15th Triennial World Congress, Barcelona, Spain

NEURAL NETWORK BASED ON-LINE RE-OPTIMISATION OF FED-BATCH PROCESSES USING ITERATIVE DYNAMIC PROGRAMMING FOR DISCRETE-TIME SYSTEMS

Zhihua Xiong and Jie Zhang

Centre for Process Analytics and Control Technology
Department of Chemical and Process Engineering
University of Newcastle, Newcastle upon Tyne, NE1 7RU, U. K.
E-mail: Zhihua.Xiong@ncl.ac.uk, Jie.Zhang@newcastle.ac.uk

Abstract: Optimisation of fed-batch processes can be described as a constrained non-linear end-point dynamic optimisation problem. Although iterative dynamic programming (IDP) is feasible, it is usually very time-consuming and very difficult to apply to on-line optimisation because of solving the non-linear differential-algebraic equations of the process model in each iteration. The replacement of a rigorous mechanistic model by an equivalent neural network (NN) model takes the advantage of high speed processing, since simulation with a NN model involves only a few non-iterative algebraic calculations. To use IDP algorithm for NN model based on-line re-optimisation, a modified algorithm is proposed and is called as iterative dynamic programming for discrete-time system (*IDP/DTS*). The novel *IDP/DTS* algorithm can obtain a reduction of many times in computational time compared to the conventional IDP algorithm. In this paper, an effective optimisation and control scheme for on-line re-optimisation of fed-batch processes is proposed based on NN models and the novel *IDP/DTS* algorithm. The proposed scheme is illustrated using simulation studies of an ethanol fermentation process. *Copyright © 2002 IFAC*

Keywords: neural networks, on-line re-optimisation, IDP, *IDP/DTS* algorithm, fed-batch processes.

1. INTRODUCTION

In recent years, there has been a growing interest in the use of optimisation in the control of fed-batch processes. In fed-batch processes, our interests are concerned with determining the feed rate to the reactor that will give the maximum amount of the desired product. Although only a single control variable in the form of the feed rate may appear to represent a simple optimal control problem, considerable difficulties have been reported in the determination of the optimal feeding policies for fed-batch processes, because fed-batch processes, in general, are non-linear dynamic systems. It is always very difficult to obtain rigorous mechanistic models due to the complexity of the processes, and also difficult to implement them for on-line optimisation and control.

When building first principles models is very costly and difficult, empirical models based on process input-output data can provide the useful alternative. Neural networks (NN) have been proposed as a promising tool for identifying empirical models (Bhat and McAvoy, 1990) and have been shown to be capable of approximating any continuous non-linear functions (Sjoberg *et al*, 1995). If properly trained and validated, these neural network models can be used to accurately predict steady-state and dynamic process behaviour, hence, leading to improved process optimisation and control performance (Narendra and Parthasarathy, 1990; Sjoberg *et al*, 1995; Zhang and Morris, 1999).

Furthermore, the most important aspects to be considered in a fed-batch process are the changes in process parameters and/or dynamics during the operation of the batch. Once the dynamic of a process is altered, the off-line calculated optimal control profile will be no longer optimal. There is a need to re-optimise the process on-line using new process dynamics (Tian *et al.*, 2001).

Optimisation of fed-batch processes is usually a constrained non-linear dynamic optimisation problem. Direct search methods could be feasible for this kind of optimisation problem. They are also reliable and the possibility of obtaining the global optimum is very high, although most of the direct search methods are very time-consuming because they usually require a large number of iterations. Many direct search methods have been reported, such as iterative dynamic programming (IDP) (Luus, 1990, 1992, 1993), heuristic random optimisation (Li and Rhinehart, 1998), and direct detailed grid search method (Nascimento *et al*, 2000).

IDP, as developed by Luus (1990), offers a good alternative for obtaining the global optimum. IDP has the advantage of not requiring additional variables to be introduced, and therefore the method does not encounter those problems as singular control. Bojkov and Luus (1996) have shown that penalty functions are very effective in handling final state constrained problems solved by IDP when an appropriate penalty function factor is selected. IDP has also been found to be well suited for an extractive fermentation optimisation problem and for general constrained optimal control problems by Dadebo and McAuley (1995). However, IDP is computationally time-consuming because of solving the non-linear differential-algebraic equations (DAEs) of process models in each iteration. The CPU time required for the whole iterations was approximately 2 hours in a typical batch process (Bojkov and Luus, 1996). It is very difficult to use IDP for on-line optimisation and control.

