Neural network model-based predictive control of a distillation column – a neural network modelling methodology

by Paul Turner*, Dr Osvaldo Agammenoni†, Prof Geoff Barton†, Prof Julian Morris* and Dr Gary Montague*

This paper describes the collaborative results of a study between The University of Newcastle, Sydney University, ICI Engineering Technology and ICI Australia Pty Ltd into the application of neural networks to Model-Based Predictive Control. The results discussed will describe the methodology of extracting data from a real industrial process, pre-processing the data, selection of key inputs using dynamic correlation and multivariate statistics, process modelling and control. The paper will emphasise the importance of combining engineering knowledge, advanced statistics and neural networks in order to obtain an extremely powerful modelling technique for dealing with non-linear systems. The implementation of the controller was carried out on a validated simulation of the actual process. This Speedup model had been developed by Sydney University over a period of 18 months and had been used previously to design other successful control strategies that are not on-line on the process.

The resultant Model-Based controller was benchmarked against a linear model-based controller and two PID controllers. The neural network controller not only outperformed the linear MBPC by a 50% reduction in standard deviation but also reduced overshoot and settling time dramatically.

Keywords: Neural networks; model based control; predictive; distillation column.

1. Introduction

This paper summarises the results of a three-month collaborative project between ICI and Universities in the United Kingdom and Australia. The work is significant not just for its application of a new technology to an industrial process, but also for the development of a methodology of applying neural networks that has arisen as a consequence. The work concentrated on the methodology for developing a closed-loop model-based predictive control scheme using neural networks. In a model-based predictive control scheme, a neural network can be used to provide a non-linear model of the process which can then be used to predict 'optimum' future control responses (Bhat and McAvoy, 1990; Hernandez and Arkun, 1992). The particular problem addressed was that of pressure control of a C3 splitter distillation column operating on one of ICI's sites in Australia. The methodology developed takes the engineer from data collection right through to on-line control. This project formed part of an overall three-year collaboration which has examined the potential of neural networks in a wide range of applications.

The C3 splitter application was a particulary interesting problem because of the complex non-linear dynamics displayed by the column. The Speedup simulation of the column took 18 months to develop and although it gave a fair representation of the process, each calculation of '1 minute data' took approximately 30 sec to calculate. The neural network produced predictions almost instantaneously. The problem was therefore difficult from a first-principles perspective and the added advantage of the neural network was the speed of calculation. This enables it to be used in a model-based control strategy since the optimisation of the objective function (see later) requires several model calculations in order to compute the optimum controller action.

2. Neural networks

Neural networks are the current emphasis of a great deal of research. Their original popularity was generated by analogies to the workings of the human brain and human behaviour. These claims may have been applicable in the signature and speech recognition application area but are far from an accurate representation for their application in the process industries. The human brain contains billions of interconnected neurons and synapses. The neural networks used in modelling rarely have more than five or six.

Neural networks, in perspective, can be thought of as non-linear approximators. Theory suggests that a neural network with a single layer of 'hidden' neurons can be used to approximate any non-linear, continuous function (Cybenko, 1989). A neural network simply maps a set of input data to a set of output data by optimising a non-linear interconnected set of equations. A neural network is therefore a non-linear mathematical equation.

3. Dynamic neural networks

The neural networks used in this project were necessarily dynamic in their architecture. Several dynamic architectures were studied during this project. These

*Departent of Chemical & Process Engineering, University of Newcastle-Upon-Tyne, UK
†Department of Chemical Engineering, University of Sydney, Australia

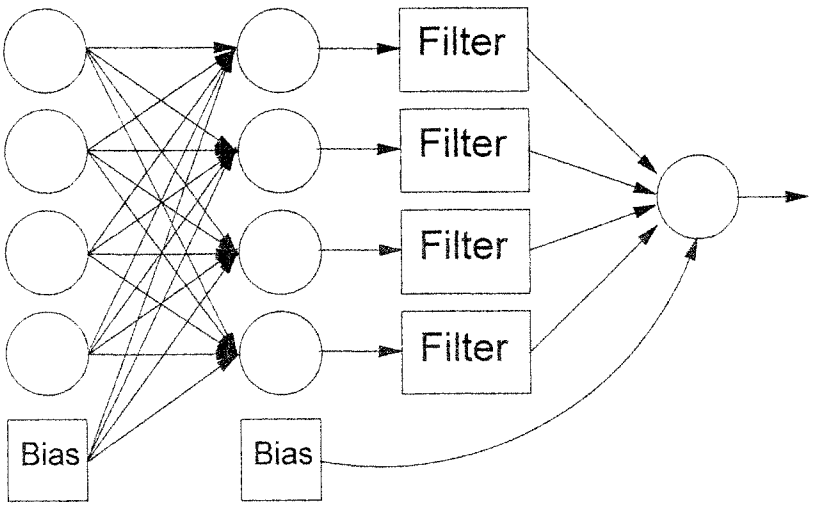


Fig 1 Filter-based network architecture

consisted of Filter-based (Willis *et al*, 1991), Time-series (Hertz *et al*, 1991), Fully recurrent (Williams and Zipser, 1989) and Hammerstein (Eskinat *et al*, 1991). The most suitable architecture for this particular application was found to be the Filter-based network. This structure offered a very efficient dynamic modelling capability, was easy to train and had a comparable number of coefficients to that required in linear ARMAX models. The potentially more powerful fully recurrent architecture was avoided because the training algorithms are computationally much more expensive. The architecture of a Filter-based network is shown in Fig 1.

