Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Reduced precision solution criteria for nonlinear model predictive control with the feasibility-perturbed sequential quadratic programming algorithm*

Jiao-na WAN†, Zhi-jiang SHAO†‡, Ke-xin WANG, Xue-yi FANG, Zhi-qiang WANG, Ji-xin QIAN

(*State Key Lab of Industrial Control Technology, Institute of Industrial Control, Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China*)

†E-mail: joyce523@126.com; zjshao@iipc.zju.edu.cn

Received Dec. 18, 2010; Revision accepted Mar. 29, 2011; Crosschecked Sept. 28, 2011

Abstract: We propose a novel kind of termination criteria, reduced precision solution (RPS) criteria, for solving optimal control problems (OCPs) in nonlinear model predictive control (NMPC), which should be solved quickly for new inputs to be applied in time. Computational delay, which may destroy the closed-loop stability, usually arises while non-convex and nonlinear OCPs are solved with differential equations as the constraints. Traditional termination criteria of optimization algorithms usually involve slow convergence in the solution procedure and waste computing resources. Considering the practical demand of solution precision, RPS criteria are developed to obtain good approximate solutions with less computational cost. These include some indices to judge the degree of convergence during the optimization procedure and can stop iterating in a timely way when there is no apparent improvement of the solution. To guarantee the feasibility of iterate for the solution procedure to be terminated early, the feasibility-perturbed sequential quadratic programming (FP-SQP) algorithm is used. Simulations on the reference tracking performance of a continuously stirred tank reactor (CSTR) show that the RPS criteria efficiently reduce computation time and the adverse effect of computational delay on closed-loop stability.

Key words: Nonlinear model predictive control (NMPC), Computational delay, Termination criteria, Continuously stirred tank reactor (CSTR)
doi:10.1631/jzus.C10a0512 **Document code:** A **CLC number:** TP273

1 Introduction

Nonlinear model predictive control (NMPC) has been an appealing field of research in the last two decades (Schäfer *et al.*, 2007), but NMPC strategies are hard to apply in practice. One of the main obstacles to the extension from linear model predictive control to NMPC is that the latter requires online real-time solution of a series of nonlinear and usually non-convex optimal control problems (OCPs). This is hard to realize because in practical applications we have to deal with model-plant mismatch, disturbance, and large-scale model equations.

Much research on NMPC is based on the assumption that the OCPs can be solved instantaneously and the impact of the computational delay can generally be neglected (DeHaan and Guay, 2006). Practically, the computation time for solving OCPs is often non-negligible and can lead to a delay between the state information and the input signal implementation on the system. NMPC requires that constrained nonlinear OCPs should be solved online, but the heavy online computational burden leads to computational delay and can give rise to the deterioration of

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (Nos. 60934007 and 60974007) and the National Basic Research Program (973) of China (No. 2009CB320603)

control performance.

Since the effect of computational delay on the performance of NMPC was noted by Santos *et al.* (2001) in a laboratory reactor, the deterioration of controller stability has been studied by some researchers. Findeisen and Allgöwer (2003) reported that the computational delay could lead to drastic performance decrease or even instability of the closed-loop, and proposed an efficient approach highlighting both the computational and measurement delay in the NMPC framework. Some studies focused on the conditions in which the closed-loop stability could be guaranteed. Chen *et al.* (2000) presented a new stability condition and modified the performance index in MPC in consideration of computational delay. When an initial control profile was chosen to satisfy an inequality condition in each online optimization procedure, the nonlinear system controlled by the proposed NMPC algorithm was asymptotically stable. From the computational point of view, Chen *et al.* (2000) emphasized that solving a nonlinear dynamic optimization problem with equality constraints was highly computationally intensive, and in many cases, impossible to finish in time.

Much computationally-oriented research has been conducted in dealing with problems on computational delay. For reducing the computation time of OCPs, two of the common approaches are suboptimal termination under mild conditions with an initial feasible solution (Scokaert *et al.*, 1999), and the reduction to a single iteration per sampling interval (Diehl *et al.*, 2002; 2005). Zavala and Biegler (2009) proposed the advanced-step NMPC strategy. Based on sensitivity information, this approach uses the current control action to predict future plant state to solve the next OCP in advance while the current sampling period evolves. Wright and Tenny (2004) developed the feasibility-perturbed sequential quadratic programming (FP-SQP) algorithm to compute iterates containing both state and control components, but perturbed these to retain feasibility with respect to the constraints at every iteration and replace the line-search globalization approach with a scaled trust-region approach. This strategy has been integrated into the NMPC framework (Tenny *et al.*, 2004). Additionally, the requirement of finding a global optimization solution has been relaxed by Chen and Allgöwer (1998), Scokaert *et al.* (1999), etc. In these works, feasibility implies stability for a particular form of MPC when a controller with linear terminal state feedback control is applied. From this point of view, it is not necessary to achieve the global, or even the local optimal solution to the given OCP. This implies that it is reasonable to terminate the optimization procedure prematurely with just a feasible solution if the closed-loop stability can be preserved well. However, how to define the termination criteria in terms of deciding which iteration is sufficient as well as whether the optimization procedure should be terminated is still being studied.

In this paper, the reduced precision solution (RPS) criteria, a kind of termination criteria, are developed for solving the OCPs. RPS criteria enable the termination of the optimization procedure in a timely way and an efficient reduction of computational cost. Unlike the traditional termination criteria (Gill *et al.*, 1981), in the RPS criteria a series of indices tailored for the FP-SQP algorithm are defined to reflect the quality of the current iterate. With the RPS criteria, the optimization procedure can be terminated with a proper approximate solution. Such a rule of termination incorporates the practical requirement of solution precision and avoids iterating too much, to satisfy the given tolerance of traditional termination criteria. Therefore, computation time is shortened and computational delay can be reduced.

2 Basic theory of NMPC

In this paper, a typical ordinary differential equation (ODE) model of the controlled plant in the following form is considered:

$$\dot{\boldsymbol{x}}(t)=g(\boldsymbol{x}(t),\boldsymbol{u}(t)),\ \ \boldsymbol{x}(0)=\boldsymbol{x}_0, \tag{1}$$

where $\boldsymbol{x}$ and $\boldsymbol{u}$ are subject to state and input constraints

$$\begin{cases}\boldsymbol{u}(t)\in\mathbb{U}^m, & \forall t\geq 0,\\ \boldsymbol{x}(t)\in\mathbb{X}^n, & \forall t\geq 0.\end{cases} \tag{2}$$

Herein, $\mathbb{X}^n\subseteq\mathbb{R}^n$ is the state constraint set and $\mathbb{U}^m\subseteq\mathbb{R}^m$ is the set of feasible inputs. We use NMPC to stabilize the origin of system (1) in an optimal way while all constraints are satisfied.

When the model equations are integrated, Eq. (1) becomes the discrete model

$$\begin{cases} \boldsymbol{x}(k+1) = G(\boldsymbol{x}(k), \boldsymbol{u}(k)) \\ \quad = \boldsymbol{x}(k) + \int_{t_k}^{t_{k+1}} g(\boldsymbol{x}(\tau), \boldsymbol{u}(k), \tau)\mathrm{d}\tau, \\ \boldsymbol{x}(0) = \boldsymbol{x}_0, \end{cases} \tag{3}$$

where k represents the sampling time step. A zero-order hold is assumed for the inputs in interval $[t_k, t_{k+1}]$.

