . \tag{10}$$

5.2 Description of Closed-Loop Behaviour

Before detailing our MPC controller, it will be useful to denote $z \triangleq [x^T,\ {t^\theta}^T,\ \theta^T]^T$ as the vector of closed-loop states. The cost function is then defined as

$$\begin{aligned} J(t,z) \ &= \textstyle\int_t^{t_N^\theta} L^a(x^p(\tau), u_\phi(\tau))\, d\tau + W^a(x^p(t_N^\theta)) && \text{(11a)}\\ s.t. \quad &\tfrac{dx^p}{d\tau} = f(x^p, u_\phi(\tau, z, \phi)), \qquad x^p|_{\tau=t} = x \ . && \text{(11b)} \end{aligned}$$

Step 1: Initialization of t^θ and θ
Let $(t_0,\, x_0) \in \mathbb{R} \times \mathbb{X}^0$ denote an arbitrary feasible initial condition for (1). The first step is to initialize the control parameters to any value in the feasible set $\Phi(t_0, x_0, \mathcal{P})$, which is guaranteed by Lemma 1 to be tractable. In the simple case where $\mathbb{X}^0 \subseteq \mathcal{X}_f$, then feasible parameter values can be obtained from forward simulation of the dynamics under the feedbacks $\kappa(\cdot)$ and $\delta(\cdot)$; otherwise a dual programming program could be solved to identify feasible initial parameter values.

Step 2: Continuous flow under dynamic feedback
At any instant $t \in [t_0,\, t_1^\theta]$ we assume that the model prediction $x^p(\tau, t, z, \phi)$ is 'instantaneously' available. This prediction information is used to update the control states in real time, so the closed-loop dynamics evolve under dynamic feedback as:

$$\dot z = \begin{bmatrix} \dot x \\ \dot t^\theta \\ \dot\theta \end{bmatrix} = \begin{bmatrix} f(x,\, \phi(t - t_0^\theta, \theta_1)) \\ \operatorname{Proj}\left\{-k_t\, \alpha(t,z)\, \Gamma_t\, \nabla_{t^\theta} J^T,\ \Xi(t)\right\} \\ \operatorname{Proj}\left\{-k_\theta \Gamma_\theta\, \nabla_\theta J^T,\ \Theta^N\right\} \end{bmatrix} \qquad \text{while } t \le t_1^\theta \tag{12a}$$

$$\alpha(t,z) \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & \operatorname{sat}\left(\frac{t_1^\theta - t}{\epsilon},\ [0,\,1]\right) & 0 \\ 0 & 0 & I \end{bmatrix}_{n_\theta \times n_\theta} \tag{12b}$$

$$\Xi(t) = \left\{ t^\theta \in \mathbb{R}^{N+1} \;\middle|\; (\pi_i(t, t^\theta) \ge 0,\ i = 1, \dots N) \text{ and } \left(\textstyle\sum_{i=1,\dots N} \pi_i \le T\right) \right\} \tag{12c}$$

where $\epsilon > 0$ is a small constant, and π represents the coordinate transformation

$$\pi_i(t, t^\theta) = \left\{ \begin{matrix} t - t_0^\theta & i = 0 \\ t_i^\theta - t_{i-1}^\theta & i = 1, \dots N \end{matrix} \right\} \in \mathbb{R}_{\ge 0}^{N+1} \tag{13}$$

The function α serves to restrict the adaptation of t_1^θ such that the intersection $t = t_1^\theta$ is transversal, resulting in deterministic closed-loop behaviour. Although (12a) appears nonautonomous, all time-dependence disappears under transformation (13).

The gradient terms $\nabla_\theta J$ and $\nabla_{t^\theta} J$ in (12a) represent sensitivities of (11a), for which differential sensitivity expressions must be solved. Fortunately, several efficient algorithms (for example [12]) exist for simultaneous solution of ODE's with their parametric sensitivity equations, which can additionally be efficiently decomposed by the intervals of t^θ. The matrices $\Gamma_{t^\theta} > 0$ and $\Gamma_\theta > 0$ define the type of descent-based optimization used. While constant matrices generating (scaled-) steepest-descent trajectories are the simplest choice, higher order definitions such as Gauss-Newton or full order Newton (appropriately convexified) could be used.

