

FAULT DETECTION AND ISOLATION IN A FED-BATCH PENICILLIN FERMENTATION PROCESS

Hongwei Zhang^{*}, Barry Lennox[†]

^{*}School of Engineering, North East Wales Institute of Higher Education, Wrexham, Wales, UK,
Email: H.Zhang@newi.ac.uk, Fax: 01978 293 392

[†]School of Engineering, University of Manchester, Manchester, UK,
Email: Barry.Lennox@man.ac.uk, Fax: 0161 275 3844

Keywords: Multi-way partial least squares, model predictive control, fault detection, fault isolation, penicillin fermentation

Abstract

This paper deals with fault detection and isolation (FDI) in a fed-batch penicillin fermentation process. The method is based on an empirical model developed using the multi-way partial least squares technique. The paper begins by demonstrating how this model can be used to provide FDI capabilities and then shows how this FDI scheme can be integrated within a model predictive controller to provide accurate control in the presence of fault conditions within the process.

1. Introduction

Due to the high demand for improved product quality and economic operation of industrial fed-batch fermentation processes, fault detection and isolation (FDI) has become an important topic of research. The early and accurate detection of fault conditions is of great benefit in fed-batch fermentation processes since the earlier that a fault can be detected and acted upon, the lower its impact will be on the process. In some situations this can be critical, for example, a drift on a pH sensor could have catastrophic results on biomass growth if this measurement is used within a feedback control scheme. The FDI approaches that have been proposed for application to fed-batch fermentation processes tend to fall into two categories: model-based and knowledge-based.

In the model based approach, a mathematical model of the process is firstly established using data collected under normal operating conditions (NOC). The model is then used to predict the behaviour of the process and detect any abnormal change by comparing the predicted and actual sensor measurements. In the knowledge-based approaches, such as expert systems, prior process knowledge and diagnosis reasoning are encoded to build a knowledge-based system (KBS) in the form of facts, rules, and heuristics which are derived from a theoretical understanding of the process or observations of process behaviour. The developed KBS is then used to reason and derive conclusions which are provided to operations staff. A major drawback with this approach is that the acquisition of data for the KBS is not trivial [8].

Among the model-based approaches, a particularly promising approach is the application of multivariate statistical process control techniques, such as Principal Component Analysis (PCA) and Partial Least Squares (PLS). The benefits of using these approaches, rather than more traditional statistical methods, such as Statistical Process Control have been demonstrated through their application to fed-batch fermentation systems. These applications have exploited a variety of multivariate statistical routines to accurately detect and isolate fault conditions within a fermenter [6, 7].

In previous studies FDI and control of fed-batch fermentation processes have been viewed as independent problems. In this paper, however, these problems are considered together and an integrated fault detection and process control scheme is developed. This scheme employs a predictive control framework based on the soft-sensing capabilities of a PLS model. By utilising a PLS model within the predictive controller it is demonstrated that the controller can be used to provide fault detection and isolation capabilities. The integration of the predictive controller and FDI scheme is also shown to provide a useful diagnostic tool in the presence of fault conditions.

In the following section, a brief description of the process that is investigated in this work is described. This is followed by an overview of the multi-way PLS (MPLS) algorithm. Section 4 demonstrates the ability of a PLS model to provide FDI capabilities when applied to a benchmark penicillin fermentation process. Section 5 then demonstrates how this PLS model can be integrated within a predictive control algorithm and applied to the simulation. Finally, the conclusions from this work are discussed.

2. Benchmark fermentation process

Secondary metabolites such as antibiotics, and in particular penicillin, have important added value, and therefore improvements in their production are of great interest to industry. For this reason there has been a great deal of research conducted during the last decade on all aspects of penicillin production [4, 11]. The work described in this paper is concerned with providing improved operating capabilities in the production of penicillin. To demonstrate the benefits of the algorithms proposed in this paper, the simulation of a penicillin fermentation process developed by the Process

Modelling, Monitoring and Control Research Group at the Illinois Institute of Technology [2] has been used.