In the NMPC framework, based on given states, the current control action is obtained by repeatedly solving a series of OCPs subject to system dynamics and constraints involving states and controls. By using the current state as the initial state, an optimal control sequence is yielded by an optimization solver. The first control of the sequence is injected into the system and the calculation is repeated at the next sampling time (Findeisen and Allgöwer, 2002; Qin and Badgwell, 2003; Bock *et al.*, 2007), as shown in Fig. 1, where Δt is the sampling interval.

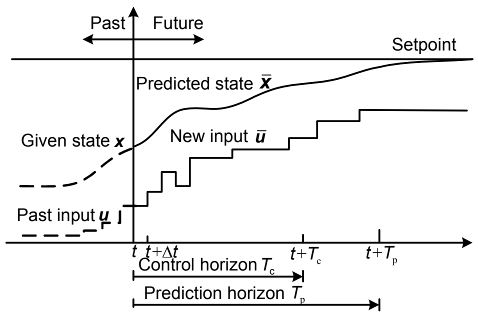

Fig. 1 Principle of nonlinear model predictive control

The OCP of system (3) minimized over the prediction horizon T_p (for convenience, control horizon and prediction horizon could be made the same) is as follows (Tenny, 2002):

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{u}} J(\boldsymbol{x},\boldsymbol{u}) &= \sum_{i=0}^{T_p-1} L(\boldsymbol{x}(k+i\,|\,k), \boldsymbol{u}(k+i\,|\,k)) + E(\boldsymbol{x}(T_p)) \\ \text{s.t.} \quad & \boldsymbol{x}(k+i+1\,|\,k) = G(\boldsymbol{x}(k+i\,|\,k), \boldsymbol{u}(k+i\,|\,k)), \\ & \quad i = 0,1,\ldots,T_p-1, \\ & \boldsymbol{x}(k+i\,|\,k) \in \mathbb{X}^n, \; i = 1,2,\ldots,T_p, \\ & \boldsymbol{u}(k+i\,|\,k) \in \mathbb{U}^m, \; i = 0,1,\ldots,T_p-1, \\ & \boldsymbol{x}(T_p) \in \mathcal{X}_f(T_p), \end{aligned} \tag{4}$$

where $L(\boldsymbol{x}(k+i|k), \boldsymbol{u}(k+i|k))$ is the stage cost, k represents the current time step, and $\boldsymbol{x}$ and $\boldsymbol{u}$ denote the sequences of vectors representing states and inputs, respectively, which are denoted as

$$\begin{cases} \boldsymbol{x} = (\boldsymbol{x}(k+1\,|\,k), \boldsymbol{x}(k+2\,|\,k), \ldots, \boldsymbol{x}(k+T_p\,|\,k)), \\ \boldsymbol{u} = (\boldsymbol{u}(k\,|\,k), \boldsymbol{u}(k+1\,|\,k), \ldots, \boldsymbol{u}(k+T_p-1\,|\,k)). \end{cases} \tag{5}$$

In these equations, $\mathcal{X}_f(T_p)$ is a compact subset of $\mathbb{R}^n$ containing the origin in its interior. Once $\boldsymbol{x}(k)$ is known, the prediction horizon is shifted forward by one sampling interval and the OCP in form of Eq. (4) is solved to obtain $\boldsymbol{u}(k)$. Consequently, such a recursive scheme produces the feedback law

$$\boldsymbol{u}(k) = h(\boldsymbol{x}(k)) \tag{6}$$

with h: $\mathbb{R}^n \rightarrow \mathbb{R}^m$.

Many approaches are proposed to solve OCP (4) through nonlinear programming methods (Henson, 1998), such as the sequential approach (Vassiliadis *et al.*, 1994a; 1994b) and the simultaneous approach (Jockenhövel *et al.*, 2003; Kameswaran and Biegler, 2006; Lang and Biegler, 2007). In this work, the sequential approach is employed due to its easy implementation. In the sequential approach, only control variables are discretized and an efficient integrator is used to obtain new state information. At every optimization iteration, system simulation and optimization are performed sequentially, i.e., one after another (Diehl *et al.*, 2008).

In consideration of the closed-loop stability of the controlled dynamic system, it is required that the final state $\boldsymbol{x}(T_p)$ should satisfy a certain inequality constraint, as discussed in the review by Mayne *et al.* (2000). Therefore, a simple method (Tenny, 2002; Tenny *et al.*, 2004) is employed to check whether the terminal constraint has been satisfied. If not, then the horizon T_p is increased and the OCP problem is resolved.

3 NMPC framework via FP-SQP with the RPS criteria

3.1 Motivation

Ideally, the OCPs can be solved instantaneously. This means that no computational delay exists. However, it is not true in practice, as solving OCP needs a certain time, which is not negligible.

We take the continuously stirred tank reactor (CSTR) (Henson and Seborg, 1997) as an example to explore the adverse effects of computational delay caused by the traditional criteria on control performance (Fig. 2). The specific parameters are listed in detail in Section 4. The tolerance of traditional termination criteria is $\varepsilon=10^{-6}$, as this is generally used as the default tolerance in many nonlinear programming (NLP) solvers.

Remark 1 If computational delay exists, then the previous control effort will act in the following sampling periods until the computation of OCP is finished and the control effort is available.

To analyze the OCP solution procedures at certain time steps where non-negligible computational delay specifically exists, we refer to the OCP objective function profiles, as shown in the sub-graphs of Fig. 2. The objective of OCP decreases quickly (in about five iterations) and slightly changes from there on. However, under the traditional termination criteria, the optimization procedure does not stop until the given tolerance is satisfied, or failure of convergence is ultimately declared. Objectively, under traditional criteria, a tiny improvement of the solution is obtained at the cost of many computational resources. Computation is extensively delayed, tracking performance deteriorates, and the system oscillates fiercely (Fig. 2). If optimization can be terminated earlier, computation time can be saved while controller performance might be improved.

Factually, the solution should be terminated when one of the following cases occurs:

1. Only a feasible solution, rather than the global or local extreme, can be obtained, but the procedure still goes on and struggles to find the optimal solution and satisfy the tolerance.

2. The constraints will not be satisfied and convergence fails eventually anyway.

3. The improvement of the solution is tiny even if the optimization procedure continues.

However, when traditional criteria are used, the solution process will not stop until the number of iterations exceeds the maximum number or the specified tolerance is met.

Moreover, the results produced by different kinds of optimization algorithms are analyzed as follows:

