The Basic Ideas of Neural Predictive Control

Leizer Schnitman [1]
Aeronautics Institute of Technology
São José dos Campos - SP, Brazil

Adhemar de Barros Fontes [2]
UFBA Department of Electrical Engineering
Salvador - BA, Brazil.

Abstract

This paper presents the application of predictive control techniques using Artificial Neural Nets (ANN). The idea is illustrate the structure of the predictive controller and the optimization functions that is usually used to update the control action, then apply the ANN technique. The ANN equations and its gradient equations are developed. Based on the ANN capacity to predict, on a optimization function and on a rule to update the control action, NPC (Neural Predictive Control) algorithms are developed and applied to control the selected plant. The paper also proposes an intelligent adaptability to ponder control action as function of dominant pole displacement.

Keywords: Predictive Control, Neural Nets, Nonlinear Systems

1. Introduction

Predictive control is characterized by accomplishing the prediction of future values through a model. Then, based on the predicted values, on a optimization function and on a control law, the controller generates a future control action. The objective of this work is to present in a brief way the implementation of predictive controllers based on the ANN . It uses some optimization indexes and control laws usually applied to the conventional predictive control.

In Figure 1 a general structure is presented to implement Neural Predictive Controllers. Note that not all the cases include the model reference block. In this cases, the ANN is properly trained and represents the dynamics of the plant in a satisfactory way.

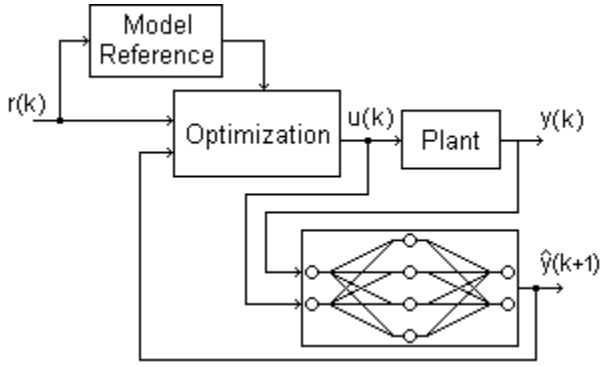

Figure 1: Blocks Diagram of Neural Predictive Controllers

2. Predictive Control

The predictive controller, in summary, is characterized by computing future control actions based on output values predicted by a model, with vast literature and academic and industrial interest (Clarke, 1987; Astrom and Wittenmark 1995; Garcia et all, 1989; Menezes, 1993; Arnaldo, 1998). This section

[1] Email: leizer@ele.ita.cta.br
[2] Email: adhemar@ufba.br

presents the concepts of predictive control based on NPC, using the usual optimization functions and control laws, applied to the conventional predictive controllers.

2.1 Optimization functions

The optimization function, usually represented by the index J, represents the function that the control action tries to minimize. In a intuitive way, the error between the plant output and the desired value is the simplest example of an optimization function, and it is expressed by:

$$J = y_{ref}(k) - y(k) = e(k) \tag{1}$$

where:

$y(k)$ represent the plant output

$y_{ref}(k)$ represent the desired response

$e(k)$ represent the estimation error

k is the sample time

One of the most usual optimization function is based on the square error and it is represented as:

$$J = \left[y_{ref}(k) - y(k)\right]^2 = \left[e(k)\right]^2 \tag{2}$$

But the optimization index can take forms of more complex functions. For predictive controllers, whose models are capable to predict N steps ahead, the simple application of the square error approach can present satisfactory results. This case admits that the optimization function is not limited to an only point, but an entire vector of N predicted errors. It seeks to optimize the whole trajectory of the future control actions in a horizon of N steps ahead.

$$J = \sum_{j=1}^{N}\left[y_{ref}(k+j) - \hat{y}(k+j)\right]^2 = \sum_{j=1}^{N}\left[e(k+j)\right]^2 \tag{3}$$

More complex optimization functions can consider the control effort. It is the specific case of GPC (Generalized Predictive Control), where the optimization index J can be expressed as:

$$J = \sum_{j=N_1}^{N_Y}\left[y_{ref}(k+j) - \hat{y}(k+j)\right]^2 + \sum_{j=1}^{N_U}\alpha(j).\left[\Delta u(k+j)\right]^2 \tag{4}$$

where:

$\bar{y}(k)$ - is the output plant estimation at instant = k

Δu - is the control action increment.

