Identification and Control of Nonlinear Systems Using Haar Wavelet Networks

Sokho Chang, Seok Won Lee, and Boo Hee Nam

Abstract: In this paper, Haar wavelet-based neural network is described for the identification and control of discrete-time nonlinear dynamical systems. Wavelets are suited to depict functions with local nonlinearities and fast variations because of their intrinsic properties of finite support and self-similarity. Due to the orthonormal properties of Haar wavelet functions, wavelet neural networks result in a greatly simplified training problem. This wavelet-based scheme performs adaptively both the identification of nonlinear functions and the control of the overall system, while the multilayer neural network is applied to the control system just after its sufficient learning of the unknown functions. Simulation shows that the wavelet network can be a good alternative to a multilayer neural network with backpropagation.

Keywords: wavelet network, neural network

I. Introduction

Recently, wavelets have been applied successfully to multiscale time-frequency analysis and synthesis in signal processing[1], function approximation[2]-[5], and fault detection and monitoring[6][7]. Wavelets are suited to depict functions with local nonlinearities and fast variations because of their intrinsic properties of finite support and self- similarity.

The goal of most modern wavelet research[8] is to create a set of basis functions and transforms that will give an informative, efficient, and useful description of a function or signal. If the signal is represented as a function of time, wavelets provide efficient localization in both time and frequency or scale. Another central idea is that of multiresolution where the decomposition of a signal is in terms of the resolution of detail.

In [9] and [10], they use Gaussian radial basis functions for system identification and control. In [5], they use also a nonorthogonal wavelet function for identification, and in [4] they use Haar basis functions for nonlinear system control, where they are dealing a multidimensional system with a 1-dimensional system by using the weighted sum of multivariables. In [2], they use a nonorthogonal wavelet function by using a weighted sum of sigmoid functions for function approximation. For function learning, we follow a procedure in [3], where they use the Lemarie-Meyer wavelet.

In this paper we use Haar wavelet functions for identification and control of discrete-time nonlinear dynamical systems as in [10], where multilayer neural networks with backpropagation are used. In section 2, we give a brief review of wavelet theory, In section 3, a wavelet network for function estimation is designed. And we present the simulation results in section 4, followed by a conclusion.

Manuscript received: April 4, 2000., Accepted: Aug. 17, 2000
Sokho Chang: DongSung Infocomm
Seok Won Lee: Dept of Control Engineering, Kangwon National University
Boo Hee Nam: BK21 Dept of Electrical and Computer Engineering, Kangwon National University

II. Haar wavelets

A wavelet is a small wave, which has its energy concentrated in time to give a tool for the analysis of transient, nonstationary, or time-varying phenomena. It still has the oscillating wave-like characteristic but also has the ability to allow simultaneous time and frequency analysis, having its finite energy concentrated around a point.

Wavelets are adjustable and adaptable. Because there is not just one wavelet, they can be designed to fit individual applications. They are ideal for adaptive systems that adjust themselves to suit the signal.

The Haar scaling function is defined as

$$\phi(x) = \begin{cases} 1 & \text{for } 0<x<1 \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

and the wavelet function is defined as

$$\psi(x) = \begin{cases} 1 & \text{for } 0<x<\frac{1}{2} \\ -1 & \text{for } \frac{1}{2}<x<1 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The wavelet functions, by dilations and translations, form an orthonormal basis of $L^2(R)$, the space of all square integrable functions on R. Specifically, there exists a function $\psi(t)$ such that

$$\psi_{m,n}(t) = 2^{\frac{m}{2}}\psi(2^m t - n) \tag{3}$$

form an orthonormal basis of $L^2(R)$. Therefore, the wavelet basis induces an orthogonal decomposition of $L^2(R)$. Let W_m be a subspace spanned by $\{2^{\frac{m}{2}}\psi(2^m t - n)_{n=-\infty,\infty}\}$ and V_m be the subspace spanned by $\phi_{m,n}(t) = 2^{\frac{m}{2}}\phi(2^m t - n)$, then we have some properties of multiresolution analysis of $L^2(R)$:

$$L^2(R) = \bigoplus_m W_m \tag{4}$$

$$W_i \perp V_i \tag{5}$$

$$V_{i+1} = V_i \oplus W_i \tag{6}$$

$$\cdots \subset V_i \subset \cdots \subset V_{i+1} \subset \cdots \subset L^2(R) \tag{7}$$
with $V_\infty = \emptyset$ and $\bigcup V_i = \emptyset$

$$L^2(R) = V_i \oplus \bigcap_{k=i}^{k=\infty} W_k \quad \text{for all } i \tag{8}$$

$$f(t) \in V_i \Leftrightarrow f(2t) \in V_{i+1} . \tag{9}$$

We will use the fact that any $f(x) \in L^2(R)$ can be approximated arbitrarily closely in V_M for some integer M. That is, for any $\varepsilon > 0$, there exists an M sufficiently large such that

$$\| f(x) - \sum_n \langle f(x), \phi_{Mn}(x) \rangle \phi_{Mn}(x) \| < \varepsilon . \tag{10}$$

In the 2-dimensional analysis and synthesis, we need one scaling function and three wavelet functions:

scaling function $a(x, y) = \phi(x)\phi(y)$ for approximation
wavelet function $h(x, y) = \phi(x)\psi(y)$ for horizontal detail
wavelet function $v(x, y) = \psi(x)\phi(y)$ for vertical detail
wavelet function $d(x, y) = \psi(x)\psi(y)$ for diagonal detail

Here, $\phi(x)$ and $\psi(x)$ are the 1-dimensional scaling function and wavelet function, respectively, defined as before. We will use only the scaling function for the identification of two dimensional systems as in the one dimensional case.

III. Wavelet networks for identification of nonlinear functions

The function $f(x) \in L^2(R)$ can be estimated with a set of given training data $T_N = \{(x_i, f(x_i))\}$, $i = 1, \cdots, N$. For a sufficiently large M, we have an estimate:

$$\begin{aligned} f(x) &\approx \sum_k \langle f(x), \phi_{Mk}(x) \rangle \phi_{Mk}(x) \\ &= \sum_k c_k \phi_{Mk}(x), \end{aligned} \tag{11}$$

where $\phi_{Mk}(x) = 2^{\frac{M}{2}} \phi(2^M x - k)$. Hence, the approximation can be implemented by a three-layer network similar to the multilayer neural networks. The input layer has one node with input x. The hidden layer contains nodes indexed by k, and the weights and nonlinearities of the hidden-layer nodes are identical, i.e., 2^M and $\phi(x)$, respectively, and the threshold for the k-th node of the hidden layer is k. The output layer is one linear node, and the weights c_k's of the node are to be found. Since we are interested in the functions of finite support, we assume that $f(x)$ is supported in [-1, 1] and that the hidden layer contains a finite number of nodes, with the indices of the hidden nodes running from $-K$ to K for some positive integer K. So, we have

$$f(x) \approx g(x) = \sum_{k=-K}^{K} c_k \phi_{Mk}(x). \tag{12}$$

First we determine the number of hidden nodes. In the 1-D case, for a given resolution M and thus a subinterval of length 2^{-M}, we have $2(2^M + 1)$ uniform subinterval in the finite support $[-1-2^{-M}, \ 1+2^M]$, with two more subintervals beyond the interval [-1 1] to provide some safeguard.
(Similarly, for 2-dimensional functions, the number of hidden layer nodes needed is $2^2(2^M + 1)^2$.)
Let the i-thsub-interval be

$$I_i = [x_i - 2^{-(M+1)}, \quad x_i + 2^{-(M+1)}]. \tag{13}$$

Let the training data be $X^T = [x_1 \ x_2 \cdots x_n]$ and $Y^T = [y_1 \ y_2 \cdots y_n]$ with $y_i = f(x_i)$, $i = 1, 2, \ldots, n$ and $n = 2(2^M + 1)$, where x_i is the center point of the i-th sub-interval in the hidden layer and is expressed as