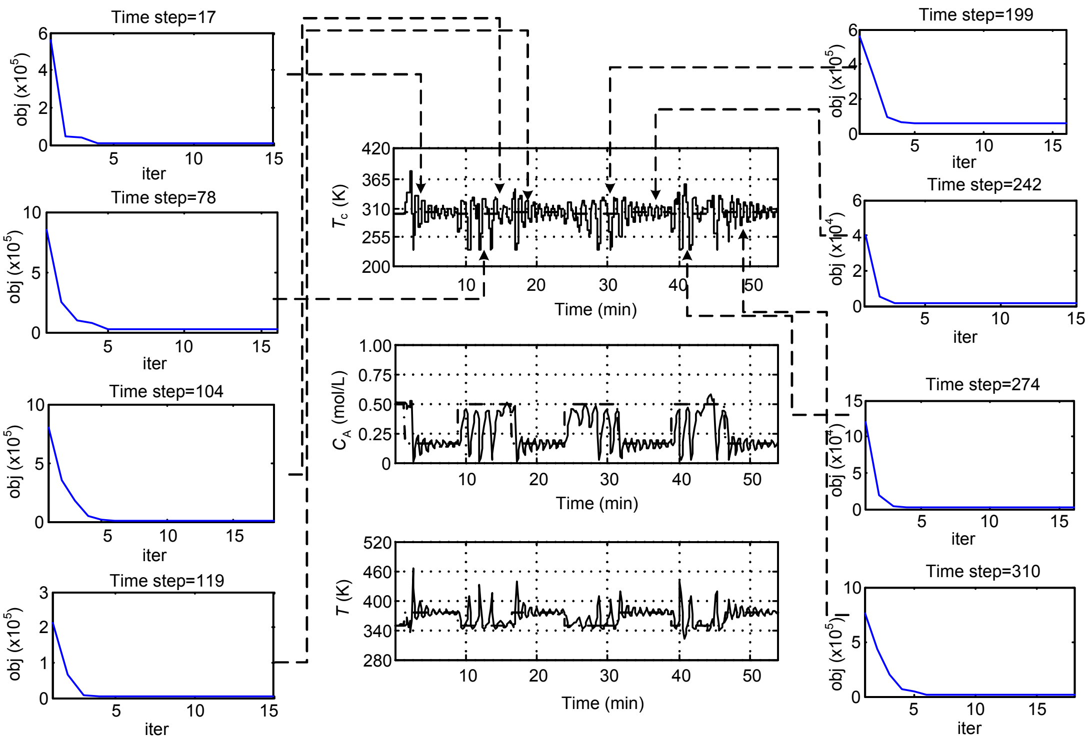

Fig. 2 Effect of computational delay on control performance

The sub-graphs show the profiles of the objective functions at some time steps where computational delay exists. T_c is the input, C_A is the state, and T is the output. obj: value of the objective function; iter: number of iterations

1. If an infeasible path method is used, each iterate is an infeasible approximation and does not satisfy the given constraints. In this case, the iterate may cause oscillation if it acts on the system when the solution procedure is terminated early.

2. If a feasible path method is used, the solution procedure can be terminated at any time when it is necessary (Tenny *et al.*, 2004) and the iterate is a kind of suboptimal approximation that can be injected to the system. However, the rules for judging whether the current iterate is good enough are not available.

As analyzed above, an infeasible path method is not suitable for solving the OCP when it needs to be terminated prematurely because of the critical requirement of computation time. Therefore, a feasible path strategy is considered in this work.

To explore the effect of termination criteria on control performance, the relationship between computational delay and solution accuracy with change of the tolerance should be considered (Fig. 3). Line 1 implies that larger tolerance results in the quick termination of the solution procedure so that computational delay becomes smaller, or vice versa. If a feasible solution can be obtained eventually, the trend of the accuracy of solution depending on different tolerances could be represented by Line 2. With a feasible iterate, the accuracy of solution may be high when tolerance is very small, as shown by the left extreme of Line 2. Its right extreme represents the contrary, that is, a coarse solution and a large tolerance. As analyzed, both long computation time and coarse solution degrade controller performance. Therefore, a tradeoff between the accuracy of solution and the computation time is necessary, as depicted by the dash dot line. Better control performance can be obtained at the tradeoff point, and the RPS criteria are proposed for finding it.

In consideration of the practical requirement for solution precision, a series of indices are defined to stop quickly the solution procedure while maintaining good approximation, so that the aforementioned tradeoff can be reached. Additionally, the feasibility of iterate should be guaranteed. Then the optimization procedure can be terminated prematurely. Here, the FP-SQP algorithm guaranteeing feasible iterate (Wright and Tenny, 2004) is employed to implement the RPS criteria.

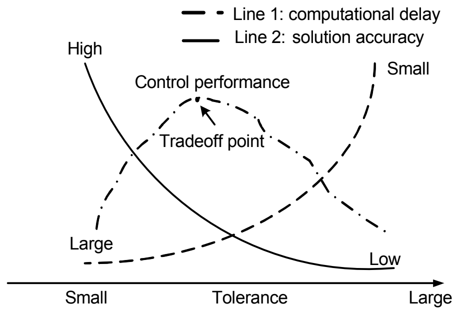

Fig. 3 Schematic of the relationship between the solution accuracy and computational delay on control performance with change of tolerance

3.2 NMPC based on the FP-SQP algorithm with the RPS criteria

Consider the OCP (4) in a general form of NLP problems:

$$\min_{z\in\mathbb{R}^n} f(z) \quad \text{s.t.} \quad d(z)=0,\ c(z)\le 0, \tag{7}$$

where $f(z)$: $\mathbb{R}^n\rightarrow\mathbb{R}$ is the objective function, $c(z)$ and $d(z)$ are the constraints, and $f(z)$, $c(z)$, and $d(z)$ are smooth (i.e., twice continuously differentiable) functions. A series of feasible iterates $\{z_j\}_{j=0,1,\ldots}$ is generated by solving the following quadratic programming sub-problem for Δz through the trust region SOP algorithm (Tenny *et al.*, 2004):

$$\min_{\Delta z} m(\Delta z) \overset{\text{def}}{=} \nabla f^{\mathrm{T}}\Delta z + \frac{1}{2}\Delta z^{\mathrm{T}}\boldsymbol{H}\Delta z \tag{8a}$$

$$\text{s.t.} \quad \begin{cases} d(z)+\nabla d(z)^{\mathrm{T}}\Delta z = 0, \\ c(z)+\nabla c(z)^{\mathrm{T}}\Delta z \le 0, \end{cases} \tag{8b}$$

$$\|\boldsymbol{S}\Delta z\|_P \le r, \tag{8c}$$

where $\boldsymbol{H}$ is the approximation to the Hessian of the Lagrangian, $\boldsymbol{S}$ is the scaling matrix for the trust region, and r represents the trust region radius. Eq. (8a) is the model function of the change in the objective function of Eq. (7) at current iterate z_j. Eq. (8b) represents the linearization of the constraints around the current iterate. Eq. (8c) is the trust region constraint, where $P\in[1, \infty)$ denotes the choice of norm.

To make the iterate feasible, the candidate step Δz is perturbed using the l_2 penalty method formulated as follows:

$$\min_{\Delta\tilde{z}} \boldsymbol{R}\left\|\Delta z - \Delta\tilde{z}\right\|_2 \quad \text{s.t. } z + \Delta\tilde{z} \in \mathbb{F}, \tag{9}$$

where $\boldsymbol{R}$ is the weight matrix, and $\mathbb{F}$ is a feasible point set of problem (4).

The acceptability of a candidate step $\Delta\tilde{z}$ depends on a 'sufficient decrease' test defined as (Nocedal and Wright, 1999)

$$\rho_j = \frac{f(z_j) - f(z_j + \Delta\tilde{z}_j)}{-m_j(\Delta z_j)}, \tag{10}$$

where $z_j \in \mathbb{R}^n$ is the current iterate, m_j represents the model function m of Eq. (8a) evaluated at the current iterate, and Δz_j, $\Delta\tilde{z}_j$ are obtained by solving Eqs. (8) and (9), respectively. Based on the principle of trust region algorithms, the step is accepted if ρ_j is positive, and then the trust region radius and scaling matrix are updated. Otherwise, the current iterate is kept unchanged, the trust region radius is decreased, and a new candidate step is calculated.

The RPS criteria define a series of indices that could depict the quality of iterates and offer an approximate solution with acceptable optimality. Considering the feasibility of the iterate, we define l_2 penalty indices without measure of constraint violation. The proposed RPS criteria are defined with the following indices:

1. Predictive improvement on the iterates and objective function at iterate z_j are given as

$$\text{ind}_z^j = \max \| z_j - z_{j-1} \|_2, \tag{11a}$$

$$\text{ind}_f^j = \| f_j - f_{j-1} \|_2, \tag{11b}$$

where ind_z^j and ind_f^j denote the change of variable z and objective function f between iterations, respectively. Eq. (11) provides the degree of improvement of the current iterate and the objective function. It is also employed to calculate the degree of convergence.

