A Neuro-Predictive Approach for Tuning Industrial Controllers

C. LAZAR*, D. VRABIE*, S. CARARI*, D. IVANA**
*Department of Automatic Control and Industrial Informatics
"Gh. Asachi" Technical University of Iasi
Blvd. D. Mangeron 53A, 700050 Iasi
**Power Plant Ltd. Iasi
Chisinaului 25, 700182 Iasi
ROMANIA

Abstract: -This paper presents a new neuro-predictive tuning procedure for PID controllers. The tuning method is based on the optimization of an objective function subject to constraints over a finite prediction horizon in time, making use of a neural process model. The performance of this new self tuning method implemented as a tuner is substantiated by experiments on a level-flow pilot plant and by comparison with a conventional controller.

Key-Words: - Neural network based model, Neural predictor, Predictive control, Self-tuning PID control

1 Introduction

Most of the industrial controllers use fixed parameters and linear structures for controlling complex nonlinear systems. These industrial controllers usually employ standard classical PID control structures. However, for nonlinear processes driven through the whole operating range, linear models and PID control become impractical and due to this reason, the use of a nonlinear process model becomes a necessity.

Reliable nonlinear models can be obtained using neural networks, which are capable of approximating nonlinear functions with high level of accuracy [1,2]. Neural networks can be also used to determine controller parameters, mainly due to their ability to learn relationships directly from the process data. Most of the neural network based control applications are using a neural network to model the controller and a neural identifier for the process model [3,4]. Unfortunately, the replacement of PID controllers with the neural network based controllers in industrial applications is quite expensive and that is why the companies do not take this option into consideration.

A solution accepted by the users is to develop schemes for on-line adaptation or self-tuning of industrial controllers, and several methods have been proposed in the last decades [5,6,7]. Different adaptive techniques are classified in [5] and it turns out that a controller with constant parameters obtained via an auto-tuning procedure should be chosen for processes with constant dynamics. However, if the process dynamics are varying, then the controller should compensate variations by adapting its parameters. There are two types of process dynamics variations: predictable and unpredictable. The predictable ones are typically caused by nonlinearities and can be handled using a gain schedule. The controller parameters are found for different operating conditions with an auto-tuning procedure that is then used to build a schedule. Unpredictable variations are caused by non-measurable variations, which cannot be handled by gain scheduling, and in this case the use of adaptive control is necessary.

This paper presents a tuner for on-line tuning of PID controllers that employs the multi-step-ahead predictive properties of neural networks in the objective function. The gain scheduling principle is replaced by using a neural network capable to capture the variations of the predictable dynamics of the process. The tuning approach uses a neural network to model the process dynamics and to develop a neuro-predictor. The tuning parameters of the controller are obtained through the optimization of the prediction error over a finite horizon in time. The neuro-predictive tuning approach is implemented as a tuner for industrial control applications with PID controllers. The advantage of such a tuner is the effective on-line adaptation of the existing industrial controllers while the process is in operation and the tracking of different process operating regimes and variations.

The proposed neuro-predictive tuner was

successfully tested on a pilot plant which models the cascade control of level and flow, a process that is often employed in industrial applications. The experimental results obtained for the pilot plant are also given.

2 Neuro-Predictive Tuner

The proposed neuro-predictive tuner is based on two parallel control structures, Fig. 1, working synchronized with the predictable dynamics process controlled by a PID controller. The first parallel control structure, which simulates the real-time control loop and uses the same sample period T, is composed of a neural network that models the process and a PID controller identical with the one from the real-time control structure

The second parallel control structure is a predictive control loop consisting of a neural predictor and a PID controller with adaptive tuning parameters. The predictive structure, with the sampling rate T_p, works faster than the real time control loop to supply the predicted control error over a finite time horizon. The tuning parameters are calculated at each sample time instant via the optimisation procedure and the values obtained are also used to update the tuning parameters of the other two control structures. Thus, the controller parameters are adapted based on predictive optimization of the control system behavior and the desired performances can be achieved over the entire operating range.

2.1 Simulated control loop

Within the simulated control loop, the neural network that models the real process with predictable dynamic variations represents the plant.

The use of neural networks for nonlinear process modeling and identification is justified by their capacity to approximate the non-linear systems.