$$x_k = 2^{-M}(0.5 + k), \quad k = -(2^M + 1), \cdots, 2^M. \tag{14}$$

If we let

$$\begin{aligned} \psi^i(x) &\equiv \phi_{Mk}(x) = 2^{\frac{M}{2}} \phi(2^M x - k) \\ &= [0, \cdots, 0, \phi_{Mk}(x_i), 0, \cdots, 0]^T, \end{aligned} \tag{15}$$

then the coefficients c_i's are calculated by

$$f(x_i) = c_i \phi_{Mk}(x_i) = 2^{\frac{M}{2}} c_i . \tag{16}$$

Or, if we let $\Phi(x) = [\psi^1(x), \cdots, \psi^n(x)]$ and $C^T = [c_1, c_2, \cdots, c_n]$, then c_i's are also calculated by $C = \Phi^{-1}(x) Y$. So we have an estimate of the function, $g(x) = \Phi^T(x) C$. Since these sampling points x_i's are the nominal points, when we have a time varying point $x_i(t)$ at time t, then by gradient descent method, $c_i(t)$ can be updated by

$$C(t+1) = C(t) + \eta \Phi(x(t)) [f(x(t)) - \Phi^T(x(t)) C(t)] \quad \text{for some } \eta, \quad 0 < \eta < 1 , \tag{17}$$

or, simply,

$$c_i(t+1) = c_i(t) + \eta 2^{\frac{M}{2}} [f(x_i(t)) - 2^{\frac{M}{2}} c_i(t)]. \tag{18}$$

For example, let's approximate the function expressed as

$$y = \frac{x}{1 + x^2} \tag{19}$$

whose graph is shown in Fig. 1, what we do first is to choose the basis functions for the approximation of this nonlinear function. For the following affine functions with dilation m and translation n,

$$\phi_{Mn}(x) = 2^{\frac{M}{2}} \phi(2^M x - n) \text{ with } \phi_{M0}(x) \equiv \phi(x), \tag{20}$$

we can have a set of basis function with an appropriate choice of m and n values, where m stands for tiling of the frequencies of the function and n stands for tiling of the time of the function. For simplicity in this example, we let $M = 1$. Then, we get six basis functions, which are translations of the mother wavelet $2^{\frac{M}{2}}\psi(2^M x)$, shown in Fig. 2.

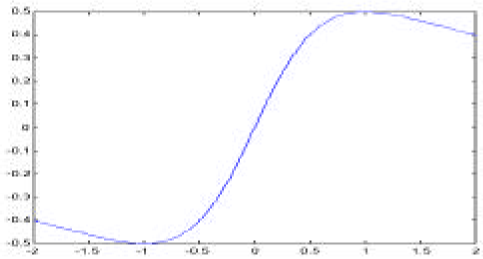

Fig. 1. The graph of an example $y = \frac{x}{1+x^2}$.

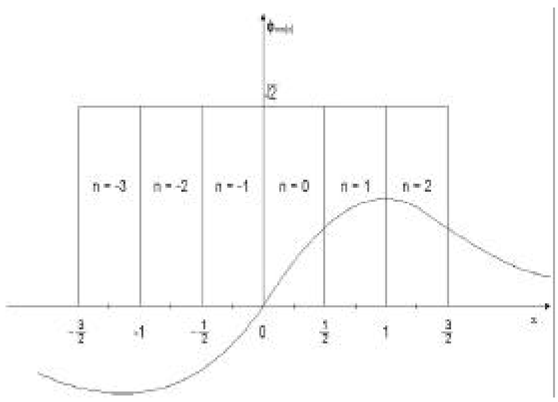

Fig. 2. Six basis function for approximation of the function $y = \frac{x}{1+x^2}$.

We could have other set of basis function with several values of M and their translations. In the appendix of this paper, the simulation program lists are given with $M=3$ for a smoother approximation of the nonlinear function than with $M=1$.

