GAUSSIAN PROCESS REGRESSION FOR BATCH PROCESS MODELLING

Xiaoling Ou, Julian Morris and Elaine Martin

Centre for Process Analytics and Control Technology,
School of Chemical Engineering and Advanced Materials,
University of Newcastle, Newcastle upon Tyne, NE1 7RU, UK

Abstract: This paper proposes the application of Gaussian process regression for the empirical modelling of batch processes to provide long range predictions. Gaussian processes are flexible, non-parametric Bayesian regression techniques. In the training stage, hyper-parameters that define the covariance structure of the Gaussian process can be obtained using Markov Chain Monte Carlo sampling. Model predictions can then be achieved by taking the average of the Monte Carlo samples. The proposed technique is evaluated by application to a benchmark simulation of a fed-batch bioreactor. The results show that comparable results can be achieved with other non-parametric modelling approaches such as recurrent neural networks. *Copyright © 2004 IFAC*

Keywords: Batch process, dynamic modelling, Gaussian process, Monte Carlo methods.

1. INTRODUCTION

It can be argued that the core task in process control and process optimisation is the establishment of an accurate model of the process. One approach has been to build a mechanistic model of the process. However, the development of mechanistic models for complex batch processes is challenging and time-consuming. In addition, the mismatch between the model and the real process is generally not negligible and hence needs to be incorporated within the analysis. Consequently a number of data-based parametric/non-parametric modelling methods have been applied in process manufacturing applications, including the linear latent variable techniques of Principal Component Analysis (PCA), Partial Least Squares (PLS) (Kourti, 2003) along with the non-linear approach of neural networks (e.g. Tian, *et al*., 2002).

More recently, Bayesian statistical tools have emerged as promising solutions to a number of modelling and model related problems in process engineering. Bayesian inference is a procedure whereby a probability model is fitted to a set of data and the results are summarised in terms of the posterior probability distribution (Gelman, *et al*., 1995; West and Harrison, 1997). The Bayesian approach provides a natural way to incorporate prior information with observed data from a real process, leading to generally better performance than non-Bayesian approaches in many applications, e.g. Nounou, *et al*., (2002).

Gaussian processes are non-parametric Bayesian regression models formulated from Gaussian prior distributions defined over the infinite-dimensional space of all possible regression functions. The behaviour of a Gaussian process model can be controlled through the covariance function in terms of the underlying hyper-parameters. A wide variety of covariance functions can be applied in Gaussian process modelling subject to the requirement that the function must result in a non-negative definite covariance matrix (Neal, 1997).

For a Bayesian approach, the hyper-parameters have an associated prior distribution with their posterior distribution being identified using Markov Chain Monte Carlo (MCMC) approaches as opposed to analytical Bayesian inference. This is because of the difficulty of integrating analytically over a posterior distribution that is of complicated form (Brooks 1998). Predictions can then be achieved by averaging over the Monte Carlo samples.

In Rasmussen’s (1996) work, it was demonstrated that Gaussian process models have comparable, if not better, predictive performance than other nonparametric regression methods such as neural networks. Their good predictive performance and analytical properties have made them increasingly attractive to users and has resulted in their widespread application in statistics, engineering and other fields (Neal, 1997; Shi, *et al*.; 2003Williams and Rasmussen, 1996).

In this paper a Gaussian process model is developed for a fed-batch bioreactor for long range predictions. The model is implemented using one of the more popular MCMC methods, the Metropolis-Hastings (MH) algorithm (Robert and Casella, 1999). Given the initial state of the fed-batch reactor, multi-step-ahead predictions can be calculated using an iterative approach. The results are sufficiently accurate to provide the basis of model based control algorithms.

2. GAUSSIAN PROCESS MODELLING WITH MCMC IMPLEMENTATION

The problem to be addressed is first described. Consider a noisy data set *D* comprising *N* pairs of *d*-dimensional input vectors and scalar outputs, i.e. $D = \left\{\mathbf{x}^{(k)}, y^{(k)}\right\}_{k=1}^{N}$. The primary objective is to establish a Gaussian process model based on this data set and use it to predict, *y*, given any new input $\mathbf{x} \notin D$. Adopting an iterative approach to the multi-step ahead case, the new input will include the previous predicted values of *y*.

