'94 KACC (1994. 10. 17 ~ 20)

Controcller Design Using Parametric Neural Networks

M. HashemiNejad , J. Murata and K. Hirasawa
Dept. of Electrical Engineering,

M. E. Banihabib
Dept. of Civil Engineering,

Faculty of Engineering, Kyushu Univ. Japan

Abstract

Neural Networks (henceforth NNs, with adjective "artificial" implied) has been used in the field of control however, has a long way to fit to its abilities. One of the best ways to aid it is "supporting it with the knowledge about the linear classical control theory". In this regard we have developed two kinds of parametric activation function and then used them in both identification and control strategy. Then using a nonlinear tank system we are to test its capabilities. The simulation results for the identification phase is promising.

1 Introduction

Our main objective is to make a good controller for nonlinear systems. Neural network is a promising tool for nonlinearity modeling and controller design; because of its capability of learning and representing a wide class of, all theoretically continuous bounded, nonlinear mappings. However, when applying a NN to identification or controller design [1]-[4], the architecture of the network has significant influence on the result, learning speed and learning error, and therefore we have to choose appropriate architecture. And the network's capability of approximating various mappings is achieved at the cost of uncleanness of its inner structure. The NNs are usually black boxes; we can not obtain any information about the inner structure of the NN corresponding to system or controller. In identification, accuracy of the derived model is very important. And if we could have some information about the *inner structure* of the system to be identified, it would be useful.

We can solve both of the above mentioned problems simultaneously, should we find an automatic method for determining the network architecture which achieves good accuracy and provides us with the system structure information. Here we restrict ourselves to choosing neuron activating functions suitable for the given task which the network should undertake. As some researchers have already had some relevant researches [5]- [8], introducing additional adjustable parameters to the activation function reduces the learning error. Here we utilize the additional parameters not only for attaining better accuracy but also obtaining information about the system structure. Also, if the additional parameters are placed suitably, they can change the shape of the function covering from a linear function to a sigmoid function with steep shape. This subject will be considered in more details in the next section.

As the final aim of modeling would be control, the problem of making an appropriate controller is important. In the most common cases designers use linear control theory, which is well established, to approximate the nonlinear systems and then making a linear controller. However if the plant can not be well approximated then reaching to a rigorous controller will be difficult. Here using the linear classical control theory we customize the parametric NN to make an appropriate initial controller. In section 3.2 we will develop such a controller in conjunction with linear truncated function. Finally we will study how we can apply this method to control water level in a nonlinear plant.

2 Parametric Neural Networks

2.1 Network Configuration

In ordinary NNs as the activation function is nonlinear there is no way to initiate the network's output with any reference value, if there is any. The possibility of starting the NN with a certain value can incorporate the existing knowledge of the linear system theory for a better transient response of the control system.

Studying the biological creatures like Aplysia buccal ganglia presents evidence that synaptic strengths are partially specified by postsynaptic neurons [9]. This fact encourages us to search for some kind of network which introduces intrinsic relationship between neurons and weights. The aims for searching new activation function are:

Firstly, we seek such a network which has good accuracy and high learning speed.

Secondly, gaining more information about the system would be a great assistance in identification.

Thirdly, having the new activation function we should be able to set it as a linear/nonlinear function, when we need so, to customize a linear initial controller.

The more number of parameters are introduced in a network, the more describing ability it obtains. Especially introduction of additional parameters to the activation functions and changing their shape by adjusting the parameters will favor in giving more flexibility to the network: it will be able to represent a wide variety of input-output mappings which are different from each other in complexity and smoothness. Also we would like to obtain some insight into the system structure from the NN model. Thus we suggest to use a parametric sigmoid function

$$f_p(x) = \frac{1}{\ln p} \cdot \tanh(\ln px) \tag{1}$$

as an activation function.

By changing p it will be an unfixed and *interactive* function. This new shape of the activation function is favorable because using $\ln p$ instead of p prevents from excessive reduction of the absolute value of the function.

