MODEL PREDICTIVE CONTROL FOR BATCH PROCESSES: A LATENT VARIABLE APPROACH

Jesus Flores-Cerrillo and John F. MacGregor

Department of Chemical Engineering
McMaster University, Hamilton, On. Canada

Abstract: A novel multivariate model predictive control strategy (LV-MPC) for trajectory tracking and disturbance rejection for batch processes, based on multiway PCA models, is presented. It directly computes the manipulated variable trajectory adjustments over a future horizon using the structure of the PCA model. The advantages and the modest data requirements are illustrated using an emulsion polymerization process for temperature tracking. *Copyright © 2002 IFAC*

Keywords: Model predictive control, Principal component analysis, Batch processes, Missing data.

1. INTRODUCTION

Batch/semi batch processes are commonly used because their flexibility to manage many different grades and types of products. In these processes, one of the requirements to achieve consistent final quality specifications and adequate operation is that the low level controllers can adequate follow the set-points determined by master controllers or optimizers. Proportional-integral (PI) and Proportional-integral-derivative (PID) controllers are by far the most common approach used in industry. However, batch processes usually exhibit large time constants, time varying dynamics and need to track complex set-point trajectories. Under this situation the standard PID controller might not achieve adequate control performance. Enhancements to conventional PID controllers have proven to lessen some of these deficiencies (Clarke-Pringle and MacGregor, 1997).

Several advanced control approaches have been presented in order to further improve PID performance. Differential geometry methods have been applied for the control of batch and continuous processes (Kravaris *et al*., 1989; Clarke-Pringle and MacGregor, 1997) while Cott and Macchietto (1989), and Aziz *et al*., (2000) used generic model control to track batch reactor temperature set points.

A considerable number of studies using nonlinear model predictive control have also been presented. Özkan *et al*., (2001), for example, used nonlinear dynamic model control to track optimal reactor temperatures in the solution polymerization of styrene.

Lee *et al*., (1999) proposed a model predictive control technique for trajectory tracking aided with iterative learning. The methodology is illustrated for the temperature control of a laboratory batch reactor.

Statistical controllers for continuous processes based on Principal Component Analysis (PCA) have also been proposed (Chen and McAvoy, 1996; Shah *et al.,* 1998). However, their objective is mainly to regulate the controlled variables around a fixed operating set point, and the use of PCA is mainly to reduce the dimension of the control variable space.

The purpose of this paper is to introduce a novel multivariate control strategy based on latent variable models for set-point trajectory tracking and disturbance rejection in batch processes. The control strategy and manipulated variable adjustments, over a future horizon, are formulated in the space of a dynamic PCA model. The outline of the paper is as follows: in section 2 the methodology is introduced; in section 3, the control approach is illustrated using an emulsion polymerization process. Conclusions and future work are stated in section 4.

2. CONTROL METHODOLOGY

2.1 Basic Methodology

Modeling

The proposed algorithm uses historical databases and a few complementary identification experiments for model building. The empirical model is obtained using PCA.

The database used for building the PCA model, denoted as $\mathbf{\Omega}$, consists of K matrices, $\mathbf{X}_k$ (k=1,…,K), where K is the number of batches used for model building.

$$\mathbf{\Omega} = \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_k \\ \vdots \\ \mathbf{X}_K \end{bmatrix}$$

Each matrix $\mathbf{X}_k$ contains the information of k-th batch. For the sake of simplicity, the subscript k is omitted in the following explanation of the structure of $\mathbf{X}_k$.

Consider a batch run. Denote N as the total number of samples collected along the time. For sampling time i, define a vector ζ_i^T such that

$$\zeta_i^T = [\mathbf{x}_{me}^T \ \mathbf{y}_{cv}^T \ \mathbf{y}_{sp}^T \ \mathbf{u}_c^T]_i$$

where $\mathbf{x}_{me}$ is a vector of measurements on S on-line (and potentially off-line) process variables such as pressure, agitation, flows, etc. that can be incorporated to give information on disturbances and process conditions changes; $\mathbf{y}_{cv}$ is a vector of R controlled process variables needed to be tracked; $\mathbf{y}_{sp}$ is the vector of corresponding set-points; and $\mathbf{u}_c$ is a vector of A manipulated variables. The length of ζ_i^T is then equal to (S+2R+A). Notice that if time delays are important, these can be easily incorporated in ζ_i by simply shifting the delayed variables.

