Robert W. Newcomb
Microsystems Laboratory
Electrical Engineering Department
University of Maryland
College Park, MD 20742
e-mail: newcomb@eng.umd.edu

Rui J.P. de Figueiredo
Department of Electrical and
Computer Engineering and
Department of Mathematics
University of California, Irvine
Irvine, CA 92717
e-mail: rui@uci.edu

A Multi-Input Multi-Output Functional Artificial Neural Network

Abstract

A generic two-layer feedforward functional neural network is proposed that processes functions rather than point evaluations of functions. Specifically, the network receives n functions as inputs and delivers m real values as outputs. Its architecture is derived using the nonlinear system identification techniques of Zyla and de Figueiredo. As such, neurons are represented by Volterra functions in Fock space, which is a reproducing kernel Hilbert space, with synaptic weights that are functions themselves. The main advantage is that this functional network can be used in the modeling of real-world (continuous-time parameter) nonlinear systems, capturing the dynamics presented in them, as well as in the simulation of their behavior in a computer-based environment. © 1996 John Wiley and Sons, Inc.

Introduction

Present day artificial neural networks normally take input data, weight them, and then sum the result into nonlinear sigmoidal-type output functions. Conventionally, the input consists of a numerical data set, such as the pixel intensity of a discretized image. As such these artificial neural networks have proven to be effective classifiers and pattern recognizers in situations where closed form mathematical solutions are hard to obtain. But in many situations the data is a set of functions, rather than their values at specific arguments. Such would be in the continuous-time parameter physical systems where one desires the input–output map as an operator rather than as a mapping corresponding to a discrete set of input–output data values (de Figueiredo and Dwyer, 1980). In the case of linear time-invariant systems such an operator could be specified by the transfer function or equivalently by the impulse response function. If we treat the latter in a functional way, for example as the kernel of the convolution input–output map, then we become interested in functional maps as descriptions of systems, and significant problems of systems modeling and identification become those of representing and identifying functional maps. Although the convolution functionals, represented by their kernels, characterize all linear time-invariant continuous systems, when we turn to the more prevalent nonlinear systems the situation is much more complex. Unfortunately the mathematics for general nonlinear systems (Holtzman, 1970) is still not fully developed in terms of obtaining practical results. However, the theory of Volterra functionals is developed to the point that, in an abstract setting, one can obtain a Volterra functional representation of a system given its sufficient input and output function pairings. Here we review the situation, as presented in Zyla and de Figueiredo (1983), for system identification via Volterra functionals in using a Fock space. Then we apply the results by introducing a functional neural network, with two hidden layers, that solves the minimum norm problem in a Bochner space related to the Fock space to which the Volterra functional belongs. In so doing the functional neural network is trained on

Journal of Intelligent and Fuzzy Systems, Vol. 4, 207–213 (1996)

CCC 1064–1246/96/030207-07

input–output exemplar function pairings to set the weights, which are functions themselves. Then the functional neural network carries out the system identification by associating a Volterra functional input–output map. The Fock concepts used are similar to the ones used for optimal interpolating (OI) (de Figueiredo, 1990) and optimal multilayer neural interpolating (OMNI) (de Figueiredo, 1992) networks.

Review of Fock Space Theory of Nonlinear System Identification

Because we are interested in characterizing dynamical systems described by nonlinear mappings V of input functions $u = u(\cdot)$ into output functions $y = y(\cdot)$, we consider the Volterra series representation as it is a description of great generality. To carry out an identification we specify a real interval I of time t over which identification is to be made. By definition we take $y(\cdot) = V_\cdot(u(\cdot))$ which when evaluated at time t is denoted $y(t) = V_t(u(\cdot))$; written as a Volterra series this is (de Figueiredo and Dwyer, 1980)

$$y(t) = V_t(u(\cdot)) = \sum_{k=0}^{\infty} \frac{1}{k!} \int_{t_1} \cdots \int_{t_k} h_k(t; t_1, \ldots, t_k) \times u(t_1) \cdots u(t_k)\, dt_1 \cdots dt_k , \quad (1)$$