Expanding equation 1 can help us to see the effect of changing p on the sigmoid function's shape

$$f_p(x) = \frac{(2x) - \frac{\ln p(2x)^2}{2!} + \frac{(\ln p)^3(2x)^3}{3!} - \cdots}{2 - (\ln p)x + \frac{2(\ln p)^2 x^2}{2!} - \frac{4(\ln p)^3 x^3}{3!} + \cdots} \tag{2}$$

Studying the above equation one can understand how easily a very changeable sigmoid function can be obtained. This variety starts from a, linear, line and can develop to nonlinear ordinary sigmoid function. By setting p to 1 we can get a linear function instead of sigmoid function. The shapes for different ps are depicted in Fig. 1. The proposed activation function will realize:

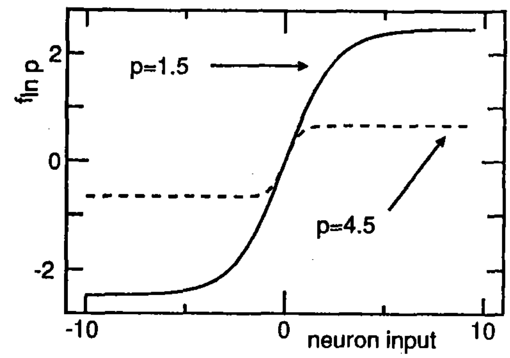

Figure 1: Two shapes of sigmoid function

- ◊ High flexibility which results in less error bound.
- ◊ Sensitizing the network to linear/nonlinear parts.
- ◊ Containing more information based on linear control theory for making initial controller.

The parameter p can be tuned along the weights to minimize the error between the network output and the teaching signal. If $p = 1$ is obtained after tuning, it means that neuron is linear.

2.2 Parameters Learning

Extending the backpropagation can be a good mean for training the new parameter. In the case of the weights it is one of the common method and here also we use that for pushing off the error function more toward , hopefully, global minimum.

In Fig. 2 the parameter p is updated so to minimize the squared sum of errors between the network outputs and their desired values:

$$E = \frac{1}{2}\sum_k \{t(k) - o_o(k)\}^2, \tag{3}$$

where $t(k)$ is the desired value for the kth output. The update will be done as,

$$p_{new} = p_{old} - \beta \frac{\partial E}{\partial p_{old}}, \tag{4}$$

where β is learning rate. Let us define error signals as,

$$\delta_{oo}(k) \triangleq \frac{\partial E}{\partial o_o(k)} = o_o(k) - t(k), \tag{5}$$

$$\begin{aligned} \delta_{oi}(k) &\triangleq \frac{\partial E}{\partial x_o(k)} = \frac{\partial E}{\partial o_o(k)} \frac{\partial o_o(k)}{\partial x_o(k)} \\ &= \delta_{oo}(k) \cdot \tanh'(\ln p_k \cdot x_o(k)), \end{aligned} \tag{6}$$

