Identification of the Tennessee Eastman challenge process with subspace methods

Ben C. Juricek[a], Dale E. Seborg[b,*], Wallace E. Larimore[c]

[a] *Toyon Research Corporation, 75 Aero Camino Road, Goleta, CA 93117, USA*
[b] *Department of Chemical Engineering, University of California, Santa Barbara, CA 93106, USA*
[c] *Adaptics, Inc. 1717 Briar Ridge Road, McLean, VA 22101, USA*

Abstract

The Tennessee Eastman challenge process is a realistic simulation of a chemical process that has been widely used in process control studies. In this case study, several identification methods are examined and used to develop MIMO models that contain seven inputs and ten outputs. ARX and finite impulse response models are identified using reduced-rank regression techniques (PLS and CCR) and state-space models identified with prediction error methods and subspace algorithms. For a variety of reasons, the only successful models are the state-space models produced by two popular subspace algorithms, N4SID and canonical variate analysis (CVA). The CVA model is the most accurate. Important issues for identifying the Tennessee Eastman challenge process and comparisons between the subspace algorithms are also discussed. © 2001 Elsevier Science Ltd. All rights reserved.

1. Introduction

Multivariable linear models are commonly used in process control applications, notably model predictive control (MPC). Most MPC applications use non-parametric linear models such as finite impulse response (FIR) or step response models. For large-dimensional applications, nonparametric models result in a very large number of parameters to be estimated. As shown by Larimore (1996b), the accuracy of the identified model measured by its variance is proportional to the number of estimated parameters. Thus, these "nonparsimonious" models can have poor robustness properties for controller design due to large parametric uncertainty; they also have limited predictive capabilities due to their model structure.

Subspace methods identify linear state-space models from input–output data without using iterative, non-linear optimization or specialized model designs, unlike

maximum likelihood methods. A major tenet of sub-space algorithms is the ability to identify large-dimensional (i.e., large numbers of inputs, outputs and states) models with little design required by the user—a significant challenge for the other techniques mentioned above. Since large dimensional applications are commonly found in many industrial processes, subspace methods are potentially quite useful for process control problems. Additionally, many modern control and monitoring applications are developed for linear state-space models. Hence, identifying parsimonious state-space models from input–output data could facilitate applications of modern control theory to industrial processes.

A previous simulation study demonstrated that subspace models can identify accurate process models for various noise models and for data collected during closed-loop operation (Juricek, Larimore, & Seborg, 1998). In that study, the simulated process was a linear, second-order state-space model with two inputs and two outputs. Due to the linear and low-order model structure, other approaches (e.g., nonlinear optimization) could also produce equivalent, if not better, process models if the model structure was known

*Corresponding author. Tel.: +1 805-893-4731; fax: +1 805-893-3352.

E-mail address: seborg@engineering.ucsb.edu (D.E. Seborg).

reasonably well. In this paper, the process is large-dimensional, nonlinear and does not have a known mathematical representation because the simulation is distributed as an intentionally undocumented FOR-TRAN program (Downs & Vogel, 1993).

Two subspace algorithms, N4SID and canonical variate analysis (CVA), are used to identify state-space models of the Tennessee Eastman challenge process (referred to as TE throughout the paper). FIR and ARX models using two reduced-rank regression methods (partial least squares and canonical correlation regression) and state-space models using optimization are also examined. In the next section, a brief description of subspace models and the relevant issues for the subspace methods are presented. The pertinent issues for identifying the TE are then described and the state-space models from the two subspace algorithms are compared.

2. Subspace methods

Subspace methods are a relatively recent development in the field of system identification. The CVA algorithm was proposed by Larimore (1983), and is based on mathematical statistics and time series analysis methods. The N4SID algorithm, developed by van Overschee and De Moor (1996), is more closely related to linear systems theory. Both algorithms identify a stochastic state-space model,

$$x(k + 1) = Ax(k) + Bu(k) + Ke(k),$$
$$y(k) = Cx(k) + Du(k) + e(k), \tag{1}$$

where x is the $(n_x \times 1)$ vector of state variables, u is the vector of measured inputs $(n_u \times 1)$, y is the vector of measured outputs $(n_y \times 1)$, e is the vector of innovations $(n_y \times 1)$, and the matrix K is the Kalman gain.[1] As described below, the state vector has a very particular meaning for subspace algorithms. The state-space model structure allows for any noise model, i.e., any linear model structure (ARX, ARMAX, OE (Ljung, 1999)) can be represented by a state-space model.

The derivation of subspace algorithms is relatively complicated compared with the traditional prediction error methods. Furthermore, the different approaches used in the derivations make comparing the subspace algorithms with one another difficult. The derivation of CVA is cast in a mathematical statistics framework that has a direct algebraic and linear algebra interpretation (Larimore, 1999). The derivation of N4SID relies on linear algebra and geometric arguments for a linear system written as a set of matrix equations.

For subspace algorithms, the state vector, $x(k)$, is defined by a linear combination of past inputs and outputs,

$$p(k) = [y^T(k-1)\ldots y^T(k-N), u^T(k-1)\ldots u^T(k-N)]^T, \tag{2}$$

$$x(k) = Jp(k), \tag{3}$$

where $p(k)$ is referred to as the "past" at sample k. The dimension of the past is specified by the number of lags, N. The state vector, $x(k)$, is computed from data, and is not specified a priori. Once J has been determined, as will be described below, the state vector can be estimated by (3). The state-space model matrices can then be estimated via linear least squares regression,

$$\begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} = \text{Cov}\left(\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix}, \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \right)$$
$$\times \text{Cov}^{-1}\left(\begin{bmatrix} x(k) \\ u(k) \end{bmatrix}, \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \right). \tag{4}$$

The parameter estimation step of subspace algorithms can vary, but all algorithms proceed in the same general fashion: estimate the state vector from the "past", and then estimate the state-space matrices using "current" values for state, input and output vectors.

The calculation of J distinguishes the various subspace algorithms from one another. In CVA, J is derived from the canonical loadings between the conditional future (i.e., subtracting the effect of the N future inputs from the N future outputs) and the past (Larimore, 1997). In the N4SID approach, J results from a series of geometric arguments based on the set of matrix equations for a linear system (van Overschee & De Moor, 1996). For CVA, N4SID, and all other subspace methods, the key step for calculating J can be written as a weighted singular value decomposition (van Overschee & De Moor, 1994),

$$\text{svd}(W_1 \text{Cov}(\tilde{f}, p) W_2) = [U_1 U_2] \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}, \tag{5}$$

where W_1 and W_2 are weighting matrices, and $\text{Cov}(\tilde{f}, p)$ is the covariance matrix of the conditional future, $\tilde{f}$, and the past, p. J is calculated from the lower dimensional subspace defined by U_1. Each subspace algorithm defines the weighting matrices, W_1 and W_2, differently.

The scaling for CVA is particularly noteworthy because it results from the maximum likelihood (ML) solution for reduced rank regression (Tso, 1981; Larimore, 1997). When CVA is applied to time series data, it is an ML procedure incorporating the correct shift structure of the regression coefficients (i.e., the precise structure of a fully parameterized state-space model of a given order). In CVA this unknown

---

[1] Eq. (1) is written in "innovations" form. The noise models for CVA and N4SID are sometimes written slightly differently, but are completely equivalent.

shift structure is not explicitly imposed, but several simulation studies demonstrate the accuracy of CVA to be essentially equal to the ML solution (Larimore, 1996b). A formal proof for this property has yet to be obtained. ML accuracy permits the computation of confidence bands on model accuracy and hypothesis tests concerning the model structure: e.g., detecting the presence of a bias, feedback, and delays. Near ML accuracy in quite small sample sizes has also been demonstrated in a number of complex systems by Larimore (1996b).

In principle, different subspace algorithms would identify the same state-space model if weighted appropriately. However, other important differences exist between the subspace algorithms: the exact numerical and statistical estimation procedures, determining the dimension of the past (N in (3)), and selecting the model order. Ideally, the model order corresponds to the number of singular values in (5) greater than zero or some very small value, ε. For N4SID, the order is selected ad hoc by the user, usually looking for a "knee" in the plot of singular values, or by noting where the singular values fall below a specified critical value. For CVA, the singular values (or equivalently the canonical correlations) can also be used for order selection based on objective tests of hypotheses rather than ad hoc user selection. For the ADAPT$_X$ software (Larimore, 1996a), the model order is automatically selected via Akaike's information criteria (AIC), an optimal procedure for model order selection (Larimore, 1999). As the TE results will show, simply using the CVA weighting and the N4SID algorithm did not produce the same model as was identified by ADAPT$_X$. Thus, other steps in the

subspace methodology apparently are necessary to achieve the near ML properties of CVA in finite sample sizes.