in which the Volterra map $V_\cdot(\cdot)$ is characterized by the kernels h_k; these latter designate k-(multi)linear maps defined on the inputs as indicated by the integrals (all of which are taken over I here and in the following). We make all of the assumptions of Zyla and de Figueiredo (1983) on the spaces to which the various varibles and operators belong, reviewing some of these as we proceed. For simplicity of the treatment, we limit our attention in this section to the single-input single-output real-valued case, that is u and y are taken to be 1-dimensional real-valued functions of the real variable time; the extension to the multiple-input multiple-output cases are discussed later. Also, on physical grounds and in line with Zyla and de Figueiredo (1983), we assume that u and y and sufficiently many, n, of the outputs' derivatives have finite energy by taking all such functions to be square integrable over I.

The identification of the system as carried out in Zyla and de Figueiredo (1983) rests crucially upon the nonlinear Volterra functional $V_t(\cdot)$ belonging to a special reproducing kernel Hilbert space, called a Fock space and designated F_r, where r is a positive constant. The more general case of a weighted (generalized) Fock space is discussed in de Figueiredo and Dwyer (1980). The developments in the present article, with minor modifications, apply to this general case. Associated with a Volterra functional $V_t(\cdot)$ in F_r there is a Volterra operator $V_\cdot(\cdot)$ belonging to a Bochner space B_n^2. The restriction that $V_\cdot(\cdot)$ is in B_n^2 is equivalent to assuming the following three physically reasonable conditions:

1. The ith partial derivative, $h_k^{(i)} = \partial^i h_k / \partial t^i$, with respect to t of h_k exists everywhere on I for $i = 0, \ldots, n-1$ as a map from I into $L^2(I^k)$, where I^k is the k-dimensional cube of sides I, and the nth partial derivative with respect to t is a map from I into $L^2(I^{k+1})$.
2. There exists a real constant r such that
$$\sum_{k=0}^{\infty} \frac{r^k}{k!} \|h_k^{(i)}(\cdot; \cdot, \cdots, \cdot)\|^2_{L^2(I^{k+1})} < \infty \quad \text{for } i = 0, \ldots, n. \quad (2)$$
3. $h_k(t; t_1, \ldots, t_k)$ is invariant to permutations (that is, symmetric) in the variables $t_1, \ldots, t_k$.

Although not so important physically, the first condition is needed for guaranteeing the mathematical existence of the reproducing kernel to be introduced while the second is needed to guarantee convergence of the Volterra series. Condition three is of secondary importance because the nonsymmetric parts cancel out in the integrals anyway.

Given r we introduce the scalar product of two elements $V_t(\cdot)$ and $W_t(\cdot)$ of Fock space F_r as follows. Let h_k be the kernels for V and g_k be those for W, then

$$\langle V_t(\cdot), W_t(\cdot)\rangle_{F_r} = \sum_{k=0}^{\infty} \frac{r^k}{k!} \langle h_k(t; ., \ldots, .), g_k(t; ., \ldots, .)\rangle_{L^2(I^k)} , \quad (3a)$$

where the scalar product of any two (Lebesgue) square integrable functions g and h of k variables, $t_1, \ldots, t_k$, is given by

$$\langle g, h\rangle_{L^2(I^k)} = \int_{t_1}\int_{t_2}\cdots\int_{t_k} g(t_1, t_2, \ldots, t_k)\, h(t_1, t_2, \ldots, t_k)\, dt_1\, dt_2 \cdots dt_k\,. \tag{3b}$$

With this latter scalar product the Fock space of the Volterra functionals $V_t(\cdot)$ becomes a Hilbert space. There is also a Hilbert space associated with the Volterra operator $V.(\cdot)$ which will be needed for the system identification; thus the Bochner space B_n^2 becomes a Hilbert space if we associate with it the scalar product

$$\langle V.(\cdot), W.(\cdot)\rangle_{B_n^2} = \sum_{i=0}^{n-1} a_i \int_t \langle V_t^{(i)}(\cdot), W_t^{(i)}(\cdot)\rangle_{F_r}\, dt\,, \tag{4}$$

where the a_i are any chosen positive constants. We note that the operator $V.(\cdot)$, which maps the full input function $u(\cdot)$ into the full output function $y(\cdot)$, represents the system as a Bochner space map taking a time t in the time interval I into the Fock space Volterra map $V_t(\cdot)$ that maps full input functions $u(\cdot)$ into output functions, evaluating them at time t, that is into $y(t)$.

