

Concentration Control of CSTR using NNAPC

Hossein Ebadi Kalhoodashti
Department of Electric Engineering,
Science and Research Branch,
Islamic Azad University, Sari, Iran

ABSTRACT

This paper presents a new advanced control algorithm for Concentration tracking of a continuous stirring tank reactor (CSTR). This algorithm called: Neural Network Approximate Generalized Predictive Control (NNAPC) that uses a combination of Artificial Neural Network (ANN) with Approximate Generalized Predictive Control technique (APC). This algorithm is based on the use of ANN as a nonlinear prediction model of the CSTR. This modeling technique is done by using the data from the system input/output information without requiring the knowledge about CSTR parameters. The outputs of the neural predictor are the future values of the controlled variables needed by the optimization algorithm. Simulation results show the effectiveness of the proposed control method.

Keywords

Continuous Stirring Tank Reactor (CSTR), Approximate Generalized Predictive Control (APC), Artificial Neural Network (ANN)

1. INTRODUCTION

Often chemical reactors have significant heat effects, so it is important to be able to add or eliminate heat from them. In a Continuously Stirred Tank Reactor (CSTR) the heat is add or removed by virtue of the temperature difference between a jacketed fluid and the reactor fluid. The heat transfer fluid often is pumped through agitation nozzle that circulates the fluid through the jacket at a high velocity. The reactant conversion in chemical reactor is a function of a residence time or its inverse, the space velocity. The product concentration for a CSTR can be controlled by manipulating the feed flow rate, which changes the residence time for a constant chemical reactor [11].

Traditionally, the predictive control method is used for industrial process control and a large number of implementation algorithms have been presented in literature such as extended prediction self adaptive control, generalized predictive control and unified predictive control [3]. The Approximate Generalized Predictive Control belongs to a class of control methods called Model Based Predictive Control (MBPC). The MBPC algorithms use an explicit process model to predict the future behavior of the plant. The classical APC had been developed with linear plant for prediction model which leads to a formulation that can be solved analytically. The use of linear model predictor becomes impractical, if the process is defined by a nonlinear model, and the design of nonlinear predictor and nonlinear algorithm for optimization are necessary.

The use of artificial neural networks for nonlinear system modeling a control has proved to be extremely successful because of their ability to learn the dynamics of the plant, adaptability to a changing environment, and their robustness

with respect to noise. In this paper, the neural model is single variable, where the input of the ANN is the coolant flow and the output is the concentration of CSTR. The ANN model is carried out offline and the control algorithm does not need to perform it continuously. Therefore, the calculation burden will be only for optimization.

This paper proposes an application of Neural Network Approximate Generalized Predictive Control for CSTR. The nonlinear prediction model of the CSTR is designed by an ANN. A Newton based Levenberg-Marquardt method is used for optimization from the model of the CSTR. The objective is to track concentration profile.

2. IDENTIFICATION OF CSTR

The procedures which must be executed when attempting to identify a dynamical system consist of four basic steps (see Figure 1):

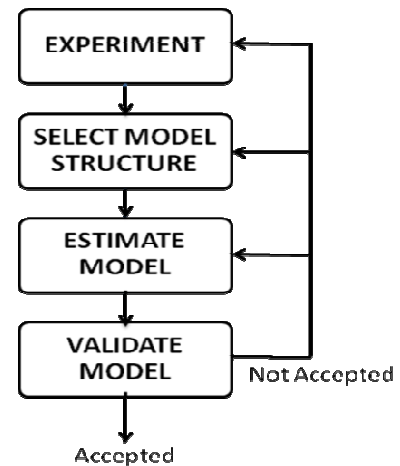


Fig. 1: The system identification procedure

2.1 Experiment

The input signal and output from the system are used as identified data.

The experiment data obtained with proper choice of sampling frequency:

$$Z^N = \{[u(t), y(t)], t = 1, \dots, N\} \quad (1)$$

$\{u(t)\}$ is no longer just a set of inputs but is now a signal, the control signal. Similarly $\{y(t)\}$ now represent the measured output signal. ' t ' specifies sampling instant number t.

Fig. 2 shows the result of this experiment only for the training data set.

