ELSEVIER

Journal of Biotechnology 69 (1999) 47–62

JOURNAL OF Biotechnology

Rolling learning-prediction of product formation in bioprocesses

J.Q. Yuan [a],*, P.A. Vanrolleghem [b]

[a] *East China University of Science and Technology, PO Box 303, 130 Meilong Lu, 200237 Shanghai, People's Republic of China*
[b] *BIOMATH Department, University Gent, Coupure Links 653, B-9000 Gent, Belgium*

Received 16 February 1998; received in revised form 4 January 1999; accepted 12 January 1999

Abstract

A rolling learning-prediction approach based on neural networks is proposed with the aim of on-line prediction of the product formation. Commercial-scale penicillin cultivations were taken as an example to test the product predictor. Raw data are pretreated in such a way that each input vector of the neural network consists of a series of time-discretised values on a specified transient of process variables. The output vector is composed of the amount of product at the next one and two prediction steps. The process variables involved in the predictor include carbon dioxide and product formation as well as oxygen, precursor and substrate consumption. Accumulated rather than instant values of these variables were used. A simple three-layer feedforward backpropagation neural network with a tangent sigmoidal transfer function in the hidden nodes and a linear one in the output nodes was used as the main frame of the product predictor. The proposed prediction procedure is called rolling learning-prediction because the training database is updated after each sampling interval and the learning-prediction is repeated thereafter. The robustness of the predictor was illustrated by its adaptive ability to widely scattered data sets and extra added noises. The testing results indicated that a prediction accuracy of 2–5% could be generally expected in the later phase of cultivation and reliable prediction time spans may take more than 10% of the cultivation period for penicillin production. An intrinsic problem of using neural networks—occasional trap of the network in bad local minima—is automatically detected and remedied. In addition, it was illustrated by example that the prediction error signal may be potentially used to detect extraordinary charges caused, for example, by contamination. Problems associated with the industrial application of the predictor are discussed. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Rolling learning-prediction; Neural network; Product prediction; Penicillin

1. Introduction

Artificial neural networks (ANNs) have been widely studied in the past decade because of their powerful input–output data mapping ability for

* Corresponding author. Tel.: +86-21-6425-3002; fax: +86-21-6425-3904.
E-mail addresses: bpc@ecust.edu.cn (J.Q. Yuan), peter.vanrolleghem@rug.ac.be (P.A. Vanrolleghem)

PII: S0168-1656(99)00002-4

nonlinear systems (Rumelhart et al., 1986; Cybenko, 1989; Bhat and McAvoy, 1990; Leonard and Kramer, 1990). Many encouraging results have been obtained by applying ANNs to bioprocess state estimation (Thibault et al., 1990; Karim and Rivera 1992), modeling (Psichogios and Ungar, 1992; van Can et al., 1997), pattern recognition and control (Raju and Cooney, 1992; Aynsley et al., 1993; Schubert et al., 1994). A promising prospect of neural networks has been shown by some industrial application oriented investigations. For instance, Montague and Morris (1994) applied neural network models in biomass prediction and fault diagnosis for the penicillin production operated by SmithKline Beecham (Irvine, UK). Linko et al. (1995) successfully applied a dynamic neural network to predict product formation and substrate consumption for commercial lysine production.

The purpose of the work presented here is to develop an on-line application oriented ANN-product predictor, which may satisfy the requirement of high prediction accuracy, strong robustness and relatively large prediction time span. The predictor will be potentially used in optimal scheduling for a multi-reactor plant (Yuan et al., 1997). A high prediction accuracy during the later phase of cultivation is especially focused upon because production scheduling is usually activated in this phase. In fact, during the earlier phases of cultivation, bioprocesses may exhibit highly individual behaviors caused by various factors, e.g. fluctuations of the seed quality. The process monitoring and control during this phase is mainly to keep the routine feeding profiles and try to recognize possible extraordinary charges. The optimal control strategy, no matter whether it is model-based or just on the basis of statistical analysis, can be usually carried out only after the process has passed approximately one-third of its cycle time, when the minimum amount of necessary data for process evaluation becomes available. For the purpose of optimal scheduling, the prediction accuracy during the later phase of cultivation should be better than 5% since the minimal process fluctuation is usually around 10%.

Two types of neural networks which have been intensively investigated, i.e. feedforward back-propagation neural networks (FBNNs) and recurrent neural networks (RNNs), may be chosen to solve the prediction problem described. Fig. 1 schematically shows these two types of neural networks. For the sake of simplicity, only three-layer networks with three input nodes, two hidden nodes and two output nodes are presented (in practical applications, the number of the input and output nodes are usually determined by the process under consideration, while the number of hidden neurons is dependent on the complexity of the problem to be solved). Generally, a RNN is different from a FBNN in that connections are allowed both ways between a pair of neurons and even from a neuron to itself. Fig. 1(b) presents only a simple architecture of RNNs in which the prior output of the hidden units (one-step delayed) is fed back to the hidden nodes on each successive calculation cycle. This specific architecture is often referred to as Elman network after its originator (Elman, 1990). Su and McAvoy (1992) have illustrated, with a waste water treatment plant as an example, that the feedforward back-propagation neural network is well suited for short-term prediction but the recurrent neural network is more powerful for long-term prediction. The long-term prediction ability of a RNN lies on

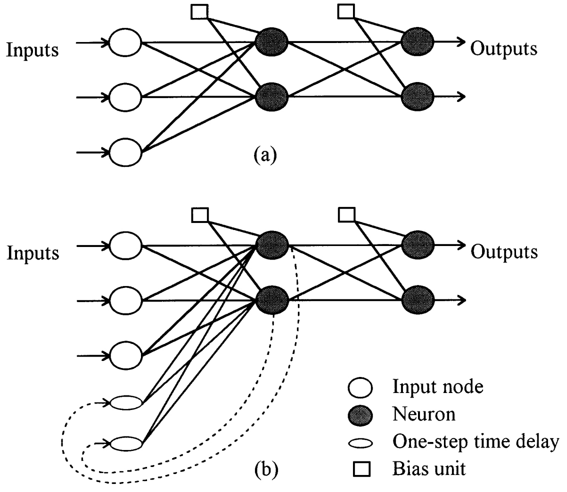

Fig. 1. Most commonly applied neural networks. (a) Standard feedforward network; (b) a simple recurrent network.