For on-line optimisation purposes, it is strongly desirable that the model can be simulated in short time while capable of describing the system accurately. In this sense, a properly trained NN model is a good substitute for a rigorous mechanistic model. High speed processing is obtained because simulation with a NN model involves only a few non-iterative algebraic calculations. IDP methods (by Luus, 1990) have been proposed only for the rigorous process models in the form of differential equations, discrete time models, such as NN models, have not been involved. Re-optimisation using IDP has not been reported either. Although some mismatches between a NN model and the real process may exist, on-line re-optimisation will overcome this disadvantage by tracking the process dynamic behaviour in real time. The aim of this paper is to present a simple and effective optimisation and control scheme for on-line re-optimisation of fed-batch processes based on neural network models, making possible to proceed the re-optimisation by *IDP/DTS*. In this study, the scheme mainly focuses on the model-plant mismatches of a neural network model, but it is also suitable for different initial conditions of batch processes, such as those caused by reactive impurities (Tian *et al.*, 2001).

The rest of this paper is structured as follows: Section 2 describes on-line optimisation based on neural network. In Section 3, the *IDP/DTS* optimisation algorithm is studied and developed based on NN model, then the whole scheme of on-line re-optimisation is proposed in detail. The proposed scheme is illustrated on a simulation of a fed-batch fermentation plant in Section 4. Finally Section 5 draws some concluding remarks.

2. NEURAL NETWORK BASED ON-LINE OPTIMISATION CONTROL

Many fed-batch processes can be considered as a class of control-affine non-linear systems described as follows

$$\dot{x} = f(x) + g(x)u, \quad x(0) = x_0 \tag{1}$$

where $x \in \boldsymbol{R}^p$, $u \in \boldsymbol{R}^q$, f and g are p-dimensional analytic vector fields. In fed-batch processes, the maximum amount of the desired product is concerned by determining feed rate to the reactor. This leads to an endpoint optimal control problem of a non-linear system, which can be mathematically formulated as

$$\min_{u(t)} J(u(t)) = \phi(x(t_f)) \tag{2}$$

where t_f is the final time of a batch and it is given.

The optimisation problem of the discrete-time system may be described as

$$\min_{u(k)} J(u(k)) = \phi(y(N)) \tag{3}$$

s. t.

$$y(k+1) = y(k) + \int_{kh}^{(k+1)h} (f(y(t)) + g(y(t))u(t))dt$$
$$= F(y(k), y(k), h) \tag{4}$$
$$H(y, u) \le 0 \tag{5}$$

where y is discrete variable of x in equation (1), h is the sampling interval, H is constraint function, time k=0 defines the start of a batch and k=N defines the end of a batch.

Neural networks have been claimed as a universal non-linear approximator (Sjoberg *et al.*, 1995) and their applications to model based control in the field of chemical engineering are growing fast (Bhat and McAvoy, 1990; Nascimento *et al.*, 2000). Because a properly trained and validated NN model can accurately predict long-range dynamic behaviour of the process, the optimisation and control profile can be calculated off-line based on a NN model. The straightforward optimisation approach is to

implement a NN model within a conventional optimisation method. This approach has been used by Altissimi *et al*. (1998) to optimise the profit a hydrocraking reactor using sequential quadratic programming (SQP). They replaced a first-principle model with a NN and obtained a reduction of at least 60 times in computational time. Nascimento *et al*. (2000) presented a study of NN based optimisation of a polymerisation process in a twin-screw extruder reactor. They used a NN model to carry out a detailed grid search in the region of interest. The use of a NN model takes advantage of the comparative rapidity of the NN model based simulation. In this paper, we use a NN model to replace a rigorous mechanistic model in the IDP scheme, and propose a modified IDP algorithm for discrete-time system, which is called as *IDP/DTS*. High speed processing of *IDP/DTS* makes it possible to be used in an on-line re-optimisation procedure for batch processes.