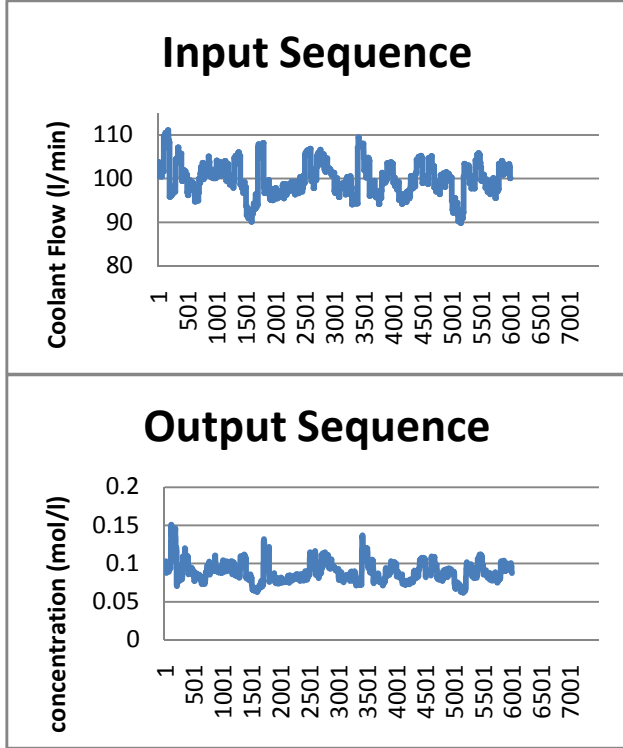


Fig. 2: Result of Experiment

2.2 Select Model Structure

The aim of using ANN to model a nonlinear system is to build a mathematical model which can be used as nonlinear predictor design. By giving some prior knowledge about the system and information on input and output, the ANN without requiring the knowledge of CSTR parameters can accurately describe the nonlinear behavior of the CSTR [2].

After the experiment part, the next step is to select a model structure. Unfortunately, in the nonlinear case this issue is much more difficult than in the linear case. Not only is it necessary to choose a set of regressors but also a network architecture is required. The idea is to select the regressors based on inspiration from linear system identification and then determine the best possible network architecture with the given regressors as inputs.

A neural network autoregressive external input (NNARX) $g[\cdot]$ is used to approximate the input/output model of the CSTR. This is basically a one-step ahead prediction structure in which we used past inputs and outputs to predict the current output. The feedforward structure is used for application in neural-predictive control [7-9], where $[\cdot]$ contains data from the plant.

Regression vector:

$$\varphi(t) = [y(t-1), \dots, y(t-n_a), u(t-n_k), \dots, u(t-n_b-n_k+1)]^T \quad (2)$$

Predictor:

$$\hat{y}(t | \theta) = \hat{y}(t | t-1, \theta) = g(\varphi(t), \theta) \quad (3)$$

$\varphi(t)$ is a vector containing the regressors, θ is a vector containing the weights and g is the function realized by the neural network. When a particular model structure has been selected, the next choice, which has to be made, is the number of past signals used as regressors, i.e., the model order. The

NNARX function is always preferred when no or only very little noise is present. Different value of lag spaces in input and output data is tested with Lipschitz ratio. This method is not always equally successful but sometimes reasonable performance is observed. In this case is $n_a = n_b = 2, n_k = 1$.

2.3 Model Estimation

The ANN used for prediction is a feedforward multilayer network with one hidden layer activated by tanh (hyperbolic tangent) function and output layer activated by linear function. The network training is carried offline in batch from using the Levenberg-Marquardt optimization. The algorithm basically seeks to minimize the prediction error over the training data test. In order to have a good training, the data must contain sufficient information about the system dynamics. The network weights and biases updating are performed only after the entire training set has been applied.

The cost function to be minimized for updating weights and biases is:

$$J(\theta) = \sum_{t=1}^p \left[\sum_{j=1}^o (d_j(t) - y_j(t))^2 \right] \quad (4)$$

Where θ is the vector of all weights and biases of the network, $d_j(t)$ is the desired value of the j^{th} output and the t^{th} pattern, $y_j(t)$ is the actual value of the j^{th} output and the t^{th} pattern, p is the number of patterns and o is the number of network outputs.

