

Control Of pH In-Line Using A Neural Predictive Strategy

J.B. Gomm, S.K. Doherty and D. Williams

Control Systems Research Group, School of Electrical and Electronic Engineering,
Liverpool John Moores University, Byrom St., Liverpool L3 3AF.

Keywords: process control, intelligent control, non-linear systems

Abstract

Control of an experimental in-line pH process exhibiting varying nonlinearity and deadtime is described. A radial basis function (RBF) artificial neural network is used to model the nonlinear dynamics of the process. Accommodation of the varying process deadtime in the neural model is achieved by the generation of a feed-forward signal, for input to the neural network, from a downstream pH measurement. The feed-forward signal is derived from a variable delay model based on process knowledge and a flow measurement. The neural model is then used to realise a predictive control scheme for the process. Development of the neural process model is described and results are presented to illustrate the performance of the neural predictive control scheme which is tested as a regulator at different setpoints.

1. Introduction

An in-line pH process is an alternative approach to the control of pH where the influent stream is treated in the pipeline as opposed to the more commonly used method of using a continuous stirred tank reactor (CSTR). In-line control of pH offers potential benefits and savings in both capital and investment compared to the conventional CSTR approach [1]. As well as the familiar strong nonlinearity associated with pH processes, due to the pH titration curve, in-line neutralisation combines several other difficult process characteristics, such as small process capacitance and flow dependant transport lags. Also, as well as measurement noise, the lack of mechanical mixing in an in-line pH process introduces a further degree of uncertainty associated with the process measurements. The combined effect of all of these features make the in-line pH process a difficult and testing environment for control strategies.

The nonlinear functional approximation

properties of artificial neural networks, together with their other associated features such as their abilities to learn by example and to generalise sensibly to novel inputs, makes them an appealing technique for nonlinear system identification. A neural network can be trained to represent the input-output dynamics of a nonlinear process using examples of process data. This is particularly relevant to chemical processes, where it is often not practical to develop a reliable analytic process model, but where historic process data is often available. Development of a neural network process model subsequently enables a nonlinear control scheme to be realised.

Predictive control is becoming increasingly popular in the chemical process industries because of its relevant and appealing features, such as the ability to accommodate input and output constraints and to handle process deadtimes and multi-variable systems in a straight forward manner, which are not easily achieved by other control strategies. Like other model based control strategies, the success of a predictive control scheme is largely dependant on the validity of the process model. Many chemical processes are highly nonlinear, and consequently predictive control based on linear process models can result in poor performance. Consequently, predictive control has been extended to accommodate nonlinear process models including neural networks [2].

This paper describes the development of a neural network model of a bench scale in-line pH process and the subsequent incorporation of the neural model in a predictive control strategy. On-line control results are presented for pH regulation in the presence of load disturbances.

2. The In-Line pH Process

The experiments described here were performed on a bench scale rig shown in

Fig.1. A wastewater stream of variable flow and pH is produced by merging an acidic feed stream (0.05 M solution of acetic acid), which is pumped at a constant flowrate (F_{feed}), with an alkaline stream (0.1 M ammonium hydroxide), whose flowrate (F_1) is variable. The resulting stream passes through a coil of tubing, which serves to mix the two streams, and then enters the tubular reactor. The pH of the influent to the tubular reactor is measured at pH₁. Only the total flowrate (F_{tub}) through the tubular reactor is measured, the component flowrates F_{feed} and F_1 are not. The stream exiting the tubular reactor is therefore of varying pH and flowrate. In closed loop, the objective is to regulate the pH of this stream (pH₂) by manipulating the flow (F_2) of neutralising reagent (0.1 M acetic acid). The process flowrates allow for setpoints of pH 5 to pH 7 to be physically realisable.

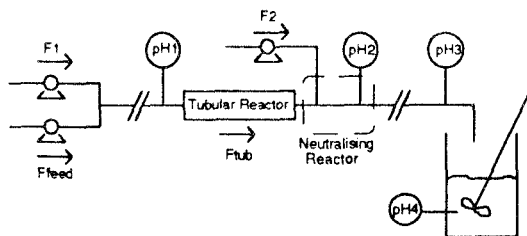


Fig.1 The In-Line pH Process

The pH₂ probe is located close to the neutralising reactor Tee so as to keep the process deadtime as small as possible. The neutralising reactor consists of the Tee junction together with the tubing connecting it to the pH₂ probe housing, and its small size can result in a particularly noisy pH₂ signal since the F_{tub} and F_2 streams will not be well mixed. For this reason another pH measurement (pH₃) is taken downstream of pH₂ for averaging purposes, after the neutralising reactor effluent has passed through a further coil of tubing to facilitate mixing. The bench scale rig also includes a CSTR at the exit of the in-line system which provides a bulk product pH measurement (pH₄) but is not considered for control purposes at present. All flowrates are set by peristaltic pumps which, together with the pH meters, are interfaced with a PC for data acquisition and closed loop control.

