Multivariable Constrained Adaptive Predictive Control based on Closed-loop Subspace Identification

Xiaosuo Luo, Xueshan Lin

Abstract**—**In order to deal with nonlinear, time-varying, and multivariable constrained characteristics in closed-loop industrial processes, a multivariable constrained adaptive predictive control (CAPC) method based on closed-loop subspace identification is proposed. The state-space model is obtained through the closed-loop subspace identification algorithm, which is regarded as the system model. The algorithm is implemented online to update the R matrix with a receding window. By comparing the prediction errors before and after updating, it considers whether or not to update the system model. The model is then used to design the model predictive controller, which involves the solution of a quadratic program solving multivariable constraints. This paper presents a comparison between the performance of the proposed control method when applied to a 2-CSTR system, and that of an open-loop subspace CAPC method. The superiority of the proposed method is illustrated by the simulation results.

Keywords**—**Closed-loop Subspace Identification, Multivariable constraint, Adaptive Predictive Control, 2-CSTR

I. Introduction

Model predictive control (MPC) has been an attractive subject in the control theory field for decades. It has become more established in the industry as the one of the choices for control architecture, especially with the improvement of computational capabilities of processors [1-4]. The traditional industrial predictive control is based on the input-output model, and includes both parametric and nonparametric models. In order to improve the control performance, a state space model should be adopted so the modern filter theory and the design method of controllers developed in recent years can play a role [5-7]. However, MPC was unable to obtain the accurate state-space model among the complex industrial targets due to the limitation of the identified method. The subspace identification method has changed this situation substantially; the control workers may be completely relieved from the tedious modeling via this mechanism. The accurate state-space model can be obtained when there is enough process input and output data [8-9].

In practice, it is often necessary to perform identification experiments on systems operating in a closed loop. This is especially true when open-loop experiments are not allowed due to reasons of safety (unstable processes) or production (undesirable open-loop behavior). System identification from closed-loop data is thus a relevant topic [10]. The identification of the subspace matrices from closed-loop data has also received attention from several researchers [11]. It has been found that the regular open-loop subspace identification algorithm yields a biased estimate when applied to closed-loop data [12]. This paper expands upon some recent ideas for developing subspace methods that can perform well on data collected in a closed-loop condition. Here, a method that aims at minimizing the prediction errors in several approximate steps is proposed. The steps involve using constrained least squares estimation on models with different degrees of structure, such as the block-Toeplitz model, and reduced rank matrices.

The batch form is employed for data processing in a basic subspace identification algorithm. The acquired input-output data is processed as a whole. It is adverse to the online implementation and adaptive identified application. Due to the time-varying nature of industrial processes, online model assessment is necessary for determining whether model re-identification is needed [13]. At present, there are two ways to conduct online adaptive subspace identification; one is the recursive identification method [14], and the other is the receding window method [15]. The primary obstacle to the implementation of adaptive subspace identification is developing online QR and SVD decomposition algorithms. A QR decomposition procedure is proposed by simultaneously applying data updating and downdating, and outperformed traditional algorithms in terms of computational efficiency. Multivariable constraints arise due to physical limitations, quality specifications, safety concerns, and limiting the wear of the equipment [16]. The prediction capability of MPC is useful in anticipating constraint violations and correcting them in an

This work was supported by the Major State Basic Research Development Program 973 (No.2012CB215202), the National Natural Science Foundation of China (No.61134001, 61773081), Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University), Ministry of Education, Chongqing Research Program of Basic Research and Frontier Exploration (No.cstc2018jcyjAX0082), The science and technology research program of Chongqing Education Commission (No.KJQN20190310 and KJQN201803112) and Research Center of Intelligent Building Engineering Technology (No.ZBPT201901).

Xiaosuo Luo is with the Chongqing College of Electronic Engineering, Chongqing 401331, China and School of Automation, Chongqing University, Chongqing 400044, China (e-mail: lock86@126.com).

Xueshan Lin is with the Chongqing College of Electronic Engineering, Chongqing 401331, China (e-mail: 826561356@qq.com).

appropriate way [17].

In this paper, by online updating the R matrix with a receding window, the state-space model can be obtained from closed-loop data. It considers whether or not to update the system model by comparing the prediction errors before and after updating. The model predictive controller is then designed, and solves an optimization problem online; MPC tools allow for the solution of control problems with multivariable constraints on input and output. The salient features and the main contribution of the proposed method can be summarized as:

- It is applicable for closed-loop systems. The closed-loop model can be obtained with QR and SVD factorization.
- It provides an attractive method for online adaptive implementation. The adaptive model is obtained by updating the modified R matrix online.
- It is an acceptable solution to the multivariable constraint problem via quadratic programming.