Every row vector of $\mathbf{X}$ is composed of vectors ζ_i^T at different sampling times (i), including the M-step future data, current data and the L-step past data.

$$\mathbf{X} = \begin{bmatrix} \zeta_1^T & \cdots & & \cdots & & \cdots & \zeta_{1+M+L}^T \\ \vdots & \ddots & & & & \iddots & \vdots \\ \zeta_{i-L}^T & \cdots & \zeta_{i-1}^T & \zeta_i^T & \zeta_{i+1}^T & \cdots & \zeta_{i+M}^T \\ \vdots & \iddots & & & & \ddots & \vdots \\ \zeta_{N-M-L}^T & \cdots & & \cdots & & \cdots & \zeta_N^T \end{bmatrix}$$

$$\underbrace{\qquad\qquad\qquad}_{\text{Past data}} \quad \underset{\text{Current data}}{\downarrow} \quad \underbrace{\qquad\qquad\qquad}_{\text{Future data}}$$

Therefore, the total dimension of matrix $\mathbf{\Omega}$ is [(N-M-L)K] × [(M+L+1) (S+2R+A)].

The data-set used for model building consists of data in which pseudo random binary sequences (PRBS) have been added on the output of the controller currently used (i.e. added on the manipulated variables). PRBS are needed to establish causal relationship between the manipulated variable changes and the tracked trajectories. Rebuilding the model by adding new batch data collected after implementing the proposed control scheme may also be done in order to further improve the causal relationship. The data requirements are further discussed in the examples.

PCA is then performed by projecting the scaled and mean centered matrix $\mathbf{\Omega}$ onto a reduced dimension score space $\mathbf{T}$:

$$\begin{aligned} \hat{\mathbf{\Omega}} &= \mathbf{T}\mathbf{P}^{\mathbf{T}} \\ \mathbf{T} &= \mathbf{\Omega}\mathbf{P} \end{aligned} \qquad (1)$$

where $\mathbf{P}$ is the loading matrix and $\mathbf{P}^T\mathbf{P} = \mathbf{I}$. In this paper the matrix $\mathbf{\Omega}$ is mean centered and scaled to unit variance. This mean centering simply subtracts the overall average value of each variable over all batches. An alternative is to mean center by subtracting the average trajectory of each variable as done in the statistical monitoring of batch processes (Nomikos and MacGregor, 1995). However, in this LV-MPC approach we are interested in performing trajectory tracking for different set-point patterns (not included in the training data), and so a PCA model for the complete trajectories rather than for the deviations about one given set-point pattern was considered more suitable. However, if several representative set-point trajectory patterns were used in the training data set, the mean centering about the average of those trajectories would be reasonable.

Control

Consider sampling time i. We can rearrange the vector $\lfloor \zeta_{i-L}^T \ \cdots \ \zeta_i^T \ \cdots \ \zeta_{i+M}^T \rfloor$ such that

$$\lfloor \zeta_{i-L}^T \ \cdots \ \zeta_i^T \ \cdots \ \zeta_{i+M}^T \rfloor \rightarrow [\mathbf{x}_1^T \quad \mathbf{x}_2^T]$$

where $\mathbf{x}_1^T$ contains the *known* information, including all past information (ζ_j^T, j=i-L, …, i-1) and future controlled variable set point trajectories ($\mathbf{y}_{sp,j}^T$, j=i,…,i+M), while $\mathbf{x}_2^T$ contains the *unknown* information, including future process variable measurements ($\mathbf{x}_{me,j}^T$,j=i,…,i+M), future controlled variables ($\mathbf{y}_{cv,j}^T$, j=i,…,i+M) and future control action adjustments ($\mathbf{u}_{c,j}^T$,j=i,…,i+M).