4. The methodology

The methodology adopted for implementing a neural model-based controller is not necessarily unique to neural networks. The procedure is applicable to any statistical based modelling exercise. At least 80% of the project time is associated with rigorous data pre-processing and analysis. The final 20% includes the neural network modelling which, in comparison, is the 'trivial' part of the methodology.

4.1 Stage 1: Data collection

Data collection can be one of the most time consuming and frustrating stages of the methodology. There are many questions that should be asked prior to collecting data. Can the data be extracted from the process monitoring computer and loaded into a PC? Are the data in the correct format? Are they collected at the correct frequency? Are all the necessary variables for the model being logged? Will the data contain sufficient variation for models to be created?

For this application, the process was variable enough to allow a model to be identified. A rule of thumb suggests a signal-to-noise ratio of greater than 6 will suffice for statistical modelling. However, for non-linear modelling, the data collected should span across the entire scope of possible variation. Any non-linear model becomes invalid if it extrapolates beyond the range of process variation that it is based on.

Thirty variables were taken from the distillation column. The decision on which variables to monitor was made by a team of process and control engineers who were experts in the operation of the column. The variables included a combination of manipulations, disturbances and auxiliary outputs. The final model inputs consisted of just disturbances and manipulations.

The frequency of data collection was determined by considering the dynamic of the process. For model-based predictive control, a dynamic model is required. In order to observe the dynamics, the collective system must subsample at a faster rate than the process time constant. In this example, the process time constants between the major disturbances/manipulations and the column pressure were between seven and eight minutes. Data were therefore collected at minutely intervals.

4.2 Stage 2: Spurious data elimination

Few data collection systems are immune to spurious data. There are many causes of spurious information finding its way into data files. Analyser calibration, computer failure, abnormal operation or manual input errors can all be sources of this problem. For dynamic modelling it is extremely important to determine whether potentially spurious data is truly spurious or if it is an actual rapid transition in a process measurement. At this stage, however, in order to examine cause and effect relationships, all spurious spikes can be removed without too much attention being paid to whether or not they are real. This is because real 'spikes' in process parameters will manifest themselves as impulse responses (rather than major changes in steady-state operation) which, provided you have enough 'good' data, should not dominate the statistical analysis.

This 'spurious' data can be removed from data by implementing the following algorithms:

(i) Define upper and lower bounds for each measurement. Any measurement exceeding these bounds can be reset to the previously 'acceptable' value.

(ii) Define a maximum allowable change between one sample and the next and also the maximum width of any potential 'spike' in the data. Any points identified exceeding these conditions can be reset to the previously 'acceptable' value.

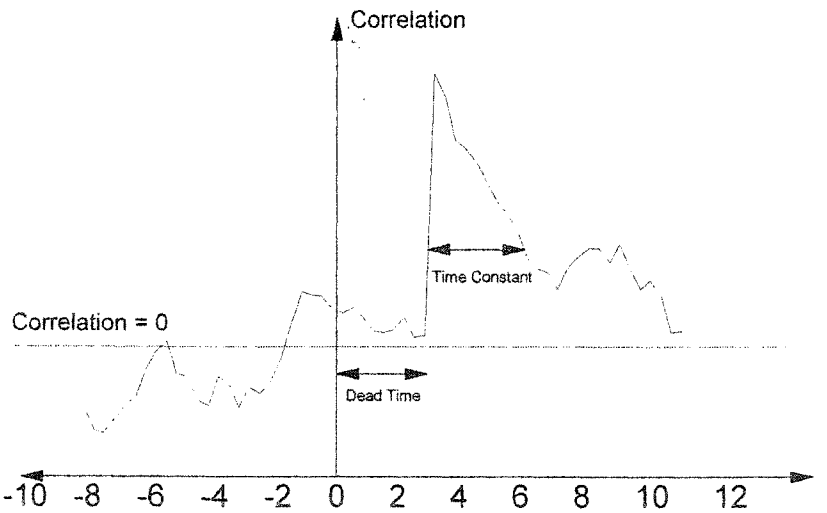


Fig 2 Typical cross-correlation curve

The elimination of any 'spikes' in the data can be achieved by either resetting the values to a previously acceptable value or by interpolating between points either side of the spurious points. Eliminating the data and therefore reducing the overall size of the data set is undesirable as this can result in valuable information being lost in other variables in the data set.

4.3 Stage 3: Cross-correlation analysis

This is the first statistical analysis technique that is employed in order to examine the dynamic characteristics of cause and effect relationships in the data. The cross-correlation coefficient is a statistical measure of the correlation between two variables x and y. The value of the coefficient is calculated as follows:

$$c = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{n\sigma_x\sigma_y} \quad \ldots(1)$$

where

$$\sigma_x = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}} \quad \text{and} \quad \sigma_y = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \bar{y})^2}{n}}$$

where c is the cross-correlation coefficient.