α - is the ponderation of the control action.

N_1 - is the minimum horizon of prediction.

N_U - is the control horizon.

N_Y - is the maximum horizon of prediction.

The objective of the control problem is to minimize the index J, with respect to the control actions, looking for the points where the first order differential is null.

2.2 Updating the control actions

Besides an index J to be minimized, the predictive controllers are based on rules to update the control action, using expressions that govern the increment in the current control action looking for to minimize the optimization indexes.

One of the most common rules is the decreasing gradient rule, in which the actualization is made in the direction of the negative gradient of the function, always seeking the minimum point - it is the basis of the delta-rule, applied in the backpropagation training method, one of the most usual method to train ANN.

One of the problem with this rule is to not be applicable in multimodal functions, being unable to guarantee the convergence to the global minimum. Several adaptations are proposed, using hybrid algorithms that, in the first stage, use methods to search global minimum, as the Genetic Algorithms (AGs), and then apply the method of the decreasing gradient.

The decreasing gradient rule can be expressed in the form:

$$u(k+1) = u(k) - \lambda . \frac{\partial J}{\partial u(k)} \tag{5}$$

where:

λ - is the consideration of the control action.

$\frac{\partial J}{\partial u(k)}$ - is the differential of the index J, with respect to the current control action.

Then future control actions can be written as a future vector, as:

$$\vec{U}(k) = \left[u(k+1) \;\; u(k+2) \;\; \cdots \;\; u(k+Nu) \right] \tag{6}$$

The expression of the differential can also be expressed in matrix form, through the Jacobian:

$$\frac{\partial J}{\partial \vec{U}(k)} = \left[\frac{\partial J}{\partial u(k+1)} \quad \frac{\partial J}{\partial u(k+2)} \quad \cdots \quad \frac{\partial J}{\partial u(k+Nu)} \right] \tag{7}$$

Some actualization rules are based on the first and second differential of the function. One of the most usual is the Newton-Raphson's rule, that can be expressed as:

$$u(k+1) = u(k) - \left(\frac{\partial^2 J}{\partial^2 u(k)} \right)^{-1} . \frac{\partial J}{\partial u(k)} \tag{8}$$

In this case, the second order differential can be written as a Hessian matrix:

$$\frac{\partial^2 J}{\partial \vec{U}^2} = \begin{bmatrix} \frac{\partial^2 J}{\partial u(k+1)^2} & \cdots & \frac{\partial^2 J}{\partial u(k+Nu).\partial u(k+1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 J}{\partial u(k+1).\partial u(k+Nu)} & \cdots & \frac{\partial^2 J}{\partial u(k+Nu)^2} \end{bmatrix} \tag{9}$$

Predictive controllers based on ANN, with applications of these actualization rules and also with different optimization functions can be found in Soloway and Haley (1997), Schnitman and Fontes (1998), Tan and Cauwenberghe (1996), Noriega and Wang (1998).

3. The ANN structure

Feedforward networks with a single hidden layer using threshold or sigmoid activation functions are universally consistent estimators of binary classifications (Farago and Lugosi, 1993). Also cited in Tan and Cauwenberghe (1996) and Hecht-Nielsen (1990), it is proven that a neural net, with only one hidden layer, is able to represent any function of $R^n \rightarrow R^m$, just limited by the number of neurons in the hidden layer.

3.1 The output general equations

Consider a real nonlinear model expressed by:

$$y(k+1) = f\left[y(k), y(k-1), .., y(k-n), u(k), u(k-1), ., u(k-m)\right] \tag{10}$$

where *f* is a nonlinear function of the system output *[y(k), y(k-1),..., y(k-n)]* and input values *[u(k), u(k-1),..., u(k-m)]*. The variable *n* is the number of auto regressors and *m* is the number of exogenous regressors.

But, to represent dynamic systems, it is necessary to introduce the feedback effect in the ANN, characterizing the application of the Recursive ANN (RANN). As in the ARX models (AutoRegressive with eXogenous inputs), the input signals of the net are associated to its own input and output past values. This structure specifically characterize a TDNN (Time Delay Neural Network). Then, let use a net with three layers, and base the study on the neural structure illustrated in the Figure 2 .

