Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

PLS and its Application within Model Predictive Controllers

Shamekh A., Lennox B., Sandoz, D. and Marjanovic, O.

Control Systems Centre (CSC), School of Electrical and Electronic Engineering, The University of Manchester, UK

E-mail: Awad.Shamekh@postgrade.manchester.ac.uk

Abstract: Many system identification techniques have been proposed over the last few decades, including ordinary and recursive least squares. Recently, Partial Least Squares (PLS) has become a popular tool in the chemometric community and is beginning to be applied to solve complex industrial process control problems. These studies have tended to ignore the issue of bias with this form of model and it is this issue that is addressed in this article. The paper describes the development of an unbiased recursive PLS algorithm that is successfully applied to two simulated processes. *Copyright © 2008 IFAC*

Keywords: recursive identification, model predictive control, partial least squares, recursive least squares

1. INTRODUCTION

In a model predictive control (MPC) system, the accuracy of the model embedded within the controller plays a crucial role in the performance of the system. The vast majority of MPC applications involve the identification of a process model using data collected from the plant. This model tends to be linear in its parameters and for time-varying and non-linear systems, the model is often designed so that it can adapt to the changing dynamics of the process. The identification of the model parameters can be achieved using the ordinary least squares (OLS) algorithm. OLS is an unbiased technique, which assumes that a linear relationship describes the system variables. However, the presence of unbiased regression coefficients does not imply minimal variance of the prediction errors (Frank 1987).

The structure of the model that is used within the MPC system can take several forms. In industrial applications, finite impulse response (FIR), auto-regressive with exogeneous signals (ARX) or auto-regressive moving average with exogeneous signal (ARMAX) model structures are typically employed. To ensure that the models are accurate representations of the plant dynamics and are not biased, the identification algorithms tend to be applied using either the instrumental variable, output error (OE) or prediction error method (PEM) techniques. These techniques have been explored in detail by Astrom *et al.* (1971), Astrom (1980), Soderstrom *et al.* (1978, 1988), Ljung, (1987, 1995). In his work, Zhu (2001) concentrated on the dynamic properties of ARX models and their use within MPC systems. Zhu highlighted the importance of developing an unbiased model and described how OE methods provided a useful approach for identifying such models. In practical model predictive control applications, recursive least squares (RLS), utilising the OE method has been demonstrated to be a robust and accurate method for developing MPC applications (Sandoz, 2003).

As MPC becomes an established tool in industry, it has begun to be applied to more complex processes. In particular, the application of MPC to processes where there exist significant amounts of correlation between the input variables to the model and where there is limited data available has recently been explored (see for example Dayal and MacGregor, 1997(b)). For these applications, an alternative identification algorithm, namely partial least squares (PLS), has emerged as an attractive modelling method. PLS is used routinely in the field of chemometrics, while its use in process control applications is a recent development. In the context of process control, model bias and other control specific issues related to PLS have tended to be ignored. In this paper, a recursive unbiased PLS algorithm is developed within the general framework of RLS modelling. The proposed algorithm addresses bias in terms of the regression coefficients. The suitability of the proposed technique is illustrated through application to two simulated systems.

2. MPC AND CONTROL LAW SYNTHESIS

2.1 MPC Algorithm

MPC operates by identifying future control action that minimises a particular cost or objective function. Based on the work of Cutler and Ramaker *et al.* (1980), Clarke *et al.* (1987) introduced the following objective function:

$$J=\sum_{j=1}^{p}\alpha_j\left(y_r(k+j/k)-\hat{y}(k+j/k)\right)^2+\sum_{j=1}^{m}\beta_j\Delta u(k+j-1/k)^2 \quad (1)$$

Where J is the cost function to be minimized, k is the sampling instant, p and m are the prediction and control horizons, respectively, y_r and $\hat{y}$ are the reference (set-point) values and estimated future output values, respectively, α_j and β_j are the weighting parameters for the controlled and manipulated variables respectively. Finally, Δu is the change in manipulated variable (incremental control move) that is to be computed by the MPC algorithm.