The loading matrix $\mathbf{P}$ can also be decomposed into two corresponding parts:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{bmatrix}$$

Two conditions need to be considered to compute $\mathbf{x}_2$. First, the unknown information $\mathbf{x}_2$ should keep the correlation structure as in the training dataset. Therefore, from the PCA model,

$$\mathbf{x}_2^T = \hat{\mathbf{t}}^T \mathbf{P}_2^T \qquad (2)$$

where $\hat{\mathbf{t}}^T$ is the estimated score vector for the current batch. Second, $\hat{\mathbf{t}}^T$ should be the projection in the score space of $[\mathbf{x}_1^T \quad \mathbf{x}_2^T]$. Therefore, $\hat{\mathbf{t}}^T$ can be obtained by solving:

$$\underbrace{\min}_{\hat{\mathbf{t}}} (\hat{\mathbf{t}}^T - \mathbf{x}_1^T\mathbf{P}_1 - \mathbf{x}_2^T\mathbf{P}_2)\mathbf{Q}(\hat{\mathbf{t}}^T - \mathbf{x}_1^T\mathbf{P}_1 - \mathbf{x}_2^T\mathbf{P}_2)^T$$
$$s.t. \quad \mathbf{x}_2^T = \hat{\mathbf{t}}^T\mathbf{P}_2^T \qquad (3)$$

The solution for (3) can be easily obtained (**Q**=**I**) as:

$$\hat{\mathbf{t}}^T = \mathbf{x}_1^T\mathbf{P}_1(\mathbf{I} - \mathbf{P}_2^T\mathbf{P}_2)^{-1} \qquad (4)$$

Then, $\mathbf{x}_2^T$ can be obtained by substituting (4) in (2):

$$\mathbf{x}_2^T = \mathbf{x}_1^T\mathbf{P}_1(\mathbf{I} - \mathbf{P}_2^T\mathbf{P}_2)^{-1}\mathbf{P}_2^T \qquad (5)$$

Equation (5) can be further restated as $\mathbf{x}_2^T = \mathbf{x}_1^T\mathbf{P}_1(\mathbf{P}_1^T\mathbf{P}_1)^{-1}\mathbf{P}_2^T$ since $\mathbf{P}_1^T\mathbf{P}_1 + \mathbf{P}_2^T\mathbf{P}_2 = \mathbf{I}$. A very important thing to note is that this equation is the same as the missing data estimation algorithm for $\mathbf{x}_2$ known as Projection to the Plane (Nelson *et al.*, 1996).

This control algorithm is recomputed at every sampling time *i* until completion of the batch. Only the current control actions ($\mathbf{u}_{c,j}^T$, *j*=*i*) are implemented.

In practice it is possible that the matrix $\mathbf{P}_1^T\mathbf{P}_1$ becomes ill conditioned. In this case it is recommended to obtain $\hat{\mathbf{t}}$ by using either a pseudo-inverse procedure based on single value decomposition or iteratively using (3) until convergence of $\hat{\mathbf{t}}$ is achieved. This last approach is the one taken in the simulation examples.

In the objective function (Equation 3), if desired, hard constraints to the scores, manipulated variables, and movement suppression factors can also be included.

An alternative modeling and control strategy is to simply build a PCA model by defining ζ_i^T as

$$\zeta_i^T = [\mathbf{x}_{me}^T \ \mathbf{y}_{cv}^T \ \mathbf{u}_c^T]_i$$

which does not contain controlled variable set point information. In the control calculation step, however, at sampling time *i*, the future controlled variable trajectories can be specified as their desired set-point trajectories, $\mathbf{y}_{cv,j}^T = \mathbf{y}_{sp,j}^T$, *j*=*i*,…,*i*+*M* and then included in $\mathbf{x}_1^T$. In this case, $\mathbf{x}_2^T$ will only include future measurements and future control actions that can be computed using (3). This is the approach used in the examples of section 3.