A reproducing kernel for F_r is the following functional $K(\cdot, \cdot)$ that maps $L^2(I) \times L^2(I)$ into the real numbers

$$K(u, v) = \exp\left(\frac{1}{r}\langle u, v\rangle_{L^2(I)}\right). \tag{5}$$

To see that this K is a reproducing kernel for the Hilbert Fock space, note that

$$K(u, \cdot) = \exp\left(\frac{1}{r}\langle u, \cdot\rangle_{L^2(I)}\right). \tag{6}$$

If we expand this exponential in a power series indexed by k and if we set $K(u, \cdot) = W_t(\cdot)$ for (3), we see by observing (1) that the kernels for W are

$$g_k = \frac{1}{r^k} u \otimes u \otimes \cdots \otimes u = \frac{1}{r^k} u \otimes^k u\,, \tag{7}$$

where $\otimes^k$ is the k-fold tensor product. In other words,

$$\langle V_t(\cdot), K(u, \cdot)\rangle_{F_r} = \sum_{k=0}^{\infty} \frac{r^k}{k!} \left\langle h_k, \left(\frac{1}{r^k} u \otimes^k u\right)\right\rangle_{L^2(I)} = V_t(u)\,. \tag{8}$$

That is, the scalar product in the Fock space of the Volterra kernel with the functional K reproduces the Volterra kernel.

The beauty of using this reproducing kernel is that it reduces the estimation problem of nonlinear dynamical systems to that of linear operators. The details are carried out in Zyla and de Figueiredo (1983) and next summarized here for use in our neural network.

We assume available m pairs of input–output test functions, $u^j(\cdot)$ and $y_j(\cdot)$ for $j = 1, \ldots, m$, with these functions (along with n derivatives of the output) being square integrable over I. We choose the m input functions to be linearly independent over I so that we have sufficient information to perform an identification. In preparation for the next section we note that these input–output function pairs serve like exemplars of artificial neural networks.

The desire is to identify a dynamical system charcterized by $V.(\cdot)$, such that

$$V.(u^j(\cdot)) = y_j(\cdot) \quad j = 1, \ldots, m\,, \tag{9a}$$

subject to V describing the "best" such system, this latter being represented mathematically by V having the smallest norm, that is,

$$\inf \|V.(\cdot)\|^2_{B_n^2} \quad \text{over all } V.(\cdot) \in B_n^2\,. \tag{9b}$$

The number n of derivatives of interest play an important mathematical role in solving this problem because this Bochner space norm is defined in terms of them by

$$\|V.(\cdot)\|^2_{B_n^2} = \sum_{i=0}^{n-1} a_i \int_{t\in I} \|V_t^{(i)}\|^2_{F_r}\, dt\,. \tag{10}$$

The problem is actually tackled by solving the equivalent problem

$$\min \|V_t^{(i)}\|_{F_r} \quad \text{over all } t \in I \text{ and all } V_t^{(i)} \in F_r\,, \tag{11a}$$

subject to

$$V_t^{(i)}(u^j(\cdot)) = y_j^{(i)}(t) \quad \text{for } i = 0, \ldots, n\,,\ j = 1, \ldots, m\,. \tag{11b}$$

The solution to the problem of eq. (11) is rather easily phrased. First we form the $m \times m$ Grammian matrix

$$G = [G_{ij}] =$$

$$\left[\exp\left(\frac{1}{r}\langle u^i(\cdot), u^j(\cdot)\rangle_{L^2(I)}\right)\right], \quad (12a)$$

where for completeness we recall, see (3b), that the L^2 scalar product of functions of one variable is just the (Lebesgue) integral over the specified interval of the scalar product entries, that is

$$\langle u^i(\cdot), u^j(\cdot)\rangle_{L^2(I)} = \int_{t \in I} u^i(t) u^j(t)\, dt\,. \quad (12b)$$