The target of the objective function in (1) is to force the future output to track the reference trajectory over the specified prediction window (p), while taking into account the balance between error energy and incremental control energy. In the case of the unconstrained control law, the optimal solution for a multivariate system can be shown to be:

$$\Delta u = (G^{\mathrm{T}} A G + B)^{-1} G^{\mathrm{T}} A (y_r - y_u) \qquad (2)$$

Where Δu is the vector of manipulated variable moves, G is the step response matrix, A and B are the weighting matrices for the controlled and manipulated variables respectively, y_r is the reference vector for the controlled variables and y_u is the unforced response of the process.

The major advantage of MPC when compared to other advanced control techniques is its ability to systematically incorporate constraints on the manipulated and control variables. However, the incorporation of constraints results in the need to use numerical optimisation, namely quadratic programming, in order to solve the corresponding MPC control problem.

2.2 Model Structure

Clarke *et al.* (1987) derived the control law (2) using the CARIMA structure of the dynamic model (prediction model) in order to cope with non-stationary disturbance models. In industrial applications such disturbances may occur as a result of changes in material quality, or Brownian movement (thermal molecular motion in liquid environments). However, it is very difficult to accurately estimate the colouring polynomial $C(q^{-1})$ in practical applications (Camacho et *al.* 1999). Hence, in such situations the ARX model structure is preferred. The structure of this model is as follows:

$$y_t = -\sum_{j=1}^{n} a_j y_{t-j} + \sum_{j=1}^{n} b_j u_{t-j} + e_t \qquad (3)$$

which can be written compactly as

$$A(q^{-1}) y_t = B(q^{-1}) u_t + \varepsilon_t \qquad (4)$$

Where y_t and u_t are the process output and input respectively. A and B are polynomials in the backwards shift operator q as follows:

$$A(q^{-1}) = 1 + a_1 q^{-1} + ... + a_n q^{-n} \qquad (5)$$

$$B(q^{-1}) = b_1 q^{-1} + ... + b_n q^{-m} \qquad (6)$$

ε_t is the residual between the actual and estimated outputs.

3. IDENTIFICATION ALGORITHMS

3.1 Recursive Least Squares

In this work the RLS algorithm is applied using the UD factorization technique (Bierman, 1977). This algorithm has been recommended by Astrom and Wittenmark *et al.* (1997) and Sandoz (2003) and has been applied in many industrial applications of MPC.

The recursive least squares algorithm can be derived from the ARX model structure as follows:

$$y_t = \varphi_t \theta + \varepsilon_t \qquad (7)$$

Where:

$$\varphi_t = \left[-y_{t-1} \ \ldots \ -y_{t-n} \ \ u_{t-1} \ \ + u_{t-n} \right]$$

$$\theta = \left[a_1 \ldots a_n \ \ b_1 \ldots b_n \right]$$

Now introduce the matrix P defined by:

$$P_t = \left[\sum_{k=1}^{t} \varphi_k{}^{\mathrm{T}} \varphi_k \right]^{-1} \qquad (8)$$

Then the regression vector can be calculated from:

$$\hat{\theta}_t = P_t \left[\sum_{k=1}^{t} \varphi_k{}^{\mathrm{T}} y_k \right] \qquad (9)$$

The P matrix can be updated at each sample point using the following expression:

$$P_{t+1}{}^{-1} = P_t^{-1} + \varphi_{t+1}{}^{\mathrm{T}} \varphi_{t+1} \qquad (10)$$

and the regression vector can be updated as follows:

$$\hat{\theta}_{t+1} = \hat{\theta}_t + K_{t+1} \varepsilon_{t+1} \qquad (11)$$

Where:

$$\varepsilon_{t+1} = y_{t+1} - \varphi_{t+1} \hat{\theta}_t$$

$$K_{t+1} = P_{t+1} \varphi_{t+1}{}^{\mathrm{T}}$$

To avoid any matrix inversion problems which may occur as a result of collinearity, equation (10) can be re-cast as follows:

$$P_{t+1} = P_t - P_t \varphi_{t+1}{}^{\mathrm{T}} \varphi_{t+1} P_t \left[1 + \varphi_{t+1} P_t \varphi_{t+1}{}^{\mathrm{T}} \right]^{-1} \qquad (12)$$

By using the estimated values of y in the formulation of φ_t it is ensured that the resulting estimator is unbiased. This formulation of RLS has been studied in depth over the last few decades and reliable techniques exist for incorporating a *forgetting factor* into equation (12). Further improvements in the robustness of the algorithm are produced by using Bierman's UD factorisation of P. When this method is used, adaptation of the model parameters is typically preceded by *blowing-up* the **D** matrix used in the factorisation (Astrom *et al.* 1997).