$$\delta_{ho}(j) \triangleq \frac{\partial E}{\partial o_h(j)}$$

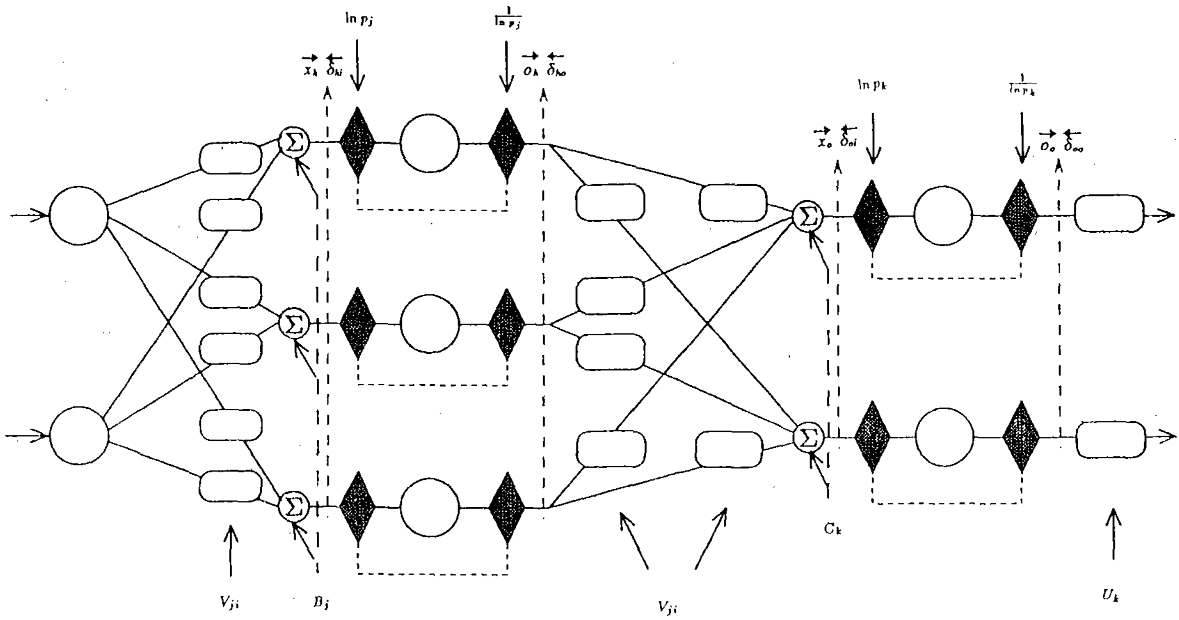

Figure 2: Network structure

$$= \sum_k \frac{\partial E}{\partial x_o(k)} \frac{\partial x_o(k)}{\partial o_h(j)} = \sum_k \delta_{oi} \cdot W_{kj}, \quad (7)$$

$$\delta_{hi}(j) \triangleq \frac{\partial E}{\partial x_h(j)} = \frac{\partial E}{\partial o_h(j)} \frac{\partial o_h(j)}{\partial x_h(j)}$$
$$= \delta_{ho}(j) \cdot \tanh'(\ln p_j \cdot x_h(j)). \quad (8)$$

These signals can be calculated in turn. This is nothing but the backpropagation. The gradient $\partial E / \partial p$ can be obtained using the above signals. For p in the kth output neuron,

$$\frac{\partial E}{\partial p_k} = \frac{\partial E}{\partial o_o(k)} \frac{\partial o_o(k)}{\partial p_k}$$
$$= \delta_{oo}(k) \langle \left(\frac{d}{dp_k} \frac{1}{\ln p_k} \right) \cdot \tanh(\ln p_k \cdot x_o(k))$$
$$+ \frac{1}{\ln p_k} \cdot \tanh'(\ln p_k \cdot x_o(k)) \frac{d \ln p_k}{dp_k} x_o(k) \rangle \quad (9)$$
$$= \frac{-1}{p_k \cdot \ln p_k} \{ \delta_{oo}(k) \cdot o_o(k) - \delta_{oi}(k) \cdot x_o(k) \}. \quad (10)$$

For p in the hidden neurons, the gradients can be calculated as,

$$\frac{\partial E}{\partial p_j} = - \frac{1}{p_j \cdot \ln p_j} \{ \delta_{ho}(j) \cdot o_h(j) - \delta_{hi}(j) \cdot x_h(j) \}. \quad (11)$$

Then the parameter p can be updated using (4). Of course, all the weights can be updated using the above error signals, too.

3 Controller Design Using Parametric NNs

3.1 System Modeling & Control

Parametric adaptive control is the problem of controlling the output of a system with a known parameters. To make the problem analytically tractable, in the classical adaptive system control theory the plant to be controlled is assumed to be linear time-invariant with unknown parameters. These parameters can be considered as the elements of a vector m. If m is known, the parameter vector θ of a controller can be chosen as θ^* so that the plant together with the fixed controller behaves like a reference model described by a linear difference (or differential) equation with constant coefficients. If m is unknown, the vector $\theta(t)$ has to be adjusted on-line using all available information concerning the system.

