International Journal of Innovative Computing, Information and Control
Volume **8**, Number **10(B)**, October **2012**

pp. 7083–7093

NONLINEAR RECEDING HORIZON CONTROL OF QUADRUPLE-TANK SYSTEM AND REAL-TIME IMPLEMENTATION

Miaomiao Ma[1,*], Hong Chen[2], Rolf Findeisen[3] and Frank Allgöwer[3]

[1]School of Control and Computer Engineering
North China Electric Power University
No. 2, Beinong Road, Huilongguan, Changping District, Beijing 102206, P. R. China
[*]Corresponding author: mamm@ncepu.edu.cn

[2]Department of Control Science and Engineering
Jilin University
No. 5988, Renmin Str., Campus NanLing, Changchun 130025, P. R. China

[3]Institute for Systems Theory and Automatic Control
University of Stuttgart
Pfaffenwaldring 970550 Stuttgart, Germany

Received August 2011; revised December 2011

Abstract. *In this paper, nonlinear receding horizon control (NMPC) is implemented on a laboratory quadruple-tank system, which is a nonlinear multi-variable process with state constraints as well as input constraints. A fast numerical algorithm called C/GMRES is employed to implement nonlinear receding horizon control of quadruple-tank system, in which the continuation method is combined with a fast algorithm for linear equations instead of the Riccati differential equation. Analysis and simulation results show that the computation time is significantly reduced and nonlinear receding horizon control can be implemented successfully in real time.*
Keywords: Quadruple-tank system, Nonlinear receding horizon control, Real-time optimization, GMRES

1. **Introduction.** Nonlinear moving horizon control, mostly referred to as nonlinear model predictive control (NMPC), has become an attractive feedback strategy for nonlinear multi-variable systems subject to input and state constraints, while application can be found not only in traditional fields [1], but also in emerging environments (see [2-7] for some new reports). Over the last few years, also academic research of NMPC has achieved significant progresses. By introducing the so-called stability constraint and appropriately computing the terminal penalty, nominal stability issues are well-addressed; for complete surveys we refer, e.g., to [8-11]. Nevertheless, because of the computational requirements of the optimizations associated with NMPC, it can be applied in the chemical industry with slow dynamics [12,13], where the sampling period is sufficiently large, e.g., several tens of seconds or longer, and an iterative optimization method can be executed to solve the optimization problem within the sampling period. However, an iterative optimization method is computationally expensive and is not suitable for fast systems controlled with a sampling period in the order of milliseconds. Although an approximate algorithm of nonlinear receding horizon control can be obtained explicitly through use of the Taylor expansion [14], the length of the horizon and the form of the performance index are restricted in such an algorithm. Therefore, efficient and reliable numerical algorithms are necessary to implement nonlinear receding control.

A real-time algorithm of nonlinear receding horizon control, in which the continuation method is combined with a fast algorithm for linear equations instead of the Riccati differential equation, is proposed in [15]. The formulation and implementation of the proposed algorithm are substantially different from the previous algorithm, although both algorithms are based on the continuation method. The problem is discretized over the horizon first, and then a differential equation to update the sequence of control inputs is obtained through use of the continuation method. Since that differential equation involves a large linear equation, we employ the GMRES (Generalized Minimum Residual) method [16] to solve the linear equation. To demonstrate the fast algorithm, we take the quadruple-tank system that is a nonlinear multi-variable system with input and state constraints as an example in this paper. The quadruple-tank process can easily be built by using two double-tank processes, which are standard processes in many control laboratories [17-19]. The setup is thus simple, but still the process can illustrate several interesting multivariable phenomena. In this paper, we apply the real-time algorithm called C/GMRES to the quadruple-tank system in order to examine the computation time. Simulation results show that nonlinear receding horizon control is possible in real time for the highly nonlinear system with the C/GMRES algorithm.

This paper is organized as follows. Section 2 describes the quadruple-tank system and formulates the control problem. In Section 3, we briefly review the basic principle of NMPC and design nonlinear receding horizon controller for the quadruple-tank system. The real-time algorithm called C/GMRES is stated in Section 4, in which the continuation method is combined with GMRES. Analysis and simulation results are given and discussed in Section 5. Section 6 presents the conclusions of the paper.

2. **Quadruple-Tank System.** A schematic diagram of the quadruple-tank system is shown in Figure 1. The quadruple-tank system is based on a laboratory experimental