The paper is organized as follows. In Section 2, the closed-loop subspace identification algorithm is given. Section 3 provides the online adaptive implementation. The constrained adaptive predictive control method is highlighted in Section 4. The simulation results are presented and discussed in Section 5. Finally, Section 6 presents the conclusions.

II. The Closed-loop Subspace Identification Algorithm

Given a general state-space system of order n in innovations form:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ke(k) \\ y(k) &= Cx(k) + Du(k) + e(k) \end{aligned} \tag{1}$$

with input $u(k) \in \mathbb{R}^m$, output $y(k) \in \mathbb{R}^l$, state $x(k) \in \mathbb{R}^n$, Kalman filter gain K, and innovation sequence $e(k)$.

The aim is to identify the system matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{l\times n}$, and $D \in \mathbb{R}^{l\times m}$ using the finite closed-loop input-output data sequences $\{u(k)\}_{k=1}^{N}$ and $\{y(k)\}_{k=1}^{N}$.

Let us first introduce some preliminary notation. The vectors of stacked inputs, outputs, and innovations are defined as

$$\begin{aligned} y_f(k) &= \begin{bmatrix} y^{\mathrm{T}}(k) & y^{\mathrm{T}}(k+1) & \cdots & y^{\mathrm{T}}(k+f-1) \end{bmatrix}^{\mathrm{T}} \\ u_f(k) &= \begin{bmatrix} u^{\mathrm{T}}(k) & u^{\mathrm{T}}(k+1) & \cdots & u^{\mathrm{T}}(k+f-1) \end{bmatrix}^{\mathrm{T}} \\ e_f(k) &= \begin{bmatrix} e^{\mathrm{T}}(k) & e^{\mathrm{T}}(k+1) & \cdots & e^{\mathrm{T}}(k+f-1) \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{2}$$

where $f > n$ is an integer chosen by the user. In addition, define

$$\begin{aligned} \tilde{A} &= A - KC \\ \tilde{B} &= B - KD \end{aligned} \tag{3}$$

It is well known that we can rewrite Eq. (1) as follows:

$$\begin{aligned} x(k+1) &= \tilde{A}x(k) + \tilde{B}u(k) + Ky(k) \\ y(k) &= Cx(k) + Du(k) + e(k) \end{aligned} \tag{4}$$

Let us now turn to the idea of the method of this paper. First, we will use Eq. (4) to form the subspace data model:

$$y_f(k) = \tilde{\Gamma}x(k) + \tilde{H}u_f(k) + \tilde{G}y_f(k) + e_f(k) \tag{5}$$

where

$$\tilde{\Gamma} = \begin{bmatrix} C \\ C\tilde{A} \\ \vdots \\ C\tilde{A}^{f-1} \end{bmatrix}, \quad \tilde{H} = \begin{bmatrix} D & 0 & \cdots & 0 \\ C\tilde{B} & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C\tilde{A}^{f-2}\tilde{B} & C\tilde{A}^{f-3}\tilde{B} & \cdots & D \end{bmatrix},$$
$$\tilde{G} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ CK & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C\tilde{A}^{f-2}K & C\tilde{A}^{f-3}K & \cdots & 0 \end{bmatrix} \tag{6}$$

Consider the Hankel representation:

$$Y_{k,i,j} = \Gamma_i X_{k,j} + H_i U_{k,i,j} + G_i Y_{k,i,j} + E_{k,i,j} \tag{7}$$

where

$$\Gamma_i = \begin{bmatrix} C \\ C\tilde{A} \\ \vdots \\ C\tilde{A}^{i-1} \end{bmatrix}, \quad H_i = \begin{bmatrix} D & 0 & \cdots & 0 \\ C\tilde{B} & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C\tilde{A}^{i-2}\tilde{B} & C\tilde{A}^{i-3}\tilde{B} & \cdots & D \end{bmatrix},$$
$$G_i = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ CK & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C\tilde{A}^{i-2}K & C\tilde{A}^{i-3}K & \cdots & 0 \end{bmatrix} \tag{8}$$

and the Hankel matrices from input-output data are $Y_{k,i,j} = \begin{bmatrix} y_k & y_{k+1} & \cdots & y_{k+j-1} \\ y_{k+1} & y_{k+2} & \cdots & y_{k+j} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k+i-1} & y_{k+i} & \cdots & y_{k+j+i-2} \end{bmatrix}$ and $U_{k,i,j} = \begin{bmatrix} u_k & u_{k+1} & \cdots & u_{k+j-1} \\ u_{k+1} & u_{k+2} & \cdots & u_{k+j} \\ \vdots & \vdots & \ddots & \vdots \\ u_{k+i-1} & u_{k+i} & \cdots & u_{k+j+i-2} \end{bmatrix}$. The state vector $X_{k,j} = \begin{bmatrix} x_k & x_{k+1} & \cdots & x_{k+j-1} \end{bmatrix}$. $E_{k,i,j}$ is the Hankel matrix of the noise value.