3. The Tennessee Eastman challenge process

The TE challenge process was published by the Tennessee Eastman Company (Downs & Vogel, 1993) as a process simulation for academic research. By academic standards, the problem is quite large: it contains 41 measured variables and 12 manipulated variables. Based on a real chemical process, the TE produces two products (labeled G and H) from four reactants (labeled A, C, D and E). An explicit mathematical representation of the process is not given; instead the simulation is distributed as purposely convoluted FORTRAN code (Ricker, 2001). In addition to the process description, the problem statement defines process constraints, 20 types of process disturbances, and six operating modes corresponding to different production rates and G/H mass ratios in the product stream. The base operating mode is a 50/50 G/H mass ratio and a production rate of 14,072 lb/h.

A simplified diagram of the process is shown in Fig. 1. The process consists of three units: a reactor, product separator, and stripper. A recycle stream returns unused reactants in the product to the reaction section. The 41 measurements are a mixture of continuous and sampled measurements. Each measurement is corrupted by additive noise; the statistical properties of the noise are unknown. The process is nonlinear, open-loop unstable, and contains a mixture of fast and slow dynamics.
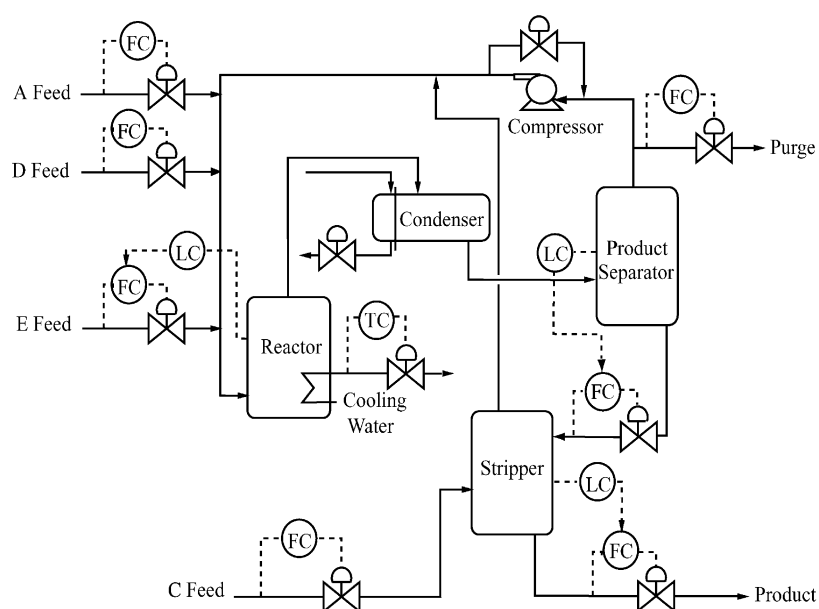

Fig. 1. A simplified diagram of the Tennessee Eastman challenge process.

Several researchers have used the TE for a variety of applications. McAvoy and Ye (1994), Ricker (1996) and Sriniwas and Arkun (1997) have designed different control systems for the process. Farina, Trierweiler, and Secchi (2000) analyzed the control properties (i.e., performance and robustness) of these control systems using the "Robust Performance Number". Ricker and Lee (1995a, b) derived a physical model of the TE and used this model to develop a nonlinear MPC algorithm. Using the economic data for the TE, Ricker (1995) optimized the steady-state conditions for the six operating modes. No detailed identification studies for the TE have been published, although Sriniwas and Arkun (1997) identified multi-input single output (MISO) models for the production rate and key compositions that were used in an MPC algorithm. Also, Chiang, Russel, and Braatz (2001) identified a state-space model using CVA in order to demonstrate process monitoring strategies. These studies emphasize the resulting applications more than the properties of the identified models.

3.1. Methodology for the TE case study

In order to generate input–output data for identifying a process model, the TE must first be stabilized. The control strategy must prevent process constraints from being violated while the TE is perturbed by the input excitation. Such a control system could serve as the base control system for an MPC controller acting at the supervisory level. The base control system for this study consisted of the "Stage One" and "Stage Two" controllers from the decentralized control strategy proposed by McAvoy and Ye (1994). Stage One provides PI controller settings for the flow controllers in Fig. 1: the A, C, D, and E feed flowrates, and the purge and product flowrates. The Stage Two loops for the reactor, product separator and stripper levels, and the reactor cooling water outlet temperature were required to prevent shutdown limits from being violated. During the excitation phase, the levels and stripper steam rate were controlled at the optimal steady-state values specified by Ricker (1995).

In order to define a process model that was large, but manageable, the seven inputs and ten outputs in Tables 1 and 2 were selected. The composition measurements in the TE were omitted to avoid a multi-rate sampling problem, which subspace methods do not readily handle. The sampling period was $\Delta t = 1$ min and 96 h (5760 samples) of data were used for the system identification.

Process nonlinearities are important characteristics for the TE if all of the possible operating modes are considered. However, for small changes (e.g., approximately 10%) from the nominal values in Table 1, a linear model describes the process reasonably well, as

Table 1
Model inputs and nominal values

u_1	Compressor recycle valve	22.2%
u_2	Condenser cooling water flow	18.1%
u_3	A feed SP[a]	0.25 kscmh
u_4	D feed SP[a]	3686 kg/h
u_5	C feed SP[a]	9.35 kscmh
u_6	Purge rate SP[a]	0.34 kscmh
u_7	Reactor CW temp. SP[a]	94.6°C

[a] SP: setpoint to flow controller.

Table 2
Model outputs and nominal values

y_1	Recycle flow	26.64 kscmh
y_2	Reactor feed rate	42.00 kscmh
y_3	Reactor pressure	2687 kPa
y_4	Reactor temp.	120.3°C
y_5	Product separator temp.	80.50°C
y_6	Product separator pressure	2616 kPa
y_7	Stripper pressure	3082 kPa
y_8	Stripper temp.	56.97°C
y_9	Compressor work	338.0 kW
y_{10}	Separator CW temp.	76.79°C

indicated by responses to the following sequence of step changes in the ith input:

$$u_i(t) = \begin{cases} u_i^0, & t < t_0, \\ u_i^0 + \delta u_i, & t_0 \leqslant t < t_0 + 24 \text{ h}, \\ u_i^0 - \delta u_i, & t_0 + 24 \text{ h} \leqslant t < t_0 + 48 \text{ h}, \\ u_i^0, & t \geqslant t_0 + 48 \text{ h}, \end{cases} \quad (6)$$

where u_i^0 is the nominal input for the base operating mode, given in Table 1. In Eq. (6), t_0 is the time when the first step change is introduced. Fig. 2 shows the response to the step sequence for the D feed flowrate, with $\delta u_4 = 300$ kg/h. The responses are nearly linear because the gains and time constants are approximately equal for the step in either direction. Fig. 3 shows the response to steps of $\delta u_3 = 0.05$ kscmh in the A feed flowrate. For this case, the static and dynamic nonlinear effects are more apparent; for example, the reactor temperature settles faster for the increase in u_3, than for the decrease. Linear process models around the base operating condition are identified in this case study.