2. To soften the indices in Eq. (11) and reduce their sensitivity to the change between iterations, the transformed sigmoid function is used as the measure of convergence degree:

$$\begin{cases} \eta_z^j = \dfrac{\tanh[\mu(\log \text{ind}_z^j) / \log \varepsilon_{\text{var}}]}{\tanh \mu}, \\ \eta_f^j = \dfrac{\tanh[\mu(\log \text{ind}_f^j) / \log \varepsilon_{\text{var}}]}{\tanh \mu}, \end{cases} \tag{12}$$

$$\eta^j = \min(\eta_z^j, \eta_f^j), \tag{13}$$

where ε_{var} is the pre-specified tolerance on the variables and μ is a transformation parameter. Indices η_z^j and η_f^j decrease monotonically with the increase of ind^j (Fig. 4). Traditional termination criteria are strict rules, and give only 'convergent' or 'not convergent' as the conclusion of the optimization. Using the sigmoid function, the extent of the convergence of iterate can be described smoothly and continuously, which is suitable for measuring the degree of convergence of the current iterate.

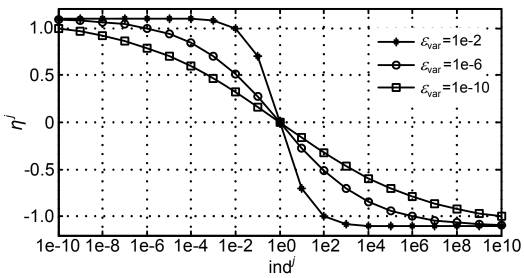

Fig. 4 Profile of the transformed sigmoid function (μ=1.5) with change of tolerance

The index in Eq. (13) is defined to judge whether the current iterate converges sufficiently. If it reaches some predefined threshold to reflect the degree of convergence, then the solution progress should be terminated and an approximate solution is obtained. Moreover, the successive improvement is small even though the process continues. Such kind of criteria, with a proper modification, is suitable for general optimization algorithms, such as the rSQP algorithm (Wang *et al.*, 2007) and the interior point method (Chen *et al.*, 2010). Usually, the specified threshold θ_0 can be determined according to user's demand by using (0, 1).

Remark 2 Apart from the termination criteria, another factor that affects algorithm performance is the initialization of the starting point. To produce a feasible iterate, the starting point is also required to be feasible. During the process of the solution for problem (4), shifted input variables of the previous time step are taken as the initialization of the current optimization procedure (Diehl *et al.*, 2008). At the very beginning, the simulation of the controlled plant can be conducted, or a linear control technique can be used (e.g., the linear quadratic regulator, or LQR) to calculate the feasible solution (Henson and Seborg, 1997; Tenny *et al.*, 2004).

To summarize, Fig. 5 illustrates the flow sheet of FP-SQP with the RPS criteria.

4 Simulation results and discussion

When linear control techniques are used, CSTR is difficult to control due to its inherent nonlinear behaviors (Bequette, 2002). Thus, research on NMPC of CSTR has been carried out in recent years (Wu, 2000; Aguilar-Lopez and Martinez-Guerra, 2005; Czeczot, 2006; Pan *et al.*, 2007; Mansour and Ellis, 2008; Barkhordari Yazdi and Jahed-Motlagh, 2009). To illustrate the efficiency of the RPS criteria, we referred to the control of a classical CSTR (Henson and Seborg, 1997; Tenny *et al.*, 2004) in which an exothermic and irreversible reaction occurs.

The temperature of the reactor can be reduced by adjusting the temperature of the coolant fluid in a heat exchange coil inside the vessel. The dynamic model is based on the component balance for reactant A and on an energy balance in the following differential model equations:

$$\begin{cases} \dot{C}_{\mathrm{A}} = \dfrac{q}{V}(C_{\mathrm{Af}} - C_{\mathrm{A}}) - k_0 \exp\left(-\dfrac{E}{RT}\right) C_{\mathrm{A}}, \\ \dot{T} = \dfrac{q}{V}(T_{\mathrm{f}} - T) + \dfrac{-\Delta H}{\rho C_{\mathrm{p}}} k_0 \exp\left(-\dfrac{E}{RT}\right) C_{\mathrm{A}} \\ \qquad + \dfrac{UA}{C_{\mathrm{p}} V \rho}(T_{\mathrm{c}} - T), \end{cases} \tag{14}$$

where UA, q, V, C_{Af}, E, R, ρ, k_0, $-\Delta H$, and C_{p} are constant parameters, as listed in Table 1. The states of the system consist of the concentration of reactant A (i.e., C_{A}) and reactor temperature T (output), and the manipulated variable is the temperature of coolant stream, T_{c}.

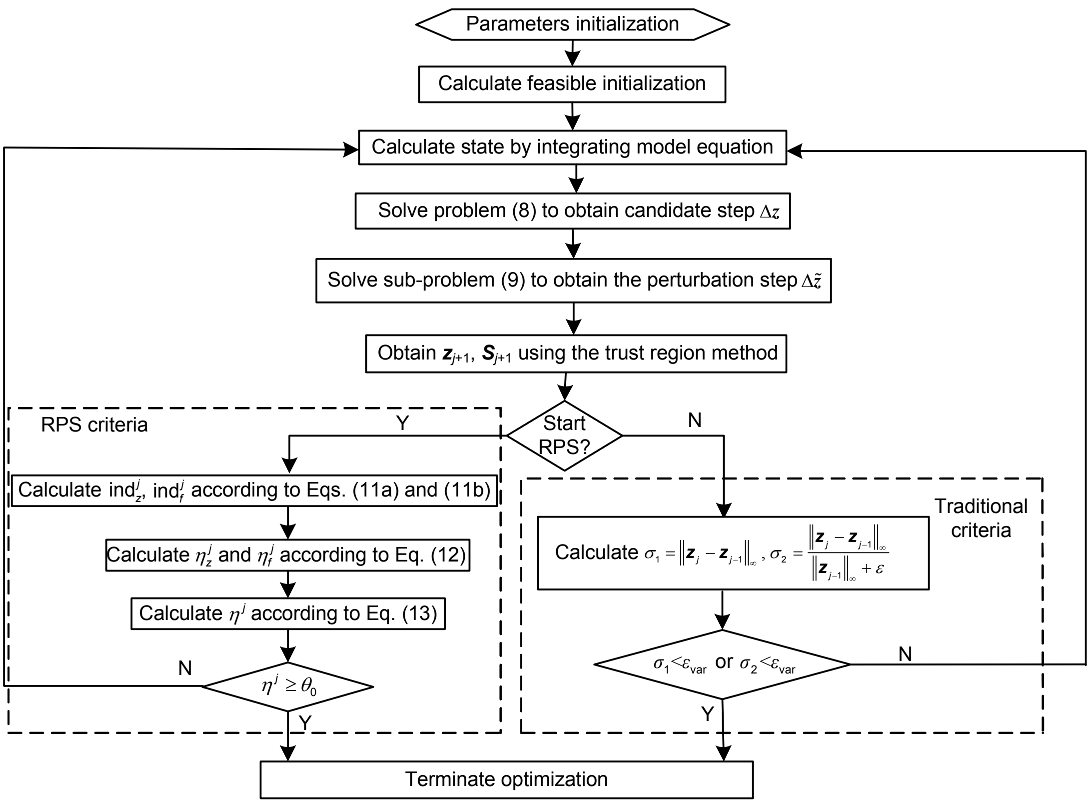

Fig. 5 Flow sheet of the feasibility-perturbed sequential quadratic programming (FP-SQP) algorithm with the reduced precision solution (RPS) criteria