2.4 Model Validation

When a network has been trained, the next step according the procedure is to evaluate it. The most common method of validation is to investigate the residuals (prediction errors) by cross validation on a test set. The test error is an important quantity since it can be viewed as an estimate of the generalization error. This should not be too large compared to training error, in which case one must suspect that the network is over-fitting the training data. If a test set is not available the average generalization error:

$$J(M) \equiv E \{ \bar{V}(\hat{\theta}) \}, \bar{V}(\hat{\theta}) = \lim_{N \rightarrow \infty} E \{ V_N(\hat{\theta}, N) \} \quad (5)$$

can be estimated from the training set alone by Akaike's final prediction error (FPE) estimate. Although a test set is available, the FPE estimate might still offer some valuable insights. For the basic unregularized criterion the estimate reads (see Ljung, 1987):

$$\hat{J}_{FPE}(M) = \frac{N+d}{N-d} V_N(\hat{\theta}, Z^N) \quad (6)$$

d denoting the total number of weights in the network. When the regularized criterion is used, the expression gets somewhat more complex (see Larsen & Hansen, 1994):

$$\hat{J}_{FPE}(M) = \frac{N+\gamma_1}{(N+\gamma_1-2\gamma_2)} V_N(\hat{\theta}, Z^N) \quad (7)$$

where

$$\gamma_1 = tr \left[R(\hat{\theta}) \left(R(\hat{\theta}) + \frac{1}{N} D \right)^{-1} R(\hat{\theta}) \left(R(\hat{\theta}) + \frac{1}{N} D \right)^{-1} \right] \quad (8)$$

and

$$\gamma_2 = tr \left[R(\hat{\theta}) \left(R(\hat{\theta}) + \frac{1}{N} D \right)^{-1} \right] \quad (9)$$

$\gamma_1 = \gamma_2$ specifies the so-called effective number of parameters in the network. It seems as if the estimate often becomes more reliable in the regularized case, which probably has to do with the regularization having a smoothing effect on the criterion. A smoother criterion function means that the assumptions on which the FPE was derived are more likely to be valid.

2.5 Going Backward in the Procedure

The fig.1 shows some “feedback” paths from validation to the previous blocks. The path from validation to training is due to the criterion having local minima. Since it is very likely that one ends up in a “bad” local minimum, the network should be trained a couple of times, starting from different initial weights. Regularization has a tremendous smoothing effect on the criterion, and several of the local minima are hence often removed by this. Local minima do, however, remain one of the major problems for nonlinear regressions, and there is no simple way of avoiding them.

Another path in fig. 1 leads back to the model structure selection block. Because of the way the model structure selection problem has been divided into two separate subproblems, this can mean two things, namely: “try another structure” or “try another network architecture”. While the regressor structure typically has to be chosen on a trial-and-error basis, it is to some extent possible to automate the network architecture selection. For this purpose Optimal Brain Surgeon (OBS) is used that it is the most important strategy, and it is consequently the only method which has been implemented for models of dynamic systems.

In order to remove the superfluous weights from the network is used pruning the network according to the optimal brain surgeon strategy to obtaining the optimal network architecture.

3. CONCENTRATION CONTROL OF CSTR

Assume that a deterministic model of the process under consideration has been established with the neural network system identification step:

$$y(t) = g(y(t-1), \dots, y(t-n), u(t-d), \dots, u(t-d-m)) \quad (10)$$

The “state” $\varphi(t)$ is then introduced as a vector composed of the arguments of the function g :

$$z(t) = [y(t-1), \dots, y(t-n), u(t-d), \dots, u(t-d-m)]^T \quad (11)$$

At time $t=\tau$ linearize g around the current state $\varphi(\tau)$ to obtain the approximate model:

$$\tilde{y}(t) = a_1 \tilde{y}(t-1) - \dots - a_n \tilde{y}(t-n) + b_0 \tilde{u}(t-d) + \dots + b_m \tilde{u}(t-d-m) \quad (12)$$

where

$$a_1 = - \left. \frac{\partial g(\varphi(t))}{\partial y(t-i)} \right|_{\varphi(t)=\varphi(\tau)} \quad (13)$$