The step responses in Figs. 2 and 3 illustrate the wide range in time constants for the TE. This range is especially difficult to model with nonparametric models such as FIR or step response models. If Δt is the same for each variable, a small Δt will result in an enormous ($O(10^3)$) number of FIR coefficients in order for all ten output variables to settle. A large Δt will neglect faster dynamics that are faster than
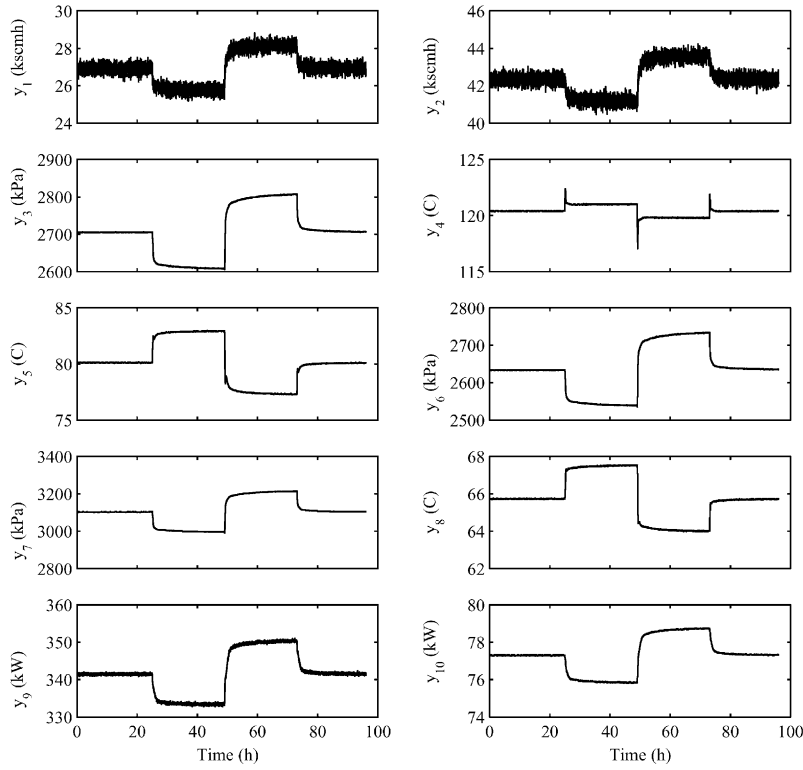
Fig. 2. The open-loop response of the outputs to a step in u_4, the D Feed Rate, indicates the fast mode ($\delta u_4 = 300$ kg/h).
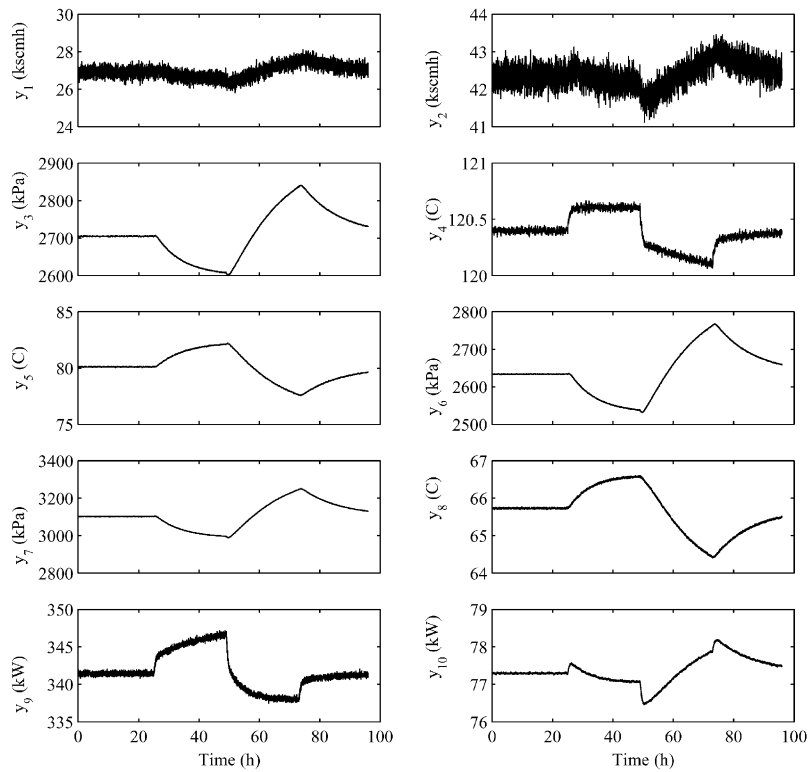

Fig. 3. The open-loop response of the outputs to a step in u_3, the A Feed Rate, indicates the slow mode ($\delta u_3 = 0.05$ km scm h).

the Nyquist frequency, $\omega_N = \pi/\Delta t$. Consequently, subspace methods are advantageous because the state-space model structure can handle the wide range of time constants better.

Identification data were generated by perturbing each manipulated variable as a three-level sequence. Three-level sequences, like pseudo random binary sequences (PRBS), are deterministic signals generated by a difference equation,

$$u_i(t) = \text{rem}(a_1 u_i(t-1) + \cdots + a_{ns} u_i(t-ns), 3), \quad (7)$$

where *ns* is the *order* of the sequence, a_i are the generating polynomial coefficients, and $\text{rem}(x, 3)$ indicates the remainder of x divided by 3. The sequence u_i only assumes three levels: $0, 1$, and 2. After the sequence was generated, these levels were translated to $u_i^0 - \delta u_i$, u_i^0, and $u_i^0 + \delta u_i$, where u_i^0 is the nominal value for the *i*th input in Table 1. Godfrey (1993) discusses the properties of three-level sequences, and provides the generating polynomial coefficients, a_i, in (7). Theoretically, three-level sequences reduce the effect of nonlinearities on the resulting linear model (Godfrey, 1993).

The seven manipulated variables were excited simultaneously and independently with seventh-order three-level sequences and a switching time of 30 min (i.e., $ns = 7$ and Eq. (7) was evaluated every 30 min). The input amplitudes were chosen to have roughly similar effects on the reactor pressure, because the pressure constraint limited the degree of excitation. The amplitudes are shown in Table 3. Fifty hours of excitation data for the inputs and outputs are shown in Figs. 4 and 5, respectively. Ninety-six hours of data (5760 samples) were used for system identification.

An attempt was made to identify N4SID models using the MATLAB System Identification Toolbox (Ljung, 1995). However, the large problem dimensions caused computational problems (out of memory errors) that

Table 3
Amplitudes for three-level sequences

δu_1	Compressor recycle valve	5%
δu_2	Condenser cooling water flow	1%
δu_3	A feed SP[a]	0.025 kscmh
δu_4	D feed SP[a]	300 kg/h
δu_5	C feed SP[a]	0.25 kscmh
δu_6	Purge rate SP[a]	0.05 kscmh
δu_7	Reactor CW temp. SP[a]	1.0°C

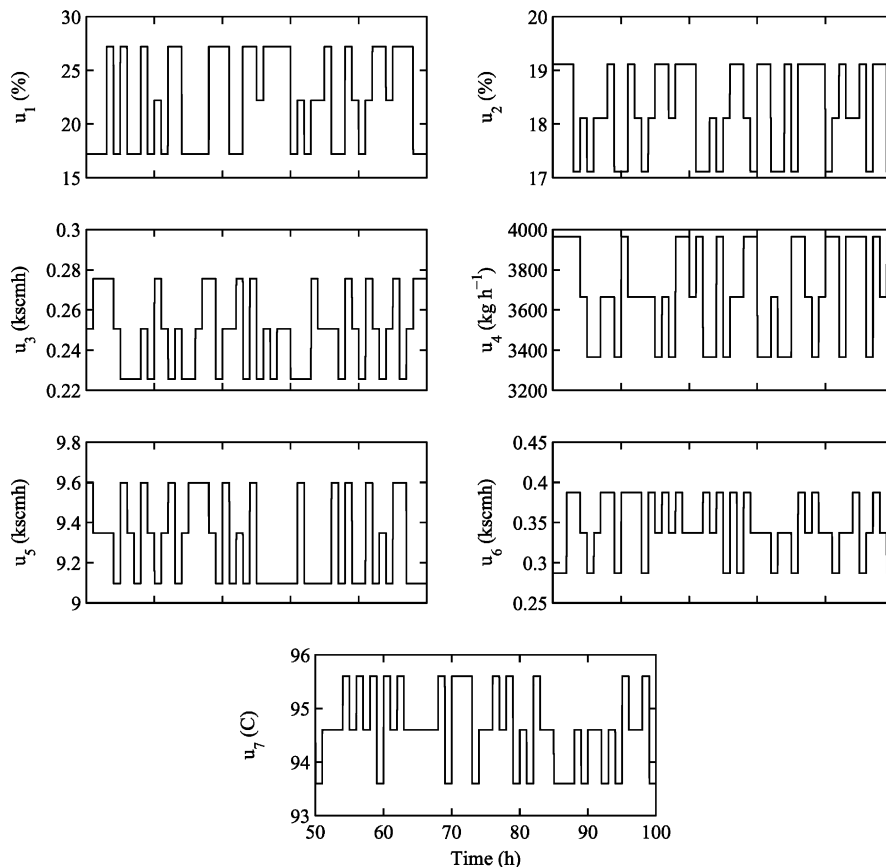[a] SP: setpoint to flow controller.