Note that G is nonsingular by virtue of the independence assumed for the input test functions. Forming the column m vector of test outputs

$$y_{\text{test}}(\cdot) = [y_j(\cdot)]\,, \quad (12c)$$

we obtain a column m vector of coefficients

$$c(t) = G^{-1} y_{\text{test}}(t) = [c_j(t)]\,, \quad (12d)$$

to place in the best estimation $V_t'(\cdot)$ of V_t. The key and end result is that this best estimate is given by eq. (20) in de Figueiredo and Dwyer (1980).

$$V_t'(\cdot) = \sum_{j=1}^{m} c_j(t) \exp\left(\frac{1}{r}\langle u^j(\cdot), \cdot\rangle_{L^2(I)}\right). \quad (13)$$

It is upon eq. (13) that we base our functional artificial neural network.

Functional Neural Network: Single-Input Single-Output Case

The functional neural network of interest results from the key functional composition described by eq. (13) explained as follows

Recall that one of the properties of neural networks is that they process all the components of the input vector in parallel. To visualize this effect in the present case, we may think of the components of the input vector $u \in L^2(I)$ to be its values $u(t_0 + (k + \frac{1}{2})\,\Delta t)$ on a mesh

$$\{t_0 + \tfrac{1}{2}\,\Delta t, t_0 + \tfrac{3}{2}\,\Delta t, \ldots, t_0 + (N - \tfrac{1}{2})\,\Delta t\}\,, \quad (14)$$

of the interval $I = (t_0, t_0 + N\,\Delta t)$.

Approximating the integral associated with the scalar product in (13) by a sum, and substituting (14) in it, yields

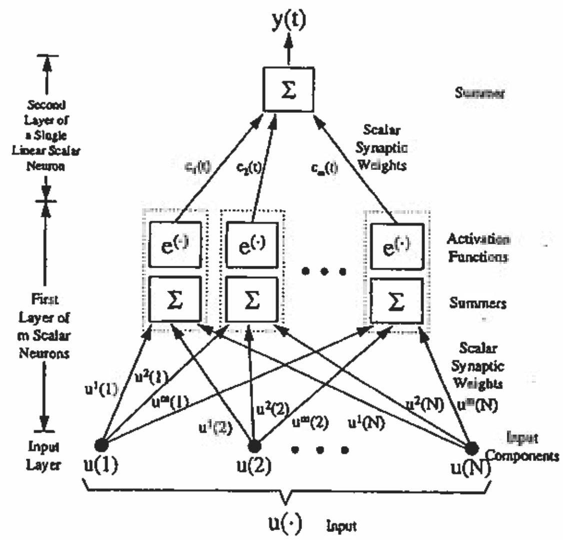

Figure 1. Discretized input FANN.

$$V_t'(\cdot) = \sum_{j=1}^{m} c_j(t) \cdot \exp\left[\frac{1}{r} \times \sum_{k=1}^{N} u^j(t_0 + \tfrac{1}{2}k\,\Delta t) u(t_0 + \tfrac{1}{2}k\,\Delta t)\,\Delta t\right], \quad (15)$$

which is illustrated diagrammatically in Figure 1, where, for simplicity in notation, we replace the argument $(t_0 + \frac{1}{2}k\,\Delta t)$ of u^j and u by k, i.e.,

$$V_t'(\cdot) = \sum_{j=1}^{m} c_j(t) \cdot \exp\left[\frac{1}{r} \sum_{k=1}^{N} u^j(k) u(k)\,\Delta t\right].$$

This figure portrays a two-hidden-layer neural network, where the synaptic weight associated with the connection from $u(k)$ and the jth neuron of the first layer representing the value $u^j(k)$ of the jth exemplary input. The second hidden layer consists of a single linear neuron, with synaptic weights $c_j(t)$, $j = 1, \ldots, m$, and output equal to $y(t)$.

We may now represent the net of Figure 1 compactly by letting $\Delta t \to 0$ and $k \to \infty$ in (15) and replacing sums by integrals. This leads us back to the representation of eq. (13) and the corresponding functional neural networks illustrated by the block diagram of Figure 2, where "functional" multiplication of the inputs by synaptic weights is interpreted as pointwise multiplication and "functional" summation of weighted inputs as an integral.