2.1 Prediction using Gaussian Process Model

As for all regression approaches, a Gaussian process model can be defined as:

$$y = f(\mathbf{x}) + \varepsilon \qquad (1)$$

where $f(\cdot)$ is a non-linear function mapping the input vector **x** to a scalar output *y*. ε is Gaussian noise with zero mean i.e. $\varepsilon \sim G(0, \sigma_v^2)$. The prior for $f(\cdot)$ is assumed to be a Gaussian process, $\{f(\cdot)\} \sim G(\mu, \mathbf{C})$, where μ is usually set to 0 and **C** is the covariance matrix. In practice normalisation is preferred for numerical stability, although it is not strictly necessary. Thus, $\left\{y^{(k)}\right\}_{k=1}^{N}$ is Gaussian with the mean and variance given by:

$$\hat{y}(\mathbf{x}) = \mathbf{k}^T(\mathbf{x})\mathbf{\Sigma}^{-1}\mathbf{y} \qquad (2)$$

$$\hat{\sigma}_{\hat{y}}^2(\mathbf{x}) = \Sigma(\mathbf{x},\mathbf{x}) - \mathbf{k}^T(\mathbf{x})\mathbf{\Sigma}^{-1}\mathbf{k}(\mathbf{x}) \qquad (3)$$

where $\mathbf{\Sigma}$ is the covariance matrix for the training data set:

$$\mathbf{\Sigma} = \mathbf{C} + \sigma_v^2\mathbf{I}\,, \quad \mathbf{I} \text{ is the identity matrix} \qquad (4)$$

The entries are defined as $\Sigma_{ij} = \Sigma(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, $\mathbf{k}(\mathbf{x}) = \left(\Sigma(\mathbf{x}, \mathbf{x}^{(1)}), \ldots, \Sigma(\mathbf{x}, \mathbf{x}^{(N)})\right)^T$, $\mathbf{y} = (y^{(1)}, \ldots, y^{(N)})^T$.

The only constraint on the choice of covariance function $\Sigma(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ is that it must generate a non-negative definite covariance matrix $\mathbf{\Sigma}$ for any set $\left\{\mathbf{x}^{(k)}\right\}_{k=1}^{N}$. A widely used form of $\Sigma(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ is:

$$\begin{aligned}\Sigma(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = {} & v_0 \exp\left\{-\frac{1}{2}\sum_{l=1}^{d} w_l \left(x_l^{(i)} - x_l^{(j)}\right)^2\right\} \\ & + a_0 + a_1 \sum_{l=1}^{d} x_l^{(i)} x_l^{(j)} + \sigma_v^2 \delta(i, j)\end{aligned} \qquad (5)$$

where

$$\delta(i, j) = \begin{cases} 1, & if \quad i = j \\ 0, & otherwise \end{cases}.$$

$\mathbf{\Theta} = \left(v_0, \{w_l\}_{l=1}^{d}, a_0, a_1, \sigma_v^2\right)$ defines the hyper-parameter vector. The covariance function consists of the following components, an exponential term ($v_0 \exp\left\{-\frac{1}{2} \cdot \sum_{l=1}^{d} w_l \left(x_l^{(i)} - x_l^{(j)}\right)^2\right\}$), a bias term ($a_0$), a linear term ($a_1 \sum_{l=1}^{d} x_l^{(i)} x_l^{(j)}$) and a jitter term ($\sigma_v^2 \delta(i, j)$). The exponential element expresses the hypothesis that cases with nearby inputs will have highly correlated outputs. The length scale, w_l, according to each input characterises the distance in that particular direction over which y is expected to significantly vary; and v_0 defines the vertical scale of variations of a typical function. The bias term adds a constant component to the regression function. The linear term expresses the linear trend of the function and finally the jitter term accounts for the noise in the data, where σ_v^2 is the variance of the noise. In practice, the covariance function can be defined as any combination of these four terms, according to the different properties of the function $f(\cdot)$.

