A NEW REAL-TIME APPROACH FOR NONLINEAR MODEL PREDICTIVE CONTROL

Darryl DeHaan[1] **Martin Guay**[2]

Department of Chemical Engineering, Queen's University, Kingston, ON, Canada K7L 3N6

Abstract: A new formulation of continuous-time nonlinear model predictive control is developed in which the parameters defining a piecewise-constant input trajectory are adapted continuously in real time. Stability is then proven when the parameter minimization evolves in the same timescale as the process dynamics. Pointwise constraints are incorporated by use of barrier functions.

Keywords: nonlinear control, predictive control, process control

1. INTRODUCTION

Model predictive control (MPC) is a control approach in which the current control move is obtained by repeatedly solving online a finite horizon, open-loop optimal control problem. At each instant, the optimal control problem is solved starting from the current state, and the input is set to the first move in the computed optimal input sequence. Nonlinear MPC has been the object of extensive study in recent years, and the theory related to stability and optimality has reached a point of relative maturity, as outlined in (Mayne *et al.*, 2000).

For general nonlinear systems, the optimal control programming problem which must be solved at every instant is nonconvex. Although much of the stability literature assumes the use of global solutions, in practice the standard approach is to identify a local optimum by linearizing the system about an initial candidate input trajectory, and solving a sequential quadratic programming problem (Biegler, 1998; Diehl *et al.*, 2002).

In order to avoid the iterative solution of nonlinear programs, (Ohtsuka, 2004) presents an approach, inspired by continuation methods, for updating the control vector by a continuous-time update law. In that work, however, dynamics associated with the minimization are eliminated by requiring that the optimal control problem is initially solved offline to provide an *optimal*, rather than simply feasible, initial control parameterization.

Instead of assuming that an optimal control parameterization is instantaneously available at the switching intervals, as is the case in standard NMPC stability proofs, in this work the evolution of the nonlinear program (NLP) is modeled as a continuous-time update law and it is demonstrated that stability can be preserved even when the NLP evolves throughout the control interval. Furthermore, the stability proof does not depend on the coarseness with which the input trajectory is discretized, and thus provides a means to reduce the dimensionality of the NLP without compromising closed-loop stability.

This paper is organized as follows. Section 2 describes the problem and relevant assumptions, while section 3 outlines the design approach. Section 4 contains a brief simulation example.

---

[1] Work supported by the National Sciences and Engineering Research Council of Canada.
[2] Corresponding author. Tel: 1-613-533-2788, email: guaym@chee.queensu.ca.

2. SYSTEM DESCRIPTION

The primary objective of the control design is assumed to be asymptotic regulation of the general nonlinear system

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

to the origin $x = 0$. The mapping $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is assumed to be smooth, and satisfies $f(0,0) = 0$. The state and input trajectories are everywhere subject to the pointwise-in-time constraints $x \in \mathbb{X} \supset \mathcal{X}_0$ and $u \in \mathbb{U} \subseteq \mathbb{R}^m$, with $\mathbb{X}$ and $\mathbb{U}$ closed, convex sets having nonempty interiors (denoted $\mathring{\mathbb{X}}$ and $\mathring{\mathbb{U}}$, respectively).

A second objective of the control is to achieve satisfactory performance with respect to a given, meaningful cost function $J_\infty = \int_{t_0}^{\infty} L(x,u)d\tau$. The function $L : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_{\geq 0}$ is assumed to be C^1, and is upper and lower bounded by class $\mathcal{K}_\infty$ functions $\gamma_U(\|x,u\|)$ and $\gamma_L(\|x,u\|)$, respectively. Since online infinite horizon calculations are impractical, this cost is approximated over a receding horizon $t_f = t_0 + T$ as

$$J_{rh}(x(\cdot),u(\cdot)) = \int_{t_0}^{t_f} L(x,u)\, d\tau + W(x(t_f)) \tag{2}$$

where the penalty $W : \mathcal{X}_f \to \mathbb{R}_{\geq 0}$ is positive semi-definite and C^1 on the terminal set $\mathcal{X}_f$.