IV. Simulations

The examples in this section are from [11]. We simulate them by using both MNN(multilayer neural networks with backpropagation) and WNN(wavelet neural network) for a comparison. The simulations are carried out in the Pentium computer using MATLAB.

Example 1: We consider here the problem of controlling the plant which is described by the difference equation

$$y_p(k+1) = f[y_p(k)] + u(k) = \frac{y_p(k)}{y_p^2(k)+1} + u(k) \quad (21)$$

where the function $f(\cdot)$ is assumed to be unknown. A reference model is described by the first-order difference equation

$$y_m(k+1) = 0.5y_m(k) + r(k) \quad (22)$$

where $r(k) = 0.3\sin(2\pi\frac{k}{25}) + 0.3\sin(2\pi\frac{k}{10})$. The nonlinear function is estimated adaptively (on-line) as $W_f[y_p(k)]$. The control input to the plant at any instant k is computed using $W_f[y_p(k)]$ in place of $f(y_p(k))$ as $u(k) = -W_f[y_p(k)] + 0.5y_p(k) + r(k)$. This results in the nonlinear difference equation

$$y_p(k+1) = f[y_p(k)] - W_f[y_p(k)] + 0.5y_p(k) + r(k)$$

governing the behavior of the plant. The overall scheme of the system is shown in Fig. 3. In this case, the identification of the nonlinear function $f(\cdot)$ and the control of the system are simultaneously obtained, while the control system using a backpropagation neural network can be

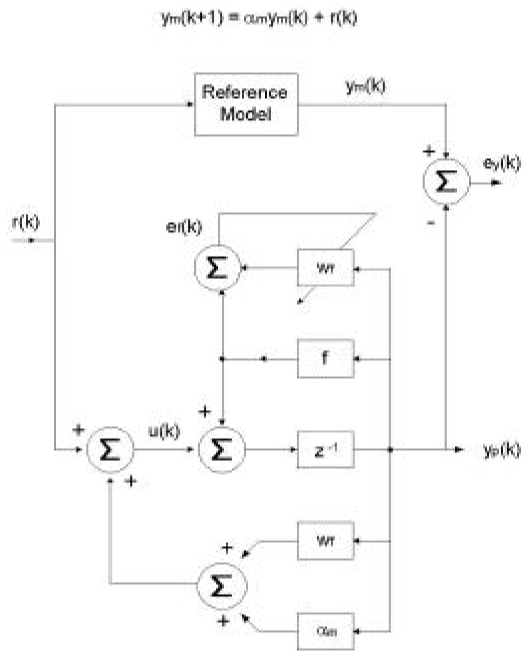

Fig. 3. The overall scheme of the controlled system with a wavelet neural network W_f.

effectively performed after the neural network N_f finishes the sufficient learning of the nonlinear function $f(\cdot)$ shown in Fig. 4.

The response of the controlled system is shown in Fig. 5. The sum-squared error between y_p and y_m is $4.9117x10^{-4}$ for WNN with $M=7$, and $1.9443\ x\ 10^{-4}$ for MNN, respectively, and the elapsed time is 0.22 for WNN and 0.49 for MNN, respectively. In the MNN, the unknown plant was identified off-line by the Levenberg-Marquardt method, And after the MNN for the nonlinear function was fully trained, the MNN was applied to the control system, while in WNN the identification was adaptively done in the overall control system. In both cases, the setup times in the program were excluded and only the execution times in the control loops were compared. In Fig. 5, the two graphs for y_p and y_m appear almost overlapped.