The operator in (12a) of the form $\dot{s} = \mathrm{Proj}(\nu, \mathbb{S})$ denotes a (Lipschitz) parameter projection like those defined in [13], where the component of ν orthogonal to $\partial\mathbb{S}$ is removed as s approaches $\partial\mathbb{S}$. This results in the properties 1) $s(t_0) \in \mathbb{S} \implies s \in \mathbb{S}$ for all $t \geq t_0$, and 2) $\nabla_s J \cdot \mathrm{Proj}(-k\Gamma\, \nabla_s J^T, \mathbb{S}) \leq 0$. For brevity, the reader is referred to [13] and reference therein for details on the design of such an operator. We note that applying this operator to θ serves simply to ensure that $\theta(t) \in \Theta$, not to enforce $u(t) \in \mathbb{U}$. Enforcing $u(t) \in \mathbb{U}$ by selection of Θ (rather than using B_u) is possible in special cases when $\mathbb{U}$ is a convex set, and ϕ is convex in both arguments.

Lemma 2. *Over any interval of existence $t \in [t_0, t_1]$ of the solution to* (12a) *starting from $(t^\theta,\ \theta)(t_0) \in \Phi(t_0, x(t_0), \mathcal{P})$, the closed-loop flows satisfy 1) $\frac{dJ}{dt} = \nabla_t J + \nabla_z J\, \dot{z} < 0$ when $x \notin \Sigma_{\mathbb{X}}$, and 2) $(t^\theta,\ \theta)(t) \in \Phi(t, x(t), \mathcal{P})$.*

Step 3: Parameter re-initialization

When the equality $t = t_1^\theta$ occurs, the n_θ parameters assigned to the first interval are no longer useful as degrees of freedom for minimizing (11a); instead, it is more beneficial to reassign these degrees of freedom to a new interval at the tail of the prediction horizon. This takes the form of the discrete jump mapping

$$z^+ = \left\{ \begin{array}{ccc} x^+ & = & x \\ (t_i^\theta)^+ & = & \begin{cases} t_{i+1}^\theta & i = 0 \ldots (N-1) \\ t_N^\theta + \delta(x^p(t_N^\theta)) & i = N \end{cases} \\ (\theta_i)^+ & = & \begin{cases} \theta_{i+1} & i = 1 \ldots (N-1) \\ \kappa(x^p(t_N^\theta)) & i = N \end{cases} \end{array} \right\} \qquad \text{if } t \geq t_1^\theta \tag{14}$$

where the feedbacks $\kappa(\cdot)$ and $\delta(\cdot)$ are used to initialize the parameters for the new interval. Following execution of (14), the algorithm repeats back to Step 2.

Lemma 3. *The jump mapping in* (14) *is such that 1) $J(t, z^+) - J(t, z) \leq 0$, and 2) $(t^\theta,\ \theta)^+ \in \Phi(t, x, \mathcal{P})$*

Remark 1. *The manner in which the horizon t_N^θ recedes (i.e. by* (14)*) differs from many other realtime approaches, in which the horizons recede continuously.*

While it may seem more natural to enforce a continuous recede $\dot{t}^\theta$, *this generally violates the dynamic programming principle, in which case stability can only be claimed if one assumes either 1)* $N = 1$ *and* (1) *is globally prestabilized [3], 2)* ϕ *contains a very large number of bases, or 3)* $\left|t_i^\theta - t_{i-1}^\theta\right|$ *is very small [2]. In contrast, we require none of these assumptions. (While Lemma 1 implies "sufficiently large N", the requirements for feasible initialization are* significantly *less conservative than for preservation of stability as in [2]).*

5.3 Hybrid Trajectories and Stability

The closed-loop behaviour resulting from the algorithm in Section 5.2 is that of a dynamic control law whose controller states exhibit discontinuous jumps. As such, neither classical notions of a "solution" nor those from the sampled-data literature apply to the closed-loop dynamics. Instead, a notion of solution developed for hybrid systems in [14] (and other recent work by the same authors) can be applied, in which trajectories are described as evolving over the "hybrid time" domain - i.e. a subset of $[0, \infty) \times \mathbb{N}_0$ given as a union of intervals of the form $[t_j, t_{j+1}] \times \{j\}$. In this context, the continuous dynamics (12a) have the form $\dot{z}_\pi = F(z_\pi)$ on the *flow domain*

