Available online at www.sciencedirect.com
SCIENCE DIRECT®
Chemometrics and Intelligent Laboratory Systems 68 (2003) 75–81
Chemometrics and intelligent laboratory systems
www.elsevier.com/locate/chemolab

Industrial fermentation end-product modelling with multilinear PLS

J.A. Lopes*, J.C. Menezes

Centre for Chemical and Biological Engineering, Technical University of Lisbon, Avenue Rovisco Pais, P-1049-001, Lisbon, Portugal

Abstract

In this paper, a trilinear version of the partial least squares (PLS) algorithm was used to model the performance of an industrial fed-batch fermentation process. Trilinear data obtained from process operation were used to derive a model for the end-process active product ingredient (API) concentration prediction. Obtained mutilinear PLS models were compared with the correspondent bilinear models. A genetic algorithm was used to select appropriate calibration sets (to reduce the influence of nominal batches). A validation coefficient of determination (Q^2_Y) of 91.4% was obtained for the multilinear PLS model after batch selection (prediction intervals were estimated using bootstrapping). Examination of the multilinear PLS model weights led to the delimitation of a small time region (from 50 to 75 processing hours) almost exclusively responsible for the fermentation performance.

Keywords: Pharmaceutical production; Multivariate modelling; Multilinear PLS; Parafac; Genetic algorithms

1. Introduction

Dynamic models for industrial fermentation processes are difficult to identify because of a wide variety of reasons: microorganisms complex dynamics, variable and ill-defined raw materials, and dependence on previous process stages (strain selection and pre-culture production) [10]. Common problems may include varying inocula quality and sensor or pump failures. Several disturbances can irreversibly influence the microorganism metabolism and lead to low product concentrations and sub-optimal batches—depending on their intensity and duration. The losses associated with sub-optimal operation increase with process scale, thus early detection of deviations or faults is crucial in large-scale industrial bioreactors. Frequently in industry, the performance of fermentation is directly related with the amount of product obtained at the end of process.

The nature of the fermentation data is trilinear (batches × variables × time). Multivariate modelling tools such as partial least squares (PLS) [15] are generally used with two-way data [4]. The common approach is to unfold the data, that is, transform a three-way array into a two-way array preserving one mode, in this case, the batch mode. However, an alternative model proposed by Bro [1] can be obtained using a trilinear (or multilinear) version of the PLS algorithm. Bro points out some advantages of the multilinear PLS algorithm: more parsimonious and less affected by noise in the original variables.

* Corresponding author. Tel.: +351-218-417-347; fax: +351-218-419-062.
E-mail addresses: joao.lopes@ist.utl.pt (J.A. Lopes), cardoso.menezes@ist.utl.pt (J.C. Menezes).

0169-7439/03/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0169-7439(03)00089-3

2. Theory

PLS is a class of regression models based on the calculation of latent variables or factors [4]. In PLS, these variables are calculated to maximize the covariance between the scores of an independent block (**X**) and the scores of a dependent block (**Y**). In this paper, the prediction of a univariate vector (***y***) from a three-dimensional matrix ($\underline{\mathbf{X}}$) will be considered. The vector ***y*** ($I \times 1$) and the tensor $\underline{\mathbf{X}}$ ($I \times J \times K$) share the batch dimension I. The vector ***y*** contains the end-process active product ingredient (API) concentration for I batches and $\underline{\mathbf{X}}$ contains the values of different variables measured at different time instants for each batch. A matricization operation unfold(**X**,a,b) with three arguments is defined [6]. This operation reshapes the three-way matrix $\underline{\mathbf{X}}$ into a two-way matrix with a rows and b columns. Elements are extracted columnwise. For three-mode arrays, the first dimension is always preserved in the present work. In the following sections, a bilinear and trilinear algorithms for PLS are presented. The general multilinear PLS algorithm is called N-PLS. In the algorithms, the symbol $\otimes$ denotes the Kronecker product.