The dynamics of the in-line process are dominated by nonlinearity and deadtime. The shape of the familiar neutralisation curve associated with pH depends on the concentrations and strengths of the different

chemical species in the system. For this single acid-base system the gain at pH 7 can be more than 40 times greater than that at pH 5. In this process the pH titration curve is not stationary because the concentrations of both the influent species vary as F_1 changes. This holds if the titration curve abscissa is the neutralisation reagent flowrate (F_2), or if it is the ratio of neutralisation reagent flowrate to influent flowrate (F_2/F_{tub}) as is commonly used. In addition, the variations in F_1 , and hence the total flowrate through the tubular reactor (F_{tub}), result in a variable transport delay between pH₁ and pH₂.

3. Neural Network Process Model Development

3.1. Neural Network Architecture

The neural architecture adopted for identifying the nonlinear pH process dynamics was a radial basis function (RBF) network which have been successfully used for nonlinear modelling (e.g. [3], [4]). The RBF network was chosen because linear algorithms can be used in the training strategy, such as least squares. The RBF network therefore has an advantage of fast convergence properties compared to alternative networks, such as the multi-layer perceptron, which utilise nonlinear learning algorithms. For this reason, RBF networks have the potential for on-line learning and adaptation to track process time variations which is a further research area in this work.

The input/output mapping for the RBF network used was of the form

$$y = \sum_{i=1}^{n_c} w_i \phi_i(\|x - c_i\|) \quad (1)$$

where

$$\phi(z) = z^2 \log_e z \quad (2)$$

x is the input vector, c_i are the centre vectors, w_i are the weights connecting the centres to the network output, n_c is the number of centres, $\phi(\bullet)$ is the thin plate spline function. Several other nonlinear basis functions have been proposed for $\phi(\bullet)$, but the choice does not seem to be crucial to the approximation capabilities of an RBF network. The mapping accuracy of the network was found to be sensitive to the scaling of the data, with best results obtained by scaling all data to (0,100) and distributing the centre positions uniformly in this range.

3.2. Neural Network Model Structure

A neural network can learn a static nonlinear function by training it with data which is representative of the function. Dynamics must be incorporated into the network if it is to be used as a dynamic model. Recurrent neural networks do this directly by introducing dynamics into the network neurons. A more common approach is to represent the process dynamics by a nonlinear difference equation, and the NARX (Nonlinear Auto Regressive eXogenous; [5]) model structure is widely used for this purpose. The multi-input, single output NARX model structure used was of the form

$$\hat{y}(t) = f[u_1(t-k_1), \dots, u_1(t-k_1-n_{u1}+1), \dots, u_m(t-k_m), \dots, u_m(t-k_m-n_{um}+1), \dots, y(t-1), \dots, y(t-n_y)] + e(t) \quad (3)$$

where $u_1 \dots u_m$ are the process inputs, y is the process output, e is the equation error, $k_1 \dots k_m$ are the model deadtimes ($k \geq 1$) and $f[\bullet]$ is a nonlinear function to be identified. The network is trained to provide a one step ahead prediction of the process output, $\hat{y}(t)$.

After training the network can be used on-line to predict future process responses based on known, or estimated, future inputs. This is required in the predictive control scheme where the control action is determined based on future process responses over a receding horizon. Long range prediction is achieved by replacing the future unknown process outputs with the network predictions, \hat{y} , at each time step. To predict N steps ahead at time t , the network makes N separate one step ahead predictions

$$\hat{y}(t+i) = f[u_1(t-k_1+i), \dots, u_1(t-k_1-n_{u1}+i+1), \dots, u_m(t-k_m+i), \dots, u_m(t-k_m-n_{um}+i+1), \dots, \hat{y}(t+i-1), \dots, \hat{y}(t+i-n_y)] \quad (4)$$

$i=1, \dots, N$

Hence, for an N step ahead prediction the network is recurred $N-1$ times. This feeding back of the network output to input can result in an accumulation of the prediction errors and consequently poor multi step ahead prediction accuracy [6]. Often, a network which performs well as a one step ahead predictor can perform poorly as a multi step ahead predictor. This accumulation of one step ahead prediction errors is reduced by making a correction to the feedback \hat{y} by adding the process/model mismatch to it [7].