In contrast we uses NNs which permit us to go beyond the linear models and controllers. There have been two major strategies treating the problem of adaptive control of an unknown plant using NNs. The first is direct control and the second is indirect control. Figure 3 shows the direct control scheme. In this method the parameters of the controller are directly adjusted to reduce some norm

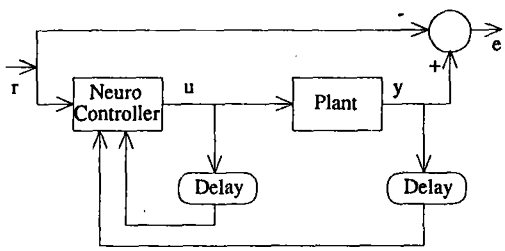

Figure 3: Direct Controller

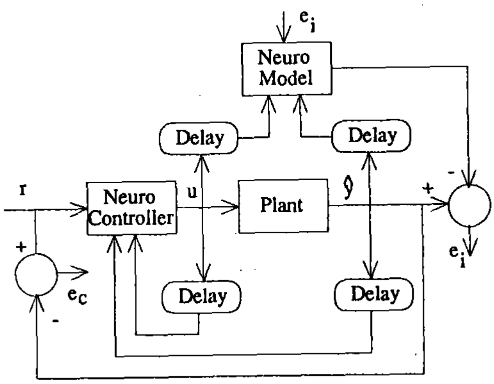

Figure 4: Indirect Controller

of the output error. However if both plant and controller are nonlinear this method will have the stability problem because, we can't adjust the controller parameters by the output error [10]. Figure 4 provides the configuration for the indirect control scheme. In this method at first by using one of the known criteria we can parameterize the plant. Then the controller parameters, in turn, are adjusted by backpropagation the error through the identified model.

3.2 Initial Controller

The adaptive control of linear system is well developed thus the principal use of NNs is concentrated on the control of nonlinear systems. However, in this relation, one of the most important problems is: "how we can use the well established linear control theory in accompany with the powerful abilities of the NNs". When we approximate a nonlinear system with a linear model although we can not cover all of the characteristics of the system by such a model but it isn't comparable with a controller that its parameters has been adjusted randomly. On the other hand if we want to control a system by the indirect method we will have the problem of the initiating the controller. One

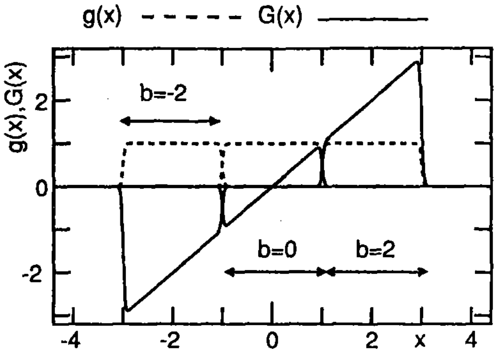

Figure 5: Shapes of the truncated linear Function

of the promising ways is to start the controlling process with a linear controller and then developing it to a NN controller. Until now in the related articles such as[11] some different sort of this approach have been discussed but there has been problem yet. It needs lots of work to find a suitable transition process, from a linear controller to a fully nonlinear controller.

Here we want to introduce a suitable bridge between the linear and nonlinear controller. In the other words we initiate the control process by a linear controller and then push it off toward a nonlinear controller. This controller will be designed as a unique NN. By using this method the knowledge of the linear control theory will be used in conjunction with the powerful abilities of NNs. As the first step let us explain a new, truncated linear, function which will be used in the initial controller,

$$G(x) = \frac{x \cdot \sinh ah}{\cosh a(x-b) + \cosh ah} \tag{12}$$

or,

$$G(x) = x \cdot g(x) \tag{13}$$

where,

$$g(x) = \frac{\sinh ah}{\cosh a(x-b) + \cosh ah} \tag{14}$$

Figure 5 represents a truncated linear function which, has been used in the first hidden layer of the new NN controller. In fact the first hidden layer of the controller, by having G(x) as activation function, divides the input space into few segments which they classify the signals and then bypass it to the next layers. By this method we can approximate an arbitrary nonlinear function with arbitrary small error. In the other word one parametric sigmoid function only has to approximate the function over each segment. Figure 6 shows such a NN controller. This net-