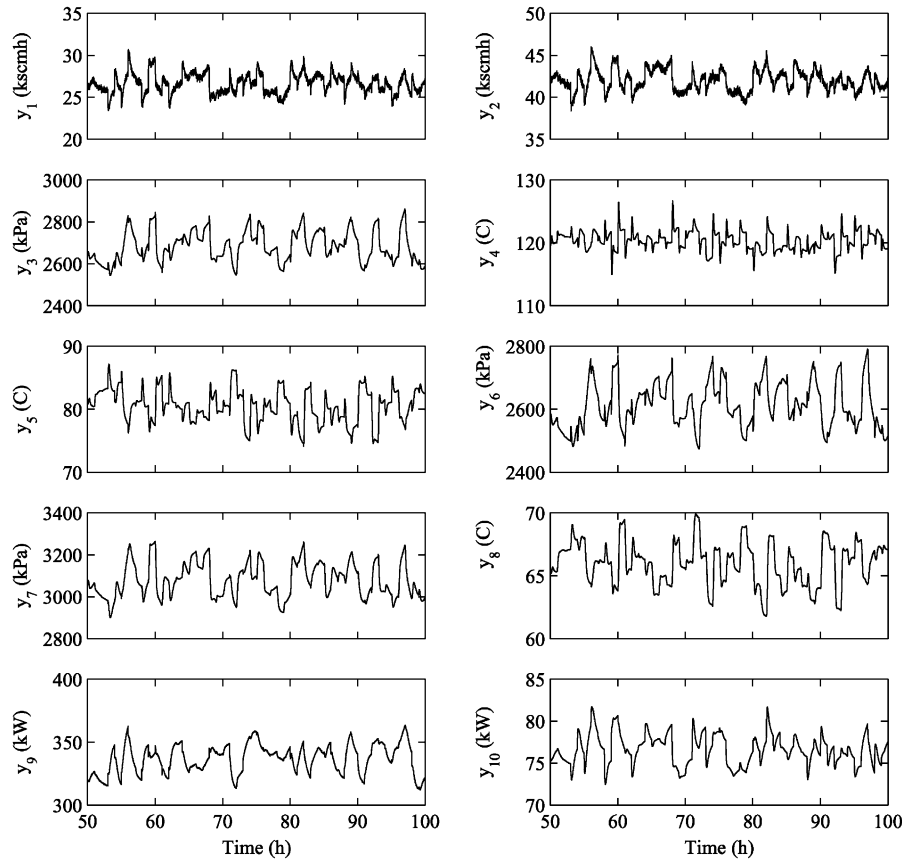Fig. 4. Excitation data for the inputs.

Fig. 5. Excitation data for the outputs.

limited the model order. Then, N4SID models were identified using the "Robust N4SID" code, that is available in the book by van Overschee and De Moor (1996). Modifications to the code were required in order to guarantee a stable transition matrix, and to guarantee a strictly proper system (i.e., no direct input–output terms). The transition matrix was stabilized with the method proposed by Maciejowski (1995). The CVA models were identified using ADAPT$_X$ version 3.5 (Larimore, 1996a).

The nominal values in Tables 1 and 2 were subtracted from the excitation data prior to identification. Since the CVA algorithm is not scale sensitive, no further data preprocessing was necessary to identify the CVA models. For the N4SID models, input and output variables were scaled by their standard deviations.

Numerical problems also occurred for MIMO ARX models estimated using canonical correlation regression (Tso, 1981) or partial least squares (PLS) methods and the MATLAB PLS Toolbox (Wise & Gallagher, 1996). However, MISO ARX models were successfully identified. The ARX model for the ith output with order m_i is given by

$$A(q^{-1})y_i(k) = B_1(q^{-1})u_1(k) + \cdots + B_7(q^{-1})u_7(k), \qquad (8)$$

$$A(q^{-1}) = 1 - A_1 q^{-1} - \cdots - A_{m_i} q^{-m_i}, \qquad (9)$$

$$B_j(q^{-1}) = B_{j,1} q^{-1} + \cdots + B_{j,m_i} q^{-m_i}. \qquad (10)$$

The ARX model parameters were estimated using ordinary least squares. The use of MISO ARX models is a more traditional identification approach than MIMO ARX or subspace methods for process applications (Rivera & Jun, 2000).

4. Results from the TE case study

For the CVA models, the minimum AIC occurred when the subspace model order was 23, as shown in Fig. 6. Based on experimentation with several other model orders, this choice appears to be reasonable. The plot of singular values in Fig. 7(a) shows a "knee" at order 23, supporting this choice. Based on the singular values for the N4SID algorithm, shown in Fig. 7(b), and experimentation with several model orders, a 15th order model was identified. The MISO ARX model orders, m_i for $i = 1 \ldots 10$, were selected according to the minimum AIC and are shown in Table 4. The CVA model had 667
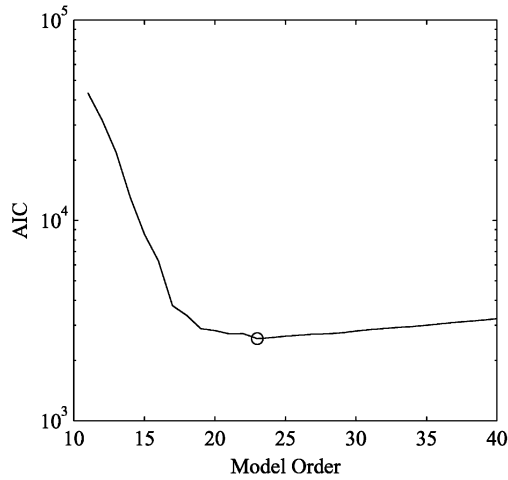
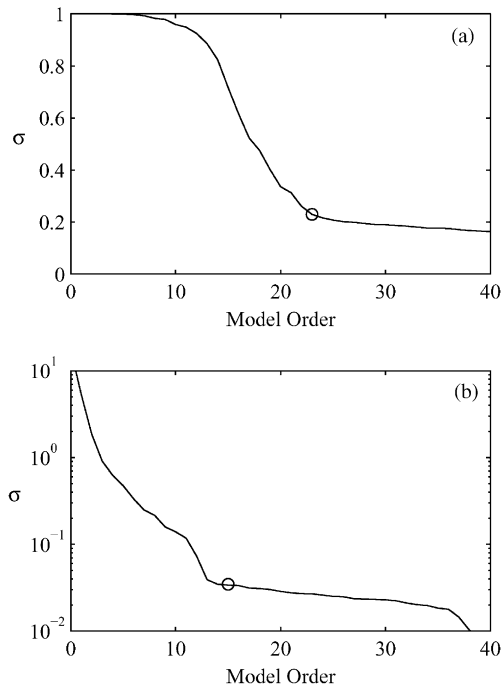Fig. 6. AIC versus model order. The circle indicates the selected model order.

Fig. 7. The singular values, σ, versus model order for (a) CVA and (b) N4SID. The circles indicate the selected model orders.

Table 4
MISO ARX model orders

Model	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}
Order (m_i)	4	4	3	5	7	3	3	4	6	6

are shown in Figs. 8 and 9 for inputs and outputs, respectively. For validation set 2, data were generated by perturbing each input individually using the sequence of steps in (6); these data are shown in Figs. 2 and 3 for u_4 and u_3, respectively. For this validation data set, each input was excited individually in order to assess the model accuracy for particular input–output pairs. Three prediction horizons were selected in order to approximately assess the model accuracy in high ($n = 1$), intermediate ($n = 15$) and low ($n = \infty$) frequencies. A classical measure of model accuracy is given by the coefficient of determination, R^2,

$$R^2(n) = \left[1 - \frac{\sum_{i=1}^{N} (y(i) - \hat{y}(i|i - n))^2}{\sum_{i=1}^{N} (y(i) - \bar{y})^2} \right] \times 100\%, \qquad (11)$$

where N is the number of data points and n is the prediction horizon. For validation set 1, the R^2 value will indicate whether a particular output variable is poorly modeled. For the step responses in validation set 2, a poor model for a particular input–output pair is indicated by a small R^2 value. Also, the R^2 value for validation set 2 weights the steady-state behavior (or, more precisely, the low frequencies) more heavily than the dynamic behavior because there are more data at steady state. Hence, a small R^2 value may be due to a steady-state error (i.e., an offset).

The R^2 results for the identification data and validation set 1 are given in Tables 5 and 6, respectively. In general, the R^2 values for the identification data are slightly better for the identification data than for validation set 1, as expected. Since the R^2 values for these validation data are comparable to the values for the identification data, the models are clearly modeling the process and not overfitting the data (i.e., "fitting the noise").

For validation set 1 and $n = 1$, most of the R^2 values are well above 95%, indicating accurate predictions. The CVA and ARX models are very accurate for most outputs with R^2 values greater than 99%. For $n = 15$ and $n = \infty$, the R^2 values decrease. However, the CVA model decreases are relatively small compared to the changes for the ARX and N4SID models; in particular, the N4SID model decreases the most. Fig. 10 shows the model residuals for y_8 and validation set 1. The residuals confirm the results of the R^2 values: the ARX and N4SID predict y_8 very poorly for $n = 15$ and ∞.

parameters, the N4SID had 460 parameters, and the ARX model had 360.