It is now necessary to define two Hankel matrices for inputs and outputs: one of past values, given by $U_{0,i,j}$ and $Y_{0,i,j}$, and another of future values, given by $U_{i,2i,j}$ and $Y_{i,2i,j}$. The instrumental variable in the subspace method is chosen as a combination of past inputs and outputs:

$$W = \begin{bmatrix} U_{0,i,j} \\ Y_{0,i,j} \end{bmatrix} \tag{9}$$

Multiplying Eq. (7) from the right by projection operators $\Pi_{U_{k,i,j}^{\perp}}$ and Π_W to remove the terms $H_i U_{k,i,j}$ and $E_{k,i,j}$ results in:

$$\lim_{N\to\infty} \frac{1}{N} Y_{i,2i,j} \Pi_{U_{k,i,j}^{\perp}} \Pi_W = \lim_{N\to\infty} \frac{1}{N} \Gamma_i X_{i,j} \Pi_{U_{k,i,j}^{\perp}} \Pi_W \tag{10}$$

This is solved in the usual way using the RQ decomposition:

$$\begin{bmatrix} U_{i,2i,j} \\ W \\ Y_{i,2i,j} \end{bmatrix} = \begin{bmatrix} U_{i,2i,j} \\ U_{0,i,j} \\ Y_{0,i,j} \\ Y_{i,2i,j} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix} \begin{bmatrix} Q_1^{\mathrm{T}} \\ Q_2^{\mathrm{T}} \\ Q_3^{\mathrm{T}} \\ Q_4^{\mathrm{T}} \end{bmatrix} \tag{11}$$

We can therefore obtain:

$$O_i = \begin{bmatrix} R_{42} & R_{43} \end{bmatrix} \begin{bmatrix} Q_2^{\mathrm{T}} \\ Q_3^{\mathrm{T}} \end{bmatrix} = \Gamma_i \hat{X}_{i,j} \tag{12}$$

To find an estimate of the column spaces of Γ_i and $\hat{X}_{i,j}$, we take the SVD of O_i:

$$O_i = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} V_1^{\mathrm{T}} \\ V_2^{\mathrm{T}} \end{bmatrix} \tag{13}$$

Given the linear system

$$\begin{bmatrix} \hat{X}_{i+1,j} \\ Y_{k,i,j} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{X}_{i,j} \\ U_{k,i,j} \end{bmatrix} + \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} \tag{14}$$

the least square method is used to obtain

$$\begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} = \arg \min_{A,B,C,D} \left\| \begin{bmatrix} \hat{X}_{i+1,j} \\ Y_{k,i,j} \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{X}_{i,j} \\ U_{k,i,j} \end{bmatrix} \right\|_F^2 \tag{15}$$

and the system matrices (A, B, C, D) can be estimated. However, it is performed “offline.” To implement “online,” we should update the R matrix.

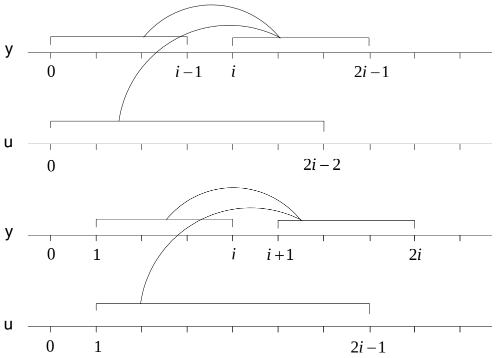

Fig. 1 The online implementation of a receding window

III. Online Adaptive Implementation

In adaptive control, model parameters are required to be updated online. In this paper, the updating law is based on a receding window, as illustrated in Fig. 1, and the explication can be seen in [15].