It can be observed from the above expression that if x and y are completely uncorrelated (ie, as x varies, y remains constant), the above expression simply becomes the summation of x minus its mean. Since $w\sum_{i=1}^{n} (x_i - \bar{x})$ (where w is a constant associated with the steady state value of y) will always equal zero, the cross-correlation coefficient equals zero. If x and y are 100% positively correlated, then we can substitute

$$\frac{x_i - \bar{x}}{\sigma_x} = \frac{y_i - \bar{y}}{\sigma_y}$$

for all i and the above expression for c cancels to equal $+1$. Similarly, if x and y are 100% negatively correlated, the expression cancels to -1. In summary, the cross-correlation coefficient can take any value between -1 and $+1$, the value of which indicates the degree and direction of correlation.

The cross-correlation analysis is completed by calculating a series of cross-correlation coefficients on sequentially timeshifted data. This is performed by shifting one of the variables in one direction and repeating the above calculation on the overlapping data. This is repeated until approximately 90% of the data are still overlapped. The procedure is then repeated by time shifting in the opposite direction. The result is known as a 'cross-correlation curve' which gives a time history of the correlation between two variables. These curves can be used to identify dead times, process time constants and feedback effects. Fig 2 displays a typical cross-correlation curve and Fig 3 displays the effect that time delays can have on a scatter plot between two variables. Without timeshifting, there is no apparent correlation between the two variables (Fig 3(a)). When one of the variables is timeshifted so that the 'cause' and 'effect' coincide, the scatter plot reveals a general trend with a positive gradient (Fig 3(b)). The scatter plot is still by no means a perfect straight line. Other causes of variation such as dynamic lag, process noise, non-linearities and multivariable effects can hinder observation of correlations in data. More powerful techniques such as Principal Component Analysis are required in these cases.

Correlation peaks on the right-hand side of $t = 0$ indicate that the causal variable (x) leads the effect variable (y) as should be expected. Peaks on the left hand side indicate that the specified effect variable is actually leading the causal variable. This can often occur in closed-loop systems where the effect variable does have an impact on a causal variable via closed-loop control action.

This procedure was carried out on all 29 process variables correlating them against the pressure measurement. Highly correlated variables were recorded and time delays identified. Time delays were removed from the data by timeshifting them accordingly.

The statistics outlined in this methodology are extremely useful in analysing and understanding process behaviour. However, the statistics cannot be applied without the support of a good engineering understanding of the process being investigated. A classic example of this occurred when the distillation column pressure was correlated against the vent flow at the top of the column. Fig 4 displays the cross-correlation curve from this analysis. As may be expected, there is an extremely

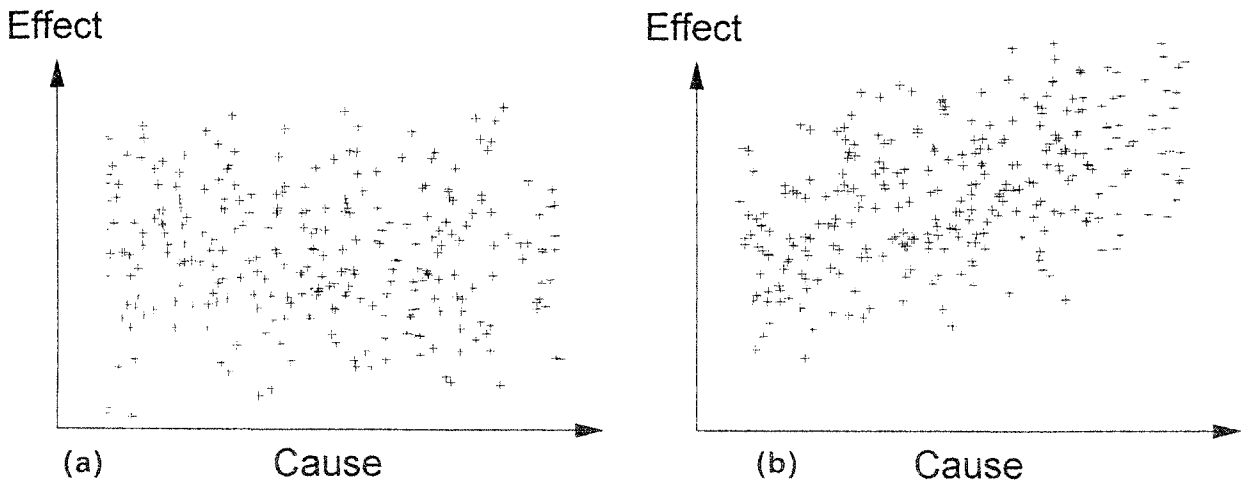


Fig 3 (a) Scatter plot with time delay of 15 samples; (b) Scatter plot with time delay removed

