

## Research Article

# Convergence Guaranteed Nonlinear Constraint Model Predictive Control via I/O Linearization

Xiaobing Kong,<sup>1</sup> Xiangjie Liu,<sup>1</sup> and Xiuming Yao<sup>2</sup>

<sup>1</sup> State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, Beijing 102206, China

<sup>2</sup> Department of Automation, North China Electric Power University, Baoding 071003, China

Correspondence should be addressed to Xiangjie Liu; liuxj@ncepu.edu.cn

Received 14 July 2013; Revised 23 October 2013; Accepted 28 October 2013

Academic Editor: Baoyong Zhang

Copyright © 2013 Xiaobing Kong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Constituting reliable optimal solution is a key issue for the nonlinear constrained model predictive control. Input-output feedback linearization is a popular method in nonlinear control. By using an input-output feedback linearizing controller, the original linear input constraints will change to nonlinear constraints and sometimes the constraints are state dependent. This paper presents an iterative quadratic program (IQP) routine on the continuous-time system. To guarantee its convergence, another iterative approach is incorporated. The proposed algorithm can reach a feasible solution over the entire prediction horizon. Simulation results on both a numerical example and the continuous stirred tank reactors (CSTR) demonstrate the effectiveness of the proposed method.

## 1. Introduction

Model predictive control (MPC) is a popular algorithm in process control which online solves an optimization problem at each time step [1]. MPC considers process input, output, and state constraints directly in the control variable calculation. Under the linear model, with a quadratic objective function, MPC utilizes the convex quadratic program (QP), which can easily find the optimal solution [2].

Industrial processes are generally nonlinear, due to the frequent changes of the operating point right across the whole operation range. In general, the nonlinear model predictive control (NMPC) also online solves an optimization problem, by using the sequential quadratic program (SQP). The resulting nonlinear programming problems are usually nonconvex, and the online computational burden is generally large for most complex systems. A general way to solve this nonlinear optimal problem is to use approximate approach. For example, the first control move is calculated exactly, since it is actually implemented. The rest control moves can be approximated, since they are not implemented [3]. Thus the number of decision variables in the online optimization problem is equal to the number of inputs, instead of the number of inputs multiplied by the control horizon for a conventional

NMPC algorithm. Paper [4] extended the first prediction of the linear constraints to the whole control horizon. Paper [5] studies the stability and region of attraction properties of a family of nonlinear MPC systems. Paper [6] presents nonlinear multivariable predictive control using neurofuzzy networks.

“Jacobian linearization” and “the input-output feedback linearization (IOFL)” are the two popular approaches in nonlinear control area. While the former can reflect the nonlinear model only at some certain point, the later represents the nonlinear system over a much wider operating range. Thus, IOFL is utilized in this paper for constituting nonlinear MPC, since the IOFL can offer a linear dynamic system so that the total optimal problem can be solved using the QP routine. Nevertheless, this can make the constraints to be nonlinear and state dependent. The approximate method developed in paper [7] can guarantee a feasible solution over the entire prediction horizon. The neural network is utilized to model the nonlinear discrete-time system. Paper [8] also introduces a technique with an affine transformation of the feasible region using feedback linearization scheme for handling input constraints. The neural network is also utilized to model the nonlinear system.

For digital MPC controller design, since the real-time control systems are in discrete-time forms, most NMPC research adopts the discrete-time feedback linearization. It should be noticed that the IOFL in differential geometry has been well developed for continuous-time system. Some famous methods in continuous-time system [9], for example, the extended system method, are not suitable to feedback linearise a general discrete-time system. Direct application of feedback linearization for a general discrete-time system may either be impossible or involve the time-consuming search algorithm. In [10], the authors proposed a linear model predictive control strategy via input/output linearization to the nonlinear process, by using one-step constraint algorithm. This paper aims to make full use of the advantage of continuous-time system IOFL and then try to reach the convergence to a feasible solution over the entire prediction horizon within the available time. Since the input constraints are transferred into nonlinear constraints from initial linear constraints, the convergent algorithm is constituted to guarantee a feasible solution without constraints violation. Simulation results on both a numerical example and the continuous stirred tank reactors (CSTR) demonstrate the effectiveness of the NMPC method.

## 2. The Linear Control Structure via Input-Output Feedback Linearization

The IOFL is to transform the original nonlinear system into linear input-output relationship, generally by using a static state feedback control law [11]. Consider the SISO affine state-space model as follows:

$$\begin{aligned}\dot{\bar{x}} &= f(\bar{x}) + g(\bar{x})u, \\ y &= h(\bar{x}),\end{aligned}\quad (1)$$

where  $\bar{x} \in R^n$  is state variables,  $u$  is the manipulated input variable,  $y$  is the controlled output variable, and  $f$ ,  $g$ , and  $h$  are smooth functions in a domain  $D \subset R^n$ .