Using a neural network to model the fed-batch process given by Eq(1), the control-affine non-linear system can be described in discrete-time function as

$$y(k+1) = f_{NN}(y(k),\ldots,y(k-n+1), u(k)) \quad (6)$$

Here *Feedforward Neural Network* (*FNN*) is selected to approximate the non-linear relationship between input and output of a fed-batch process. Using a properly trained and validated *FNN* model, long-range dynamic behaviour of a process can be predicted accurately (Zhang, 2001).

To reduce the complexity of the optimisation problem, the set of possible control trajectories is restricted to a finite-dimensional space. The control input is here parameterised as a piecewise constant function.

3. IDP FOR DISCRETE-TIME SYSTEM (*IDP/DTS*) OPTIMISATION ALGORITHM

The IDP algorithm developed by Luus (1990) has been reported only involving the continuous-time rigorous mechanistic models or first-principle models. As to discrete-time model, such as NN model, there are some steps that should be modified if IDP algorithm is to be used. Firstly, the method of constructing grid-points is to be revised. In IDP, for a rigorous model, the vector of state-space variables $\chi(t) = (x_1(t),\ldots,x_n(t))$, where n is the order of continuous-time system, is the basement of grid-points. But for a discrete-time system, the model is described as

$$y(k+1) = f(y(k),\ldots,y(k-n_y+1), u(k),\ldots,u(k-n_u+1)) \quad (7)$$

where n_u and n_y are the maximum lags in the input and output respectively, known as the orders of the discrete-time system, and subject to $(n_y+n_u) \geq n$. Then the basement of grid points is changed into

$$\chi(k)= (y(k),\ldots,y(k-n_y+1), u(k),\ldots,u(k-n_u+1)) \quad (8)$$

Secondly, replace a NN model for the rigorous mechanistic model at each time stage and at each iteration. It only involves a few algebraic calculations, not to solve non-linear DAEs of process model, thus the time cost decreases significantly. It benefits to construct on-line re-optimisation using this kind of modified IDP algorithm. Thirdly, when used in an on-line re-optimisation procedure, time stage P decreases gradually. Finally, neural network should be built using input/output data set, in order to predict output of the process in *IDP/DTS* procedure.

The *IDP/DTS* procedure can be outlined in the following algorithm:

(1) Divide the time interval of a batch $[0, t_f]$ into P time stage and each of length L, guess the initial values of control policy u^0, and set index j to 0.

(2) Choose the number of χ-grid points Q at each time stage and the number of allowable values M for the control u. If the constraints are existed in the non-linear optimisation problem and are violated for the components of these M allowable values, the clipping technique is used to substitute the constraints for the values. Choose also the initial region $r(0)$, over which the allowable values for control can be selected.

(3) Choose Q nominal initial allowable values of control, where Q is odd, and subsequently perturbing $u^0(i)$ uniformly for each time stage inside the allowable region for control to generate Q control trajectories $u(k,i)$

$$u(k,i) = \begin{cases} u^0(i) - \dfrac{k}{(Q-1)/2} r(j), k = 1,3,\cdots,Q \\ u^0(i) + \dfrac{k}{(Q-1)/2} r(j), k = 2,4,\cdots,Q-1 \end{cases} \quad (9)$$

where i=1,2,…,P, and j is the iteration index. Using the Q control trajectories, based on the NN model, predict recursively Q times to generate and store the χ-grid for each time stage.

(4) Starting at the last time stage P, corresponding to time t_f -L, for each χ-grid point, predict based on NN model from t_f -L to t_f once with each of the M allowable values for control. Choose the value of control u that minimises the performance index and store the corresponding value of control for use in step (5).

(5) Step back to stage P-1, corresponding to time t_f -2L, and predict based on NN model from t_f -2L to t_f -L for each χ-grid point once with each of the M allowable values for control. To continue to predict based on NN model from t_f -L to t_f, choose the control from step (4) that corresponds to the grid point closest to the resulting χ at t_f -L. Compare the M values of the performance index and store the value of control that gives the minimum value.