Assumption 1. $T, \mathbb{X}, \mathcal{X}_f, \mathbb{U}$ are such that for every initial state $x_0 \in \mathring{\mathbb{X}}$, there exists a trajectory $(x(t),u(t))$ of (1) such that $(x(t),u(t)) \in \mathring{\mathbb{X}} \times \mathring{\mathbb{U}}$, $\forall t \geq t_0$ and $x(t_0+T) \in \mathring{\mathcal{X}}_f$

The next assumption is a slightly strengthened version of the sufficient conditions for MPC stability given in (Mayne *et al.*, 2000).

Assumption 2. A *known* feedback $u = \alpha(x)$, where $\alpha : \mathbb{X} \to \mathbb{R}^m$ is locally Lipschitz, satisfies

(1) $\mathcal{X}_f \subset \mathring{\mathbb{X}}$, $\mathcal{X}_f$ closed, $0 \in \mathcal{X}_f$.
(2) $\alpha(x) \in \mathring{\mathbb{U}}$ for all $x \in \mathcal{X}_f$.
(3) $\dot{x} = f(x, \alpha(x))$ renders $\mathcal{X}_f$ positive invariant.
(4) $\exists \lambda > 0$ such that $\forall x \in \mathcal{X}_f$,
$\nabla W^T f(x,\alpha(x)) + L(x,\alpha(x)) \leq -\lambda \|x\|^2$.

For the remainder of the paper, we denote by $f_\alpha(x,v) \triangleq f(x,\, \alpha(x)+v)$ the nominal closed-loop dynamics with control $u = \alpha(x) + v$.

3. DESIGN APPROACH

3.1 Barrier functions

Constraints are incorporated by augmenting the functions L and W with *gradient-recentered* barrier functions of the form (Wills and Heath, 2004)

$$\begin{aligned}
&L^a(x,u) = L(x,u) + \mu\left(\bar{B}_x(x) + \bar{B}_u(u)\right) \\
&W^a(x_f) = W(x_f) + \mu \bar{B}_{x_f}(x_f) \\
&\bar{B}_i(i) = B_i(i) - B_i(0) - \nabla B_i(0)^T i,\ i \in \{x, u, x_f\}
\end{aligned}$$

The functions B_x, B_u, B_{x_f} can be any appropriate barrier functions defined on the convex sets $\mathbb{X}$, $\mathbb{U}$, and $\mathcal{X}_f$, with weighting constant $\mu > 0$. By construction, gradient-recentering the barriers preserves $L^a(0,0) = 0$ and $W^a(0) = 0$. For computational and scaling reasons, use of self-concordant barrier functions is preferable when possible (Nesterov and Nemirovskii, 1994).

Assumption 3. The barriers B_x, B_u, B_{x_f} and weighting μ are chosen to satisfy

$$\nabla \bar{B}_{x_f}(x)^T f_\alpha(x,0) + \bar{B}_x(x) + \bar{B}_u(\alpha(x)) \leq \frac{\lambda}{\mu}\|x\|^2$$

for all $x \in \mathcal{X}_f$.

Remark 4. It can be shown from the closedness of $\mathcal{X}_f$, its invariance under $\dot{x} = f_\alpha(x,0)$, and the strict containments $\mathcal{X}_f \subset \mathring{\mathbb{X}}$ and $\alpha(x) \in \mathring{\mathbb{U}}$, that for any given barriers B_x, B_u, B_{x_f}, and given set $\mathcal{X}_f$ satisfying Assumption 2, there exists $\mu^* > 0$ such that Assumption 3 holds for all $\mu \in (0,\, \mu^*)$.

Using Assumption 3, the statement of Assumption 2.4 is equivalent to

$$\nabla {W^a}^T f(x,\alpha(x)) + L^a(x,\alpha(x)) \leq 0,\ \forall x \in \mathcal{X}_f \tag{3}$$

Remark 5. A complementary approach to attain (3) is outlined in (Wills and Heath, 2004), where for a system with a controllable linearization, choices $\alpha = Kx$, $L = \|x\|_Q^2 + \|u\|_R^2$, and self-concordant barrier functions with *specified* μ, the set $\mathcal{X}_f$ satisfying (3) is solved for as a level set of $W = \|x\|_P^2$, where P solves a Lyapunov equation.