This simulator is based on the unstructured mechanistic model of Bajpai and Reuss [1] and is capable of simulating a controlled fed-batch fermentation system. The load variables are: aeration rate, agitator power, substrate feed rate and substrate feed temperature; the manipulated variables are: acid/base and heating/cooling water flow rates; the internal state variables are: culture volume, generated heat, carbon dioxide, dissolved oxygen, biomass, penicillin and substrate feed concentrations; and the controlled variables are: pH and bioreactor temperature.

3. Partial least squares

To develop a predictive controller for application to the benchmark simulation, it is first necessary to develop a model, capable of predicting the required control variable. In this paper, the work focuses on the development of a controller capable of regulating the biomass concentration, which represents an important quality variable from the process. This model was developed using the regression technique referred to as partial least squares, which is now described.

3.1 Basic algorithm of PLS

PLS is a regression tool that is ideally suited for situations where high levels of correlation exist between cause variables. The approach works by selecting factors of the cause variables in a sequence that successively maximises the explained covariance between the cause and effect variables. The factors of the cause and effect variables are defined as follows:

$$\mathbf{X} = \sum_{k=1}^{np < nx} \mathbf{t}_k \mathbf{p}_k^T + \mathbf{E} \quad \text{and} \quad \mathbf{Y} = \sum_{k=1}^{np < ny} \mathbf{u}_k \mathbf{q}_k^T + \mathbf{F} \quad (1)$$

where \mathbf{X} and \mathbf{Y} represent the cause and effect variables respectively (typically scaled to zero mean and unit variance); \mathbf{t}_k and \mathbf{u}_k represent the cause effect factors respectively; \mathbf{p}_k and \mathbf{q}_k represent the set of orthogonal vectors, of length nx and ny respectively, referred to as *loadings*; \mathbf{E} and \mathbf{F} are residual matrices; np is the number of inner components that are used in the model; nx is the number of causal variables.

The equations defined in (1) are referred to as the *outer relationships*. The \mathbf{t}_k vectors, which are mutually orthogonal, and \mathbf{u}_k are selected so as to maximise the covariance between each pair, $(\mathbf{t}_k, \mathbf{u}_k)$. Linear regression is performed between the \mathbf{t}_k and the \mathbf{u}_k vectors to produce the *inner relationship*, defined as:

$$\mathbf{u}_k = b_k \mathbf{t}_k + \varepsilon_k \quad (2)$$

where b_k is a regression coefficient, and ε_k refers to the prediction error. The number of latent variables that are used in the model is an important specification and in this work is made through the use of cross-validation. Geladi [3] provides further details regarding the PLS algorithm.

3.2 Multi-way partial least squares (MPLS)

PLS is a linear regression tool and therefore its application to non-linear, time-varying processes, such as fed-batch fermentation systems is limited. To overcome this problem, Nomikos and MacGregor [10] developed a modified PLS approach, termed multi-way PLS. MPLS uses a technique referred to as *unfolding* to re-arrange the cause data collected from the batch, which can be considered to be a 3-dimensional array of size $mx \times nx \times nb$, where mx is the number of samples taken during a batch, nx is the number of cause variables that are measured and nb is the number of batches for which data is available. This 3-dimensional array is unfolded into a 2-dimensional array, of size $nb \times (nx \cdot mx)$, as shown in figure 1. The effect data is unfolded in a similar way to create a 2-dimensional array of size $nb \times (ny \cdot my)$, where ny and my are the number of effect variables that are measured and the number of samples that are taken of the effect variables respectively. It is worth noting that mx does not need to be, and in most cases will not be, equal to my .

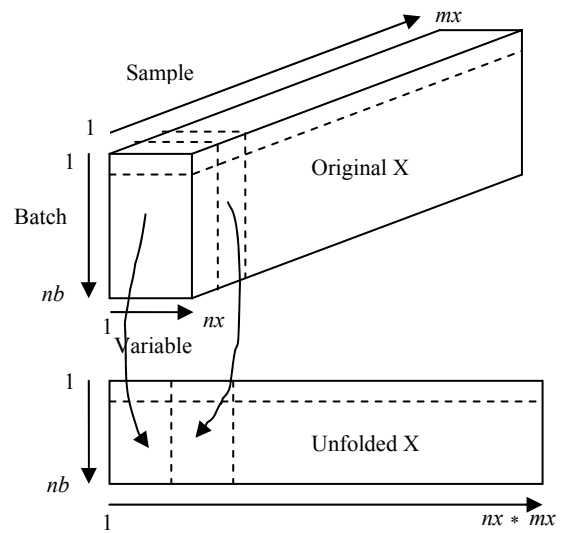


Figure 1: Unfolding

Following the unfolding of the data, the columns of both data sets are scaled to zero mean and unit variance and the standard PLS algorithm applied.