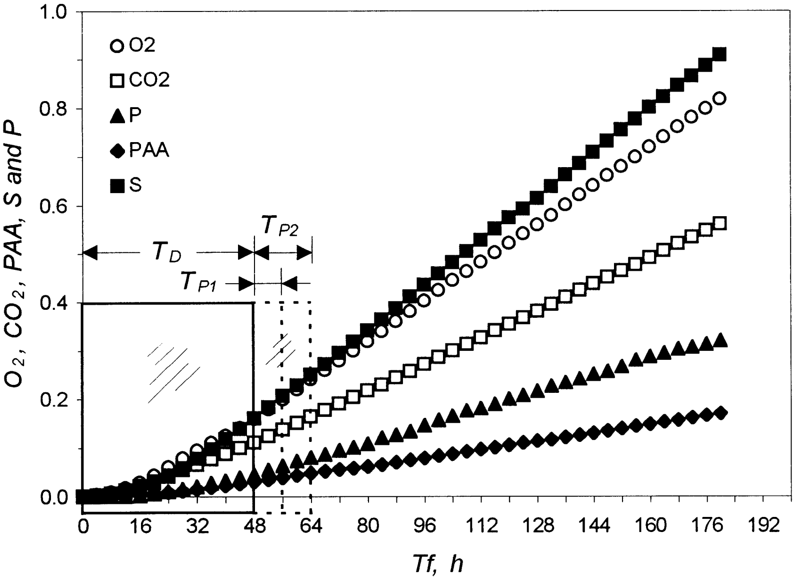

Fig. 2. Time course of some process variables found in a commercial penicillin cultivation and the first input and output data window.

the fact that the feedback paths in a RNN allow the network to learn to recognize and generate both spatial and temporal patterns so that the process dynamics can be accounted for (Karim et al., 1997). Despite the large and growing body of work in the training and use of RNNs (Williams and Ziper, 1989; Williams and Peng, 1990; Su and McAvoy, 1992; Karjala and Himmelblau, 1994), the feedforward backpropagation neural network was chosen for the present study because a short-term product prediction is essentially focused upon in this paper.

For neural network training, data can come from both historical charges and the data collected so far during the charge of present interest. Variables which are not routinely available, such as biomass concentration in mycelia cultivations, must be excluded. The incorporation of data of the present charge into the training database is most important in order to obtain a highly accurate and robust predictor, since these data contain individual characteristics of the present charge which may not have occurred in the historical charges. Such individual characteristics are usually caused by inherent quality fluctuations of precultures, composition changes of substrates and other unmeasured disturbances during the earlier phase of cultivation. The proposed prediction procedure is called rolling learning-prediction because the training database is extended as the process progresses and the learning-prediction is iteratively repeated every time the database is updated with the analysis results of the latest sample.

Taking penicillin fermentation as an example, the establishment of the training database, the rolling learning-prediction for product formation and its error analysis, testing the robustness of the predictor, the potential use of the prediction error signal in fault diagnosis as well as automatic detection and remediation of eventual malfunction will be demonstrated in the following sections. Determination of the database size as well as problems associated with industrial implementation of the approach will be discussed.

2. Establishment of training database

Fig. 2 shows the time course of the main process variables (accumulated values) for penicillin production. These data come from a commercial production charge in a Chinese pharmaceutical factory. For confidentiality reasons, the data (as well as the data used in the following text) have been normalized. In the following discussions, a constant sampling time interval T_S is adopted ($T_S = 4$ h in Fig. 2). Two data windows may be found in Fig. 2, i.e. an input data window with solid frame and an output data window (or prediction window) with dotted frame. The width of the input data window is T_D ($T_D = 48$ h in Fig. 2) while that of the output window is T_P. For a two-step prediction, T_P is equal to T_{P1} for the first step and T_{P2} for the second step (in Fig. 2, $T_{P1} = 8$ h and $T_{P2} = 16$ h).

The database is defined as the set of input–output data pairs. Each individual data pair is obtained using a moving data windows technique. Both input and output data windows move along the time scale with a fixed moving step T_M. By discretising the transients of process variables covered by each data window, one obtains a series of input–output data pairs—elements of the database. The input–output data pair corresponding to the kth data window $\{X(T_k), Y(\mathrm{T_k})\}$ is given by Eqs. (1)–(3).

$$X(T_k) = \begin{bmatrix} T_k \\ x(T_k) \\ x(T_k\text{-}1\tau) \\ x(T_k\text{-}2\tau) \\ \vdots \\ x(T_k\text{-}m\tau) \end{bmatrix} \tag{1}$$

$$x(T_k) = [\mathrm{O_2}(T_k)\ \mathrm{CO_2}(T_k)\ P(T_k)\ \mathrm{PAA}(T_k)\ \mathrm{S}(T_k)\ \mathrm{Nit}(T_k)\ \mathrm{Temp}(T_k)\ p\mathrm{O_2}(T_k)\ \mathrm{pH}(T_k)\ \ldots\ldots]^T \tag{2}$$

$$Y(T_k) = [P(T_k + T_{P1})\ P(T_k + T_{P2})]^T \tag{3}$$

where T_k is the cultivation time at the right border of the input data window so that we have $T_1 = T_D$, τ is the discretisation time interval for process variables covered by the input data window, and m is the discretisation step length which equals T_D/τ. The output data vector is composed of the amount of product at the next one and two steps, respectively. The meaning and units of other symbols are presented in Appendix A.

For a historical charge with a cultivation period T_f, the number of input–output data pairs N is readily calculated by:

$$N = \mathrm{int}\left(\frac{T_f - T_D - T_{P2}}{T_M}\right) \tag{4}$$

The training database for rolling learning-prediction, θ, is expressed by Eq. (5). It contains two parts: the set of all input–output data pairs of n historical charges, $\theta_{1\sim n}$, and all input–output data pairs available at the moment of prediction for the $(n+1)$th charge (i.e. the charge of present interest), θ_{n+1}, see Eqs. (6) and (7). The subscript i in Eq. (6) represents the charge number and N_i the number of input–output data pairs of the ith historical charge. Suppose the most recent measurement for the $(n+1)$th charge is at T_k, then in the case of $T_{P1} = 8$ h, $T_{P2} = 16$ h and $T_M = 4$ h, θ_{n+1} may be expressed by Eq. (7), where only the input–output data pairs up to T_{k-4} are available. The output data pairs for $X_{n+1}(T_{k-3}) \sim X_{n+1}(T_k)$ do not exist since the future measurements are not yet available. Rather, they will be predicted.

$$\theta = \{\theta_{1\sim n}\theta_{n+1}\} \tag{5}$$

$$\theta_{1-n} = \{X_i(T_k),\ Y_i(T_k)\}$$

$$k = 1, 2, \ldots, N_i, \qquad i = 1, 2, \ldots, n \tag{6}$$

$$\theta_{n+1} = \{X_{n+1}(T_1),\ Y_{n+1}(T_1);\ X_{n+1}(T_2),\ Y_{n+1}(T_2)\ ;\ \ldots;\ X_{n+1}(T_{k-4}),\ Y_{n+1}(T_{k-4})\} \tag{7}$$

The first product prediction for the $(n+1)$th charge can be made only when its cultivation time $T_{f\ n+1}$ has surpassed T_D so that the first input vector is complete. On the other hand, on-line updating of the training database θ by adding the input–output data pairs of the $(n+1)$th charge only begins when $T_{f\ n+1}$ has become larger than $(T_D + T_{P2})$.