2.2 Adaptive multi-model methods

By using the structure in the last section, a single set of **P** coefficients is determined for the whole duration of the batch. Depending on the process, improved control may be obtained by using adaptive modeling. This can be accomplished by using a moving window (such as shown in figure 1) to model only the local behavior of the batch over the window centered at each time *i*. This will lead to multiple models, each one centered at a time point during the batch. To build these models, the window is placed over the data centered at time *i,* and the window weights applied to the past and future data relative to time *i*. This gives the following data matrix for a batch at time *i*:

$$\mathbf{X}^{(i)} = \begin{bmatrix} w(L+1-i)\times\left[\zeta_1^T \ \cdots \ \cdots \ \cdots \ \zeta_{1+M+L}^T\right] \\ \vdots \\ w(0)\times\left[\zeta_{i-L}^T \ \cdots \ \zeta_i^T \ \cdots \ \zeta_{i+M}^T\right] \\ \vdots \\ w(N+M-i)\times\left[\zeta_{N+M-L}^T \ \cdots \ \cdots \ \cdots \ \zeta_N^T\right] \end{bmatrix}$$

where $\mathbf{X}^{(i)}$ is the matrix **X** for sampling time *i*. Then a training dataset for sampling time *i* is obtained by

$$\mathbf{\Omega}^{(i)} = \begin{bmatrix} \mathbf{X}_1^{(i)} \\ \vdots \\ \mathbf{X}_k^{(i)} \\ \vdots \\ \mathbf{X}_K^{(i)} \end{bmatrix}$$

Loading matrix for sampling time *i*, $\mathbf{P}^{(i)}$, is computed by performing PCA on $\mathbf{\Omega}^{(i)}$. In the control algorithm, one only need to replace **P** by $\mathbf{P}^{(i)}$ to compute the control action at sampling time *i*.

Different weighting vectors can be used. Here we use a weighting vector as shown in Figure 1. This weighting vector consists of two mirrored sigmoid functions, which can be calculated using:

$$w(\theta) = \begin{cases} \dfrac{1}{1+\exp(6.9068\dfrac{\theta-\mu_1}{\mu_2})} & \theta \geq 0 \\ \dfrac{1}{1+\exp(-6.9068\dfrac{\theta+\mu_1}{\mu_2})} & \theta < 0 \end{cases}$$

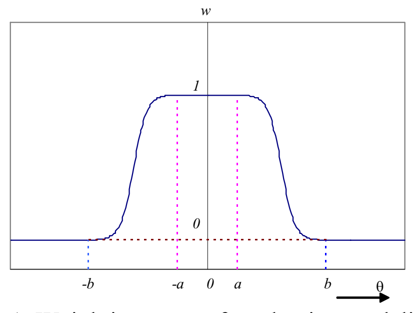

Figure 1. Weighting vector for adaptive modeling

where $\mu_1 = (a+b)/2$, $\mu_2 = (b-a)/2$ (a and b are two adjusting factors, which can be used to change the shape of the weighting vector). For other adaptive PCA methods, please refer to Rännar *et al*., (1998).

3. SIMULATION EXAMPLES

The control algorithm was tested using two non-linear simulators: the emulsion polymerization of styrene (Lynch and Kiparissides, 1981) and the solution polymerization of vinyl acetate (Teymor, 1997). However, for brevity, only results on the emulsion polymerization process are presented.

Lynch and Kiparissides (1981) developed a non-linear model, with simple kinetics, for the styrene emulsion polymerization. This model, originally developed for tubular reactors with full recycle, has been adapted for use in batch and semi-batch processes. For a complete description of the model and model parameters the reader is referred to the original publication. The non-linear simulator is used for data generation and controller performance evaluation. The control objective is to perform reactor temperature (T_r) trajectory tracking by adjusting the inlet jacket temperature (T_j). On-line T_r measurements, considered to be available every 30sec, are corrupted by normally distributed random error with standard deviation σ=0.15°K. Simulation time is 300min. Control action is taken every 30sec.