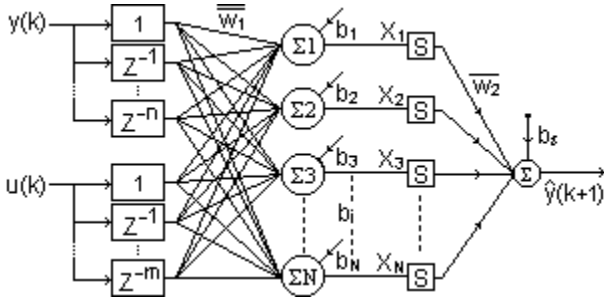

Figure 2: The structure of used TDNN

where *N* represents the number of the hidden layer neurons, the b_i are the hidden layer neurons bias and the *S* blocks represents a single-valued monotonic nonlinear sigmoid function to be applied at the output of each hidden layer neuron. b_s represents the bias of the output linear neuron. The matrixes of

weights w_1 and w_2 have dimensions *N* x (*n+m*) and *1* x *N* respectively and represent the weights of the connections among layers 1-2 and 2-3.

The generic expression of the proposed TDNN model is:

$$\hat{y}(k+1) = bs + \sum_{i=1}^{N} w_2(1,i).S(Xi) \tag{11}$$

where:

$$Xi = b(i,1) + \sum_{j=1}^{n} w_1(i,j).y(k-j+1) + \sum_{j=1}^{m} w_1(i,n+j).u(k-j+1) \tag{12}$$

that represents the estimated output $\hat{y}(k+1)$ as a nonlinear function of the system input and output.

3.2 Neural system differential equations

In general, the application of a control law requires to compute the expression of the output differential with respect to the process input. Based on (11) and differentiating it with respect to input values, we can generalize the expression:

$$\frac{\partial \hat{y}(k+1)}{\partial u(k)} = \frac{\partial}{\partial u(k)}\left[bs + \sum_{i=1}^{N} w_2(1,i).S(Xi) \right] \tag{13}$$

and it can be rewritten as:

$$\frac{\partial \hat{y}(k+1)}{\partial u(k)} = \sum_{i=1}^{N} w_2(1,i).S'(Xi).\frac{\partial Xi}{\partial u(k)} \tag{14}$$

where

$$S' = \frac{dS}{dXi} \tag{15}$$

Then, (14) represents the expression of the first order differential and S' is defined as Based on (12), the expression can be expanded:

$$\frac{\partial Xi}{\partial u(k)} = \frac{\partial}{\partial u(k)}\{b(i,1)\} + \frac{\partial}{\partial u(k)}\left\{ \sum_{j=1}^{n} w_1(i,j).y(k-j+1) \right\} + \frac{\partial}{\partial u(k)}\left\{ \sum_{j=1}^{m} w_1(i,n+j).u(k-j+1) \right\} \tag{16}$$

Note that the terms *y(k-1), y(k-2),..., y(k-n),* as well as the terms *u(k-1), u(k-2),..., u(k-m),* are past values, therefore, they do not depend on *u(k).* This way, the sum is allways null, except for *j=1,* in term *u(k)* whose differential is equal to 1. Hence:

$$\frac{\partial Xi}{\partial u(k)} = w_1(i, n+1) \tag{17}$$

substituting in (14):

$$\frac{\partial \hat{y}(k+1)}{\partial u(k)} = \sum_{i=1}^{N} w_2(1,i).S'(Xi).w_1(i,n+1) \tag{18}$$

The expression (18) generalizes the differential of used TDNN.

4. Neural predictive control

4.1 1 step ahead

The predictive controller, by definition, takes control actions based on predicted values for the system. Then, properly designed and well trained (Chauvim and Rumelhart, 1995), the neural net can represent the proposed nonlinear dynamic model. In so doing, control algorithms can be implemented based on errors between the reference signal, and the predicted value. See Figure 3.

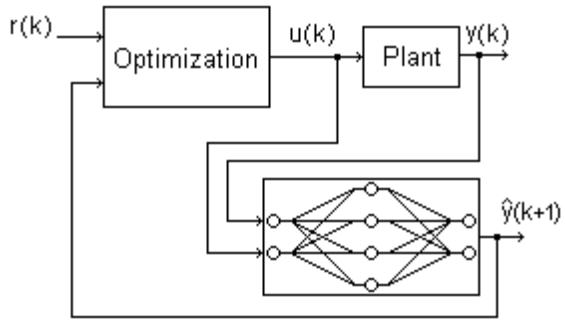

Figure 3: The NPC blocks diagram

where:

k = sample time
r(k) = reference signal
u(k) = control action
y(k) = plant output
e(k) = error between r(k) and the reference signal
$\hat{y}$ (*k+1*)= predicted plant output.