**Definition 1** (relative degree). The nonlinear system (1) is said to have relative degree  $\gamma$ ,  $1 \leq \gamma \leq n$ , in a region  $D_0 \subset D$  if

$$\begin{aligned}L_g L_f^i h(\bar{x}) &\equiv 0, \quad i = 0, \dots, \gamma - 2; \\ L_g L_f^{\gamma-1} h(\bar{x}) &\neq 0, \quad \forall \bar{x} \in D_0.\end{aligned}\quad (2)$$

If the relative degree  $\gamma = n$ , then for every  $x_0 \in D$ , a neighborhood  $N$  of  $x_0$  exists such that the map  $\Phi : \bar{x} \rightarrow x = [h \quad L_f h \quad \dots \quad L_f^{n-1} h]^T$  restricted to  $N$  is a diffeomorphism on  $N$ .

The mapping  $\Phi$  transfers the system (1) to a new system as follows:

$$\begin{aligned}\dot{x}_1 &= x_2, \\ \dot{x}_2 &= x_3, \\ &\vdots \\ \dot{x}_n &= b(x) + a(x)u, \\ y &= x_1,\end{aligned}\quad (3)$$

where  $a(\bar{x}) = L_g L_f^{n-1} h(\bar{x})$ ,  $b(\bar{x}) = L_f^n h(\bar{x})$ , which can also be expressed as  $a(x)$ ,  $b(x)$  in the new coordinates.

The feedback law is

$$u = \frac{1}{a(x)} [-b(x) + v], \quad (4)$$

where  $v$  is the transformed input variable. The new linear system is

$$\begin{aligned}\dot{x} &= Ax + Bv, \\ y &= Cx,\end{aligned}\quad (5)$$

where

$$A = \begin{bmatrix} 0 & 1 & & 0 \\ & \ddots & \ddots & \\ 0 & & \ddots & 1 \\ & & & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, \quad C = [1 \quad 0 \quad \dots \quad 0]. \quad (6)$$

The resulting linear state-space system (5) could be used for constituting “standard” linear MPC [1]. Discretization (5) can result in

$$\begin{aligned}x(k+1) &= A_d x(k) + B_d v(k), \\ y(k) &= C_d x(k).\end{aligned}\quad (7)$$

The matrices  $A_d$ ,  $B_d$ , and  $C_d$  are expressed as  $A_d = e^{AT}$ ,  $B_d = \int_0^T e^{At} dt \cdot B$ ,  $C_d = C$ , where  $T$  is sampling time.

Define  $\Delta x(k+1) = x(k+1) - x(k)$ ;  $\Delta x(k) = x(k) - x(k-1)$ ;  $\Delta v(k) = v(k) - v(k-1)$ . Choose a new group of state variables  $x_u(k) = [\Delta x(k)^T \quad y(k)]^T$ ; the augmented system is

$$\begin{aligned}\begin{bmatrix} x_u(k+1) \\ \Delta x(k+1) \\ y(k+1) \end{bmatrix} &= \begin{bmatrix} A_u & & \\ A_d & 0_d^T & \\ C_d A_d & 1 & \end{bmatrix} \begin{bmatrix} x_u(k) \\ \Delta x(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_u \\ B_d \\ C_d B_d \end{bmatrix} \Delta v(k), \\ y(k) &= \begin{bmatrix} C_u \\ 0_d & 1 \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix},\end{aligned}\quad (8)$$

where  $0_d = \overbrace{[0 \ 0 \ \cdots \ 0]}^n$ . Based on the state-space model  $(A_u, B_u, C_u)$ , the future outputs along the predictive horizon  $N_p$  are calculated sequentially as

$$\begin{aligned} y(k+1|k) &= C_u A_u x_u(k) + C_u B_u \Delta v(k), \\ y(k+2|k) &= C_u A_u^2 x_u(k) + C_u A_u B_u \Delta v(k) \\ &\quad + C_u B_u \Delta v(k+1) \end{aligned} \quad (9)$$

$\vdots$

$$\begin{aligned} y(k+N_p|k) &= C_u A_u^{N_p} x_u(k) + C_u A_u^{N_p-1} B_u \Delta v(k) \\ &\quad + C_u A_u^{N_p-2} B_u \Delta v(k+1) + \cdots \\ &\quad + C_u A_u^{N_p-N_c} B_u \Delta v(k+N_c-1). \end{aligned} \quad (10)$$

Define vectors

$$Y = [y(k+1|k) \ \cdots \ y(k+N_p|k)]^T,$$

$$\Delta V = [\Delta v(k) \ \Delta v(k+1|k) \ \cdots \ \Delta v(k+N_c-1|k)]^T, \quad (11)$$

where  $N_c$  is the control horizon.