In other words, given an arbitrary input (of the class allowed by the system) this neutral network gives an output that is an approximation to the

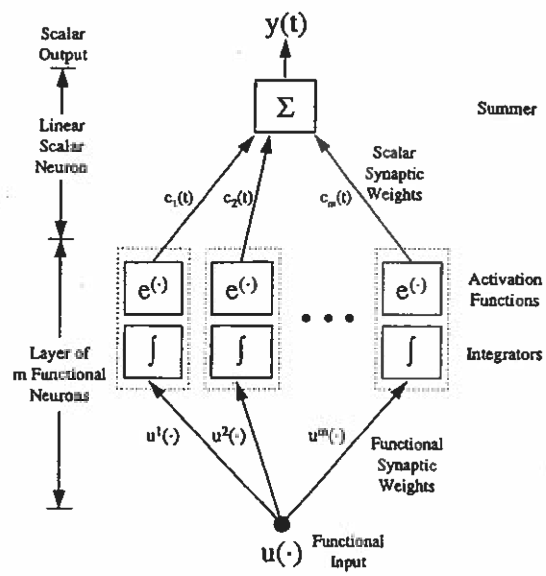

Figure 2. Basic FANN.

output of the true system that the neural network is approximating or simulating. This approximation is based upon forcing the neural network to give the desired output on the exemplary inputs from which the system is approximated by the neural network. The approximation is in terms of functionals and as a consequence attempts to incorporate the nonlinearities and dynamics of the system that is being approximated. The key ideas are best illustrated by a simple example.

EXAMPLE. Letting $1(\cdot)$ denote the unit step function, consider a system for which $u^1(t)=e^{-t}1(t)$ gives $y_1(t)=0.5(1-e^{-2t})1(t)$ and $u^2(t)=0.5e^{-t/2}1(t)$ yields $y_2(t)=0.25(1-e^{-t})1(t)$, all defined for time in the unit interval $I=[0,1]$. We choose $m=2$ and find

$$\|u^1\|^2_{L^2(I)}=\int_0^1 e^{-2t}\,dt=(1-e^{-2})/2\,,$$

$$\|u^2\|^2_{L^2(I)}=\int_0^1 0.25\,e^{-t}\,dt=(1-e^{-1})/4\,,$$

$$\langle u^1,u^2\rangle_{L^2(I)}=\langle u^2,u^1\rangle_{L^2(I)}=\int_0^1 0.5\,e^{-1.5t}\,dt=(1-e^{-1.5})/3\,,$$

from which G is calculated according to eq. (12a) after choosing $r=0.9$ and writing to three decimal places although carrying the calculation to eight,

$$G=\begin{bmatrix}1.617 & 1.333\\ 1.333 & 1.192\end{bmatrix},\quad G^{-1}=\begin{bmatrix}7.997 & -8.950\\ -8.950 & 10.847\end{bmatrix}.$$

In turn the synaptic weights $c(t)=G^{-1}y(t)$ are given as

$$c(t)=\begin{bmatrix}(1.761+2.238\,e^{-t}-3.999\,e^{-2t})1(t)\\ (-1.763-2.711\,e^{-t}+4.475\,e^{-2t})1(t)\end{bmatrix}.$$

For the two input neurons we calculate the exponentials of eq. (13) to be

$$\exp\left(\frac{1}{r}\langle u^1,u\rangle_{L^2(I)}\right)=\exp\left(1.111\int_0^1 e^{-t}u(t)\,dt\right),$$

$$\exp\left(\frac{1}{r}\langle u^2,u\rangle_{L^2(I)}\right)=\exp\left(0.556\int_0^1 e^{-2t}u(t)\,dt\right).$$

We note that these terms, although nonlinear in the input, are independent of time, all of the time dependence having now been placed in $c(t)$. For reference we state that the original system for which this neural network is an approximation is a squaring device followed by an integrator, both with unity gain.