3. Rolling learning-prediction of product formation

Fig. 3 schematically shows how the ANN-based product predictor works. The initial state is referred to as the initial values of medium volume and concentrations of sugar, precursor and peni-

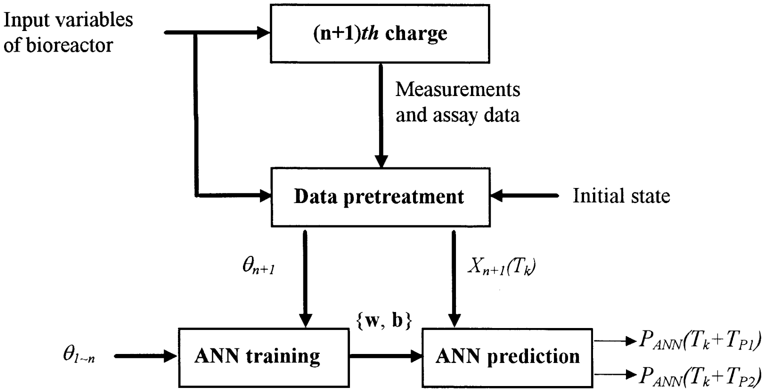

Fig. 3. Schematic description of the rolling learning-prediction mechanism.

cillin. The input variables of the bioreactor include aeration rate, which will be used for calculation of oxygen consumption and carbon dioxide evolution, feeding rates of carbon source, precursor, ammonia and ammonium sulfate solutions and relevant concentrations. In practical operation, a complex substrate is used which is a ropy solution of glucose, hydrolyzed corn mash, soybean cake powder and so on. However, in this paper, substrate only refers to the total reducible sugar. The other input information to the data pretreatment block, measurements and assay data, refers to medium volume of the bioreactor, flow rate of withdrawal, oxygen and carbon dioxide content in waste gas as well as the sampling analysis results of substrate(s), precursor, product (it may also include some byproducts for other bioprocesses) concentration in the medium. All these data are essential for making mass balances so that at any sampling time, one knows the outcome of the accumulated values of the most important process variables, e.g. how much product and carbon dioxide has been produced and how much sugar, precursor and oxygen has been consumed (as indicated in Fig. 2). The accumulated process variables resulting from the different mass balances have different units and order of magnitude. For example, the penicillin produced by a charge is typically several thousand kilograms, but the corresponding sugar consumption may be as high as 20 000 kg. As usual, when applying ANNs, all process variables as well as fermentation time are scaled to vary between 0 and 0.9. This is done by dividing the value of a process variable by 1.3 times its relevant maximal value.

The input and output data vectors of the ANN are then generated by the data pretreatment block with the principles described in the previous section. Based on the step-by-step updated training database $\{\theta_{1\sim n}\ \theta_{n+1}\}$, the ANN training block identifies the weighting factors and biases {**w**, **b**} by repeated learning. The latest input data vector $X_{n+1}(T_k)$ (as already stated, it does not have a corresponding measured output data pair) is fed as input to the ANN prediction block so that the prediction of the total product at time (T_k+T_{P1}) and (T_k+T_{P2}) is obtained. The whole learning-prediction procedure is therefore characterized by the use of data of both historical charges $(\theta_{1\sim n})$ and data of the present charge acquired up to time T_k (θ_{n+1}). The prediction is knowledge-based because of the learning ability of ANN and the rich information content contained in the training database.

For the neural network training, the Levenberg–Marquart optimization (a modified Gauss–Newton method) was applied (Demuth and Beale, 1994). For one- and two-step predictions, the error goal of the network training is to minimize the sum of square errors *ssv*, represented by:

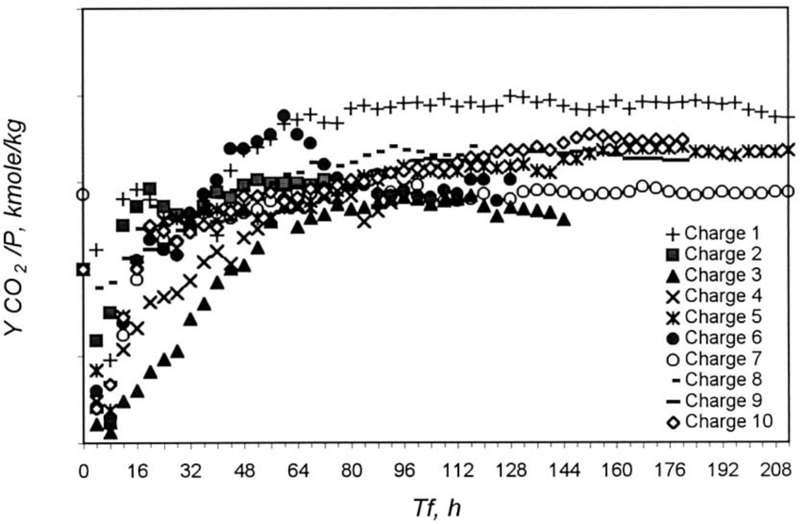

Fig. 4. CO_2 production over penicillin formation of 10 commercial charges. Accumulation values of the two variables are used.

$$ssv = \sum_{k=1}^{N(\theta)} \{(P_{\mathrm{M}}(T_k + T_{\mathrm{P1}}) - P_{\mathrm{ANN}}(T_k + T_{\mathrm{P1}}))^2 + (P_{\mathrm{M}}(T_k + T_{\mathrm{P2}}) - P_{\mathrm{ANN}}(T_k + T_{\mathrm{P2}}))^2\}$$

where $N(\theta)$ is the number of the input–output data pairs in the training database. It is noticed that the error signals are equally weighted in *ssv*.

4. Results

Ten commercial fed-batch cultivations were used for training and testing. They were carried out in continuous stirred fermenters with a volume exceeding 100 m^3. Penicillin G was the product of interest. Complex substrate was quasi-continuously fed into the fermenter by using a gauging-cup technique. The feeding rate was adjusted by changing the frequency of cup emptying. Ammonia and ammonium sulfate solutions were fed into the fermenter proportionally to the substrate feeding. The feeding rate of the precursor solution was manually controlled with the aim of keeping the phenyl acetic acid concentration at a predetermined low level because of the known inhibitory effect of the precursor to micro-organisms. These 10 charges behaved very differently, as illustrated in Fig. 4, showing a large variation in the ratio of carbon dioxide to product formation (accumulated values of both variables are used). Similarly large fluctuations in product formation, product yield and respiration quotient (*RQ*) were also observed (results not shown). It seems very difficult to fit all these largely scattered charges with a conventional mechanistic model and fixed model parameters, if possible at all.

A three-layer tansig/purelin neural network (i.e. a tangent sigmoidal transfer function for the hidden layer and a linear transfer function for the output layer) was used as the kernel of the predictor. By setting $T_{\mathrm{D}} = 24$ h, $T_{\mathrm{P1}} = 8$ h, $T_{\mathrm{P2}} = 16$ h and $T_{\mathrm{M}} = 4$ h, the number of input–output data pairs available for these 10 charges is 345. Choosing the first five elements in Eq. (2) as input variables (because their measurements are available for the 10 cultivations studied) and setting $m = 3$, $\tau = 8$ h, the number of input nodes (Fig. 1a) becomes 21. The output layer contains two neurons since a one and two steps ahead prediction for product formation is desired. Theoretically, the number of neurons in the hidden layer can be arbitrarily chosen. However, too many hidden neurons may result in large computational efforts for training and possible over-fitting. In our earlier work, it was shown that for penicillin production three hidden neurons could already give satisfactory fitting and prediction results

(Yuan and Vanrolleghem, 1998). Therefore, a 21–3–2 tansig/purelin network was adopted. Accordingly, there are $(21 \times 3 + 3 \times 2) = 69$ weights and $(3 + 2) = 5$ biases to be determined.

