Model predictive control of a CSTR: a comparative study among linear and nonlinear model approaches

Ashok Krishnan*, Bhagyesh V. Patil†, P. S. V. Nataraj‡, Jan Maciejowski§ and K. V. Ling*

*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.
†Cambridge Centre for Advanced Research in Energy Efficiency, 1 Create Way, CREATE Campus, Singapore.
‡IDP in Systems and Control Engineering, IIT Bombay, Mumbai.
§Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, United Kingdom.

***Abstract*—This paper presents a comparative study of two widely accepted model predictive control schemes based on mixed logical dynamical (MLD) and nonlinear modeling approaches with application to a continuous stirred tank reactor (CSTR) system. Specifically, we approximate the nonlinear behavior of a CSTR system with several local linear models in a MLD framework. The main benefit of a such scheme is the significant improvement in the model accuracy compared with a single linearized model. The benefits and trade-off associated with a predictive control law synthesized using MLD and nonlinear modeling approaches are also compared.**

I. Introduction

Model predictive control (MPC) is an advanced control methodology for multivariable control systems. It is also known as receding horizon control. MPC generates control actions by optimizing a specific performance index based on a dynamic model of the system to be controlled over a finite-time moving window within system constraints [1], [2]. The application of MPC on a large scale can be found in process industries, especially petrochemical plants [3]. In the last one decade, MPC has also gained prominence in the other industries including automotive, aerospace and robotics (see, for instance [4], and reference therein). A more recent survey on the current trends in the field of MPC can be found in [5].

In practice, many MPC applications prefer linear models which are obtained by linearizing the original nonlinear system description around a single operating point. The inherent simplicity of linear models facilitates the use of convex optimization techniques for the online solution of optimization problems. Typically, such MPC applications are also known as *linear MPC* (LMPC) schemes (cf. [6]). However, linear models provide good approximations of the nonlinear system only when the system is operating close to the chosen linearized point. As such, once the system moves away from this linearized point, closed-loop performance deteriorates quickly thereby negatively impacting process productivity. To overcome this limitation, the concept of hybrid systems is used in literature to approximate nonlinear system behavior using multiple linearizations interconnected with a set of logical rules [7]. These rules act as guidelines for switching from one linear model to another depending on the state of the system at a given time instant. Such MPC applications are known as mixed-logical dynamical MPC (MLD-MPC) scheme (cf. [8]).

Another interesting approach followed by many researchers is to use the nonlinear system model. This approach comes with attractive benefits such as higher product quality, tighter regulation of process parameters and the possibility of operating the process (with good control authority) in different operating regimes. Consequently, the model predictive control using nonlinear system models, usually called *nonlinear MPC* (or NMPC) has also attracted many researchers over the past decade [9], [10], [11]. It is worth noting that, an NMPC formulation requires the solution of a (usually *nonconvex*) nonlinear optimization problem at each sampling instant. As such, NMPC is a challenging field and is dependent on the adoption of good optimization techniques.

The scope of the present work involves a systematic study of the aforementioned MLD-MPC and NMPC schemes in the context of a CSTR system. First, we detail modeling of a CSTR system in the MLD framework. Subsequently, we compare the resultant MLD model with a single linearized model and justify via model validation, the choice of MLD model as a good prediction model for deriving the MPC control law. Finally, we compare the performance of MLD-MPC and NMPC schemes for the set-point tracking problem of a CSTR system and study the benefits and associated trade-offs among them.

II. Theory

In this section, we first introduce the nonlinear model of a CSTR system. Next, we briefly present the nonlinear MPC (NMPC) and mixed-logical dynamical MPC (MLD-MPC) formulations in the context of a CSTR system.