Collect (9) and (10) together in a compact matrix form as

$$Y = F x_u(k) + \Psi \Delta V, \quad (12)$$

where

$$F = \begin{bmatrix} C_u A_u \\ C_u A_u^2 \\ C_u A_u^3 \\ \vdots \\ C_u A_u^{N_p} \end{bmatrix}, \quad \Psi = \begin{bmatrix} C_u B_u & 0 & 0 & \cdots & 0 \\ C_u A_u B_u & C_u B_u & 0 & \cdots & 0 \\ C_u A_u^2 B_u & C_u A_u B_u & C_u B_u & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_u A_u^{N_p-1} B_u & C_u A_u^{N_p-2} B_u & C_u A_u^{N_p-3} B_u & \cdots & C_u A_u^{N_p-N_c} B_u \end{bmatrix}. \quad (13)$$

Similarly, we can have the future state variables along the predictive horizon  $N_c$  based on the model of  $(A_d, B_d, C_d)$

$$X = \tilde{A} \Delta V + \gamma, \quad (14)$$

where

$$\begin{aligned} X &= [x(k) \ x(k+1|k) \ \cdots \ x(k+N_c-1|k)]^T, \\ \tilde{A} &= \begin{bmatrix} 0 & 0 & \cdots & 0 \\ B_d & 0 & \cdots & 0 \\ (A_d + I) B_d & B_d & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \left(\sum_{i=1}^{N_c-1} A_d^{i-1}\right) B_d & \left(\sum_{i=1}^{N_c-2} A_d^{i-1}\right) B_d & \cdots & 0 \end{bmatrix}, \end{aligned} \quad (15)$$

$$\gamma = \begin{bmatrix} x(k) \\ x(k) + A_d \Delta x(k) \\ x(k) + \left(\sum_{i=1}^2 A_d^i\right) \Delta x(k) \\ \vdots \\ x(k) + \left(\sum_{i=1}^{N_c-1} A_d^i\right) \Delta x(k) \end{bmatrix}.$$

The object function to be minimised is a quadratic criterion on  $\Delta V$  as

$$J = (R_s - Y)^T (R_s - Y) + \Delta V^T \bar{R} \Delta V. \quad (16)$$

By using (12), the optimization problem can be attributed to

$$J_{\min} = \min_{\Delta V} \left\{ \frac{1}{2} \Delta V^T H \Delta V + \Gamma^T \Delta V \right\} \quad (17)$$

$$\text{subject to } \underline{\Delta V} \leq \Delta V \leq \overline{\Delta V},$$

where  $H = \Psi^T \Psi + \bar{R}$ ,  $\Gamma = \Psi^T (-R_s + F x_u(k))$ ,  $\underline{\Delta V}$  and  $\overline{\Delta V}$  indicates the minimal and maximum value of  $\Delta V$ .

### 3. Convergence Algorithm for Constraint Optimal MPC

**3.1. The Nonlinear Constraint Handling.** In practice, the process inputs are frequently subjected to the following level inequality constraints:

$$\underline{U} \leq U \leq \bar{U}, \quad (18)$$

where  $U = [u_k \ \cdots \ u_{k+N_c-1}]^T$  represents the control inputs, and  $\underline{U}$  and  $\bar{U}$  indicate the minimal and maximum value of  $U$ .

After the realization of the IOFL, the input vector  $U$  is transformed into a new one  $V$ . With this nonlinear and state dependent input constraint, traditional QP cannot be used to calculate the optimal control sequence. Though the sequential quadratic programming (SQP) technique is generally used on nonlinear constraint MPC, it is often nonconvex and can cause large computation burden. This property will be demonstrated later in the simulation example of CSTR.

In order to solve this nonlinear constraints problem, an iterative QP is first used to try to get the optimal solution. The key issue is to establish the nonlinear relationship between vectors  $V$  and  $U$  and then get the explicit expression of  $U$  from  $V$ .

To establish the nonlinear relationship between vectors  $V$  and  $U$ , expand the feedback law (4) over the control horizon  $N_c$ ,

$$\begin{aligned} v(k) &= a(x(k))u(k) + b(x(k)), \\ v(k+1) &= a(x(k+1))u(k+1) + b(x(k+1)), \\ &\vdots \end{aligned}$$

$$\begin{aligned} v(k+N_c-1) &= a(x(k+N_c-1))u(k+N_c-1) + b(x(k+N_c-1)). \end{aligned} \quad (19)$$

Considering the expression (14), (19) can be rearranged in the matrix form

$$V = G_u [U, \Delta V]. \quad (20)$$

The above inequality can be rewritten in the following form:

$$G [U, \Delta V] = 0 \quad (21)$$

due to

$$\begin{aligned} V &= \begin{bmatrix} v(k) \\ v(k+1) \\ \vdots \\ v(k+N_c-1) \end{bmatrix} \\ &= \begin{bmatrix} 1 & & & & 0 \\ 1 & 1 & & & \\ 1 & 1 & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \\ 1 & 1 & \dots & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta v(k) \\ \Delta v(k+1) \\ \vdots \\ \Delta v(k+N_c-1) \end{bmatrix} + v(k-1). \end{aligned} \quad (22)$$

In general,  $v(k+i-1)$  can be rewritten as

$$\begin{aligned} v(k+i-1) &= v(k-1) + \sum_{j=1}^i \Delta v(k+j-1) \quad (i=1, \dots, N_c), \end{aligned} \quad (23)$$

with the constraints

$$\underline{v} \leq v \leq \bar{v}, \quad (24)$$

where  $\underline{v}$ ,  $\bar{v}$  indicate the minimal and maximum value of  $v$ , which are state dependent.