Observe that the control action is based on the predicted values and not in the real output. The model consider a TDNN properly trained and capable to satisfactorily represent the plant.

By definition, the predictive controllers seeks to minimize an performance index, considering aspects as error, control effort,.... For application, it is considered an index J as:

$$J = \frac{1}{2} e^2(k+1) \tag{19}$$

where:

$$e(k+1) = r(k+1) - \hat{y}(k+1) \tag{20}$$

And the rule that is used to update the control action is based on the decreasing gradient, so:

$$u(k+1) = u(k) - \lambda \frac{\partial J}{\partial u(k)} \tag{21}$$

Based on (19) and (20):

$$\frac{\partial J}{\partial u(k)} = \frac{1}{2}\left[-2.e(k+1).\frac{\partial \hat{y}(k+1)}{\partial u(k)}\right] = -e(k+1).\frac{\partial \hat{y}(k+1)}{\partial u(k)} \tag{22}$$

For the proposed TDNN, the expression of differential is already known. Then, using (18):

$$\frac{\partial J}{\partial u(k)} = -e(k+1).\sum_{i=1}^{N} w_2(1,i).S'(Xi).w_1(i,n+1) \tag{23}$$

With the obtained results, returning to (21), the control law can be written as:

$$u(k+1) = u(k) + \lambda.e(k+1).\left[\sum_{i=1}^{N} w_2(1,i).S'(Xi).w_1(i,n+1)\right] \tag{24}$$

And the control algorithm is implemented by the following steps:

1. Select λ
2. Use (11) and (12) to compute $\hat{y}(k+1)$
3. Use (20) to compute $e(k)$
4. Use (24) to compute the new control signal
5. Apply the new control signal to the system input.
6. Return to step 2.

4.2 T steps ahead

The preceding algorithm can be improved using predictive control techniques. Thus, scalar values can now be expressed as predicted vectors:

$$\vec{R} = \left[r(k+1), r(k+2), \ldots, r(k+T)\right] \tag{25}$$

$$\vec{Y}e = \left[\hat{y}(k+1), \hat{y}(k+2), \ldots, \hat{y}(k+T)\right] \tag{26}$$

$$\vec{U} = \left[u(k), u(k+1), \ldots, u(k+T-1)\right] \tag{27}$$

obtaining:

$$\vec{E} = \left[e(k+1), e(k+2), \ldots, e(k+T)\right] \tag{28}$$

Thus, the index to be minimized be in the form:

$$J = \frac{1}{2}\left[\vec{E}.\vec{E}^{T}\right] \tag{29}$$

Using the decreasing gradient rule, as (21):

$$\vec{U}(k+1) = \vec{U}(k) - \lambda \frac{\partial J}{\partial \vec{U}(k)} \tag{30}$$

where:

$$\frac{\partial J}{\partial \vec{U}(k)} = -\vec{E}.\frac{\partial \vec{Y}e}{\partial \vec{U}(k)} \tag{31}$$

The Jacobian matrix $\frac{\partial \vec{Y}e}{\partial \vec{U}(k)}$ can be found by differentiating (26) with respect to (27) and can be expanded as:

$$\frac{\partial \vec{Y}e}{\partial \vec{U}(k)} = \begin{bmatrix} \frac{\partial \hat{y}(k+1)}{\partial u(k)} & 0 & \cdots & 0 \\ \frac{\partial \hat{y}(k+2)}{\partial u(k)} & \frac{\partial \hat{y}(k+2)}{\partial u(k+1)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \hat{y}(k+T)}{\partial u(k)} & \frac{\partial \hat{y}(k+T)}{\partial u(k+1)} & \cdots & \frac{\partial \hat{y}(k+T)}{\partial u(k+T-1)} \end{bmatrix} \tag{32}$$

Some methods to compute the Jacobian can be found in Schnitman (1998) and Noriega and Wang (1998).

The NPC control algorithm can be implemented by the following steps:

1. Select λ and T.
2. Use (26) and (27) to compute $\vec{Y}e$ e $\vec{U}_k$.
3. Compute $\vec{E}$ using (28) .
4. Use (32) to compute $\frac{\partial \vec{Y}e}{\partial \vec{U}(k)}$
5. Use (31) to compute $\frac{\partial J}{\partial \vec{U}(k)}$.
6. Use (30) to compute the new control vector .
7. Apply the new control signal to the system input.
8. Return to step 2.

