IFAC MCPL 2007
The 4th International Federation of Automatic Control Conference on Management and Control of Production and Logistics
September 27-30, Sibiu - Romania

NEURAL-ADAPTIVE CONTROL BASED ON ADALINE NEURONS WITH APPLICATION TO A POWER SYSTEM

Ioan Filip, Octavian Prostean, Florin Dragan, Cristian Vasar

Department of Control System Engineering,
"Politehnica" University from Timisoara,
Faculty of Automation and Computer Science,
300223 Timisoara, Romania, Phone: (+40) 0256-403238,
E-Mail: ioan.filip@aut.upt.ro

Abstract: A neural-adaptive control solution is exposed in this paper. The control strategy is based on the linear adaptive neuron, which is called ADALINE. Unlike other neural control solutions, based on perceptrons neurons characterized by a long time learning process and a difficult on-line tuning of weights, this approach uses a fast algorithm, which adapts on-line the neuron's weights. Therefore the non-linear character of control law is induced by the permanent changes of neuron weights, which are variable parameters of controller. A set of study cases is done, with application to the excitation control of a synchronous generator.

Keywords: adaptive control, neural networks, ADALINE, modelling, simulation, synchronous generator.

1. INTRODUCTION

Using artificial neural networks to design viable process control strategies represents alternative solutions for classical control approaches. However neural network training process, due to its relative long time interval, represents in most of cases an impediment that limits its applicability domains. The previous observation concerns the applications of artificial neural networks based on perceptron neurons (having nonlinear activation functions and distributed in one or more hidden layers) to real time control of nonlinear complex process, where the time factor is critical. In such cases, the high computational training methods involve a long time, and many times the training results are dependent of the initial values chosen for network weights. (Ardalani *et al.*, 2005). Therefore, an on-line control strategy based on such artificial neural network becomes difficult to be implemented in many cases concerning the control of plants characterized by a complex functionality. However, if such control method is chosen from different reasons, after a successful off-line training of a neural network (learning only a limited desired functionality), to reach the objectives of a control strategy and to extend the functionality domain, classical control components (proportional, integrative) are often attached to the neural kernel of controller (somehow similar to fuzzy controller with external dynamic). (Filip *et al.*, 2006a). In this context, an artificial neural network based on ADALINE neuron (Adaptive Linear Neuron) can be chosen as efficient and viable solution, to design and implement adaptive control strategies for nonlinear process with a complex dynamic. Unlike the neural network based on perceptrons, which involves a long time adapting of the weights, the main characteristic of ADALINE neuron consist of its possibility to on-line adapt the neuron weights using a relative simple computational algorithm, easily to be implemented, relatively similar to a process parameters estimator used in a classic self-tuning control strategy. The aim of this paper is to analyse the possibility of using such adaptive control system based on ADALINE network with application to the excitation control system of synchronous generator, a process with a complex dynamic, time varying parameters and stochastically perturbed. Also, due to similarities with a self-tuning controller, some remarks between the two control structures performances will be presented. The self-tuning control already represents a classical solution with outstanding performances for the cases of synchronous generator connected to an electric power system (Filip *et al*., 2006b). The inconvenience of this classical approach consists in the high computation effort corroborated with a potential numerical instability of the parameters estimation algorithm due to varying regimes of such process, problem solved by a control strategy based on ADALINE. (Ardalani *et al*., 2005)

2. ADALINE NEURAL NETWORKS

The power of ADALINE consist in the on-line training capacity, through a permanent adjusting of its weights based on a supplementary adapting mechanism, externally attached to the neuron. Considering a certain evolution of neuron output ($y(t)$), the aims of the training (adapting) process consist in a continuous tuning of neuron weights to minimize the learning error ($e(t)$) as fast as possible. This adapting method, is very often named "supervised learning", involving the presence of a "trainer", able to provide the desired output for the full on-line learning process. (Benedito and Eduardo, 1998; Derrick *et al*., 1990; Widrow and Stearns., 1985) The mathematical model of an ADALINE neuron is practically identical with that of a McCulloch-Pitts neuron (perceptron); the differences are the type of the activation function and the on-line weights adapting mechanism, structural attached to neuron. The ADALINE neuron structure is presented in figure 1.