MULTIPLE-INPUT MULTIPLE-OUTPUT CASE

The preceding functional neural network can be generalized to the multiple-input multiple-output case, where the input $u(\cdot)$ consists of n functions, i.e.

$$u(\cdot)=\mathrm{col}[u_1(\cdot),\ldots,u_n(\cdot)]\,, \qquad (16)$$

which are mapped by the network into M values

$$y(t)=\mathrm{col}[y_1(t),\ldots,y_M(t)]\,. \qquad (17)$$

This generalization is achieved by letting the input u belong to the space $L^2_n(I)$ of n-tuples of functions in $L^2(I)$ with a scalar product between any u and v in $L^2_n(I)$ defined by

$$\langle u,v\rangle=\sum_{k=1}^{n}\langle u_k,v_k\rangle_{L^2(I)}\,. \qquad (18)$$

Following a reasoning analogous to the one before with $L^2(I)$ replaced by $L^2_n(I)$, we arrive at the following generalization of eq. (13) for the nonlinear functional that sends the input u to the value $y_i(t)$ of the ith output at time t. In this formula,

$$u^q(t) = \text{col}[u_1^q(t), \ldots, u_n^q(t)], \quad q = 1, \ldots, m, \tag{19}$$

are the m exemplary test inputs that give rise to the m examplary test outputs

$$y^q(t) = \text{col}[y_1^q(t), \ldots, y_M^q(t)], \quad q = 1, \ldots, m. \tag{20}$$

Thus eq. (13) takes the form

$$y_i(t) = V'_{i,t}(u) = \sum_{j=1}^{m} c_j(t) \exp\left(\frac{1}{r}\langle u^j(\cdot), u(\cdot)\rangle_{L_n^2(t)}\right), \quad i = 1, \ldots, M, \tag{21}$$

where

$$c_{i\cdot}(t) = \text{col}[c_{i1}(t), \ldots, c_{im}(t)] = G^{-1} y_{i,\text{test}}(t), \tag{22}$$

where

$$G = [G_{ij}] = \left[\exp\left(\frac{1}{r}\langle u^i(\cdot), u^j(\cdot)\rangle_{L_n^2(t)}\right)\right], \tag{23}$$

and

$$y_{i,\text{test}}(t) = \text{col}[y^1_{i,\text{test}}(t), \ldots, y^m_{i,\text{test}}(t)]. \tag{24}$$

That is, $y_{i,\text{test}}(t)$ is the m-tuple of the ith component of the test outputs $y_i^1(t), \ldots, y_i^m(t)$.

A block diagram of the functional network represented by eq. (21) is shown in Figure 3. The network consists of two layers. The first layer consists of functional neurons with functional synaptic weights given by the appropriate components of the exemplary inputs. After going through an exponential activation function, these outputs of the firts layer go through a second layer of linear neurons after being weighted with scalar synaptic weights $c_{ij}(t), i = 1, \ldots, M, j = 1, \ldots, m$.

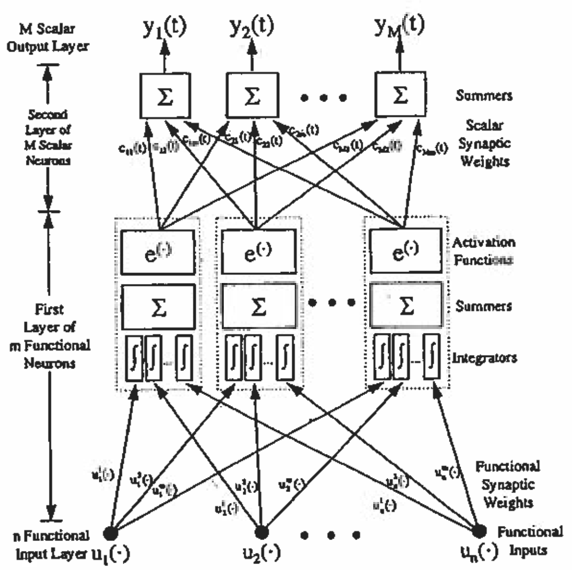

Figure 3. MIMO FANN.