The subsequent use of this model on-line poses the problem that it is necessary to know the values of all process measurements through to the end of the batch. This necessitates the need to estimate the future values of the measured variables as each batch progresses. The estimation of future process values is referred to as *filling up* the array. Nomikos and MacGregor [10] suggested three alternative methods for filling up the array and in this work, the most appropriate method was to assume that the values of all process measurements remain at their current offset from the mean trajectory through to the end of the batch. It is worth mentioning that a problem frequently encountered with the application of MPLS to fed-batch fermentation processes is that the run length of the batch can often vary. Fortunately, this was not a problem in the applications explored in this paper. However, various solutions exist to accommodate such conditions, see for example Lennox *et al* [8].

3.3 MPLS for fault detection

In addition to providing predictive capabilities, process models developed using the MPLS algorithm may also be used for detecting and isolating fault conditions. The approach for achieving this is very similar to that adopted when using Principal Component Analysis (PCA) [12]. Rather than attempting to detect the presence of any fault condition by monitoring each process variable independently, two univariate statistics, referred to as the *SPE* (Squared Prediction Error) and T^2 statistics, are monitored. These statistics are defined as follows:

$$SPE = \mathbf{E}^2 \quad (3a)$$

$$T^2 = \sum_{k=1}^{np} \mathbf{t}_k \sigma_k^{-2} \mathbf{t}_k^T \quad (3b)$$

where σ_k^2 is the variance of the k -th principal component.

Confidence limits can be placed around these statistics, which if violated indicate deviations from the process conditions that were recorded in the data used to develop the PLS model. The character of these deviations will be reflected in the particular chart that is violating its confidence limit. Goulding *et al* [5] demonstrated that changes in the relationships between variables, such as would be experienced if for example a sensor failed, tended to be detected on the *SPE* chart, whilst changes in operating condition, for example a grade change, were typically identified on the T^2 chart. There will be exceptions to this, for example a high impact fault which significantly affects a number of variables is likely to be detected on both the T^2 and *SPE* charts.

4. Application to fed-batch fermenter

4.1 PLS model development

The first stage in the development of the PLS model is to generate suitable training data. In this application data from 30 batches was collected. For each batch a pseudo-random signal (PRS) was applied to the substrate feedrate to ensure the data was sufficiently rich. 20 of these batches were used to train the PLS model (training batches) with the remainder used to validate the model (validating batches).

A PLS model, containing 3 latent variables, was then developed using this data. In this model the following measurements were used as input, or cause, variables: substrate feed rate aeration rate, agitator power, substrate feed temperature, substrate concentration, dissolved oxygen concentration, culture volume, pH, fermenter temperature and generated heat. The concentration of biomass and Penicillin were selected as 'effect' variables. The accuracy of this model has been tested in simulation batches to compare the actual biomass/Penicillin concentration with that predicted by the PLS model and it has been found to be satisfactory.

The PLS model was then integrated within a relatively standard model predictive controller (MPC). By using the PLS model to predict the future behaviour of the process, the predictive control algorithm is able to determine suitable control moves that will minimise the value of a pre-

determined cost function. In this application, the following cost function was utilised:

$$J = \sum_{k=t}^{t+N} [\hat{y}_k - w_k]^2 + \lambda \Delta u_k^2 \quad \text{for } t + N < t_{end} \quad (4)$$

$$J = \sum_{k=t}^{t_{end}} [\hat{y}_k - w_k]^2 + \lambda \Delta u_k^2 \quad \text{for } t + N \geq t_{end}$$

where: t is the current sample time of the batch; N is the length of the prediction horizon, which in this application was set to 10; w_k is the desired biomass concentration at sample time, k ; \hat{y}_k is the biomass concentration that is predicted by the PLS model at sample time k ; Δu_k is the change in the manipulated variable (the substrate feed) made at sample time, k ; λ is a tuning parameter.