$$b_1 = \left. \frac{\partial g(\varphi(t))}{\partial u(t-d-i)} \right|_{\varphi(t)=\varphi(\tau)} \quad (14)$$

and

$$\tilde{y}(t-i) = y(t-i) - y(\tau-i) \quad (15)$$

$$\tilde{u}(t-i) = u(t-i) - u(\tau-i) \quad (16)$$

Separating the portion of the expression containing components of the current state vector, the approximate model may alternatively be written as:

$$y(t) = \left(1 - A(q^{-1}) \right) y(t) + q^{-d} B(q^{-1}) u(t) + \zeta(\tau) \quad (17)$$

where the bias term, $\zeta(\tau)$, is determined by

$$\zeta(\tau) = y(\tau) + a_1 y(\tau-1) + \dots + a_n y(\tau-n) - b_0 u(\tau-d) - \dots - b_m u(\tau-d-m) \quad (18)$$

and

$$A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_n q^{-n} \quad (19)$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \dots + b_m q^{-m} \quad (20)$$

The approximate model may thus be interpreted as a linear model affected by an additional DC disturbance, $\zeta(\tau)$, depending on the operating point.

It is straightforward to apply this principle to the design of control systems. The idea is illustrate in Fig. 3.

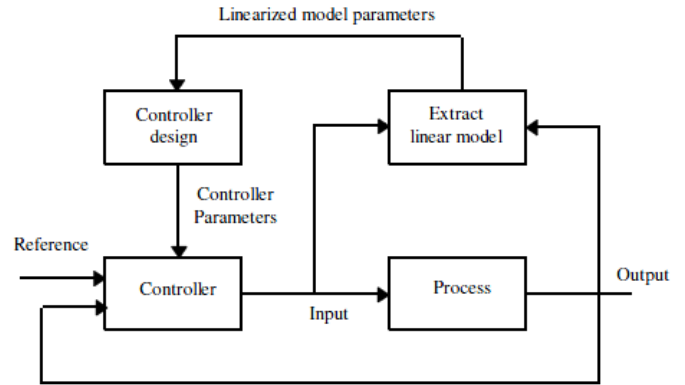


Fig. 3: Instantaneous linearization applied to control system design

3.1 Approximate GPC (APC)

The idea behind generalized predictive control (GPC), is to at each iteration minimize a criterion of the following type:

$$J(t, U(t)) = \sum_{i=N_1}^{N_2} [r(t+i) - \hat{y}(t+i)]^2 + \rho \sum_{i=1}^{N_u} [\Delta u(t+i-1)]^2 \quad (20)$$

with respect to the N_u future controls

$$U(t) = [u(t), \dots, u(t+N_u-1)]^T \quad (21)$$

and subject to the constraint

$$\Delta u(t+i) = 0, \quad N_u \leq i \leq N_2 - d \quad (22)$$

N_1 denotes the minimum prediction (or costing) horizon, N_2 the maximum prediction (or costing) horizon, and N_u the (maximum) control horizon. ρ is a weighting factor for penalizing variations in the control input. $\zeta(\tau)$ is modeled as integrated white noise and the predictions of future outputs, $\hat{y}(t+i)$, are determined as the minimum variance predictions. The optimization problem (which must be solved on-line since a new linear model is obtained at each sample) results in a sequence of future controls, $U(t)$. From this sequence the first component, $u(t)$, is then applied to the process.

4. SIMULATION RESULTS

The simulations have been carried out to evaluate the performance of neural network based input-output modeling and predictive control for Concentration tracking.

Before controlling the system, the CSTR is modeled by an ANN. The number of neurons in hidden layer (15) was chosen by trial and error in order to have an acceptable value of error between the output of the motor model and ANN output. The training data sets are collected from CSTR under sampling time

$$T_s = 6 \text{ s.}$$

Three hundred (300) training epochs are performed in order to get an acceptable mean square error (MSE). At this point, the optimal network weights for the neural model are stored and used for validation. Validation and cross-validation respectively consist of applying the training and test data to the neural model in order to see how closely it's the CSTR data in each case.