Thus

$$\begin{aligned} &\underline{v}_{k+i-1}(x(k+i-1)) - v(k-1) \\ &\leq \sum_{j=1}^i \Delta v(k+j-1) \\ &\leq \bar{v}_{k+i-1}(x(k+i-1)) - v(k-1). \end{aligned} \quad (25)$$

The above inequality can be rewritten over the entire horizon,

$$\Lambda^T \Delta V(k) \leq c(X(\Delta V(k)))^T, \quad (26)$$

where

$$\Lambda = [L^T \quad -L^T], \quad L = \begin{bmatrix} 1 & & & 0 \\ 1 & 1 & & \\ \vdots & & \ddots & \\ 1 & 1 & \dots & 1 \end{bmatrix}, \quad (27)$$

$$c = [\bar{v}_k - v(k-1) \quad \dots \quad \bar{v}_{k+N_c-1} - v(k-1) \quad v(k-1) - \underline{v}_k \quad \dots \quad v(k-1) - \underline{v}_{k+N_c-1}].$$

From (26), the above equation can be rewritten as

$$\Lambda^T \Delta V(k) \leq c(X(\Delta V(U(k))))^T. \quad (28)$$

This iterative QP algorithm is presented below:

Step 1: Initializing  $c(X(\Delta V(U^0(k))))$  within the input constraint.

Step 2: Solve QP for  $\Delta V^i(k)$  subject to  $\Lambda^T \Delta V^i(k) \leq c(X(\Delta V(U^{i-1}(k))))^T$

Step 3: Calculate  $c(X(\Delta V(U^i(k))))$

Step 4: Test if  $\Lambda^T \Delta V^i(k) \leq c(X(\Delta V(U^i(k))))^T$ , end; otherwise,  $i = i + 1$ , go to Step 2.

The above iterative QP is effective for solving the problem if the initial condition is properly chosen. Nevertheless, it has no guaranteed convergence to a feasible solution. To overcome this problem, another iterative algorithm is needed to guarantee a feasible solution over the entire prediction horizon is incorporated. This idea is previously presented by paper [7] on a input/output model. Extending it to the state-space equation is straightforward.

In order to solve the convergence problem, linear approximation is necessary. Now consider the approximation by linearization through Taylor's expansion about  $U_0$  of the nonlinear constraints (21) as

$$\Delta V = \Delta V_0 + g[U_0](\tilde{h}) + r(\tilde{h}), \quad (29)$$

where  $U_0$  is the chosen initial operating trajectory,  $\Delta V_0$  is obtained from (21) when  $U = U_0$ , the  $(N_c \times N_c)$  dimensional matrix  $g[U_0]$  represents the Jacobian matrix  $\partial \Delta V / \partial U$  in (21) at the operating trajectory, and  $r(\tilde{h})$  corresponds to the higher order terms of the approximation given by

$$r(\tilde{h}) = \frac{1}{2} \tilde{h}^T g[U_0 + \theta \tilde{h}] \tilde{h}, \quad (30)$$

where  $\theta \in [0, 1]$ ,  $\tilde{h} = U - U_0$ .

Neglect the higher order terms as

$$\Delta V = \Delta V_0 + g[U_0](U - U_0). \quad (31)$$

Equation (31) can lead to an explicitly linear function relationship between the control sequence  $U$  and  $\Delta V$ :

$$U = g^{-1}[U_0] \Delta V + U_0 - g^{-1}[U_0] \Delta V_0. \quad (32)$$

Now the optimization problem is defined as

$$J_{\min} = \min_{\Delta V} \left\{ \frac{1}{2} \Delta V^T H \Delta V + \Gamma^T \Delta V \right\}. \quad (33)$$

subject to the convex set of approximate linear constraints

$$\underline{U} \leq M \Delta V + m_0 \leq \bar{U}, \quad (34)$$

with  $M = g^{-1}[U_0]$ ,  $m_0 = U_0 - g^{-1}[U_0] \times \Delta V_0$ .

In order for the algorithm to reach the convergence to a feasible solution, the optimal problem can be rewritten as

$$\Delta V^* = \arg \min_{\Delta V} \left\{ \frac{1}{2} \Delta V^T H \Delta V + \Gamma^T \Delta V \right\} \quad (35)$$

$$\text{subject to } \underline{U} \leq M^\alpha \Delta V + m_0^\alpha \leq \bar{U}, \quad (36)$$

where  $0 < \alpha < 1$  is decreasing parameter with initial value  $\alpha = 1$ . Define  $M^\alpha$  and  $m_0^\alpha$ , respectively, by

$$\begin{aligned} M^\alpha &= (g^{-1}[U_0])^\alpha (I - (g^{-1}[U_0])^\alpha)^{-1}, \\ m_0^\alpha &= U_0 - (g^{-1}[U_0])^\alpha (I - (g^{-1}[U_0])^\alpha)^{-1} \Delta V_0. \end{aligned} \quad (37)$$

Define  $U_l^* = M^\alpha \Delta V^* + m_0^\alpha$  to be the linear control sequence and  $U_{nl}^*$  to be the nonlinear control sequence obtained from  $G[U, \Delta V^*] = 0$ . Then the major goal is to keep

$$\underline{U} \leq U_{nl}^* \leq \bar{U}. \quad (38)$$

Thus the optimization problem with the approximate constraints  $\Delta V^*$  is a feasible solution, which can be solved iteratively using QP.