2.2 MCMC Implementation

According to the Bayesian approach, the training process for the Gaussian process model commences with the definition of a prior distribution $P(\mathbf{\Theta})$ over the hyper-parameter space that is updated using the training data, D, to produce a posterior distribution $P(\boldsymbol{\Theta} \mid D)$:

$$p(\boldsymbol{\Theta} \mid D) \propto p(\boldsymbol{\Theta})\,p(D \mid \boldsymbol{\Theta}) \qquad (6)$$

where $p(D \mid \boldsymbol{\Theta})$ is the likelihood of the training data for a given $\boldsymbol{\Theta}$. Here the log-likelihood is used to avoid large rounding errors:

$$L = -\frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\mathbf{y}^T\boldsymbol{\Sigma}^{-1}\mathbf{y} - \frac{N}{2}\log 2\pi \qquad (7)$$

This procedure requires the computation of the inversion of the covariance matrix $\boldsymbol{\Sigma}$ of order $N \times N$, which takes time, $O(N^3)$. This means that the implementation time increases cubically as the size of the training data set increases, and hence problems may result when dealing with large data sets.

To make predictions, integration over the posterior distribution of the hyper-parameters, $\boldsymbol{\Theta}$, is performed:

$$\bar{y}(\mathbf{x}) = \int \hat{y}_{\boldsymbol{\Theta}}(\mathbf{x})P(\boldsymbol{\Theta} \mid D)d\boldsymbol{\Theta} \qquad (8)$$

$$\bar{\sigma}_y^2(\mathbf{x}) = \int \hat{\sigma}_{\hat{y}_{\boldsymbol{\Theta}}}^2(\mathbf{x})P(\boldsymbol{\Theta} \mid D)d\boldsymbol{\Theta} \qquad (9)$$

Because of the difficulty of performing this integration analytically, Markov Chain Monte Carlo (MCMC) is used to generate a series of samples for $\boldsymbol{\Theta}$ according to the posterior distribution, equation (6). The predictions are then calculated by averaging over the MCMC samples of $\boldsymbol{\Theta}$. Details of MCMC are given in Robert and Casell, (1999).

Priors for $\boldsymbol{\Theta}$. The priors for $\boldsymbol{\Theta}$ are defined as given by Rasmussen (1996) whilst the priors for $\{w_l\}$ are given by inverse Gamma distributions:

$$w_l^{-1} \sim Gamma\left(\frac{\alpha}{2}, \frac{\alpha}{2u}\right) \text{ with } u = u_0 d^{\,2/\alpha} \qquad (10)$$

where u is the mean of w_l^{-1}. A small value of α results in a vague prior. $\alpha = 1$ and $u_0 = 1$ were selected. The priors for $\log(v_0)$, $\log(a_0)$ and $\log(\sigma_v^2)$ are all Gaussian distributions with large variances corresponding to fairly vague priors, i.e. $\log(v_0) \sim G(-3, 3^2)$, $\log(a_0) \sim G(-3, 3^2)$ and $\log(\sigma_v^2\, v_0) \sim G(-1, 1^2)$.

The MCMC method. The Metropolis-Hastings algorithm is used to address the relationship in equation (6), which uses a candidate distribution $q(\boldsymbol{\Theta}, \boldsymbol{\Theta}')$ from which to sample, $p(\boldsymbol{\Theta} \mid D)$. The algorithm can be described as follows:

```
1. Initialization Step:
   Set $i = 0$ and generate a random value for $\boldsymbol{\Theta}^{(0)}$;
2. Iteration:
   For $i = 1:K$
      Sample a candidate $\boldsymbol{\Theta} \sim q(\boldsymbol{\Theta}^{(i-1)}, \cdot)$;
      Evaluate the acceptance probability $\alpha(\boldsymbol{\Theta}^{(i-1)}, \boldsymbol{\Theta})$:

      $\min\left\{ \frac{p(\boldsymbol{\Theta} \mid D) / q(\boldsymbol{\Theta}^{(i-1)}, \boldsymbol{\Theta})}{p(\boldsymbol{\Theta}^{(i-1)} \mid D) / q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(i-1)})}, 1 \right\}$      (11)

      Sample a uniformly distributed $u$ as $u \sim U[0,1]$, if $u \le \alpha(\boldsymbol{\Theta}^{(i-1)}, \boldsymbol{\Theta})$, then
      $\boldsymbol{\Theta}^{(i)} = \boldsymbol{\Theta}$, otherwise, $\boldsymbol{\Theta}^{(i)} = \boldsymbol{\Theta}^{(i-1)}$.
   End
```