5. Plant for control

5.1 Plant differential equations

To TDNN control proposal is defined the nonlinear dynamic system according Figure 4, which is represented by the differential equation:

$$\frac{dh}{dt} = \frac{q_i . h^{-2}}{\beta} - \frac{K . h^{-3/2}}{\beta} \tag{33}$$

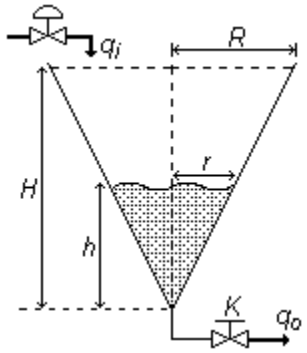

Figure 4: Model plant for control

where:

H = tank height R = tank radius
h = liquid level K = valve constant

$\beta = \pi . \left(\frac{R}{H} \right)^2$, tank constant

We propose to control the liquid level through the input volumetric flow rate (q_i). The valve that controls the output volumetric flow rate (q_o) is always open and presents nonlinear characteristic: $q_o = K\sqrt{h}$.

The characteristics of the system are:

$$H = 10m \quad , \qquad R = 5m \quad , \qquad K = 10$$

The operating point is set to h = 3m, then, for $\bar{h} = 3m$, $q_i = q_0 = K\sqrt{h} = 10\sqrt{3}$.

A level sensor represented by a first order system is integrated to the plant. Then *hr* represents the liquid level and *h* the measured level, in meters.

5.2 Linearization

From (33), at the operating point:

$$\frac{dh}{dt} = 0 \quad , \qquad \bar{h} = \textit{constant}, \qquad \bar{u} = K\sqrt{\bar{h}} .$$

The transfer function (34) is obtained by linearizing the system around the operating point:

$$\frac{H(s)}{U(s)} = \frac{\frac{4}{\pi}\left(\bar{h}\right)^{-2}}{S - \frac{20}{\pi}\left(\bar{h}\right)^{-5/2}} \tag{34}$$

then the static gain K_p and the *Pole* are functions of $\bar{h}$.

5.3 Proposed λ adaptability

It is observed that the plant nonlinearity and nonsymmetry make the plant output behaves differently for rising edge and falling edge of the reference signal. The idea is try to found an perfect adaptability to λ to compensate the plant characteristics.

Some empiric rules were used with satisfactory practical results. But appropriate results were obtained considering the adaptability of λ as a function of the dominant pole

displacement. Considering the plant linearization around the operating point (34), it is observed that the dominant pole behavior reflects exactly the necessary λ adaptability.

Using (34), to the proposed system we tuned in λ_0 and done:

$$\lambda = \lambda_0 . \left[\frac{20}{\pi} \left(y(k) \right)^{-5/2} \right] \tag{35}$$

These methods requires do adapt the proposed algorithms to calculate λ each step, and the practical results become excellent as showed in Section 7.

6. The used TDNN

According to Section 3, the neural net will be characterized by the number of auto regressors, the number of exogenous regressors and the number of hidden layer neurons. In the proposed application, we select the TDNN with only 6 hidden layer neurons (*N=6*), 4 auto regressors (*n=4*), 4 exogenous regressors (*m=4*) and use the hyperbolic tangent as sigmoid function.

After appropriate training, the validity of the model is verified in a large operation strip. The obtained results are presented in the Figure 5, it shows the level changing from 2 to 9m, that practically represents the whole tank. So, the proposed TDNN properly trained can satisfactorily represent the real model in a quite satisfactory way. Hence the net is ready to NPC implementation.

7. Results

The proposed control system is based on Figure 3 block implementation. The control algorithm will be executed according to (30) developed in the Section 4. The results are showed in this section. Just to compare, it also shows the result using the GPC and DMC controllers.

7.1 With constant λ

Figure 6 shows the plant output behavior with respect to reference signal and Figure 7 clearly shows the control action stability. It also illustrate de satisfactory performance of the NPC

Note in Figure 6 the value of λ is large enough to generate an output overshoot when the reference signal rises. However, it is too small to guarantee an effective time to stabilize the output plant when the reference signal falls.