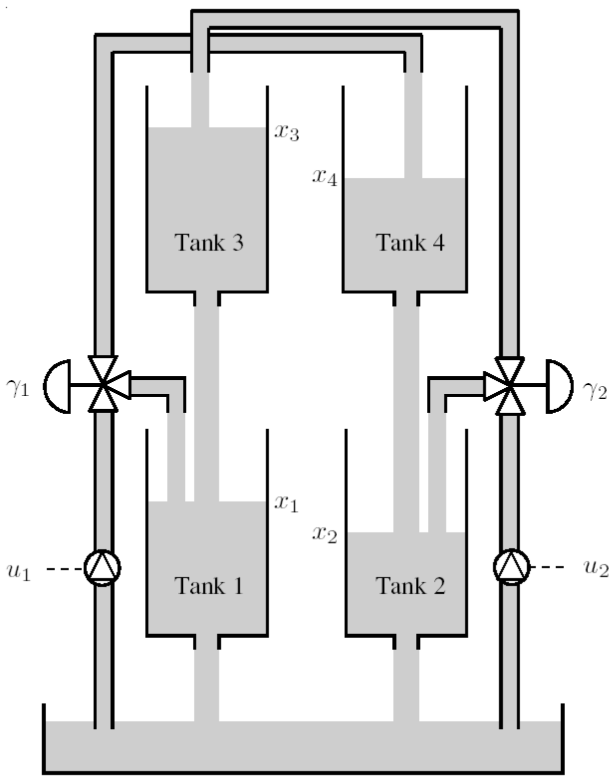

FIGURE 1. Schematic diagram of the quadruple-tank system

setup, which is used in control education at the Institute for Systems Theory and Automatic Control (IST), at the University of Stuttgart. The system is built up of four interconnected water tanks. The two inputs of the tank system are the the flow rates u_1 and u_2. The control variables are the fill levels x_1 and x_2 of the lower two tanks. Each input flow rate is separated by a valve into two flow rates. The valve position parameters $\gamma_1, \gamma_2 \in (0, 1)$ specify the flow partitioning of water up to the upper and the lower tanks. The flow of water to tank 1 is $\gamma_1 u_1$ and the flow to tank 4 is $(1 - \gamma_1)u_1$ and respective to tank 2 and tank 3. Mass balances and Bernoulli's law yield under the assumption of frictionless flow and the neglect of the unsteady parts to the nonlinear system equation [20]

$$\dot{x}_1 = -\frac{a_1}{A_1}\sqrt{2gx_1} + \frac{a_3}{A_1}\sqrt{2gx_3} + \frac{\gamma_1}{A_1}u_1, \tag{1a}$$

$$\dot{x}_2 = -\frac{a_2}{A_2}\sqrt{2gx_2} + \frac{a_4}{A_2}\sqrt{2gx_4} + \frac{\gamma_2}{A_2}u_2, \tag{1b}$$

$$\dot{x}_3 = -\frac{a_3}{A_3}\sqrt{2gx_3} + \frac{(1-\gamma_2)}{A_3}u_2, \tag{1c}$$

$$\dot{x}_4 = -\frac{a_4}{A_4}\sqrt{2gx_4} + \frac{(1-\gamma_1)}{A_4}u_1, \tag{1d}$$

where x_i denotes the fill level of tank i, u_i the flow rate of pump i, A_i the cross-section of tank i, a_i the cross-section of the outlet hole of tank i, g the acceleration due to gravity (981cm/s^2). The fluid flow is not steady, unless the system is in steady state, but nevertheless the unsteady part of the Bernoulli equation is neglected to get an easier model. The measured parameters A_i and the identified parameters a_i of the four tank system are given in the Table 1 [21].

Table 1. Parameters of the four tank system

	A_i	a_i
$i = 1$	50.27 cm^2	0.233 cm^2
$i = 2$	50.27 cm^2	0.242 cm^2
$i = 3$	28.27 cm^2	0.127 cm^2
$i = 4$	28.27 cm^2	0.127 cm^2

The objective to control the water levels of the two lower tanks around the setpoints $x_{1f} = 14$cm and $x_{2f} = 14$cm with the valve parameters $\gamma_1 = \gamma_2 = 0.4$. This results in the setpoint $\boldsymbol{x}_f = [14\text{cm } 14\text{cm } 14.2\text{cm } 21.3\text{cm}]^T$ for the state and $\boldsymbol{u}_f = [43.4\text{ml/s } 35.4\text{ml/s}]^T$ for the control inputs. The control inputs of the quadruple-tank system are constrained as $u_b \le u_1, u_2 \le u_a$, where $u_b = 0$ml/s and $u_a = 65$ml/s [21].

3. **Nonlinear Receding Horizon Control.** In this section, the nonlinear receding horizon control problem is briefly summarized. We consider a general nonlinear system governed by the state equation

$$\dot{x}(t) = f(x(t), u(t), p(t)), \tag{2}$$

subject to the input and state constraints $u \in \mathcal{U}$, $x \in \mathcal{X}$, where $x \in \mathbb{R}^n$ is the system state and $u \in \mathbb{R}^{m_u}$ the control input, and $p(t) \in \mathbb{R}^{m_p}$ the vector of given time-dependent parameters, respectively. Furthermore, it is assumed that the equilibrium point of the system (2) is at the origin, i.e., the vector field f satisfies $f(0, 0, 0) = 0$. The set $\mathcal{X}$ is a closed subset of $\mathbb{R}^n$ and the set $\mathcal{U}$ is a compact subset of $\mathbb{R}^m$, both containing the origin. Suppose that the full state x of the system (2) can be measured. Then in NMPC the control input applied to the system (2) at each time t is given by the repeated solution of the finite horizon optimal control problem:

$$\min_{u(\cdot)} J(x(t), u(\cdot), p(t))$$

subject to

$$\dot{x}(t') = f(x(t'), u(t'), p(t')), \quad x(t; x(t), t) = x(t) \tag{3a}$$

$$u(t') \in U, \quad t' \in [t, t+T] \tag{3b}$$

$$x(\tau; x(t), t) \in X, \quad t' \in [t, t+T] \tag{3c}$$

$$x(t+T; x(t), t) \in \mathcal{E}. \tag{3d}$$

We introduce a finite horizon objective functional for the open-loop optimal control problem at time t in the framework of nonlinear MPC by

$$J = \varphi(x^u(t+T; t, x(t), p(t+T)) + \int_t^{t+T} L(x^u(t'; t, x(t)), u(t'), p(t'))dt' \tag{4}$$