2.1. Bilinear PLS

The unfolding of the trilinear structure of $\underline{\mathbf{X}}$ ($I \times J \times K$) preserving the batch dimension yields a **X**($I \times JK$) array. Hence, a bilinear PLS (U-PLS) can be used to model ***y***.

$$\mathbf{X} = \boldsymbol{T}\mathbf{W}^T + \mathbf{E} \tag{1}$$

The weights matrix **W**, in Eq. (1), captures the structure of the variables and time modes. The orthogonal loadings U-PLS algorithm is presented here for comparison with the trilinear version which is intrinsically related to the orthogonal loading version of two-way PLS [2,8]. A regression vector of the form $\hat{\boldsymbol{y}} = \mathbf{X}b_{\text{U-PLS}}$ can be obtained. For the orthogonal loadings algorithm, $\hat{\boldsymbol{y}} = \boldsymbol{T}\mathrm{q} = \mathbf{X}\mathbf{W}q$, hence $b_{\text{U-PLS}} = \mathbf{W}q$ (see Table 1). A more interesting expression is given by Eq. (2) where only the weights and covariances are used [5].

$$b_{\text{U-PLS}} = \mathbf{W}(\mathbf{W}^T\mathbf{S}_{xx}\mathbf{W})^{-1}\mathbf{W}^T\mathbf{S}_{xy} \tag{2}$$

Table 1
Algorithms for U-PLS (bilinear) and N-PLS (trilinear)

let $\mathbf{X}$ = unfold($\underline{\mathbf{X}}$,I,JK) and $\boldsymbol{y_0} = \boldsymbol{y}$ for each factor (LV) do	
U-PLS [2,8]	N-PLS [1]
$\boldsymbol{w} = \mathbf{X}^T\boldsymbol{y}$	$\boldsymbol{Z}$ = unfold ($\mathbf{X}^T\mathbf{y}$,J,K) $\max_{\boldsymbol{w}^J,\boldsymbol{w}^K}$ $[(\boldsymbol{w}^J)^T\boldsymbol{Z}\boldsymbol{w}^K]$ where $\boldsymbol{w}^J$ and $\boldsymbol{w}^K$ are the first components of the singular value decomposition of $\boldsymbol{Z}$ ($\boldsymbol{Z} = \mathbf{W}^J\mathbf{S}\mathbf{W}^K$)
$\boldsymbol{w} = \boldsymbol{w}/\|\boldsymbol{w}\|$	$\boldsymbol{w} = \boldsymbol{w}^K \otimes \boldsymbol{w}^J$[a]
$\boldsymbol{t} = \mathbf{X}\boldsymbol{w}$	
$\boldsymbol{T} = [\boldsymbol{T}\|\boldsymbol{t}]$	
$q = (\boldsymbol{T}^T\boldsymbol{T})\boldsymbol{T}^T\boldsymbol{y}_0$	
	For each row i of $\mathbf{X}$ do
$\mathbf{X} = \mathbf{X} - \boldsymbol{t}\boldsymbol{w}^T$	$\mathbf{X}_i = \mathbf{X}_i - \text{unfold}(\boldsymbol{t}_i \cdot \boldsymbol{w}^J(\boldsymbol{w}^K)^T, 1, JK)$
$\boldsymbol{y} = \boldsymbol{y}_0 - \mathbf{T}q$	
$\mathbf{W} = [\mathbf{W}\|\boldsymbol{w}]$	
return to first step to include more factors	

[a] $\boldsymbol{w}^J$ and $\boldsymbol{w}^K$ are the first column of $\mathbf{W}^J$ and $\mathbf{W}^K$, respectively.

2.2. Trilinear PLS

A more suitable model for these data is Bro's N-PLS algorithm [1].

$$\mathbf{X} = \boldsymbol{T}(\mathbf{W}^K \otimes \mathbf{W}^J)^T + \mathbf{E} \tag{3}$$

The weights in Eq. (3) ($\mathbf{W}^J$ and $\mathbf{W}^K$) are related with the variables and time modes, respectively. These are obtained as the first left and right singular vectors of $\mathbf{X}^T\boldsymbol{y}$. In contrast with U-PLS, these weights are non-orthogonal $(w_p^J)^T w_r^J \neq 0$ and $(w_p^K)^T w_r^K \neq 0$ for each $p \neq r$. The analysis of the weights $\mathbf{W}^J$ and $\mathbf{W}^K$ provides an indication of the importance of each variable and time region for the model.