This approximate constraints algorithm is presented follows.

Step 1: assure feasibility of the new operating trajectory:  $\underline{U} \leq U_i \leq \bar{U}$ . Solve for  $\Delta V_0$ ,  $M$ ,  $m_0$

Step 2:  $\alpha_{i-1} = 1$

Step 3: actualize the squeezing factor  $\alpha$ :

$$\alpha = \alpha \times \lambda. \quad (39)$$

Step 4: new linearized constraints at  $U_0$ :  $\underline{U} \leq M^\alpha \Delta V + m_0^\alpha \leq \bar{U}$ .

Step 5: Solve QP for the new linearized constraints:

$$\Delta V_{i+1}^* = \arg \min_{\Delta V} J(\Delta V). \quad (40)$$

Step 6: test if  $U_{nl}^*$  violate the original constraint. If:  $\underline{U} \leq U_{nl}^* \leq \bar{U} \Rightarrow \text{End}$ .

Step 7: let  $i = i + 1$ , go to 3.

**3.2. The Convergence Property.** With the object function (35), assume the practical input sequence constraints

$$|U| \leq 1. \quad (41)$$

Assume a given initial vector representing the operating trajectory,  $U_0$ , verifying the conditions

$$|U_0| \leq 1 - \varepsilon_1, \quad (42)$$

where  $\varepsilon_1$  is small positive scalar values,  $\varepsilon_1 > 0$ .

For any given scalar  $0 < \alpha < 1$ , the solution of the optimization problem (35) is subjected to

$$\left| U_0 + (g^{-1}[U_0])^\alpha (I - (g^{-1}[U_0])^\alpha) (\Delta V^* - \Delta V_0) \right| \leq 1. \quad (43)$$

which can be further written as

$$|\Delta V^*| \leq \left[ I - (g^{-1}[U_0])^\alpha \right] (g[U_0])^\alpha (1 - U_0) + \Delta V_0. \quad (44)$$

In the limit, as  $\alpha \rightarrow 0$ ,  $(g^{-1}[U_0])^{-\alpha} \rightarrow I$ ,  $(g[U_0])^\alpha \rightarrow I$ , the solution of the optimization problem is given by

$$\lim_{\alpha \rightarrow 0} |\Delta V^*| \leq \Delta V_0. \quad (45)$$

Due to the smooth function  $G[U, \Delta V] = 0$  and definition (42), this last result leads to

$$|U^*| = |U_0| \leq 1 - \varepsilon_1, \quad (46)$$

signifying that the resulting practical nonlinear control sequence is always within the bounds, and so a feasible solution is always guaranteed.

**3.3. The Algorithm for Constraint Optimal MPC.** The MPC algorithm, which can guarantee convergence to a feasible solution, has to deal with the constraints as the following steps:

Step 1:  $i = 0$ .

Step 2: if  $(N - i < 0)$ , go to 8.

Step 3: assure feasibility of the new operating trajectory:  $\underline{U} \leq U_i \leq \bar{U}$ .

Step 4: solve QP at  $U_i$ :

$$\Delta V_{i+1}^* = \arg \min_{\Delta V} J(\Delta V) \quad (47)$$

$$\text{subject to } \Lambda^T \Delta V_{i+1}^* \leq c(X(\Delta V(U_i)))^T. \quad (48)$$

Step 5: test if  $\Delta V_{i+1}^*$  violates the constraints: if  $\Lambda^T \Delta V \leq c(X(\Delta V(U_{i+1})))^T \Rightarrow \text{END}$

where  $U_{i+1} = U_{nl}^*$  is nonlinear control sequence, obtained from  $G[U_{nl}^*, \Delta V_{i+1}^*] = 0$ .

Step 6:  $i = i + 1$ . Go to 2

Step 7:  $U_0 = U_i$ ,  $\Delta V_0 = \Delta V_N^*$ . Solve for  $M, m_0$ .

Step 8:  $\alpha = 1$ .

Step 9: actualize the squeezing factor  $\alpha$ :

$$\alpha = \alpha \times \lambda. \quad (49)$$

Step 10: new linearized constraints at  $U_0$ :  $\underline{U} \leq M^\alpha \Delta V + m_0^\alpha \leq \bar{U}$

Step 11: Solve QP for the new linearized constraints:

$$\Delta V_{i+1}^* = \arg \min_{\Delta V} J(\Delta V). \quad (50)$$

Step 12: test if  $U_{nl}^*$  violate the original constraint. If:  $\underline{U} \leq U_{nl}^* \leq \bar{U} \Rightarrow \text{End}$

Step 13: let  $i = i + 1$ . Go to 9.

The total algorithm performance is quite related to the tuning of two parameters:

$N$ : the number of iterations of the QP algorithm;

$\lambda$ : the squeezing parameter of the convergence guaranteed algorithm.

The convergence speed of this iterative algorithm can be set using the decreasing rate  $\alpha$ , defined here through parameter  $0 < \lambda < 1$ .