In validation, it is used the training data set as an input to the neural network model of the system and compare the output obtained with what was used during training. Fig. 4 shows the validation result for concentration with the training data set. The result is considered satisfactory. Then, the network's weights are stored and this neural is tested in order to check if it emulates the behavior of the CSTR. To better characterize the network's modeling ability, cross-validation is performed by applying other data set not used in the training of the neural network. Fig. 5 shows the result for concentration with the testing data set. For the response, we can observe that the fit is almost good and the errors are minimal.

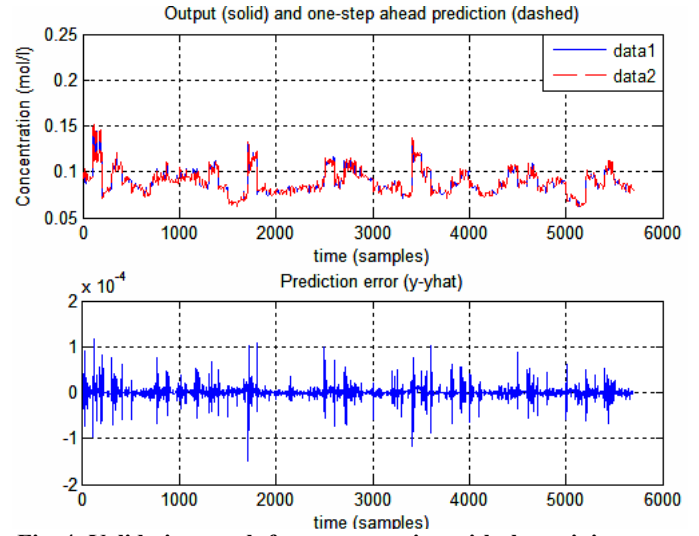


Fig. 4: Validation result for concentration with the training data set

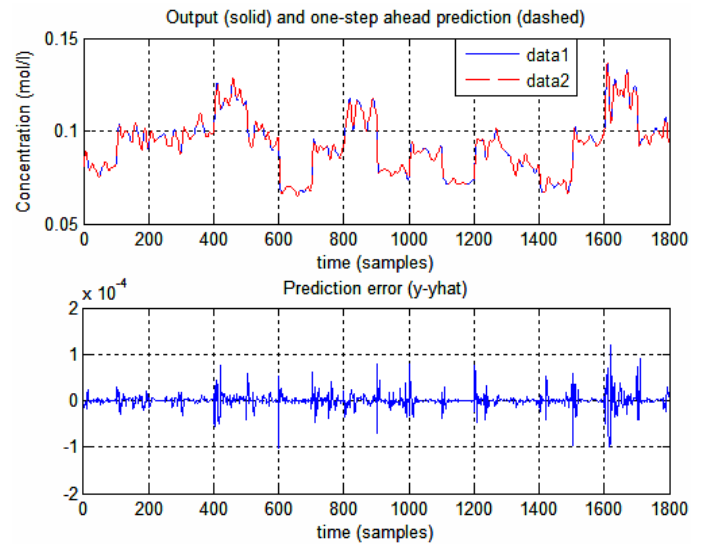


Fig. 5: Validation result for concentration with the testing data set

$(N_2 + 1)$ identical neural models of the CSTR is placed to create a nonlinear predictor (NNP), which is placed into the NNAPC control loop. The system was tuned by varying N_1 , N_2 , N_u and ρ to produce a desirable response. The final choice of controller parameters are $N_1=1$, $N_2=15$, $N_u=2$ and $\rho=0.001$. The controlled response of the CSTR for reversal concentration of operation is shown in Fig. 6. The tracking performance of the proposed NNAPC system is satisfactory achieved.