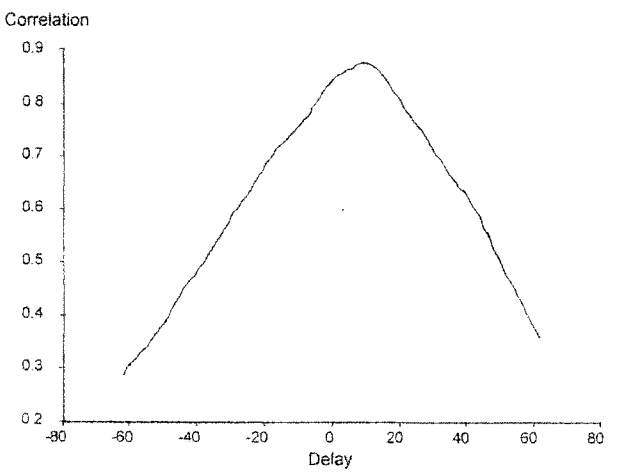


Fig 4 Cross-correlation curve of vent flow to column pressure

strong correlation between vent flow and column pressure with the vent flow changes leading pressure changes. However, the observed correlation is positive which indicates that as the vent flow is increased, the column pressure increases. This does not comply with engineering knowledge or even basic common sense. Opening a vent in a pressurised column should cause the pressure to decrease!

An understanding of the control system was necessary in order to explain this phenomenon. The vent flow was ratioed to the temperature of the cooling water utilised at the top of the distillation column. When a disturbance occurred (eg, an increase in cooling temperature), the control system knew that this would cause an increase in the pressure in the column. The control system therefore increased the vent flow in order to counteract this effect. Since the control system wasn't perfect, the residual effect was still observed as an increase in pressure. Statistically, every time the vent flow increased so did the pressure. The fact that the increase in pressure was less than it would have been if the vent had not opened, could not be identified statistically. The statistics simply represented the information that was contained in the data, ie, every time the pressure increased, this was pre-empted by an increase in vent flow. A model-based controller could not assume a positive correlation between vent flow and pressure without serious consequences. The system was therefore modelled by assuming that the vent flow would be a permanent fixture on the process and its effects would be an integral part of the dynamics between the disturbance variable and the column pressure. Without further process experiments, the vent flow could not have been used as an actuator in a model-based controller.

4.4 Stage 4: Multivariable data analysis

Cross-correlation analysis, although a powerful tool for identifying dynamics and strongly correlated variables, is nevertheless a univariate analysis technique. Most chemical processes suffer from multivariable interactions and, as such, appropriate techniques are required for identifying these relationships.

Principal Components Analysis (Jackson, 1981), Projection to latent structures (Frank *et al*, 1983) and Principal Component Regression are commonly used multivariable analysis techniques. The main objective here is to reduce the dimensionality of a problem to a sensible number of variables. It is a fallacy to suggest that a neural network can take 30 inputs and produce an accurate and robust model of a real industrial process. The correlations between variables need to be taken into account prior to model building so that confusion over feedback and closed-loop phenomena does not occur.

Principal Component Analysis is a technique used to transpose a data set from its original format to a form of reduced dimensionality. Each Principal Component is a linear combination of the original normalised variables and each Principal Component is orthogonal (uncorrelated) to all other components. The variance of each Principal Component decreases so that the first component has maximum standard deviation and the last component has minimum standard deviation. For highly correlated data, only the first few Principal Components are required in order to describe the vast majority of variation in all of the original variables.

The above procedure is based on the principle of singular value decomposition and can essentially be visualised as a re-orientation of co-ordinate axes in n-dimensional space (where n is the number of original