3.2 Partial Least Squares

PLS is a recently proposed technique that combines the features of Principle Component Analysis and regularised regression. It has been proposed as a regression tool for use in situations where there are significant levels of collinearity or when there are few observations. The technique was originally proposed for static systems where the NIPALS algorithm was employed (Lindgren, *et. al*., 1993). Geladi and Kowalski, (1986) introduced an overview of PLS which included a comparative discussion of OLS and PLS algorithms and showed the deficiency of the former in the case of correlated variables. To tackle such situations in the process industries, and also to provide accurate process models when observations are limited, the PLS algorithm is beginning to be applied within model predictive control schemes.

To reduce the computational expense when identifying a PLS model, Dayal and MacGregor (1997a) proposed a modified kernel PLS method. The proposed method was based on the algorithm developed by Lindgren *et al.* (1993) but with the reduced deflation step in the algorithm. The kernel PLS algorithm is described in detail by Dayal and MacGregor (1997a) and only a simple overview is provided here.

The kernel PLS algorithm estimates the values of the modelling parameters, $\hat{\theta}$ using the matrices $\Phi^{\mathrm{T}}\Phi$ and $\Phi^{\mathrm{T}}Y$. To introduce a recursive element into the model, Dayal and MacGregor (1997a), proposed re-calculating the PLS model at each sampling instant using updated $\Phi^{\mathrm{T}}\Phi$ and $\Phi^{\mathrm{T}}\mathrm{Y}$ matrices, as defined below:

$$\left(\Phi^{\mathrm{T}}\Phi\right)_{t+1} = \lambda\left(\Phi^{\mathrm{T}}\Phi\right)_t + \varphi_{t+1}{}^{\mathrm{T}}\varphi_{t+1} \qquad (13)$$

$$\left(\Phi^{\mathrm{T}}\mathrm{Y}\right)_{t+1} = \lambda\left(\Phi^{\mathrm{T}}\mathrm{Y}\right)_t + \varphi_{t+1}{}^{\mathrm{T}}\mathrm{y}_{t+1} \qquad (14)$$

Where λ is a forgetting factor that varies between 0 and 1.

PLS models are typically biased estimators as the measurement noise is included in the measurements of y, which is then used to estimate the modelling parameters. However, as with the RLS algorithm described in section 3.1, if the model estimates for y are used in equation 14 rather than the measured values then an unbiased estimator results. In fact, there are potentially further benefits in using the P matrix determined in the UD RLS algorithm to determine the values for $\Phi^{\mathrm{T}}\Phi$ and $\Phi^{\mathrm{T}}\mathrm{Y}$, i.e.:

$$\left(\Phi^{\mathrm{T}}\Phi\right)_{t+1} = P(t)^{-1} \qquad (15)$$

$$\left(\Phi^{\mathrm{T}}Y\right)_{t+1} = P(t+1)^{-1}\hat{\theta}_t \qquad (16)$$

The benefit in using this approach is that the established mechanisms for adapting a RLS model can be applied to a PLS model. For example, the forgetting factor can be implemented within the parameter update equations and the diagonal matrix **D** used in the UD factorisation can be manipulated in the same way as with RLS models. A further advantage with this approach is that many other identification techniques, such as instrumental variables and the prediction error method, can be employed within PLS identification. It is also possible to use standard techniques (such as Himmelblau, 1970) to compute confidence limits on the predictions made by the PLS model. Calculation of confidence limits for PLS models is an area of active research (Martens and Martens, 2001).