One of the most general non-linear models, which includes the largest class of non-linear processes, is the NARMAX model [8,9]:

$$y(k) = f[y(k-1),..,y(k-n),u(k-d),..,u(k-d-m)] \quad (1)$$

where $f(.)$ is some nonlinear function, d is the dead time, n and m are the orders of the nonlinear system model, u and y being the input and the output of the process. A neural network based model, i.e. NNARMAX, corresponding to the NARMAX model, can be obtained by adjusting the weights of a multi-layer perceptron (MLP) architecture with adequately delayed inputs. The neural network output is given by:

$$y(k) = f_N[\mathbf{u}(k-d-1), \mathbf{y}(k-1)] \quad (2)$$

where f_N denotes the input-output transfer function of the neural network which replaces the non-linear function f in (1) and $\mathbf{u}(k\text{-}d\text{-}1)$ and $\mathbf{y}(k\text{-}1)$ are:

$$\begin{aligned} \mathbf{u}(k-d-1) &= [u(k-d-1)u(k-d-2)...u(k-d-m)]^T \\ \mathbf{y}(k-1) &= [y(k-1)y(k-2)...y(k-n)]^T \end{aligned} \quad (3)$$

For a neural network with one hidden layer, the following expression is obtained for equation (2):

$$\begin{aligned} y(k) = \sum_{j=1}^{h} w_j \sigma_j (\mathbf{w}_j^u \mathbf{u}(k-d-1) + \\ + \mathbf{w}_j^y \mathbf{y}(k-1) + b_j) + b \end{aligned} \quad (4)$$

where h is the number of neurons in the hidden layer, σ_j is the activation function for the j-th neuron from the hidden layer, $\mathbf{w}_j^u$ the weight vector for the j-th neuron with respect to the inputs stored in $\mathbf{u}(k\text{-}d\text{-}1)$, $\mathbf{w}_j^y$ the weight vector

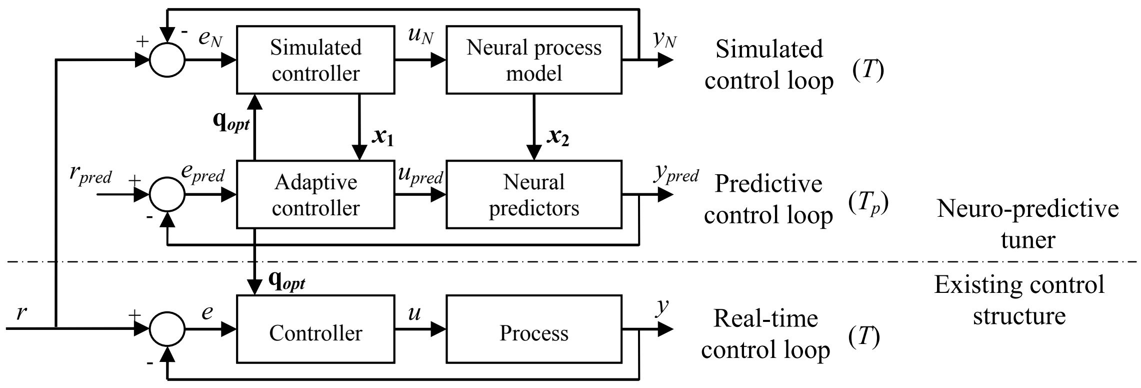


Fig. 1 Control structure with neuro-predictive tuner

for the j-th neuron with respect to the inputs stored in $\mathbf{y}(k-1)$, b_j the bias for the j-th neuron from the hidden layer, w_j the weight for the output layer corresponding to the j-th neuron from the hidden layer and b the bias for the output layer. Such structures with a single hidden layer are considered satisfactory for most of the cases.

Since all industrial processes are working in closed loop, in order to obtain the neural model of the process, it was considered appropriate to perform closed loop identification. In order to result a most accurate model for the process nonlinear dynamics, the training data had to be obtained around different operating points such that the entire variation range of the process output is covered. For this reason a stepwise reference for the system was chosen summed with a pseudo random binary signal that was generated with a shifting register [10].