Before testing the product predictor, the prediction error should be defined. Denote $P_{\mathrm{M}}(T_k + T_{\mathrm{P}})$ as the measured product corresponding to the kth data window of an arbitrary charge, and $P_{\mathrm{ANN}}(T_k + T_{\mathrm{P}})$ as its ANN prediction, then the relative prediction error $e(T_k + T_{\mathrm{P}})$ is defined as:

$$e(T_k + T_{\mathrm{P}}) = \frac{P_{\mathrm{ANN}}(T_k + T_{\mathrm{P}}) - P_{\mathrm{M}}(T_k + T_{\mathrm{P}})}{P_{\mathrm{M}}(T_k + T_{\mathrm{P}})} \qquad (8)$$

The average of relative prediction errors corresponding to q prediction points is defined as $\bar{e}$:

$$\bar{e} = \sqrt{\frac{\sum_{k=1}^{q} e(T_k + T_{\mathrm{P}})^2}{q}} \qquad (9)$$

Especially for the sake of optimal scheduling, we divide a commercial penicillin cultivation into two parts, i.e. an earlier phase with $T_{\mathrm{f}} \leq 96$ h and a later phase with $T_{\mathrm{f}} > 96$ h. The average of relative prediction errors during the earlier phase is defined as $\bar{e}_1$ and that during the later phase as $\bar{e}_2$.

4.1. Learning accuracy of the neural model

How good could the 21–3–2 backpropagation network fit the 10 commercial charges? To answer this question, we made a self-testing study. At first, the network was trained with all 345 input–output data pairs as the training database. Then the same 345 input data vectors were fed as input to the trained neural network and the self-testing output was produced. The average of the prediction errors (to be more exact, the fitting errors in the case of self-testing) was thereafter calculated and found to be 0.04. In other words, a data fitting accuracy of 4% was reached. Considering the fact that the process has a great intrinsic uncertainty (see Fig. 5) the fitting ability of the simple neural model is excellent.

4.2. Charge-wise prediction

In contrast to rolling learning-prediction, charge-wise prediction means that the input–output data pairs of the testing charge are not incorporated into the training database. In other words, the network is trained by using only $\theta_{1\sim n}$ as database, then simulation is performed to get predicted output vectors for the testing charge corresponding to $X_{n+1}(T_k)$, $k = 1, 2, \ldots, N_{n+1}$. Hence, both training and prediction is carried out once. For the given example, an arbitrary charge out of the 10 available can be chosen as the testing charge, while the other nine charges automatically become the database for neural network training. Table 1 shows the average of the relative prediction errors after charge-wise prediction for Charges 1–10, respectively, where + 8 and + 16 h mean 8 and 16 h ahead prediction, respectively.

It can be found that $\bar{e}_1$ is either about the same or greater than $\bar{e}_2$ (Charges 2 and 4 were terminated at about 100 h so that practically they did not have a later phase). The poor prediction accuracy during the earlier phase of cultivation ($\bar{e}_1$) is mainly intrinsic process uncertainty related. Meanwhile, no significant difference of $\bar{e}_2$ could be found between one- and two-step prediction, a good sign for the applicability of a two-step prediction. However, the $\bar{e}_2$ values of + 8 h prediction for Charges 1 and 5 are as high as 11.1 and

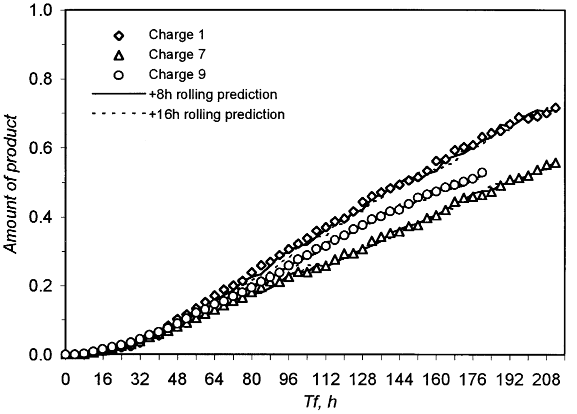

Fig. 5. Comparison between rolling learning-prediction and measurements of product formation for Charges 1, 7 and 9. Symbols are measured data, lines are predictions.

Table 1
Average of relative prediction errors for charge-wise prediction (%)

		Charge number										Mean
		1	2	3	4	5	6	7	8	9	10	
+8 h prediction	$\bar{e}_1$	10.7	8.3	11.2	7.8	7.4	9.2	6.3	5.3	6.5	4.9	7.66
	$\bar{e}_2$	11.1	–	4.4	–	7.8	2.5	3.1	2.3	1.8	1.4	4.17
+16 h prediction	$\bar{e}_1$	14.6	3.7	11.4	9.7	6.1	15.0	5.2	4.2	6.1	9.6	8.56
	$\bar{e}_2$	12.0	–	3.1	–	6.4	6.1	3.2	2.6	2.2	1.4	4.63

7.8%, respectively. Since these two charges look to have normal process characteristics, the charge-wise prediction can therefore not be accepted as a routine prediction procedure.

4.3. Rolling learning-prediction

Table 2 shows the average prediction errors when the rolling learning-prediction procedure described in Fig. 3 is applied. Compared with Table 1, $\bar{e}_1$ and $\bar{e}_2$ were generally improved (see the mean values). Especially, all $\bar{e}_2$ values became less than 5%, both for one-step and two-step predictions. Fig. 5 shows a comparison of the one and two steps ahead predicted product and the measurements for Charges 1, 7 and 9. It can be concluded that the rolling learning-prediction appears to work well.

4.4. Extension of prediction horizon

Because of its high accuracy (Table 2) for +8 and +16 h prediction, one would like to apply the rolling learning-prediction procedure for a multi-step prediction. Simulation was done to test two and three steps ahead prediction (i.e. +16 and +24 h, respectively) using the same rolling learning-prediction procedure but, evidently, a revised database (see Fig. 2). Again, it was found (Table 3), that the $\bar{e}_2$ values were less than 5% in most cases except for Charge 6. Charge 6, however, is actually an abnormal charge which will be dealt with in the next section. Therefore, performance for three steps (up to 24 h or 1 day) ahead prediction is acceptable. Nevertheless, one should be careful if a prediction over three steps is made, because the mean of average prediction errors in Tables 2 and 3 indicates an increasing tendency, which is summarized in Fig. 6. Moreover, the uncertainty of the future working conditions (feeding rates, etc.) increases with the width of the prediction window T_P.

4.5. Qualitative diagnosis of extraordinary charges by using prediction error signals

Charge 6 is an abnormal charge because of its dual growth phases which may be recognized by the second lag phase of product formation occurring at about 68 h as shown in Fig. 7. The neural network is apparently able to adapt to such an abnormal situation as indicated by the good fitting (i.e. self-testing) result. Also, the +8 h rolling learning-prediction tracks the abnormal time course very well after a short-term adaptation. However, the two- and three-step predictions take much longer time to follow this abnormal charge and have much higher prediction errors.

The sustained extraordinarily high prediction errors shown in Fig. 6 can be used for fault diagnosis. By applying the rolling learning-prediction, the relative prediction error for normal charges converges generally along the cultivation time, like the case of Charge 1 in Fig. 8(a). In the case of Charge 6 (see Fig. 8b), however, although the relative prediction errors converge at the beginning of cultivation, as of 64 h, the prediction errors exhibit a divergent tendency. Fault detection can be done according to the characterized description of the prediction errors' divergence.