For the trilinear version of the PLS algorithm, it is also possible to compute a regression vector. Because the weights w_i are non-orthogonal, the regression vector cannot be given as $\mathbf{W}q$ where $\mathbf{W} = [w_1|w_2|\cdots|w_r]$. Nevertheless, there is a way to compute $b_{\text{N-PLS}}$ (Eq. (4) refers to a model with r factors) [14].

$$b_{\text{N-PLS}} = [\boldsymbol{w}_1 \,|\, (I - \boldsymbol{w}_1\boldsymbol{w}_1^T)\boldsymbol{w}_2 \,|\cdots|\, (I - \boldsymbol{w}_1\boldsymbol{w}_1^T) \times (I - \boldsymbol{w}_2\boldsymbol{w}_2^T)\cdots(I - \boldsymbol{w}_{r-1}\boldsymbol{w}_{r-1}^T)\boldsymbol{w}_r]q \tag{4}$$

2.3. Model predictions and cross validation

Before the projection of new data ($\underline{\mathbf{X}}^{\text{new}}$) into the model, it is required that the centring and scaling coefficients (obtained from calibration data) are applied. Predictions for each new sample (for the U-PLS and N-PLS models) can be produced using Eq. (5).

$$\hat{\boldsymbol{y}} = (\underline{\mathbf{X}}_i^{\text{new}}, I, JK) b_{\text{U-PLS/N-PLS}} \tag{5}$$

Selection of the appropriate number of factors is achieved by cross validating the models. The leave-one-block-out strategy consists of dividing the entire data set on n blocks. The model is calibrated with each $n-1$ set of blocks and tested on the remaining block [9]. Two measures of goodness of fit are used: the root mean squares (Eq. (6)) and the amount of variance predicted (Eq. (7)).

$$\text{RMS}_{\text{cv}} = \left[\frac{\text{trace}[(\boldsymbol{y}-\hat{\boldsymbol{y}})^T(\boldsymbol{y}-\hat{\boldsymbol{y}})]}{n}\right]^{0.5} \tag{6}$$

$$Q_Y^2 = 1 - \frac{\text{trace}[(\boldsymbol{y}-\hat{\boldsymbol{y}})^T(\boldsymbol{y}-\hat{\boldsymbol{y}})]}{\text{trace}(\boldsymbol{y}^T\boldsymbol{y})} \tag{7}$$

2.4. Prediction intervals for PLS estimates

There are several methods to estimate prediction intervals for PLS estimates [13]. PLS can be viewed as combination of a projection in latent structures followed by an ordinary regression between components. The problem is that the number of degrees of freedom is unknown. A bootstrap-based strategy can be used to replace the t-student parameter with a constant that approximates the true distribution of the residuals [9].

In the approach described by Denham [3], a constant c_α is estimated from n different PLS models, where n is the number of batch samples. Each model is obtained removing sample i. The constant c_α is obtained for a confidence level of $100(1-\alpha)\%$. A prediction interval for an estimate $\hat{\boldsymbol{y}}_{\text{i}}$ is given by Eq. (8), where s_{e} is the residual sum of squares divided by $n-r-1$ (r is the number of factors).

$$\text{PI}_\alpha(\boldsymbol{y}_i) : \hat{\boldsymbol{y}}_i \pm c_\alpha s_{\text{e}} \left[(n+1)/n + (x_i-\bar{x})S_{\mathbf{X}}^{-1}(x_i-\bar{x})^T\right]^{0.5} \tag{8}$$

2.5. Batch selection

A genetic algorithm was used to select a better set of batches to include in model calibration [11] instead of using every available batch. If a large number of normal operating condition (NOC) batches are present, the model may not capture relevant information contained in the less numerous batches operated on non-nominal conditions (imposed steps on temperature, feed rates, and aeration over time). To select the best calibration set, different models are tested. A loss function $F = Q_Y^2(\eta + (1-\eta)m/n)$ was maximized. The η parameter balances the importance of the goodness of fit and of the number of batches discarded. m is the number of batches included and n the total number of batches. The genetic algorithm will find a set of batches that maximizes F, based on a leave-one-block-out cross validation procedure. We are aware that this method cannot avoid a certain model overfit since the fitting error is included in the loss function. With this strategy, we offer an alternative way to the traditional methods of model validation (e.g., random splitting cross validation).