In this study, $q(\boldsymbol{\Theta}, \boldsymbol{\Theta}')$ is set as the transition distribution of a random walk with a small perturbation, $\mathbf{z}$. A normally distributed $\mathbf{z}$ with a zero mean vector implies that $q(\boldsymbol{\Theta}, \boldsymbol{\Theta}') = q(\boldsymbol{\Theta}', \boldsymbol{\Theta})$. Then the acceptance probability becomes:

$$\begin{aligned} \alpha(\boldsymbol{\Theta}, \boldsymbol{\Theta}') &= \min\left\{ \frac{p(\boldsymbol{\Theta}' \mid D)}{p(\boldsymbol{\Theta} \mid D)}, 1 \right\} \\ &= \min\left\{ \frac{p(D \mid \boldsymbol{\Theta}')\,p(\boldsymbol{\Theta}')}{p(D \mid \boldsymbol{\Theta})\,p(\boldsymbol{\Theta})}, 1 \right\} \end{aligned} \qquad (12)$$

A simple slowly decreasing variance for $\mathbf{z}$ is selected although more elaborate strategies, such as simulated annealing, could be applied (e.g. Robert and Casella, 1999). Assuming the initial variance for $\mathbf{z}$ is $\sigma_z^2(0)$, at step i, the variance $\sigma_z^2(i)$ can be defined as:

$$\sigma_z^2(i) = \begin{cases} \sigma_z^2(0) & \textit{if} \quad i \le T \\ \sigma_z^2(0) / (i - T)^{0.1} & \textit{otherwize} \end{cases} \qquad (13)$$

where $T < K$ is the start point for σ_z^2 to decrease.

After generating K samples of the hyper-parameters $\boldsymbol{\Theta}$, the predictions can be made by taking an average over the samples:

$$\bar{y}(\mathbf{x}) = \frac{1}{K}\sum_{i=1}^{K} \hat{y}_{\boldsymbol{\Theta}}^i(\mathbf{x}) \qquad (14)$$

where $\hat{y}_{\boldsymbol{\Theta}}^i(\mathbf{x})$ is given by equation (2) for a particular sample $\boldsymbol{\Theta}^{(i)}$. The variance of the prediction can then be calculated in a similar way:

$$\bar{\sigma}_{\hat{y}}^2(\mathbf{x}) = \frac{1}{K}\sum_{i=1}^{K} \left(\hat{\sigma}_{\hat{y}}^{i\,2}(\mathbf{x}) \right) \qquad (15)$$

where $\hat{\sigma}_{\hat{y}}^{i\,2}(\mathbf{x})$ is given by equation (3).

2.3 Model for Processes with Multi-outputs.

The theory described holds for scalar outputs. For the modelling of processes with multiple outputs, although it is possible to define Gaussian processes with multi-outputs, it is not clear how the covariance function should be defined. Therefore, a separate Gaussian process model is established for each output to simplify the problem. However, since multiple outputs can share some common features, it may be advantageous to build this correlation structure into the analysis. This new approach is currently under investigation.

3. EXPERIMENTS AND RESULTS

3.1 The Fed-batch Bioreactor

The process used in the study is a benchmark fed-batch bio-reactor for the production of secreted protein (Park and Ramirez, 1988). They considered the secretion of foreign protein using Baker's yeast as the host organism. The mechanistic model that describes the dynamics of the process can be expressed by the following differential equations:

$$\dot{P}_M = A(S)(P_T - P_M) - \frac{q}{V}P_M \qquad (16)$$

$$\dot{P}_T = B(S)X - \frac{q}{V}P_T \qquad (17)$$

$$\dot{X} = C(S)X - \frac{q}{V}X \qquad (18)$$

$$\dot{S} = -YC(S)X + \frac{q}{V}(m - S) \qquad (19)$$

$$\dot{V} = q \qquad (20)$$

where

$$A(S) = \Phi(\mu_x) = \frac{4.75C(S)}{0.12 + C(S)} \qquad (21)$$

$$B(S) = f_P(S) = \frac{Se^{-5.0S}}{0.1 + S} \qquad (22)$$

$$C(S) = \mu_x(S) = \frac{21.78S}{(S + 0.4)(S + 62.5)} \qquad (23)$$

The state variables are the amount of secreted protein on a unit culture volume basis (P_M), the total protein amount on a unit volume basis (P_T), the culture cell density (X), the culture glucose concentration (S), and the culture volume (V). q is the feed flow rate which is used as the control variable, m is the glucose concentration of the feed stream, Y is the yield of glucose per cell mass and Φ , f_P , and μ_x are the protein secretion rate, the protein expression rate and the specific growth rate of the host cell, respectively.