Let

$$\mathrm{G} = [G_1 \ \ G_2 \ \cdots \ G_j] \in \mathbb{R}^{4i \times j} \tag{16}$$

Where

$$G_c = \begin{bmatrix} u_c \\ w_c \\ y_c \end{bmatrix} = [g_{1c} \ \ g_{2c} \cdots g_{4ic}]^{\mathrm{T}}, \ 1 \le c \le j \tag{17}$$

Suppose a new set of input and output data vector is

$$G_{j+1} = \begin{bmatrix} u_{j+1} \\ w_{j+1} \\ y_{j+1} \end{bmatrix} = [g_{1(j+1)} \ \ g_{2(j+1)} \cdots g_{4i(j+1)}]^{\mathrm{T}} \tag{18}$$

In the receding window approach, the input and output data of the current process are augmented and the oldest data is discarded. The new data G_{j+1} is then added to the G matrix in the last column, and the new G' is

$$\mathrm{G}' = [G_2 \ \ G_3 \ \cdots \ G_{j+1}] \in \mathbb{R}^{4i \times j} \tag{19}$$

The key of QR decomposition in the receding window is to obtain the decomposition results of G' from the results of G via a series of appropriate transformations. The QR decomposition of G' is

$$\mathrm{G}' = \mathrm{R}'Q'^{\mathrm{T}} \tag{20}$$

Introduce intermediate matrices $\mathrm{R}^{+} = [R_1 \;\; R_2 \;\; \cdots \;\; R_{j+1}] \in \mathbb{R}^{4i\times(j+1)}$ and $A = \begin{bmatrix} \mathrm{R} & R_{j+1} \end{bmatrix}$, $B = \begin{bmatrix} R_1 & \mathrm{R}' \end{bmatrix}$. Define a_i and b_i as the i rows of A and B, respectively: $a_i = [r_{i1} \; r_{i2} \cdots r_{i(j+1)}]$, $b_i = [r_{i1} \; r'_{i2} \cdots r'_{i(j+1)}]$.

According to the QR decomposition, then

$$\begin{aligned} \mathrm{R}^{+} = [R_1 \;\; R_2 \;\; \cdots \;\; R_{j+1}] &= \begin{bmatrix} \mathrm{R}Q^{\mathrm{T}} & R_{j+1} \end{bmatrix} = A \begin{bmatrix} Q^{\mathrm{T}} & 0 \\ 0^{\mathrm{T}} & I \end{bmatrix} \\ &= \begin{bmatrix} R_1 & \mathrm{R}'Q'^{\mathrm{T}} \end{bmatrix} = B \begin{bmatrix} I & 0^{\mathrm{T}} \\ 0 & Q'^{\mathrm{T}} \end{bmatrix} \end{aligned} \tag{21}$$

According to the orthogonality of Q and Q', we can obtain

$$AA^{\mathrm{T}} = BB^{\mathrm{T}} \tag{22}$$

by expanding Eq. (21), the variables of R' can be obtained:

$$r'_{ii} = \left[a_i a_i^{\mathrm{T}} - r_{i1}^2 - \sum_{k=1}^{i-1} r'^2_{ik} \right]^{\frac{1}{2}} \tag{23}$$

$$r'_{ij} = \frac{1}{r'_{jj}} \left[a_i a_j^{\mathrm{T}} - r_{i1} r_{j1} - \sum_{k=1}^{j-1} r'_{ik} r'_{jk} \right] \tag{24}$$

The above method adds new data at each time while simultaneously deleting the oldest data. It would result in the whole window losing information, and would produce greater prediction error [18]. The existence of online disturbance influences identification precision. Therefore, we must decide whether or not to use the new R' matrix to update the model with the inspection strategy of model precision. This strategy depends on the prediction error.

Calculate the following prediction error before updating the model:

$$\mathrm{ess}_1 = \left\| y_k - \hat{y}_{k|k-1} \right\| \tag{25}$$

where y_k is the process output at k time, $\hat{y}_{k|k-1}$ is the output at $k-1$ time that predicts k time before updating, and $\|\cdots\|$ is the 2-norm.

Similarly, the prediction error after updating the model ess_2 can be also introduced:

$$\mathrm{ess}_2 = \left\| y_k - \hat{y}'_{k|k-1} \right\| \tag{26}$$

where $\hat{y}'_{k|k-1}$ is the output at $k-1$ time that predicts k time after updating.

While $\mathrm{ess}_1 \le \mathrm{ess}_2$, maintain the R matrix and the system model invariably. However, when $\mathrm{ess}_1 > \mathrm{ess}_2$, use the new R' matrix to update the system model. At the next sample time, when the new data arrives, recycle the above process.

IV. Multivariable constrained Adaptive Predictive Control

The model predictive control strategy is an interesting alternative for the control design method. It allows for the incorporation of not only reference trajectories, but also multivariable constrains control signals and outputs. It is used to implement an indirect adaptive control method using the online subspace algorithm with MPC tools. These MPC tools allow for the solution of control problems with multivariable constraints. A block diagram of the proposed system is illustrated in Fig. 2.