3.3. Compensation of Variable Deadtime

The pH of the influent to the neutralisation reactor can be approximated by a delayed pH₁ measurement since the tubular reactor acts as a transportation delay. Hence, pH₁ can be used as a feedforward input to the neural network process model. Feeding forward pH₁ through a pure time delay does not produce a pH signal which is in phase with the neutralisation reactor influent pH since the tubular reactor's transportation delay changes as its throughput (F_{tub}) changes. Three methods have been investigated and compared for representing this variable time delay in a neural network model [8]. The approach adopted in the neural predictive controller described is to represent the variable deadtime using a transport lag volume array (TLVA). This technique was observed to provide satisfactory network performance in the comparative tests performed.

In the TLVA method, the deadtime is compensated by representing the transport of fluid through the tubular reactor by an array of lagged pH₁ values, with each array element representing a unit volume. The technique makes use of process knowledge regarding the volume of the tubular reactor. At each sample interval a number of array elements (N_V), corresponding to the volume of influent to the tubular reactor since the previous sample, is calculated as $N_i = (F_{\text{tub}}(t) \cdot \Delta T) / \text{vol}$, where ΔT = sample time and vol = a unit volume. A feedforward signal, $\text{pH}_{1\text{ff}}$, is calculated as the mean value of the last N_V elements of the delay array. All array elements are now shifted forward N_V places, corresponding to the reactor throughput since the last sample. Finally, the current value of the delayed variable, $\text{pH}_1(t)$, is copied to the first N_V array elements.

3.4. Process Excitation Signal

Process input-output data for network training was collected in open loop by applying an excitation signal to the process input. F_2 . A random amplitude signal (RAS) has been widely used to provide suitable excitation for nonlinear system identification. A conventional RAS consists of a uniformly distributed random input bounded by the limits of the process input operating range. However, when a highly nonlinear process, such as a pH process, is excited using a RAS, little output data is generated in the high gain

region leading to poor predictions from a network trained with this data. One practical way of improving the quality of the training data, and consequently the network prediction accuracy, is to force the process output through the high gain region on each clock pulse [9]. Hence, this 'forced' RAS is uniformly distributed above and below some threshold which, for the in-line pH process, was chosen to be the steady state for the high gain region.

3.5. Sample Time Selection

Appropriate selection of a data sample time is an important aspect for the reliability of an identified process model. In this work, several neural models were identified with data collected at different sample times. It was generally observed that faster sample times gave improved network performance in terms of the mean square error (MSE) on test data. However, further investigation showed that whilst some of the networks exhibited small prediction errors, they failed to adequately model the cause and effect relationship between the manipulated variable and the process output. This aspect is important since a predictive controller utilising one of these models would adjust the manipulated variable in the wrong direction, as suggested by the model, in order to affect the control.

A sample time of 2 secs. was chosen for the identification of the in-line pH process. This was selected on the basis of the 95% process step response times, which range from 15 to 70 seconds, and the manipulated variable deadtime, which is also flow dependant, and was estimated to range from 3 to 4 seconds. Hence, for the model to predict the process response to a change in the manipulated variable it must be recurred at least once, eqn.(4). Model validation tests confirmed that a model employing a 2 sec. sample time was capable of adequate network predictions and could also accurately predict the correct direction of the process output trajectory to step changes in the manipulated variable. This accuracy deteriorated for smaller sample intervals, which required more than one recursion to predict over the manipulated variable deadtime, as it did for longer sample intervals, where the model could predict the direction of the process output in one step.

3.6. Selection of the Neural Network Model Inputs

When a neural network is configured to

represent a NARX model, the determination of the model orders and delay in the NARX model, $n_{u1}, \dots, n_{um}, k_1, \dots, k_m, n_y$ in eqn.(3), is equivalent to determining the input node assignments for the neural network. There are no established procedures for determining the size of the NARX model required for an individual problem. It is generally recognised that a parsimonious neural network is desirable because over parameterisation leads to an unnecessarily large network, with subsequent increased computational overheads, and can also degrade a neural network's generalisation abilities. Determination of a suitable NARX model structure can be simplified by setting $n_{u1} = \dots = n_{um} = n_y = n$, where n is termed the order of the model. When a suitable model order has been chosen the individual n_u and n_y values can then be further adjusted if necessary.

Selecting the model order may be guided by knowledge of the process, however it is usually necessary to test networks with different model orders to establish the best NARX model structure. Several statistical metrics from linear system identification, such as Akaike's Final Prediction Error Criterion (AFPE), can be employed to assist the selection of a parsimonious model structure [10]. The AFPE attempts to penalise marginal reductions in the mean squared prediction error of a model which occur from increasing the model order. Thus, the trade off between model size and accuracy is clarified by comparing the AFPE for different network models.