$$S_F \triangleq \{\ z_\pi\ :\ \pi_0 \leq \pi_1 \text{ and } (t^\theta,\, \theta) \in \Phi(t, x, \mathcal{P})\ \}, \quad t^\theta \equiv t^\theta(t, \pi), \quad t \text{ arbitrary} \tag{15}$$

where z_π denotes a coordinate change of z with t^θ transformed by (13). Likewise, (14) has the form $z_\pi^+ = H(z_\pi)$ on the *jump domain*

$$S_H \triangleq \{\ z_\pi\ :\ \pi_0 \geq \pi_1 \text{ and } (t^\theta,\, \theta) \in \Phi(t, x, \mathcal{P})\ \}, \quad t^\theta \equiv t^\theta(t, \pi), \quad t \text{ arbitrary} \tag{16}$$

Lemmas 2 and 3 guarantee the invariance of $S_F \cup S_H$, the domain on which either a flow or jump is always defined. Although S_F and S_H intersect, uniqueness of solutions results from the fact that $F(z_\pi)$ points out of S_F on $S_F \cap S_H$ [15, Thm III.1]. In the language of [15], the resulting closed-loop system is a nonblocking, deterministic hybrid automaton which accepts a unique, infinite execution. Using this notion of solution, the behaviour can be summarized as follows:

Theorem 1. *Let an input parameterization* $\mathcal{P}$ *be selected to satisfy Assumption 1, and assume that a corresponding local stabilizer* $\kappa(x)$, $\delta(x)$ *and penalty function* $W(x)$ *are found which satisfy Assumption 2 on the* $\mathcal{X}_f$. *Furthermore, let the constraints in* (2c) *be enforced by barrier functions satisfying Criteria 1 and 2. Then, using the dynamic feedback algorithm detailed in Section 5.2, the target set* Σ *is feasibly, asymptotically stabilized with domain of attraction* $\mathbb{X}_{doa}(N)$ *containing* $\mathcal{X}_f$. *Furthermore,* $\exists N^* \geq 1$ *such that* $\mathbb{X}_{doa}(N) \equiv \mathbb{X}^0$ *for* $N \geq N^*$.

6 Simulation Example

To illustrate implementation of our approach, we consider regulation of the stirred tank reactor from [16], with exothermic reaction $A \longrightarrow B$ resulting in dynamics

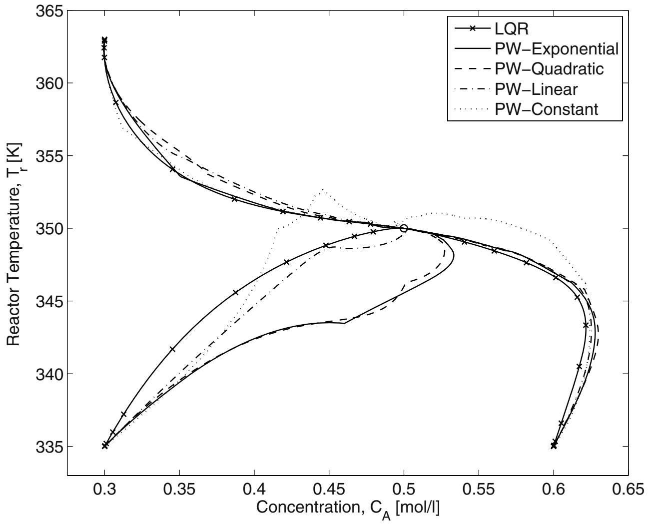

Fig. 1. Closed-loop state profiles from three different x_0, using various ϕ

$$\dot{C}_A = \frac{v}{V}\left(C_{Ain} - C_A\right) - k_0 \exp\left(\frac{-E}{RT_r}\right) C_A$$

$$\dot{T}_r = \frac{v}{V}\left(T_{in} - T_r\right) - \frac{\Delta H}{\rho\, c_p} k_0 \exp\left(\frac{-E}{RT_r}\right) C_A + \frac{UA}{\rho\, c_p\, V}\left(T_c - T_r\right)$$

Constants are taken from [16]: v=100 ℓ/min, V=100 ℓ, $\rho\, c_p$=239 J/ℓ K, E/R = 8750 K, k_0=7.2×10^{10} min^{-1}, UA=5×10^4 J/min·K, ΔH=−5×10^4 J/mol, C_{Ain}=1 mol/ℓ, T_{in}=350 K. The target is to regulate the unstable equilibrium C_A^{eq}=0.5 mol/ℓ, T_r^{eq}=350 K, T_c^{eq}=300 K, using the coolant temperature T_c as the input, subject to the constraints $0 \leq C_A \leq 1$, $280 \leq T_r \leq 370$ and $280 \leq T_c \leq 370$.