where $x^u(\tau; t, x(t))(x \leq t' \leq t+T)$ denotes the state trajectory by the input function u starting from $x(t)$ at time t, T is the control/prediction horizon and L is the stage cost which is in general a positive definite funciton in the variables x, u. The optimal control u_{opt} is determined as a function over the horizon and also depends on t and $x(t)$ as $u_{opt}(t'; t, x(t))(t \leq t' \leq t+T)$. However, the actual input to system is given only by the value of u_{opt} at time t, that is, $u(t) = u_{opt}(t; t, x(t))$, which results in a state feedback control law. Since u_{opt} is updated at each time, the predicted trajectory $x^{u_{opt}}(t'; t, x(t))(t \leq t' \leq t+T)$ is not necessarily identical to the actual trajectory of the closed-loop system. The terminal region $\mathcal{E}$ and the terminal penalty term φ of the finite horizon optimal problem (4) are used to enforce closed loop stability and to increase the performance in NMPC. A wide variety of approaches have been developed in the literature to achieve asymptotic closed loop stability [8,11]. All these approach are based, implicitly or explicitly, on three ingredient: a terminal cost φ, a terminal region $\mathcal{E}$ and a locally stabilizing control law $\psi = Kx$.

In the so called quasi-infinite horizon NMPC approach [22] the terminal cost and the terminal region are chosen as

$$\varphi(x) = x^T P x \tag{5}$$

$$\mathcal{E} = \{x \in \mathbb{R}^n | x^T P x \leq \alpha\} \tag{6}$$

with the state feedback $\psi = Kx$. The terminal cost $\varphi(x)$, the terminal region $\mathcal{E}$, and the linear feedback $\psi = Kx$ are calculated off-line by a procedure described in [22,23].

4. **Continuation/GMRES Method for Fast Algorithm of NMPC.** The receding horizon control problem is essentially a family of finite horizon optimal control problems along a fictitious time τ as follows:

$$\min J = \varphi(x^*(T, t)) + \int_0^T L(x^*(\tau, t), u^*(\tau, t))d\tau \tag{7}$$

subject to

$$\dot{x}^*_\tau(\tau, t) = f(x^*(\tau, t), u^*(\tau, t)), \tag{8a}$$

$$x^*(0, t) = x(t), \tag{8b}$$

$$C(x^*(\tau, t), u^*(\tau, t)) = 0. \tag{8c}$$

The new state vector $x^*(\tau, t)$ denotes the trajectory along the τ axis starting from $x(t)$ at $\tau = 0$. The optimal control input $u^*(\tau, t)$ is determined on the τ axis as the solution of the finite horizon optimal control problem for each t, and the actual control input is given by $u(t) = u^*(0, t)$. The horizon T is a function of time, $T = T(t)$, in general, as is explained later.

Equation constraints are also imposed in the general as Equation (8c) where C is an m_c dimensional vector-valued function. All functions are assumed to be differentiable as many as necessary. In the case of an inequality constraint, we can convert it to the equation constraint by introducing a penalty function method or some heuristic modification of the problem, as have also be discussed in [24].

Real-time algorithm is a major issue for nonlinear receding horizon control. Now we briefly review the algorithm called C/GMRES [15], which will be employed to solve the optimal control problem at each time. First, we divide the horizon into N steps and discretize the optimal control problem with the forward differences as follows:

$$x^*_{i+1}(t) = x^*_i(t) + f(x^*_i(t), u^*_i(t))\Delta\tau, \tag{9}$$

$$x^*_0(t) = x(t), \tag{10}$$

$$C(x^*(t), u^*_i(t)) = 0, \tag{11}$$

$$J = \varphi(x^*_N(t)) + \sum_{i=0}^{N-1} L(x^*_i(t), u^*_i(t))\Delta\tau, \tag{12}$$

where $\Delta\tau := T/N$, and $x^*_i(t)$ corresponds to the discrete-time trajectory starting from $x(t)$ at $i = 0$. Since the horizon length T depends on time t in general, so does $\Delta\tau$. Note that only the problem over the horizon is discretized, and the dependence of $x^*_i(t)$ and $u^*_i(t)$ on the actual time t remains continuous at the stage of problem formulation. That is, the discretized problem is supposed to be solved at each continuous time t, although the time t will also be discretized eventually in the actual implementation. Given the initial state of the discretized problem, $x^*_0(t) = x(t)$, the control input sequence $\{u^*_i(t)\}_{i=0}^{N-1}$ is optimized at each time t. The actual control input to the system is given by $u(t) = u^*_0(t)$.