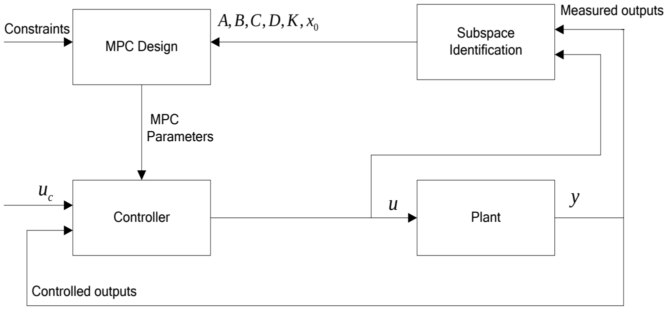

Fig. 2 The structure of the multivariable constrained adaptive predictive control system

Constraints on the inputs and outputs of the system are based on the physical limits of the process being controlled. The limits fall into the following categories:

$$\begin{aligned} u_{\min} &\le \;\; u \;\; \le u_{\max} \\ \Delta u_{\min} &\le \; \Delta u \; \le \Delta u_{\max} \\ y_{\min} &\le \;\; y \;\; \le y_{\max} \end{aligned} \tag{27}$$

as the purpose is to optimize the cost function through the set of future control variables ΔU_f, the effects of these constraints on ΔU_f must be determined.

Given $F_m = \begin{bmatrix} I_m & I_m & I_m & I_m \end{bmatrix}^{\mathrm{T}}, F_l = \begin{bmatrix} I_l & I_l & I_l & I_l \end{bmatrix}^{\mathrm{T}}$, we can write Eq. (27) as a model predictor:

$$\begin{aligned} F_m u_{\min} &\le \;\; U_f \;\; \le F_m u_{\max} \\ F_m \Delta u_{\min} &\le \; \Delta U_f \; \le F_m \Delta u_{\max} \\ F_l y_{\min} &\le \;\; Y_f \;\; \le F_l y_{\max} \end{aligned} \tag{28}$$

After determining the constraints on ΔU_f, these constraints can be incorporated into a matrix inequality:

$$\begin{bmatrix} T_m \\ -T_m \\ I_m \\ -I_m \\ T_l L_u \\ -T_l L_u \end{bmatrix} \Delta U_f \le \begin{bmatrix} F_m u_{\max} - F_m u_k \\ F_m u_k - F_m u_{\min} \\ F_m \Delta u_{\max} \\ -F_m \Delta u_{\min} \\ F_l y_{\max} - T_l L_w \Delta W_p - F_l y_k \\ T_l L_w \Delta W_p + F_l y_k - F_l y_{\min} \end{bmatrix} \tag{29}$$

where $T_m = \begin{bmatrix} I_m & 0 & \cdots & 0 \\ I_m & I_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_m & I_m & I_m & I_m \end{bmatrix}$, $T_l = \begin{bmatrix} I_l & 0 & \cdots & 0 \\ I_l & I_l & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_l & I_l & I_l & I_l \end{bmatrix}$, and $\Delta W_p = \begin{bmatrix} \Delta y_{k-P+1} \\ \vdots \\ \Delta y_k \\ \Delta u_{k-M+1} \\ \vdots \\ \Delta u_k \end{bmatrix}$, with P and M being the prediction and control horizons, respectively. L_w and L_u are the subspace prediction matrices, and can be obtained from the least square method [19]:

$$\min_{L_w, L_u} \left\| \Delta Y_f - (L_w, L_u) \begin{pmatrix} \Delta W_p \\ \Delta U_f \end{pmatrix} \right\|_F^2 \tag{30}$$

where ΔY_f is the set of future output variables.

We are looking for a cost function of a standard quadratic objective

$$J = \frac{1}{2} \Delta U_f^{\mathrm{T}} h \Delta U_f + \Delta U_f^{\mathrm{T}} f \tag{31}$$

where $h = (T_l L_u)^{\mathrm{T}} Q (T_l L_u) + R$, $f = (T_l L_u)^{\mathrm{T}} Q \left(T_l L_w \Delta W_p + F_l y(k) - Y_d \right)$, Q and R are weighting matrices, and Y_d is the reference output. Equation (24) also can be denoted as

$$A \Delta U_f \le b \tag{32}$$

This inequality can be solved by a quadratic programming (QP) algorithm [20], which can be defined in MATLAB as

$$\Delta U_f = quadprog(h, f, A, b) \tag{33}$$

After adding the first control variable Δu_k of ΔU_f to the previous plant input u_k, the new plant input is $u_{k+1} = u_k + \Delta u_k$. Measure the new plant output y_{k+1}, update ΔW_p, and repeat the calculation to solve the optimization.