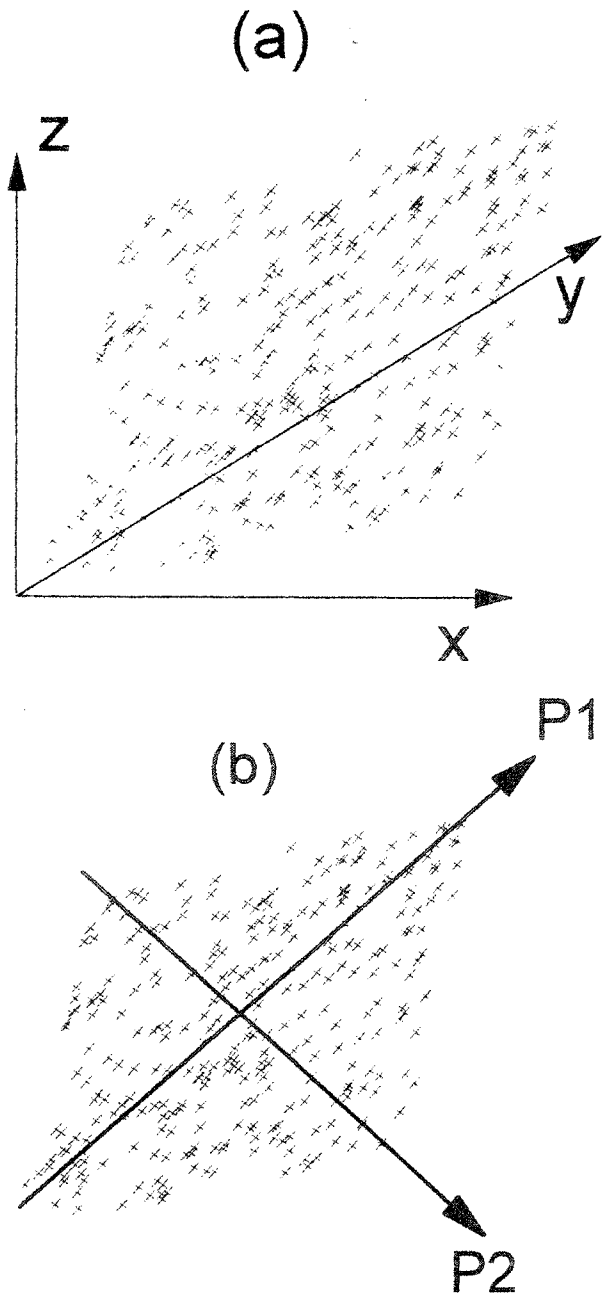


Fig 5 Three-dimensional scatter plot: (a) data points represented by three variable axes; (b) fewer axes needed using Principal Component Analysis

variables). The new axes are aligned so that as many axes as possible lie along 'planes of correlation' in the n-dimensional space. Since all axes are orthogonal, the remaining axes will be orthogonal to these 'planes' and thus contain little or no information about variation in the data. These components can be ignored and the dimensionality of the problem is reduced. This explanation can be made clearer by considering the three-dimensional case. Consider the following relationship between three variables x, y and z.

$$z = x + y \qquad \ldots(2)$$

Fig 5(a) displays a three-dimensional scatter plot of the above relationship. The scattered data lie on the plane as described in Eqn (2). Fig 5(a) uses three variable/axes (x, y and z) in order to represent all of the data points. Principal Component Analysis allows you to represent all of the points in Fig 5(a) but using fewer axes. This is achieved by creating a new coordinate reference frame of lower dimension to xyz. This can be illustrated by considering a new set of axes *P1*, *P2* and *P3* which are linear combinations (ie, re-orientations) of the original x, y and z axes. Fig 5(b) shows where these axes may be aligned. *P1* points in the direction of maximum variation in the data, *P2* is orthogonal to *P1* and contains smaller variation along its axis and *P3* is normal to the plane and thus has no variation about its axis.

Fig 5(b) displays a two-dimensional plot of *P1* against *P2*. All 100 data points appear on this plot. No information has been lost yet the dimensionality of the problem has been reduced from three variables to two new 'latent' variables *P1* and *P2*.

The algorithm has identified correlations between x, y and z. The coefficients that describe the axes *P1*, *P2* and *P3* contain information concerning these correlations. This information can be used to identify correlations in the data. For more information, consult Jackson (1981).

Principal Component Regression simply uses Principal Components as opposed to the original variables in a Multiple Linear Regression equation which maps input variables to output variables. The contributions of original variables to the model can be easily extracted from the coefficients generated.

The 30 original distillation column variables were reduced to just three key variables using the above techniques. The proposed inputs to a neutral network were as follows:

(i) Reboiler Energy (Manipulated Input)
(ii) Column Feed Flow (Disturbance)
(iii) Cooling Water Inlet Temperature (Disturbance)

All of the above techniques are based on linear algorithms. The relevance to non-linear systems is therefore a sensible question. Although these techniques are linear in their structure, they can be used to identify non-linear relationships provided that the non-linearity has a continuous positive/negative correlation between input and output.

For example, the cross-correlation coefficient between a variable z and some non-linear transformation of this variable such as the sigmoid

$$\left(\frac{1}{1+\exp(-z)}\right),$$

produces a correlation coefficient exceeding 0.95. Similar results are obtained for quadratics, exponentials and in fact any function with a constant sign derivative. The sigmoid of z, although not a linear function, always increases as z increases. This is identified by cross-correlation.

4.5 Stage 5: Process modelling

A filter-based dynamic neural network was trained using the above three inputs to model column pressure. As a rule of thumb, the number of hidden nodes in the network was initially set to equal the number of inputs.

This was found to be sufficient for modelling purposes. The network was trained using Levenburg-Marquardt optimisation (Marquardt, 1963), which is a well established non-linear least squares optimisation algorithm. Data from the real process were used for modelling as well as data from a validated Speedup dynamic simulation of the distillation column. This non-linear model had been developed and validated by ICI personnel over a number of years and was regarded as an acceptable representation of the real process. For modelling purposes, data from the real process were used to perturb the column vent, the bottoms flow, reboiler energy, cooling water temperature, product flow and feed flow. The simulation was then used to predict the effects these conditions would have on the distillation column pressure. The results were similar to those observed on the real process. Fig 6 displays a comparison between a linear dynamic model of the pressure and the neural network prediction. The linear model was a time-series, finite impulse response model which was generated with as much attention as the neural network model. The error on the linear model exceeds $\pm 12\%$ whereas the error on the neural network model is less than $\pm 0.5\%$.