(6) Repeat this procedure for stage P-2, P-3, etc., until stage 1, corresponding to the initial time t=0, is reached. Store the control policy that minimises the performance index.

(7) Reduce the region for the allowable values of control by a factor α, i.e., $r(j+1) = \alpha\, r(j)$, where j is the iteration index, and α=0.7 here. Use the optimal control policy obtained in step (6) as the nominal value for u^0.

(8) Increase the iteration index, j, by 1. If $j < T$ (e.g. T = 20), go to step (3); else record both the values of performance index and the best control policy at each iteration, and stop.

4. OPTIMAL CONTROL OF A FED-BATCH FERMENTATION PROCESS

The fed-batch fermentation process was taken from Hong (1986) and Luss (1993). The mechanistic model of the fed-batch ethanol fermentation process is described as follows

$$\frac{dx_1}{dt} = Cx_1 - \frac{x_1}{x_4}u \tag{10}$$

$$\frac{dx_2}{dt} = -10Cx_1 + \frac{(150 - x_2)}{x_4}u \tag{11}$$

$$\frac{dx_3}{dt} = Dx_1 - \frac{x_3}{x_4}u \tag{12}$$

$$\frac{dx_4}{dt} = u \tag{13}$$

with

$$C = \frac{0.408x_2}{(1 + x_3/16)(0.22 + x_2)} \tag{14}$$

$$D = \frac{x_2}{(1 + x_3/71.5)(0.44 + x_2)} \tag{15}$$

where x_1 is the cell mass concentration, x_2 is the substrate concentration, x_3 is the product concentration, x_4 is the liquid volume of the reactor. The initial condition is specified as $x(0) = [1\ 150\ 0\ 10]^T$, with the feed rate to the reactor u constrained by $0 \le u \le 12$, the liquid volume of the reactor is limited by the 200l vessel size, and t_f is set to 63.0 h.

The performance index is the yield of the reactor, which is to be maximised by choosing the feed rate u at the end of a batch

$$\min_{u(t)} J(u) = -x_3(t_f)x_4(t_f) \tag{16}$$

In this study, we assume that the above mechanistic model, Eq(10~15) is not available, thus an empirical model has to be utilised. Because the performance index only include variables x_3 and x_4, we use a *FNN* to model the non-linear relationship between x_3 and u. And x_4 is the liquid volume of the reactor, then $x_4(t_f)$ can be integrated by u easily. Thus the whole discrete-time model of fed-batch process is described

$$y_p(k+1) = f_{NN}\,(y_p(k), y_p(k-1), y_p(k-2), u(k)) \tag{17}$$

$$V(N) = \sum_{k=0}^{N-1} u(k)h + V(0) \tag{18}$$

where $y_p = x_3$, $V = x_4$, $y_p(0) = 0$, $V(0) = 10$, N=10, h is interval time.

Several batches of the process operation under different feeding policies are simulated from the mechanistic model to produce the data set for *FNN* modelling. The appropriate topology and weights of *FNN* model is determined by examining the least SSE on testing data. *FNN* is trained using the Levenberg-Marquardt algorithm with regularisation and using an "early stopping" mechanism to prevent over-fitting. After *FNN* training, predictions of y_p from a typical batch run is shown in Figure 1. It can be seen that these predictions are quite accurate.

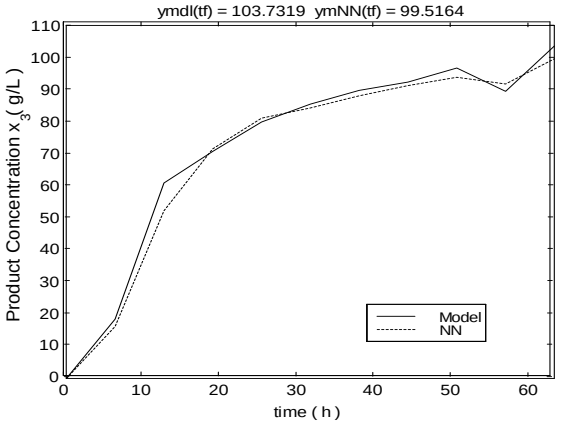

Fig. 1. Predictions at a typical batch run of NN model compared to real model

According to the discrete-time model, this optimisation problem is a final state constrained non-linear dynamic optimisation. Using the penalty function technique (Bojkov and Luus, 1996), and taking into account the final state inequality constraint, the optimisation problem with the augmented performance index is chosen in the form

$$\min_{u(k)} J(u) = -y_p(N)V(N) + \lambda \sum_{k=1}^{P} p(k) \tag{19}$$

s. t.