Using the cost function $L(x,u) = x'Qx + u'Qu$, with $x = [C_A - C_A^{eq},\ T_r - T_r^{eq}]'$, $u = (T_c - T_c^{eq})$, $Q = diag(2,\ 1/350)$, $R = 1/300$, the linearized local controller $k_f(x) = [109.1,\ 3.3242]\,x$ and cost $W(x) = x'Px$, $P = [17.53,\ 0.3475;\ 0.3475,\ 0.0106]$, were chosen. Four different choices of the basis $\phi(\tau, \theta_i)$ were tested,

$$\phi_C = \theta_{i1} \quad \phi_L = \theta_{i1} + \theta_{i2}\tau \quad \phi_Q = \theta_{i1} + \theta_{i2}\tau + \theta_{i3}\tau^2 \quad \phi_E = \theta_{i1}\exp\left(-\theta_{i2}\tau\right)$$

with N chosen (intentionally small) such that the total size of θ remained similar (N_C=8, N_L=N_E=4, N_Q=3). In each case, the gains k_θ=0.1 and k_t=0.5 were used in the update laws, with $\Gamma_t \equiv I$ and Γ_θ chosen as a diagonally scaled identity matrix (i.e. scaled steepest-descent updates). The feedbacks $\kappa(x)$ were derived by analytically solving (7), while $\delta(x)$ was chosen using forward simulation as described in Section 4.1. In all cases, initial conditions for t^θ and θ were chosen to approximate the trajectory $T_c(t)$, $t \in [0, 1.5]$, resulting under LQR feedback $u = k_f(x)$.

Three different initial conditions were tested, and the closed-loop state profile for each parameterization are shown in Figures 1 and 2, with corresponding

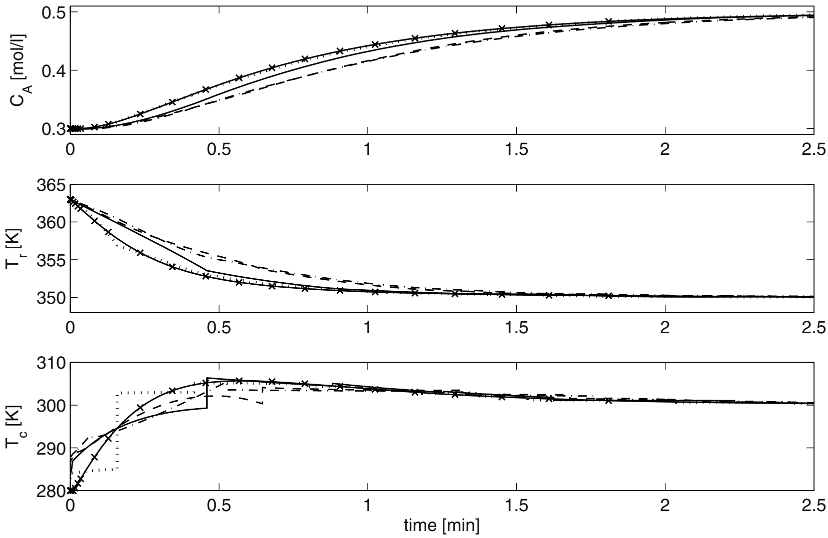

Fig. 2. Closed-loop trajectories from $(C_A, T) = (0.3, 363)$. Symbols same as Fig. 1.

Table 1. Actual closed-loop cost to practically (at t=10min) stabilize to setpoint

$(C_A, T_r)_0$	LQR	ϕ_C	ϕ_L	ϕ_Q	ϕ_E
(0.3, 363)	0.285	0.310	0.281	0.278	0.279
(0.3, 335)	1.74	1.80	1.55	1.42	1.41
(0.6, 335)	0.596	0.723	0.570	0.567	0.558

closed-loop costs reported in Table 1. Using higher-order parameterizations such as ϕ_E and ϕ_Q over coarse time-intervals resulted in lower cost than ϕ_C (which used smaller intervals), thus making better use of approximately the same number of optimization parameters. Although the equilibrium is open-loop unstable, large interval-lengths are not problematic since (12a) provides a continuous-time state-feedback for $k_\theta > 0$.