3.2 Control parametrization

Let $N \in \mathbb{N}$ be a design parameter, relative to which we define the time sequence $t^\theta = t^\theta(t_0^\theta, N) \triangleq \{t_i^\theta = t_0^\theta + i\frac{T}{N} : i = 1, 2 \ldots N\}$. Define θ as a vector of length mN of the form

$$\theta = \begin{bmatrix} \theta_1^1 \\ \vdots \\ \theta_N^1 \\ \vdots \\ \theta_1^m \\ \vdots \\ \theta_N^m \end{bmatrix} \triangleq \begin{bmatrix} \theta^1 \\ \theta^2 \\ \vdots \\ \theta^m \end{bmatrix}$$

where each θ^j is a vector of N parameters corresponding to input u_j. The control is then

$$u(t,x,t^\theta,\theta) = \alpha(x) + v(t,t^\theta,\theta) \tag{4}$$

where for each $j \in \{1, \dots m\}$

$$v_j(t, t^\theta, \theta) = \begin{cases} \theta_1^j & t \in [t_0^\theta,\, t_1^\theta] \\ \theta_i^j & t \in (t_{i-1}^\theta,\, t_i^\theta],\ i = 2 \dots N \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Essentially, $v(t, t_0, \theta)$ corresponds to a piecewise constant (in time) parametrization of each of the m control inputs. For every (t_0, x_0) initial condition and (t^θ, θ) control parametrization (in general, t_0 and t_0^θ need not be equal) we denote the corresponding solution (in the classical sense) to (1), (4), (5) by $x^p(t, t_0, x_0, t_0^\theta, \theta)$ and $u^p(t, t_0, x_0, t_0^\theta, \theta)$. At times we may condense this notation to $x^p(t, \cdot)$, $u^p(t, \cdot)$, when the omitted arguments are obvious.

Since every $x_0 \in \mathring{\mathbb{X}}$ is assumed to admit an input trajectory $u(t)$ which is feasible with respect to the *open* constraint sets $(\mathring{\mathbb{X}}, \mathring{\mathbb{U}}, \mathring{\mathcal{X}}_f)$, we appeal to the recent literature on piecewise constant control (see (Clarke and Stern, 2003) and references therein) to claim without proof that for $N \geq N^*(x_0, T)$ sufficiently large, there exists a θ for which the solutions $x^p(t, t_0, x_0, t_0, \theta)$ and $u^p(t, t_0, x_0, t_0, \theta)$ satisfy all input, state and terminal constraints. We define $\Theta \equiv \Theta(x_0, N) \subseteq \mathbb{R}^{mN}$ as the set of all such feasible θ, which we note is an open set due to the openness of $(\mathring{\mathbb{X}}, \mathring{\mathbb{U}}, \mathring{\mathcal{X}}_f)$.

3.3 Overall Algorithm

For any given $(t_0, x_0) \in \mathbb{R} \times \mathring{\mathbb{X}}$, we assume that an initial feasible control parametrization (t^θ, θ_0) is known, satisfying $\theta_0 \in \Theta(x_0, N)$ and $t_0^\theta = t_0$. For the important case in which $\mathcal{X}_f = \mathbb{X}$ (i.e. $u = \alpha(x)$ is stabilizing over all of $\mathbb{X}$), a natural initial choice would be $\theta_0 = 0$.

For the interval $t \in [t_0,\, t_1]$, with $t_1 = t_0 + \frac{T}{N}$, we modify (2) as

$$J(t, x, \theta, t_0^\theta) = W^a(x^p(t_f, t, x, t_0^\theta, \theta)) + \int_t^{t_f} L^a(x^p(\tau, t, x, t_0^\theta, \theta), u^p(\tau, t, x, t_0^\theta, \theta))d\tau \quad (6)$$

where the terminal time $t_f = t_0 + T$ *remains constant* in time. The vector θ in (6) is a state of the controller, which is modified in continuous time according to

$$\dot{\theta} = -k_\theta \Gamma \nabla_\theta J \quad (7)$$

where $k_\theta > 0$ is a design parameter, and $\Gamma = \Gamma(t, \theta) > 0$ is a (time-varying) positive definite matrix function, the design of which will be discussed in section 3.7. Calculation of $\nabla_\theta J$ is discussed in section 3.6.