V. Simulations

The 2-CSTR system is a system that incorporates nonlinear, time-varying, and multivariable characters. The approach of multivariable constrained adaptive predictive control is applied to a 2-CSTR process. The process is schematically presented in Fig. 3. A full description of the system can be found in [21-22]. The control problem is to maintain both tank temperatures at desired values. The inputs are two cooling-water flowrates, denoted by $u = [Q_{cw1}, Q_{cw2}]^{\mathrm{T}}$, and the outputs are two tank outlet temperatures, denoted by $y = [T_{o1}, T_{o2}]^{\mathrm{T}}$.

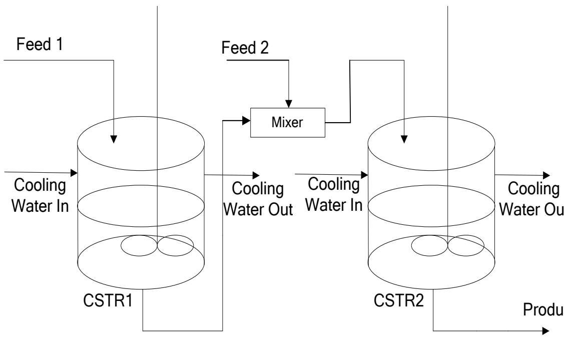

Fig. 3 2-CSTR plant

The nonlinear differential equations of this system can be described as follows:

$$\begin{aligned}
V_1 \frac{dx_1}{dt} &= -K_1 V_1 x_1 + Q_{I1}(C_{I1} - x_1) \\
V_1 \frac{dx_2}{dt} &= \Delta H K_1 V_1 x_1 + Q_{I1}(T_{I1} - x_2) - U_{a1}(x_2 - x_3) \\
V_{J1} \frac{dx_3}{dt} &= Q_{CW1}(T_{CW1} - x_3) + U_{a1}(x_2 - x_3) \\
V_2 \frac{dx_4}{dt} &= -K_2 V_2 x_4 + Q_{I2}(C_{I2} - x_4) + K_{V1} x_1 \sqrt{V_1} \\
V_2 \frac{dx_5}{dt} &= \Delta H K_2 V_2 x_5 + Q_{I2}(T_{I2} - x_5) - U_{a2}(x_5 - x_6) + K_{V1} x_2 \sqrt{V_1} \\
V_{J2} \frac{dx_6}{dt} &= Q_{CW2}(T_{CW2} - x_6) + U_{a2}(x_5 - x_6)
\end{aligned} \tag{34}$$

where $K_1 = K_0 \exp(-E / R x_2)$ and $K_2 = K_0 \exp(-E / R x_5)$. The six states are $x_1 = C_{O1}$ (outlet concentration of CSTR1), $x_2 = T_{O1}$ (outlet temperature of CSTR1), $x_3 = T_{CWO1}$ (cooling water outlet temperature of CSTR1), $x_4 = C_{O2}$ (outlet concentration of CSTR2), $x_5 = T_{O2}$ (outlet temperature of CSTR2), and $x_6 = T_{CWO2}$ (cooling water outlet temperature of CSTR2). The system stability analysis can refer to paper [21].

The input constraints are

$$\begin{aligned}
0.05 \quad &\le \quad Q_{cw1} \quad \le \quad 0.8\,[\mathrm{m}^3/\mathrm{s}] \\
0.05 \quad &\le \quad Q_{cw2} \quad \le \quad 0.8\,[\mathrm{m}^3/\mathrm{s}]
\end{aligned} \tag{35}$$