The identified models were compared by examining the n-step ahead model predictions for two sets of validation data and different values of n. For validation set 1, the data were generated using the same conditions that generated the identification data: inputs were excited simultaneously as three-level sequences using the amplitudes in Table 3, and each sequence had a switching time of 30 min. Validation set 1 data
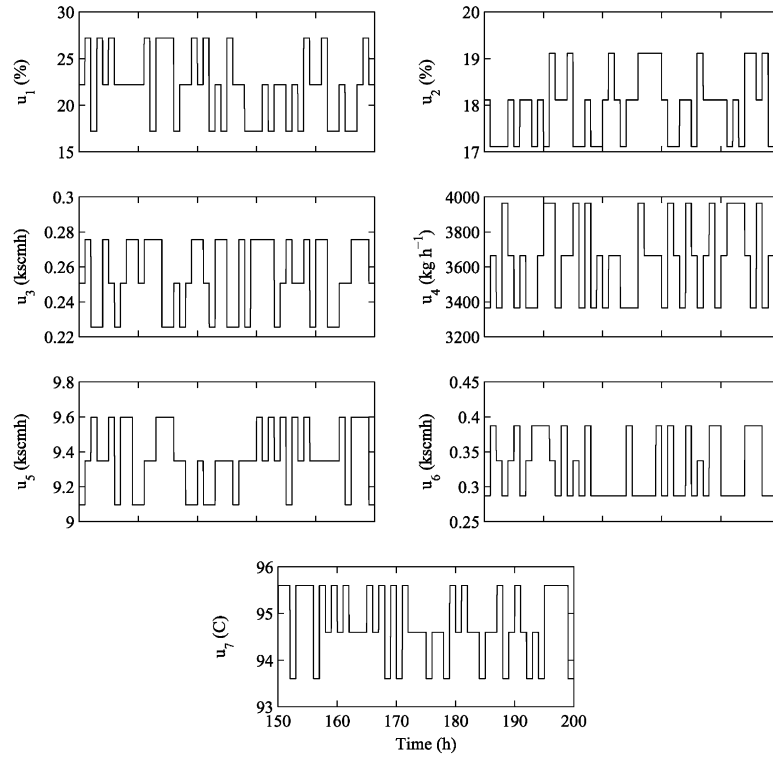
Fig. 8. Validation set 1 input data.

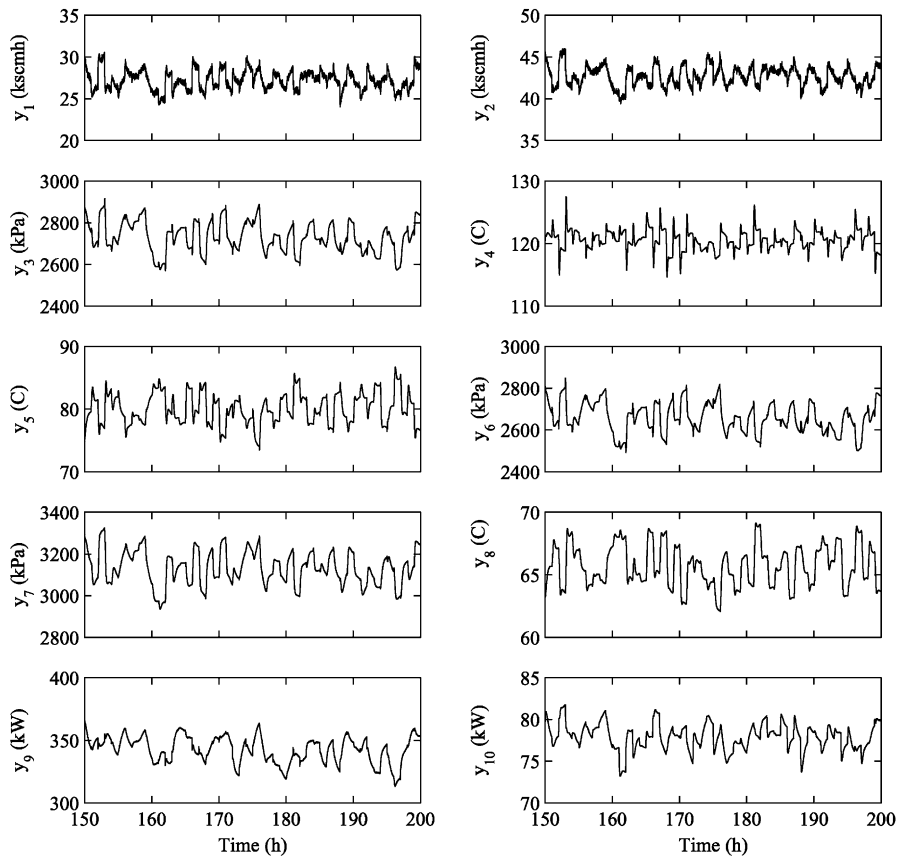Fig. 9. Validation set 1 output data.

Table 5
R^2 values (%) for identified models using identification data (R^2 values less than zero are indicated by a (—) entry.)

	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	
$n=1$	97.1	97.4	99.9	99.8	99.9	99.9	99.9	99.9	99.9	99.9	CVA
	87.7	91.8	99.7	98.9	99.8	99.7	99.5	97.4	99.4	99.9	N4SID
	96.3	96.8	99.9	99.8	99.9	99.9	99.9	99.9	99.9	99.9	ARX
$n=15$	96.5	97.3	99.8	99.4	99.5	99.7	99.8	99.7	99.5	99.3	CVA
	77.2	80.6	82.3	94.5	92.4	82.8	82.6	2.9	84.2	86.7	N4SID
	92.1	93.5	92.2	68.8	97.7	92.2	89.8	81.6	98.6	98.8	ARX
$n=\infty$	95.4	96.8	97.4	99.4	99.1	97.3	97.5	99.0	97.5	99.1	CVA
	76.7	75.4	78.1	92.0	90.8	78.4	78.0	(—)	75.2	79.3	N4SID
	90.5	91.4	74.5	61.0	95.1	74.8	66.6	(—)	95.5	97.5	ARX

Table 6
R^2 values (%) for identified models using validation set 1 (R^2 values less than zero are indicated by a (—) entry.)

|	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	
$n=1$	96.9	97.0	99.9	99.7	99.9	99.9	99.9	99.9	99.9	99.9	CVA
	89.1	90.8	99.7	98.8	99.7	99.7	99.5	96.8	99.4	99.9	N4SID
	96.2	96.3	99.9	99.8	99.9	99.9	99.9	99.9	99.9	99.9	ARX
$n=15$	96.5	96.9	99.7	99.1	99.0	99.7	99.7	99.5	99.4	98.9	CVA
	80.3	80.1	81.0	94.3	91.5	81.1	81.0	(—)	84.9	84.4	N4SID
	91.5	92.8	92.6	67.6	96.9	92.5	90.7	81.8	98.2	98.2	ARX
$n=\infty$	96.2	96.0	91.4	99.2	97.6	91.2	91.7	98.0	98.5	97.2	CVA
	80.0	74.6	76.5	92.8	89.6	76.5	76.2	(—)	76.2	75.2	N4SID
	90.7	89.9	74.2	61.2	93.4	74.3	68.7	(—)	95.1	95.1	ARX
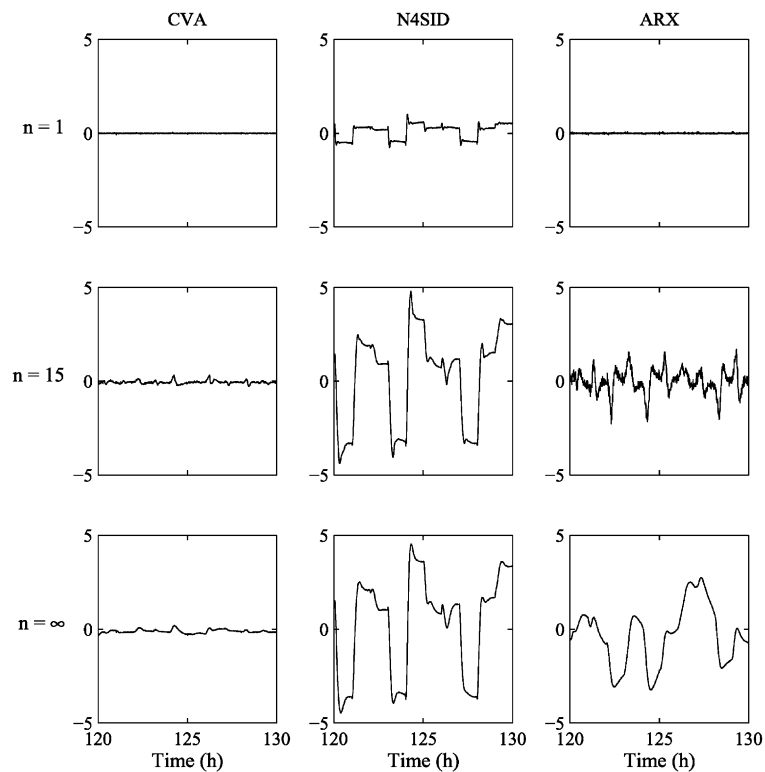
Fig. 10. The y_8 model residuals for validation set 1 and three prediction horizons: $n = 1, 15, \infty$.