At $t = t_1$, the control parametrization (t^θ, θ) is "shifted" by the following jump map

$$(t_i^\theta)^+ = \begin{cases} t_{i+1}^\theta & i < N \\ t_N^\theta + \dfrac{T}{N} & i = N \end{cases} \quad (8)$$

$$(\theta_i^j)^+ = \begin{cases} \theta_{i+1}^j & i < N \\ 0 & i = N \end{cases}, \quad j = 1, \dots m \quad (9)$$

where the notation $(t^{\theta+}, \theta^+)$ denotes the new parametrization after the jump. If $\theta(t_1) \in \Theta(x_0, N)$ immediately prior to reset, then it follows from (5),(9) and Assumption 2 that $\theta^+ \in \Theta(x(t_1), N)$. The algorithm therefore repeats back to the beginning, with (t_0, x_0, θ_0) re-initialized as the post-jump values $(t_1, x(t_1), \theta^+)$ from the previous iteration. Since t_f is defined relative to t_0, the prediction horizon therefore recedes in discrete jumps.

3.4 Existence and Uniqueness of Solutions

Since the controller state θ exhibits both continuous flows as well as discrete jumps, the classical notion of "solution" does not apply to the closed-loop response. Instead, we will adopt a notion of solution developed for hybrid systems.

Define $\omega = [z, x^T, \theta^T]^T$, where z is the "time since last reset" ($\dot{z} = 1$, $z^+ = 0$), and define the domains over which flow and jumps occur as $\{\omega \,|z \leq \frac{T}{N}\}$ and $\{\omega \,|z \geq \frac{T}{N}\}$, respectively. We can then use the framework of (Goebel *et al.*, 2004) to state that solutions of the form $\omega(t, k)$ exist in the sense of Filippov or Krasovskii, where t denotes ordinary time (evolution along continuous flows), while k denotes event time (evolution according to jumps). Despite the apparent nondeterminism at $z = \frac{T}{N}$ (i.e. both a flow and a jump are defined), uniqueness of the solution follows from the fact that the flow field $\dot{z} = 1$ points out of the flow domain and into the interior of the jump domain (Goebel *et al.*, 2004). Finally, we point out that by construction our hybrid dynamics are incapable of multiple successive jumps at a single moment in ordinary time, so zeno behaviour (i.e. infinite jumps in finite ordinary time) is excluded (Lygeros *et al.*, 2003).

3.5 Main Result

Theorem 6. If $\alpha(x)$ satisfies Assumption 1, and (x_0, θ_0) are a feasible, stabilizing set of initial conditions for the selected (T,N), then stability and feasibility are preserved under (7), (9).

PROOF. As is standard in MPC approaches, we will prove stability by using the finite horizon cost (6). However, since our closed loop trajectory $\omega(t, k)$ evolves in the hybrid time domain, it will be necessary to show decrease of J with respect to both ordinary and event time. We will therefore use the notation $x(t,k)$, $u(t,k)$, $\theta(t,k)$ to denote the resulting solutions to (1), (4), (7), (9) in the hybrid time domain.

Event time
At $t = t_1$, we denote the change in J under the jump mapping (and corresponding redefinition of t_f) as $\Delta J(t_1) = J(t_1, x(t_1, k+1), \theta(t_1, k+1)) - J(t_1, x(t_1,k), \theta(t_1,k))$. Since $\theta_N^+ = 0$, it can be seen that

$$\begin{aligned}\Delta J(t_1) &= \int_{t_f(k)}^{t_f(k+1)} L^a(x^p, u^p) d\tau \\ &\quad + W^a(x(t_f(k))) - W^a(x(t_f(k+1))) \\ &= \int_{t_f(k)}^{t_f(k+1)} L^a(x^p, \alpha(x^p)) + \nabla W^{a\,T} f_\alpha(x^p, 0) d\tau \\ &\leq 0 \end{aligned} \tag{10}$$

where the inequality follows from (3).