Table 1 Parameters of the continuously stirred tank reactor (CSTR) model*

Symbol	Description	Value
UA	U: overall heat transfer; A: area	5e4 J/min·K
q	Volumetric flow rate	100 L/min
V	Volume of CSTR	100 L
$-\Delta H$	Heat of reaction for A→B	5e4 J/mol
C_{Af}	Feed concentration	1 mol/L
E/R	E: activation energy; R: universal gas constant	8750 K
T_{f}	Feed temperature	350 K
ρ	Density of A, B mixture	1000 g/L
C_{p}	Heat capacity of A, B mixture	0.239 J/g·K
K_0	Pre-exponential factor	7.2e10 min^{-1}

* Tenny *et al.*, 2004

Assume that the output reference is a piecewise constant, and is composed of two operating points (OPs) (Findeisen and Allgöwer, 2003; Tenny *et al.*, 2004). The locations of OPs are illustrated in Fig. 6. To test the tracking performance, we switch references between the two OPs (Fig. 7). The purpose is to drive the controlled system to the given operating points which require both state variables and input variables achieve the target values. The corresponding parameters are set as listed in Table 2.

Given target values $C_{\mathrm{A,\,target}}$ and T_{target} for the states, as well as $T_{\mathrm{c,\,target}}$ for the input, the deviations of state and input are defined as follows (Tenny, 2002):

$$\boldsymbol{x}=\begin{bmatrix} C_{\mathrm{A}}-C_{\mathrm{A,target}} \\ T-T_{\mathrm{target}} \end{bmatrix},\quad u=T-T_{\mathrm{c,target}}. \tag{15}$$

The parameters including weight matrices $\boldsymbol{Q}$, R, and S, sampling time, and prediction horizon (for convenience, prediction horizon and control horizon are made the same) are listed in Table 2.

All numerical results are obtained on the computer running Windows 7 with Intel® Dual Core™ 2 Duo 2.66 GHz CPU and 2 GB RAM. MATLAB 7.4 is used as the computation environment and SUNDIALS (Hindmarsh *et al.*, 2005) is used to calculate sensitivity information during simulations.

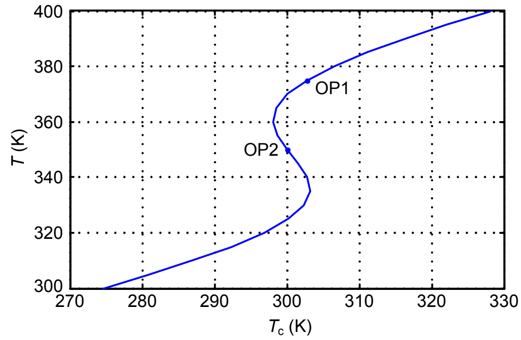

Fig. 6 Steady-state map between temperature of coolant stream T_{c} (input) and the reactor temperature T (output)

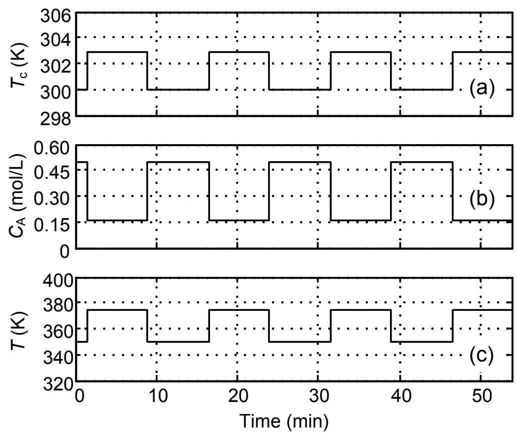

Fig. 7 Reference profiles of input T_{c} (a), state C_{A} (b), and output T (c)

Remark 3 To reveal the effect of computation delay on the control performance, the simulation in this section is made in the nominal case in which perturbation, disturbance, and model-plant mismatch are not considered.

The FP-SQP algorithm with the traditional criteria (Tenny *et al.*, 2004) is used for comparison. The traditional criteria are defined as follows:

Table 2 Control parameters corresponding to different operating points

Operating points 1*	Operating points 2**	Weight matrices $\boldsymbol{Q}$	R	S	Sampling interval	Prediction horizon	Control horizon	Boundary of input
T_{target}=375 K $C_{\mathrm{A,\,target}}$=0.159 mol/L $T_{\mathrm{c,\,target}}$=302.84 K	T_{target}=350 K $C_{\mathrm{A,\,target}}$=0.5 mol/L $T_{\mathrm{c,\,target}}$=300 K	$\begin{bmatrix}10 & 0\\ 0 & 50\end{bmatrix}$	2	3	9 s	90 s	90 s	[230, 427]

* For time steps 11–60, 111–160, 211–260, and 311–360; ** For time steps 61–110, 161–210, and 261–310

$$\sigma_1 = \| z_j - z_{j-1} \|_\infty,\ \sigma_2 = \frac{\| z_j - z_{j-1} \|_\infty}{\| z_{j-1} \|_\infty + \varepsilon}, \tag{16}$$

where ε is defined as the smallest real number in floating point arithmetic. The iterate is acceptable if one of the two indices in Eq. (16) is not larger than the given tolerance $\varepsilon_{\text{var}}=10^{-6}$.

The integral square error (ISE) is used to measure the reference tracking performance of the controller:

$$\text{ISE} = \sum_{k=1}^{N} (x(k) - x_{\text{ref}}(k))^2, \tag{17}$$

where N is the length of simulation horizon, and x and x_{ref} represent state variable and reference, respectively. To show the general influence of computational delay caused by different solution precisions, we changed the tolerance of the traditional criteria from 10^{-1} to 10^{-9}. The statistics of the corresponding ISE of the state variables and computation time are shown in Table 3.

Table 3 Comparison of computation time and the integral square error (ISE) obtained from the traditional criteria with different tolerances

Tolerance	Computation time (s)			ISE	
	Min	Max	Ave	C_A	T
1e-1	0.39	6.58	1.41	2.584	1.83e4
1e-2	0.41	21.11	1.79	3.519	3.95e4
1e-3	0.42	8.18	2.55	3.338	3.33e4
1e-4	1.17	13.50	3.74	2.980	6.11e4
1e-5	1.97	9.81	5.15	3.251	4.19e4
1e-6	2.07	12.45	5.78	8.893	1.01e5
1e-7	2.14	47.33	6.04	8.546	1.31e5
1e-8	1.81	24.18	6.11	9.610	1.61e5
1e-9	2.41	22.34	6.22	11.211	1.99e5

To describe the proportion of computational delay along with the simulation horizon, we define an index referred to as rate of 'efficient receding horizon optimization':

$$r_{\text{ERHO}} = \frac{N_{\text{ERHO}}}{N}, \tag{18}$$

where N_{ERHO} is the number of time steps wherein the associated OCP is solved successfully in one sampling period, and N is the total number of time steps along the simulation horizon. This means that if the OCPs at all time steps can be solved in time, then $r_{\text{ERHO}}=1$; otherwise, the OCP at a certain time step needs more than one sampling period and the following OCP would be neglected, resulting in r_{ERHO} within (0, 1). Fig. 8 shows the relationship between ISE of output T and index r_{ERHO} with change of tolerance.