2.2 Predictive control loop

In order to obtain the variations of the predictable dynamics at the time instants k, a neural predictor based on the neural model of the process was used. A sequential algorithm that uses the knowledge of current values of u and y together with the neural network system model gives the i-step ahead neural predictor:

$$\begin{aligned} y(k+i) = \sum_{j=1}^{n}[w_j\sigma_j(\mathbf{w}_j^u\mathbf{u}(k-d+i-1)+ \\ \mathbf{w}_j^y\mathbf{y}(k+i-1)+b_j)]+b \end{aligned} \tag{5}$$

where:

$$\begin{aligned} \mathbf{u}(k-d+i-1) &= [u(k-d+i-1)...u(k-d+i-m)]^T \\ \mathbf{y}(k+i-1) &= [y(k+i-1)y(k+i-2)...y(k+i-n)]^T \end{aligned} \tag{6}$$

The future control $\mathbf{u}(k-d+i-1)$ from (5) is obtained by operating the predictive control loop, at time instant k, faster than the real-time control loop, so that the predicted output $y(k+i)$ could be determined in a shorter period of time, where $i=\overline{N_1, N_2}$, N_1 and N_2 are the prediction horizons. If T_p is the sampling time with which the predictive control loop operates, this must satisfy:

$$(N_2 - N_1)T_p << T \tag{7}$$

Placing the neural model of the process to operate in the simulated control loop, in parallel with the real time control loop, it is possible to transfer at each time instant k the state $\boldsymbol{x}_2$ of the neural model to the neural predictor and the state $\boldsymbol{x}_1$ of the PID controller to the adaptive controller (Fig. 1). Thus, at each time instant k, the predicted behavior of the process is obtained in the vector form:

$$\mathbf{y}_{pred} = [y(k+N_1)y(k+N_1+1)...y(k+N_2)]^T \tag{8}$$

The process output $\mathbf{y}_{pred}$ predicted by the neural predictor is used to calculate the predicted control error based on the controller set-point.

Consider the discrete form of a PID controller:

$$u(k) = u(k-1) + q_0e(k) + q_1e(k-1) + q_2e(k-2) \tag{9}$$

Substituting (9) in (5) yields predicted control error:

$$\begin{aligned} e_{pred}(k+i) = \sum_{j=1}^{n}[w_j\sigma_j(\mathbf{w}_j^u\mathbf{u}(k-d+i-1)+ \\ \mathbf{w}_j^y\mathbf{y}(k+i-1)+b_j)+b]-r(k+i) \end{aligned} \tag{10}$$

where the vector $\mathbf{u}(k-d+i-1)$ depends on the tuning parameters vector $\mathbf{q}=[q_0\ q_1\ q_2]$.

By optimizing, with respect to $\mathbf{q}$, the cost-function:

$$J = \frac{1}{2}\sum_{i=N_1}^{N_2} e_{pred}^2(k+i) \tag{11}$$

the optimal tuning parameters $\mathbf{q}_{opt}$ are obtained also taking into account the process predictable dynamics variations. For the next time instant, $k+1$, the tuning parameters $\mathbf{q}_{opt}$ are transferred to the real time control loop and to the simulated control loop.

The self-tuning method has the advantage that it does not imply a pre-tune phase because $\mathbf{q}_{opt}$ is known after the first sampling period. However, the initialization of the tuning parameters $\mathbf{q}$ with adequate values is necessary when the self-tuning procedure is started.

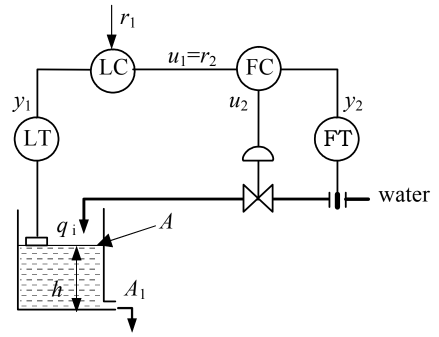
3 Experimental results

The neuro-predictive tuner developed for on-line tuning of PID controllers was tested on a level-flow pilot plant. The schematic diagram of the pilot plant is presented in Fig. 2.

The level is controlled using a cascade control structure that has as internal variable the feed water flow. The inner loop controls the feed water flow and rejects the disturbances caused by the pressure variations in the water pipe.

The outer loop PI controller determines the feed water flow reference signal r_2 for the inner loop based on the measured water level signal.