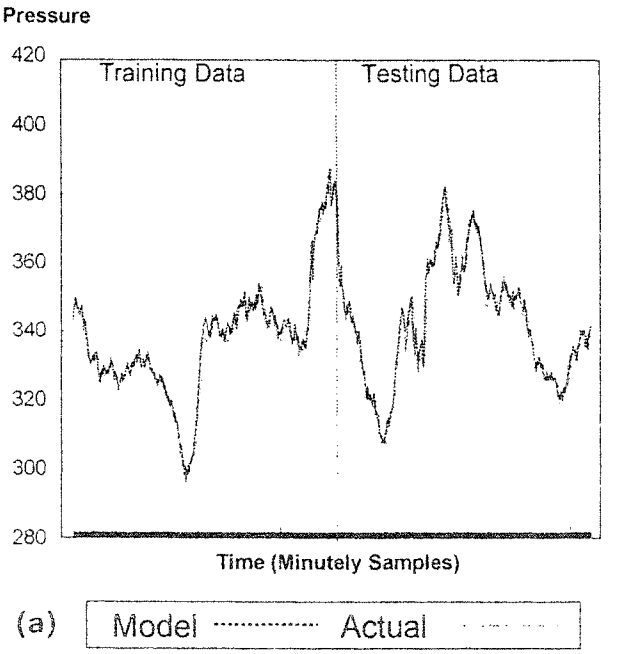
The inputs and calculated dead-times for both the linear model and neural network model were identical. All disturbance/feedforward information was available to both models.

4.6. Stage 6: Model validation

The stages of neural network modelling usually consist of training, testing and validation. The training stage is the optimisation of network coefficients to minimise the error between predicted outputs and actual outputs. The testing stage is the cross-checking of the error during training with an 'unseen' data set to check for over-training. Training ceases when the error on the testing data is seen to be a minimum. Validation is usually just a final application of the model to a third data set to show that the model fits this completely 'unseen' data set. This is *not* model validation, this is 'fitting' validation. There are many reasons why a good fit to data can be observed even though the model is totally unacceptable. A few of these reasons are listed below. Consider a simple single input (u), single output (y) modelling exercise:

(i) The model $y(t) = y(t-1)$ usually a good 'fit' to dynamic data but is totally unsuitable for prediction. For example, a good model of the weather may be that the weather in half an hour will be exactly the same as it is now. The rms error in this prediction over say 1000 hours would be very small (a good solution for least square regression). However, the model would be totally inappropriate for predicting the weather tomorrow.

(ii) A model of feedback $u = fn(y)$ can often give a better 'fit' than the actual true process relationship $y = fn(u)$. A model based on feedback usually assumes an incorrect correlation between u and y (as discussed in Section 4.3) and could be disastrous in a model-based control scheme. For example, a good 'fit to data' may be observed if a graph of someone's monthly expenditure is overlapped with their monthly income – an obvious positive correlation (The more money you earn, the more you spend). Unfortunately, this does not mean that the more money you spend (manipulated variable), the more monthly income you will receive (effect variable?). The observed position correlation between manipulation and effect is feedback.



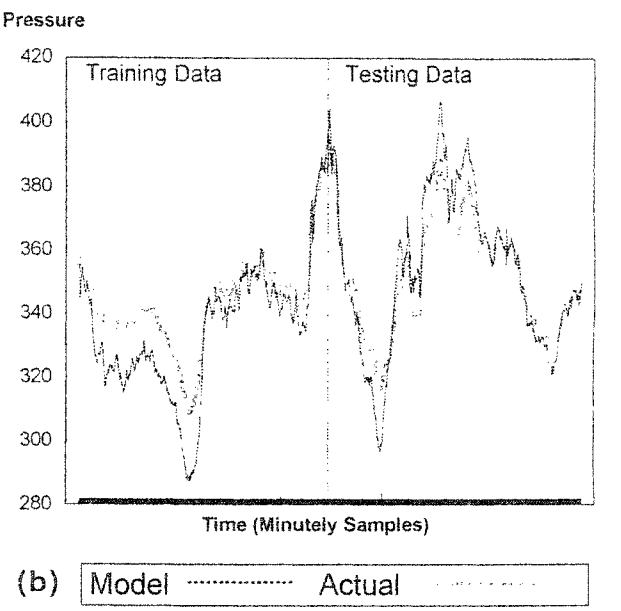


Fig 6 (a) Dynamic neural network model of distillation column pressure; (b) linear model of distillation column pressure

In other words, for control it is necessary to understand the true dynamic relationship between a control actuator and the response variable, not an apparent one that exists because of inadequate data pre-processing.

A more accurate method of model validation is to use the model to predict the step response of the process and