3. Experimental

The performance of an industrial fed-batch fermentation process was monitored as the final concentration of the produced API, which is an antibiotic-like β-lactam [12]. The objective is to be able to produce estimates of the final API concentration using the data monitored during the fermentation. Experiments are divided in two groups: A and B. Group A contains 80 batches operated on pilot-scale bioreactors while group B contains 22 batches operated on semi-industrial bioreactors (10 times greater in volume). Previous results suggested a notable difference between the batches according to the scale [7] even after geometric ratios and other engineering dimensional corrections were introduced in all extensive measured variables. The total duration of each batch is 132 h. A total of 10 variables was monitored during the fermentation time. Some data pre-treatment was applied to eliminate outliers and reduce noise. A sampling frequency of 4 h was found to be adequate. For each batch, the API concentration was monitored and the value at the end of the batch was stored to be used as the quality

Table 2
Cross-validation errors obtained with U-PLS and N-PLS models for the final API concentration

RMS_{cv}	U-PLS (three factors)	N-PLS (four factors)
Scale A	0.075	0.074
Scale B	0.052	0.052

Table 3
Amount of variance predicted of the final API concentration by the U-PLS and N-PLS models using all available batches

Q_Y^2 (%)	U-PLS (three factors)	N-PLS (four factors)
Scale A	57.0	57.2
Scale B	64.3	64.3

variable. Group A is represented by an independent array $\mathbf{X}_A$ with dimensions (80 × 10 × 34) and a quality variable vector $\boldsymbol{y}_A$ with dimensions (80 × 1). For group B, $\mathbf{X}_B$(22 × 10 × 34) and $\boldsymbol{y}_B$(22 × 1). The data were mean centred and slab-scaled across the variables dimension (slab scaling consists of dividing by the correspondent standard deviation every value in a data slice [6].

4. Results and discussion

The analysis was performed in four steps:

(1) selection of the appropriate number of components to include in U-PLS and N-PLS;
(2) compare the result for U-PLS and N-PLS models;
(3) build-up of a better calibration set to optimize N-PLS models (using a genetic algorithm);
(4) analysis of N-PLS model weights.

Initially, the best number of factors was selected by a leave-one-block-out cross validation strategy for both data sets and considering all batches. This was performed with U-PLS and N-PLS models. A different optimal number of factors was determined for each model. While for U-PLS, the best number of components is 3; for N-PLS, the best number of components was found to be 4. However, the RMS_{cv} is approximately the same for U-PLS with three factors and N-PLS with four factors (see Table 2). Fig. 1 shows the results obtained for each scale with N-PLS models. The difference in the optimum number of components might be related with the way the weights are computed in the N-PLS algorithm.

The amount of variance predicted (Q_Y^2) of the U-PLS (three factors) and N-PLS (four factors) models is presented in Table 3 (leave-one-out cross validation strategy). In terms of prediction, both models seem to be equivalent. A better amount of variance predicted was obtained for scale B data set. These models were obtained using all batches.

As stated before, an appropriate selection of batches might improve the obtained models. A genetic algorithm was used to select a better set of batches to calibrate N-PLS models with four factors (from now on, only the N-PLS model was used since the prediction ability is very similar to the U-PLS model). The maximization of the cost function (see Section 2.5) resulted in the selection of 47 batches (for scale A) and 15 batches (for scale B). In Fig. 2, the selected