The tank and the inner loop represent the plant for the outer controller.



A - cross section area
A_1- circular orifice area
q_i - flow rate

LC - Level Controller
FC - Flow Controller
LT - Level Transducer
FT - Flow Transducer

Fig. 2 Schematic diagram of the pilot plant

The mathematical model of the open tank with an orifice is:

$$A\frac{dh}{dt} = q_i - C_d A_1 \sqrt{2gh} \tag{12}$$

The tank parameters are: A=203.4 cm^2, A_1=2.26 cm^2, h_{max}=13.5 cm, C_d=0.6.

Due to the non-linearity described in equation (12), this plant represents a suitable test-bed for the proposed self-tuning control method. The level and flow controllers, together with the neuro-predictive tuner are implemented on an IBM PC compatible computer with 12 bits A/D and D/A interface.

3.1 Development of the neural process model

A MLP neural network is configured to represent the NNARMAX model (2) by applying n delayed process output and m delayed process input $y(t)$.

In order to estimate the parameters of the neural model, a training sequence was built so that the process output should explore its whole operating range. Thus, the reference r_1 from Fig. 3,a was applied to the real time control loop and by monitoring the control signal u_1 and the output y_1, with a sampling rate of 1 sec, a training sequence with 3600 successive samples, for each variable was obtained. Using the collected input-output data, a two layer neural network was trained off-line.

The model parameters m, n and d were estimated using an ARX type of identification with the Matlab System Identification Toolbox. It was found that the process has a delay d=5 and m=2, n=2. So the best training results should be obtained with a model that has the form:

$$y(k) = f_N(y(k-1), y(k-2), u((k-5), u(k-6)) \tag{13}$$

Thus, the resulting neural network model of the process has four inputs. Based on a series of successive training and testing experiments it was determined the structure of the network that gave the best results, i.e. a network with 15 neurons in the hidden layer.