Table 7
R^2 values of identified models for prediction horizon $n = 1$ (R^2 values less than zero are indicated by a (—) entry.)

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	
y_1	81.2	95.3	67.9	93.8	92.4	83.4	98.4	CVA
	23.2	90.1	0.3	89.7	57.2	41.4	93.1	N4SID
	77.8	94.9	59.7	92.3	90.4	78.9	98.0	ARX
y_2	86.6	94.5	60.4	93.5	96.3	83.1	98.2	CVA
	52.3	87.9	18.4	90.0	83.6	67.5	91.6	N4SID
	84.2	93.2	51.2	91.9	95.4	78.9	97.8	ARX
y_3	99.9	99.9	99.9	99.9	99.9	99.9	99.9	CVA
	99.4	99.8	99.4	99.9	99.4	99.0	99.9	N4SID
	99.9	99.9	99.9	99.9	99.9	99.9	99.9	ARX
y_4	98.6	87.4	97.9	99.8	99.8	58.2	99.9	CVA
	82.2	64.3	48.4	99.6	94.6	(—)	99.9	N4SID
	98.8	98.0	98.2	99.8	99.9	92.6	99.9	ARX
y_5	99.9	99.9	99.9	99.9	99.9	99.9	99.9	CVA
	99.7	99.7	99.0	99.9	98.6	98.3	99.9	N4SID
	99.9	99.9	99.9	99.9	99.9	99.9	99.9	ARX
y_6	99.9	99.9	99.9	99.9	99.9	99.9	99.9	CVA
	99.7	99.8	99.6	99.9	99.5	99.0	99.9	N4SID
	99.9	99.9	99.9	99.9	99.9	99.9	99.9	ARX
y_7	99.9	99.9	99.9	99.9	99.9	99.9	99.9	CVA
	95.3	99.5	99.1	99.8	98.9	98.6	99.5	N4SID
	99.9	99.9	99.9	99.9	99.9	99.9	99.9	ARX
y_8	99.9	99.9	99.9	99.9	99.9	99.9	99.9	CVA
	87.5	98.3	83.2	99.9	83.4	65.5	99.7	N4SID
	99.7	99.9	99.9	99.9	99.9	99.8	99.9	ARX
y_9	99.9	99.9	99.1	99.8	99.9	99.3	99.9	CVA
	99.9	99.1	98.8	99.7	99.8	98.9	99.8	N4SID
	99.9	99.9	98.9	99.8	99.9	99.2	99.9	ARX
y_{10}	99.9	99.9	99.8	99.9	99.9	99.9	99.9	CVA
	99.4	99.9	99.0	99.9	99.8	99.5	99.9	N4SID
	99.9	99.9	99.8	99.9	99.9	99.9	99.9	ARX

Table 8
R^2 values of identified models for prediction horizon $n = 15$ (R^2 values less than zero are indicated by a (—) entry.)

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	
y_1	79.2	92.8	67.6	93.7	92.2	81.8	98.2	CVA
	(—)	65.6	3.5	85.0	42.6	9.5	85.7	N4SID
	76.6	93.3	50.8	93.0	88.8	69.6	97.5	ARX
y_2	85.3	94.2	59.8	93.5	96.3	82.9	98.2	CVA
	3.6	58.7	19.2	83.2	67.3	32.1	78.9	N4SID
	82.9	92.1	47.7	92.3	95.3	74.7	97.5	ARX
y_3	99.9	99.2	99.8	99.9	99.9	99.3	99.8	CVA
	60.0	55.3	23.2	96.2	58.3	20.9	87.1	N4SID
	92.5	93.4	89.0	99.6	92.5	85.5	98.5	ARX
y_4	96.0	(—)	88.1	99.4	98.0	(—)	99.8	CVA
	81.0	(—)	(—)	98.6	88.8	(—)	99.0	N4SID
	(—)	(—)	(—)	99.1	86.1	(—)	91.2	ARX
y_5	99.5	96.2	99.6	99.9	99.5	95.7	99.9	CVA
	88.8	69.9	8.8	96.7	51.2	28.3	98.1	N4SID
	98.0	98.2	82.3	99.3	95.0	86.6	99.6	ARX
y_6	99.9	99.3	99.8	99.9	99.9	99.3	99.8	CVA
	68.1	57.5	23.9	96.4	58.2	19.8	86.8	N4SID
	93.7	93.0	88.6	99.6	92.1	85.0	98.3	ARX
y_7	99.8	99.2	99.8	99.9	99.9	99.3	99.9	CVA
	(—)	55.7	24.1	95.6	56.0	18.5	85.0	N4SID
	66.6	92.2	88.4	99.6	91.1	84.2	98.1	ARX
y_8	98.8	99.7	99.5	99.9	99.6	98.3	99.9	CVA
	(—)	(—)	(—)	99.4	(—)	(—)	95.9	N4SID
	67.3	97.6	94.7	97.4	94.3	91.2	98.9	ARX
y_9	99.7	99.2	96.7	99.7	99.9	98.3	99.9	CVA
	93.8	74.9	2.6	89.9	89.8	51.0	88.5	N4SID
	99.4	99.4	97.7	99.5	99.7	96.2	99.6	ARX
y_{10}	98.9	99.5	99.3	99.8	99.9	98.5	99.4	CVA
	(—)	94.9	49.9	88.3	78.8	42.6	71.3	N4SID
	97.5	99.9	91.8	99.8	99.7	96.6	98.9	ARX

Tables 7–9 show the R^2 values for validation set 2 and $n = 1$, 15, and ∞. For $n = 1$, the R^2 values for each model are very good (greater than 95%) for many of the input–output pairs. The R^2 values for N4SID and ARX decrease significantly when the prediction horizon increases to $n = 15$ for several pairs, while the CVA model has approximately the same accuracy as for $n = 1$; for example, the $y_3 - u_6$ (the reactor temperature — purge rate) and $y_9 - u_3$ (compressor work — A feed) pairs demonstrate this behavior. This type of decrease occurred for several pairs for the N4SID and ARX models. The R^2 values continue to decrease for larger prediction horizons. For $n = \infty$, the predicted outputs are inaccurate for the inputs with very large time constants, indicating that the very slow dynamics have not been modeled accurately. The input variables with

these slow dynamics are u_3 (A feed), u_5 (C feed), and u_6 (purge rate). Based on the R^2 values for these inputs in Table 9, the CVA model clearly is the most accurate model for the slow dynamics; the N4SID and ARX models are of approximately the same accuracy for u_3, u_5 and u_6.

Fig. 11 shows $y_3 - u_6$ predictions for different prediction horizons. These plots demonstrate the same general trends as for the R^2 metrics. For $n = \infty$, the ARX and N4SID predictions are especially poor. The slow dynamics and nonlinear gain are difficult to model well. The slow dynamics could be modeled better by increasing the sampling period, which would improve the resolution of the poles near the unit circle (i.e., the slow dynamics) at the expense of poorer resolution of poles near the origin (i.e., the fast dynamics). In Figs. 12 and 13, the predictions of each model for $n = 15$ are shown for four representative input–output pairs. These pairs were selected to show results for different variable types, and for slow and fast dynamics. The N4SID model predictions clearly indicate that the gain (e.g., for $y_2 - u_2$) and slow dynamics (e.g., for $y_9 - u_3$) are poorly modeled. The predictions of the

Table 9
R^2 values of identified models for prediction horizon $n = \infty$ (R^2 values less than zero are indicated by a (—) entry.)