From knowledge of the process operation, a RBF neural network was configured as a NARX model to predict the process output, $pH_2(t)$, based on lagged values of pH_2 , F_2 , F_{1ub} , and pH_{1ff} . Forced RASs were applied to F_1 and F_2 to obtain identification data, sampled at 2 sec. intervals, for training and testing different neural network process models. The model deadtime for the manipulated variable, F_2 , in all of the trained networks was 2 samples, corresponding to the known delay for the process. Based on some preliminary results, the model deadtime for F_{1ub} was selected as 1 sample. The TLVA effectively removes the varying process deadtime from the feedforward signal pH_{1ff} , hence a delay of 1 sample was used for this signal. This small delay is mainly due to the dynamics of the pH_2 measurement sensor which are not insignificant for the process at the sample time used.

Neural networks with different model orders, $n = n_{Ftub} = n_{F2} = n_{pH1} = n_{pH2}$, were compared using the AFPE criterion to choose an initial value for n , Fig.2. This indicated $n=2$ as an appropriate model configuration. The model orders were then individually fine tuned to observe if a smaller network could be obtained without significant loss of prediction accuracy. The final network configuration chosen was $n_{Ftub} = 1$, $n_{F2} = 2$, $n_{pH1} = 1$ and $n_{pH2} = 2$.

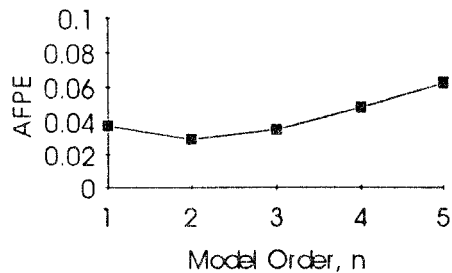


Fig.2 Selection of neural network model order using AFPE

5. Predictive Control of the In-Line pH Process

In predictive control the control inputs are determined by minimisation of a performance index based on evaluating the effects of different inputs on predictions of the process output up to a fixed horizon. The predictions of the process output are obtained by using a dynamic process model, in this case the identified neural network model described in the previous section. In this application to the in-line pH process, the predictive control scheme naturally takes account of the effects of future process disturbances by the use of the feedforward signal pH_{1ff} in the neural network model. Thus, the control scheme can apply early compensation to the process disturbances reducing their effect which is a desirable attribute.

The predictive control scheme is illustrated in Fig.3. The controller, C, computes the control input, $u(t)$, using an iterative optimisation algorithm to minimise the following cost function at each sampling instant, t :

$$J(N1, N2, Nu) = \sum_{i=t+N1}^{t+N2} [r(i) - \hat{y}(i)]^2 + \lambda \sum_{i=t}^{t+Nu} [u(i) - u(i-1)]^2 \quad (5)$$

where r is a modified process setpoint, \hat{y} is the neural network model predicted output, u is the process input, $N1$ and $N2$ define the prediction horizon, Nu is the control horizon and λ is a control weighting factor. A suitable choice for $N1$ is to make it equal to the process delay between the manipulated variable and the process output, k in eqn.(3). $N2$ is then set to define the prediction horizon beyond this point. The purpose of the weighting factor, λ , is to penalise large changes in the manipulated variable, the severity of the penalty is set by the value of λ .

Differences between the process and model that occur in practice are compensated by error feedback as shown in Fig.3. The low pass filter improves the robustness of the scheme by attenuating the effects of random disturbances, and the filtered error is also used to correct the predictions from the neural network model as described in section 3.2. Often, pH_2 oscillates around the setpoint, particularly for setpoints near the high gain region ($pH 7$). The uncertain effects of mixing in the pipeline and asymmetry of the pH nonlinearity around the setpoint can result in an offset in the downstream pH, measured at pH_3 . Hence, there is an additional feedback path for a filtered pH_3 signal to reduce the offset at pH_3 from the required set point.

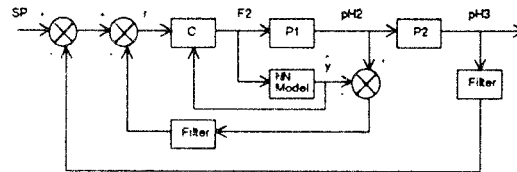


Fig.3 Neural Network Predictive Control Scheme for the In-line pH Process

The predictive control scheme was tested as a regulator for setpoints between $pH 5$ and $pH 7$ for a variety of disturbances in F_{tub} and pH_1 . A control horizon of $Nu=0$ was used as this significantly reduced the amount of computation required by the nonlinear optimiser to find a minimum of the cost function, eqn.(5). Fig.4 shows on-line results for regulation at a setpoint of $pH 6$ with $N1=2$, $N2=3$ and $\lambda=0$. After initialisation of the control algorithm pH_2 is maintained within ± 1.2 pH units of the setpoint. The erratic nature of pH_2 , caused by the close vicinity of the measurement probe to the neutralising injection point (section 2), is evident in the figure. The effect of the control action is

