ELSEVIER

Chemometrics and Intelligent Laboratory Systems 23 (1994) 149–161

Chemometrics and intelligent laboratory systems

Exponentially weighted moving principal components analysis and projections to latent structures

Svante Wold

Research Group for Chemometrics, Umeå University, S-901 87 Umeå, Sweden

(Received 15 June 1993; accepted 22 November 1993)

Abstract

For stable (non-dynamic) chemical processes characterized by multivariate data, principal components analysis (PCA) and projections to latent structures (PLS) have recently been shown to provide useful monitoring schemes. In this work, PCA and PLS are generalized to dynamically updated models for modelling processes with memory and drift. These models are based on exponentially weighted observations, and are formulated as multivariate generalizations of the exponentially weighted moving average (EWMA). Principles and estimation algorithms for EWM-PCA and EWM-PLS are presented, and predictive control schemes based on these models are discussed.

1. Introduction

With process computers and electronic sensors, highly multivariate data are routinely collected in processes manufacturing chemicals, polymers, pulp and paper, ore, oil and petroleum, and many other products. As discussed by Wise and Ricker [1] and others [2–5], the multivariate analysis of these data by principal components analysis (PCA) and projections to latent structures (PLS) provides models that allow the interpretation of the data and scores that are useful summaries of the state of the process. The standard PC and PLS models, however, assume independence of the process time points (no process memory). Since the PC and PLS scores comprise good cross section summaries of the process data, a natural way to model 'memory effects' would be to develop simple time series models in these scores. One of the simplest such models is provided by the exponentially weighted moving average (EWMA) which gives both a good picture of the current status of the process, and a one-step-ahead forecast. Hence, EWMA models based on the multivariate scores from PCA or PLS would be a natural extension of the standard multivariate models for process applications.

The generalization of EWMA to EWM-PCA and EWM-PLS consists of two parts. The first relates to the use of scores instead of individual variables in control charts and predictions. The second is the dynamic updating of PC and PLS models themselves, to allow for drift in the process.

The present article concerns how to set up and estimate EWM-PC/PLS models.

2. Examples of applications

The obvious applications area of EWM-PCA/PLS is process monitoring and control.

SSDI 0169-7439(93)E0075-F

Multiple responses are common in all areas today, both because it is simple and cheap to measure many performance indicators, and because complicated products need many criteria to be monitored and controlled to assure high quality. As a small example illustrating EWM-PCA we use a 49×17 matrix of paper machine data collected over a time period of 49 even time intervals. The 17 variables include the weight of the pulp, moisture, burst tension, etc., until machine speed *.

A particular type of processes of increasing interest are those in our environment. In the monitoring of pollution in rivers, lakes, oceans, air, etc., many responses are usually measured, and EWM-PCA/PLS would provide a compact and easy way to view a representation of the data.

EWMA was first used to monitor and predict economic processes as reviewed by Hunter [6]. Since many economic indicators usually are monitored together, EWM-PCA/PLS would be an interesting alternative to EWMAs of the single indicators.

In chemistry and chemical engineering, other sequences than those of time are often studied. Natural polymers, e.g., cellulose, DNA and proteins, are sequences of varying monomers where local EWM-PCAs of monomer properties might be used to obtain information about such things as binding sites, sites of folding, etc. In such applications, it may be natural to extend the exponentially decreasing weights in both directions from the 'center point' of the model.

3. Brief review of exponentially weighted moving averages (EWMA)

A lucid overview of EWMA is given by Hunter [6], and applications are discussed by Lucas and Saccucci [7] and MacGregor and Harris [8]. EWMA can be viewed as having two components. The first concerns the modelling of a variable, y, and predicting it at the next time point. The second concerns the construction of a control chart on the basis of the model.

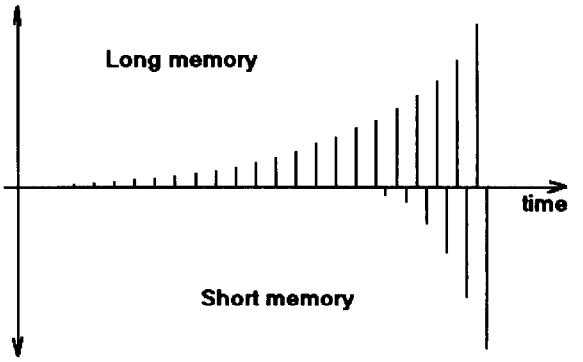


Fig. 1. Exponentially decreasing weights with long memory (small) and short memory (large).

The basic idea of EWMA is to model y as a weighted moving average, with recent observations weighted more heavily than earlier observations. Exponential weights

$$v_i = \lambda(1-\lambda)^{t-i} \tag{1}$$

are used for the ith observation preceding the present one ($i = t$), see Fig. 1. This gives the *predicted* value at time $t+1$ as Eqs. (2) and (3). These equations can at the same time be used to recursively update the EWMA model from time t to time $t+1$:

$$\hat{y}_{t+1} = \lambda y_t + (1-\lambda)\hat{y}_t \tag{2}$$

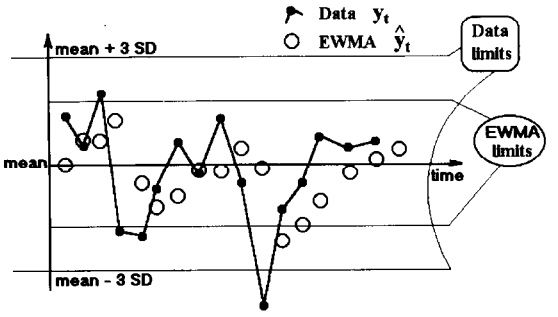
$$= \hat{y}_t + \lambda(y_t - \hat{y}_t) = \hat{y}_t + \lambda e_t \tag{3}$$

Assuming that the residuals, e_i, have a constant variance σ^2, the variance of the EWMA is

$$\mathrm{Var(EWMA)} = \sigma^2\lambda/(2-\lambda) \tag{4}$$

The corresponding standard deviation (SD) can be used to construct control limits as, for instance, 3σ limits. Thus, the EWMA chart, like Shewhart and Cusum charts, can be used as a *monitoring* device to indicate if the process is significantly off target to warrant an action (Fig. 2). However, since the model also provides a prediction of y at the next time point, EWMA can also be used as a basis to modify the process to decrease the difference between the prediction and the target value, i.e., dynamic process con-

* I am indebted to John MacGregor for access to these data.



$$\text{EWMA} = \hat{y}_{t+1} = \hat{y}_t + \lambda e_t = \hat{y}_t + \lambda(y_t - \hat{y}_t)$$

Fig. 2. The exponentially weighted moving average (EWMA) control chart together with the ordinary Shewhart chart.

trol. For that purpose, Hunter [6] recommends a modified EWMA corresponding to the PID controller, as

$$\hat{y}_{t+1} = \hat{y}_t + \lambda_1 e_t + \lambda_2 \Sigma e_t + \lambda_3 (e_t - e_{t-1}) \quad (5)$$

The values of the parameters λ_1 to λ_3 are estimated from the process history.