Basically, there are three symptoms for extraordinary charges:

Table 2
Average of relative prediction errors for one and two steps rolling learning-prediction (%)

		Charge number										Mean
		1	2	3	4	5	6	7	8	9	10	
+8 h prediction	$\bar{e}_1$	7.3	2.9	7.5	7.2	5.1	10.4	4.4	4.0	5.0	5.6	5.94
	$\bar{e}_2$	2.3	–	4.2	–	1.8	2.1	1.6	2.8	2.0	1.6	2.30
+16 h prediction	$\bar{e}_1$	11.8	2.9	6.6	8.1	5.9	15.2	4.9	4.2	5.8	10.9	7.63
	$\bar{e}_2$	2.8	–	4.9	–	2.1	4.6	2.4	2.6	2.2	2.6	3.03

1. The absolute value of the prediction error is increasing constantly.
2. After constant increase, the absolute value of prediction error exceeds a predetermined threshold, e.g. 0.1.
3. The $1 \sim 3$-step predictions' error diverges consistently.

If symptoms (1)–(3) are satisfied simultaneously for a charge (e.g. Charge 6), then an alarm for an eventually abnormal charge will be given. A quantitative description of the fault detection procedure is out of scope of this paper, because it can only be done after more extraordinary charges are examined so as to determine values of some empirical factors for confirming symptoms (1) and (2).

4.6. Influence of measurement noise on the prediction accuracy

The continuously measured variables and laboratory assay data are polluted by measurement noise. Actually, the raw industrial data used in this paper are already noise corrupted. In spite of this, extra Gaussian noise was added to test the robustness of the predictor under a more noisy environment. Noise was added to the sampling time and all process variables in the database as well as in input data vectors. However, it must be understood that when calculating the relative prediction error the measured product $P_M(T_k + T_P)$ (see Eq. (8)) was given the original value not polluted by the extra noises. Adding noise to the sampling time has a practical background since the sampling time interval in commercial production is not strictly equidistant. The average prediction errors were examined with the rolling learning-prediction procedure under natural noise, 5 and 10% additional noise, respectively. Fig. 9 shows the results (only $\bar{e}_2$ was plotted). Keeping Charge 6 in mind as an abnormal charge, it may be concluded from Fig. 9 that the rolling learning-prediction procedure is very tolerant to noise. This is a distinguished characteristic of the ANN predictor over other conventional methods like polynomial extrapolation (the comparison will be given later). In fact, 10% additional noise may correspond to an extremely noisy situation. We recall that Fig. 2 shows the process variables of an industrial charge with natural noise. If 10% extra noise is added, a very different view will emerge (see Fig. 10).

As a comparison, short term prediction of product formation was also made by a linear extrapolation technique in which only the previous measurements of P are needed. The data window for linear extrapolation was chosen as $T_D = 24$, $m = 6$ and $\tau = 4$. The moving step length of the data window was set to $T_M = 4$, the same as the case of rolling learning-prediction. Table 4 shows nine normal charges' mean of $\bar{e}_1$ and $\bar{e}_2$ corresponding to linear extrapolation (linex for abbreviation) and rolling learning-prediction (rolep for abbreviation), respectively. Here, the abnormal Charge 6 is reasonably excluded. It can be found that, if no extra noise is added, $\{\bar{e}_2$-linex$\}$ is comparable with $\{\bar{e}_2$-rolep$\}$ but $\{\bar{e}_1$-linex$\}$ is much higher than $\{\bar{e}_1$-rolep$\}$ which indicates the poor prediction accuracy of linear regression during the earlier phase of cultivation. Then, if extra noise is applied, $\{\bar{e}_2$-linex$\}$ increases rapidly with the intensity of noise. Corresponding to 10% extra noise, $\{\bar{e}_2$-linex$\}$ is as high as 11.2 and 12.4% for + 16 and + 24 h extrapolation, respectively. Evi-

Table 3
Average of relative prediction errors for two- and three-step rolling learning-prediction (%)

		Charge number										Mean
		1	2	3	4	5	6	7	8	9	10	
+16 h prediction	$\bar{e}_1$	12.0	3.4	7.7	10.2	8.4	13.7	6.5	5.1	5.9	12.1	8.50
	$\bar{e}_2$	3.0	–	3.2	–	3.0	5.7	2.2	2.0	2.5	2.2	2.98
+24 h prediction	$\bar{e}_1$	13.9	4.1	8.5	10.1	9.9	16.6	6.7	4.6	4.9	9.8	8.91
	$\bar{e}_2$	4.3	–	4.5	–	3.1	10.8	2.4	1.8	2.3	2.1	3.91

dently, linear extrapolation is noise intolerant. In contrast, {$\bar{e}_2$-rolep} is always below 5% in all cases. Further simulation revealed that higher order polynomial extrapolation cannot improve the average prediction accuracy either. Generally speaking, linear extrapolation can be applied only during the middle phase of cultivation where all process variables behave quasi linearly (see Fig. 2).

4.7. Predictor malfunction detection and remediation

The nonlinear transfer functions in the neural network may introduce many local minima in the error surface (Demuth and Beale, 1994). Although the local minima in our application were found to be very close to the global minimum in most cases, it happened from time to time that the solution was trapped in bad local minima. Evidence for such a local minimum is that the predicted product is far away from its should-be value. We define it as 'malfunction' of the predictor. For off-line training prediction, one can simply restart the training process when malfunction occurs. However, in on-line applications, one cannot compare the prediction with its should-be value since the future measurement is still not available. Here, a malfunction detector was designed.

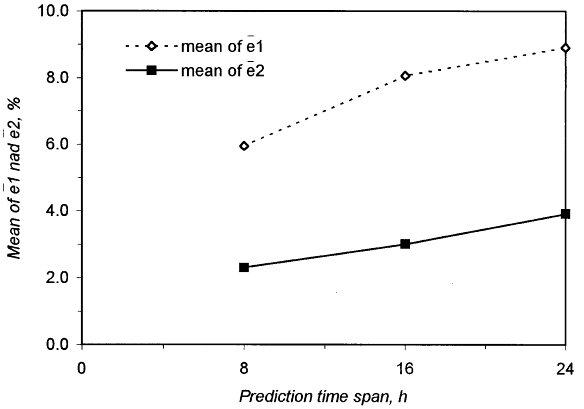

Fig. 6. Increasing tendency of the mean of $\bar{e}_1$ and $\bar{e}_2$ along prediction time span.