Let H denote the Hamiltonian defined by

$$H(x, \lambda, u, \mu) := L(x, u) + \lambda^T f(x, u) + \mu^T C(x, u),$$

where $\lambda \in \mathbb{R}^n$ denotes the costate, and $\mu \in \mathbb{R}^{m_c}$ denotes the Lagrange multiplier associated with the equality constraint. The first-order necessary conditions for the sequences of optimal control $\{u^*_i(t)\}_{i=0}^{N-1}$, multiplier $\{\mu^*_i(t)\}_{i=0}^{N-1}$ and costate $\{\lambda^*_i(t)\}_{i=0}^{N-1}$ are obtain by the calculus of variation [25] as

$$H_u(x^*_i(t), \lambda^*_{i+1}(t), u^*_i(t), \mu^*_i(t)) = 0, \tag{13}$$

$$\lambda^*_i(t) = \lambda^*_{i+1}(t) + H^T_x(x^*_i(t), \lambda^*_{i+1}(t), u^*_i(t), \mu^*_i(t))\Delta\tau, \tag{14}$$

$$\lambda^*_N(t) = \varphi^T_x(x^*_N(t)). \tag{15}$$

The sequence of the optimal control $\{u^*_i(t)\}_{i=0}^{N-1}$ and the multiplier $\{\mu^*_i(t)\}_{i=0}^{N-1}$ have to satisfy (9)-(11), (13)-(15), which defines a two-point boundary-value problem (TPBVP) for the discretized optimal control problem. It should be noted that the TPBVP for the discretized problem is identical to a finite difference approximation of the TPBVP for the original continuous-time problem. Therefore, the solution of the discretized problem converges to the solution of the continuous-time problem as $N \to \infty$ under mild conditions [26]. Other higher order discretization schemes can also be employed at the expense of simplicity and computational time.

We define a vector of the inputs and the multipliers as

$$U(t) := \left\{u_0^{*T}(t), \mu_0^{*T}(t), u_1^{*T}(t), \mu_1^{*T}(t), \ldots, u_{N-1}^{*T}(t), \mu_{N-1}^{*T}(t)\right\} \in \mathbb{R}^{(m+m_c)N},$$

and we also define a projection P_0: $\mathbb{R}^{(m+m_c)N} \to \mathbb{R}^m$ as

$$P_0(U(t)) := u_0^*(t).$$

For a given $U(T)$ and $x(t)$, $\{x_i^*(t)\}_{i=0}^N$ are calculated recursively by Equations (9) and (10), and then, $\{\lambda_i^*(t)\}_{i=0}^N$ are also calculated recursively by from $i = N$ to $i = 0$ by Equations (14) and (15). Since $x_i^*(t)$ and $\lambda_i^*(t)$ are determined by $x(t)$ and $U(t)$ through Equations (9), (10), (14) and (15). Equations (11) and (13) can be regarded as an equation defined as

$$F(U(t), x(t), t) := \begin{bmatrix} H_u^T(x_0^*(t), \lambda_1^*(t), u_0^*(t), \mu_0^*(t)) \\ C(x_0^*(t), u_0^*(t)) \\ \vdots \\ H_u^T(x_{N-1}^*(t), \lambda_N^*(t), u_{N-1}^*(t), \mu_{N-1}^*(t)) \\ C(x_{N-1}^*(t), u_{N-1}^*(t)) \end{bmatrix} = 0. \tag{16}$$

The equation also depend on time t through T and $\Delta\tau$ in general. If the equation is solved with respect to $U(t)$ for the measured $x(t)$ at each time t, then the control input $u(t) = P_0(U(t))$ is determined.

Since such an iterative method as Newton method is numerically expensive to solve Equation (16) in real time, we employ another equivalent condition to trace the time-dependent solution as follows:

$$\begin{aligned} &\dot{F}(U(t), x(t), t) = -\zeta F(U(t), x(t), t), \quad \zeta > 0, \\ &F(U(0), x(0), 0) = 0, \end{aligned}$$

where the right-band side in the first equation is added to stabilize $F = 0$. If the Jacobian F_U is non-singular, we obtain a differential equation of $U(t)$ as

$$\dot{U} = F_U^{-1}(-\zeta F - F_x\dot{x} - F_t). \tag{17}$$

Then, $U(t)$ can be updated without any successive approximation by integrating Equation (17) in real time, which is a kind of the continuation method [27]. The derivative $F_x\dot{x} + F_t$ in Equation (17) can be approximated efficiently by a forward difference [15].

Since Equation (17) still involves numerically expensive operations to solve the linear equation associated with F_U^{-1}, we also employ generalized minimum residual method (GMRES) [16], which is a kind of Krylov subspace methods and is efficient for large-scale linear equations. Then, the real-time algorithm is summarized as follows.

Algorithm (C/GMRES)

(i) Measure the initial state $x(0)$ and find $U(0)$ analytically or numerically such that $\|F(U(0), x(0), 0)\| \leq \delta$ for sufficiently small $\delta > 0$.

(ii) With the state $x(t)$ measured, integrate in real time $\dot{U}(t)$ obtained by solving Equation (17) with GMRES. The control input $u(t)$ is given by $u(t) = P_0(U(t))$.

For example, real-time state integration of $\dot{U}(t)$ can be carried out with the Euler method as $U(t_{k+1}) := U(t_k) + \dot{U}(t_k)\Delta t$, where t_k denotes the sampling time and Δt denotes the sampling period. It should be noted that $\dot{U}(t_k)$ and the measured state $x(t_k)$ by Equation (17) and, consequently, $U(t_k)$ can be updated successively once $U(0)$ is given.