$$y_p(k+1) = f_{NN}\,(y_p(k), y_p(k-1), y_p(k-2), u(k)) \tag{20}$$

$$V(N) = \sum_{k=0}^{N-1} u(k)h + V(0) \tag{21}$$

$$0 \le u(k) \le 12 \tag{22}$$

$$p(k) = \begin{cases} 0, & if\,(V(k) \le 200) \\ V(k) - 200, & if\,(V(k) > 200) \end{cases} \tag{23}$$

where λ is the penalty function factor and is set to 1000 here.

To investigate the convergence of performance index for this problem and compare all three cases, listed in Tabel 1, the results of off-line optimal control profile and actural output of process are obtained by using IDP algorithm based on the mechanistic model, i. e.,

Table 1 Three cases list

Case 1:	off-line optimisation, *IDP*, rigorous model
Case 2:	off-line optimisation, *IDP/DTS*, NN model
Case 3:	on-line re-optimisation, *IDP/DTS*, NN model

DAEs model Eq(10~15). All the calculations were carried out using MATLAB5.2 running on a Pentium 800 Personal Computer. The MATLAB command ODE45 was used for integration of the differential equations in the mechanistic model.

We choose the following parameters: time stage N=10, a reduction factor α=0.7, initial control policy u^0 =4.0, and an initial control region size $r(0)$=5.0, which are the same as those used in Luus (1993). Detailed results showing the effects of the numbers of grid points Q, allowable control values M, and iterations T, are given in Table 2. In Case 1, when values of Q, M, T are increased, the convergence properties of iteration calculation are increased correspondingly and better values of performance index are obtained, but CPU time also raises significantly. When Q is 81 and M is 17, it is too much time-consuming, about 5.5 hours, to be applied to on-line optimisation. In Case 2 and Case 3, the performance of optimisation are also better and time-cost has not been the problems when parameters are increased. To compare performance of all methods in detail, the suitable parameters, Q=41, M=7, T=20, are chosen. In Case 1, the performance index value is 20706, which is close to the result reported by Luus (1993), but it costs 142 minutes CPU time because it has to solve those differential equations at each iteration. It is therefore very difficult to be used on-line. On the other hand, in Case 2, using *IDP/DTS* based on NN model, the off-line optimisation only takes 7 minutes CPU time, almost 20 times reduction compared to Case 1. Although the result of off-line optimisation in Case 2, 16680, is not as good as that in Case 1, on-line re-optimisation can overcome this disadvantage. In Case 3, the control profiles gained in Case 2 are chosen to be the initial trajectories to search optimum by *IDP/DTS*. The result is improved further as shown in Table 2, 20134, close to the result of Case 1, and it only takes 28 minutes CPU time to complete all on-line re-optimisation procedure, almost decreases 5 times compared to Case 1. The piecewise constant optimal control policies of all three cases are shown in Figure 2.

The corresponding outputs $V(k)$ under these optimal control polices are shown in Figure 3. The final volumes $V(N)$ in all cases are exactly 200 l. It means that the final state constraints are satisfied by using the penalty function technique.

It is interesting to note the end time stage where the feed rate $u(N\text{-}1)$ is zero when volume $V(N)$ reach the constraints. Figures 4 and 5 compare the actual and predicted values of output $y(k)$ based on NN model in Cases 2 and 3, and they are both compared to the results of Case 1, where "o" stands for $y(N)$ at the end of batch in Case 1, and "x" and "∇" stand for actual output $y(N)$ and NN model prediction $y_{NN}(N)$ at the end of batch in Cases 2 and 3 respectively. In spite of the poor results in Case 2, the results obtained in Case 3 are gradually improved under the on-line re-optimisation scheme. This demonstrates that the on-line re-optimisation procedure is very beneficial.