3.1 Control studies

In order to evaluate the control algorithm, several studies involving the effect of information content of data available for model building and the type of model (adaptive versus fix) were performed for different set points and disturbances. In what follows some of these studies are presented.

As shown in section 2, PCA modeling allows having different species for model building. For example, one can use $\boldsymbol{\zeta}_i^T = [\mathrm{T_{r,sp}}\ \mathrm{T_j}\ \mathrm{T_{r,cv}}]_i^T$ or simply $\boldsymbol{\zeta}_i^T = [\mathrm{T_j}\ \mathrm{T_{r,cv}}]_i^T$ ($\boldsymbol{\zeta}_i^T = [\mathbf{u}_c^T\ \mathbf{y}_{cv}^T]_i$). Both situations were studied. For brevity only results in which $\zeta_i^T = [\mathrm{T_j}\ \mathrm{T_{r,cv}}]_i^T$ are presented.

The first step in model building is to obtain a data set. In most of the situations (at least for temperature control of batch reactors), the base controller is PI or PID. Therefore, it is feasible to assume that closed loop data containing some of the effects that T_j and normal operating disturbances have on T_r is available. However, in order to further establish a casual relationship between u_c= T_j and y_{cv}= T_r, a few complementary experiments, in which PRBS are added on the top of the output of the PI controller (u_c) are needed. Two scenarios are illustrated here: in the first one a dataset is obtained from a system in which a fine tuned PI (denoted as fine PI) is employed together with PRBS, while in the second one a slightly sluggish PI (denoted as sluggish PI) is used together with smaller PRBS. Comparison between the proposed algorithm and PI is given (PID performance is similar to those obtained from PI and therefore not shown).

Figure 2a shows the performance of the model predictive control algorithm for the T_r trajectory tracking of Model 1, set-point 1 of Table 1 (set-point 1 is the set-point from which the model was identified, Figure 3) versus that obtained from a fine tuned PI. Figure 2b shows their corresponding manipulated variable. It is evident that the trajectory tracking of T_r from the LV-MPC (Model 1) is better than that of PI, while having a much smoother manipulated variable behavior.

The good control performance can be further observed by inspecting the mean absolute error $MAE_y = \sum_{i=1}^{N} \left| y_{cv,i} - y_{sp,i} \right| \big/ N$ for the CV and the $MAE_\nabla = \sum_{i=1}^{N} \left| u_{c,i} - u_{c,i-1} \right| \big/ N$ for the changes in the MV in Table 1. It can be seen that smaller values of $\mathrm{MAE_y}$ and $\mathrm{MAE_\nabla}$ are obtained by using the LV-MPC algorithm.

The data requirements to build Model 1 are small considering that *only 3 batches* were used and that the PRBS do not affect substantially the batch operation as can be seen in Figure 3. In this Figure (···) indicates the control obtained when in the MV (T_j) has been added PRBS, and (—) that under normal operation (no PRBS in the PI output). The MAE of the CV for this data set is $\mathrm{MAE_y}$=0.85 (only 30% higher than when no PRBS is added).

One of the advantages in using dynamic PCA is that the usefulness of the model is not limited to only the set point trajectory from which the model was identified. This is illustrated in Figure 4a for a different set point (set point 2 of Table 1) using Model 1 (model identified using the data shown in Figure 3) and compared to that obtained using a fine tuned PI. Figure 4b shows their corresponding manipulated variables. Clearly, the LV-MPC can be used for tracking different set-points. Some reasons for this are: 1) causation is introduced into the model by the PRBS's, which give information on how the $\boldsymbol{u}_c$ trajectory affects the $\boldsymbol{y}_{cv}$ trajectory, 2) the model focuses on local trajectory information rather than in the shapes of the complete trajectory, and 3) the correlation structure of the model over the local horizons (i-L,…i+M) does not change significantly for the different set-point conditions. If any deterioration in the control is evident for a new set-point trajectory, the result can be added to the previous training data and the model refitted.