The intention of this work was to develop a model predictive control system that was tolerant to the introduction of process faults into the process. Therefore, following the development of the predictive controller, which was found to provide acceptable control performance, its ability to initially detect process faults and then to compensate for them so that their impact on the process was minimised was investigated.

4.2 Fault detection and isolation

To test the ability of the PLS model to detect and isolate fault conditions within the fermenter, a series of faults were introduced into the process, while it was under closed loop control. The faults that were introduced were those that were suggested and analysed by the authors of the simulation [11]. Online multivariate statistics for monitoring the batch in real time consist of the *SPE* and T^2 charts as well as the individual variable contributions charts.

Figure 2 illustrates the ability of the PLS model to correctly detect and isolate one process fault, a step fault of 5% applied to the aeration measurement between 50 and 100 hours. Figure 2a shows that the *SPE* chart violates the 95% confidence limit immediately after the fault condition occurs, while the T^2 chart remains well below its limit. The inability of the T^2 chart to detect this fault condition is consistent with the results of Goulding *et al* [5] who demonstrated that the *SPE* chart is much more sensitive to the detection of faults than the T^2 chart. Figure 2b shows the *SPE* contribution chart that was produced immediately after the *SPE* limit was violated. This chart indicates that variable 1, which is the aeration measurement, is the likely cause of the fault condition.

Figures 3 and 4 illustrate the control charts that were recorded during two further faults conditions. The first, figure 3, is the result of a ramp increase of 20% applied to the aeration measurement between 50 and 100 hours and figure 4 is produced when there is a 20% step increase in agitator power. These figures show that the PLS model is able to detect these faults. It is also evident from figures 3b and 4b (both at hour 50) that the PLS model has correctly identified the two conditions as a fault with the aeration measurement (variable 1) and agitator power (variable 2). In further tests it was

found that the PLS model was able to detect and isolate all the faults that were investigated by Undey *et al* [11].

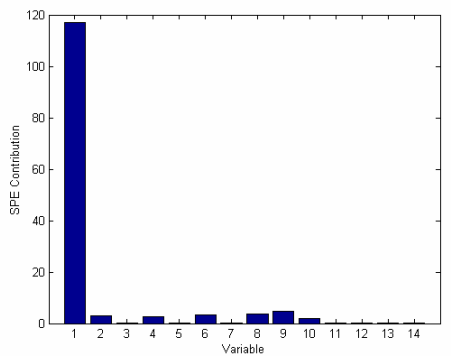
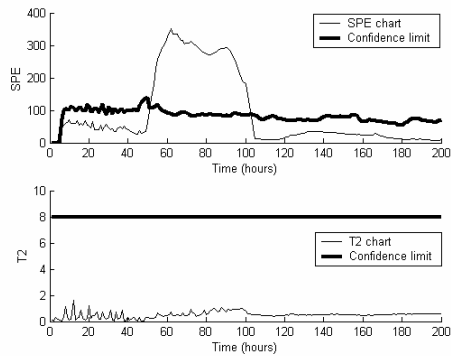


Figure 2: PLS monitoring and contribution charts for 5% step aeration fault

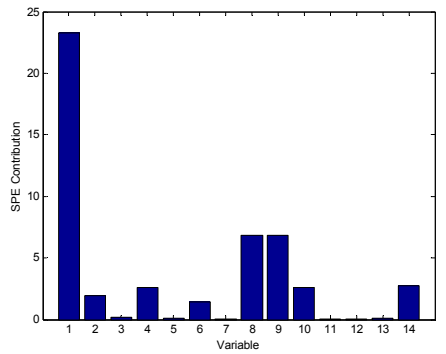
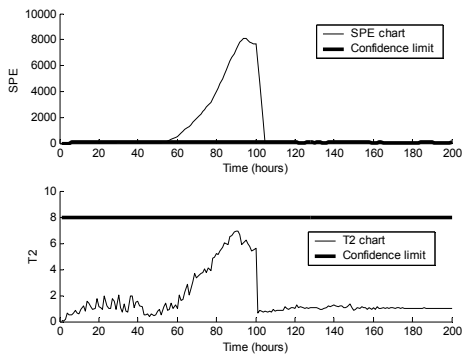