Based on the above model, the performance of Gaussian process modelling of the bioreactor was investigated. The initial conditions of the state variables and parameters in the model were given by:

$$P_M(t_0) = 0.0 \qquad V(t_0) = 1.0\,\text{L}$$

$$P_T(t_0) = 0.0 \qquad m = 20.0\,\text{g/L}$$

$$X(t_0) = 1.0\,\text{g/L} \qquad Y = 7.3$$

$$S(t_0) = 5.0\,\text{g/L}$$

Data for 10 batches, each of 15 hours duration were generated. The data comprised the following simulated measurements, P_M, S , V , and q . It is assumed that the measurements of V and q are obtained on-line while the measurements of P_M and S are recorded off-line. This reflects current industrial practice where many bio-product quality variables are measured off-line through laboratory analysis. The data were generated by adding normally distributed random noise with zero mean to the nominal control policies (q) as reported in Tian, *et al.* (2002). Zero mean normal random noise was also added to all the measurements to simulate the effects of measurement noise. The standard deviations of the noise for q , P_M, S and V , were 0.02L/hr, 0.01, 0.02g/L and 0.05L respectively. The sampling time is 6 min, i.e. 150 samples per batch.

3.2 Simulation Study.

There are two states of interest to be predicted, the amount of secreted protein P_M and the culture glucose concentration S , which cannot be measured on-line. As described in Section 2.3, separate Gaussian models were constructed for each state, *GP_S* and *GP_ P_M*. The inputs for each model comprised the feed flow rate *q*, and the predicted value of *V* ($\hat{V}$) using:

$$\hat{V}(t) = \hat{V}(t-1) + q(t-1)\Delta t, \qquad \Delta t = 6\,\text{min} \qquad (24)$$

and the previous predictions from the model (denoted $\hat{S}$ and $\hat{P}_M$), so that the model can make long-range predictions for the whole batch given the initial measurement of *V* and the initial values for S and P_M . As the number of inputs is related to the dimension of the hyper-parameter $\mathbf{\Theta}$, it is preferred that fewer input variables are included in the model to reduce computation time. Also it was observed that prediction performance was not compromised. Thus the model inputs are selected as follows. For *GP_S*, the inputs are:

$$q(t-1), q(t-2), \hat{V}(t-1), \hat{S}(t-1)\,.$$

The culture glucose concentration S may have an impact on P_M , consequently two models were developed to predict P_M. $GP\_\ P_M\_1$ does not include information on S:

$$q(t-1), q(t-2), \hat{V}(t-1), \hat{P}_M(t-1)\ ,$$

whilst $GP\_\ P_M\_2$ includes the previous predictions of S from GP_S:

$$q(t-1), q(t-2), \hat{V}(t-1), \hat{S}(t-1), \hat{P}_M(t-1)\ .$$

For the fed-batch process, the regression model is based on a smooth, continuous function, thus the covariance function is chosen as a simple form without the linear term:

$$\Sigma(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = v_0 \exp\left\{-\frac{1}{2}\sum_{l=1}^{d} w_l \left(x_l^{(i)} - x_l^{(j)}\right)^2\right\} \qquad (25)$$
$$+ a_0 + \sigma_v^2 \delta(i, j)\ .$$

and thus $\mathbf{\Theta}$ is $\left(v_0, \{w_l\}_{l=1}^{d}, a_0, \sigma_v^2\right)$. During the training stage, 1200 iterations were run for the MCMC sampling of $\mathbf{\Theta}$. The algorithm converged quickly and tended to stabilise after about 200 iterations according to the values of the log-likelihood. These initial 200 values were discarded. One sample of $\mathbf{\Theta}$ was then selected from each consecutive set of 20 iterations resulting in a total of 50 samples being selected to perform the prediction. These samples can be approximately treated as independent and identically distributed, according to the relevant posterior distribution.

3.3 Results and Discussions.

Five batches were used to train the model with a further 5 (batches 6-10) being used for model validation. The initial state values S and P_M were set to the same values as in the simulation, i.e., $S_0 = 5.0$ g/L and $P_{M0} = 0.0$. Using the samples of $\mathbf{\Theta}$ obtained from the model training and the given initial states, the predictions for the whole batch were calculated using equations (2), (3), (14) and (15). The performance of the models was evaluated using the average Root Mean Square Error (RMSE) over the five validation batches, Table 1.