however, more evident in the downstream pH₃ measurement by which time the chemicals are better mixed. This signal is much less erratic than pH₂, due to the averaging effect of the improved mixing, and shows improved control of the effluent pH within ± 0.5 pH units from the setpoint. The quality of the control was observed to deteriorate at higher set points, where the process gain is at its maximum, with pH₃ maintained within ± 1 pH units for the maximum setpoint of pH 7.

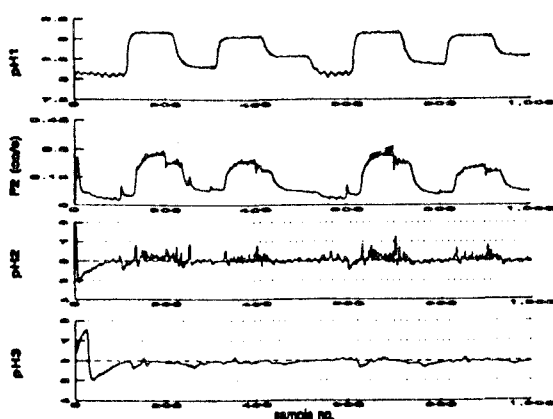


Fig.4 On-line results for predictive control of the in-line pH process

Increasing λ from 0 had the expected effect of smoothing the control input, F_2 , and thus, reducing the fluctuations in pH₂. However, making λ too large caused the control input to saturate. Increasing the prediction horizon, N_2 , also had the effect of reducing the control effort and prediction horizons between 3 and 5 gave satisfactory control over the range of setpoints tested. The control weighting factor, λ , needed to be tuned separately to obtain the best results for each prediction horizon.

6. CONCLUSIONS

The development and performance of a neural network predictive control scheme for an in-line pH process was described. The process exhibits significant nonlinearity, which is time variant, and variations in deadtime. A method for compensating the variable process delay in a neural network model was described and results demonstrating regulatory control of the process in the presence of load disturbances were presented. The performance of other neural network control strategies for the process and application of the controllers to a pilot-scale in-line pH process are currently

under investigation.

7. ACKNOWLEDGEMENTS

This work is funded by Liverpool John Moores University Research Fund and British Nuclear Fuels plc (Sellafield).

8. REFERENCES

1. R.R. Rhinehart and J.Y. Choi, Process model based control of wastewater pH neutralization, *Advances in Instrumentation*, 1988, **43**, 351-388.
2. Hunt, D. Sbarbaro, R. Zbikowski and P.J. Gawthrop, Neural networks for control systems - a survey, *Automatica*, 1992, **28**(6), 1083-1112.
3. S. Chen, S.A. Billings, C.F.N. Cowan and P.M. Grant, Practical identification of NARMAX models using radial basis functions, *Int. J. Control*, 1990, **52**(6), 1327-1350.
4. M. Pottmann and D.E. Seborg, Identification of non-linear processes using reciprocal multiquadratic functions, *Journal of Process Control*, 1992, **2**(4), 189-203.
5. I.J. Leontaritis and S.A. Billings, Input-output parametric models for non-linear systems, parts 1 and 2, *Int. J. Control*, 1985, **41**, 303-344.
6. J.B. Gomm, P.J.G. Lisboa, D. Williams and J.T. Evans, Accurate multi-step ahead prediction of non-linear systems using the MLP neural network with spread encoding, *Trans.Inst.M.C.*, 1994, **16**(4), 203-213.
7. M. Pottmann and D.E. Seborg, Proc. IFAC Conf. DYCORN '92, 1992, 351-356.
8. S.K. Doherty, J.B. Gomm and D. Williams, Neural network identification of an in-line pH process, Proc. IFAC Conf. DYCORN'95, Helsingor, Denmark, 1995.
9. S.K. Doherty, J.B. Gomm and D. Williams, Practical considerations on the implementation of neural networks for non-linear system identification, Proc. IEEE/IMACS Int. Symp. Sig. Proc., Robotics and Neural Nets (SPRANN'94), Lille, France, 1994, 564-567.
10. S.K. Doherty, J.B. Gomm, D. Williams and D.C. Eardley, Design issues in applying neural networks to model highly non-linear processes, Proc. IEE Int. Conf. Control '94, Warwick, UK, 1994, 1478-1483.