Figure 3 PLS monitoring and contribution charts for 20% ramp aeration fault

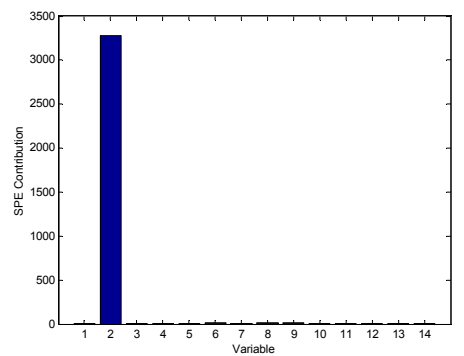
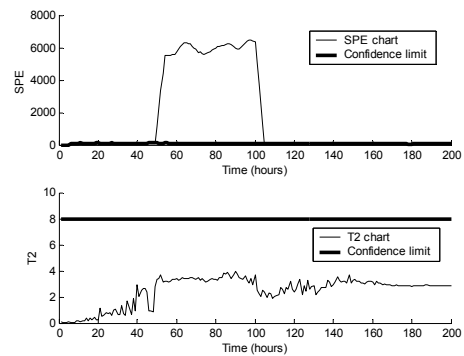


Figure 4 PLS monitoring and contribution charts for 20% step agitator fault

5. Intelligent fault detection and process control

A weakness of most process control and FDI schemes is that following the detection of a fault condition, such as a drift on a sensor, the scheme is rendered useless until the particular fault is resolved. This is of particular concern if there is likely to be a delay between the fault being detected and it being repaired.

This problem is highlighted in the following example, where a step increase of 0.5 is applied to the pH sensor measurement, 100 hours into a batch. This fault creates a major disturbance to the process as the pH measurement is used in two automatic control loops. The pH measurement is firstly used in a single loop feedback, PID control scheme that regulates the pH in the fermenter to 5. Since there is a drift of 0.5 on the pH sensor then this controller will actually maintain the pH within the reactor at 4.5, which could result in an adverse response by the bio-organisms inside the fermenter. This is illustrated in figure 5 which shows the *actual pH* in the fermenter. It can be seen from this figure that when the drift is introduced into the sensor measurement at a time of 100 hours, the pH control system gradually reduces the pH in the fermenter to 4.5. Figure 6 shows the corresponding acid and base flow rates during this fault. A further concern is that the pH measurement is used by the PLS model to predict the biomass concentration and this prediction is used within the predictive controller to regulate the production of biomass. Since the pH measurement is incorrect then so too will be the biomass prediction made by the model. As a consequence the predictive controller will operate poorly and the productivity of the batch may be

affected. This is reflected in figure 7 which shows that if the MPC controller uses the raw pH measurement, the penicillin production reduces significantly.

The ideal solution for this problem is for the pH sensor fault to be quickly detected and isolated and a mechanism put in place so that the two control loops that utilise the pH measurement can function normally in spite of this fault. Figure 8 displays the PLS monitoring charts produced by the PLS model during this fault. The charts clearly show that the fault has been detected following its introduction after 100 hours. Figure 8b, the contribution chart, shows the fault is caused by variable 9, which is the pH measurement.