A. CSTR model

We consider a CSTR where an exothermic reaction $A \rightarrow B$ takes place. The dynamic behavior of such a system is described by a nonlinear differential equation [12]

$$\dot{x} = f(x, u), \quad x(0) = x_0. \tag{1}$$

*,† *Correspondance*: ashok004@e.ntu.edu.sg, bhagyesh.patil@gmail.com
*,†,§Authors acknowledge the support for research from the National Research Foundation, Prime Ministers Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

where

$$f(x,u) = \left(\begin{array}{c} -\phi x_1 e^{\left(\frac{x_2}{1+\frac{x_2}{\lambda}}\right)} + q(x_{1f} - x_1) \\ \beta\phi x_1 e^{\left(\frac{x_2}{1+\frac{x_2}{\lambda}}\right)} - (q+\delta) + \delta u + q x_{2f} \end{array} \right).$$

Here, $x_1 = C_A$ is the concentration of A in the reactor, $x_2 = T$ is the temperature of reaction mixture and $u = T_C$ is the temperature of the coolant stream. Numerical values of the parameters corresponding to the nominal operating point condition are $\phi = 0.072, \beta = 8, \delta = 0.3, \lambda = 20, q = 1, x_{1f} = 1, x_{2f} = 0$.

The system is subjected to the following constraints on states and input:

$$x = \left\{ \left[\begin{array}{c} x_1 \\ x_2 \end{array} \right] \in \mathbb{R}^2 \mid 0 \le x_1 \le 1, \, 0 \le x_2 \le 6 \right\}. \quad (2)$$

$$u \in \mathbb{R}\ , \ -2 \le u \le 2. \quad (3)$$

B. NMPC formulation

We formulate NMPC problem for continuous-time systems described by the nonlinear model (1), where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ denote the vectors of states and control inputs, respectively. The state of the system and the control input applied at a sampling instant k are denoted by $x(k)$ and $u(k)$, respectively. The system is subject to state and input constraints of the following form:

$$x(k) \in \mathcal{X}, \ \ \forall\, k \ge 0 \quad (4)$$

$$u(k) \in \mathcal{U}, \ \ \forall\, k \ge 0 \quad (5)$$

where $\mathcal{X} \subseteq \mathbb{R}^n, \mathcal{U} \subseteq \mathbb{R}^m$, and are given by bound constraints of the form:

$$\mathcal{X} := \{x \in \mathbb{R}^n \mid x_{\min} \le x \le x_{\max}\}.$$

$$\mathcal{U} := \{u \in \mathbb{R}^m \mid u_{\min} \le u \le u_{\max}\}.$$

We consider the design of an NMPC controller for (1) to track a desired reference x_s, subject to constraints of the form (4)-(5). Here, the general form of the NMPC control law can be derived at each sampling instant k by the solution of the following NLP problem:

$$\min_{u_i} \sum_{i=0}^{N-1} \left(Q \parallel x_i - x_{i,s} \parallel_p + R \parallel \Delta u_i \parallel_p\right) \quad (6)$$

$$\text{subject to} \quad (1),\ (4),\ \text{and}\ (5)\ \text{for}\ i = 0, 1, \ldots, N-1$$

where $x_{i,s}$ denotes the set-point (reference) at instant i; $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ denote positive definite, symmetric weighting matrices; $\Delta u_i = u_i - u_{i-1}$ denotes the control increment, $N (\ge 1)$ denotes the prediction horizon, and $\parallel . \parallel_p$ represents a standard vector p-norm with $p = 2$.

We follow the approach presented in [13] to compute an optimal control sequence based on the NMPC optimization problem formulated in (6).

C. MLD-MPC formulation

Hybrid systems integrate the continuous aspects of physical systems (like temperature and pressure of a system) as well as discrete decisions (like on/off decisions for solenoid valves) which need to be taken to drive the system from one operating point to another. Mixed Logical Dynamical (MLD) provides a natural modeling framework for such hybrid systems. The general MLD form of hybrid systems as described in [7] is as follows:

$$x(k+1) = Ax(k) + B_u u(k) + B_{aux} w(k) + B_{aff} \quad (7a)$$

$$y(k) = Cx(k) + D_u u(k) + D_{aux} w(k) + D_{aff} \quad (7b)$$

$$E_x x(k) + E_u u(k) + E_{aux} w(k) \le E_{aff} \quad (7c)$$

where $x = [x_c \quad x_d]^{\mathrm{T}}$, $x_c \in \mathbb{R}^{n_x^c}$, $x_d \in \mathbb{R}^{n_x^d}$, and $u = [u_c \quad u_d]^{\mathrm{T}}$, $u_c \in \mathbb{R}^{n_u^c}$, $u_d \in \mathbb{R}^{n_u^d}$, $n_i := n_i^c + n_i^d$ $(i = \{x, u\})$ represent states (both continuous and discrete) and inputs (both continuous and discrete), respectively. $w \in \mathbb{R}^{n_w^c + n_w^d}$ represent continuous and discrete auxiliary variables which are necessary to convert propositional logic into linear inequalities represented by (7c).