4. Principal components analysis, PCA

The analysis of an $N \times K$ data matrix, **Y**, usually starts with the matrix centered and scaled to unit column variance. PCA models this normalized matrix as a product of an $N \times A$ score matrix, **T**, and an $A \times K$ loading matrix, **P**′, plus an $N \times K$ residual matrix, **E**. Reviews of PCA are given by, e.g., Jackson [9], Jolliffe [10], and Wold, Esbensen and Geladi [11]. The number of product terms, components, A, defines the dimensionality of the PC model. If the number of product terms, A, equals the smaller of the dimensions of **X**, N or K, the residuals **E** are identically zero. The number of *significant* components, A, can be estimated in many ways; we advocate cross-validation [12,13].

$$\mathbf{Y} = \sum_a \boldsymbol{t}_a * \boldsymbol{p}_a' + \mathbf{E} = \mathbf{TP}' + \mathbf{E} \quad (6)$$

The scores (columns of **T**) are orthogonal, and in many ways provide the best summary of the data. Hence, plotting these scores against time gives a good picture of how the process evolves, as discussed by Wise and Ricker [1] and others [2–5].

For the unweighted calculation of the principal components vectors, $\boldsymbol{t}_a$, and $\boldsymbol{p}_a$, singular value decomposition (SVD) is the preferred method if all components are wanted ($a = 1, 2, \ldots, \min(N, K)$). If we are interested only in the first few PCs, however, the nonlinear iterative partial least squares (NIPALS) method originated by Fisher and MacKenzie [14] and H. Wold [15] is faster than SVD, because only these first PCs are actually calculated. The NIPALS interpretation of the loading values (p_{ak}) as partial regression coefficients makes the calculation of weighted PC models straightforward as discussed below.

$$\mathbf{E}_{a-1} = \{e_{ik,a-1}\} = \mathbf{X} - \sum_b^{a-1} \boldsymbol{t}_b * \boldsymbol{p}_b' \quad (7)$$

$$e_{ik,a-1} = t_{ia} * p_{ak} + e_{ik,a} \quad (8)$$

$$p_{ak} = \sum_i^N (e_{ik,a-1} * t_{ia}) / \sum_i^N (t_{ia} * t_{ia}) \quad (9)$$

The elements in $\boldsymbol{p}_a$ are usually normalized to unit length ($\| \boldsymbol{p}_a \| = 1$), giving

$$t_{ia} = \sum_k y_{ik} p_{ak} \quad (10)$$

The row i standard deviation (SD) of the residuals, s_i, is a measure of the distance between the ith observation vector and the PC model. Hence, this is often called DMod for distance to model.

5. Exponentially weighted moving principal components analysis (EWM-PCA)

To develop EWM-PCA we need two parts. The first, updating and forecasting the process values in the next time point, $t + 1$, is straightforward if we assume the existence of a PC model of the process. The second part, to update this PC model for a drifting process, turns out to be more complicated.

5.1. The prognostic part

With K response multivariate data, $\mathbf{Y} = \{y_1, y_2, \ldots, y_k, \ldots, y_K\}$, and a PC model with A components estimated from the **Y** data, a process time point has with it A associated score values,

t_{ia} ($a = 1, 2, \ldots, A$), constituting a row of the score matrix **T**.

If we now assume some autoregressive autocorrelation structure in the data, but a stable cross-section correlation structure and hence a stable PC model, EWMAs in the scores, t_a, will provide a basis for multivariate monitoring and dynamic process control. This is similar to the multivariate control charts proposed by Crosier [16], Hawkins [17], and Lowry et al. [18], except that they construct a single control chart based on all variables and the inverse of their covariance matrix, **S**. This corresponds to assuming the process variables to be independent.

Here we assume that the process actually is driven by only A independent 'latent' variables — conceptually similar to state variables — that are indirectly observed by the **Y** variables, and estimated as the scores t_a ($a = 1, 2, \ldots, A$). This gives two alternatives for control charts. Either we make one control chart for each PC component, a, which makes sense if these components have a separate physical meaning. An additional control chart can be constructed from the residual SDs, the DMod chart. A second alternative is obtained by combining all significant t values and DMod to a single chart, which, however, leads to a loss of the information about the separate model dimensions.

The forecast of the score vector $\boldsymbol{t}$ (with A elements) at time $t + 1$ is, in analogy to Eqns. (2) and (3),

$$\hat{\boldsymbol{t}}_{t+1} = \lambda \boldsymbol{t}_t + (1 - \lambda)\hat{\boldsymbol{t}}_t \tag{11}$$

$$= \hat{\boldsymbol{t}}_t + \lambda(\boldsymbol{t}_t - \hat{\boldsymbol{t}}_t) \tag{12}$$

The more elaborate form analogous to Eqn. (5) is obvious. These predicted score values, in turn, forecast the vector of M variables, $\boldsymbol{y}$, as

$$\hat{\boldsymbol{y}}_{t+1} = \hat{\boldsymbol{t}}_{t+1}\mathbf{P}' \tag{13}$$

The variance of $\hat{t}_{t+1,a}$ is given directly by Eq. (4), with σ_a^2 estimated from the scores of a longer data history. Because of the non-full rank of the matrix **Y**, classical variances of $\hat{\boldsymbol{y}}_{t+1}$ cannot be estimated without some further assumptions. If we make the 'partial least squares assumptions' [15] about some independent regularity of each y_k, a reasonable variance of the forecasted y_k would be

$$\mathrm{Var}(\hat{y}_{k,t+1}) = \sum_a p_{ka}^2 \sigma_a^2 \tag{14}$$

5.2. Updating the model

Second, we need to update the dynamic exponentially weighted PCA, a way to deal with the rotation problem discussed below, specifications of centering and scaling, and a few other things. These pieces will now be dealt with one at a time.