Choosing the initial value  $U_0$  is a key issue in solving the optimal problem, since it is quite related to the resulting computation burden. Suppose that the final input sequence at instant  $k$  can be expressed as  $U_{nl}^* = [u(0) \ u(1) \ \cdots \ u(N_c - 2)]^T$ . Then, the initial value  $U_0$  at instant  $k + 1$  should be chosen as  $U_0 = [u(1) \ \cdots \ u(N_c - 2) \ 0]^T$ ; for example, the first  $N_c - 1$  factors are taken directly from the prediction of  $U_{nl}^*$ .

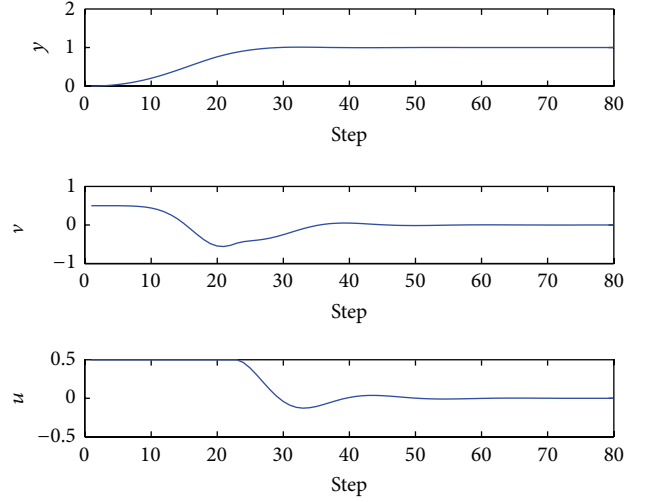


FIGURE 1: The closed-loop response for a step signal under the proposed IOFL MPC.

## 4. Case Studies

Two examples are presented to illustrate the implementation and the performance of the proposed nonlinear MPC. In example 1, a numerical state-space equation is used, and in example 2, the control problem of CSTR is considered.

**4.1. Numerical Example.** Assume the state-space equations to be

$$\begin{aligned} \dot{\bar{x}}_1 &= \bar{x}_2, \\ \dot{\bar{x}}_2 &= -3\bar{x}_1^2 \bar{x}_2 - \bar{x}_1^3 \sin(\bar{x}_2) + u, \\ y &= \bar{x}_1. \end{aligned} \quad (51)$$

The input constraint is  $-0.5 \leq u \leq 0.5$ .

Using (4), the static feedback law:  $v = u + (-3x_1^2 x_2 - x_1^3 \sin x_2)$

The resulting feedback linearization linear system is

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v, \\ y &= x_1. \end{aligned} \quad (52)$$

Choose the predictive horizon  $N_p = 8$ , the control horizon  $N_c = 5$ , and the sampling time  $\tau = 1$  s. Figure 1 shows the system response for a step signal using the proposed IOFL MPC method. The algorithm guarantees that the process input does not violate the constraints. The system reaches stability in 3 seconds without overshoot. Figure 2 shows the number of iterations at each optimization step. From Figure 2, the process needs 65 iterations at the beginning but soon reduces the iteration and finally reaches the convergence using only one iteration.

Simulations were then repeated under different predictive horizons. Table 1 lists the comparison of the control performance. From Table 1, the computing burden increases when



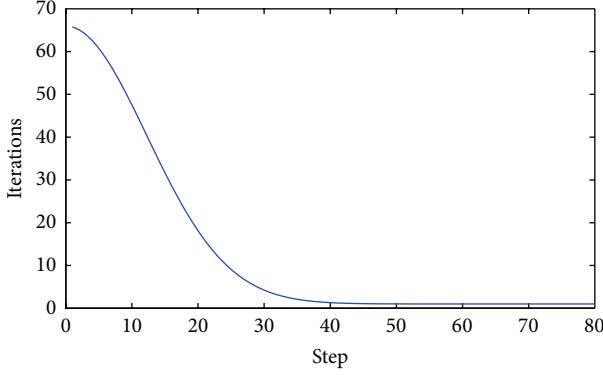


FIGURE 2: The number of iterations at each optimization step.

TABLE 1: The comparison of control performance at different prediction horizon.

Prediction horizon	Computing burden (CPU time)	Sum squared error
8	1.0469	11.4042
10	1.1875	11.4036
12	1.2813	11.4032

the prediction horizon increases, meanwhile the tracking property improves with the increment of the prediction horizon.

**4.2. The Control of a First-Order CSTR.** CSTRs are typical chemical processes representing a variety of complex industry systems. The CSTR problem discussed here represents a first-order, irreversible, exothermic kinetics reaction, which can be described by the following equation [12]:

$$\begin{aligned}
 \frac{dx_1}{dt} &= -x_1 + D_\alpha \cdot \exp\left(\frac{x_2}{1 + x_2/\gamma}\right), \\
 \frac{dx_2}{dt} &= -x_2(1 + \beta) + H \cdot D_\alpha \cdot (1 - x_1) \cdot \exp\left(\frac{x_2}{1 + x_2/\gamma}\right) \\
 &\quad + \beta u, \\
 y &= x_1,
 \end{aligned} \tag{53}$$

where the two state variables  $x_1$  and  $x_2$  are the reactor temperature and the normalized reactant concentration. The control variable  $u$  is the normalized cooling water temperature.  $D_\alpha$ ,  $\beta$ , and  $\gamma$  are the constants of the system.