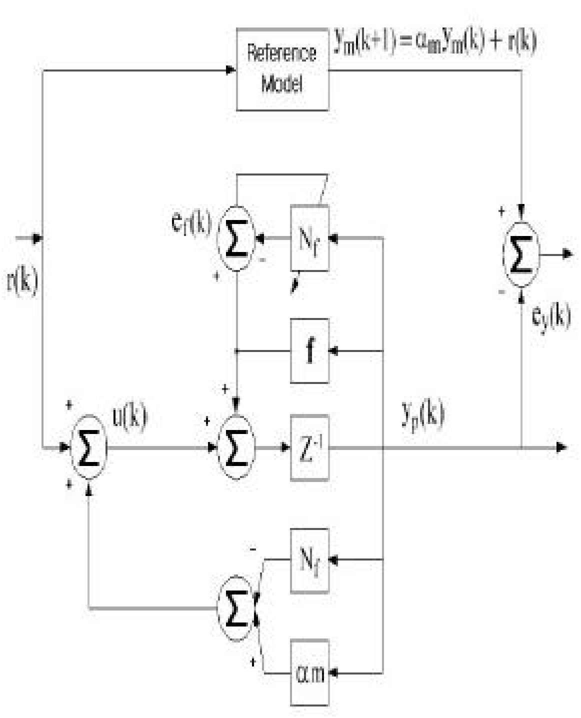

Fig. 4. The overall scheme of the controlled system with a backpropagation neural network N_f.

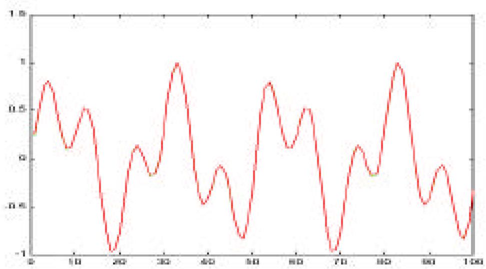

Fig. 5. The outputs of the target system and plant in Example 1.

Example 2: The plant we deal with is

$$y_p(k+1) = f[y_p(k), y_p(k-1)] + u(k) \tag{23}$$

where

$$f[y_p(k), y_p(k-1)] = \frac{a y_p(k) y_p(k-1)[a y_p(k) + 2.5]}{1 + a^2 y_p(k)^2 + a^2 y_p^2(k-1)},$$
$$a = 4,$$

and $f[y_p(k), y_p(k-1)]$ is adaptively estimated as $W_f[y_p(k), y_p(k-1)]$. The reference model is

$$y_m(k+1) = 0.6 y_m(k) + 0.2 y_m(k-1) + r(k) \tag{24}$$

and the reference input $r(k)$ is an alternating rectangular wave with an amplitude 0.2. In this case the control input is:

$$u(k) = -W_f[y_p(k), y_p(k-1)] + 0.6 y_p(k) + 0.2 y_p(k-1) + r(k). \tag{25}$$

The overall scheme of the system is shown in Fig. 6. In this case, the identification of the nonlinear function $f(\cdot)$ and the control of the system are simultaneously obtained, while the control system using a backpropagation neural network can be effectively performed after the neural network N_f finishes the sufficient learning of the nonlinear function $f(\cdot)$ shown in Fig. 7. The response of the controlled system is shown in Fig. 8. The sum-squared error between y_p and y_m is 0.001 for WNN and 0.0313 for MNN, respectively, and the elapsed time is 0.27 for WNN and 0.60 for MNN, respectively. In both examples, the graphs of the target system and the plant are seen to be almost overlapped. Learning a 2-dimensional function becomes considerably difficult, since the available training data is generally sparse and do not fill out the domain of the function. But for many applications, good results can be obtained by observing that the training data needed are concentrated in a small number of regions in the overall domain $[-1, 1]^2$. In this example, the data interested are concentrated on and around the diagonal in the plane. The 2-dimensional scaling function used is