Remark 4.1. *C/GMRES solves the linear Equation (17) only once at each sampling time and, therefore, requires much less computational burden than such iterative methods as Newton's method which solves a linear equation several times to determine search* *directions. In addition, C/GMRES involves no line search, which is also a significant difference from standard optimization methods.*

A simple method for initializing $U(0)$ is to choose a time-dependent horizon $T(t)$ as a smooth function such that $T(0) = 0$ and $T(t) \to T_f (t \to \infty)$, e.g., $T(t) = T_f(1 - e^{\alpha t})$ $(T_f, \alpha > 0)$. Then, since the horizon length is zero at $t = 0$, we have $\mu_i^*(0) = u(0)$, $\mu_i^*(0) = \mu(0)$ $(i = 0, \dots, N-1)$, $x_i^*(0) = x(0)$, $\lambda_i^*(0) = \varphi_x^T(x(0))$ $(i = 0, \dots, N)$, and the initialization reduces to finding only $m+m_c$ unknown quantities, $u(0)$ and $\mu(0)$, such that

$$\left\| \begin{bmatrix} H_u^T(x(0), \varphi_x^T(x(0)), u(0), \mu(0)) \\ C(x(0), u(0)) \end{bmatrix} \right\| \leq \frac{\delta}{\sqrt{N}}.$$

Error analysis of the entire C/GMRES algorithm shows that the error $\|F\|$ is bounded under some assumptions if the parameter ζ is chosen so that $0 < \zeta\Delta t \leq \bar{\zeta}\Delta t$ for given sampling period Δt and some $\bar{\zeta}$ such that $1 \leq \bar{\zeta}\Delta t < 2$ [15].

Remark 4.2. *Although the proposed algorithm is expressed as a differential equation obtained from the continuation method, it must be discretized with respect to time for implementation, and the algorithm may fail because of discretization error. In addition, the algorithm without stability constraints does not guarantee closed loop asymptotic stability and can even destabilize a stable system.*

Remark 4.3. *The algorithm, C/GMRES, for nonlinear receding horizon control is a numerical algorithm anyway and can fail depending on the simulation conditions.*

Remark 4.4. *At the present time, there is no systematic methodology to determine those free parameters. In general, the number of grids on the horizon N is restricted by available computational power and a desirable sampling period. Then the maximum horizon length T_f is restricted according to desirable accuracy of discretization and numerical stability of the algorithm. There is also a limitation on α because the algorithm fails to trace the solution if α is so large that the horizon length T increases too rapidly. Moreover, a designer needs extensive simulation to find appropriate weights in the performance index to achieve satisfactory closed-loop response. The weights on the inputs should be chosen to be small as far as the algorithm can be executed so that the approximation is accurate. If their values are sufficiently small, they have only slight influence over the control performance. A large value of iteration number in GMRES may also be necessary for an accurate solution. An appropriate value of the parameter for stabilization of the solution, ζ would be about $1/\Delta t$.*

5. **Simulation Results.**

5.1. **NMPC of the quadruple-tank system.**

The objective to control the water levels of the two lower tanks around the setpoints $x_{1f} = 14$cm and $x_{2f} = 14$cm with the valve parameters $\gamma_1 = \gamma_2 = 0.4$. This results in the setpoint $\boldsymbol{x}_f = [14\text{cm } 14\text{cm } 14.2\text{cm } 21.3\text{cm}]^T$ for the state and $u_f = [43.4\text{ml/s } 35.4\text{ml/s}]^T$ for the control inputs. Hence, with the coordinate change $z_i = x_i - x_{if}$ and $v_i = u_i - u_{if}$ the dynamics of the quadruple-tank system (1) is

$$\dot{z}_1 = -\frac{a_1}{A_1}\sqrt{2g(z_1 + x_{1f})} + \frac{a_3}{A_1}\sqrt{2g(z_3 + x_{3f})} + \frac{\gamma_1}{A_1}(v_1 + u_{1f}), \tag{18a}$$

$$\dot{z}_2 = -\frac{a_2}{A_2}\sqrt{2g(z_2 + x_{2f})} + \frac{a_4}{A_2}\sqrt{2g(z_4 + x_{4f})} + \frac{\gamma_2}{A_2}(v_2 + u_{2f}), \tag{18b}$$

$$\dot{z}_3 = -\frac{a_3}{A_3}\sqrt{2g(z_3 + x_{3f})} + \frac{(1-\gamma_2)}{A_3}(\upsilon_2 + u_{2f}), \tag{18c}$$

$$\dot{z}_4 = -\frac{a_4}{A_4}\sqrt{2g(z_4 + x_{4f})} + \frac{(1-\gamma_1)}{A_4}(\upsilon_1 + u_{1f}). \tag{18d}$$

The stage cost of the considered control objective is

$$L(z, \upsilon) = z_1^2 + z_2^2 + 0.01(\upsilon_1^2 + \upsilon_2^2). \tag{19}$$

Furthermore, the input and state constraints arise from the pump characteristics and minimal and maximal water levels of the tanks. In particular, the input and state constraints are

$$\begin{aligned}
\upsilon_{\min} &= [-43.4\text{ml/s} \quad -35.4\text{ml/s}]^T, \\
\upsilon_{\max} &= [21.6\text{ml/s} \quad 29.6\text{ml/s}]^T, \\
z_{\min} &= [-6.5\text{cm} \ -6.5\text{cm} \ -10.7\text{cm} \ -16.8\text{cm}]^T, \\
z_{\max} &= [14\text{cm} \quad 14\text{cm} \quad 13.8\text{cm} \quad 6.7\text{cm}]^T.
\end{aligned}$$

The terminal penalty term $\varphi(x)$ and the terminal region $\mathcal{E}$ of the quasi-infinite horizon NMPC controller are calculated via the procedure described in [23] and are given by

$$P = \begin{bmatrix} 6.55 & 0 & 0 & 0 \\ 0 & 6.55 & 0 & 0 \\ 0 & 0 & 7.92 & 0 \\ 0 & 0 & 0 & 31.7 \end{bmatrix}, \quad \alpha = 792.4. \tag{20}$$