The CSTR control objectives contain two aspects: set point tracking and regulation of reactant concentration under the perturbation of feed water temperature.

With the static feedback law,

$$v = au + b, \tag{54}$$

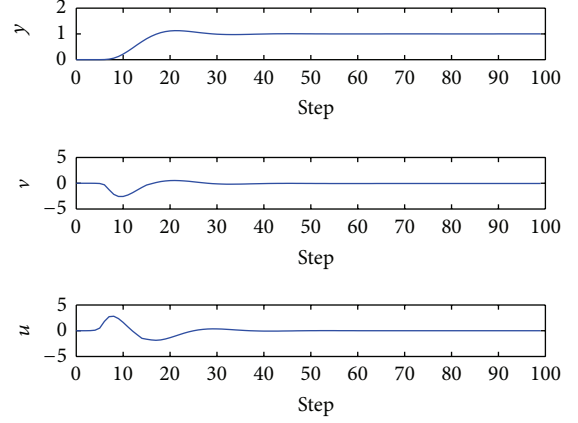


FIGURE 3: The system output and the control under step changes in the reference signal.

where

$$\begin{aligned}
 a &= \beta(\xi_1 + \xi_2) \cdot \left(1 - \left(\frac{1}{r}\right) \ln\left(\frac{(\xi_1 + \xi_2)}{D_\alpha}\right)\right)^2, \\
 b &= \xi_1 - (\xi_1 + \xi_2) + (\xi_1 + \xi_2) \\
 &\quad \cdot \left[ \frac{\ln((\xi_1 + \xi_2)/D_\alpha)}{1 - (1/r) \ln((\xi_1 + \xi_2)/D_\alpha)} (1 + \beta) \right. \\
 &\quad \left. + H \cdot (1 - \xi_1)(\xi_1 + \xi_2) \right].
 \end{aligned} \tag{55}$$

The resulting feedback linearization linear system is

$$\begin{aligned}
 \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v, \\
 y &= [1 \ 0] \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}.
 \end{aligned} \tag{56}$$

Suppose the constraint to be  $-3 \leq u \leq 3$ . Let  $r, D_\alpha, H = 1$ ,  $\beta = 0.01$ , the predictive horizon  $N_p = 10$ , the control horizon  $N_c = 5$ , and the sampling time  $\tau = 1$  s. Figure 3 shows the system output and the control under step changes in the reference signal.

From the simulation result, the manipulated variable  $u$  does not violate the defined constraint. The system reaches stability in 30 steps without obvious overshoot.

As mentioned before, there exist two general approaches for NMPC. One is to use the SQP method. Another is to approximate the first constraints as the constraints over the entire control horizon. The proposed method is then compared with these two methods, considering this CSTR plant. The simulation results using the three predictive control schemes for  $N_c = 5$  are given in Figure 4, adopting the best  $N$  and  $\lambda$  for each  $N_p$  in the proposed IOFL NMPC.

Computation burden is one of the most concerning problems in CSTR real-time control. In using MPC, though choosing larger predictive horizon can improve system performance, it can, meantime, result in the increment of

TABLE 2: Left: CPU time of the entire simulation. Right: sum squared output error (SSE) in the simulation.

Predictive horizon	CPU time			SSE		
	One-step constraints	IOFL MPC	SQP	One-step constraints	IOFL MPC	SQP
10	0.1563	0.2011	0.3054	11.4181	11.4045	11.4042
12	0.1719	0.2407	0.3602	11.4180	11.4034	11.4031
14	0.2031	0.2795	0.4415	11.4177	11.4026	11.4018
16	0.2614	0.3306	0.5728	11.4130	11.4019	11.4003

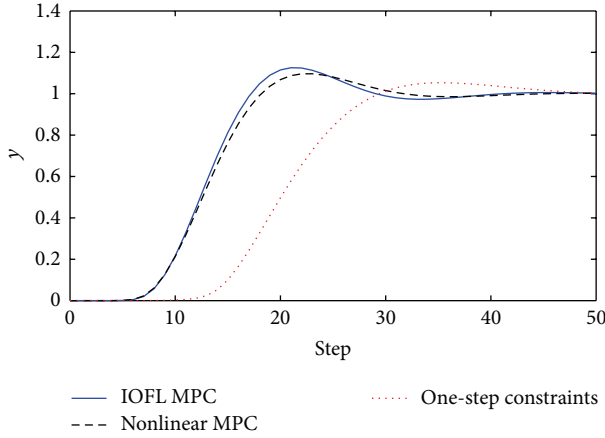


FIGURE 4: The comparison of the three responses.