$$a(x, y, j, k) = 2^M \phi(2^M x - j)\phi(2^M y - k). \tag{26}$$

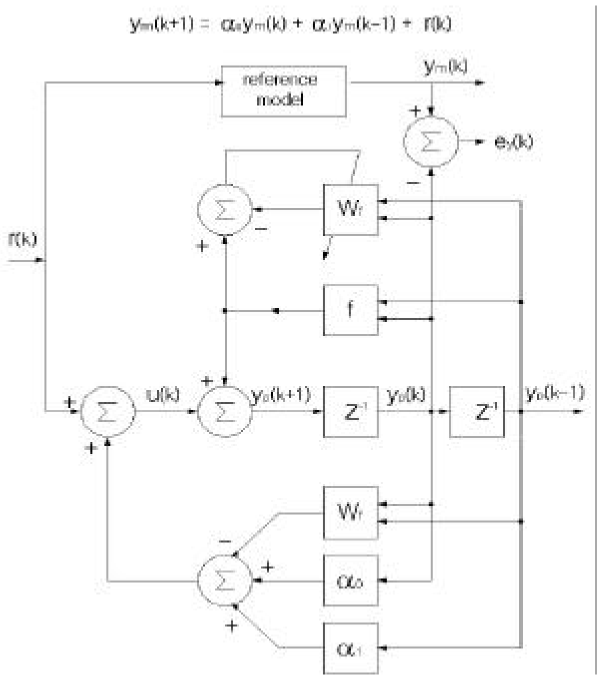

Fig. 6. The overall scheme of the controlled system with a wavelet neural network W_f for Example 2.

V. Conclusions

In this paper we implement the wavelet network identifier for nonlinear function using the Haar wavelet basis functions. Due to the orthonormal properties of Haar wavelet functions, wavelet networks result in a greatly simplified training problem. These wavelet networks are used for the adaptive control of unknown nonlinear dynamical systems. Simulation shows that the wavelet

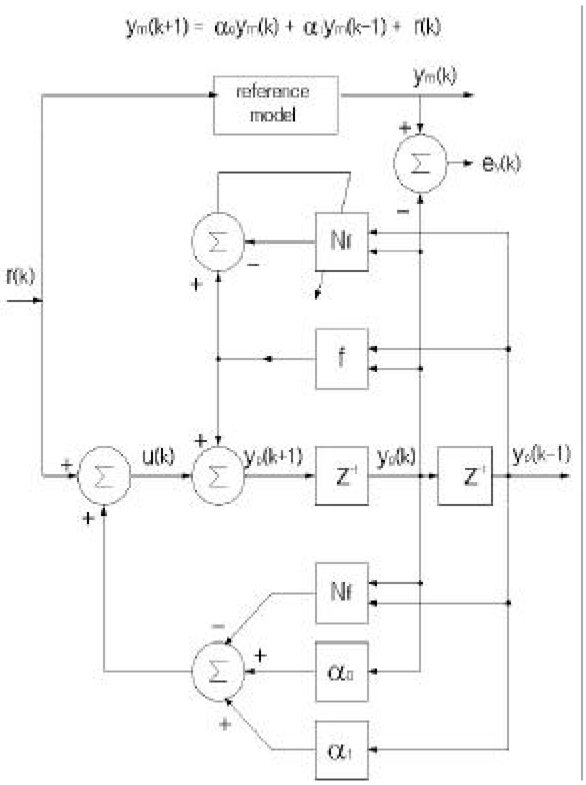

Fig. 7. The overall scheme of the system with a backpropagation neural network N_f for Example 2.

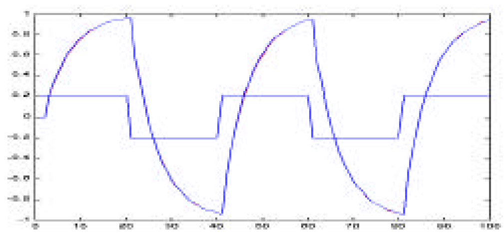

Fig. 8. The outputs of the target system and plant in Example 2.

network can be a good alternative to a multilayer neural network.

References

[1] MathWorks Inc., MATLAB Wavelet Toolbox.

[2] Y. C. Pati and P. S. Krishnaprasad, "Analysis and synthesis of feedforward neural networks using discrete affine wavelet transforms," *IEEE Trans. NN*, vol. 4, no. 1, pp. 73-85, 1993. 1991.

[3] J. Zhang, G. G. Walter, Y. Miao, and W. N. W. Lee, "Wavelet neural networks for function learning," IEEE *Trans. Signal Processing*, vol. 43, no. 6, 1995.