When using a two-step prediction procedure, the product formation at each future sampling time point is actually predicted twice. Suppose the simulation conditions are the same as above (i.e. $T_D = 24$ h, $T_{P1} = 8$ h, $T_{P2} = 16$ h and $T_M = 4$ h) and the present time is T_k, then there are two predictions for the product at time $(T_k + T_{P1})$, i.e. the +16 h prediction made at time T_{k-2}, $P_{ANN}(T_{k-2} + T_{P2})$ corresponding to the second open circle in Fig. 11(a), and the +8 h prediction made at T_k, $P_{ANN}(T_k + T_{P1})$ corresponding to the first open triangle in Fig. 11(a), respectively. If the predictor works properly, these two values should be very close to each other so that the ratio $P_{ANN}(T_k + T_{P1})/P_{ANN}(T_{k-2} + T_{P2})$ should be nearby 1.0. Define D_i as the distance between $P_{ANN}(T_k + T_{P1})/P_{ANN}(T_{k-2} + T_{P2})$ and 1.0 (see Eq. (10)), then once malfunction happens, D_i will depart from zero significantly.

$$D_i = \left| 1 - \frac{P_{ANN}(T_k + T_{P1})}{P_{ANN}(T_{k-2} + T_{P2})} \right| \tag{10}$$

For automatic malfunction detection, a critical distance, D_{ic}, is empirically determined. If $D_i > D_{ic}$, the present prediction will be regarded as a malfunction. Malfunction remediation is carried out by repeating the last training-prediction. D_{ic}

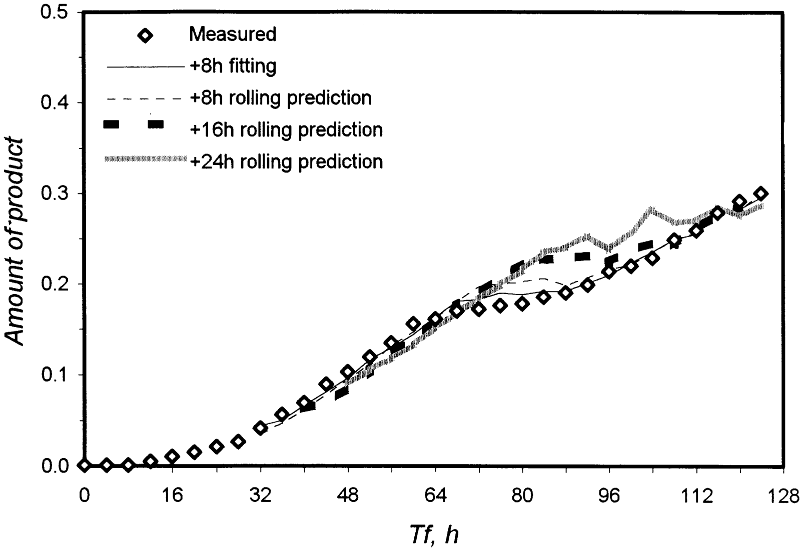

Fig. 7. Fitting and 1 ∼ 3-step prediction results by applying the rolling learning-prediction procedure for Charge 6. Symbols are measurements, lines are simulations.

was set to be 0.1 for our example, see the dotted boundary in Fig. 11(b). For Charge 9, D_i was once found to be 0.143 ($> D_{ic} = 0.1$!) at 156 h and the + 8 h prediction was a malfunction indeed. It was detected in time and remedied by retraining. Since the probability of malfunction's occurrence was observed to be approximately 0.1%, the probability that the restarted training prediction becomes another malfunction is extremely low (one in a million). Therefore, one time retraining is enough for remedying. It should also be pointed out that during the earlier phase of cultivation, D_i values can be high and sometime may exceed D_{ic}. This is usually not the result of malfunction, rather of the intrinsic uncertainty of the process. In that case, the prediction accuracy can no longer be increased by retraining. Nevertheless, no special measures are necessary to be taken to stop the retraining since occasional retraining of the network is not very harmful—one round of learning-prediction only needs a few minutes with a low-end (486DX) personal computer.

5. Discussion and conclusion

Since the rolling learning-prediction procedure is on-line application oriented, it must meet some important criteria, such as computing time, accuracy and robustness. The performance of the product predictor is largely dependent on the topology of the neural network and the database. A 21–3–2 topology of the neural network was chosen in this study. More complicated topologies have been tested, e.g. with more hidden neurons or with more hidden layers, but no significant improvement of prediction accuracy was found. The ratio of the number of input–output data pairs to the number of unknowns in the present ANN is 345:75 ≈ 4.5:1. Intuitively, this is more realistic than to determine hundreds of unknown factors (in the case of a more complicated topology) on the basis of the same amount of input–output data pairs. Keeping the topology of a neural network as simple as possible is the principle of a network design. This may avoid eventual over fitting and reduce computation time. For the given problem in this paper, a round of rolling learning-prediction took 4 min on average when using the Levenberg–Marquart optimization (Demuth and Beale, 1994) and a 486DX computer. In future applications, a larger training database (the size may be three to four times

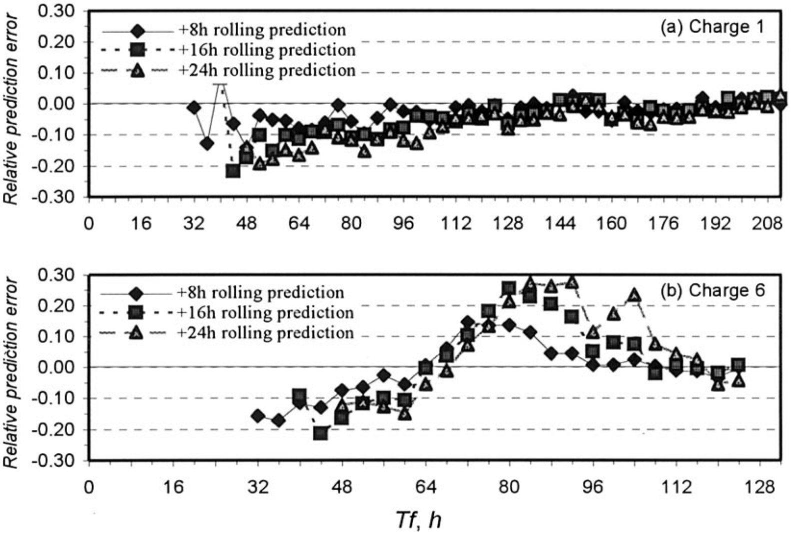

Fig. 8. Comparison of relative prediction errors for a normal charge (a) and an abnormal charge (b).

as big as used now) and eventually a more complicated topology may be compensated by faster computers with more memory.

The use of accumulated process variables is a distinguished feature of the product predictor. It is advantageous to reduce the influence of measurement noise but by no means at the cost of dynamic information. The accumulated process variables are broken down into a series of dynamic pieces via a moving data windows technique so as to get input–output data pairs. There are some principles to determine the width of the data and prediction windows as well as the moving speed. A larger T_D involves more dynamic process information, but it may increase the dimension of the input data vectors since, in order to keep the discretisation accurate, the discretising time interval can not be chosen too small so that m must increase. If T_D is too small, then the network may be too sensitive to measurement errors. As for T_P, generally speaking, it should not exceed T_D. It has been shown that the prediction window can be as large as 10% of the process cycle time with high prediction accuracy (higher than 5%). Longer term prediction is significant to indicate the future trends of a process, but other network architectures (such as recurrent neural networks) should eventually be considered. The moving step length T_M of the data window may be chosen the same as the assay sampling interval—the case of full use of the measurements. If the measurement noise is reasonably low, a larger moving step may be chosen. This may lead to a reduction of the sampling frequency and therefore the assay labor intensity. For penicillin cultivation, we have done another set of simulations with $T_M = 8$ h instead of 4 h. The results revealed that the + 8 and + 16 h prediction accuracy during

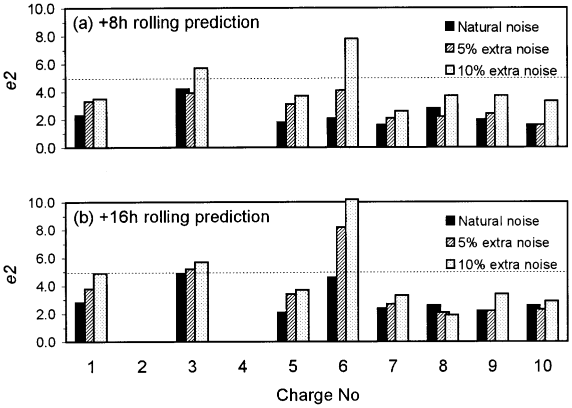

Fig. 9. Rolling learning-prediction accuracy under different noise levels.