In spite of that the data requirements for Model 1 are not very demanding, it is not uncommon to find processes in which the PI performance is sluggish. Model 2 and 3 of Table 1 were designed in order to evaluate the performance of the algorithm when the *data* used for model building is obtained with a more sluggish PI.

Table 1 Control Results for the emulsion polymerization of styrene

Control	Model	Model type	Data from PI tuned	Set point 1		Set point 2	
				MAE_y	MAE_∇	MAE_y	MAE_∇
PCA	1	Fix	fine	0.49	4.5	0.32	2.8
	2	Fix	sluggish	0.72	4.7	0.48	2.9
	3	Adaptive	sluggish	0.34	4.3	0.37	4.3
PI			fine	0.66	13.4	0.35	14.1
			sluggish	1.09	5.9	0.54	6.1

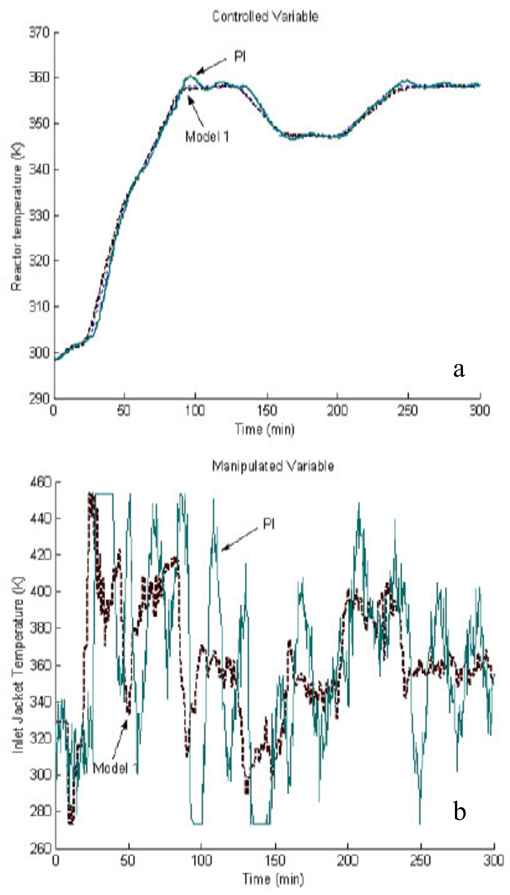

Fig. 2a-b. Control performance for Model 1 vs. fine-tuned PI for set-point 1. In Figure 2a (···) represents T_{sp}, while in Figure 2a/2b (– –) T_r/T_j from Model 1 and (—) from the fine-tuned PI.