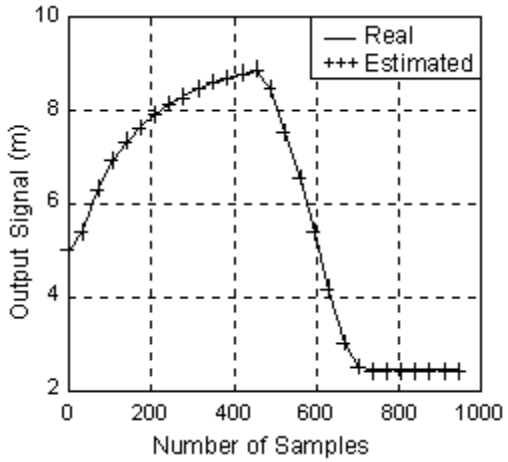

Figure 5: Real and estimated output signal

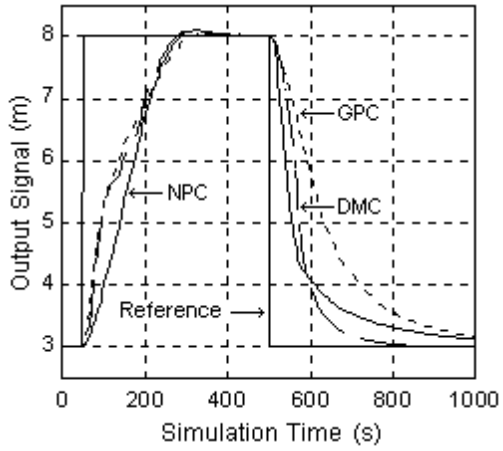

Figure 6: Reference signal and plant output with constant λ

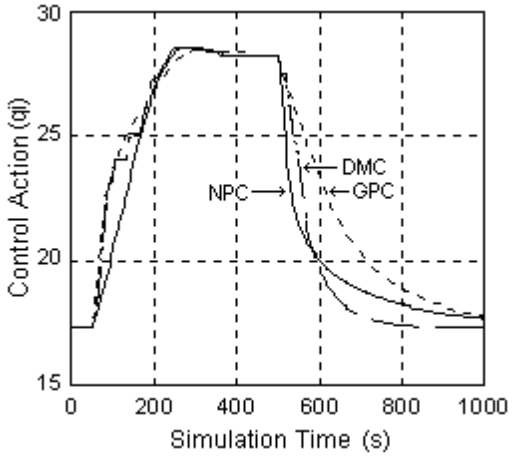

Figure 7: Control action with constant λ

7.2 With the proposed adaptability for λ

To enhance the system control action, as in Section 5-5.3, we propose an adaptive λ. It compensates the overshoot at the reference signal rises and the stabilization time at the falls. Then λ is maximum at the tank low limit and decreases when the output level rises.

As in (35) to the proposed system we tuned in $\lambda_0 = 8x10^{-6}$ and the obtained results is showed in Figure 8. Figure 9 illustrates the respective control action behavior.

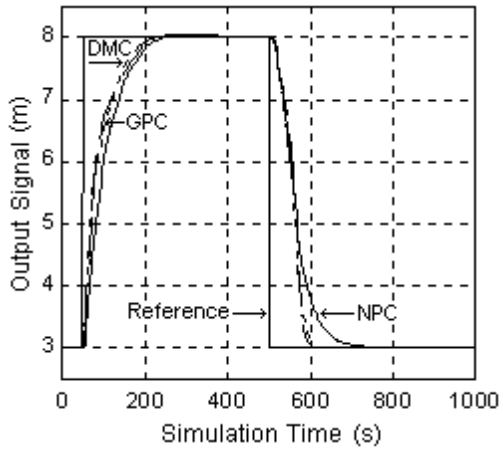

Figure 8: Reference signal and plant output with adaptable λ

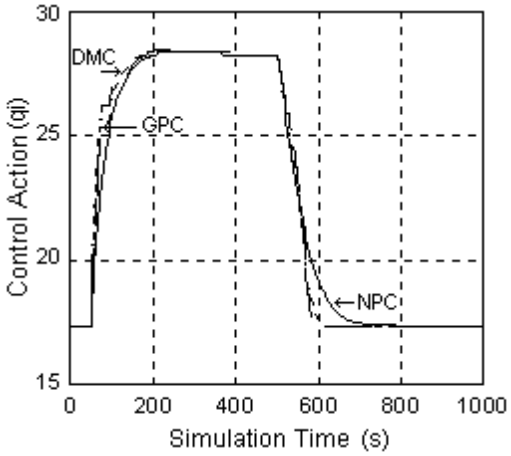

Figure 9: Control action with adaptable λ

8. Conclusions

It was clearly shown how to apply the predictive control concepts using the ANN.