Weighted PCA

Using exponentially decreasing object weights, v_i, we obtain directly from the weighted least squares formalism and Eq. (9):

$$p_{ak} = \frac{\sum_i^N (v_i * e_{ik} * t_{ia})}{\sum_i^N (v_i * t_{ia} * t_{ia})} \tag{15}$$

Hence, the NIPALS algorithm can be used directly with only a minor modification in the calculation of the EWM-PCA loadings using the exponentially decreasing weights, v_i, according to Eq. (1). For a single, fixed **Y**, the other NIPALS steps remain unchanged. The unweighted scores, t_a, are no longer orthogonal, however, but the weighted ones, $t_{ia}v_i^{1/2}$, are. This is as it should.

Centering and scaling

Before any multivariate modelling, the data are usually centered by subtracting the column means from the data matrix. The mean vector can be interpreted as a first loading vector, $\boldsymbol{p}_0$, with the corresponding score vector, $\boldsymbol{t}_0$, having all elements equal to $1/N$.

In the present context, there are two natural ways to calculate the centering vector; either to use a constant vector of means estimated from a long history of process data, or to use the EWMA for each variable (y_k) with a much smaller λ than used in the EWM-PCA weights (v_i). To stabilize the estimation of this EWMA_k, it will be calculated using the residuals of the PC model instead

of the (normalized) raw data. Thus, the observation vector y_{t+1} is first centered and scaled using the parameters at time t. Thereafter the predicted values (Eq. (13)) are subtracted to give residuals, e_{t+1}, which are used to update the EWMA_k according to Eqs. (1) and (2).

After the centering, the data are usually scaled by multiplying each column in the data by a scaling weight, ϕ_k. With variance scaling (autoscaling), ϕ_k is calculated as $1/s_k$, with s_k being the column standard deviation. This leads again to two obvious choices; to calculate s_k from a long process history, or to use an updating computation of s_k based on the weighted local data. A third choice based on a slowly updated 'spanning data base' will be discussed further below. Important variables can be scaled up by multiplying ϕ_k by a value between 1 and 3, and conversely, other 'unimportant' variables can be suitably scaled down.

The rotation problem and its stabilization

Any bilinear model — PCA or PLS — is undefined with respect to a rotation (Fig. 3). In dynamically updated models this leads to a potential instability; when a new process point is brought in it can lead to a rotation of the previous model even if the new point is very close to the model plane. This will be displayed as a jump

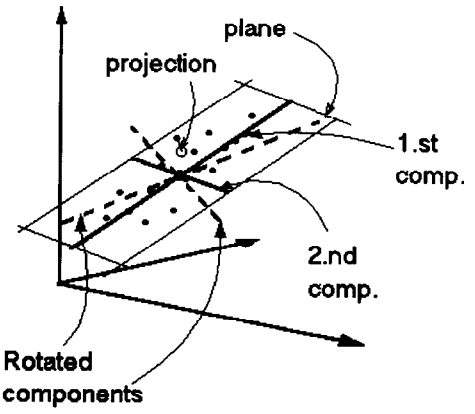


Fig. 3. A projection (PCA or PLS model) is the same even if it is expressed as two different coordinate systems rotated with respect to each other.

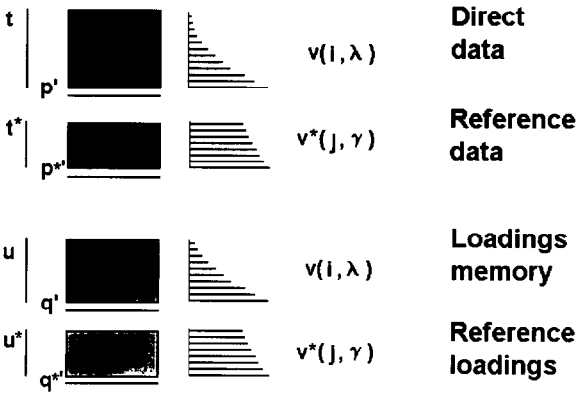


Fig. 4. The data and loading blocks in the EWM-PCA algorithm.

in score plots, and hence incorrectly interpreted as a change of the process.

To avoid this unwarranted rotation, the loading vectors from the previous models are saved in an auxiliary '**P** memory matrix' ('**W** memory' in PLS), here denoted by **Q**. Then, in the calculation of the updated model, this **P** memory matrix — exponentially weighted — is included in the estimation according to the multi-block PCA/PLS algorithm published by Wold et al. [19,21]. This can be seen as an empirical Bayesean estimation of the PC model, where prior information about the loadings is given by **Q**, the **P** memory matrix (see Fig. 4).

The consequence of including the memory matrix is that the updated loadings, p_a (or w_a in PLS) are forced not to differ too much from the previous ones. The balance between new and old values are controlled by an adjustable parameter, α. The full algorithm is given in the appendix.

Loss of process memory in periods of stability

In periods of stability, any recursive model estimation has a tendency to lose the information about previous periods [20]. This is because if the process is stable long enough, only data without appreciable variation are retained, and earlier data are weighted down to have insignificant influence by the exponentially decreasing weights.

To force the model to remember important events further back in its history, a second auxil-

iary matrix, a reference data matrix, $\mathbf{Y}^*$, is here included in the model estimation. This matrix contains the process data points that span the data space, and it is updated whenever a process point has a score value (t_a) that exceeds a preset limit. This limit can be derived from the data history, or be preset by the operator.

Analogously, an additional reference matrix for the loadings, $\mathbf{Q}^*$, is included so that the process memory about the loadings does not disappear in periods of stability (see Fig. 4).

As in the **P** memory matrix (**Q**), the rows in the reference matrices ($\mathbf{Y}^*$ and $\mathbf{Q}^*$) are exponentially weighted, but with a slower decreasing rate, using a smaller value γ instead of λ in Eq. (1).