It is a well-known fact that while over-parameterized, i.e. over-fitted, models can satisfactorily characterize the training data set, they are unable to generalise to data sets which were not used in the model development (Kresta, Marlin et al. 1994). In this sense, the unbiased PLS technique can achieve both the accuracy and precision, which characterize the two key elements of any identification algorithm (Frank 1987). In particular, the PLS technique can establish an accurate relationship between the predictor and response variables (accuracy) by efficiently estimating the regression coefficients, thereby minimizing the bias. At the same time, PLS can improve the predictive ability (precision) by minimizing the variance of the prediction errors.

3.3 Real-Time Adaptation

On-line model adaptation, or recursive identification is an important issue in process control problems. A serious issue associated with closed loop identification schemes is the correlation that exists between the non-stationary noise and the input signals under the feedback action (Ljung, 1987 and Soderstrom and Stoica, 1989). This correlation causes the estimated model coefficients to converge to incorrect values. Continuous adaptation of the process model is common in adaptive control techniques. However, such an approach can have a detrimental effect on model accuracy when the data is not rich, i.e. when there is no sufficient external excitation of the system.

Ideally, external excitation, or a *dither signal*, should be applied to the output of the control system. This signal provides a rich collection of data which can then be used to update the model. Unfortunately, many situations exist, particularly in batch processing, where the introduction of such excitation is not possible. In this paper, two forms of adaptation are considered. The first is *continuous* adaptation,

which involves updating the covariance matrix using routine closed loop data and updating the model at each sampling instant. The second approach is *short term* adaptation. This method involves collecting a small sample of normal closed loop data and then updating the model using this data. This process of adaptation is repeated on a routine basis. These two techniques are described in more detail in (Sandoz, 2003).

4. CASE STUDIES

In the following section the performance of four model predictive control systems are compared. The difference with each of these controllers is the model that is used within it. These models are biased RLS (BIRLS), unbiased RLS (URLS), biased recursive PLS (BIRPLS) and unbiased recursive PLS (URPLS). The difference between the biased and unbiased models is in the computation of the covariance matrices. For the biased models, the actual output measurements are used and for the unbiased models, the estimated output measurements are used to compute the covariance matrices. To compare the performance of the different models, two different processes have been investigated. The first is a two-input, two-output, first order linear system and the second is the non-isothermal CSTR investigated by Dayal and MacGregor (1997b).

4.1 MIMO First Order Linear System

This system has been taken from Huang *et al.* (1997) and the open-loop transfer function matrix G and disturbance transfer function matrix D are given as follows:

$$G = \begin{bmatrix} \frac{q^{-1}}{1-0.4q^{-1}} & \frac{q^{-2}}{1-0.1q^{-1}} \\ \frac{0.3q^{-1}}{1-0.1q^{-1}} & \frac{q^{-2}}{1-0.8q^{-1}} \end{bmatrix} \quad D = \begin{bmatrix} \frac{1}{1-0.5q^{-1}} & \frac{-0.6}{1-0.1q^{-1}} \\ \frac{0.5}{1-0.5q^{-1}} & \frac{1}{1-0.5q^{-1}} \end{bmatrix}$$

An MPC control system was designed for this system with a prediction horizon of 20 (based on the slowest settling time of the system), an output horizon of 1, a control move weighting of 0, a sample time of 1 second and with no process constraints. The structure of the model for each identification algorithm was chosen to be ARX. The order of both A and B polynomials was set to one. Hence, there were three coefficients in each model. For the PLS models, three latent variables were used.

For identification purposes, a constant reference signal of 1.0 was used for each output and a pseudo-random binary sequence (PRBS) was applied to the system. This signal varied between -0.25 and +0.25. A white noise signal was added to the output measurements to simulate the effect of measurement noise. The noise- to- signal ratio was 0.2 for each output. To test the performance of the developed controllers, a coloured noise disturbance was added to the output of the process. The parameters of the estimated model were initially assumed to be equal to one and the diagonal coefficients (**D**) are specified to be 10^6 (Astrom *et. al.*, 1997).

4.2 Results and Discussion

Figure (1) compares the step response of the four models with that of the actual plant. This figure shows the step response between *u1*, *u2* and *y1*. However similar results were obtained for all the step response functions. Figure (1) highlights the following:

1. The response of the BIRLS and BIRPLS models is almost identical, as are the URLS and URPLS models. This is expected as the PLS models should tend towards the RLS models unless collinearity is a major problem. In this case study collinearity in the data was not an issue; hence all the latent variables were included in the PLS model.
2. The unbiased models have a very similar step response to the actual system and hence provide a significant improvement in modelling accuracy over the biased models.