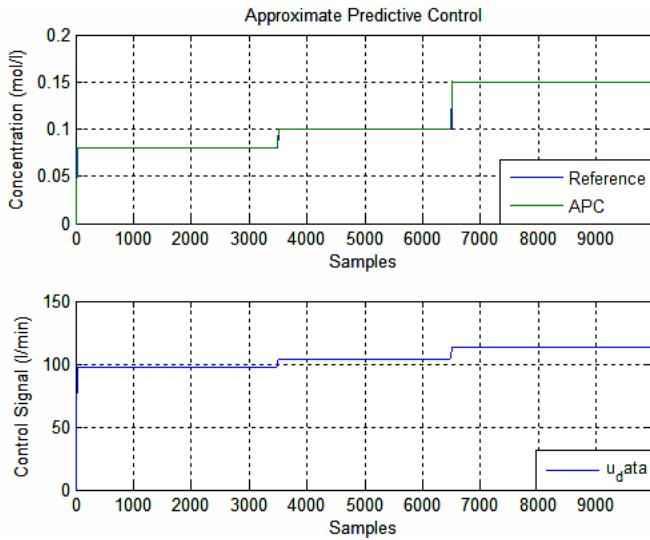


Fig. 6: Result of NNAPC and control signal

5. CONCLUSIONS

A new advanced control algorithm: named neural networks approximate generalized predictive control algorithm, for concentration tracking of a CSTR is presented in this paper. It is based on a combination of artificial neural networks and model predictive control technique.

The ANN is used for input-output modeling of the process, which permits to approximate is by a NNARX model and use it as nonlinear predictor. The choice of neural network structure depends upon the performance needed and its training is done off-line using the Levenberg-Marquardt algorithm. The advantage of this modeling by ANN is the ability to have a nonlinear model with good performance without requiring the knowledge of CSTR parameters which may vary in time.

Simulation results have been provided to verify the effectiveness of the proposed controller with ANN modeling for concentration tracking in CSTR. Because of the neural network, which is characterized by a nonlinear equation, the closed loop stability is difficult to study. So, the research area for the neural network application is depending to verify the effectiveness of this kind of intelligent control.

6. REFERENCES

[1] M. NØrgaard. 2000. Neural Network Based System Identification Toolbox: Tech.Report. 00-E-891, Department of Automation, Technical University of Denmark.

[2] Constant, L., Lagarrigues, P., Dagues, B., Rivals, I. and Personnaz, L. 1999. Modeling of Electromechanical Systems Using Neural Networks: Computational Intelligence and Applications, P. S. Szczepaniak (ed.), Physica-Verlag, c/o Springer-Verlag.

[3] Garcia, C.E, Prett, D.M. and Morari, M. 1989. Model Predictive Control: Theory and Practice- A Survey. Automatica, 3, 334-348.

[4] Larsen & L.K. Hansen. 1994. Generalization Performance of Regularized Neural Network Models: proc. Of the IEEE Workshop on neural network for Signal Proc. IV, Piscataway, New Jersey, pp.42-51.

[5] M. NØrgaard, O. Ravn, N. K. Poulsen, L. K. Hansen. 2000. Neural networks for modeling and Control of Dynamic Systems: Springer-Verlag, London, UK.

[6] R. Fletcher. 1987. Practical Methods of Optimization: Wiley.

[7] K.J. Hunt, D. Sbarbaro. 1991. Neural Networks for Nonlinear Internal Model Control: IEE Proceedings-D, Vol. 138, No. 5, PP. 431-438.

[8] Psalties, A. Sideris, A.A. Yamamura. 1988: A Multilayered Neural Network Controller: Control Sys. Mag., Vol. 8, No. 2, pp. 17-21.

[9] L. Ljung. 1987. System Identification- Theory for the User: prentice-Hall, 1987.

[10] M. W. Pedersen, L.K. Hasen, J. Larsen. 1995. Pruning with Generalization Based Weight Saliences: □OBD, □OBS, Proc. Of the Neural Information Processing Systems 8.

[11] Farhad Aslam, Gagandeep Kaur. 2011. Comparative analysis of conventional P, PI, PID and fuzzy logic controllers for the efficient control of concentration in CSTR. International Journal of Computer Applications. Volume 17-No.6.

[12] J.D. Morningred, B.E.Paden, D.E. Seborg and D.A. Mellichamp. 1990."An adaptive nonlinear predictive controller" in. Proc. of the A.C.C. vol.2 pp.1614-1619.

[13] G.Lightbody and G.W.Irwin. Nonlinear Control Structures Based on Embedded Neural System Models, IEEE Tran. on Neural Networks Vol.8 No.3 pp.553-567.

[14] J.Espinosa and J. Vandewalle, Predictive Control Using Fuzzy Models, Submitted to the 3rd. On-Line World Conference on Soft Computing in Engineering Design and Manufacturing.