Table 1 Results for the Prediction of S and P_M

Model Type	Input variables	RMSE
GP_S	$q(t-1), q(t-2), \hat{V}(t-1),$ $\hat{S}(t-1)$	0.1218
$GP\_P_M\_1$	$q(t-1), q(t-2), \hat{V}(t-1),$ $\hat{P}_M(t-1)$	0.0435
$GP\_P_M\_2$	$q(t-1), q(t-2), \hat{V}(t-1),$ $\hat{S}(t-1), \hat{P}_M(t-1)$	0.0384

The results show that the models exhibit quite good performance with the RMSEs being comparable with those obtained by the augmented recurrent neural network models proposed in Tian, *et al.* (2002). It is also observed that the RMSE of model $GP\_P_M\_2$ is smaller than for model $GP\_P_M\_1$, which implies that S has a direct effect on P_M and more accurate predictions of P_M can be obtained by including S as an additional model input.

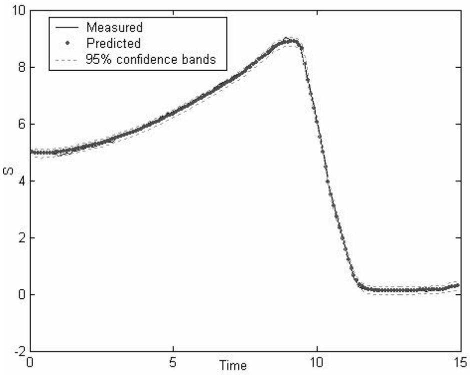

Fig. 1. Multi-step-ahead predictions for S on validation batch 9 (GP_S)

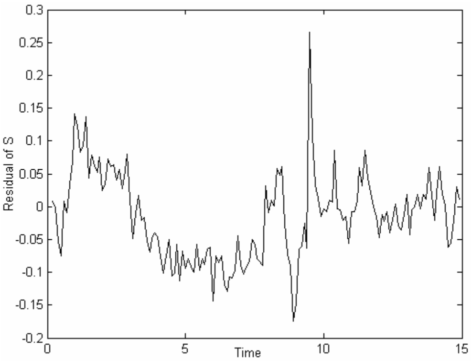

Fig. 2. Multi-step-ahead prediction residuals of S on validation batch 9 (GP_S)

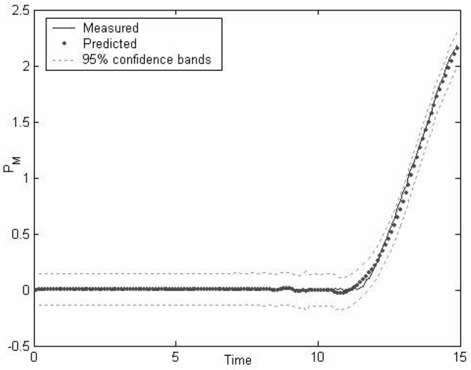

Fig. 3. Multi-step-ahead predictions for P_M on validation batch 9 ($GP\_P_M\_1$)

To further assess the performance of the model, the predictions with 95% confidence bands of each model (without considering the uncertainty of **x**) are compared with the actual output values. Long range predictions of glucose concentration (S) for one of the validation data set (batch 9) and the corresponding residuals are plotted in Fig. 1 and Fig. 2. Fig. 3 - Fig. 5 show the long range predictions and the corresponding residuals of secreted protein (P_M) for the cases where the models do not have any information on S and where information on S is included.

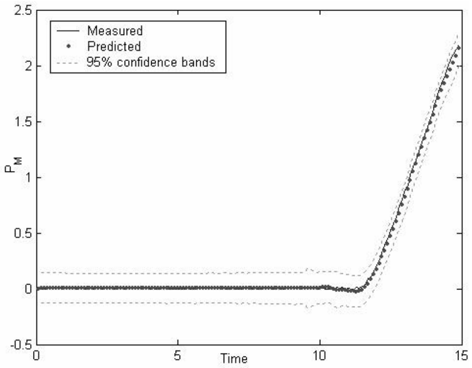

Fig. 4. Multi-step-ahead predictions for P_M on validation batch 9 (*GP_P_M_2*)