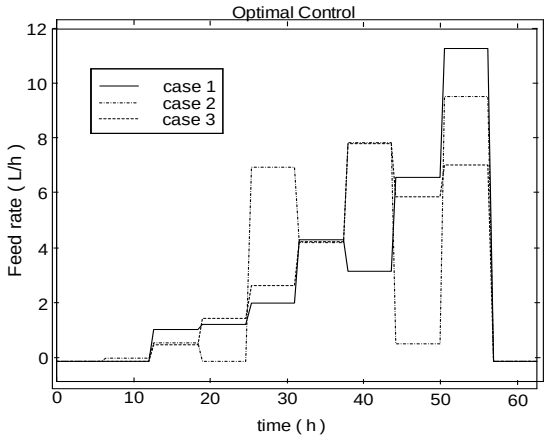

Fig. 2. Piecewise constant optimal control policies of all cases

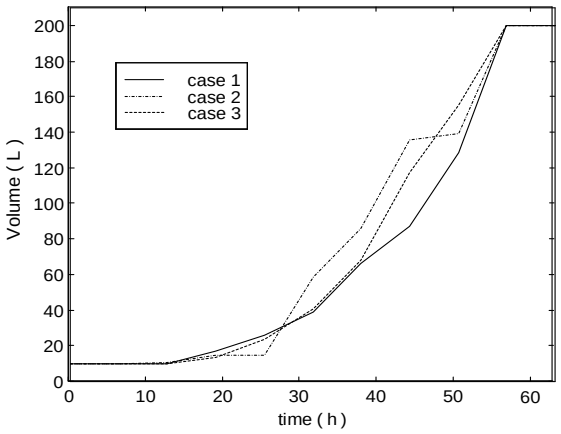

Fig. 3. Corresponding output $V(k)$ under optimal control profiles in all cases

Table 2 Performance comparison of different parameters

	Performance Index			CPU time(min)		
	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
Q=41, M=5, T=20	20601	15947	18668	143.34	4.54	18.09
Q=35, M=5, T=20	20622	17460	19859	111.69	4.90	19.96
Q=41, M=7, T=10	20574	16408	19807	76.29	3.18	13.53
Q=41, M=7, T=20	20706	16680	20134	142.38	7.25	28.12
Q=81, M=17, T=20	20721	16806	20175	326.74	7.49	32.24

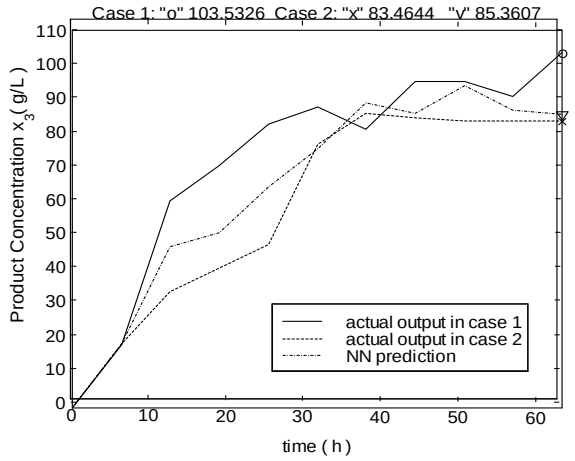

Fig. 4. Actual and predicted values of output $y(k)$ in Case 2 compared to Case 1

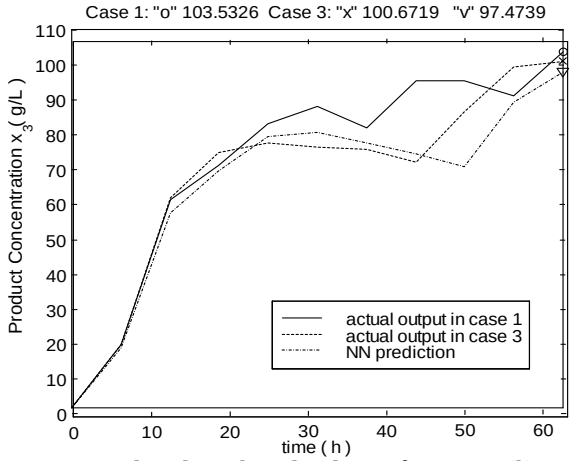

Fig. 5. Actual and predicted values of output $y(k)$ in Case 3 compared to Case 1