As discussed earlier in the Section I, nonlinear models may be effectively approximated by the use of the multiple-linear models. As such, switching between these multiple-linear models is vital in ensuring that the behavior of the nonlinear model is closely approximated. This switching can be easily facilitated by adopting the MLD modeling approach.

The MLD model incorporating all the linear models was formulated by adapting the piecewise affine (PWA) approach described in [12]. The two states (x_1, x_2) and system input (u) along with their respective bounds are the same as reported in Section II-A. The steady-state operating points of the system were identified as x_{s1} = [0.856, 0.886], x_{s2} = [0.5528, 2.7517], and x_{s3} = [0.2353, 4.705] [12]. The system state-space was subsequently partitioned into three regions with each region containing a stable operating point. The partitioned regions were defined as R_1 = [0.78, 1]×[0, 6], R_2 = [0.35, 0.78]×[0, 6], and R_3 = [0, 0.35]×[0, 6], receptively.

Auxiliary variables δ_1, δ_2 and δ_3 were used to detect the current state of the system as shown below in (8a)-(8c) In conjunction with δ_4 and δ_5 in (8d)-(8e), the exact region of operation in the state-space was identified. Based on this, the appropriate linear model from the bank of multiple-linear models was selected for performing the state update. A total of 11 auxiliary variables were defined. The remaining 6 auxiliary variables were used to perform the state updates for the two continuous states depending on the region of operation.

$$\delta_1(k) = 1 \Leftrightarrow x_1(k) \le 0.35 \quad (8a)$$

$$\delta_2(k) = 1 \Leftrightarrow x_2(k) \le 0.78 \quad (8b)$$

$$\delta_3(k) = 1 \Leftrightarrow x_1(k) \le 1 \quad (8c)$$

$$\delta_4(k) = \delta_2(k) \wedge \sim \delta_1(k) \quad (8d)$$

$$\delta_5(k) = \delta_3(k) \wedge \sim \delta_2(k) \wedge \sim \delta_1(k) \quad (8e)$$

A unified MLD model based on combining all the linear models can be obtained from (7a)-(7c). The whole process of obtaining this unified MLD model is automated using the software package Hybrid System Description Language (HYSDEL) [14]. HYSDEL generates all the matrices in the MLD model from a high level description of the system behavior. The matrices of the MLD model are omitted due to the space restrictions.

The design of a MLD-MPC controller for (1) to track a desired reference x_s is formulated as follows:

$$\min_{u_i} \sum_{i=0}^{N-1} (Q \parallel x_i - x_{i,s} \parallel_p + R \parallel \Delta u_i \parallel_p) \tag{9}$$

subject to $(1),\ (2),\ (3)$, and $(7a)$-$(7c)$ for $i = 0, 1, \ldots, N-1$

where $x_{i,s}$ denotes the set-point (reference) at instant i; $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ denote positive definite, symmetric weighting matrices; $\Delta u_i = u_i - u_{i-1}$ denotes the control increment, $N(\geq 1)$ denotes the prediction horizon, and $\parallel . \parallel_p$ represents a standard vector p-norm with $p = \infty$.

III. Simulation results and discussion

In this section, we present our studies on a CSTR system (1) in two different cases. In the first case, we briefly justify our selection of the MLD model over the single linearized model. This is demonstrated with the help of a simple model validation study. Next, we present the set-point tracking problem under MLD-MPC and NMPC schemes while comparing the benefits and trade-offs among them.

A. Model validation study

To demonstrate the advantages of the MLD approach, we compare the original nonlinear model of a CSTR system in (1) against a single linearized model (SLP) and the MLD model developed in Section II-C. We achieve this by perturbing the model in (1) under a uniform random sequence of inputs (u's) starting with a steady-state point (x_{s1}) reported in Section II-C with a sampling time of 0.1 second[1]. The SLP was developed by linearizing the model (1) around steady-state operating point x_{s1}. Figure 1 depicts the evolution of states, C_A and T. It is worth nothing that the single linearized model (dotted line) performs poorly against the MLD model (dash-dotted line) which closely approximates the original nonlinear model (blue solid line). To further justify the choice of the MLD model, we compare the corresponding approximation error (e) between the two schemes computed with the following performance criterion:

$$\text{e} = \parallel f(x, u) - f_{\text{SLP / MLD}}(x, u) \parallel_p \tag{10}$$

where $f_{\text{SLP / MLD}}$ are the evolutions with SLP and MLD models under a uniform random sequence of inputs (u's), and $\parallel . \parallel_p$ represents a standard vector p-norm with $p = 1$.