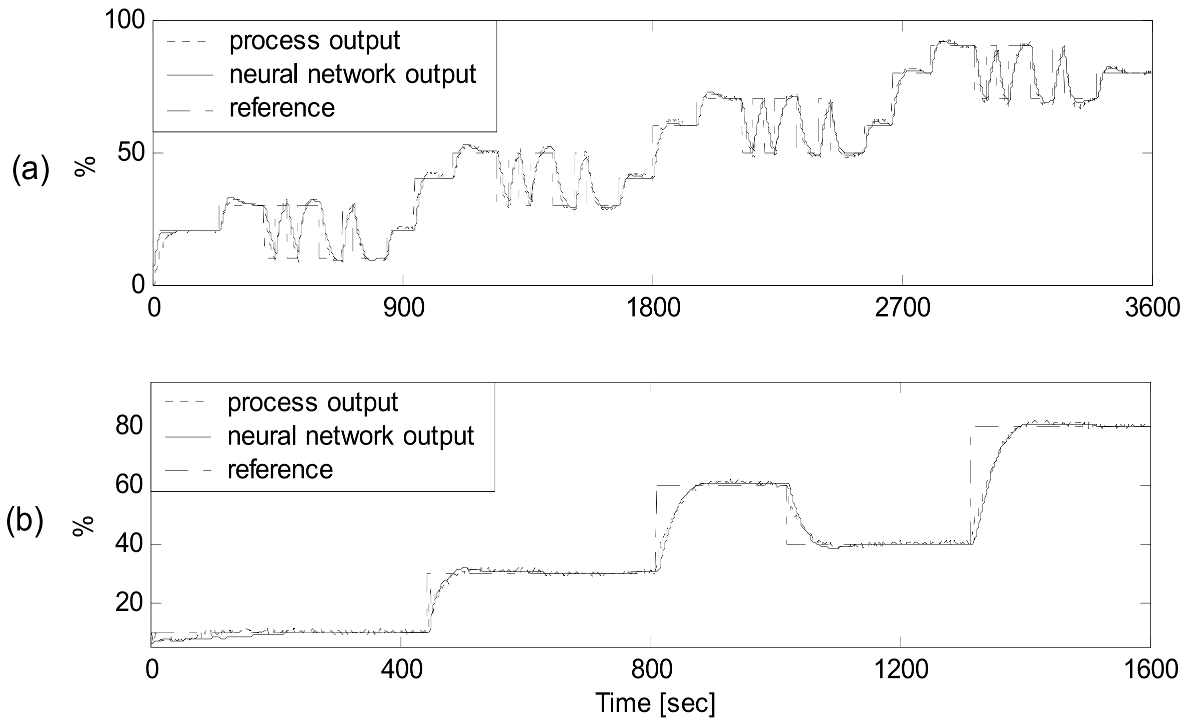


Fig. 3: Neural process model validation

The network parameters (weights and biases) were calculated such that the mean squared error between the desired response (the values from the target vector) and the network output is minimized. Thus, when a certain stop criterion is satisfied, the training algorithm gives a set of sub-optimal values of the weight vectors.

The neural network with the constant parameters, obtained after the training, represents the process neural model. Before this model is used in the simulated control loop and before obtaining the neural predictions, the validation of the neural model is required. Two validation tests were performed. Because the training sequence was obtained by a closed loop identification method, the first validation test consisted in applying the same reference used in identification to the simulated control loop and to the real time control loop. The experimental results are depicted in Fig. 3(a). The second validation test was similar with the first, but the reference was changed, as is shown in Fig. 3(b). These tests used the following error index to appreciate the quality of the model:

$$\varepsilon_{index} = \sqrt{\frac{\sum_{k=1}^{N}[y_N(k) - y(k)]^2}{\sum_{k=1}^{N} y(k)^2}} \tag{14}$$

The best results were obtained while training the network for the whole data set, 300 epochs and a mean squared error of 10^{-5}.

In order to obtain the on-line validation of the neural model, a Matlab/Simulink scheme was implemented, which makes use of the Real Time Windows Target kernel. This allows the closed loop operation of the real and neural closed loop systems using the same PI controller.

The software instruments presented in [12] were used for the training and the validation of the neural network that models the process.

3.2 Adaptive procedure results

The parameters q_0 and q_1 of the PI controller were obtained by optimizing an objective function via a numerical procedure. The cost function implements the mean squared prediction error over a finite prediction horizon as given in (11). A neural model was used to predict the future outputs in order to obtain a reliable nonlinear model of the process. In order to avoid the saturation of the actuator, the optimization was carried out considering constraints for the control input and for the controller parameters.

For a fair comparison with the existing control structures, the PI level controller was first tuned using the relay method of Astrom and Hagglund [7]. Fig. 4 shows the resulting closed loop stepwise responses obtained with the simulated control loop for fixed tuning parameters of the PI controller and with continuous adaptation of the tuning parameters based of the proposed self-tuning method.