It was observed the capacity of TDNN to model nonlinear plants. The expressions of TDNN and its respective prediction and control algorithms were developed and can be generalized to other control systems. They characterize a structured form to control nonlinear dynamic systems based on predictive techniques using ANN.

The proposed adaptability for λ based on the placement of the dominant pole have demonstrated excellent practical results in the control of the studied plant as shown in Figure 8 and Figure 9.

References

[1] Ahmed, M.S. and Anjum, M.F.: "*Neural-net-based direct self-tuning control of nonlinear plants*". International Journal of Control. Vol. 66, nº 1, pp. 85-104. 1997.

[2] Arnaldo, M.C.M.: "*Avaliação comparativa de desempenho entre controladores preditivos adaptativos GPC e DMC*". Dissertação de Mestrado do curso de Pós Graduação em Eng. Elétrica da UFBA. 1998.

[3] Astrom, K.J. and Wittenmark, B.: "*Adaptive control*". Addison-Wesley Publishing Company. 1995.

[4] Brown, M.D.; Lightbody, G. and Irwin, G.W.: "*Nonlinear internal model control using local model networks*". IEE Proceedings. Control Theory Application. Vol. 144, nº 6, pp. 505-514. November, 1997.

[5] Brown, M. and Harris, C.: "*Neurofuzzy adaptive modelling and control*". Prentice Hall. 1994.

[6] Canney, W.M. and Morari, M.: "*Experimental study of internal model control*". Industrial Eng. Chem. Process Des. Dev. Vol 25, nº 1, pg 102-108. 1986.

[7] Chauvim, Y. and Rumelhart, D.E.: "*Backpropagation theory: architectures and applications*". Laurence Erlbaum Associates Inc. 1995.

[8] Chen, C.L. and Chang, F.Y.: "*Design and analisis of neural/fuzzy variable structural PID control systems*". IEE Proceedings. Control Theory Application. Vol. 143 nº 2, pp. 200-208. March, 1996.

[9] Cichocki A. and Unbehauen R.: "*Neural networks for optimization and signal processing*". John Wiley & Sons Ltd. & B.G. Teubner. 1995.

[10] Clarke, D.W.; Mohtadi, C. and Tuffs, P.S.: "*Generalized predictive control. Part I: the basic algorithm*". Automatica, Vol. 23 nº 2, pp. 137-148. 1987.

[11] Clarke, D.W.; Mohtadi, C. and Tuffs, P.S.: "*Generalized predictive control. Part II: extensions and interpretations*". Automatica, Vol. 23 nº 2, pp. 149-160. 1987.

[12] Farago, A. and Lugosi, G.: "*Strong universal consistency of neural network classifiers*". IEEE Transactions on Information Theory. Vol. 39, pp. 1146-1151. 1993.

[13] Garcia, C.E.; Prett, D.M. and Morari, M.:"*Model predictive control: theory and practice - a survey*". IFAC Workshop on Model Based Control, Atlanta-USA, pg 335-348. June, 1989.

[14]Haykin, S.: "*Neural Networks: a comprehensive foundation*". Macmillian College Publishing Company Inc. 1994.

[15]Hecht-Nielsen, R.: "*Neurocomputing*". University of California Inc. 1990.

[16] Hunt, K.J.; Sbarbaro, D.; Zbikowski, R. and Gawthrop, P.J.: "*Neural networks for control systems - a survey*". Automatica. Vol. 28 nº 6. pp. 1083-1112. 1992.

[17] Hyland, D.C.; Collins, E.G.Jr.; Haddad, W.M. and Hunter, D.L.: "*Neural network system identification for improved noise rejection*". International Journal of Control. Vol. 68, nº 2, pp. 233-258. 1997.

[18] Jagannathan, S.; Lewis, F.L. and Pastravanu, O.: "*Discrete-time model reference adaptive control of nonlinear dynamical systems using neural networks*". International Journal of Control. Vol. 64, nº 2, pp. 217-239. 1996.