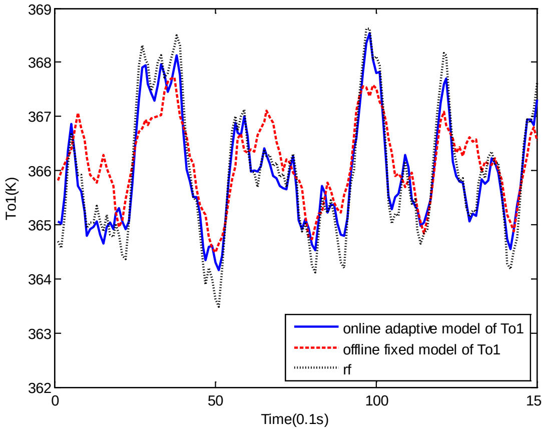

Fig. 4 The online and offline models of T_{o1}

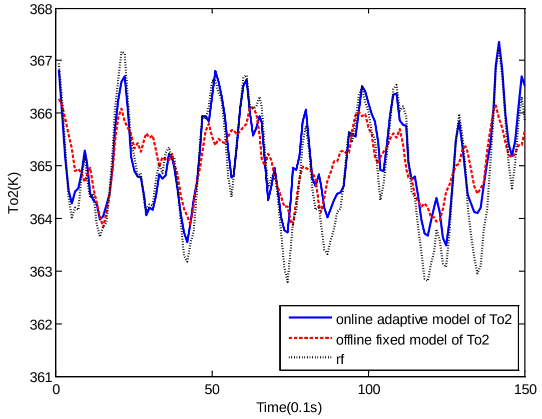

Fig. 5 The online and offline models of T_{o2}

The input Q_{cw1} was chosen to be a Gaussian random signal that shifts by $\pm 20\%$. The lengths of the samples and sampling time were set to 300 and 0.1 s, respectively, and 150 samples were used to verify the identification accuracy. The comparisons in Fig. 4 and Fig. 5 demonstrate the responses of the online adaptive model and the offline fixed model of T_{o1} and T_{o2}, respectively, where "rf" is the process output.

The online and offline prediction errors in Fig. 4 and Fig. 5 are defined as

$$\varepsilon_j = 100 * \sqrt{\frac{\sum_{i=1}^{N}\left(y_{ij}-y_{ij}^m\right)^2}{\sum_{i=1}^{N}\left(y_{ij}\right)^2}} \qquad (36)$$

where N is the number of samples, and y_{ij}^m is the "one-step-ahead" predicted output. The prediction error of the validation data set is provided in Table 1, and the adaptive algorithm reduces the error considerably.

Table 1. The prediction errors of the validation data set

Outputs	Online	Offline
T_{o1}	$\varepsilon_1 = 2.0958$	$\varepsilon_1 = 0.8013$
T_{o2}	$\varepsilon_2 = 1.8045$	$\varepsilon_2 = 0.9403$

The online adaptive models were then used to design the predictive controllers. The parameters of the proposed closed-loop subspace CAPC (CSCAPC) method are set as follows. The prediction horizon $P = 10$, and the control horizon $M = 3$. To turn the weighting matrices, $Q = 1.5 * I_{20}$ and $R = I_6$. In total, 100 seconds are conducted in the simulation. For comparison, an open-loop subspace CAPC (OSCAPC) method is used to control this system, and the parameters are selected to be the same as those set in the previous method. In the set-point test, the performance of outputs using the CSCAPC method is better than that of the OSCAPC method, as demonstrated in Fig. 6 and Fig. 7 where "Rf" is the setting trajectory.

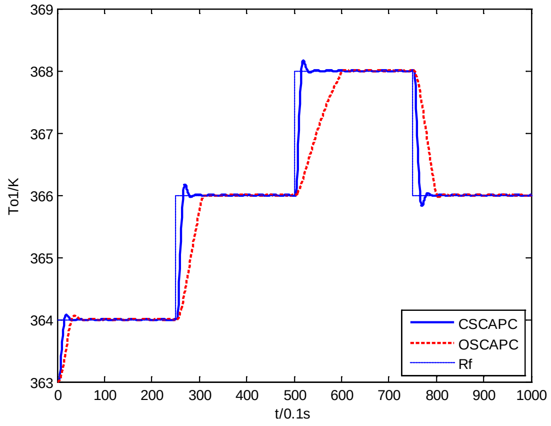

Fig. 6 The tracking performances of T_{o1}

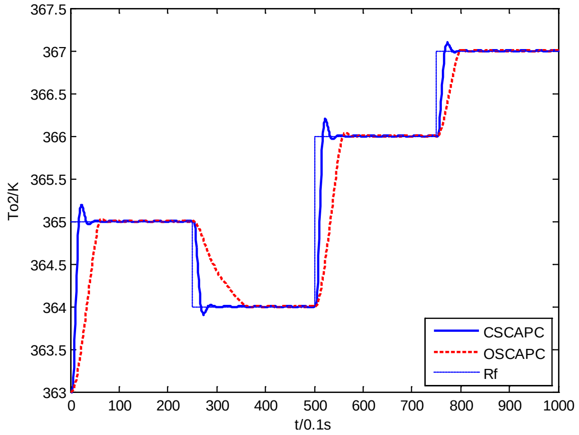

Fig. 7 The tracking performances of T_{o2}

VI. Conclusion

In this paper, the design of a multivariable constrained adaptive predictive controller based on closed-loop subspace identification is addressed. The system matrices are identified via online closed-loop subspace identification, and a constraint handling method is then introduced to handle constraints for input and output variables. The proposed controller is successfully applied to a 2-CSTR process. It is demonstrated to be more efficient than the open-loop subspace CAPC, and the excellence of the proposed controller is explained. In an extension, this controller can be further applied to 2-D system, and the application of neural network to the controller will also be an interesting and meaningful topic.