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	
y_1	66.0	89.6	18.3	93.3	86.3	52.7	94.9	CVA
	(—)	63.0	5.5	84.9	41.5	2.5	84.7	N4SID
	66.8	85.3	(—)	93.2	76.7	15.4	94.7	ARX
y_2	76.8	87.6	13.6	93.1	93.6	51.5	95.0	CVA
	(—)	41.6	0.2	79.9	56.8	(—)	72.5	N4SID
	73.0	81.7	(—)	91.9	91.4	22.2	93.7	ARX
y_3	84.2	87.3	32.1	99.8	88.2	50.2	89.8	CVA
	50.7	41.7	11.1	95.0	48.0	(—)	83.6	N4SID
	48.7	53.4	16.5	99.1	43.4	(—)	89.9	ARX
y_4	92.3	94.6	87.2	99.7	99.3	(—)	99.9	CVA
	(—)	(—)	23.1	95.7	90.5	34.2	98.5	N4SID
	(—)	(—)	(—)	99.3	72.6	(—)	86.5	ARX
y_5	95.2	98.9	34.9	99.9	86.5	53.3	99.3	CVA
	84.3	73.9	9.0	95.7	28.6	(—)	97.4	N4SID
	91.7	94.1	(—)	96.4	71.5	23.7	98.1	ARX
y_6	87.3	87.3	32.2	99.8	88.1	50.2	89.8	CVA
	60.2	43.9	11.7	95.2	47.3	(—)	83.2	N4SID
	58.0	52.4	16.7	99.1	42.9	(—)	89.0	ARX
y_7	42.4	87.1	31.9	99.8	88.6	50.1	89.9	CVA
	(—)	40.7	12.6	94.3	44.6	(—)	80.7	N4SID
	(—)	46.8	17.3	98.9	36.8	(—)	88.1	ARX
y_8	84.7	99.8	(—)	95.8	62.7	33.0	98.7	CVA
	(—)	(—)	(—)	99.3	(—)	(—)	95.2	N4SID
	(—)	(—)	(—)	(—)	(—)	58.0	(—)	ARX
y_9	95.2	94.5	76.2	98.8	99.5	71.8	98.2	CVA
	89.6	54.4	6.0	80.9	76.0	(—)	78.4	N4SID
	92.9	94.0	88.3	98.1	98.4	59.0	96.2	ARX
y_{10}	87.7	98.0	18.0	99.5	97.4	61.5	98.4	CVA
	(—)	91.7	(—)	80.3	61.4	(—)	50.9	N4SID
	73.8	97.9	(—)	99.5	96.5	43.4	83.6	ARX

ARX and CVA models indicate that these models are more accurate.

A plot of the poles for the CVA and N4SID models is shown in Fig. 14. The CVA model places several poles closer to the unit circle; these poles are related to the slower system dynamics. Based on the step responses shown in Fig. 3, the TE exhibits slow dynamics. The absence of poles near the unit circle for the N4SID models explains the poor predictions for variables with slow dynamics, even for $n = 1$ and 15.

As noted earlier, the transition matrix of the model identified with the "Robust N4SID" code was stabilized by modifying the method for estimating of A in (1), as suggested by van Overschee and De Moor (1994). When the algorithm was not modified, the identified transition matrix had poles outside the unit circle, and the model for multiple step ahead ($n = 15$ and ∞) predictions was unstable. As described by Maciejowski (1995), this modification can bias the estimates of the poles, which explains why the slow modes were not modeled well by the N4SID model. It may be possible to decrease this bias in the estimated transition matrix by modifying the algorithm differently, with methods such as those proposed by Chui and Maciejowski (1996).

The effect of weighting in the N4SID algorithm was examined by identifying a model with the N4SID algorithm and CVA weighting (referred to as "weighted
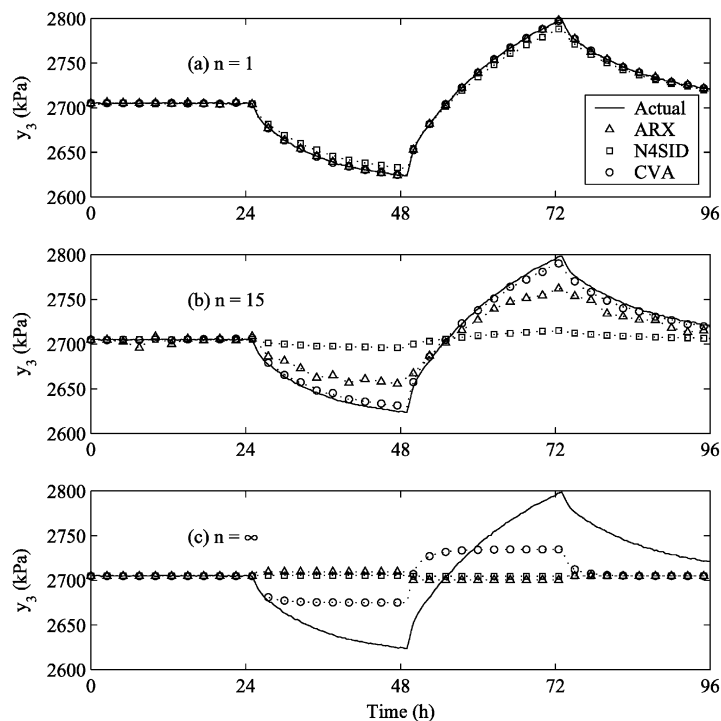

Fig. 11. Comparisons of the actual and predicted reactor pressure response to a sequence of steps in the purge rate and three prediction horizons: $n = 1, 15, \infty$.
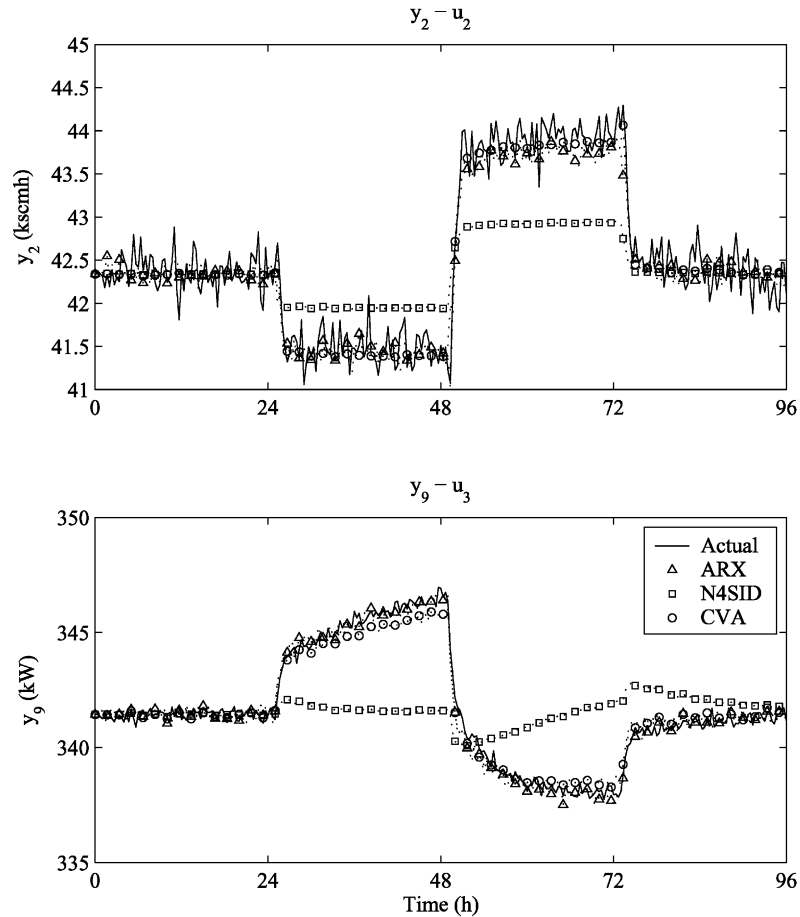
Fig. 12. Comparisons of the actual and predicted responses to a sequence of steps for $n = 15$ for $y_2 - u_2$ (reactor feed—condensor CW) and $y_9 - u_3$ (compressor work—A feed).

N4SID"). Interestingly, the results using the weighted N4SID model were *not* the same as the CVA models obtained using the ADAPT$_X$ code. The singular values (i.e., canonical correlation coefficients) and identified models of the same order were different. For all of the simulations, the dimension of the past in (2) was the same, $N = 10$, as selected according to the AIC statistic calculation in the ADAPT$_X$ software. The difference between the two subspace algorithms may be due to specific numerical steps (e.g., whether a QR or SVD algorithm was used for a particular step) and the sample statistical properties, which are unknown.

In summary, based on the R^2 values, the CVA state-space model was the most accurate, followed by the set of MISO ARX models, and the N4SID state-space model. The MISO ARX models performed reasonably well for short prediction horizons, and had larger R^2 values than the N4SID models for several input–output pairs. However, the set of identified ARX models is more difficult to interpret than a single state-space model. Thus, when compared with the MISO

ARX model, the N4SID state-space model might still be preferred because it is a more parsimonious representation.