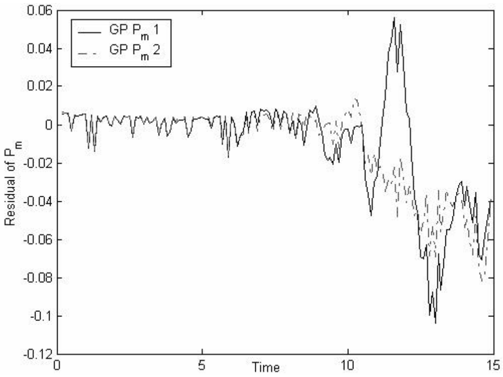

Fig. 5. Multi-step-ahead prediction residuals of P_M on validation batch 9 (*GP_P_M_1* and *GP_P_M_2*)

It can be seen that the predictions are satisfactory. It is also apparent that the predictions of *GP_P_M_2* are more accurate than those of *GP_P_M_1*, especially around 10hr30min to 12hr where the secreted protein is beginning to increase sharply.

4. CONLUSIONS AND DISCUSSION

This paper has proposed the use of Gaussian process regression for batch process modelling. Gaussian processes are non-parametric Bayesian regression models which are sufficiently flexible to represent a wide variety of data by defining an appropriate covariance matrix. Hyper-parameters, which determine the covariance functions can be estimated by Markov chain Monte Carlo simulation. The model can be implemented using matrix computations that are feasible for datasets comprising over a thousand samples. The effectiveness of the Gaussian process approach has been demonstrated by its application to the modelling of a fed-batch bioreactor.

Currently the research is aimed at extending Gaussian process approaches to mixtures of Gaussians (Shi, *et al*., 2003) to account for heterogeneity between batches. A useful aspect of Gaussian process mixtures is that they can dramatically reduce the computational cost by splitting the training data. Conventional mixtures are defined according to different batches. The extension to Gaussian process mixtures for data monitored across batches is also on-going.

5. ACKNOWLEDGEMENTS

X. Ou acknowledges the financial support from CPACT and Newcastle University, for her PhD. The authors would also like to thank Dr. Jianqing Shi for discussions relating to Gaussian processes.

REFERENCES

Brooks, S. (1998). Markov Chain Monte Carlo Method and Its Application, *The Statistician*, **47**, pp. 69-100.

Gelman, A., J.B. Carlin, H.S. Stern and D.B. Rubin (2000). *Bayesian Data Analysis*, Chapman & Hall/CRC.

Neal, R.M. (1997). *Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification*, Technical Report No. 9702, Dept. of Statistics, University of Toronto, available from http://www.cs.toronto.edu/~radford/publications.html.

Nounou M.N., B.R. Bakshi, P.K. Goel and X.T. Shen (2002). Bayesian Principal Component Analysis, *Journal of Chemometrics*, **16**, pp. 576-595.

Kourti T. (2003). Multivariate Dynamic Data Modelling for Analysis and Statistical Process Control of Batch Processes, Start-ups and Grade Transitions, *Journal of Chemometrics*, **17**, pp. 93-109

Park, S. and W.F. Ramirez (1988). Optimal Production of Secreted Protein Fed-Batch Reactors, *A.I.Ch.E Journal*, **39**, pp. 1550-1558.

Robert, C. and G. Casella (1999). *Monte Carlo Statistical Methods*, Springer, New York

Rasmussen, C.E. (1996). Evaluation of Gaussian Process and Other Methods for Non-linear Regression, PhD Thesis, University of Toronto, available from http://www.cs.toronto.edu/0~carl/gp.html.

Shi, J.Q., R. Murray-Smith and D.M. Titterington (2003), Bayesian Regression and Classification Using Mixtures of Gaussian Processes, *International Journal of Adaptive Control and Signal Processing*, **17**, pp.1-16.

Tian, Y., J. Zhang and A.J. Morris (2002). Optimal Control of A Fed-Batch Bioreactor Based upon An Augmented Recurrent Neural Network Model, *Neurocomputing*, **48**, pp. 919-936.

West, M. and J.F. Harrison (1997). *Bayesian Forecasting and Dynamic Models*, Springer Series in Statistics, 2nd edition.

Williams, C.K.I. and C.E. Rasmussen (1996). Gaussian processes for regression, *Neural Information Processing Systems*, **8**, pp. 514-520.