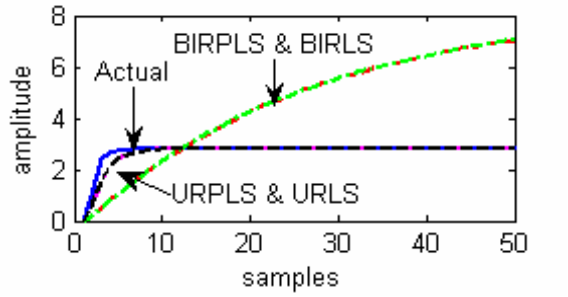

Fig (1), Step response test for *y1*

The importance of an accurate model is illustrated in figures (2) and (3), which compare the ability of the URPLS and BIRPLS MPC systems to maintain the set-point of *y1* at 1.0. These figures clearly show the benefit that the proposed URPLS based MPC system has over the control system that utilises BIRPLS prediction model.

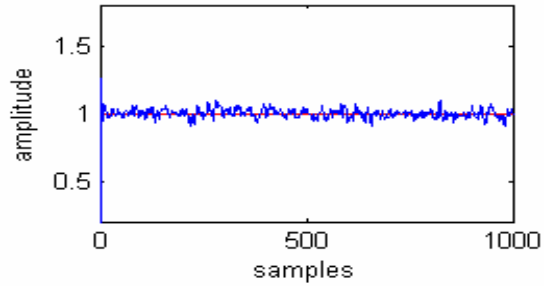

Fig (2), Closed-loop response of *y1* using URPLS

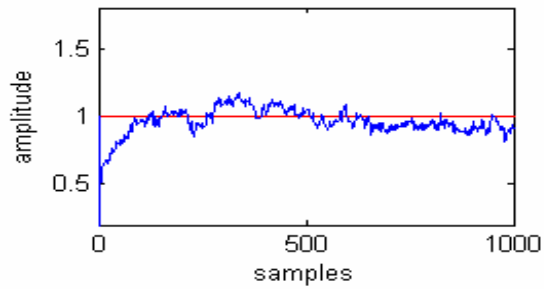

Fig (3), Closed-loop response of *y1* using BIRPLS.

Table (1) summarises the results from this investigation. These results show the integral square error (ISE) statistic for

each of the model predictive controllers. The error in the ISE statistic is the difference between the output and set-point.

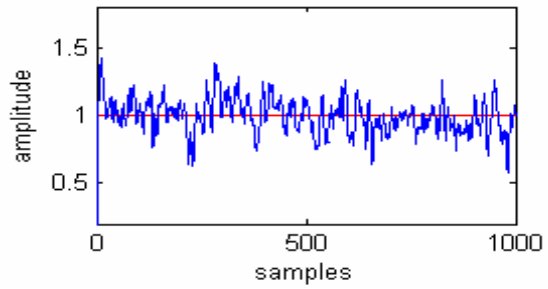

Fig (4), Closed-loop response of *y2* using URPLS

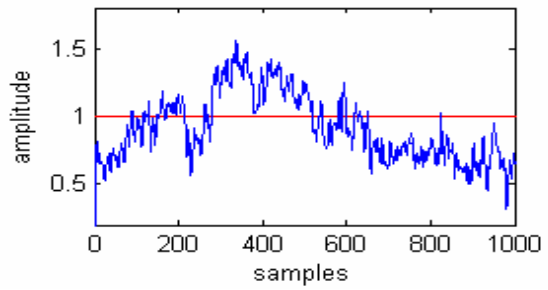

Fig (5), Closed-loop response of *y2* using BIRPLS

From the results in table (1) it is possible to conclude that the unbiased models have performed significantly better than the biased models. This highlights the importance of using unbiased models in MPC systems, which is something that has been ignored in the past when using PLS models. As expected, the results for the RLS and PLS models are very similar since collinearity amongst predictor variables is not an issue in this particular case study. This is an important result as it suggests that PLS models do not cause any significant reduction in control performance.