Protecting against spikes

Many processes now and then produce 'spikes', i.e., aberrant values that should not be included in the modelling. The easiest way to cope with these spikes is to calculate the distance in Y space between the new and the previous observation. Data with widely different values, including those producing score values (t_a) far outside the 'normal ranges' as judged from the score values of the reference data (see above), are discarded after a signal to the operator, unless several process points in a row show this widely differing pattern.

Estimating the parameters λ_1, λ_2, and λ_3

Applying EWM-PCA (or PLS) on a set of historical data with a given set of parameter values of λ_1 to λ_3 gives prediction errors for the one-ahead forecasts for each score, t_a, and for each y variable. The sum of squares of the differences between actual values and forecasts thus constitutes a measure of the predictive power of the model in the same way as with cross-validation. This sum, PRESS, has components from each score, or y variable, or both, weighted according to their perceived importance. To find the best combination of the values of λ_1 to λ_3, a response surface modelling (RSM) approach is recommended. In this approach, 15 models with different parameter combinations are evaluated in parallel. The 15 parameter combinations ($j = 1, 2, \ldots, 15$) are selected according to a central composite inscribed (CCI) design, with low and high values being around, say, 0.15 and 0.45. This is followed by a regression of $y = \log(\mathrm{PRESS}_j)$ against the extended design matrix of $\mathbf{X} = \lambda$ (including squares and cross-terms), giving a predicted combination of parameter values that will give a minimal PRESS.

6. EWM-PCA / PLS; putting it all together

To develop a dynamic PCA, or PLS, model that meets the objectives to (i) be a good summary of the local process data, (ii) be stable against unwarranted rotation, (iii) have an optional long term memory, and (iv) be robust against spikes, the following is proposed:

(a) PCA or PLS is used as model of aggregation, suitably modified to meet the other requirements. The choice of PCA or PLS depends on the structure of the problem; whether 'predictor' variables **X** are present in addition to the responses, **Y**.

(b) Saving previous loadings (PCA) or weights (PLS) in a memory matrix, **Q** ($\mathbf{P}_{\mathrm{mem}}$ or $\mathbf{W}_{\mathrm{mem}}$). In the calculations this memory matrix, appropriately weighted, is included as a block in the multi-block PCA or PLS modelling. This encourages the updated model to look like the previous one. The balance between the 'new' loading and the one derived from the memory block is controlled by an adjustable parameter α.

(c) Saving 'spanning' loadings or PLS weights in a long term memory matrix, $\mathbf{Q}^*$ ($\mathbf{P}_{\mathrm{ref}}$, or $\mathbf{W}_{\mathrm{ref}}$). This long term memory matrix, appropriately weighted, is also included as a block in the calculations (see Fig. 4).

(d) Saving 'spanning' data vectors in data memory matrices, $\mathbf{Y}^*$ and $\mathbf{X}^*$ ($\mathbf{Y}_{\mathrm{ref}}$ in PCA, or $\mathbf{X}_{\mathrm{ref}}$ in PLS). These data memory matrices, appropriately weighted, are included as an additional data block in the calculations.

(e) Checking incoming data vectors for spikes by allowing a maximum distance to previous data or to the data in the data memory matrix.

The algorithm in its simpler form for EWM-PCA becomes a four object block PCA, very similar to a multi-block PLS [19,21] but in the

object direction. For EWM-PLS, additional blocks are included for the Y side. Overviews are given below, and the details are given in the appendix.

7. Overview of the EWM-PCA and EWM-PLS algorithms

1. Select the values of the parameters λ_1 to λ_3 in Eq. (5) (or Eqs. (11) and (12) in the simpler case). This is done based on experience or estimation of the values giving the best one-ahead predictions in a longer process history.
2. Select a starting matrix, $\mathbf{Y}_0$, as the process data in the beginning of the time interval of interest. For PLS, the two starting matrices $\mathbf{X}_0$ and $\mathbf{Y}_0$ are needed. From these, calculate column averages and standard deviations for the centering and scaling of the data.
3. Use weighted PCA (or PLS) to derive an initial model of the process from the normalized data of step 2. Use cross-validation to select the appropriate number of components, A.
4. Start the data memory matrix by including the data vectors from $\mathbf{Y}_0$ in PCA and $\mathbf{X}_0$ in PLS that correspond to the maximum and minimum score values of each model dimension, a.
5. Start the loading or weight memory matrices, $\mathbf{Q}_a$ ($\mathbf{P}_{\mathrm{mem},a}$ or $\mathbf{W}_{\mathrm{mem},a}$), one for each component, a, with $\boldsymbol{p}_a'$ or $\boldsymbol{w}_a'$ as the first and single rows.
6. Start $\mathbf{Q}^*$, the long term spanning $\mathbf{P}_a$ or $\mathbf{W}_a$ matrices identical to those in step 5 above.
7. Make the one-ahead forecasts of the scores; $\hat{\boldsymbol{t}}_{a,t+1}$. Calculate predicted y values from $\hat{\boldsymbol{t}}_{a,t+1}$ and $\mathbf{P}'$ (PCA) or $\mathbf{C}'$ (PLS).