Table I report the numerical results for the approximation error. The state approximation error with MLD model is observed to be 40-96 % lesser than SLP.

[1]The reported model validation can be performed with various steady-state points and random input sequences to bring out the strength of the MLD modeling approach. However, due to a space constraints, we limit ourselves to model validation along the lines of a single equilibrium point.

TABLE I
Comparison of approximation errors between single linearized model (SLP) and MLD model.

Modeling scheme	Approximation error (e)	
	C_A	T
SLP	0.029	0.181
Multiple-MLD	0.001	0.004

B. Set-point tracking study

We simulate both MLD and NMPC schemes for the set-point tracking control problem, which involves multiple set-point changes for x_2. For an NMPC scheme, the nonlinear model in (1) is used as a system model for the simulation, and the NMPC control law is derived by solving an NLP of the form (6). Similarly, for the MLD-MPC scheme, the formulation (7a)-(7c) is used as a system model for the simulation and the MPC control law is derived by solving (9). Further, the solution for the updated states is computed based on the set of given initial conditions and the first optimal control input derived based on the MPC control law.

Table II shows the parameter values adopted for the simulation:

TABLE II
Paramters for set-point tracking simulations (Figures 1 and 2).

Paramters	NMPC	MLD-MPC
sampling time (sec)	0.3	0.1
N	3	3
Q	diag(0 0.1)$^{\mathrm{T}}$	diag(0 0.1)$^{\mathrm{T}}$
R	0.1	0.1

For the simulation studies, the MLD-MPC scheme is formulated using YALMIP. Subsequently, *intlinprog* solver from the MATLAB Optimization Toolbox was used to solve resulting MILPs at each sampling instant. Similarly, for the NMPC scheme, we follow the approach based on Bernstein global optimization reported in [13] to solve NLPs at each sampling instant.

Figures 2a and 2b show the evolution of states starting from an initial operating point ($C_A = 0.5$, $T = 2.7$) followed by a series of set-point transitions. Specifically, we compare the results between NMPC and MLD-MPC schemes for a CSTR system. We observed a smooth transition for both states under multiple set-point changes under NMPC scheme. On the other hand, with the MLD-MPC scheme, a slow response was observed for the first set-point change (samples 0-20). The settling time remains almost the same in both cases. Further, we notice an improvement in the settling-time for the second and third set-point changes with the MLD-MPC scheme. However, this could be due to a difference in the sampling time between the two schemes. It is worth nothing that in the case of NMPC scheme, we are actually solving *nonconvex* optimization problems (due to the nonlinear nature of a CSTR model). As such, it is justifiable to consider a sampling time of 0.3 seconds from the practical point of view.

Figure 2c illustrates the control performance of the two schemes. It is apparent that the MLD-MPC stabilizes the system very quickly. On the other hand, NMPC scheme takes some time to achieve the stable state. This is evident from Figure 2d, where the NMPC scheme takes an average 0.1 seconds to solve optimization problems (almost 50 % slower than the MLD-MPC scheme). However, this is a trade-off to be paid at the cost of arriving at a globally optimal solution. Similarly, to observe the consistency of our findings reported above, we choose a different series of set-point transitions whose results are reported in Figure 3.

In summary, we note that the MLD model provides a good approximation to the original nonlinear CSTR model with promising results for the set-point tracking control problem. Although it would be interesting to compare the behavior of MLD-MPC with NMPC under short and long term transients, we restrict ourselves to the results presented earlier owing to space limitations.