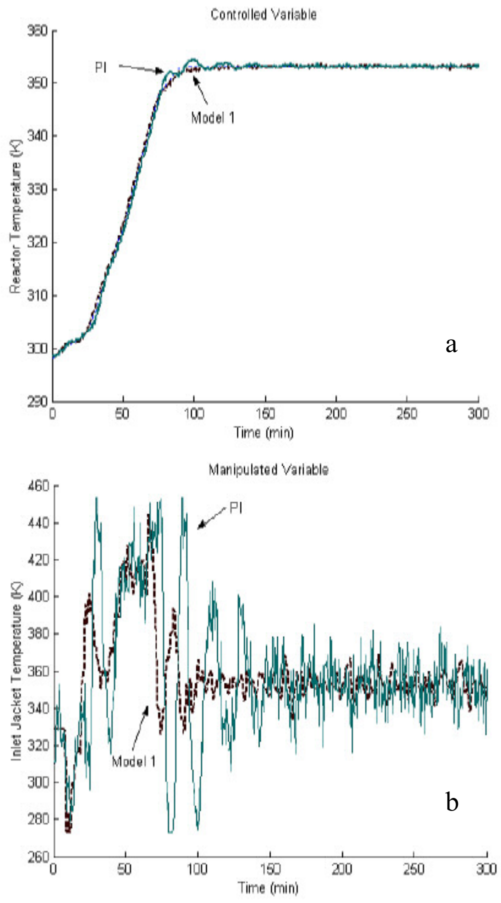

Fig. 4a-b. Control performance for Model 1 vs. fine-tuned PI for set point 2. In Figure 4a (···) represents T_{sp}, while in Figure 4a/4b (– –) T_r/T_j from Model 1 and (—) from the fine-tuned PI.

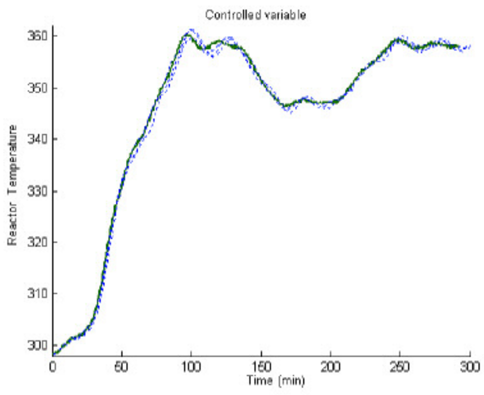

Fig. 3. Dataset used for building Model 1. (···) indicates the control obtained when in the MV (T_j) has been added PRBS, and (—) that under normal operation for the fine-tuned PI.

Figure 5a shows the performance of the control algorithm for the T_r tracking of Model 2 (set-point 1) versus that obtained from the sluggish PI. Figure 5b shows their corresponding manipulated variable. It is clear (see Table 1) that also in this situation the performance of the proposed control algorithm (Model 2, MAE_y=0.72) is better than that of PI (MAE_y=1.09), while also having a smother manipulated variable behavior. However, it seems that at certain instances, the control from the data based algorithm is a little sluggish. This can be further corrected by using adaptive modeling (Model 3), which as can be seen in Figure 5a and Table 1 (MAE_y=0.34), it practically overlaps on the set points. Notice that the advantage of the predictive part of the algorithm is evident in Figure 5b, where the manipulated variable obtained from the algorithm anticipates the changes in the set-points, while the MV of the PI responds much slower.

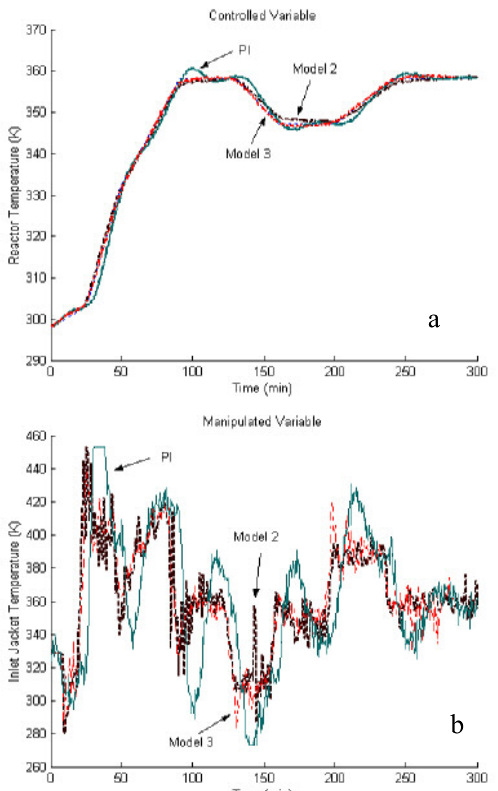

Fig. 5a-b. Control performance for Model 2 and 3 vs. sluggish PI. In Figure 5a (···) represents T_{sp}, while in Figure 5a/5b (**— —**) T_r/T_j from Model 2, (**—·—**) form Model 3 (Adaptive) and (—) from the PI.