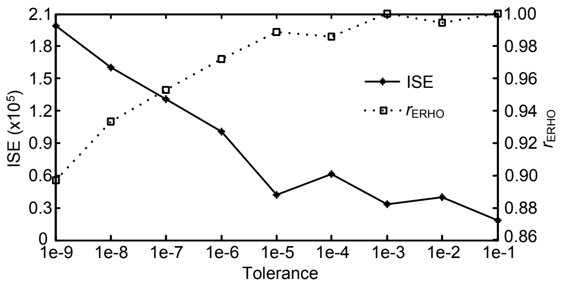

Fig. 8 Relationship between the integral square error (ISE) of output T and r_{ERHO} with change of tolerance

From Fig. 8, the large tolerance results in high r_{ERHO} and small ISE, and vice versa. It is deduced that the CSTR is more sensitive to computation time than to the accuracy of solution because much smaller ISE is produced, as shown by the right extreme of the ISE profile, although a coarse solution is obtained. Based on the trends of ISE and r_{ERHO}, it is concluded that even the OCP at one time step is solved thoroughly, as shown by the left extremes of two lines, large computational delay causes large ISE; it is true for the contrary. However, the control performance is reflected not only by the value of ISE. To analyze further controller performance influenced by different tolerances of the traditional criteria, we employ the profiles of input, state, and output variables obtained from traditional criteria with different tolerances (Fig. 9).

Fig. 9 shows that the controlled variables (i.e., state variables) oscillate fiercely when a large computational delay exists (e.g., Fig. 9i), especially when reference switches from OP1 to OP2, as illustrated in Fig. 9f. When a larger tolerance is used, as shown in Figs. 9a–9c, the controlled variables show better tracking performance because the computational delay is small. Specifically, when OP1 is tested, almost the expected performance can be obtained. This demonstrates that high real-time performance resulting from a large tolerance improves controller performance. However, taking Fig. 9a as an example, the coarse solution obtained with a large tolerance makes

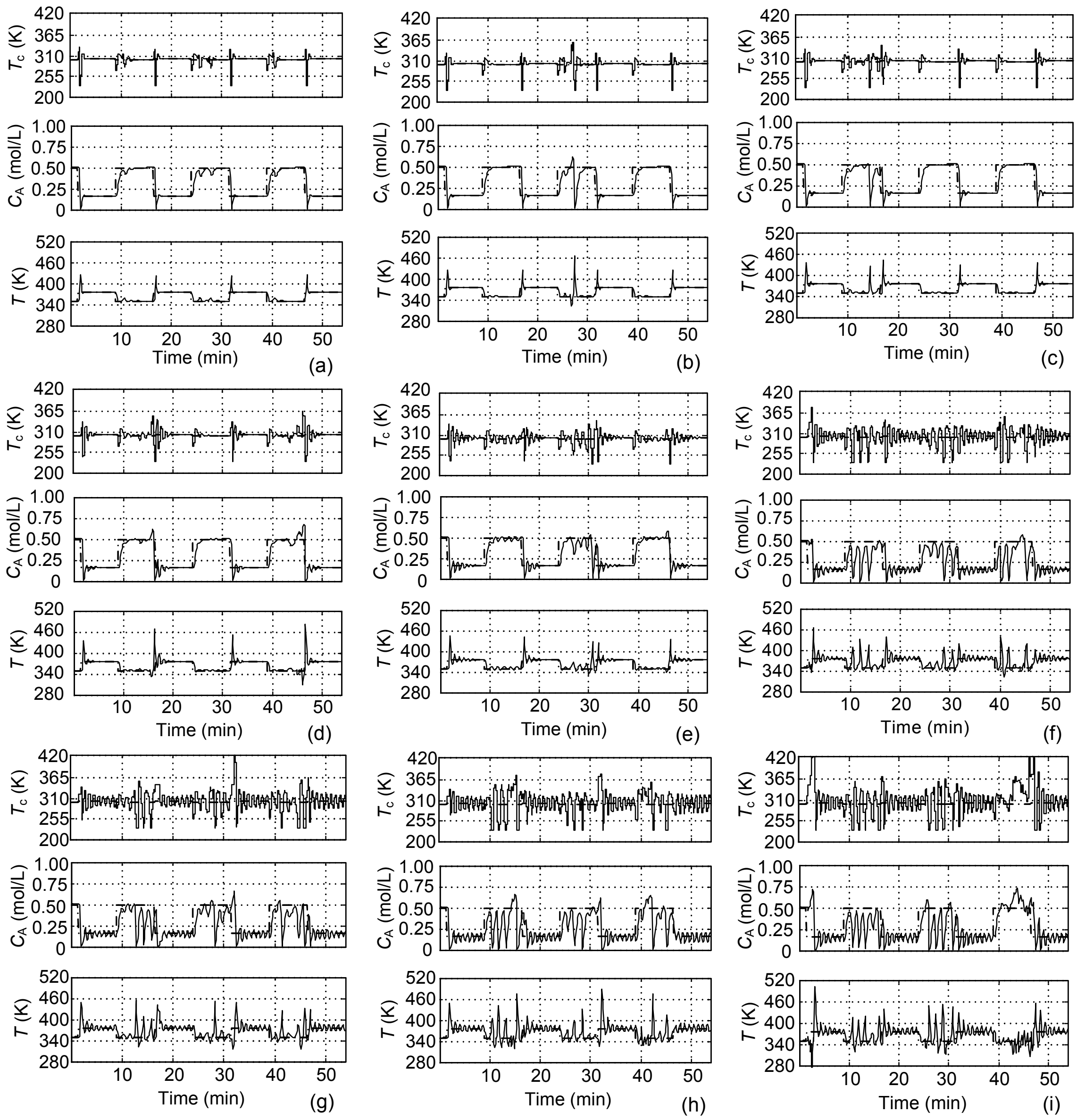

Fig. 9 Numerical results of reference tracking obtained from traditional criteria with different tolerances
(a) to (i) demonstrate results with tolerance changing from 10^{-1} to 10^{-9} in descending order. Solid line: plant response; dashed line: reference. T_c is the input, C_A is the state, and T is the output

the controlled system achieve the given steady state slowly and unstably, hence degrading controller performance as well. It shows that just a feasible solution is not enough to guarantee good control performance; the accuracy of solution is also important. Therefore, it is deduced that even when a feasible path method is used, the optimization procedure cannot be stopped at any time. As a feasible solution, which is significantly different from the optimal one, it may not improve the closed-loop control performance or keep the system stable.

By taking as an example the results of the traditional criteria of $\varepsilon_{var}=10^{-6}$ (which is usually considered as the default tolerance in most NLP solvers), comparisons are made on computation time produced from the traditional criteria and the RPS criteria with threshold 0.5 (Fig. 10). The computation time for the RPS criteria is much shorter than that for the traditional criteria. At certain time steps, the computation time is 0 because the OCP solution in the previous time step remains incomplete, and thus the OCP at the current time step is neglected. Therefore, only the computation time of the solved OCP is compared. As shown in Fig. 10, when the RPS criteria are adopted, a lot of computing resources are saved.

The numerical results are listed in Table 4. For the RPS criteria, the average time is reduced by almost 80%, and the ISE of state variables is reduced by more than 70%. Moreover, index r_{ERHO} for the RPS criteria reaches 1, indicating that the OCPs at all time steps are calculated in time and that the corresponding computation time is less than one sampling interval.