Ordinary time
We will use the notation $\dot{J}(k)$ to denote evolution of J along continuous flows of the system *for a period of constant event-time k*. Then

$$\dot{J}(k) = \nabla_t J + \langle \nabla_x J,\, f^\omega(t,k)) \rangle + \langle \nabla_\theta J, \dot{\theta} \rangle$$

where $f^\omega(t,k) \triangleq f_\alpha(x(t,k), \theta_1(t,k))$, and $\dot{\theta}$ is given by (7). From (6) it can be shown that $\nabla_t J = \langle \nabla_x J,\, f^\omega(t,k)) \rangle - L^a(x(t,k), u(t,k))$, so

$$\begin{aligned}\dot{J}_k &= -L^a(x(t,k), u(t,k)) + \langle \nabla_\theta J, \dot{\theta} \rangle \\ &= -L^a(x(t,k), u(t,k)) - k_\theta \, \nabla_\theta J^T \, \Gamma(t) \, \nabla_\theta J \\ &< 0. \end{aligned} \tag{11}$$

It then follows by a hybrid systems version of LaSalle's Invariance principle (Lygeros *et al.*, 2003) that the system converges asymptotically to $x = 0$. Feasibility of $x(t,k)$ and $u(t,k)$ for an initial feasible θ_0 follows immediately from (10),(11), and the unboundedness of $J(t, x, \theta, t_0^\theta)$ when a constraint is approached. ∎

3.6 Calculation of $\nabla_\theta J$

For notational simplicity, we define $L_\alpha^a(x,v) \triangleq L^a(x, \alpha(x) + v)$. The gradient vector $\nabla_\theta J$ is then

$$\nabla_\theta J = \int_t^{t_f} \frac{\partial L_\alpha^a}{\partial x}\frac{\partial x^p}{\partial \theta} + \frac{\partial L_\alpha^a}{\partial v}\frac{\partial v^p}{\partial \theta} d\tau + \frac{dW^{a\,T}}{dx}\frac{\partial x^p}{\partial \theta}(t_f) \tag{12}$$

where $\frac{\partial L_\alpha^a}{\partial x}$, $\frac{\partial L_\alpha^a}{\partial v}$ and $\frac{dW^a}{dx}$ are evaluated along $x^p(\tau, \cdot)$, $u^p(\tau, \cdot)$ (and $v^p(\tau, \cdot)$), the simultaneously-calculated prediction trajectory. The sensitivity matrix has initial condition $\partial x^p / \partial \theta \, (t) = 0$ and evolves according to

$$\frac{d}{d\tau}\frac{\partial x^p}{\partial \theta} = \frac{\partial f_\alpha}{\partial x}\frac{\partial x^p}{\partial \theta} + \frac{\partial f_\alpha}{\partial v}\frac{\partial v^p}{\partial \theta} \tag{13}$$

In both (12) and (13), elements $\partial v_l^p / \partial \theta_i^j$ are given by

$$\frac{\partial v_l^p}{\partial \theta_i^j} = \begin{cases} 1 & l = j, \text{ and } \tau \in [t_{i-1}^\theta,\, t_i^\theta] \\ 0 & \text{otherwise} \end{cases}$$

Instead of solving the (mN) sensitivity equations of (13) on the entire prediction interval $\tau \in [t,\, t_f]$, it is generally more efficient to decompose $\partial x^p / \partial \theta$ on the interval $[t_{i-1},\, t_i]$ as

$$\frac{\partial x^p}{\partial \theta} = \frac{\partial x^p}{\partial x^p(t_{i-1})}\frac{\partial x^p(t_{i-1})}{\partial \theta} + \frac{\partial x^p}{\partial v^p}\frac{\partial v^p}{\partial \theta} \tag{14}$$

Since the intervals are solved sequentially, the term $\partial x^p(t_{i-1}) / \partial \theta$ is a known constant and thus only the $(m+n)$ sensitivities $\partial x^p / \partial x^p(t_{i-1})$ and $\partial x^p / \partial v^p$ need to be solved on $\tau \in [t_i,\, t_{i+1}]$. The term $\nabla_\theta J$ in (12) can be decomposed in a similar fashion, so that only $(m+n)$ elements require integrating.