IV. Conclusions

This work demonstrated the MPC problem of a CSTR system. Since the original CSTR system is highly nonlinear, the optimal control problem naturally becomes a nonlinear programming problem in the MPC framework. To circumvent this issue, the present work investigated a well-known multiple-model approach in the MLD formalism. The benefits accrued by the multiple-model approach over the classical single linearized model approach in terms of approximating the behavior of the original nonlinear model were demonstrated. Later, different simulation studies for the set-point tracking problem were conducted to compare MLD-MPC and NMPC schemes. The simulation results demonstrated that the MLD-MPC approach holds good promise as a viable alternative for the NMPC scheme. The main benefit of the approach noted was that solving time for the optimization problems at each sampling instant was significantly reduced.

Overall, it may be concluded that the NMPC scheme has good potential to deliver optimal solutions. However, this comes at the cost of computational time, and is the price to be paid in order to solve NLPs. While the MLD-MPC scheme delivered promising results in our studies, it would be interesting to observe how the approach performs under transients in comparison to the NMPC scheme.

References

[1] J. M. Maciejowski, *Predicitve control with constraints*. UK, Harlow: Prentice Hall, 2000.

[2] E. F. Camacho and C. Bordons, *Model predictive control, 2^{nd} ed.* London: Springer-Verlag, 2004.

[3] M. L. Darby and M. Nikolaou, "MPC: Current practice and challenges," *Control Engineering Practice*, vol. 20, no. 4, pp. 328–342, 2012.

[4] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, pp. 733–764, 2003.

[5] T. A. Badgwell and S. J. Qin, "Model-predictive control in practice," *Encyclopedia of Systems and Control*, pp. 1–6, 2014.

[6] C. V. Rao and J. B. Rawlings, "Linear programming and model predictive control," *Journal of Process Control*, vol. 10, no. 2-3, pp. 283–289, 2000.

[7] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[8] B. Aufderheide and B. W. Bequette, "Extension of dynamic matrix control to multiple models," *Computers and Control Engineering*, vol. 27, no. 8-9, pp. 1079–1096, 2003.

[9] F. Martinsen, L. T. Biegler, and B. A. Fossa, "A new optimization algorithm with application to nonlinear MPC," *Journal of Process Control*, vol. 14, no. 8, pp. 853–865, 2004.

[10] R. Findeisen, F. Allgöwer, and L. Biegler. Assessment and future directions of nonlinear model predictive control, *Lecture Notes in Control and Information Sciences*: Springer-Verlag, 2007.

[11] J. D. Hedengren, R. A. Shishavana, K. M. Powell, and T. F. Edgar, "Nonlinear modeling, estimation and predictive control in APMonitor," *Computers and Chemical Engineering*, vol. 70, no. 5, pp. 133–148, 2014.

[12] L. Özkan, M. V. Kothare, and C. Georgakis, "Control of a solution copolymerization reactor using multi-model predictive control," *Chemical Engineering Science*, vol. 58, pp. 1207–1221, 2003.

[13] B. V. Patil, J. Maciejowski, and K. V. Ling, "Nonlinear model predictive control based on Bernstein global optimization with application to a nonlinear CSTR," *15^{th} Annual European Control Conference*, Aalborg, Denmark, 2016 (Accepted).

[14] F. D. Torrisi and A. Bemporad, "HYSDEL-a tool for generating computational hybrid models for analysis and synthesis problems," *IEEE Transactions on Control System Technology*, vol. 12, pp. 235–249, 2004.