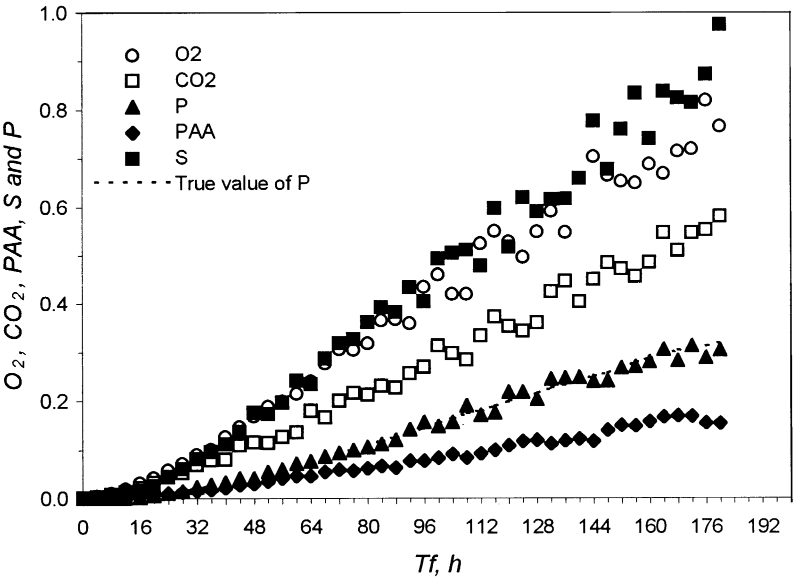

Fig. 10. Extra noise corrupted process variables of an industrial charge when 10% Gaussian noise is added. For original raw data see Fig. 2.

the later phase of cultivation is about the same as for the case of $T_{\mathrm{M}} = 4$ h.

Besides the topology of the neural network, another important factor which may influence the robustness of the predictor is the training database. In our training database $\{\theta_{1\sim n}\ \theta_{n+1}\}$, $\theta_{1\sim n}$ is charge-wise updated while θ_{n+1} is obligatorily on-line updated. Careful choice of the n historical charges is very important for success of the predictor. When using neural networks, the training database must be a representative one. That means it should include as many situations occurring in industrial cultivations as possible (except contaminations and some other extraordinary charges). Before being incorporated into the database, a historical charge should be evaluated according to yield coefficients, characteristics of transients of key process variables and so on. The aim of the evaluation is to find the charges which have similar performance. The database will be kept representative by limiting the number of similar charges to one or two. On the other hand, the database should consist of recent historical charges so that some gradually changing factors, such as climate and degeneration of equipment could be excluded as much as possible. Therefore the time span covered by the database should be reasonably short and it should be updated as soon as a new charge is finished. It could also happen that a certain bioreactor, for structural reasons, has evidently different behavior as the average level. In this case, a special database, which consists of charges carried out only in this bioreactor, should be established. More details concerning the database may be found elsewhere (Yuan et al., 1997).

For penicillin production, we have chosen O_2, CO_2, P, PAA and S as process variables. The biomass concentration was not taken into account since it is usually not regularly measured. The product predictor presented here may be regarded as a software sensor which is ‘calibrated’ by routine product analysis data supplied during industrial production. Incorporating some other process variables such as consumption of nitrogen source and sulfate would be favorable both for enhancing the robustness and safety of the predictor. The gas balance data (O_2, CO_2) should be treated carefully in applications because of the disturbances during repeated calibration of the gas analyzers. For other bioprocesses, the number of process variables may probably be limited to

Table 4
Comparison of linear extrapolation (linex) and rolling learning-prediction (rolep)[a]

	Natural noise T_P (h)			5% extra noise T_P (h)			10% extra noise T_P (h)		
	+8	+16	+24	+8	+16	+24	+8	+16	+24
$E\{\bar{e}_1\text{-linex}\}$	12.5	19.1	24.6	13.4	20.0	25.4	14.6	20.9	26.2
$E\{\bar{e}_1\text{-rolep}\}$	5.4	6.9	8.1	7.5	7.8	8.7	9.0	8.4	7.7
$E\{\bar{e}_2\text{-linex}\}$	2.0	2.8	3.7	4.3	5.5	6.6	7.6	11.2	12.4
$E\{\bar{e}_2\text{-rolep}\}$	2.3	2.8	2.9	2.6	3.1	3.6	3.5	3.7	4.0

[a] The abnormal Charge 6 is excluded. $E\{x\}$, mean of x (%).

the same level (five or six) as in our example after careful investigation on the process kinetics and working conditions. τ and m are two process dynamics dependent factors. For most industrial bioprocesses, τ may vary between 1 and 8 h and m between 2 and 5.

In summary, the rolling learning-prediction procedure proposed in this paper can give a highly accurate and noise-tolerant prediction for product formation during the second half of a cultivation. The prediction accuracy is largely improved by involving the previous data of the present charge into the database—on-line updating of the database and repeated training. The reliable prediction time span can be as large as 10% of the whole process cycle time, long enough for application in optimal production scheduling. The prediction accuracy during the first half of the cultivation is not always high because it is largely influenced by the intrinsic uncertainties of individual charges such as inherent quality fluctuations of the precultures. However, as explained earlier, because an efficient dynamic profit optimization can only be carried out during the later phase of cultivation, the rolling learning-prediction procedure could already provide an excellent support. The ANN model described in this paper is also capable of predicting substrate consump-

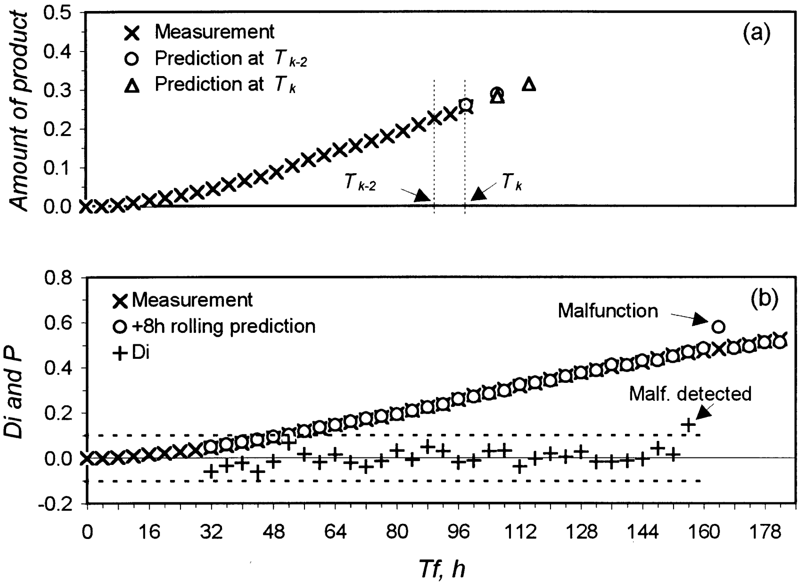

Fig. 11. Automatic detection of predictor malfunction. (a) Duplicated prediction at each sampling time point (here at time $T_k + T_{P1}$); (b) Detection of malfunction by on-line comparison of D_i with D_{ic}.

tion so that, to some extent, it may be used to determine optimal feeding rates within the prediction window. However, long-term prediction of these two variables are generally discouraged since the uncertainty of feeding profiles increases along with T_P. In fact, the prediction made in this paper is under the assumption that the feeding profiles in the prediction window are in their ordinary level. Besides total product of a charge, people in industry may also be interested in the titre prediction at T_P (h). This is easily obtained from the neural net predictions.