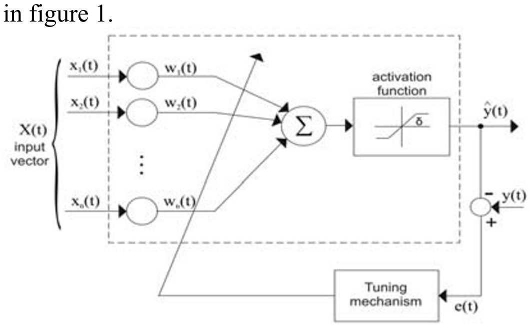

Fig. 1. ADALINE neuron structure

The training phase, takes place on-line in case of ADALINE neuron, unlike the McCulloch-Pitts neuron where the training is an off-line preliminary phase. Even the name of the neural element indicates that its activation function is linear, its input-output transfer characteristic become non-linear due to permanently varying of the weights (which are on-line adapted). An ADALINE input-output function is described by a mathematical relationship having the following form:

$$\hat{y}(t) = \delta[\hat{w}_1(t)x_1(t) + \hat{w}_2(t)x_2(t) + ... + \hat{w}_n(t)x_n(t)] \quad (1)$$

where: $x_1(t), x_2(t),..., x_n(t)$ - neuron inputs; $\hat{w}_1(t), \hat{w}_2(t),..., \hat{w}_n(t)$ - on-line tuning weights calculated by a learning mechanism; $\hat{y}(t)$ - neuron trained output; δ - linear characteristic of activation function (usually inferior and superior limited).

From this point it will be assumed $\delta = 1$ (without affecting the generality of the considered problem). Also, two basic variables can be noticed in figure 1:

- $y(t)$ -plant output (set point);
- $e(t) = y(t) - \hat{y}(t)$ - learning error.

3. ADALINE CONTROLLER

ADALINE controller design strategy is based mostly on fundamental principles of classical adaptive self-tuning controllers. (Hunt *et al*., 1992) On-line parameters estimation algorithm is substituted, in this case, with an on-line learning mechanism (computing the neuron weights which models the controlled process). This neural model, implemented using an ADALINE neuron, has the role to learn as accurate as possible the dynamic of real process.

In closed loop, the real process represents practically a reference model, whose dynamic must be on-line learned by a neural model. (Marei *et al*., 2004) The learning phase consist practically in a continuous adjusting of ADALINE neuron weights, that models the real process, in order to minimize, as fast as possible, the error between the real process output and the neural model output (the learning error). The weights of identified neural model, represents the primary information that, based on an adequate computing strategy (implemented by the computing block within figure 2), provides the controller's adaptable parameters (practically the weights of the second ADALINE neuron that implements the controller). The parallelism between the classical adaptive controllers and the considered neural-adaptive controller is evident, based on the presented designing and functioning strategy. The general structure of such neural-adaptive control system is depicted in figure 2.

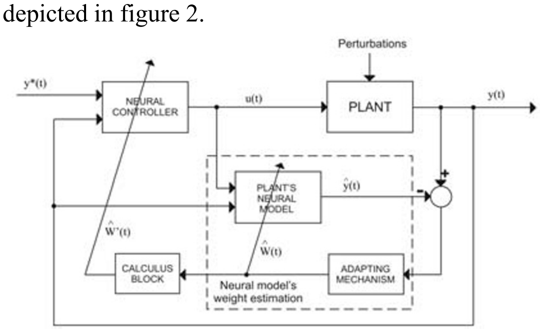

Fig. 2. The structure of neural-adaptive control system

We assume the following notation:

$\hat{W}(t) = [\hat{w}_1(t), \hat{w}_2(t),..., \hat{w}_p(t)]^T$ - the weights vector of neural model identifying the real process (on-line tuneable) and

$\hat{W}'(t) = [\hat{w}_1'(t), \hat{w}_2'(t),..., \hat{w}_p'(t)]^T$ -the neural controller's weights vector. The calculation of