From Table 1 and Figures 2,4 and 5, it can be seen that the control performance of the proposed LV-MPC control strategy is generally superior to that obtained using PI even in cases that the data set used for model building contains no information on new set-points.

The effect of the information content on the data-set is evident from the results: if a model is obtained using an initially better tuned PI the resulting LV-MPC would perform better than when the model is obtained using a sluggish PI (This situation can be corrected, to a considerable extent, by using adaptive modeling as can be seen in Table 1 and Figure 5a). Moreover, independently of the PI and PRBS used to generate the model, the performance obtained from the data based method generally outperforms that obtained from PI (Similar conclusion were observed when using other recipes and when using a solution polymerization system (Teymor, 1997). However, these results are not shown here for brevity.)

4. CONCLUSIONS AND FUTURE WORK

A novel and simple multivariate model predictive control strategy (LV-MPC) for trajectory tracking based on PCA models is presented. It computes M step ahead trajectory manipulated variable adjustments using the structure of the PCA model. The advantages of the control algorithm and data requirements are illustrated using an emulsion polymerization process for reactor temperature tracking.

The performance of the LV-MPC is shown to be very good in comparison to traditional PI controllers, not only for achieving tighter trajectory tracking, but also by doing so with much less effort in the manipulated variables. Perhaps the most interesting aspect of designing these LV-MPC algorithms is that the data requirements for model identification are very modest.

REFERENCES

Aziz, N., M. A. Hussain and I. M. Mujtaba (2000). Performance of different types of controllers in tracking optimal temperature profiles in batch reactors. *Computers chem. Eng*., **24**, 1069.

Chen, G. and T. J. McAvoy (1996). Process control utilizing data based multivariate statistical models. *The Canadian Journal of Chemical Engineering*. **74**, 1010.

Clarke-Pringle, T. and J. F. MacGregor (1997). Nonlinear adaptive temperature control of multi-product, semi-batch polymerization reactors. *Computers chem. Eng.,* **21** (12), 1395.

Cott, B. J. and I. M. Mujtaba (1989). Temperature control of exothermic batch reactors using generic model control. *Ind. Eng. Chem. Res.,* **28**, 1177.

Kravaris, C., R. A. Wright and J.F. Carrier (1989). Nonlinear controllers for trajectory tracking in batch processes. *Computers Chem. Eng*., **13** 73.

Lee, K. S., I. Chin, H. J. Lee and J. H. Lee (1999). Model Predictive Control Technique Combined with Iterative Learning for Batch Processes. *AIChE J.* **45** (10) 2175.

Lynch, D. and C. Kiparissides (1981). Numerical Simulation of a Tubular Polymerization Reactor. *J. of Applied polymer Science*, **26**, 1283.

Nelson, P., J. F. MacGregor and P.A. Taylor (1996). Missing Data Methods in PCA and PLS: Score Calculations with Incomplete Observations. *Chem. & Intel. Lab. Sys*., **35**, 45.

Nomikos, P., and J.F. MacGregor (1995). Multi-way Partial Least Squares in Monitoring Batch Processes, *Chemometrics and Intelligent Laboratory Systems*, **30**, 97.

Özkan, G., S. Özen, S. Erdoğan, H. Hapoğlu and M. Alpbaz (2001). Nonlinear control of polymerization reactor. *Computers chem. Eng*., **25**, 757.

Rännar, S., J. F. MacGregor and S. Wold (1998). Adaptive batch monitoring using hierarchical PCA. *Chem. & Intel. Lab. Sys*., **41**, 73.

Shah, S., M. Randy, H. Takada, K. Morinaga and T. Satou (1998). Modeling and Control of a Tubular Reactor: A PCA Based Approach. *Proceedings of the 5th IFAC Symposium on Dynamics and Control of Process Systems*, 17.

Teymour, F. (1997). Dynamics of Semibatch Polymerization Reactors: I. Theoretical Analysis. *AIChE J.* **43** (1) 145.