Acknowledgements

Applitek NV, Deinze/Belgium and Alexander von Humboldt Foundation/Germany are acknowledged for financial support. Part of this work was supported by The Natural Science Foundation of China and The Fund for Scientific Research, Flanders/Belgium. The authors are grateful to Y.Q. Shen of Harbin Pharmaceutical factory, People's Republic of China for his valuable assistance and discussions.

Appendix A. Nomenclature

$\mathbf{b}$	biases of the neural network
$CO_2(t)$	total carbon dioxide production at time t (kmole)
D_{ic}	critical distance for judgement of malfunction
$e(t)$	relative prediction error at time t (%)
$\bar{e}$	average of relative prediction errors (%)
$\bar{e}_1$	average of relative prediction errors during earlier phase of cultivation (%)
$\bar{e}_2$	average of relative prediction errors during later phase of cultivation (%)
$E\{x\}$	mean of x
i, $n+1$	charge number
m	dating back steps when discretising the transients covered by data window
N	number of the input–output data pairs of a charge
$\mathrm{Nit}(t)$	total nitrogen source consumption at time t (kg)
$N(\theta)$	number of input–output data pairs in database θ
$O_2(t)$	total oxygen consumption at time t (kmole)
$P(t)$	total product formation at time t (kg)
$\mathrm{PAA}(t)$	total phenyl acetic acid consumption at time t (kg)
$P_{ANN}(t)$	predicted total product by ANN at time t (kg)
$\mathrm{pH}(t)$	average pH value at time t
$P_m(t)$	measured total product at time t (kg)
$pO_2(t)$	average dissolved oxygen at time t (%)
$S(t)$	total reducible sugar consumption at time t (kg)
$\mathrm{Temp}(t)$ °C	average temperature of the medium at time t
T_D	width of data window (h)
T_f	cultivation time (h)
T_k	cultivation time at the right border of kth data window (h)
T_M	moving step length of data window (h)
T_P	width of prediction window (h)
T_{P1}	width of one step prediction window (h)
T_{P2}	width of two steps prediction window (h)
T_S	sampling interval (h)
$\mathbf{w}$	weighting factors of the neural network
$x(t)$	vector of discretised process variables at time t
$X(k)$	kth neural network's input vector of a charge
$Y(k)$	kth neural network's output vector of a charge
Greeks	
θ	database network training
$\theta_{1\sim n}$	collection of input–output data pairs of $1\sim n$ historical charges
θ_{n+1}	collection of input–output data

pairs of the present charge

τ discretising time interval of data window (h)

References

Aynsley, M., Hofland, A., Morris, A.J., Montague, G., Di Massimo, C., 1993. Artificial intelligence and the supervision of bioprocesses (Real-time knowledge-based systems and neural networks). Adv. Biochem. Eng./Biotechnol. 48, 1–27.

Bhat, N., McAvoy, T.J., 1990. Use of neural nets for dynamic modeling and control of chemical process systems. Comp. Chem. Eng. 14, 573–583.

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Math. Control Signal Systems 2, 303–314.

Demuth, H., Beale, M., 1994. Neural Network Toolbox User's Guide. The Math Works.

Elman, J.L., 1990. Finding structure in time. Cognitive Sci. 14, 179–211.

Karjala, T.W., Himmelblau, D.M., 1994. Dynamic data rectification by recurrent neural networks vs. traditional methods. AIChE J. 40, 1865–1875.

Karim, M.N., Rivera, S.L., 1992. Artificial neural networks in bioprocess state estimation. Adv. Biochem. Eng./Biotechnol. 46, 1–31.

Karim, M.N., Yoshida, T., Rivera, S.L., Saucedo, V.M., Eikens, B., Oh, G.-S., 1997. Global and local neural network models in biotechnology: application to different cultivation processes. J. Ferment. Bioeng. 83, 1–11.

Leonard, J.A., Kramer, M.A., 1990. Improvements of the backpropagation algorithm for training neural networks. Comp. Chem. Eng. 14, 337–341.

Linko, S., Zhu, Y.-H., Linko, P., 1995. Neural networks in lysine fermentation. In: Munack, A., Schügerl, K. (Eds.), Computer Applications in Biotechnology. IFAC Symposium Series. Pergamon Press, Oxford, pp. 336–339.

Montague, G., Morris, J., 1994. Neural-network contributions in biotechnology. Trends Biotechnol. 12, 312–324.

Psichogios, D.C., Ungar, L.H., 1992. A hybrid neural network-first principles approach to process modeling. AIChE J. 38, 1499–1511.

Raju, G.K., Cooney, C.L., 1992. Using neural networks for the interpretation of bioprocess data. In: Karim, M.N., Stephanopoulos, G. (Eds.), Modeling and Control Biotechnical Processes 1992. IFAC Symposia Series. Pergamon Press, Oxford. Number 10, pp. 425–428.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J. (Eds.), Parallel Data Processing, vol. 1. M.I.T. Press, Cambridge, MA, pp. 318–362. Chapter 8.

Schubert, J., Simutis, R., Dors, M., Havlik, I., Lübert, A., 1994. Bioprocess optimization and control: application of hybrid modeling. J. Biotechnol. 35, 51–68.

Su, H.-T., McAvoy, T.J., 1992. Long-term predictions of chemical processes using recurrent neural networks: a parallel traing approach. Ind. Eng. Chem. Res. 31, 1338–1352.

Thibault, J., Van Breusegem, V.C., Cheruy, A., 1990. On-line prediction of fermentation variables using neural networks. Biotechnol. Bioeng. 43, 1041–1048.

van Can, H.J.L., te Braake, H.A.B., Hellinga, C., Luyben, K.C.A.M., Heijinen, J.J., 1997. An efficient model development strategy for bioprocesses based on neural networks in balances. Biotechnol. Bioeng. 54, 549–566.

Williams, R.J., Peng, J., 1990. An efficient gradient based algorithm for on-line training of recurrent network trajectories. Neural Comput. 2, 490–501.

Williams, R.J., Ziper, D., 1989. A learning algorithm for continually running fully recurrent neural networks. Neural Comput. 1, 270–280.

Yuan, J.Q., Vanrolleghem, P.A., 1998. One-step-ahead product predictor for profit optimization of penicillin fermentation. In: Yoshida, T., Shioya, S. (Eds.), Computer Applications in Biotechnology. IFAC Symposium Series. Pergamon, Oxford, pp. 183–188.

Yuan, J.Q., Guo, S.R., Schügerl, K., Bellgardt, K.-H., 1997. Profit optimisation for mycelia fed-batch fermentation. J. Biotechnol. 54, 175–193.