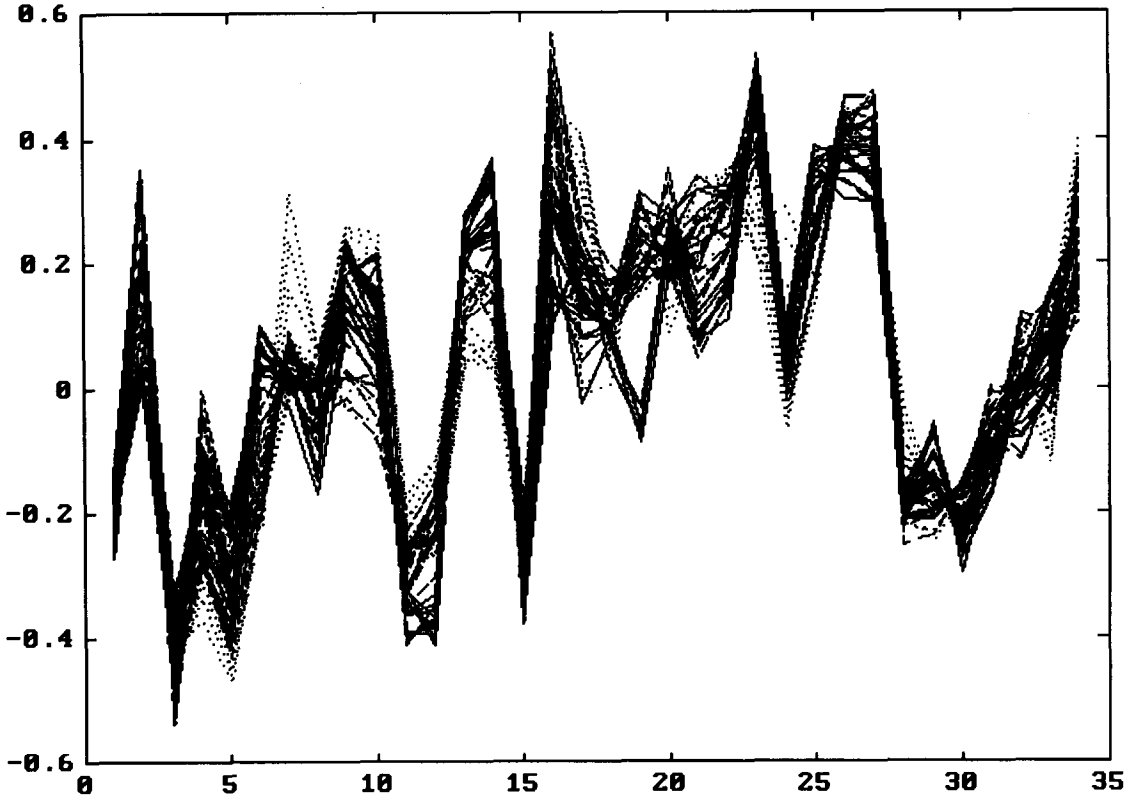


Fig. 5. The loading vectors of the 42 EWM-PCA models of $t = 8, 9, \ldots, 49$. Points 1–17 are the elements of the first loading vector ($a = 1$) and points 18 to 34 are the elements of the second loadings ($a = 2$). $\lambda = 0.3$.

8. Get the observed values, y_{t+1} (and x_{t+1} for PLS), check them against spikes, center and scale them using the normalization parameters from the previous step (time = t), and calculate the actual scores, $t_{a,t+1}$, and the resulting residuals, $e = y_{t+1} - t_{a,t+1}\mathbf{P}_t'$.
9. Update the centering parameters by means of the residuals e.
10. Update the EWM-PCA or PLS model by iterating the algorithm till convergence.
11. Update the memory matrices $\mathbf{Q}_a$ ($\mathbf{P}_{\text{mem},a}$ or $\mathbf{W}_{\text{mem},a}$ for PLS), and, *if warranted*, also the data memory **Y**, and long term **P** or **W** matrices, $\mathbf{Q}_a^*$, and the spanning data memory matrix, $\mathbf{Y}^*$.

7.1. Difference between EWM-PCA and EWM-PLS

In the PLS situation, the process data have been divided in two (or more) blocks; **X** referring to inputs, and **Y** referring to outputs, and performance and quality measures on the products. One may here wish to monitor and forecast the process (**X**), the results (**Y**), or both. The model updating algorithm becomes a little (but not much) more complicated by the inclusion of the **Y** block. The data memory will now also have a **Y** block. Forecasts of **Y** are made directly from the forecasted **X** scores (t) as shown above in step 7, and the **X** data in the same way as with EWM-PCA. The inclusion of a **Y** block substantially stabilizes the model and decreases (but does not eliminate) the needs of special constraints on **P**.

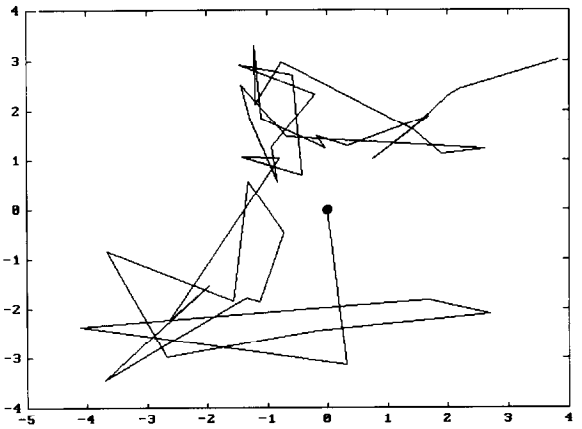


Fig. 6. The score plot (t_1 horizontally and t_2 vertically) of the paper machine data. The black dot indicates the starting point. $\lambda = 0.3$.

7.2. Useful displays

To follow the process, the display of the first two scores (t_1 and t_2) against each other or separately versus time provides a good picture of how the process evolves. The distance to the model — i.e., the standard deviation of the **Y** residuals, and **X** residuals for PLS — can be included as a separate plot, or included in the score plot(s) as color.

8. The paper machine illustration

As a first step, the first 15 of the 49 observations (objects, time points) were subjected to PCA after centering and unit variance scaling (autoscaling). Two significant components were obtained, explaining 51% (32 and 19) of the **Y** variance. Thereafter three EWM-PCAs were run with $\lambda_1 = [0.3, 0.25, \text{and } 0.2]$, λ_2 and $\lambda_3 = 0$, and no long term reference matrices ($\mathbf{Y}^*$ or $\mathbf{Q}^*$). Thus, only the **P** memory matrices $\mathbf{Q}_a$ were used to stabilize the model. Two components ($A = 2$) were used throughout, and the initial $\mathbf{Y}_0$ had eight objects (rows).

Fig. 5 shows the resulting loadings p_1 and p_2 for the 42 resulting models. The strongest coefficients (nos. 2, 3, 14–16 in p_1, and nos. 6, 9, 10, 17 in p_2) are seen to be fairly stable, while the smaller coefficients show a substantial variation. The score plot of t_1 and t_2 (Fig. 6) shows a clear jump in t_2 around $i = 22$ to 23. The DMod plot is uninteresting and is not shown.