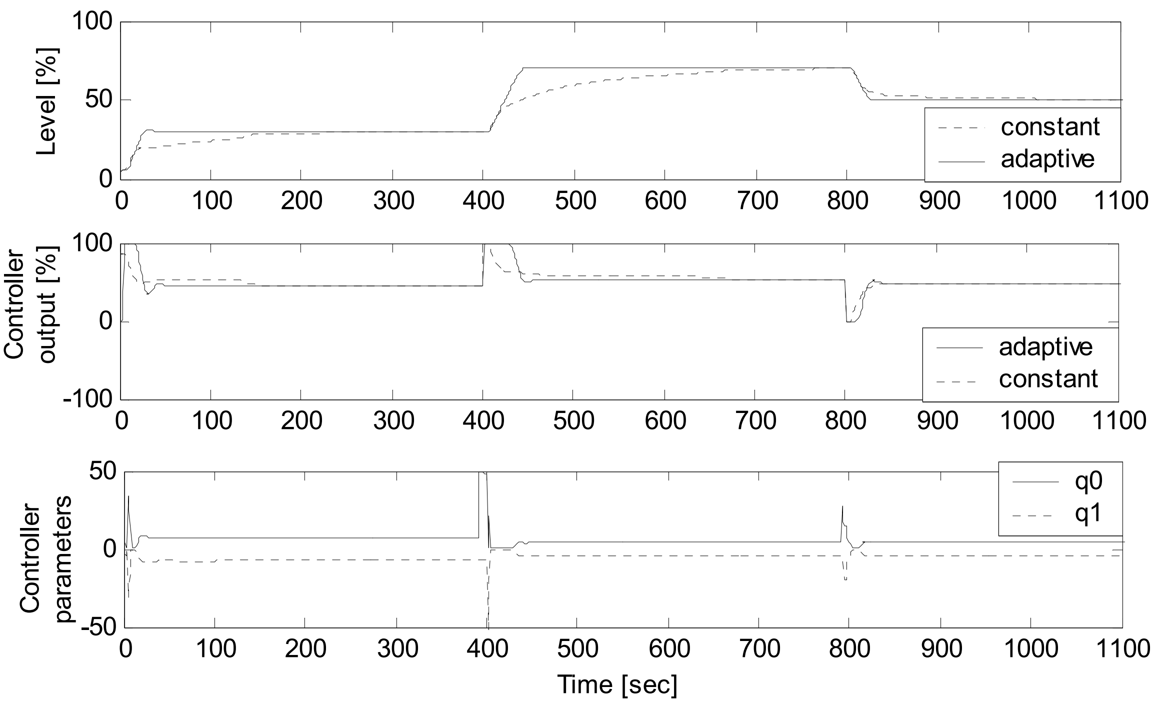


Fig. 4: Comparative results and parameters for the adaptive discrete PI controller

The control inputs for the two loops and the tuning parameters of the PI controller are also depicted in Fig. 4. As seen in the figure, the fixed parameters controller gave a sluggish control response. With the tuner, the controller had continuous adaptation of the tuning parameters resulting in a much faster control.

4 Conclusions

A tuner for PID controllers based on a neuro-predictive control approach has been derived. The advantage of the method consists in the on-line adaptation of the controller parameters and in tracking different process operating regimes.

The proposed neuro-predictive tuner was successfully tested on a pilot plant, which models the cascade control of level and flow, a process that is often employed in industrial applications. A comparison with the existing control structures was also given.

The neuro-predictive approach for tuning controllers can also be used for more sophisticated control algorithms than the PID. Optimal tuning parameters of such controllers, with many more parameters, can be obtained by minimizing an objective function based on the mean squared prediction error over a prediction horizon using numerical procedures.

Acknowledgements

The authors would like to acknowledge CNCSIS for financing the work under the grant no. 763/2003.

References:

[1] Chen S, Billings S.A. and P.M. Grant, Nonlinear system identification using neural networks, *Int. J. Control*, Vol. 49, No.3 , 1990, pp. 1013-1032.

[2] Narenda K.S. and Parthasarathy K., Identification and Control of Dynamical Systems using Neural Networks, *IEEE Trans. Neural Networks*, Vol.1, 1990, pp. 4-28.

[3] Hunt K.J., Sbarbaro D., Zbikowski R. and Gawthrop P.J., Neural Networks for Control Systems – A Survey, *Automatica*, Vol. 28, No. 6, 1992, pp. 1083-1142.

[4] Narenda K.S., Neural Network for control: Theory and applications, *Proc. IEEE*, Vol. 48, 1996, pp. 1385 – 1406.

[5] Hagglund T. and Astrom K.J., Industrial Adaptive Controllers Based on Frequency Response Techniques, *Automatica*, Vol. 27, No.4, 1991, pp. 599-609.

[6] Cheng-Ching Yu, *Autotuning of PID Controllers*, Springer-Verlag, London, 1999.

[7] Astrom K.J. and Hagglund T., *PID Controllers: Theory, Design and Tuning*, Instrument Society of America: Research Triangle Park, 1995.

[8] Chen S. and Billings S.A., Representations of nonlinear systems: the NARMAX model, *Int. J. Control*, Vol. 49, No. 3, 1989, pp. 1013 – 1032.

[9] Sales K.R. and Billings S.A., Self-Tuning Control of Nonlinear ARMAX Models, *Int. J. Control*, Vol. 51, No. 4, 1990, pp. 753-769.

[10] Landau I.D., *System Identification and Control Design*, Prentice Hall, 1990.

[11] Liu G.P., Kadirkamanathan V. and Billings S.A., Predictive control for nonlinear systems using neural networks, *Int. J. Control*, Vol. 71, 1998, pp. 1119 -1132.

[12] Kloetzer, M., Ardelean, D. and Păstrăvanu, O., Developing Simulink tools for teaching neural-net-based identification, *The 9th Mediterranean Conference on Control and Automation*, 63, 2001.