5. CONCLUSIONS

IDP algorithm using penalty function provides an effective way of solving a rather difficult optimisation problem of non-linear control-affine fed-batch process. However, IDP with mechanistic models is usually very time-consuming due to integrating differential equations. NN models can take the advantage of high speed proceeding and are very useful in on-line re-optimisation. A novel *IDP/DTS* algorithm is proposed to utilise a NN model in stead of a mechanistic model of a process in IDP. An effective on-line re-optimisation scheme based on NN model using *IDP/DTS* is also presented. All are demonstrated in a benchmark fed-batch process. It has been shown that using off-line optimisation with *IDP/DTS*, almost 20 times CPU time are reduced compared to using IDP based on the rigorous mechanistic model and on-line re-optimisation procedure reduces computation time by almost 5 times. In spite of poor results in off-line optimisation using *IDP/DTS*, the performance index is improved by using on-line re-optimisation.

ACKNOWLEDGEMENT

The work was supported by the UK EPSRC through the Grant GR/N13319 – "Non-linear Optimising Control in Agile Batch Manufacturing".

REFERENCES

Altissimi R., Brambilla A., Deidda A., and Semino D. (1998). Optimal operation of a separation plant using artificial neural networks, *Comput. Chem. Eng.*, Vol. **22**, 939-942.

Bhat, N., and McAvoy, T. (1990). Use of neural nets for dynamic modelling and control of chemical process systems, *Comput. Chem. Eng.*, Vol. **14**, 573-583.

Bojkov, B. and R. Luus (1996). Optimal control of non-linear systems with unspecified final times, *Chem. Eng. Sci.*, Vol. **51**, 905-919.

Dadebo, S. A. and McAuley, K. B. (1995). Dynamic optimisation of constrained chemical engineering problems using dynamic programming, *Comput. Chem. Eng.*, Vol. **19**, 513-525.

Hong, J. (1986). Optimal substrate feeding policy for fed batch fermentation with substrate and product inhibition kinetics, *Biotechnol. Bioeng.*, Vol. **27**, 1421-1431.

Li, J. and Rhinehart R. R. (1998). Heuristic random optimisation. *Comput. Chem. Eng.*, Vol. **22**, 427-444.

Luus, R. (1990). Optimal control by dynamic programming using systematic reduction in grid size, *Int. J. Control*, Vol. **51**, 995-1013.

Luus, R. (1992). On the application of iterative dynamic programming to singular optimal control problems. *IEEE Trans. Autom. Control*, Vol. **37**, 1802-1806.

Luus, R. (1993). Application of dynamic programming to differential algebraic process systems, *Comput. Chem. Eng.* Vol. **17**, pp 373-377.

Narendra, K. S., and Parthasarathy K. (1990). Identification and control of dynamic systems using neural network, *IEEE Trans. on Neural Network*, Vol. **1**, 4-27.

Nascimento, C. A., R. Giudici, and R. Guardani (2000). Neural network based approach for optimisation of industrial chemical processes, *Comput. Chem. Eng.*, Vol. **24**, 2303-2314.

Sjoberg, J., Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Y. Glorennec, H. Hjalmarsson, and A. Juditsky (1995). Non-linear black-box modeling in system identification: a uinfied overview, *Automatica*, Vol. **31**, 1691-1724.

Tian, Y., J. Zhang, and J. Morris (2001). On-line re-optimisation control of a batch polymerisation reactor based on a hybrid recurrent neural network model, *American Control Conference*, Arlington, Virginia, USA, 25-27 June, 2001, pp350-355.

Zhang J., and Morris A. J. (1999). Recurrent neuro-fuzzy networks for non-linear process modeling, *IEEE Trans. on Neural Networks*, Vol. **10**, 313-326.