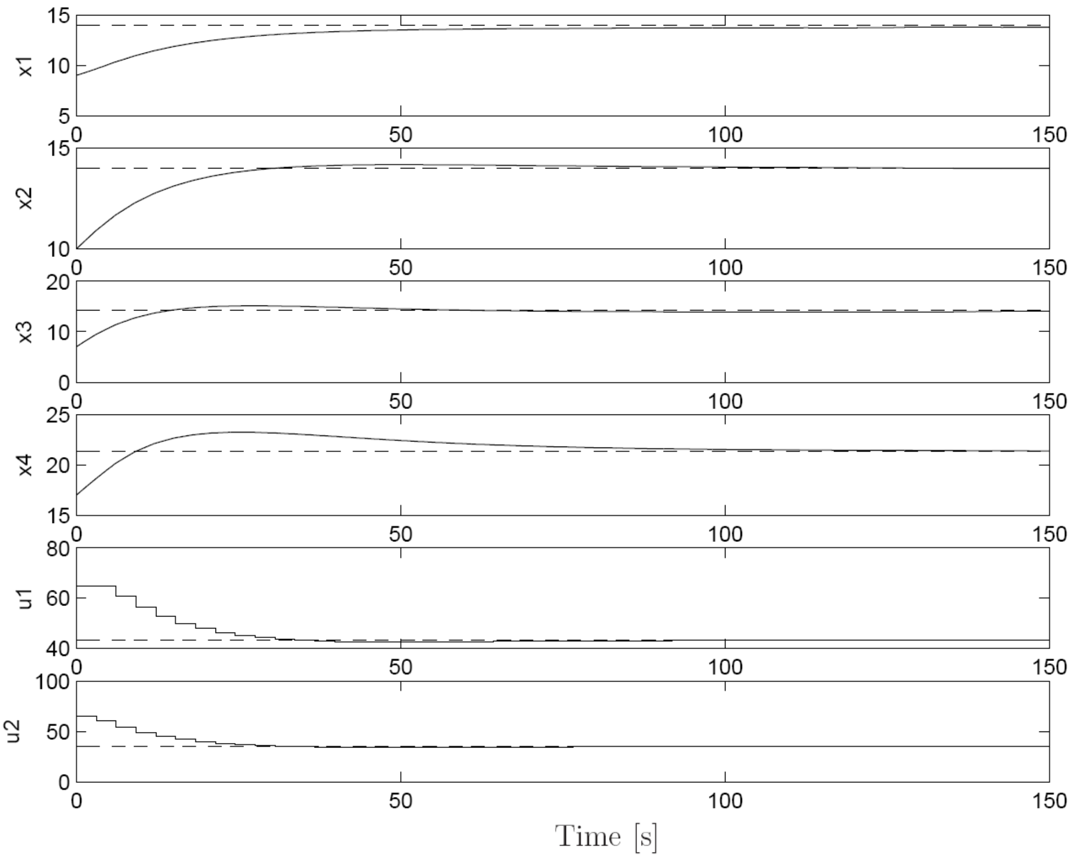

FIGURE 2. Simulation results with quasi-infinite horizon NMPC: setpoint (dash-dotted), simulation (solid)

Figure 2 shows the simulation results using the quasi-infinite horizon NMPC controller with a control/prediction horizon of $T = 15$s and the sample time δ is chosen as 3s. It can be seen that the quasi-infinite horizon NMPC controller is able to stabilize the quadruple-tank system while satisfying the input and state constraints (20). For a total simulation of 150s, the elapsed CPU time is about 220s in a system with 2.1GHz Intel Core 2 Duo T657 CPU and 2G RAM, where the controller calls *fmincon* for solving the LMI optimization problem with given numerical parameters (relative accuracy = 0.01, etc.). The heavy on-line computation burden arises partially from the M-files management. It is clear that the computation times of the optimizer with the NMPC controllers takes more time to find the optimal solution than sample time, which indicates that real-time control is impossible with a sampling period of 3s.

5.2. **Real-time implementation of the quadruple-tank system.** The control inputs of the quadruple-tank system are constrained as $u_b \leq u_1, u_2 \leq u_a$, where $u_b = 0$ml/s and $u_a = 65$ml/s [21]. The inequality constraint is converted into an equality constraint by introducing a dummy input v_1 and v_2 as follows [24]:

$$C(u,v) := \begin{bmatrix} (u_1 - u_{av})^2 + v_1^2 - (u_a - u_{av})^2 \\ (u_2 - u_{av})^2 + v_2^2 - (u_a - u_{av})^2 \end{bmatrix} = 0,$$

where $u_{av} := (u_a + u_b)/2$, and u is the augmented input vector defined by $u := [u_1, u_2, v_1, v_2]^T$.

In order to evaluate the computational time of the algorithm C/GMRES, we attempt to control this system with receding horizon control. The functions in the performance index J are chosen as

$$\begin{aligned} \varphi &:= (x - x_f)^T S_f (x - x_f) \\ L &:= (x - x_f)^T Q (x - x_f) + (u - u_f)^T R (u - u_f) - p_1 \nu_1 - p_2 \nu_2 \end{aligned}$$

where p_1 and p_2 are small positive numbers for avoiding the singularities [24]. The state vector of the present system is $x = [x_1\ x_2\ x_3\ x_4]^T \in \mathbb{R}^4$, $x_f \in \mathbb{R}^4$ denote the objective state, and S_f, Q and R are weighting matrices.