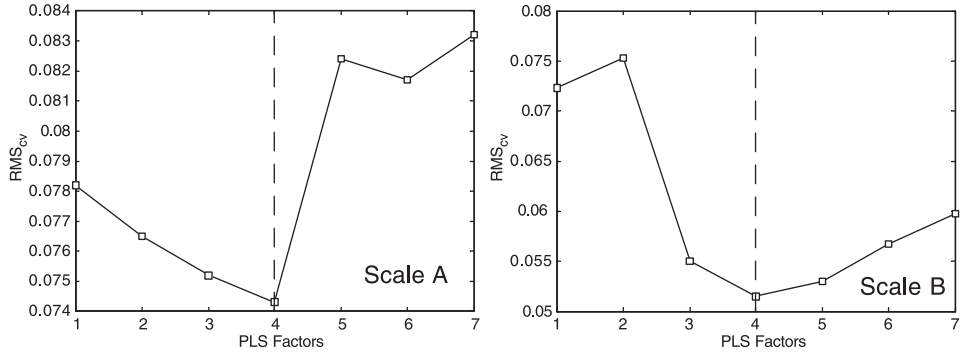

Fig. 1. Cross-validation charts for N-PLS models (groups A and B). The best number of factors is 4 for both data sets.

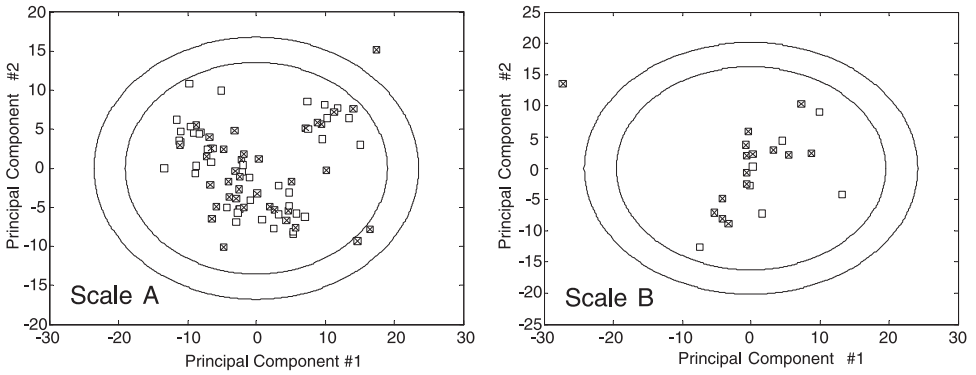

Fig. 2. First versus second principal components for groups A and B. Batches marked as (⊠) were selected by the genetic algorithm to be included in the calibration set (the captured variance in the two principal components is 44.1% for data set A and 47.6% for data set B).

batches for both scales are depicted (principal components are used to map the batches). The algorithm selected a small fraction of the available batches presumably due to the presence of a large number of NOC batches. It is clear from Fig. 2 that the algorithm selected both nominal and non-nominal batches. A large number of NOC batches are left out from the calibration set for both data sets. Fig. 3 shows actual versus predicted values for the final API concentration for the batches selected by the algorithm. Predictions are obtained by leave-one-out cross validation strategy (for data set A, $Q_Y^2 = 86.6\%$ and for data set B, $Q_Y^2 = 85.8\%$). These values are significantly higher than those presented in Table 3 (where all batches were used).

Considering a model calibrated with the selected batches and validated with the rest, a determination coefficient of 82.2% was obtained for data set A and 81.8% for data set B.

The N-PLS model weights $\mathbf{W}^K$ can be used to determine time points that are more important for the regression. For both data sets, it was found that the weights are near zero after 80 h. Fig. 4 refers to the model for scale B batches. It is clear that for the first and second factors, the weights are more important in the range from 0 to 80 h. This clearly shows that the fermentation later instants do not have information to explain the API concentration variability as expected from a batch bioprocess such as the one considered here. Therefore, a new set of models was built using a sliding time window instead of the entire fermentation. Several window lengths were tested. A 12-h window length was found optimal for modelling both data sets (using the selected batches for calibration and the rest