7 Conclusions

In this work, a framework has been proposed for continuous-time NMPC in which the dynamics associated with the nonlinear program are allowed to evolve in the same timescale as the process dynamics, without compromising closed-loop stability. The unique manner in which the prediction horizon recedes accommodates the use of efficient basis functions capable of parameterizing the input trajectory over large intervals using relatively few parameters. Adapting the time support of the parameterization, if desired, helps to maximize the efficiency of the parameterization. By allowing for stabilization to a general target set, a broad class of control problems can be addressed within the given framework.

References

[1] Diehl, M., Bock, H., and Schloder, J., "A real-time iteration scheme for nonlinear optimization in optimal feedback control", *SIAM Journal of Control and Optimization*, **43(5)**, 1714–1736 (2005).

[2] Ohtsuka, T., "A continuation/GMRES method for fast computation of nonlinear receding horizon control", *Automatica*, **4(40)**, 563–574 (2004).

[3] Cannon, M. and Kouvaritakis, B., "Continuous-time predictive control of constrained non-linear systems", in: F. Allgower and A. Zheng (eds.), *Nonlinear Model Predictive Control: Assessment and Future Directions for Research*, Birkhauser-Verlag (2000).

[4] Findeisen, R. and Allgöwer, F., "Stabilization using sampled-data open-loop feedback – a nonlinear model predictive control perspective", in: *Proc. IFAC Symposium on Nonlinear Control Systems*, 735–740 (2004).

[5] DeHaan, D. and Guay, M., "A real-time framework for model predictive control of continuous-time nonlinear systems", in: *Proc. IEEE Conf. on Decision and Control* (2005), To Appear.

[6] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M., "Constrained model predictive control: Stability and optimality", *Automatica*, **36**, 789–814 (2000).

[7] Clarke, F., Ledyaev, Y., Sontag, E., and Subbotin, A., "Asymptotic controllability implies feedback stabilization", *IEEE Trans. Automat. Contr.*, **42(10)**, 1394–1407 (1997).

[8] Nešić, D. and Teel, A., "A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models", *IEEE Trans. Automat. Contr.*, **49(7)**, 1103–1122 (2004).

[9] Wills, A. and Heath, W., "Barrier function based model predictive control", *Automatica*, **40**, 1415–1422 (2004).

[10] Grimm, G., Messina, M., Tuna, S., and Teel, A., "Examples when model predictive control is non-robust", *Automatica*, **40(10)**, 1729–1738 (2004).

[11] Nesterov, Y. and Nemirovskii, A., *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia (1994).

[12] Leis, J. and Kramer, M., "The simultaneous solution and sensitivity analysis of systems described by ordinary differential equations", *ACM Transactions on Mathematical Sofware*, **14(1)**, 45–60 (1988).

[13] Krstic, M., Kanellakopoulos, I., and Kokotovic, P., *Nonlinear and Adaptive Control Design*, Wiley and Sons, New York (1995).

[14] Goebel, R., Hespanha, J., Teel, A., Cia, C., and Sanfelice, R., "Hybrid systems: generalized solutions and robust stability", in: *Proc. IFAC Symposium on Nonlinear Control Systems*, 1–12 (2004).

[15] Lygeros, J., Johansson, K., Simić, S., Zhang, J., and Sastry, S., "Dynamical properties of hybrid automata", *IEEE Trans. Automat. Contr.*, **48(1)**, 2–17 (2003).

[16] Magni, L., De Nicolao, G., Magnani, L., and Scattolini, R., "A stabilizing model-based predictive control for nonlinear systems", *Automatica*, **37(9)**, 1351–1362 (2001).