The simulation program in C is generated by an automatic code generation system called AutoGenU [15]. AutoGenU is a Mathematica program for generating a simulation program for nonlinear receding horizon control. The parameters in the performance index are chosen as: $T = T_f(1 - e^{-\alpha t})$, $T_f = 15$s, $\Delta t = 0.01$s, $\alpha = 0.5$, $S_f = \text{diag}[6.55\ 6.55\ 7.92\ 31.7]$, $Q = \text{diag}[1\ 1\ 0\ 0]$, $R = \text{diag}[0.01\ 0.01]$, $p_1 = p_2 = 0.1$, and $\zeta = 1/\Delta t$. The initial state is $x_0 = [9\text{cm}\ 10\text{cm}\ 7\text{cm}\ 17\text{cm}]^T$. The simulation result with C/GMRES is shown in Figure 3. We can see that nonlinear receding horizon control steers the state sufficiently close to the objective state. The computation time by C/GMRES is less than 2.94s for a total simulation of 150s, which indicates that the computation time of NMPC with C/GMRES algorithm is significantly reduced compared with NMPC without C/GMRES algorithm and C/GMRES algorithm can be implemented successfully in real time with a sampling period of 3s [28].

6. **Conclusions.** Nonlinear receding horizon control has been applied to control the water level of the quadruple-tank system. A real-time algorithm called C/GMRES has been successfully implemented for nonlinear receding horizon control. Although asymptotic stability of the objective state is not guaranteed, nonlinear receding horizon control steers the state sufficiently close to the objective state, which shows practical effectiveness of the control method. Analysis and simulation results show that the computation time is significantly reduced and nonlinear receding horizon control can be implemented successfully in real time. Nevertheless, many unsolved questions remain. The algorithm, C/GMRES, is a numerical algorithm anyway and can fail depending on the simulation condition and

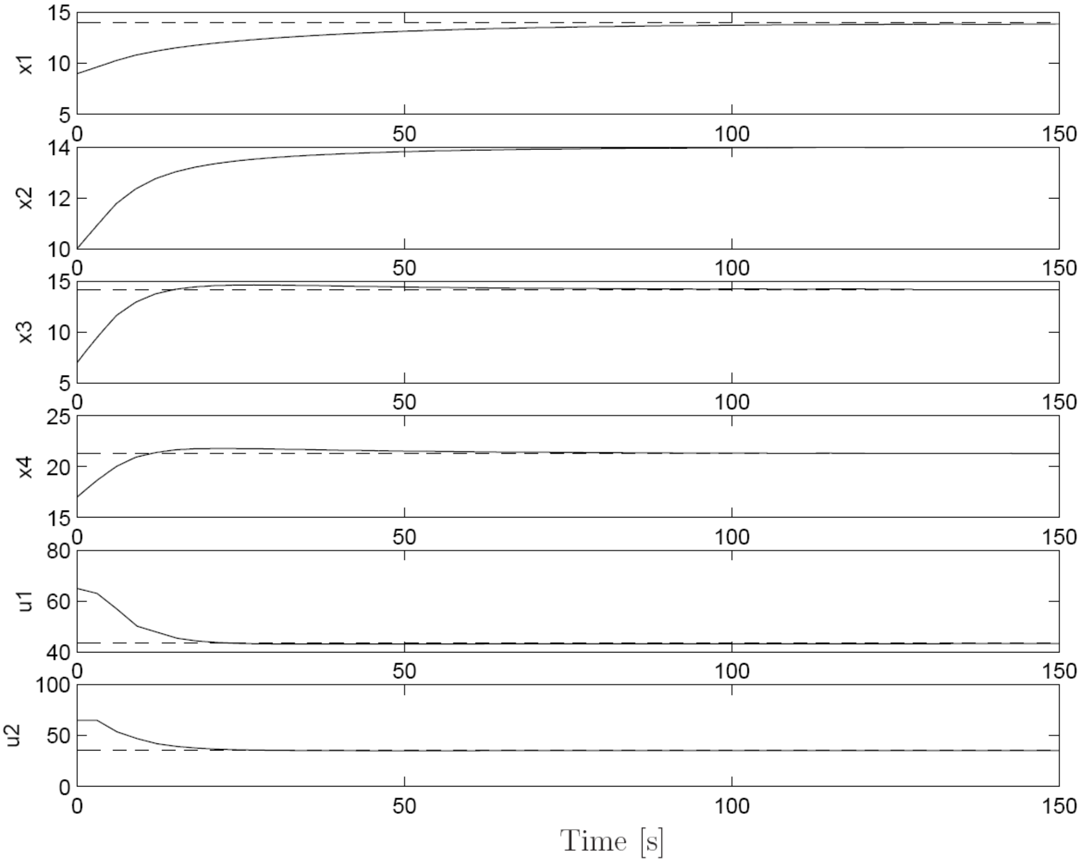

Figure 3. Simulation results with C/GMRES: setpoint (dash-dotted), simulation (solid)

there is no systematic methodology to determine those free parameters, which motivates our further efforts on the fast algorithm.

Acknowledgment. The work is financially supported by the Fundamental Research Funds for the Central Universities (No. 12MS49), the Program for Excellent Talents of Beijing (No. 2011A009005000001) and Overseas Training Project for Young Key Teachers (No. [2011]3022).