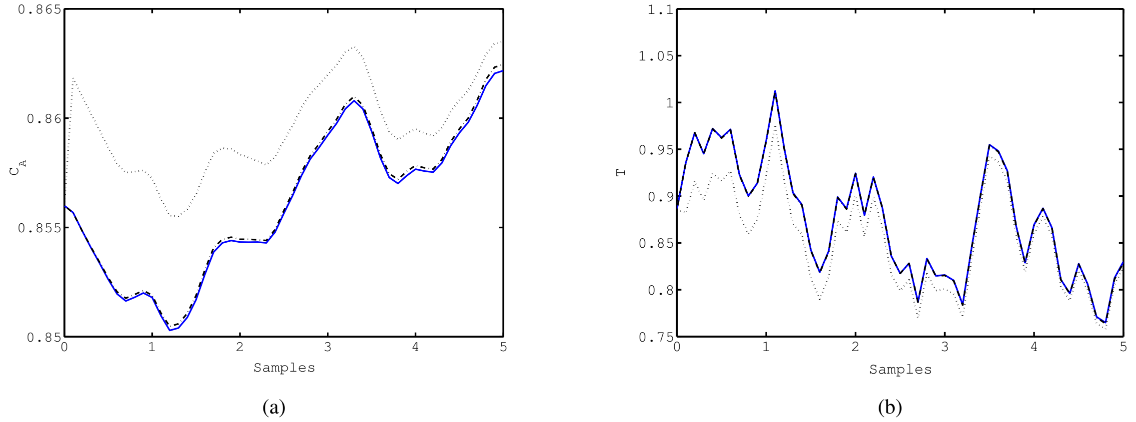

Fig. 1. Model validation: evolution of states $x = [C_A \quad T]^{\mathrm{T}}$ with single linear model (dotted line), MLD model (dash-dotted line), nonlinear model (blue solid line).

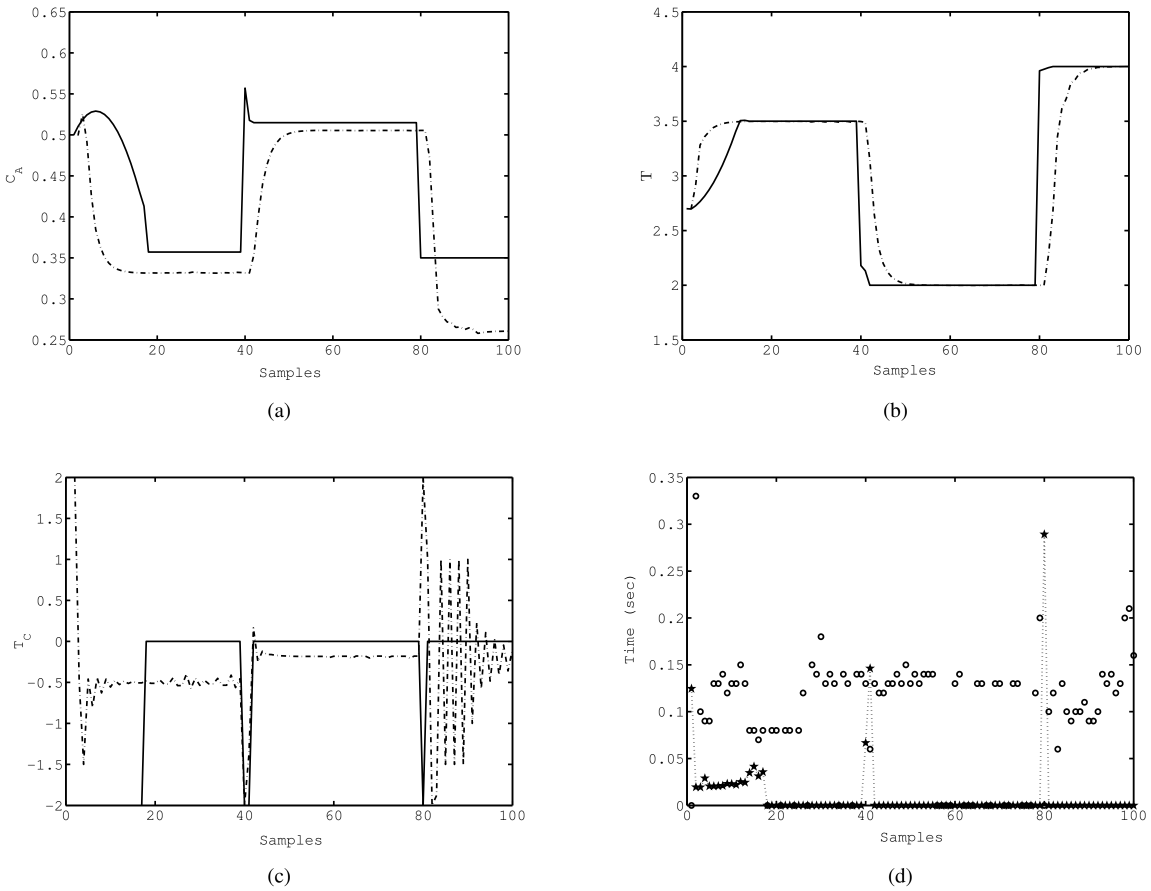

Fig. 2. Close-loop results under multiple set-point changes in T. The set-points are as follows: at 0 sec 3.5, at 40 sec 2, and at 80 sec 4. (a) C_A profile for MLD-MPC (solid line) and NMPC (dash-dotted line). (b) T profile for MLD-MPC (solid line) and NMPC (dash-dotted line). (c) T_C profile for MLD-MPC (solid line) and NMPC (dash-dotted line). (d) Computation times for MLD-MPC (star-dotted line) and NMPC (circles).