A Proof of Lemma 2

It can be shown from (11a) that $\nabla_t J = -L^a(x^p, u_\phi) - \langle \nabla_x J,\, f(x^p, u_\phi)\rangle$. From (12a),

$$\begin{aligned}\frac{dJ}{dt} &= \nabla_t J + \nabla_z J\, \dot{z} \\ &= -L^a(x^p, u_\phi) - \big\langle \nabla_{t^\theta} J,\ \mathrm{Proj}\{k_t\, \alpha\, \Gamma_t\, \nabla_{t^\theta} J^T,\ \Xi\}\big\rangle - \big\langle \nabla_\theta J,\ \mathrm{Proj}\{k_\theta \Gamma_\theta\, \nabla_\theta J^T,\ \Theta^N\}\big\rangle \\ &\le -\gamma_L(\|x^p, u_\phi\|_\Sigma)\end{aligned}$$

The conditions of the lemma guarantee that $J(t_0, z_0)$ is bounded (although not uniformly), and the above ensures that $J(t, z) \le J(t_0, z_0)$, for all $t \in [t_0, t_1]$. Since all dynamics in (12a) are locally Lipschitz on the set $\mathcal{Z} = \{\ z\ :\ (t^\theta,\ \theta) \in \Phi(t, x(t), \mathcal{P})\ \}$, continuity of the solution implies that the states can only exit $\mathcal{Z}$ by either 1) reaching the boundary $\mathcal{A} = cl\{\mathcal{Z}\} \setminus \mathcal{Z}$ (i.e. the set where $x \in \partial\mathbb{X}$, $u_\phi \in \partial\mathbb{U}$, or $x^p(t_N^\theta) \in \partial\mathcal{X}_f$), or 2) passing through the boundary $\mathcal{B} = \mathcal{Z} \setminus \mathring{\mathcal{Z}}$. The first case is impossible given the decreasing nature of J and $\lim_{z\to\mathcal{A}} J(t, z) = \infty$, while the second case is prevented by the parameter projection in (12a).

B Proof of Lemma 3

The first claim follows from

$$\begin{aligned}J(t, z^+) - J(t, z) &= \int_{t_N^\theta}^{t_N^{\theta+}} L^a(x^p(\tau, t, z^+, \phi),\, u_\phi(\tau, z^+, \phi))\, d\tau + W^a(x_f^{p+}) - W^a(x_f^p) \\ &= \int_0^{\delta(x_f^p)} L(x_\kappa(\tau),\, \phi(\tau, \kappa(x_f^p)) + \mu\Big(B_x(x_\kappa(\tau)) + B_u(\phi(\tau, \kappa(x_f^p)))\Big)\, d\tau \\ &\qquad + W(x_f^{p+}) - W(x_f^p) + \mu\Big(B_{x_f}(x_f^{p+}) - B_{x_f}(x_f^p)\Big) \\ &\le 0 \qquad \text{(by (4) and (9))}\end{aligned}$$

where $x_f^p \triangleq x^p(t_N^\theta, t, z, \phi)$, $x_f^{p+} \triangleq x^p(t_N^{\theta+}, t, z^+, \phi)$, and $x_\kappa(\cdot)$ is the solution to $\dot{x}_\kappa = f(x_\kappa, \phi(t, \kappa(x_f^p)))$, $x_\kappa(0) = x_f^p$. The second claim follows by the properties of $\kappa(x)$ guaranteed by Assumption 2, since the portion of the $x^p(\tau)$ and $u_\phi(\tau)$ trajectories defined on $\tau \in (t,\, t_N^\theta]$ are unaffected by (14).

C Proof of Theorem 1

Using the cost $J(z_\pi)$ as an energy function (where $J(z_\pi) \equiv J(s, x, t^\theta - s, \theta)$ from (11a), with s arbitrary), the result follows from the Invariance principle in [15, Thm IV.1]. The conditions of [15, Thm IV.1] are guaranteed by Lemmas 2, 3, and the boundedness of the sets $\mathbb{X}$, $\mathbb{U}$, Θ and Ξ (which ensures that trajectories remain in a compact subset of $S_F \cup S_H$). Thus, z_π asymptotically converge to M, the largest invariant subset of $\{z_\pi \mid \dot{J} = 0 \text{ under (12a)}\} \cup \{\ z_\pi\ :\ J^+ - J = 0 \text{ under (14) }\}$. Since H maps into the interior of S_F (strictly away from S_H), zeno solutions are not possible. This implies $M \subset \{z_\pi \mid \dot{J} = 0\}$, and thus from the proof of Lemma 2 it follows that $M = \{\ z_\pi\ :\ (x, u_\phi) \in \Sigma\ \}$. Feasibility holds from Lemmas 2 and 3, while the last claim follows from Lemma 1 and the compactness of $\mathbb{X}^0$.