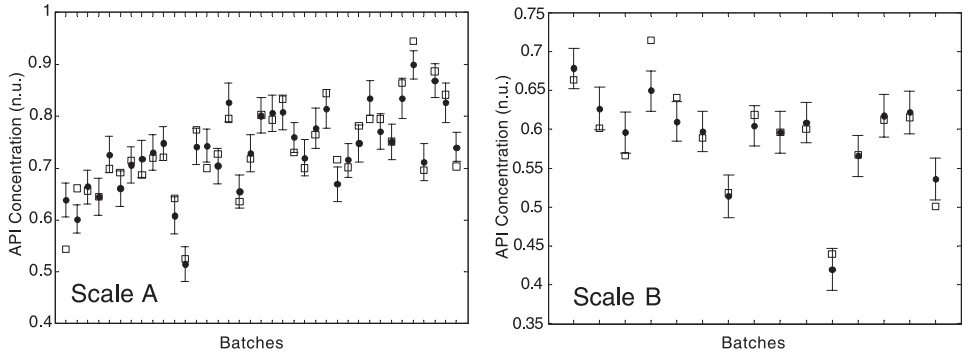

Fig. 3. Final API concentration predictions for batches of data sets A and B (normalized data). Each prediction was obtained from an N-PLS model with four factors, built with the remaining batches. A bootstrap strategy was used to estimate prediction intervals.

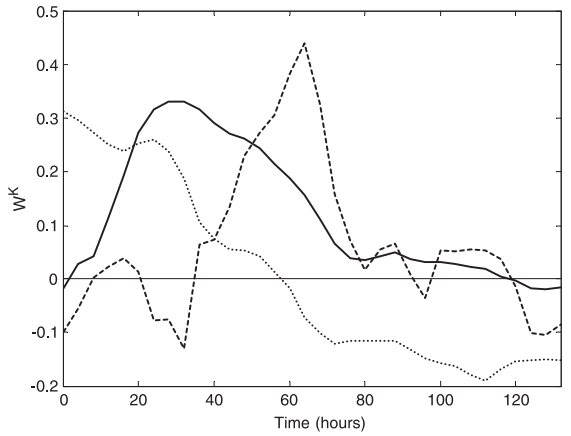

Fig. 4. Weights for the time dimension ($\mathbf{W}^K$). The model was calibrated with selected scale B batches (—— factor 1, — — factor 2, · · · · · factor 3).

for validation). Each model uses only three time points to predict the final API concentration.

The results obtained for the N-PLS models using a sliding window (for data sets A and B) are depicted on Table 4. For the smaller scale batches, the best model uses points around hour 44 ($Q_Y^2=0.82$), while for the larger scale batches, the best region is located at hour 60 ($Q_Y^2=0.91$). It seems that for the smaller scale tanks, the phenomena that determine the performance of the fermentation happen slightly before than for the larger scale tanks. Note that for the smaller scale batches, if region around hour 60 was used, the Q_Y^2 was only 0.35. Similarly, using the region around hour 44 for the larger scale batches, the obtained Q_Y^2 was 0.32. The observed times in each scale correspond to the beginning of the API production phase. It is thus very important to improve the operation of the process during the first process phases (exponentially growth and transition phases)—e.g., investing in process optimization and process consistent operation in those process phases.

5. Conclusions

A bilinear PLS and multilinear PLS algorithms were compared, in terms of regression performance and parameter interpretability, using industrial fermentation data. Even if both algorithms produced the same results in terms of prediction, the multilinear model needed one more factor. This is probably due to the way the weights are computed.

With the implementation of a genetic algorithm, an appropriate calibration set could be extracted from the available batches (reducing the information redundancy). The same results were obtained with two independent data sets (different tank volumes).

The analysis of multilinear PLS weights was found to be very important to determine a region in the process that could explain the observed variability in the quality variable. A 16-h difference was found between the two scales. The optimal time region for making significant process corrections corresponds to the onset of API production phase. It is thus very important to improve the process monitoring in the observed regions to generate more precise models for the process. To improve the production (and decrease production variability), the process should be controlled more accurately during the first half of the fermentation.