Compared with Table 3, it is clear that the ISE obtained using RPS criteria is smaller than all of those resulting from traditional criteria of different tolerances, and that the average computation time is much shorter than one sampling interval (9 s). The high real-time performance and good solution accuracy imply better control performance.

The result of reference tracking performance for the RPS criteria is given in Fig. 11. Specifically, the comparison of objective profiles corresponding to the

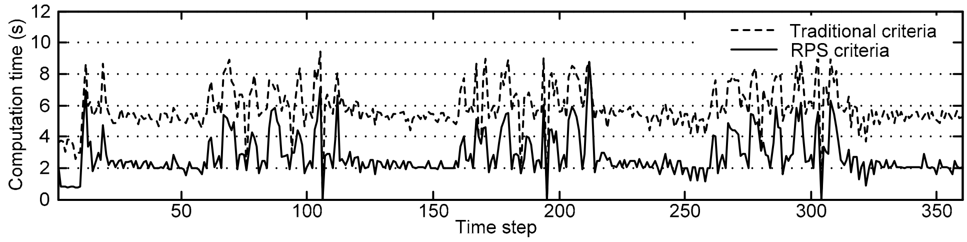

Fig. 10 Comparison of computation time for solving the optimal control problem under the traditional criteria and the reduced precision solution (RPS) criteria

Table 4 Comparison of computation time and the integral square error (ISE) index under the traditional criteria ($\varepsilon_{\mathrm{var}}$=10^{-6}) and the reduced precision solution (RPS) criteria

Termination criteria	Computation time (s)			ISE		r_{ERHO}
	Minimum	Maximum	Average	C_{A}	T	
Traditional	2.07	12.45	5.78	8.893	1.01e5	0.97
RPS	0.73	8.02	1.28	2.469	1.75e4	1

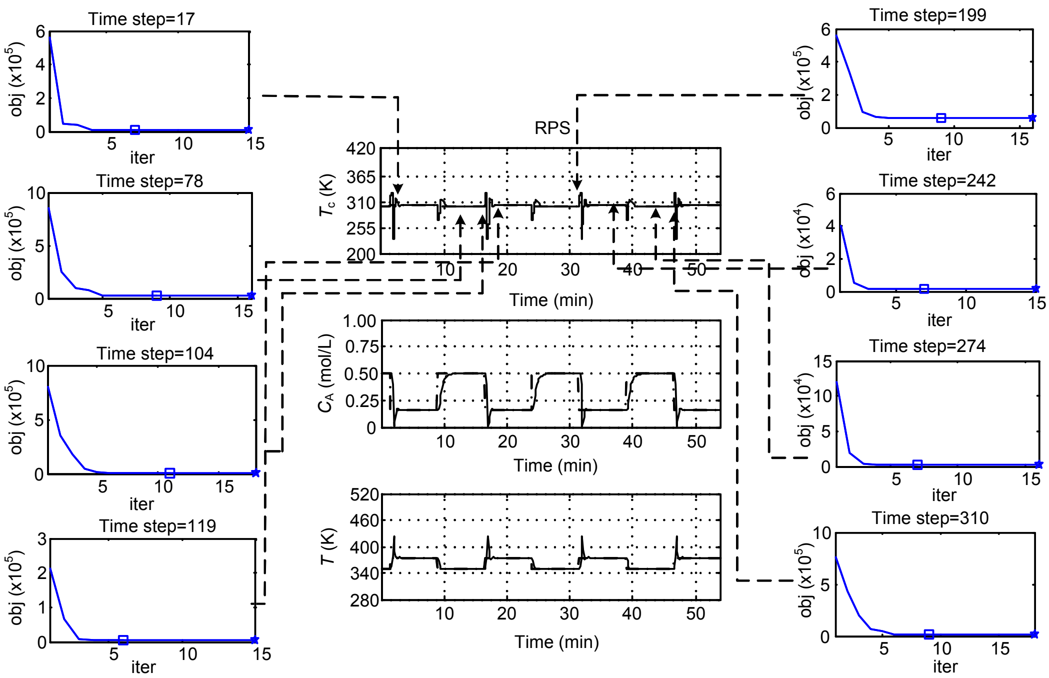

Fig. 11 Reference tracking performance for the reduced precision solution (RPS) criteria and the optimal control problem objective function profiles at the time step corresponding to Fig. 2

The termination points of the runs under traditional criteria or RPS criteria are marked by '☆' or '□', respectively. T_{c} is the input, C_{A} is the state, and T is the output. obj: value of the objective function; iter: number of iterations sub-graphs of Fig. 2 is also shown.

Compared with Fig. 2, the adverse effect caused by computational delay has been removed completely using the RPS criteria. The overshoot is also reduced. In the nominal case, the system is driven quickly and stably to the required steady state. To conclude, the runs under RPS criteria terminate much earlier than the ones under traditional criteria. NMPC based on RPS criteria outperforms that using traditional criteria, as seen from the comparative results in Tables 3 and 4. The simulations demonstrate that using the RPS criteria could reduce computation time efficiently with a good feasible solution. Consequently, the controller performance is improved.

5 Conclusions

We analyze the effect of computational delay caused by the traditional termination criteria on control performance of NMPC controllers. Usually, the traditional termination criteria result in a slow solution process with tiny improvement of solution after a few iterations. This causes a waste of computing resources and degrades controller performance. Detailed analysis of this study shows that both the low real-time performance and coarse solution of OCP could degrade controller performance, as shown by simulation with traditional criteria with different tolerances. The RPS termination criteria are proposed for solving the OCP by employing the FP-SQP algorithm. Indices are constructed to judge whether the optimization procedure should terminate and whether the current iterate is a good approximation, and the threshold can be used to adjust the degree of the approximation of the iterate according to the user's needs. The performances of the traditional and the RPS criteria are compared by applying the FP-SQP algorithm with both criteria to solve the reference tracking problem of CSTR, respectively. Simulation results demonstrate a higher efficiency of the RPS criteria. Numerical results show that the RPS criteria guarantee a timely termination of the optimization procedure, implying that the control performance is improved.

The RPS criteria can be used as an alternative to the termination criteria of NLP solvers, especially in meeting the real-time requirements of the application of NMPC. In this paper, however, only the simulation in the nominal case (in which no disturbance, model-plant mismatch, or perturbation exists) is made to exemplify the efficiency of the proposed RPS criteria in reducing computational delay. For future work, to make the criteria more applicable and robust, perturbation, disturbance, and model-plant mismatch will be considered. The closed-loop stability of the NMPC with the RPS criteria will also be studied. The input-to-state stability (ISS) and the related extended theories can be powerful tools for analyzing the stability and robustness. Special attention will be paid to the effect that the sub-optimal solution obtained from RPS criteria has on the ISS of the closed-loop system. The relationship between the deviations of the sub-optimal solution from the optimal and the resulting deterioration of the ISS property will be studied further. Even the precision of chemical instruments can be introduced to make the RPS criteria suitable for practical applications.

Acknowledgements

The authors particularly thank Prof. L. T. BIEGLER from Carnegie Mellon University for his kind suggestions.

References

Aguilar-Lopez, R., Martinez-Guerra, R., 2005. State estimation for nonlinear systems under model unobservable uncertainties: application to continuous reactor. *Chem. Eng. J.*, **108**(1-2):139-144. [doi:10.1016/j.cej.2005.01.008]

Barkhordari Yazdi, M., Jahed-Motlagh, M.R., 2009. Stabilization of a CSTR with two arbitrarily switching modes using modal state feedback linearization. *Chem. Eng. J.*, **155**(3):838-843. [doi:10.1016/j.cej.2009.09.008]

Bequette, B.W., 2002. Behavior of CSTR with a Recirculating Jacket Heat Transfer System. American Control Conf., p.3275-3280.