The EWMA in the second score (t_2) is plotted together with the unsmoothed t_2 in Fig. 7. The jump between 22 and 23 is seen together with the one-ahead forecasts. The sum of squares of the prediction errors (PRESS) is 67.9 compared to the total SS of 171.2, showing that the EWMA of t_2 has a substantial predictive capability in this case. The EWMA of the first score, t_1, has only

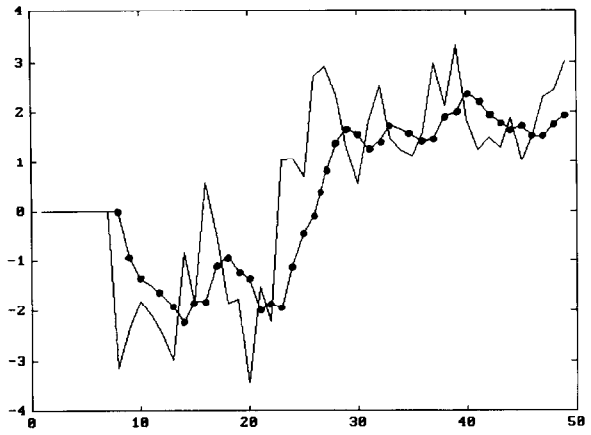


Fig. 7. A plot of the second score vector (t_2) of the EWM-PCA of the paper machine data together with the one-ahead EWMA forecast (dotted line). $\lambda = 0.3$.

little predictive capability (see Fig. 8); PRESS = 104.4 versus SS of 136.9.

The interpretation of these results together with the loadings, **P** (not shown), show that the second dimension is related to porosity and smoothness of the paper and colour, while dimension one models strength, thickness, and some other related properties. Thus, going from right to left in Fig. 6 corresponds to having a stronger and thicker paper, while going from down to up corresponds to a smoother paper. Values of t_1 larger than around 2 seem dangerous, and larger than 3 are alarming (the paper broke after observation 49).

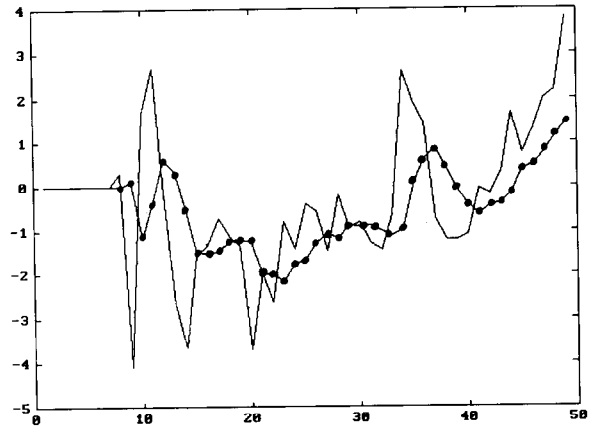


Fig. 8. A plot of the first score vector (t_1) of the EWM-PCA of the paper machine data together with the one-ahead EWMA forecast (dotted line). $\lambda = 0.3$.

9. Summary

The proposed EWM-PCA and EWM-PLS modelling provides multivariate windows on a dynamic process, where the dominant features of the evolving process are displayed as scores plus a measure of how far the process data are from the model. If there is autocorrelation structure in the scores, one-ahead forecasts of the process scores (t_a) and process variables (y or x) are useful for diagnosis and sometimes even control. The latter demands the deliberate multivariate disturbance of the process according to some statistical design as discussed by Kettaneh-Wold et al. in another article in this volume [22].

10. Notation

- **X** Matrix of process variables (inputs and others used to predict **Y**)
- i, j Indices of objects, rows in **X** and **Y** ($i, j = 1,2,\ldots,N$)
- N Number of objects, observations, samples, process time points (rows in **X** and **Y**)
- k Index of **X** and **Y** variables ($k = 1, 2,\ldots, K$)
- K Number of **X** or **Y** variables (columns in **X** or **Y**)
- **Y** Matrix of result variables in PLS (responses, outputs, product properties)
- * Used to denote long term memory matrices and corresponding parameters
- m Index of response variables ($m = 1, 2,\ldots, M$)
- M Number of PLS **Y** variables (columns in **Y** in PLS)
- v_i Weight of observation i
- a Index for components ($a = 1, 2,\ldots, A$)
- A Number of components
- **W** Matrix of PLS weights, dimension $K \times A$
- w_a Column of **W**, the **X** weights of component a
- **P** Matrix of loadings, dimension $K \times A$
- **Q** Memory matrix of loadings (or PLS weights)
- **C** Matrix of PLS **Y** weights, dimension $M \times A$

$\mathbf{c}_a$ Column of **C**, the **Y** weights of component a
T Matrix of scores, dimension $N \times A$
$\mathbf{t}_a$ Column of **T**, the scores of component a
U Matrix of other scores, dimension $L \times A$
$\mathbf{u}_a$ Column of **U**, other scores of component a
$\mathbf{E}_a$ **X** or **Y** residuals after ath component, dimension $N \times K$
$\mathbf{F}_a$ PLS **Y** residuals after ath component, dimension $N \times M$

11. Appendix

It is assumed that appropriate values of the parameters λ_1 to λ_3 are available, as well as starting values of the centering and scaling constants, EWMA_k and ϕ_k, for the kth variable.

11.1. The EWM-PCA algorithm

1. Select appropriate parameter values (λ_j, γ, etc.)
2. We start with an initial matrix, $\mathbf{Y}_0$, size $N_0 \times K$. We set $\mathbf{Y} = \mathbf{Y}_0$. The memory matrices, $\mathbf{Y}^*$ and **Q**, are initialized as empty.
3. Weights v_i and v_i^* are calculated as Eq. (1) with parameters λ and γ, respectively. The weighted average for each variable (k) is calculated from **Y**:

$$\mathrm{EWMA}_k = \sum_i v_i^* y_{ik} / \sum_i v_i^*$$

The scaling weights (ϕ_k) are calculated from both **Y** and $\mathbf{Y}^*$ (note that $\mathbf{Y}^*$ is centered):

$$s_k^2 = \frac{\beta \sum_i v_i (y_{ik} - \mathrm{EWMA}_k)^2 + (1-\beta) \sum_j v_j^* y_{jk}^{*2}}{\beta \sum_i v_i + (1-\beta) \sum_j v_j^*}$$

$$\phi_k = 1/s_k$$

Constant variables are left as zero with zero weights. Important variables may be scaled up or down by multiplying the above scaling weight by a suitable modifier between, say, 0.3 and 3. The parameter β that determines the relative influence of present and reference data can be anywhere between 0.1 and 0.9 depending on the stability of the process.