Nomenclature

$b_{\text{N-PLS}}$	N-PLS regression coefficients
$b_{\text{U-PLS}}$	U-PLS regression coefficients
c_α	Statistical parameter determined by cross validation (prediction intervals)
$\mathbf{E}$	PLS residuals unfolded matrix

Table 4
Amount of variance predicted for the final API concentration

Q_Y^2	Process time (h)															
	12	20	28	36	44	52	60	68	76	84	92	100	108	116	124	132
Scale A	0.10	0.54	0.62	0.73	**0.82**	0.68	0.35	0.32	0.30	0.14	0.15	0.17	0.18	0.17	<0	–
Scale B	<0	<0	<0	0.22	0.32	0.73	**0.91**	0.66	0.43	0.11	<0	<0	<0	<0	<0	–

Each N-PLS model was calibrated with a data time window centered at a specified time point.
The time window width was set to 12 h ($t-6$,$t+6$).

F	Loss function (batch selection)
m	Batches included in batch selection procedure
n	Number of batch samples
$\mathrm{PI}_{\alpha}(y_i)$	Prediction interval for sample i
q	PLS loadings for the dependent block
Q_Y^2	Amount of predicted variance (cross validation)
r	Number of components in a PLS model
$\mathrm{RMS}_{\mathrm{cv}}$	Root mean squares (cross validation)
s_{e}	Sum squares error divided by $n-r-1$ (cross validation)
$\mathbf{S}_{xx}$	Covariance matrix for independent block (unfolded)
$\mathbf{S}_{xy}$	Covariance matrix between the dependent and independent blocks (unfolded)
$\boldsymbol{T}$	PLS scores
$\mathbf{W}$	U-PLS weights
$\mathbf{W}^J$	N-PLS weights for variables mode
$\mathbf{W}^K$	N-PLS weights for time mode
$\underline{\mathbf{X}}$	Independent data block
$\overline{\mathbf{X}}$	Unfolded independent data block
$\boldsymbol{y}$	Independent data vector
$\hat{\boldsymbol{y}}$	PLS prediction vector
η	Balance parameter (batch selection)

Acknowledgements

The authors would like to thank Companhia Industrial Produtora de Antibióticos SA (CIPAN) in Portugal for providing the data. The financial support from the Foundation for Science and Technology (PRAXIS XXI BD/18471/98) is gratefully acknowledged.

References

[1] R. Bro, Multiway calibration: multilinear PLS, J. Chemom. 10 (1996) 47–61.
[2] M. Denham, Implementing partial least squares, Stat. Comput. 5 (1995) 191–202.
[3] M. Denham, Prediction intervals in partial least squares, J. Chemom. 11 (1997) 39–52.
[4] P. Geladi, B. Kowalsky, Partial least squares regression: a tutorial, Anal. Chim. Acta 185 (1986) 1–17.
[5] I. Helland, On the structure of partial least squares regression, Communications in Statistics—Elements of Simulation and Computation 17 (1988) 581–607.
[6] H. Kiers, Towards a standardized notation and terminology in multiway analysis, J. Chemom. 14 (2000) 105–122.
[7] J. Lopes, J. Menezes, Faster development of fermentation processes. Early stages process diagnostics, AIChE Symp. Ser. 94 (320) (1998) 391–396.
[8] H. Martens, T. Naes, Multivariate Calibration, Wiley, Chichester, 1989.
[9] D. Massart, B. Vandeginste, S. Deming, Y. Michotte, L. Kaufman, Chemometrics: A Textbook, Elsevier, Amsterdam, 1988.
[10] J. Menezes, S. Alves, J. Lemos, S. Azevedo, Mathematical modelling of industrial pilot-plant penicillin-G fed-batch fermentations, J. Chem. Technol. Biotechnol. 64 (1994) 123–138.
[11] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, 3rd ed., Springer, Berlin, 1996.
[12] A. Neves, L. Vieira, J. Menezes, Effects of preculture variability on clavulanic acid fermentation, Biotechnol. Bioeng. 72 (6) (2001) 628–633.
[13] A. Phatak, P. Reilly, A. Pendilis, An approach to interval estimation in partial least squares regression, Anal. Chim. Acta 277 (1993) 495–501.
[14] A. Smilde, Comments on multilinear PLS, J. Chemom. 11 (1997) 367–377.
[15] H. Wold, Nonlinear estimation by iterative least squares procedures, in: F. David (Ed.), Research Papers in Statistics, Wiley, New York, 1966, pp. 411–444.