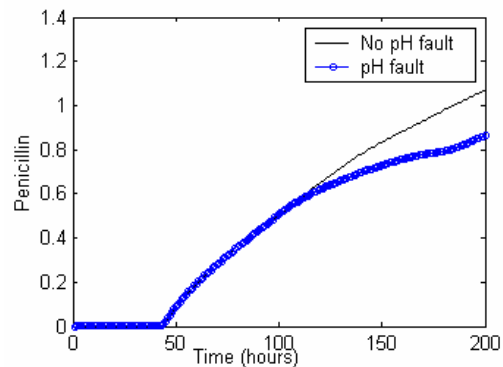


Figure 7 Penicillin production

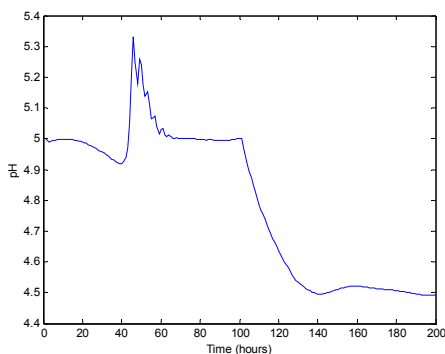
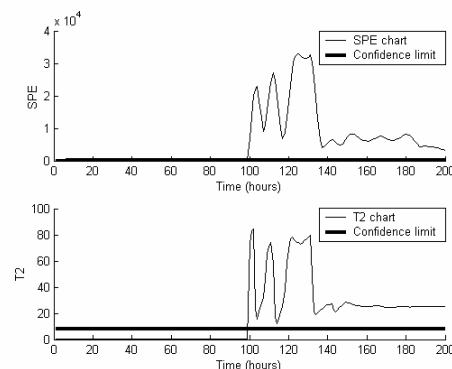
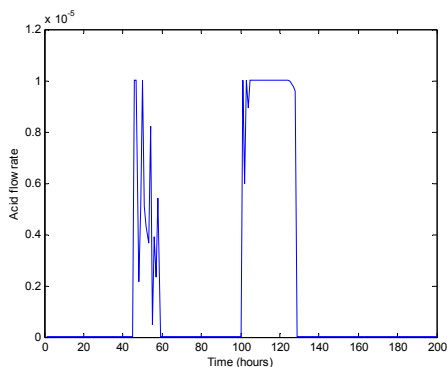


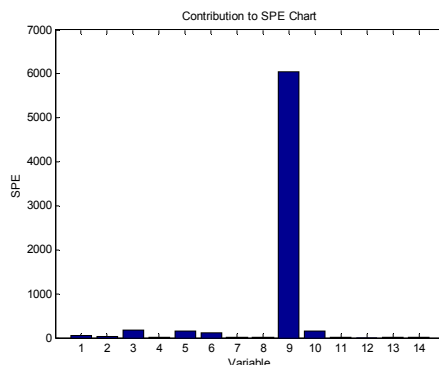
Figure 5 Actual pH in the fermenter during a pH fault



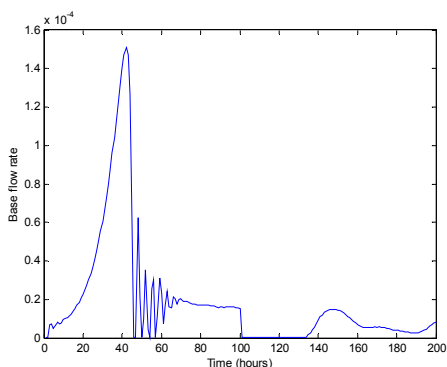
(a)



(a)



(b)



(b)

Figure 6 Acid and base flow rate during a pH fault

Figure 8: PLS monitoring and contribution charts during a pH fault

An advantage in using a PLS model is that if a sensor measurement is in question then it can be used to infer this measurement, using the techniques proposed by Nelson et al [9], until the fault is resolved. In [9], several methods are presented - single component projection, projection to the model plane and data replacement by the conditional mean, to estimate scores from an existing PCA or PLS model when new observation vectors are incomplete. In this work the following procedure is adopted following the detection of a fault condition on the pH sensor:

- The local PID controller is switched to manual until the sensor fault can be confirmed. The reason for switching the controller to manual is to prevent the PID algorithm from reacting to a sensor fault.
- Following the confirmation that the sensor is at fault then the PLS model is used to infer the pH sensor measurement.

The local PID controller is then switched back to automatic and the inferred pH sensor measurement is then supplied to both the local and MPC controllers. In this work it is assumed that should the confidence limit be exceeded in 5 continuous samples then this represents a real sensor fault, rather than a false alarm.

The inference capabilities of the PLS model were found to be very good in this application, this is illustrated in figure 9 which compares the actual pH measurement with that predicted, or inferred, using the PLS model and that measured using the pH sensor. As a result of the accurate inference of the pH measurement the performance of the controller, and subsequent productivity of the batch, was found to be unaffected by the pH sensor fault.