Most importantly, we note that several efficient algorithms exist for simultaneously solving ODEs and their parametric sensitivity equations. Examples include the packages ODESSA (Leis and Kramer, 1988), DASPK (Li and Petzold, 2000), or ESDIRK (Kristensen *et al.*, 2004).

3.7 Selecting and Computing Γ

3.7.1. Steepest Descent The simplest approach to selecting Γ is to fix it as the constant (diagonally scaled) identify matrix. The update law (7) then becomes a steepest descent approach for minimizing J with respect to θ. While simple to implement, the poor scaling generally associated with steepest descent may result in slow convergence of θ towards any (locally) optimal control parametrization.

3.7.2. Newton Method with Trust Region Maximum performance is achieved when Newton's method is used to generate Γ. Since the cost surface J may not be convex in the parameter space, it is necessary to modify the standard Newton's method using a trust-region approach to ensure positive definiteness of Γ. This has the form

$$\Gamma = \left[\nabla_{\theta\theta}^2 J + \left(\left\| \nabla_{\theta\theta}^2 J \right\|_F + \varepsilon \right) I \right]^{-1} \tag{15}$$

where $\|\cdot\|_F$ denotes the Frobenius matrix norm, and $\varepsilon > 0$ is a small design constant.

Calculating the true Hessian $\nabla_{\theta\theta}^2 J$ requires solving the second-order sensitivity equations analogous to (13),(14). While generally computationally expensive, it has been shown in (Guay and McLean, 1995) that the same numerically efficient methods for generating first order sensitivities can be extended to the second order case as well.

3.7.3. Gauss-Newton Method If (6) can be rewritten as $J = \int_t^{t_f} \|l^a(x^p, u^p)\|_2^2 \, d\tau + \|w^a(x_f)\|_2^2$, then a Gauss-Newton approximation

$$G = 2\int_t^{t_f} \mathbb{L}(\tau)^T \mathbb{L}(\tau) d\tau + \mathbb{W}^T \mathbb{W}$$

$$\mathbb{L} = \frac{\partial l^a}{\partial(x,v)} \begin{bmatrix} \frac{\partial x^p(\tau)}{\partial \theta} \\ \frac{\partial v^p(\tau)}{\partial \theta} \end{bmatrix} \qquad \mathbb{W} = \frac{\partial w^a}{\partial x}\frac{\partial x}{\partial \theta}(t_f),$$

can be substituted for $\nabla_{\theta\theta} J$ in (15). The dimension of the quadrature can again be reduced by decomposing as in (14) and exploiting symmetry. This is essentially a sequential application of the Gauss-Newton method of (Diehl *et al.*, 2002).

4. SIMULATION EXAMPLE

To illustrate the implementation of our approach, we consider the problem of state-feedback regulation of a jacketed non-isothermal reactor with van de Vusse kinetics. The reaction mechanism is

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C$$
$$2A \xrightarrow{k_3} D$$

The states of the system consist of concentrations of components A and B, as well as the temperatures T and T_K occurring in the reactor and cooling jacket, respectively. The manipulated variables consist of the dilution rate $\frac{\dot{V}}{V_R}$ and the rate of heat removal from the jacket. All four states are assumed to be measured, and evolve according to

$$\dot{C}_A = \frac{\dot{V}}{V_R}(C_{A_0} - C_A) - k_1(T)C_A - k_3(T)C_A^2$$
$$\dot{C}_B = -\frac{\dot{V}}{V_R}C_B + k_1(T)C_A - k_2(T)C_B$$
$$\dot{T} = \frac{\dot{V}}{V_R}(T_0 - T) + \frac{k_w A_R}{\rho c_p V_R}(T_k - T) - \frac{1}{\rho c_p}\left(k_1(T)C_A \Delta H_{R_{AB}} + k_2(T)C_B \Delta H_{R_{BC}} + k_3(T)C_A^2 \Delta H_{R_{AD}}\right)$$
$$\dot{T}_k = \frac{1}{m_k c_{pk}}\left(\dot{Q}_k + k_w A_R (T - T_k)\right)$$