[19] Jang, J.R. and SUN, C.: "*Neuro-fuzzy modeling and control*". Proceedings of the IEEE. Vol.83, nº 3, pp. 378-406. 1995.

[20] Leu, Y.; Lee, T. and Wang, W.: "*On-line tunning of fuzzy-neural network for adaptive control of nonlinear dynamical systems*". IEEE Transactions on System, Man and Cybernetics. Vol 27, nº 6, pp. 1034-1043. December, 1997.

[21] Miller, W.T.; Sutton, R.S. and Werbos, P.J.: "*Neural networks for control*". MIT Press, Cambridge, MA. 1990.

[22] Mills, P.M.; Zomaya, A.Y. and Tadé, M.O.: "*Adaptive model-based control using neural networks*". International Journal of Control. Vol. 60, nº 6, pp. 1163-1192. 1994.

[23] Narendra, K.S. and Parthasaranthy, K.: "*Identification and control of dynamical systems using neural networks*". IEEE Transactions on Neural Networks. Vol 1, nº 1, pp. 4-27, 1990.

[24] Narendra, K.S.: "*Neural networks for control: theory and practice*". Proceedings of the IEEE. Vol.84, nº 10, pp. 1385-1406. 1996.

[25] Narendra, K.S. and Mukhopadhyay, S.: "*Adaptive control using neural networks and approximate models*". IEEE Transactions on Neural Networks. Vol. 8, nº 3, pp. 475-485. May, 1997.

[26] Noriega, J.R. and Wang, H.: "*A direct adaptive neural-network control for unknown nonlinear systems and its application*". IEEE Transactions on Neural Networks. Vol 9, nº 1, pp. 27-33. 1998.

[27] Richalet, J.; Rault, A.; Testud, J.L. and Papon, J.: "*Model predictive heuristic control: application to industrial processes*". Automatica, Vol. 14, pp. 413-418. 1978.

[28]Schnitman, L.: "*Controle preditivo baseado em redes neurais artificiais*". Dissertação de Mestrado do curso de Pós Graduação em Eng. Elétrica da UFBA. 1998.

[29] Schnitman, L and Fontes, A.B.: "*Predictive and adaptive nonlinear dynamic control system using neural networks*". V Simpósio Brasileiro de Redes Neurais. Minas Gerais, Dezembro - 1998.

[30] Schnitman, L; Arnaldo, M.C.M. and Fontes, A.B.: "*Controladores preditivos: GPC, DMC, redes neurais*". Seminário de Instrumentação e Automação, Instituto Brasileiro de Petróleo. Rio de Janeiro, Outubro - 1998.

[31] Soloway, D. and Haley, P.: "*Neural generalized predictive control: a Newton-Raphson implementation*". NASA Technical Memorandum 110244. February, 1997.

[32] Tan, Y. and Cauwenberghe, A.V.: "*Nonlinear one-step-ahead control using neural networks: control strategy and stability design*". Automatica, Vol. 32 pp.1701-1706, nº12. 1996.

[33] Widrow, B. and Lehr, M.A.: "*30 years of adaptive neural networks: Perceptron, Madaline and backpropagation*". Proceedings of the IEEE, Vol. 78, Nº 9, pp. 1415-1441. September, 1990.

[34]Widrow, B. and Lehr, M.A.: "*Adaptive neural networks and their applications*". International Journal of Intelligent Systems. Vol. 8, pp. 453-507. 1993.

[35] Widrow, B.; Rumelhart, D.E. and Lehr, M.A.: "*The basic ideas in neural networks*". Communications of the ACM, Vol. 37, Nº 3, pp. 87-92. March, 1994.

[36] Widrow, B.; Rumelhart, D.E. and Lehr, M.A.: "*Neural networks: applications in industry, business and science*". Communications of the ACM, Vol. 37, Nº 3, pp. 93-105. March, 1994.

[37] Willis, M.J.; Montague, G.A.; Di Massimo, C.; Tham, M.T. and Morris, A.J.: "*Artificial neural networks in process estimation and control*". Automatica. Vol. 28 nº 6. pp. 1181-1187. 1992.

[38] Yu, S.H. and Annaswamy, A.M.: "*Adaptive control of nonlinear dynamic systems using θ-adaptive neural networks*". Automatica, Vol. 33, nº 11, pp. 1975-1995. 1997.