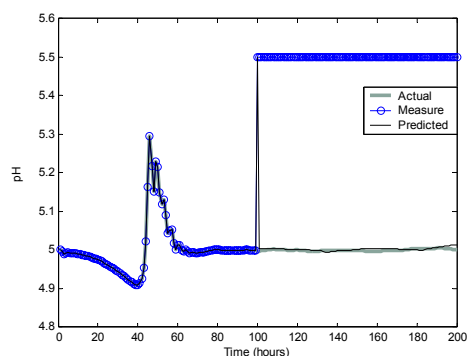


Figure 9: pH Measurement

6 Conclusions

This paper has demonstrated the capabilities of the MPLS technique for fault detection and isolation of fed-batch fermentation systems. It is shown, through application to a benchmark simulation, that multi-way PLS can be used to provide early detection and isolation of fault conditions within the fermenter. A further advantage of using a model developed using the PLS algorithm is that it can be integrated within a fairly standard model predictive control strategy to maintain the control performance in the event of sensor failures.

Acknowledgements

This work has been completed thanks to the funding provided by EPSRC (grant number GR/N24858). The authors would also like to acknowledge The Process Modelling, Monitoring, and Control Research Group at IIT who generously provided the source code for their *PenSim*© simulator.

References

- [1] R.K. Bajpai, M. Reuss. "A mechanistic model for penicillin production", *Journal of Chemical Technology and Biotechnology*, **30**, pp.332-344, (1980).
- [2] I. Birol, C. Undey, G. Birol, A. Cinar. "A web-based simulator for penicillin fermentation". *International Journal of Engineering Simulation*, **2**, pp. 24-30, (2001).
- [3] P. Geladi, B.R. Kowalski. "Partial least squares regression: a tutorial", *Anal. Chim. Acta.*, **185**, pp. 1-17, (1986).
- [4] E. Gomez Sanchez, M.J Arauzo Bravo, J.M. Cano Izquierdo, Y.A. Dimitriadis, J. Lopez Coronado, M.J. Lopez Nieto. "Control of the penicillin production using fuzzy neural networks", *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, IEEE, **6**, pp. 446-450, (1999).
- [5] P.R. Goulding, B. Lennox, D.J. Sandoz, K. Smith, O. Marjanovic. "Fault detection in continuous processes using multivariate statistical methods", *International Journal of Systems Science*, **31**, 1459-1471, (2000).
- [6] S. Lakshminarayanan, R.D. Gudi, S.L. Shah, K. Nandakumar. "Monitoring batch processes using multivariate statistical tools: extensions and practical issues". *Proceeding of IFAC World Congress*, San Francisco, pp. 241-246, (1996).
- [7] B. Lennox, G.A. Montague, H.G. Hiden, G. Kornfeld, P.R. Goulding. "Process monitoring of an industrial fed-batch fermentation", *Biotechnology and Bioengineering*, **74**, pp. 125-135, (2001).
- [8] B. Lennox, K. Kipling, J. Glassey, G. Montague. "Automated production support for the bioprocess industry", *Biotechnology Progress*, **18**, pp. 269-275, (2002).
- [9] P.R.C. Nelson, P.A. Taylor, J.F. Macgregor. "Missing data methods in PCA and PLS: score calculations with incomplete observations", *Chemometrics and Intelligent Laboratory Systems*, **35**, pp. 45-65, (1996).
- [10] P. Nomikos, J.F Macgregor. "Monitoring batch processes using multi-way principal component analysis". *AIChE Journal*, **40**, pp. 1361-1375, (1994).
- [11] C. Undey, E. Tatara, B.A. Williams, G. Birol, A. Cinar. "Hybrid supervisory knowledge-based system for monitoring penicillin fermentation", *Proceedings of the American Control Conference*, IEEE, **6**, pp. 3944-3948, (2000).
- [12] B.M. Wise, N.B. Gallagher. "The process chemometrics approach to process monitoring and fault detection", *Journal of Process Control*, **6**, pp. 329-348, (1996).