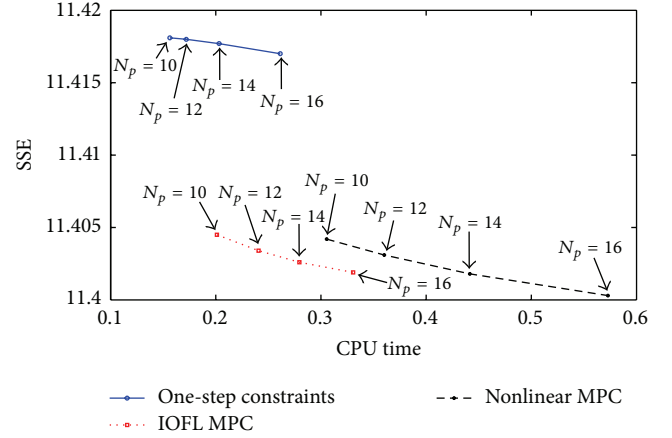


FIGURE 5: The comparison of control performance.

computation burden, especially in nonlinear system. For further comparison, the controller performance is analysed with respect to two variables, for example, sum squared output error (SSE) and the relative optimization time for the simulation. Table 2 shows the performance of each controller configuration for various prediction horizons. The three methods, namely, the one-step constraints, the MPC with IOFL and the nonlinear MPC, can all guarantee a feasible control solution over the complete prediction horizon. The comparison of the three shows a similar trend performance in terms of the system tracking error, although the one-step constraint is less computational time demanding, as shown in Figure 5. With the increase of the prediction horizon, the time increase in nonlinear MPC is obvious, while the time increase in one-step constraints and MPC with IOFL optimization is quite minor. This is due to the efficient iterative process of the proposed IOFL MPC. The reduced computing burden for the one-step constraints method is obvious, since only the first time-step-ahead inputs are constrained, leaving the other  $N_p - 1$  inputs unconstrained. The shortcoming is that the resulting optimum may not be feasible. From Figure 5, the total trade-off between the computational demands and the gain in optimality is obviously favourable to the proposed MPC with IOFL.

## 5. Conclusions

MPC has been widely used in industrial process. One limitation of MPC is that it is mostly based on a linear model.

The performance will deteriorate when the current output is relatively far away from the operating point at which the linear control model was generated, especially in the presence of constraints. The approach presented in this paper aims at combining the nonlinear differential geometry technique with existing MPC technique to avoid nonlinear optimization when using a nonlinear model of the process. Considering the continuous-time system, the detailed routine for reaching a feasible solution through iterative QP optimization is presented in this paper. The initial value is properly chosen to greatly reduce the computation burden. Comparing results on the CSTR considering, both the SSE and the relative optimization time needed for completing the simulation have also been addressed in detail. The advantage is that it avoids a direct nonlinear optimization on a nonlinear model of the process and incorporates the linear optimization technique in the controller design.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 60974051, 61273144, and 61203041, the Natural Science Foundation of Beijing under Grant 4122071, the Chinese National Postdoctor Science Foundation under Grants 2011M500217 and 2012T50036, the Doctoral Fund of Ministry of Education of China under Grant 20120036120013, and the Fundamental Research Funds for the Central Universities 13XS09.



## References

- [1] W. Liuping, *Model Predictive Control System Design and Implementation Using MATLAB*, Springer, New York, NY, USA, 2009.
- [2] T. Zou, "Offset-free strategy by double-layered linear model predictive control," *Journal of Applied Mathematics*, vol. 2012, Article ID 808327, 14 pages, 2012.
- [3] A. Zheng, "A computationally efficient nonlinear MPC algorithm," in *Proceedings of the American Control Conference*, pp. 1623–1627, Albuquerque, NM, USA, June 1997.
- [4] M. J. Kurtz, G.-Y. Zhu, and M. A. Henson, "Constrained output feedback control of a multivariable polymerization reactor," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 1, pp. 87–97, 2000.
- [5] A. Jadbabaie and J. Hauser, "On the stability of receding horizon control with a general terminal cost," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 674–678, 2005.
- [6] X. J. Liu, P. Guan, and C. W. Chan, "Nonlinear multivariable power plant coordinate control by constrained predictive scheme," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 5, pp. 1116–1125, 2010.
- [7] M. A. Botto, T. J. J. Van Den Boom, A. Krijgsman, and J. Sá Da Costa, "Predictive control based on neural network models with I/O feedback linearization," *International Journal of Control*, vol. 72, no. 17, pp. 1538–1554, 1999.
- [8] J. Deng, V. Becerra, and R. Stobart, "Input constraints handling in an MPC/feedback linearization scheme," *International Journal of Applied Mathematics and Computer Science*, vol. 19, no. 2, pp. 219–232, 2009.
- [9] M. A. Henson and D. E. Seborg, "Input-output linearization of general nonlinear processes," *AIChE Journal*, vol. 36, no. 11, pp. 1753–1757, 1990.
- [10] X.-B. Kong, Y.-J. Chen, and X.-J. Liu, "Nonlinear model predictive control with input-output linearization," in *Proceedings of the 24th Chinese Control and Decision Conference (CCDC '12)*, pp. 688–693, Taiyuan, China, May 2012.
- [11] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, Upper Saddle River, NJ, USA, 2002.
- [12] T. D. Knapp, H. M. Budman, and G. Broderick, "Adaptive control of a CSTR with a neural network model," *Journal of Process Control*, vol. 11, no. 1, pp. 53–68, 2001.