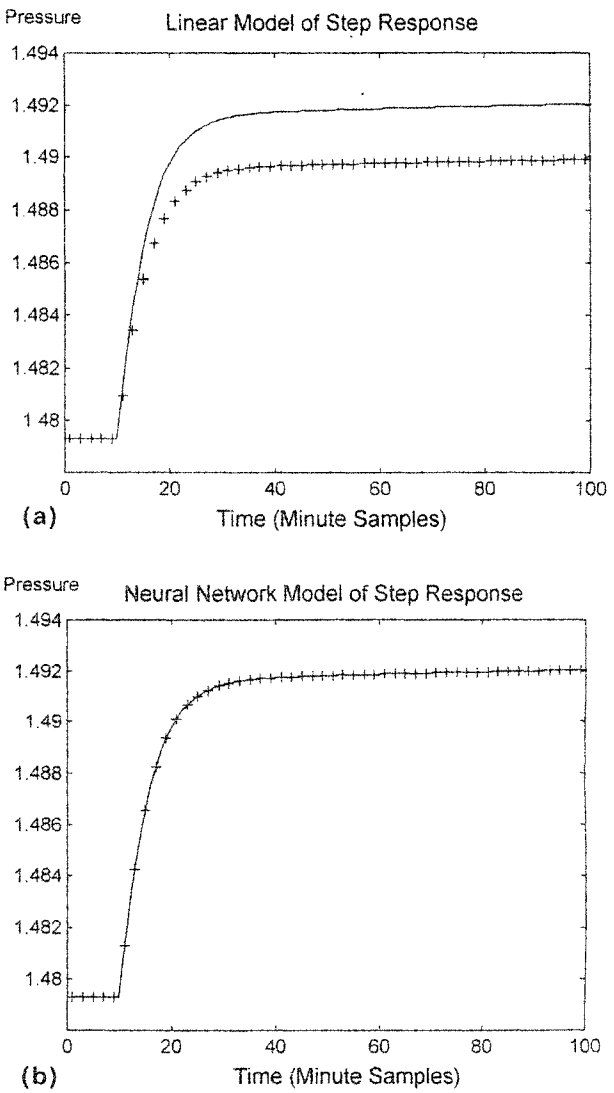


Fig 7 Step response between reboiler energy and column pressure: (a) linear model; (b) neural network model

check that the result makes physical sense. A model based on either of the above scenarios would fail this test. Model (i) would simply not respond to a step change in the input variable u and model (ii) would respond in the opposite direction to that expected.

Fig 7 displays the prediction of the step response between the manipulated variable (reboiler energy) and the column pressure. The linear model predicts the process time constant to a reasonable accuracy but fails to predict accurately the gain of the system. The neural network predicts both time constant and gain with acceptable accuracy.

4.7 Stage 7: Model-based predictive control

The control algorithm implemented on the Speedup model was the same for both the linear and neural network process models. At each stage of execution, the next control action was calculated according to the optimum trajectory back to setpoint. This trajectory was calculated by evaluating the movement in the manipulated variable over a given prediction horizon necessary to return the pressure back to setpoint while minimising the following objective function:

Model-Based Controller Objective Function:

$$J = \frac{1}{2}\sum_{k=1}^{\text{horiz}_y} W_y\left(y_m(k) + off - y_{set}(k)\right)^2 + \frac{1}{2}\sum_{k=1}^{\text{horiz}_u} W_u\left(\Delta u(k)\right)^2 \quad \ldots(3)$$

Where:

W_y, W_u = Controller weights
y_m = Model output
y_{set} = Setpoint
off = Model offset
horiz_y = Output horizon
horiz_u = Input horizon
$u(k)$ = Manipulated variable
$\Delta u(k)$ = $u(k+1) - u(k)$

The offset is evaluated as the mean of the error between the model predictions and the actual column pressure over a period equal to the time constant of the process.

$$off = \sum_{t=t-\tau-1}^{t=t-1}\left(\frac{y(t) - y_m(t)}{\tau}\right) \quad \ldots(4)$$

The offset is required in order to account for deterioration or inaccuracies in the process model. The offset is important to the objective function since it is a potentially varying property of the process and affects the relative weightings attributed to the model output and the model setpoint.

Once the optimum trajectory has been evaluated, only the first calculated movement in the manipulated variable is actually initiated. The entire optimisation process is repeated at each time step.

An important consideration in non-linear model-based predictive control is the length of time it takes to optimise the objective function. For the control system to work, the minimum has to be identified in a time faster than the period between control actions. It was found in this application that once a local solution had been obtained by the optimiser, further solutions were found in a matter of seconds compared with control action occurring every minute. A simple steepest-descent algorithm was used to find the minimum. However, this may not be the most appropriate algorithm to use in a true on-line application.

Fig 8 shows the comparison of the performance of a robust PI controller, the linear model-based controller and the neural network model-based controller. The model-based controllers are seen to outperform the traditional PI controller by approximately 50%. The neural network controller reduces the pressure standard deviation by a further 50% over the linear model-based controller. The benefits associated with reducing the pressure from conventional PI to model-based control exceed AUS$700k per annum. The 50% further reduction from linear model-based control to neural network is worth a further AUS$300k per annum. This is significant when considering that the neural network model took only as long to generate as the linear model.