5. Conclusions

The TE challenge process was used to compare dynamic models identified using the CVA, N4SID and ARX methods. Although the TE is a nonlinear system, linear models for the base operating mode were reasonably accurate for most of the seven inputs and ten outputs that were included in the model. The models identified by the CVA algorithm were particularly accurate, and should be well-suited for model-based control and monitoring applications. In general, the set of MISO ARX models was less accurate than the CVA model, but better than the N4SID state-space model, as indicated by the R^2 values for two validation data sets. Compared with a set of MISO ARX models, the single state-space model identified by CVA was more accurate and simpler to interpret. For example, the statistical
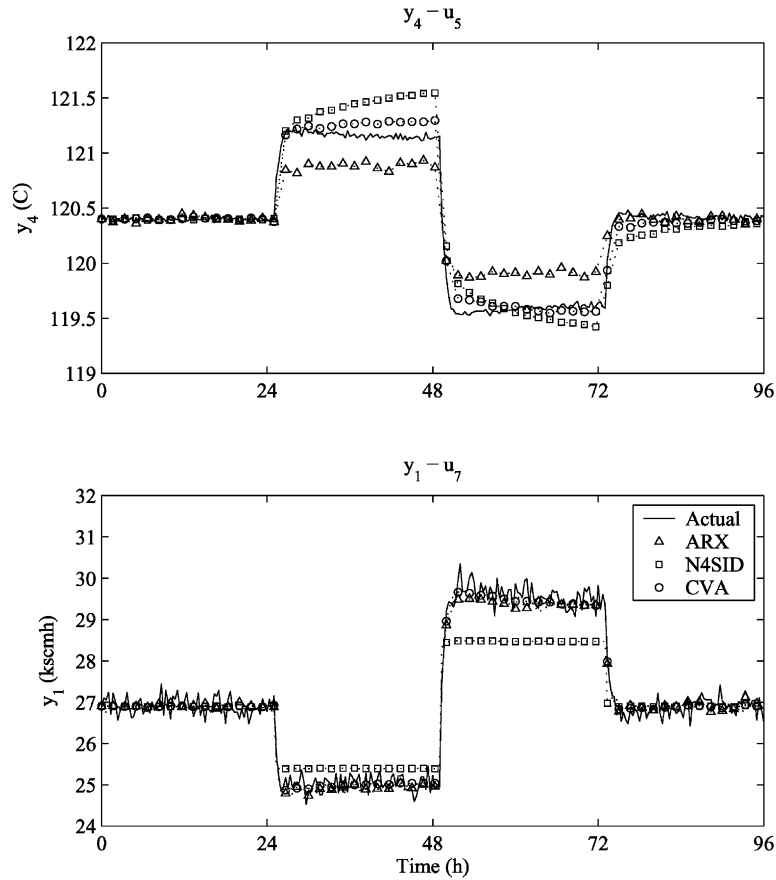
Fig. 13. Comparisons of the actual and predicted responses to a sequence of steps for $n = 15$ for $y_4 - u_5$ (reactor temperature—C feed) and $y_1 - u_7$ (recycle flow—reactor CW temperature).
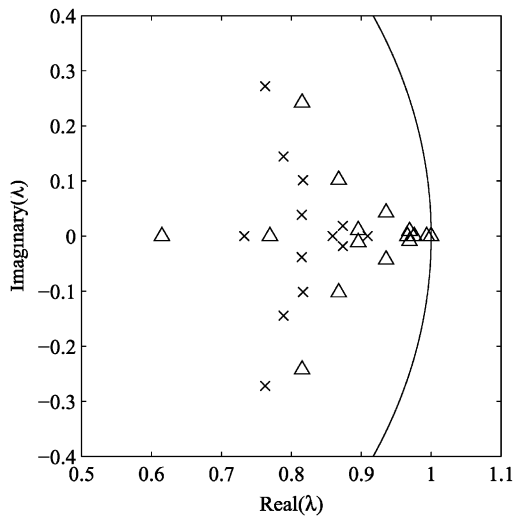

Fig. 14. The CVA poles (denoted by $\triangle$) are closer to the unit circle than the N4SID poles (denoted by $\times$).

properties of the noise and the modes of the system are easier to analyze using the single state-space model than seven ARX models. Although the subspace algorithms are related by the generalized singular value decomposi-tion, using the N4SID algorithm with the CVA weighting did not produce the same state-space model as the CVA model that was identified using the ADAPT$_X$ software.

Acknowledgements

References

Chiang, L. H., Russel, E. L., & Braatz, R. D. (2001). *Data-driven techniques for fault detection and diagnosis in chemical processes.* London: Springer.

Chui, N. L. C., & Maciejowski, J. M. (1996). Realization of stable models with subspace methods. *Automatica, 32*, 1587–1595.

Downs, J. J., & Vogel, E. (1993). A plant-wide industrial process control problem. *Computers and Chemical Engineering, 17*, 245–255.

Farina, L. A., Trierweiler, J. O., & Secchi, A. R. (2000). A systematic comparison of control structures proposed to the Tennessee Eastman benchmark problem. *Proceedings of ADCHEM 2000* (pp. 629–634). Pisa, Italy.

Godfrey, K. Ed. (1993). *Perturbation signals for system identification*. NY: Prentice-Hall.

Juricek, B. C., Larimore, W. E., & Seborg, D. E. (1998). Reduced-rank ARX and subspace models for process control. *Proceedings of IFAC Symposium on Dynamics and Control of Process Systems (DVCOPS-V)* (pp. 247–252). Corfu, Greece.

Larimore, W. E. (1983). System identification, reduced-order filtering and modeling via canonical variate analysis. *Proceedings of 1983 American Control Conference* (pp. 445–451). San Francisco.

Larimore, W. E. (1996a). *ADAPT$_X$ automated system identification software users manual*. Adaptics, Inc.. 1717 Briar Ridge Rd., McLean, VA, 22101.

Larimore, W. E. (1996b). Statistical optimality and canonical variate analysis system identification. *Signal Processing, 52*, 131–144.

Larimore, W. E. (1997). Optimal reduced rank modeling, prediction, monitoring, and control using canonical variate analysis. *Proceedings of IFAC 1997 International Symposium on Advanced Control of Chemical Processes.* (pp. 61–66). Banff, Canada.

Larimore, W. E. (1999). Automated multivariable system identification and industrial applications, Tutorial Paper. *Proceedings of 1999 American Control Conference* (pp. 1148–1162). San Diego, CA.

Ljung, L. (1995). *System identification toolbox user's guide* Nantick, MA.

Ljung, L. (1999). *System identification* (2nd ed.). Upper Saddle River, NJ: Prentice-Hall.

Maciejowski, J. M. (1995). Guaranteed stability with subspace methods. *Systems and Control Letters, 26*, 153–156.

McAvoy, T. J., & Ye, N. (1994). Base control for the Tennessee Eastman problem. *Computers and Chemical Engineering, 18*, 383–413.

Ricker, N. L. (1995). Optimal steady-state operation of the Tennessee Eastman challenge process. *Computers and Chemical Engineering, 19*, 949–959.

Ricker, N. L. (1996). Decentralized control of the Tennessee Eastman challenge process. *Journal of Process Control, 6*, 205–221.

Ricker, N. L. (2001). Tennessee Eastman challenge archive. Available at: http://depts.washington.edu/control/LARRY/TE/download.html.

Ricker, N. L., & Lee, J. H. (1995a). Nonlinear model predictive control of the Tennessee Eastman challenge process. *Computers and Chemical Engineering, 19*, 961–981.

Ricker, N. L., & Lee, J. H. (1995b). Nonlinear modeling and state estimation for the Tennessee Eastman challenge process. *Computers and Chemical Engineering, 19*, 983–1005.

Rivera, D. E., & Jun, K. S. (2000). An integrated identification and control design methodology for multivariable process system applications. *IEEE Control Systems Magazine, 20*, 25–37.

Sriniwas, R., & Arkun, Y. (1997). Control of the Tennessee Eastman process using input–output models. *Journal of Process Control, 7*, 387–400.

Tso, M. K. S. (1981). Reduced-rank regression and canonical analysis. *Journal of the Royal Statistical Society B, 43*, 183–189.

van Overschee, P., & De Moor, B. (1994). N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica, 30*, 75–93.

van Overschee, P., & De Moor, B. (1996). *Subspace identification for linear systems*. Boston, MA: Kluwer Academic Publishers.

Wise, B. M., & Gallagher, N. B. (1996). *PLS toolbox*. Manson, WA: Eigenvector Research, Inc..