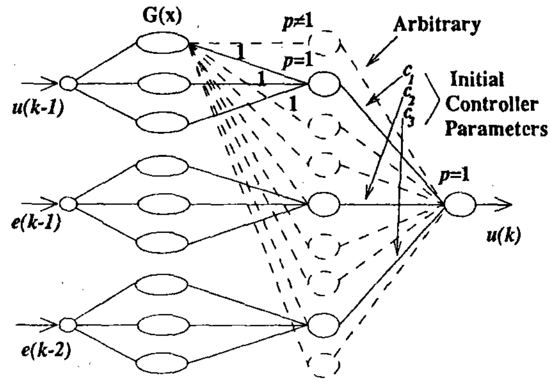

Figure 6: Initial Controller

work differs from Fig. 2. Here let us concentrate only on the bold lines, the dash lines will be considered in the next section. Neurons with circle, in the second and third layers, represent the parametric activation which was discussed in the second section. The p shows the neuron parameters' values which in the bold circles are chosen $p \simeq 1$ that corresponds to linear neuron(dash neurons will be considered in the next section).

3.3 Improved Controller

Starting the control process by linear controller would be the first step however, we will complete it by a nonlinear, NN, controller. The dash lines in the Fig. 6 has very small random values which means at the first the plant will be stimulated only by the linear part of the controller. Then, after passing the few seconds the learning will affect the dashed lines and they will become similar to the bold lines which means they has entered to, improve, the control strategy.

In this stage the parameters' learning will be start and not only the random setting but also the initial predetermined parameters like p_1, p_2, and $p_3$will be changed by the backpropagation of the error through the network.

4 Nonlinear Plant Control

4.1 Tank System

For testing the proposed controller we have found the tank system in Fig. 7 a suitable plant for the simulation. Here we have input u and outputs o_1 and o_2. This plant has two discontinuity which will occur when $l = 0$ or $l = 2R$ where $A(l) = 0$. Thus it would be a proper nonlinear

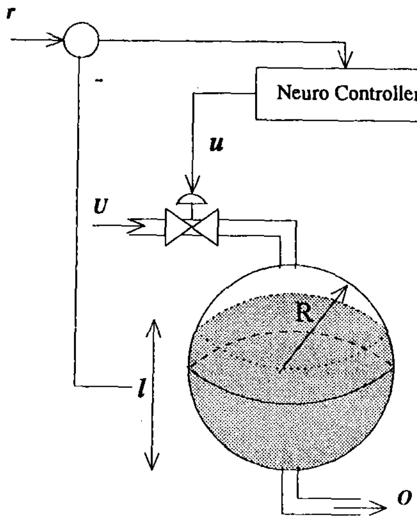

Figure 7: Tank System Control

plant to examine parametric NN controller. This model has a practical shape which in industry is used widely for its strong, spheric, structure. At first we have made a mathematical model for this tank which will be used for the NN training and its differential equation is:

$$\frac{dl}{dt} = \frac{K_v \frac{C_v}{100} - M \cdot \sqrt{2gl}}{\pi \cdot l(2R - l)} \tag{15}$$

where the parameters are define in table 1. Since the input discharge can be controlled by an electrical valve, it may be considered as a function of the time which has two operational ranges: a) $0 \leq s_h \leq 1$ and b) $1 \leq s_h \leq 100$ as bellow,

a) $C_v = \frac{1}{10000}(3228.65 s_h)$
b) $C_v = \frac{1}{10000}(-3168 s_h + 6610 s_h - 216 {s_h}^2 + 0.00265 {s_h}^3)$

Then the following scheme is used to calculate water level as a function of time:

$$l^{n+1} = \frac{l^n + \Delta t[(1-\theta)U^{n+1} + \theta U^n]}{[(1-\theta)A^{n+1}(l) - \theta A^n(l)]} - \frac{[(1-\theta)O^{n+1} + \theta O^n]}{[(1-\theta)A^{n+1}(l) - \theta A^n(l)]} \tag{16}$$

By this way we have chosen the mixed implicit & explicit [12] numerical methods for calculating the differential equation which has had very good results.

After making this model we used this model to train the parametric NN (as section 2.1-2.2). The results of this simulation is shown in Figs. 8 9.