where the rate constants $k_i(T)$ follow the Arrhenius law, and the values of all constants are taken from (Chen *et al.*, 1995). The steady state to be regulated is $x_r = [C_A, C_B, T, T_k]_r = [2.14\frac{mol}{L}, 1.09\frac{mol}{L}, 114.2°C, 112.9°C]$ from initial conditions $(x_0 + x_r) = [1, 0.5, 100, 100]$. The inputs are $u = [\frac{\dot{V}}{V_R}, \frac{1}{100}\dot{Q}_k]$, where u_2 is rescaled for numerical considerations. At steady state, $[\frac{\dot{V}}{V_R}, \dot{Q}_k]_r = [14.19\, hr^{-1}, -1118\frac{kJ}{hr}]$. The inputs are constrained by the hypercube $3 \le \frac{\dot{V}}{V_R} \le 35\, hr^{-1}$ and $-9000 \le \dot{Q}_k \le 0\, \frac{kJ}{hr}$, which was enforced using a logarithmic barrier for B_u.

The cost function is taken to be $L(x,u) = x^T Q x + u^T R u$, with diagonal matrices $Q = \text{diag}(0.2,\ 1,\ 0.5,\ 0.2) \times 10^3$ and $R = \text{diag}(500, 5)$. The terminal cost $W = x^T P x$ and local controller $\alpha(x) = \text{sat}(Kx, \mathbb{U})$ are given by

$$K = \begin{bmatrix} -0.0381 & -0.0405 & -0.1004 & -0.0244 \\ 12.7532 & 6.2581 & 5.9558 & 3.6523 \end{bmatrix}$$

$$P = \begin{bmatrix} 70.1 & 36.6 & 20.1 & 6.4 \\ 36.6 & 33.6 & 9.9 & 3.1 \\ 20.1 & 9.9 & 10.6 & 3.0 \\ 6.4 & 3.1 & 3.0 & 1.8 \end{bmatrix}$$

where $\text{sat}(\cdot, \mathbb{U})$ denotes componentwise saturation. Although $\alpha(x)$ is not necessarily globally stabilizing, $\theta_0 = 0$ is still a feasible initial parametrization of v for the given x_0.

Figure 1 depicts the closed-loop results for a control parameterization involving 30 intervals of 20 seconds each. A Gauss-Newton update law was used, with an adaptation gain of $k_\theta = 10$. The combination of continuous-time evolution and discrete jumps are evident the input trajectories. Figure 2 depicts the resulting dilution rate u_1 and accumulated cost for different values of k_θ. The trajectory labeled OPT represents the ideal MPC input generated by instantaneous minimization at the switching points. As k_θ is increased, the performance of our proposed method improves from that of the nominal control $u = \alpha(x)$ (i.e. $k_\theta = 0$) towards that of the ideal trajectory.

All calculations were performed on an Athlon 1.6 Ghz, using the Fortran package ODESSA in conjunction with Matlab. Approximate computing time (ms) per iteration (Matlab measurements, not true CPU times) for different horizon lengths involving 20s intervals were

No. Intervals	30	50	100
Steepest Descent	$<$20ms	$<$30ms	$<$50ms
Gauss-Newton	$<$50ms	$<$80ms	$<$500ms

We note that part of the Gauss-Newton calculation was performed in Matlab, which is less efficient than an entirely Fortran implementation.

5. CONCLUSIONS

In this work an approach has been developed for incorporating the optimization stage associated with standard MPC into the controller design. This allows for the control of faster processes, by allowing the minimization to enter into the same timescale as the process dynamics, rather than requiring that it be completed within a faster timescale. In effect, the results of the parameter minimization are implemented as they are being generated, rather than waiting for successful termination of the nonlinear program.

Stability and feasibility of the proposed method is proven using slightly modified standard sufficient conditions, and does not require any "sufficiently fine" discretization of the input, other than to assume the existence of a feasible initial parameterization. While the method only guarantees improvement of an initial feasible parameterization up to a local minimum, this limitation is shared by most other practical approaches.