The reason for the higher error statistics for *y2* is simply a phenomena associated with multivariate systems. It is quite common to find that one loop is faster or more tightly controlled than another. MPC has a systematic policy to cope with this problem by appropriately selecting the output weighting matrix A.

Table (1), performance of designed MPCs

Model	ISE_y1	ISE_y2
BIRLS	11.1989	68.4458
URLS	02.2646	18.6745
BIRPLS	11.1881	69.0225
URPLS	02.2580	18.5913

4.3 Non-isothermal CSTR

This process is described in detail in Luyben (1995) and also in Dayal and MacGregor (1997b). Because of space limitations only a brief overview of this system is described here. The process is defined as irreversible with material *A* being transformed into *B*. This reaction is carried out in a perfectly mixed CSTR. The flow of component *A* into the system is $F_o C_{Ao}$ where C_{Ao} is the concentration of *A* in the inlet flow and the output flow is given by FC_A, where C_A is the concentration of *A* in the reactor. In this system the liquid height is assumed to be constant. The targets for the controller in this application are to maintain the reactor temperature *T*, at 305 *K* and to keep the residual conversion to a rate of 0.6. To control the heat of reaction, cooling water of constant volumetric flow V_J is added to the jacket that surrounds the reactor at flow rate F_J and with an inlet temperature T_{Jo}.

The manipulated variables in this process are the feed flow of material *A* and the inlet water coolant flow $(F = u_1, F_J = u_2)$, and reactor temperature and conversion rate are the control variables $(T = y_1, conv_A = y_2)$ with dead times of two sample periods for the former and five for the latter. White noise with variances of 0.002 and 4e-6 are added to the system outputs y_1 and y_2 respectively, while a random walk fluctuation in the Arrhenius rate constant (k_o) was also applied to represent a non-stationary disturbance to the system.

In the work presented by Dayal and MacGregor (1997b), they compared the performance of biased RLS and PLS models. In this work, the potential benefit of using unbiased models was to be investigated. Similarly to the approach taken by Dayal and MacGregor (1997b), an initial model for each output is identified by exciting the open-loop system with a PRBS. 300 samples with a sampling period of 20 seconds are generated from this test. A sixth order ARX structure was identified for each output variable. This order of model was chosen to match that used by Dayal and MacGregor, (1997b). The models were subsequently updated by using continuous and short term methods by means of BIRLS, BIRPLS, URLS, and URPLS algorithms. Six latent variables were considered in the PLS methods.

The short-term adaptation suggests updating the model parameters at regular intervals with a specific length of samples, in such a way to allow good tracking of the system dynamics and at the same time preventing a high computational load, and the consequences of discharging the essential covariance matrix information when no significant information has been added.

4.4 Results and discussion

Table (2) shows the performance of each controller using the ISE statistic. The main findings from this study were:

1. When using the continuous adaptation method, the RLS models caused the process to go unstable. This is believed to be because there is no independent excitation in the manipulated variables and hence these variables are highly correlated. This correlation appears to be handled well by the BIRPLS model. However, the URPLS algorithm fails to produce a stable response with this method of adaptation. The reason for this is unclear and is the subject of continued research.

2. For the short adaptation approach the unbiased models produce better results than those obtained using the biased models.
3. The URPLS model provides better control than the BIRPLS when the short adaptation method is used.
4. The results suggest that unless continuous adaptation is to be used then there is little advantage in using a PLS model. This highlights that the RLS algorithm is itself very capable of coping with significant levels of collinearity in data sets.

Table (2), the ISE factors

Model		ISE_T	ISE_Conv	comments
BIRLS	Cont	NA	NA	unstable
	Short	11.3700	0.4650	–
BIRPLS	Cont.	06.3800	0.0333	–
	Short	06.0200	0.0317	–
URLS	Cont.	NA	NA	unstable
	Short	05.0200	0.2401	–
URPLS	Cont.	NA	NA	unstable
	Short	05.4400	0.0288	–

During this study it was found that the BIRLS method required some caution in the selection of the control weighting factor to ensure stability. In contrast the BIRPLS and URPLS did not show high sensitivity to the controller weighting factor. This suggests that the resulting models using the PLS algorithm were more robust.