References

[1] T. Sun, Y. Pan, J. Zhang, and H. Yu, "Robust model predictive control for constrained continuous-time nonlinear systems," *International Journal of Control*, vol. 91, pp. 1–16, 2018.

[2] X. S. Luo and Y. D. Song, "Data-driven Predictive Control of Hammerstein-Wiener Systems Based on Subspace Identification," *Information Sciences*, vol. 422, pp. 447–461, 2018.

[3] Y. C. Li, P. Voulgaris, D. Stipanovic, and Z. H. Gu, "Decentralized Model Predictive Control of Urban Drainage Systems," *WSEAS Transactions on Systems and Control*, Volume 14, pp. 247-256, 2019.

[4] K. C. Schwab, L. Schräder, P. Mercorelli, and J. T. Lassen, "Sliding Mode and Model Predictive Control for Inverse Pendulum," *WSEAS Transactions on Systems and Control*, Volume 14, pp. 190-195, 2019.

[5] X. S. Luo, "Data-driven predictive control for continuous-time linear parameter varying systems with application to wind turbine," *International Journal of Control, Automation and Systems*, vol. 15, pp. 619-626, 2017.

[6] P. Chaupa, J. Novak, and P. Januska, "State space MPC Using State Observers," *International Journal of Circuits, Systems and Signal Processing*, vol. 8, pp. 9–14, 2014.

[7] X. S. Luo, "Subspace based model predictive control for linear parameter varying systems," *International Journal of Circuits, Systems and Signal Processing*, vol. 10, pp. 321–325, 2016.

[8] C. P. Yu, M. Verhaegen, and A. Hansson, "Subspace Identification of Local Systems in 1D Homogeneous Networks," *IEEE Transactions on Automatic Control*, vol. 63, no. 4, pp. 1126-1131, 2018.

[9] J. W. van Wingerden and M. Verhaegen, "Subspace identification of bilinear and LPV systems for open- and closed-loop data," *Automatica*, vol. 45, no. 2, pp. 372-381, 2009.

[10] K. Batselier, C. Y. Ko, and N. Wong, "Tensor network subspace identification of polynomial state space models," *Automatica*, vol. 95, pp. 187-196, 2018.

[11] B. Huang, S. X. Ding, and S. J. Qin, "Closed-loop subspace identification: an orthogonal projection approach," *Journal of Process Control*, vol. 15, no. 1, pp. 53-66, 2005.

[12] B. Gunes, J. W. Van Wingerden, and M. Verhaegen, "Tensor Nuclear Norm LPV Subspace Identification," *IEEE Transactions on Automatic Control*, vol. 63, pp. 3897-3903, 2018.

[13] S. J. Qin, "An overview of subspace identification," *Computers and Chemical Engineering*, vol. 30, no. 10-12, pp.1502-1513, 2006.

[14] K. Kameyama, A. Ohsumi, Y. Matsuura, and K. Sawada, "Recursive 4SID-based identification algorithm with fixed input-output data size," *International Journal of Innovative Computing Information and Control*, vol. 1, no. 1, pp. 17-33, 2005.

[15] H. Yang and S.Y. Li, "Subspace-based adaptive predictive control for a class of nonlinear systems," *International Journal of Innovative Computing, Information and Control*, vol. 1, no. 4, pp.743-753, 2005.

[16] X. R. Wang, B. Huang, and T. W. Chen, "Data-driven predictive control for solid oxide fuel cells," *Journal of Process Control*, vol. 17, no. 2, pp. 103-114, 2007.

[17] T. Wang, H. J. Gao, and J. B. Qiu, "A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 416-425, 2016.

[18] B. Baykal and A. Constantinides, "Sliding window adaptive fast QR and QR-lattice algorithms," *IEEE Transactions on Signal Processing*, vol. 46, no. 11, pp.2877-2887, 1998.

[19] V. Vajpayee, S. Mukhopadhyay, and A. P. Tiwari, "Data-driven subspace predictive control of a nuclear reactor," *IEEE Transactions on Nuclear Science*, vol. 65, no. 2, pp. 666-679, 2018.

[20] J. S. Zeng, C. H. Gao, and H. Y. Su, "Data-driven predictive control for blast furnace ironmaking process," *Computers and Chemical Engineering*, vol. 34, no. 11, pp. 1854-1862, 2010.

[21] Y. Cao and Z. J. Yang, "Muiltobjective process controllability analysis", *Computers and Chemical Engineering*, vol. 28, no. 1-2, pp. 83-90, 2004.

[22] R. A. Seyab and Y. Cao, "Differential recurrent neural network based predictive control," *Computers and Chemical Engineering*, vol. 32, no.7, pp. 1533-1545, 2008.