4. Center and scale **Y** with the centering parameters EWMA_k and the scaling parameters ϕ_k.

$$y_{ik}(\text{normalized}) = [\, y_{ik}(\text{raw}) - \mathrm{EWMA}_k] * \phi_k$$

5. Here begins the central part of the EWM-PCA algorithm; the determination of the weighted PCA model. The additional steps due to cross-validation are not explicitly shown; they basically comprise the rerunning of the algorithm below several times (typically 7 or 9) with different parts of the data deleted, and afterwards predicting the deleted data from the model as described by Wold [12] and Eastment and Krzanowski [13]. The model dimension, A, with the smallest prediction error (PRESS) is selected, with preference for the smaller A if PRESS is almost the same for several model dimensions.

(i) Set the dimension index, a, to one.

(ii) As a starting $\mathbf{p}_a$ and $\mathbf{q}_a$ (the loading vector), use the one from the previous time point. For the very first time, use the last row of $\mathbf{Y}_0$, normalized to length 1.

(iii) Calculate the scores, t_{ia}. To compensate for missing data, use the dummy variables d_{ik}, which are zero if element y_{ik} is missing, otherwise one.

$$t_{ia} = \frac{\sum_k d_{ik} y_{ik} p_{ka}}{\sum_k d_{ik} p_{ka}^2}$$

If there is a reference data matrix, $\mathbf{Y}^*$, calculate the corresponding scores, t_{ia}^* for this matrix using d_{jk} and y_{jk}^* instead of d_{ik} and y_{ik} in the equation above.

(iv) Calculate the loadings, p_{ka}, using the same d_{ik} for missing data compensation:

$$p_{ka} = \frac{\sum_i d_{ik} y_{ik} t_{ia}}{\sum_i d_{ik} t_{ia}^2}$$

Normalize $\mathbf{p}_a$ to length one; $\mathbf{p}_a = \mathbf{p}_a / \| \mathbf{p}_a \|$.

If there is a reference data matrix, $\mathbf{Y}^*$, calculate the corresponding loadings, p_{ka}^*

for this matrix using d_{jk}, y^*_{jk}, and t^*_{ia}, instead of d_{ik}, y_{ik}, and t_{ia} in the equation above.

Form $\boldsymbol{p}_a$ as the weighted combination of the two calculated $\boldsymbol{p}_a$ values.

$$\boldsymbol{p}_a = \beta \boldsymbol{p}_a + (1-\beta)\boldsymbol{p}^*_a$$

Normalize this new $\boldsymbol{p}_a$ to length one.

(v) Check the convergence on $\| \boldsymbol{p}_{a,\text{new}} - \boldsymbol{p}_{a,\text{old}} \| / \| \boldsymbol{p}_{a,\text{new}} \|$, which should be smaller than 10^{-6} to indicate convergence. If convergence, continue to step ix, otherwise step vi.

(vi) If this is the very first time, return to step iii, else step vii.

(vii) Calculate the scores, $\boldsymbol{u}_a$ and $\boldsymbol{u}^*_a$ for the loading memory and reference matrices, **P**, and **P***, respectively.

$$u_{ia} = \frac{\sum_k p_{\text{mem},a,ik}\, p_{ka}}{\sum_k p^2_{ka}}$$

$$u^*_{ja} = \frac{\sum_k p_{\text{ref},a,jk}\, p_{ka}}{\sum_k p^2_{ka}}$$

(xiii) Calculate 'loading loadings' of the two loading memory and reference matrices:

$$q_{ka} = \frac{\sum_i p_{\text{mem},a,ik}\, u_{ia}}{\sum_i u^2_{ia}}$$

$$q^*_{ka} = \sum_j p_{\text{ref},a,jk} u^*_{ja} / \sum_j (u^*_{ja})^2$$

Form $\boldsymbol{q}_a$ as the weighted combination of the two calculated $\boldsymbol{q}_a$ values.

$$\boldsymbol{q}_a = \beta \boldsymbol{q}_a + (1-\beta)\boldsymbol{q}^*_a$$

Normalize this new $\boldsymbol{q}_a$ to length one. Use the weighted combination (weight α) of this vector and $\boldsymbol{p}_a$ (weight $1-\alpha$) as the new $\boldsymbol{p}_a$, and return with it to step iii.

(ix) After convergence, calculate the final scores, $\boldsymbol{t}_a$, and $\boldsymbol{t}^*_a$ for the two data blocks (**Y** and **Y***), and from these, temporary loadings that are used only to form the residuals to constitute the data in the next model dimension calculations. This is necessary to preserve orthogonality of the scores, and is analogous to the orthogonalization step $\boldsymbol{w} \rightarrow \boldsymbol{t} \rightarrow \boldsymbol{p}$ in PLS regression.

Thereafter, form the residuals, $\mathbf{Y} - \boldsymbol{t}_a \boldsymbol{p}'_a$, add one to the model dimension ($a = a + 1$), and continue with the next dimension, using the residuals of **Y** and **Y*** as the data matrices **Y** and **Y*** in this next dimension.

(x) The algorithm is terminated when the model dimension, a, equals the desired number of 'significant' model dimensions, A, as determined by cross-validation, or, perhaps, by experience.

11.2. The EWM-PLS algorithm

The only difference between the PCA and the PLS algorithm is that the latter includes both **X** and **Y** blocks for the data and reference data. Substituting **X** for **Y**, and **X*** for **Y***, PLS weights for loadings, and $\boldsymbol{w}$ for $\boldsymbol{p}$, in the algorithm above, some additional sub-steps are added in step iii. After calculating the scores, $\boldsymbol{t}_a$ and $\boldsymbol{t}_a{}^*$, these are used to calculate **Y** weights, $\boldsymbol{c}_a$ and $\boldsymbol{c}_a{}^*$, respectively, which, in turn, give **Y** scores, here denoted by $\boldsymbol{r}_a$ instead of the standard $\boldsymbol{u}_a$.