The results from this work can be summarised as follows:

- The upper diagonal factorization with RLS offers an effective tool against data collinearity.
- Equivalent performance can be achieved by RLS and RPLS in the open-loop system estimation.
- When the URPLS was applied in long-term fashion, the results were poor. The specific reasons for this, is the subject of future research.
- The results show that URPLS is a reliable algorithm, and can provide good results for open and closed-loop data.

5. CONCLUSIONS

In this work, an unbiased recursive partial least squares algorithm has been developed for application within a model predictive control system. The proposed algorithm provides a direct link between the industrially accepted RLS algorithm and the more recently developed PLS routine. By using a kernel based algorithm to identify the PLS model, it is possible to utilise the covariance matrices that are formulated when using RLS. This means that techniques, such as the output error method, can be utilised when designing PLS models. This ability offers significant benefits when using PLS models in industrial control problems. The benefits of the proposed PLS algorithm were illustrated through its application to two simulated systems, where its performance was compared favourably with that of RLS when the plant variables were highly correlated.

REFERENCES

Astrom, K. (1980) 'Maximum likelihood and prediction error methods', *Automatica*, **16** (5), 551-574.

Astrom, k. and Eykhoff, P. (1971), 'System identification-a survey', *Automatica***, 7** (2), 123- 162.

Astrom, K. and Wittenmark, B. (1997) *Computer controlled systems, theory and design*, Third edition, Prentice Hall.

Camacho, E. F. and Bordons, C. (1999) *Model Predictive Control*, Springer-Verlag, London.

Clark, D. W., Mohtadi, C. and Tuffs, P. S. (1987), 'Generalized predictive control-part I. The basic algorithm', *Automatica,* **23** (2), 137-148.

Cutler, C. R. and Ramaker, B. L. (1980), 'Dynamic matrix control-a computer control algorithm'. JACC, San Francisco.

Dayal, B. S. and MacGregor, J. F. (1997a), 'Improved PLS algorithms', *Journal of Chemometrics,* **11** (1)*,* 73-85.

Dayal, B. S. and MacGregor, J. F. (1997b), 'Recursive exponentially weighted PLS and its applications to adaptive control and prediction', *Journal of Process control***, 7** (3), 169-179.

Frank, I. E. (1987) Intermediate least squares regression method. *Chemometrics and Intelligent Laboratory Systems* **1**(3): 233-242.

Geladi, P. and Kowalski, B. (1986) 'Partial least squares regression: a tutorial' *Analytical Chemical Acta***, 35**, 1-17.

Himmelblau, D, 1970, *Process Analysis by Statistical Methods*, Wiley.

Huang, B., Shah, S. L. and Kwok E. K. (1997) Good, Bad or optimal? Performance Assessment of Multivariable Processes*, Automatica* , **33** (6), pp. 1175-1183.

Kresta, J. V., T. E. Marlin and MacGregor, J. F. (1994).Development of inferential process models using PLS. *Computers & Chemical Engineering.***18**(7): 597-611.

Lindgren, P., Geladi P. and Wold, S. (1993), 'The kernel algorithm for PLS', *Journal of Chemometrics***, 7** (1)**,** 45-59.

Ljung, L. (1987) *System Identification: Theory for the User*, Englewood Cliffs, NJ Prentice-Hall.

Ljung, L. (1995) *System identification*, McGraw-Hill

Luyben, W. (1995). *Process Modelling, Simulation, and Control for Chemical Engineers*, McGraw-Hill.

Martens, H and Martens, M, (2001), *Multivariate Analysis of Quality*, Wiley.

Sandoz, DJ, (2003), 'The exploitation of adaptive modeling in the model predictive control environment of Connoisseur', in *Techniques of Automatic Control*, Ed. VanDoren V, Butterworth.

Soderstrom, T. and Stoica, P. (1989), *System Identification*, Englewood Cliffs, NJ Prentice-Hall.

Soderstrom, T. Ljung, L. and Gustavsson, I. (1978) 'A theoretical analysis of recursive identification method'*, Automatica,* **14,** 231-244.

Zhu, Y. (2001), *Multivariable System Identification .for Process Control.*