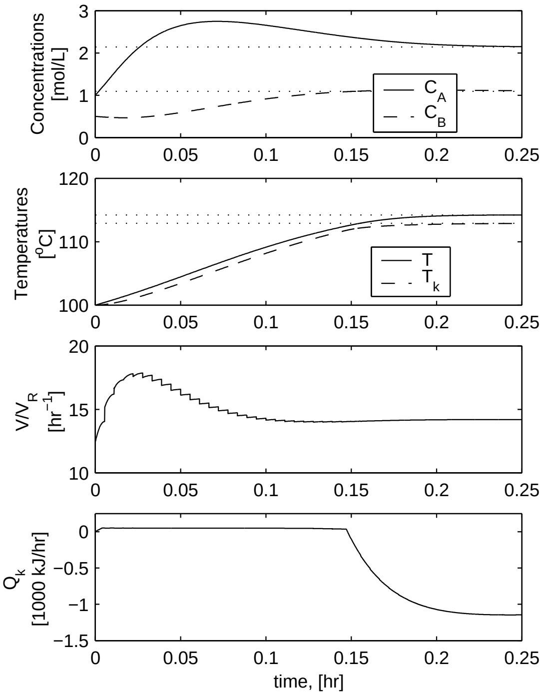

Fig. 1. Closed-loop time profiles using Gauss-Newton parameter update, with $k_\theta = 10$.

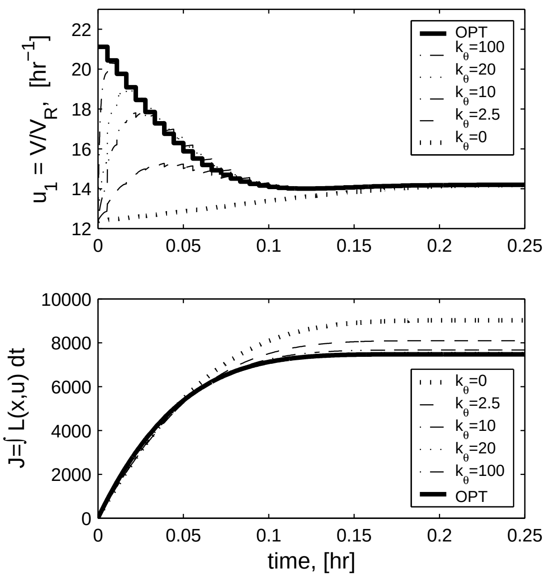

Fig. 2. Effect of increasing k_θ, using Gauss-Newton update law with 30 interval prediction.

REFERENCES

Biegler, L.T. (1998). Advances in nonlinear programming concepts for process control. *Journal of Process Control* **8**(5-6), 301–311.

Chen, H., A. Kremling and F. Allgöwer (1995). Nonlinear predictive control of a benchmark CSTR. In: *3rd European Control Conference*. pp. 3247 – 3252.

Clarke, F. H. and R. J. Stern (2003). State constrained feedback stabilization. *SIAM Journal of Control and Optimization* **42**(2), 422–441.

Diehl, M., G. Bock, J. Schloder, R. Findeisen, Z. Nagy and F. Allgöwer (2002). Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control* **12**(4), 577–588.

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

Guay, M. and D. McLean (1995). Optimization and sensitivity analysis for multiresponse parameter estimation in systems of ordinary differential equations. *Computers and Chemical Engineering* **19**(12), 1271–1285.

Kristensen, M.R., J.B. Jørgensen, P.B. Thomsen and S.B. Jørgensen (2004). Efficient sensitivity computation for nonlinear model predictive control. In: *Proc. IFAC Symposium on Nonlinear Control Systems*. pp. 723–728.

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

Li, S. and L. Petzold (2000). Software and algorithms for sensitivity analysis of large-scale differential algebraic systems. *J. Comput. Appl. Math.* **125**, 131–145.

Lygeros, J., K.H. Johansson, S.N. Simić, J. Zhang and S.S. Sastry (2003). Dynamical properties of hybrid automata. *IEEE Transactions on Automatic Control* **48**(1), 2–17.

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

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

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

Wills, A.G. and W.P. Heath (2004). Barrier function based model predictive control. *Automatica* **40**, 1415–1422.