And then we have used the initial controller scheme to build the linear part of the controller. After this scheme we would improve it to obtain the final controller.

variable	definition	dim.
l	level of the water	m
$A(l)$	water surface for water level l	m^2
U	input water	m^3
O	output water	m^3
dx/dt	rate of water level change in time	m/s
R	radius of the sphere tank	m
M	effective area of outlet cross section	m^2
C_v	effective area of inlet cross section	m^2
g	acceleration due to gravity	m/s^2
Δt	time step	s
s_h	input voltage of the valve	v
n	superscript for nth time step	—
$n+1$	superscript for $n+1$th time step	—
θ	parameter of the mathematical scheme	—

Table 1: Parameters & variables definitions

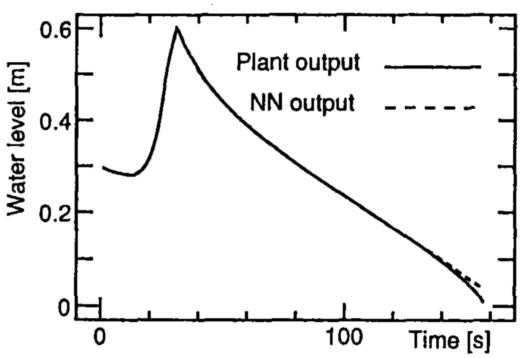

Figure 8: NN and plant output for tank system

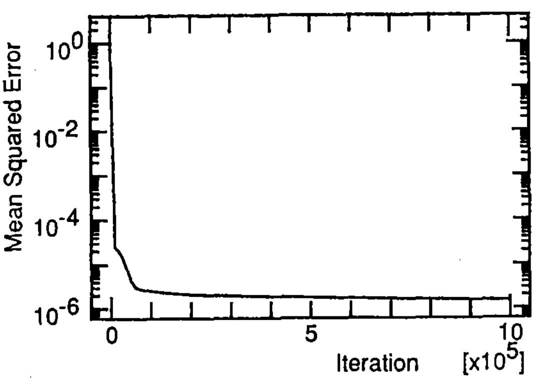

Figure 9: Error for the tank simulation

5 Conclusion

In this paper we have studied a new scheme for the controller design and then simulation results shows its properness for the first phase, or Identification.

References

[1] D. Psaltis, A. Sideris and A. Yamamura, "A Multilayered NN Controller" in IEEE Contr. Syst. Mag., vol. 8, pp. 17-21, Apr. 1988.

[2] M. Kawato et al. "Hierarchical NN model for Voluntary Movement with Application to Robotics" in IEEE Contr. Syst. Mag., vol. 8, pp. 8-16, Apr. 1988.

[3] D. H. Nguyen and B. Widrow, "NN for Self-Learning Control Systems, in IEEE Contr. Sys. Mag., vol. 10, pp. 18-23, Apr. 1988.

[4] K. S. Narendra and K. Parthasrathy, "Identification and Control of Dynamical Systems using NN" IEEE Trans. NN vol. 1, pp. 4-27, Mar. 1990.

[5] J. M. Zurada "Artificial Neural Systems" West Publishing Company, 1992.

[6] M. Arai, et al. "Adaptive Control of a NN with a Variable Function of a Unit and its Application" Trans. Inst. Electron. Inform. and Commun. Eng., vol. J74-A, no. 3, pp. 551-559, 1991(in Japanese)

[7] K. Song and J. Shieh "A NN Controller Based on Autotuning the Gain of the Activation Function" Proc. of the Int. Joint Conf. on NN, vol. 3, pp. 2787, 1993.

[8] T. Yamada and T. Yabuta "Neural Network Controller using Autotuning Method for Nonlinear Functions" Trans. on NN.,vol. 3, no. 4, pp. 595-601, 1992.

[9] D. Gardner "The Neurobiology of Neural Networks" MIT press, Cambridge, MA, 1993.

[10] W. Thomas Miller, III, Richard S. Sutton, and Paul J. Werbos "Neural Network for Control" The MIT press, Cambridge, Ma, 1990.

[11] G. M. Scott "Refining PID controller Using Neural Netwoks" Neural Computation, vol. 4, pp. 746-757, MIT, Sept. 1992.

[12] W. H. Press et al. "Numerical Recipes n C, the Art of Scientific Computing" Cambridge Univ. Press, 1992.