Application in Nonlinear Control

In addition to conventional applications in which neural networks are used for detection and classification of events in data, the functional neural network presented here may be applied to model dynamical systems representing plants in nonlinear feedback control systems. One such application is illustrated in Figure 4 where, in a self-tuned regulator system, a functional neural network is used to model a plant, the parameters of which are captured using the input–output data of the nonlinear plant being regulated by the system. The model is then used to design and adjust the parameters of the regulator controlling the plant. This relates to the type of models discussed by Sanchez (1994). Other types of configurations can be considered, depending on specific control applications (de Figueiredo and Chen, 1993; de Figueiredo, 1994).

Discussion

Because of the inherent importance of using neural networks for predicting the performance

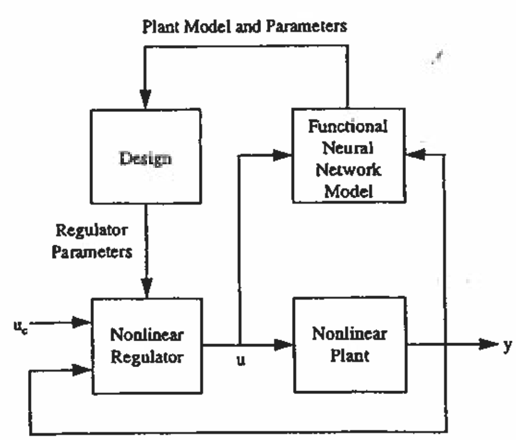

Figure 4. FANN in self-tuned regulator.

of nonlinear dynamical systems, here we generalized the theory of artificial neural networks to incorporate that of functional input–output pairing of nonlinear dynamical systems. The theory is based upon the theory of Volterra kernels but using a very important difference of viewpoint than one finds in much of the literature on Volterra series outside those related to Zyla and de Figueiredo (1983). The key idea is the reproducing kernel within the mathematical framework of Fock Hilbert space concepts, although the reader should not let the technical details of the mathematical spaces involved get in the way of the fundamental ideas. The use of the reproducing kernel allows the estimation of nonlinear systems to revert to that of linear dynamical systems while still incorporating the nonlinearities for which the Volterra series is tailored. The theory was previously developed and expounded in Zyla and de Figueiredo (1983) and adapted here to fit within the framework of neural network theory. The generalization to the case of a multilayer functional network, consisting of a cascade of networks of the type described here, can be made as in de Figueiredo (1992).

References

de Figueiredo RJP (1990): "An optimal matching score net for pattern classification." In *Proceedings of the 1990 International Joint Conference on Neural Networks (IJCNN)*, IEEE, San Diego, CA, June 17–21, 1990. Vol. 2, pp. 909–916.

de Figueiredo RJP (1992): "An optimal multilayer neural interpolating (OMNI) net in a generalized Fock space setting." In *Proceedings of the International Joint Conference on Neural Networks*, IEEE, Baltimore, MD, June 7–11, 1992. pp. I-111–120.

de Figueiredo RJP (1994): "Neural modeling and identification of nonlinear systems in an abstract space setting." In R.S. Gutalu (ed.): *Mechanics and Control*, Vol. V. New York: Plenum Press, pp. 309–317.

de Figueiredo RJP, Chen G (1993): *Nonlinear Feedback Control Systems: An Operator Theory Approach.* Cambridge, MA: Academic Press.

de Figueiredo RJP, Dwyer TAW III (1980): "A best approximation framework and implementation for simulation of large-scale nonlinear systems." *IEEE Trans. Circuits Syst.* **CAS-27**: 1005–1014.

Holtzman JN (1970): *Nonlinear System Theory: A Functional Analysis Approach.* Englewood Cliffs, NJ: Prentice–Hall.

Sanchez EN (1994): "Dynamic neural networks for nonlinear systems identification." In *Proceedings of the 33rd IEEE Conference on Decision and Control*, Lake Buena Vista, FL, December 14–16, 1994. New York: IEEE, pp. 2480–2481.

Zyla LV, de Figueiredo RJP (1983): "Nonlinear system identification based on a Fock space framework." *SIAM J. Control Optimization* **21**: 931–939.

Received December 15, 1995
Accepted February 15, 1996