(iiia) **Y** and **Y*** weights

$$c_{ma} = \frac{\sum_i d_{im} y_{im} t_{ia}}{\sum_i d_{im} t^2_{ia}}$$

and analogously for c^*_{ma}.

(iiib) Scores, $\boldsymbol{r}_a$ and $\boldsymbol{r}^*_a$

$$r_{ia} = \frac{\sum_m d_{im} y_{im} c_{ma}}{\sum_m d_{im} c^2_{ma}}$$

and analogously for r^*_{ja}.

These scores, $\boldsymbol{r}$ and $\boldsymbol{r}^*$, are then used instead of $\boldsymbol{t}$ and $\boldsymbol{t}^*$, respectively, to calculate the PLS weights in step iv.

Finally, after convergence, the residuals ($\mathbf{F}_a$) of each **Y** block are formed by subtracting the pertinent t vector times the pertinent c vector. These residuals are then used as **Y** or **Y*** in the next dimension.

11.3. Updating the memory and reference matrices

After convergence of the above algorithm, the resulting scores (only the t^2 values and the u^2 values) are compared with the maximum and minimum values of the corresponding scores for the reference data and reference loading matrices. Thus, when the reference matrices are initially empty, the data vectors corresponding to the largest and smallest t values for each model dimension are saved in the data reference matrix, **Y***. These extreme scores are saved for later comparisons. In following updates, a score value that is below the minimum or above the maximum previous scores of that dimension makes the corresponding data vector be included in the reference matrix and the new score value saved at the place of the old one. Two variants of **Y*** are seen; one where the old data vector is deleted from **Y***, and no exponential weighting of **Y*** is done. The second, presently recommended variant, is to just extend **Y*** with the new 'spanning' vector, and use a slowly decreasing exponential weighting of **Y***.

Acknowledgements

Support from NFR (Swedish Natural Science Research Council) is gratefully acknowledged. I am indebted to John MacGregor for giving me access to the paper machine data.

References

[1] B.M. Wise and N.L. Ricker, Feedback strategies in multiple sensor systems, Presented at AIChE Conference, Washington, DC, November 1988.

[2] J.V. Kresta, J.F. MacGregor and T.E. Marlin, Multivariate statistical monitoring of process operating performance, *Canadian Journal of Chemical Engineering*, 69 (1991) 35–47.

[3] M.B. Priestley, T.S. Rao and H. Tong, Applications of principal components analysis and factor analysis in the identification of multivariable systems, *IEEE Transactions on Automatic Control*, 19 (1974) 730–734.

[4] B. Skagerberg, J.F. Macgregor and C. Kiparissides, Multivariate data analysis applied to low-density polyethylene reactors, *Chemometrics and Intelligent Laboratory Systems*, 14 (1992) 341–356.

[5] S. Wold, J. Jonsson, M. Sjöström, M. Sandberg and S. Rännar, DNA and peptide sequences and chemical processes multivariately modelled by PCA and PLS, *Analytica Chimica Acta*, 277 (1993) 239–253.

[6] J.S. Hunter, The exponentially weighted moving average, *Journal of Quality Technology*, 18 (1986) 203–209.

[7] J.M. Lucas and M.S. Saccucci, Exponentially weighted moving average control schemes: Properties and enhancements, *Technometrics*, 32 (1990) 1–12.

[8] J.F. MacGregor and T.J. Harris, Discussion of Lucas and Saccucci, *Technometrics*, 32 (1990) 23–26.

[9] J.E. Jackson, *A User's Guide to Principal Components*, Wiley, New York, 1991.

[10] I.T. Jolliffe, *Principal Component Analysis*, Springer Verlag, Berlin, 1986.

[11] S. Wold, K. Esbensen and P. Geladi, Principal Component Analysis, *Chemometrics and Intelligent Laboratory Systems*, 2 (1987) 37–52.

[12] S. Wold, Cross validatory estimation of the number of components in factor and principal components models, *Technometrics*, 20 (1978) 397–406.

[13] H. Eastment and W. Krzanowski, Crossvalidatory choice of the number of components from a principal component analysis, *Technometrics*, 24 (1982) 73–77.

[14] R. Fisher and W. MacKenzie, Studies in Crop Variation. II. The manurial response of different potato varieties, *Journal of Agricultural Science*, 13 (1923) 311–320.

[15] H. Wold, Nonlinear estimation by iterative least squares procedures, in F. David (Editor), *Research Papers in Statistics*, Wiley, New York, 1966, pp. 411–444.

[16] R.B. Crosier, Multivariate generalizations of cumulative sum quality-control schemes, *Technometrics*, 30 (1988) 291–303.

[17] D.M. Hawkins, Multivariate quality control charts based on regression adjusted variables, *Technometrics*, 33 (1991) 61–75.

[18] C.A. Lowry, W.H. Woodall, C.W. Champ and S.E. Rigdon, A multivariate exponentially weighted moving average control chart, *Technometrics*, 34 (1992) 46–53.

[19] S. Wold, S. Hellberg, T. Lundstedt, M. Sjöström and H. Wold, PLS model building: Theory and application. PLS modeling with latent variables in two or more dimensions, in H. Wold and W. Melssner (Editors), PLS Symposium, Frankfurt am Main, Sept. 23–25, 1987.

[20] K.J. Åström and B. Wittenmark, *Computer Controlled Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1984.

[21] S. Wold, H. Martens and H. Wold, The multiblock PLS algorithm, in S. Wold (Editor), *MULDAST NEWS, Report from the MULDAST Symposium in Umea, June 4–8, 1984*, Institute of Chemistry, Umea University, 1984 (see also: L.E. Wangen and B.R. Kowalski, A multiblock PLS algorithm for investigating complex chemical systems, *Journal of Chemometrics*, 3 (1988) 3–20).

[22] N. Kettaneh-Wold, J.F. MacGregor and S. Wold, Multivariate design of process experiments, *Chemometrics and Intelligent Laboratory Systems*, 23 (1994) 39–50.