REFERENCES

[1] S. J. Qin and T. A. Badgwell, A survey of industrial model predictive control technology, *Control Engineering Practice*, vol.11, no.7, pp.733-764, 2003.
[2] A. N. Venkat, I. A. Hiskens, J. B. Rawlings and S. J. Wright, Distributed MPC strategies with application to power system automatic generation control, *IEEE Transactions on Control System Technology*, vol.16, no.6, pp.1192-1206, 2008.
[3] H. Chen, M. M. Ma, H. Wang, Z. Y. Liu and Z. X. Cai, Moving horizon H_∞ tracking control of wheeled mobile robots with actuator saturation, *IEEE Transactions on Control System Technology*, 2009.
[4] M. S. Tsoeu and T. Koetje, Unconstrained mpc tuning for prediction accuracy in networked control systems, *IEEE International Conference on Networking, Sensing and Control*, pp.474-479, 2011.
[5] L. Wang and J. Liu, Further studies on min-max MPC algorithm for LPV systems subject to input saturation, *ICIC Express Letters*, vol.4, no.3(A), pp.839-844, 2010.
[6] Y. X. Zhang and M. Liu, Model predictive control with state estimator for polytopic uncertain system with actuator saturation, *IEEE Transactions on Automatic Control*, vol.6, pp.2417-2426, 2010.
[7] O. P. Palsson, H. Madsen and H. T. Sogaard, Generalized predictive control for non-stationary systems, *Automatica*, vol.30, no.12, pp.1991-1997, 1994.

[8] F. Allgöwer, T. Badgwell, J. Qin, J. Rawlings and S. Wright, Nonlinear predictive control and moving horizon estimation – An introductory overview, in *Advances in Control, Highlights of ECC'99*, P. Frank (ed.), Springer Verlag, 1999.

[9] H. Chen and F. Allgöwer, A computationally attractive nonlinear predictive control scheme with guaranteed stability for stable systems, *Journal of Process Control*, vol.8, no.5-6, pp.475-485, 1998.

[10] A. Bemporad and M. Morari, Robust model predictive control: A survey, *Robustness in Identification and Control, Lecture Notes in Control and Information Sciences*, vol.245, pp.207-226, 1999.

[11] D. Mayne, J. Rawlings, C. Rao and P. Scokaert, Constrained model predictive control: Stability and optimality, *Automatica*, vol.36, pp.789-814, 2000.

[12] E. S. Meadows and J. B. Rawlings, Model predictive control, in *Model Predictive Control*, M. A. H. D. E. Seborg (ed.), NJ, Englewood Cliffs, Prentice-Hall, 1997.

[13] S. Qin and T. Badgwell, An overview of nonlinear model predictive control applications, in *Nonlinear Model Predictive Control*, F. Allgöwer and A. Zheng (eds.), Birkhäuser, 2000.

[14] P. Lu, Optimal predictive control of continuous nonlinear systems, *International Journal of Control*, vol.62, no.3, pp.633-649, 1995.

[15] T. Ohtsuka, Continuation/GMRES method for fast algorithem of nonlinear receding horizon control, *The 39th IEEE Conference on Decision and Control*, Australia, pp.766-771, 2000.

[16] C. T. Kelley, *Iterative Mrthod for Linear and Nonlinear Equations, Frontiers in Applied Mathetatics*, SIMA, Philadelphia, PA, 1995.

[17] K. J. Aström and M. Lundh, Lund control program combines theory with hands-on experience, *IEEE Control System Magazine*, vol.12, no.3, pp.22-30, 1993.

[18] K. J. Aström and A. B. östberg, A teaching laboratory for process control, *IEEE Control System Magazine*, vol.16, pp.37-42, 1986.

[19] D. Rosinova and M. Markech, Robust control of quadruple: Tank process, *ICIC Express Letters*, vol.3, no.4, pp.231-238, 2008.

[20] K. H. Johansson, The quadruple-tank process: A multivariable laboratory process with an adjustable zero, *IEEE Transactions on Control System Technology*, vol.8, pp.456-465, 2000.

[21] S. Huber, *Nonlinear Model Predictive Control of a Quadruple-Tank System*, Student Thesis, University of Stuttgart, 2006.

[22] H. Chen and F. Allgöwer, A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability, *Automatica*, vol.34, no.10, pp.1205-1217, 1998.

[23] W. Chen and D. J. Ballance, On attraction domain of model predictive control of nonlinear systems with input/state constraints, *Technical Report, CSC Report: CSC-99099*, 1999.

[24] H. Seguchi and T. Ohtsuka, Nonlinear receding horizon control of an underactuated hovercraft, *International Journal of Robust and Nonlinear Control*, vol.13, pp.381-398, 2003.

[25] A. Bryson and Y.-C. Ho, *Applied Optimal Control*, Ginn and Company, Waltham, Massachusetts, 1969.

[26] U. M. Ascher, R. M. M. Mattheij and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, SIAM, Philadelphia, PA, 1995.

[27] S. L. Richter and R. A. DeCarlo, Continuation methods: Theory and applications, *IEEE Transactions on Automatic Control*, vol.28, pp.660-665, 1983.

[28] T. Raff, S. Huber, Z. K. Nagy and F. Allgöwer, Nonlinear model predictive control of a four tank system: An experimental stability study, *Proc. of the 2006 IEEE International Conference on Control Applications*, Munich, Germany, pp.237-242, 2006.