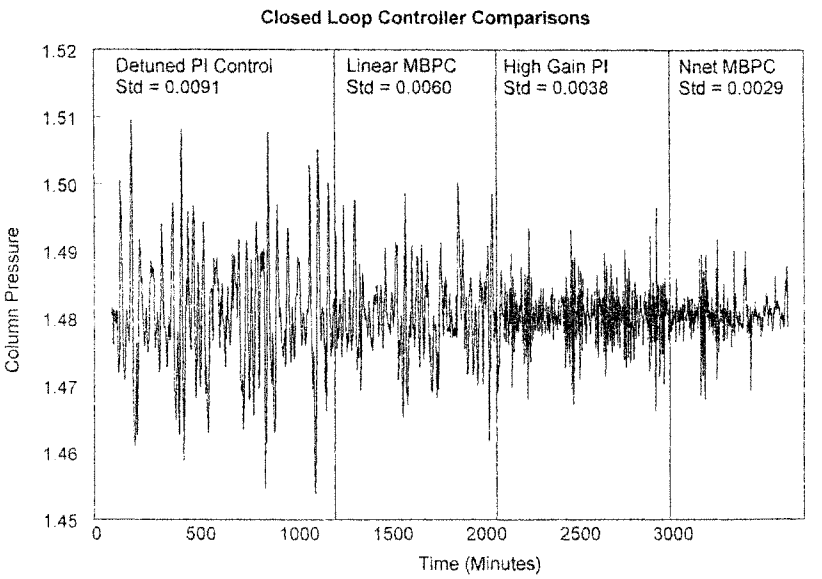


Fig 8 Closed-loop controller comparisons

5. Conclusions

A procedure has been described for implementing neural-network-based control to real processes. Neural networks are often claimed to be the panacea to all our problems. Analogies to the workings of the human brain supplement the recondite propaganda that is far too often over-exaggerated. This paper has adopted a practical methodology for the application of neural networks to chemical processes. They are treated as powerful non-linear approximators and implemented with engineering judgement and validation. The combination of statistics, engineering understanding and neural networks removes some of the mystique and 'black-box' nature attributed to this technology.

Neural network model-based control is seen to outperform linear control algorithms on non-linear processes. This should yield little surprise since neural networks are inherently non-linear. However, the ease of implementation and understanding of the predictive capability of the neural model is a major step forward in their potential application to real processes. Many of the major control companies are now actively incorporating neural networks into their software. In several years time neural network models in control could be as standard as linear models.

There is, however, a word of warning. Neural networks cannot be used to extrapolate beyond the range that they were trained on. Extrapolation is not simply a case of exceeding maximum and minimum values. Extrapolation can occur within the boundaries of a defined training data input space. Regions or 'pockets' of the input space may exist where little information was available during training. Research is currently being made into identifying these regions so that appropriate actions can be made if a network extrapolates on-line.

6. Acknowledgements

The authors wish to express their thanks to the following people and institutions. Their help and support for this project is greatly appreciated: ICI Australia Pty Ltd; The Royal Academy of Engineering, UK; The Teaching Company Directorate of Great Britain.

7. References

Agamennoni, O. and Romagnoli, J. 1993. 'DICOF: A package for multivariable controller design in the frequency domain', *Cache News*, 36, Spring.

Bhat, N., and McAvoy, T. J. 1990. 'Use of neural nets for dynamic modelling and control of chemical process systems', *Computer Chem Engng*, 14(4/5), 573–583.

Bhat, N., Minderman, P., and McAvoy, T. J. 1989. 'Use of neural nets for modelling of chemical process systems', IFAC Symp on Dycord+89, Maastricht, The Netherlands, 21–23 August, 1989. 147–153.

Birky, G. J., and McAvoy, T. J. 1989. 'A neural net to learn the design of distillation controls', IFAC Symp on Dycord+89, Maastricht, The Netherlands, 21–23 August. 205–213.

Cybenko, G. 1989. 'Approximation by superposition of sigmoidal function', *Math Control Signal Syst* **2**, 303–314.

Eskinat, E., Johnson, S., and Luyben, W. 1991. 'Use of Hammerstein Models in identification of non-linear systems', *AIChE Journal*, **37**(2), Feb.

Frank, I. E., Feikema, J., Constantine, N., and Kowalski, B. R. 1983. 'Prediction of product quality from spectral data using the partial least squares method.' *J of Chemical Information and Computer Sci*, **24**, 20–24.

Hernandez, E., and Arkun, Y. 1992. 'Study of the control relevant properties of backpropogation neural network models of non-linear dynamical systems', *Computer Chem Engng*, **16**(4), 227–240.

Hertz, J., Krogh, A., and Palmer, R. 1991. *Introduction to the Theory of Neural Computation*, Addison-Wesley.

Jackson, J. E. 1981. 'Principal components and factor analysis: Part 1–Principal components', *J Qual Tech* **13**(2).

Marquardt, D. 1963. 'An algorithm for least squares estimation of non-linear parameters', *SIAM J Appl Math*, **11**, 431.

Montague, G. A., and Willis, M. J. 1993. 'Artificial neural network model-based control', Advances in Model Based Predictive Control Conference Proceedings, September, Department of Electrical Engineering, University of Oxford, UK. Volume 2, 65–76.

Ungar, L. H., Powell, B. A., and Kamens, S. N. 1990. 'Adaptive networks for fault diagnosis and process control', *Computers Chem Engng*, **14**(4,5), 561–572.

Williams, R. J., and Zipser, D. 1989. 'A learning algorithm for continually running, fully recurrent neural networks', *Neural Computation*, 1270–1289.

Willis, M. J., Massimi, C. Di., Montague, G. A., Tham, M. T., and Morris, A. J. 1991. 'Artificial neural networks in process engineering', *IEE Proceedings-D*, **138**(3), May.