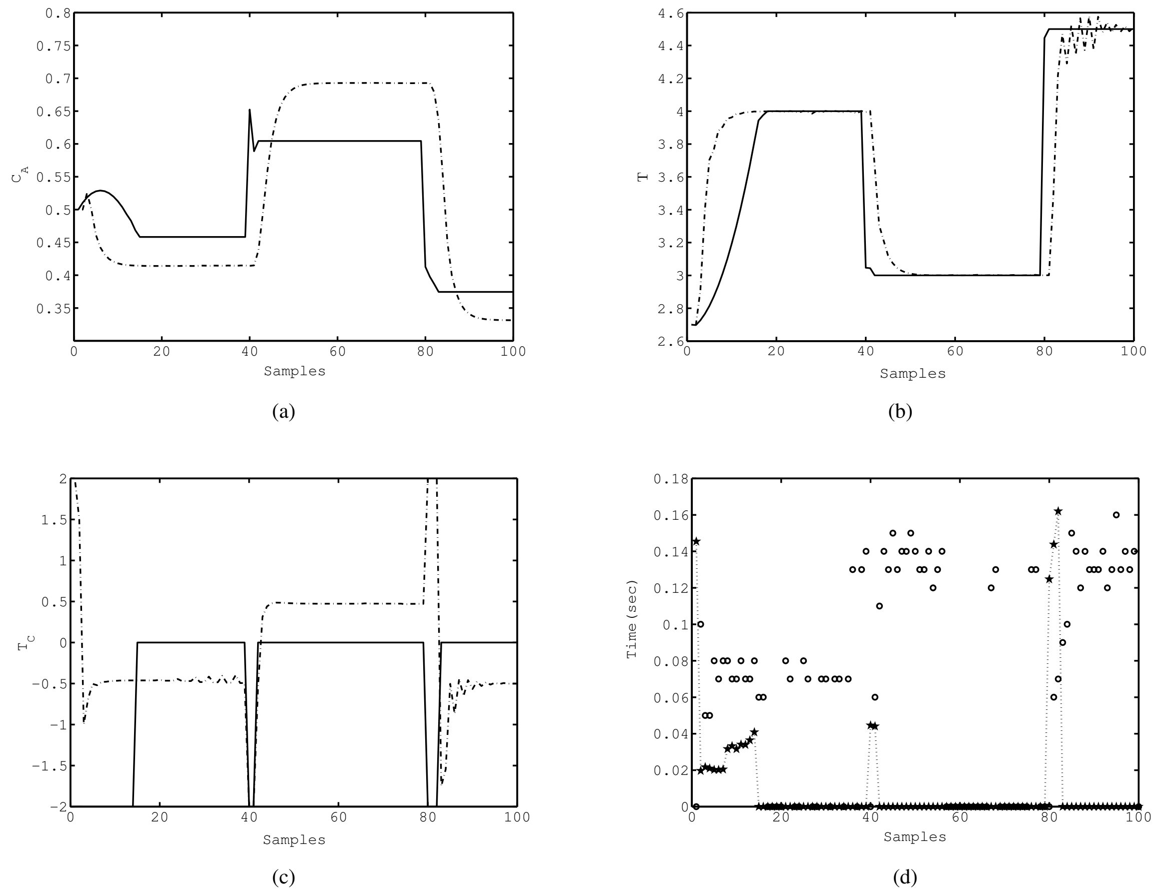

Fig. 3. Close-loop results under multiple set-point changes in T. The set-points are as follows: at 0 sec 4, at 40 sec 3, and at 80 sec 4.5. (a) C_A profile for MLD-MPC (solid line) and NMPC (dash-dotted line). (b) T profile for MLD-MPC (solid line) and NMPC (dash-dotted line). (c) T_C profile for MLD-MPC (solid line) and NMPC (dash-dotted line). (d) Computation times for MLD-MPC (star-dotted line) and NMPC (circles).