Bock, H.G., Diehl, M., Kühl, P., Kostina, E., Schlöder, J.P., Wirsching, L., 2007. Numerical methods for efficient and fast nonlinear model predictive control. *Lect. Notes Control Inform. Sci.*, **358:**163-179. [doi:10.1007/978-3-540-72699-9_13]

Chen, H., Allgöwer, F., 1998. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, **34**(10):1205-1217. [doi:10.1016/S0005-1098(98)00073-9]

Chen, W., Shao, Z., Wang, K., Chen, X., Biegler, L.T., 2010. Convergence depth control for interior point methods. *AIChE J.*, **56**(12):3146-3161. [doi:10.1002/aic.12225]

Chen, W.H., Ballance, D.J., O'Reilly, J., 2000. Model predictive control of nonlinear systems: computational burden and stability. *IEE Proc.-Control Theory Appl.*, **147**(4):387-394. [doi:10.1049/ip-cta:20000379]

Czeczot, J., 2006. Balance-based adaptive control methodology and its application to the non-isothermal CSTR. *Chem. Eng. Process.*, **45**(5):359-371. [doi:10.1016/j.cep.2005.10.002]

DeHaan, D., Guay, M., 2006. A new real-time perspective on non-linear model predictive control. *J. Process Control*, **16**(6):615-624. [doi:10.1016/j.jprocont.2005.10.002]

Diehl, M., Bock, H.G., Schlöder, J.P., Findeisen, R., Nagy, Z., Allgöwer, F., 2002. Realtime optimization and non-linear model predictive control of processes governed by differential-algebraic equations. *J. Process Control*, **12**(4):577-588. [doi:10.1016/S0959-1524(01)00023-3]

Diehl, M., Bock, H.G., Schlöder, J.P., 2005. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM J. Control Optim.*, **43**(5):1714-1736. [doi:10.1137/S0363012902400713]

Diehl, M., Ferreau, H.J., Haverbeke, N., 2008. Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation. Workshhop on Assessment and Future Directions of NMPC.

Findeisen, R., Allgöwer, F., 2002. An Introduction to Nonlinear Model Predictive Control. 21st Benelux Meeting on Systems and Control, p.119-141.

Findeisen, R., Allgöwer, F., 2003. Computational Delay in Nonlinear Model Predictive Control. Proc. Int. Symp. on Advanced Control of Chemical Processes, p.427-452.

Gill, P.E., Murray, W., Wright, M.H., 1981. Practical Optimization. Academic Press, London.

Henson, M.A., 1998. Nonlinear model predictive control: current status and future directions. *Comput. Chem. Eng.*, **23**(2):187-202. [doi:10.1016/S0098-1354(98)00260-9]

Henson, M.A., Seborg, D.E., 1997. Nonlinear Process Control. Prentice Hall PTR, Upper Saddle River, New Jersey.

Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., Woodward, C.S., 2005. SUNDIALS, suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Software*, **31**(3):363-396. [doi:10.1145/1089014.1089020]

Jockenhövel, T., Biegler, L.T., Wächter, A., 2003. Dynamic optimization of the Tennessee Eastman process using the OptControlCentre. *Comput. Chem. Eng.*, **27**(11):1513-1531. [doi:10.1016/S0098-1354(03)00113-3]

Kameswaran, S., Biegler, L.T., 2006. Simultaneous dynamic optimization strategies: recent advances and challenges. *Comput. Chem. Eng.*, **30**(10-12):1560-1575. [doi:10.1016/j.compchemeng.2006.05.034]

Lang, Y.D., Biegler, L.T., 2007. A software environment for simultaneous dynamic optimization. *Comput. Chem. Eng.*, **31**(8):931-942. [doi:10.1016/j.compchemeng.2006.10.017]

Mansour, M., Ellis, J.E., 2008. Methodology of on-line optimization applied to a chemical reactor. *Appl. Math. Model.*, **32**(2):170-184. [doi:10.1016/j.apm.2006.11.014]

Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M., 2000. Constrained model predictive control: stability and optimality. *Automatica*, **36**(6):789-814. [doi:10.1016/S0005-1098(99)00214-9]

Nocedal, J., Wright, S.J., 1999. Numerical Optimization. Springer, New York. [doi:10.1007/b98874]

Pan, T., Li, S., Cai, W., 2007. Lazy learning-based online identification and adaptive PID control: a case study for CSTR process. *Ind. Eng. Chem. Res.*, **46**(2):472-480. [doi:10.1021/ie0608713]

Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. *Control Eng. Pract.*, **11**(7):733-764. [doi:10.1016/S0967-0661(02)00186-7]

Santos, L.O., Afonso, P.A., Castro, J.A., Oliveira, N.M., Biegler, L.T., 2001. On-line implementation of nonlinear MPC: an experimental case study. *Control Eng. Pract.*, **9**(8):847-857. [doi:10.1016/S0967-0661(01)00049-1]

Schäfer, A., Kühl, P., Diehl, M., Schlöder, J., Bock, H.G., 2007. Fast reduced multiple shooting methods for nonlinear model predictive control. *Chem. Eng. Process.: Process Intens.*, **46**(11):1200-1214. [doi:10.1016/j.cep.2006.06.024]

Scokaert, P.O.M., Mayne, D.Q., Rawlings, J.B., 1999. Suboptimal model predictive control (feasibility implies stability). *IEEE Trans. Autom. Control*, **44**(3):648-654. [doi:10.1109/9.751369]

Tenny, M., 2002. Computational Strategies for Nonlinear Model Predictive Control. PhD Thesis, University of Wisconsin-Madison, Madison, Wisconsin, USA.

Tenny, M., Wright, S.J., Rawlings, J.B., 2004. Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming. *Comput. Optim. Appl.*, **28**(1):87-121. [doi:10.1023/B:COAP.0000018880.63497.eb]

Vassiliadis, V.S., Sargent, R.W.H., Pantelides, C.C., 1994a. Solution of a class of multistage dynamic optimization problems. 1. Problems without path constrints. *Ind. Eng. Chem. Res.*, **33**(9):2111-2122. [doi:10.1021/ie00033a014]

Vassiliadis, V.S., Sargent, R.W.H., Panteides, C.C., 1994b. Solution of a class of multistage dynamic optimization problems. 2. Problems with path constraints. *Ind. Eng. Chem. Res.*, **33**(9):2123-2133. [doi:10.1021/ie00033a015]

Wang, K., Shao, Z., Zhang, Z., Chen, Z., Fang, X., Zhou, Z., 2007. Convergence depth control for process system optimization. *Ind. Eng. Chem. Res.*, **46**(23):7729-7738. [doi:10.1021/ie070073s]

Wright, S.J., Tenny, M., 2004. A feasible trust-region sequential quadratic programming algorithm. *SIAM J. Optim.*, **14**(4):1074-1105. [doi:10.1137/S1052623402413227]

Wu, W., 2000. Nonlinear bounded control of a nonisothermal CSTR. *Ind. Eng. Chem. Res.*, **39**(10):3789-3798. [doi:10.1021/ie990186e]

Zavala, V.M., Biegler, L.T., 2009. The advanced-step NMPC controller: optimality, stability and robustness. *Automatica*, **45**(1):86-93. [doi:10.1016/j.automatica.2008.06.011]