[4] B. S. Chen and Y. M. Cheng, "Adaptive wavelet network control design for nonlinear systems," *Proceedings of the 35th Conference on Decision and Control*, pp. 3224-3229, 1996.

[5] L. F. Yeung and X. W. Li, "Multi-input system identification and its applications using wavelet constructive method," *Proceedings of the 35th Conference on Decision and Control*, pp. 3230-3235, 1996.

[6] C. M. Chen and K. A. Loparo, "Electric fault detection for vector-controlled induction motors using the discrete wavelet transform," *Proceedings of the American Control Conference*, pp. 3297-3301, June, 1998.

[7] H. Du and S. S. Nair, "Identification of friction at low velocities using wavelet basis function network," *Proceedings of the American Control Conference*, pp. 1918-1922, June, 1998.

[8] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms*, Prentice Hall, 1998.

[9] R. M. Sanner and J. J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, Nov., 1992.

[10] M. Cannon and J. J. E. Slotine, "Space-frequency localization networks for nonlinear system estimation and control," *Neurocomputing*, vol. 9, no. 3, 1995.

[11] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. NN*, vol. 1, no. 1, 1990.

Appendix

CEMTOOL Program List for the approximation of the nonlinear function $y = \frac{x}{1+x^2}$

```
EX1.cem
clear
M=3; //M=7
t=[];
for(k=-(2^M+1);k<=2^M;k=k+1)
    t=[t; 2^(-M)*(0.5+k)];
y=t./(1+t.^2);
eta=0.01;
C=zeros(2*(2^M+1), 1);
basis=[];
for(n=-(2^M+1);n<=2^M;n=n+1)
    basis=[basis;Hphi(M,n,t)'];
f=basis'*C;
sse=sum(abs(sqrt(y-f)).^4);
while (sse > 10^(-8) ){
    error=y-f;
    delC=eta*basis*error;
    C=C+delC;
    f=basis'*C;
    sse = sum(abs(sqrt(y.-f)).^4); }
f=basis'*C;
plot(t,[f,y]);
sse = sum(abs(sqrt(f-y)).^4)

EX2.cem
clear
a=4;
y(1)=1;
```

```
y(2)=-1;
for(k=2;k<=100;k=k+1){
y(k+1)=a*y(k)*y(k-1)*(a*y(k)+2.5)/(1+y(k)^2
          +a^2*y(k-1)^2);
plot(k, [y(k), y(k-1)]);      }

Hphi.cem
// Haar affine scaling functions
// m: dilation    n: translation function
function;
y <> m,n,t
y=2^(m/2)*Hsc(2^m*t-n);
Hpsi.cem
// Haar affine wavelet functions
function;
y <> m,n,t
y=2^(m/2)*Hwv(2^m*t-n);
Hsc.cem
// Haar scaling function
function;
y <> t
y=[];
for(i=1;i<=length(t);i=i+1){
    if(t(i) >= 0 && t(i) < 1)
        y(i)=1;
    else
        y(i)=0;   }
[m,n]=size(t);
[o,p]=size(y);
if(m != o)
   y=y';
Hwv.cem
// Haar wavelet function
function;
y <> t
y=[];
for(i=1;i<=length(t);i=i+1){
    if (t(i) >= 0 && t(i) < 0.5)
        y(i)=1;
    else{
        if (t(i) >=0.5 && t(i) < 1)
            y(i)=-1;
        else y(i)=0;
}
}
```

Sokho Chang
He was born in Korea in 1969. He received the B. S. in control and instrumentation engineering, and M. S. and Ph.D. degrees in electronics engineering from Kangwon National University, in 1993 and 1998, repectively. From 1998 to 2000, he did his Post-Doctoral work at the advanced robotics research center, KIST(Korea Institute of Science and Technology). Since March 2000, he has been with the institute of systems engineering, DongSung Inforcomm. His research interests include human-computer interaction, telerobotics, and communication.

Boo Hee Nam
He was born in 1950. He received the B. S. and M. S. degrees from Seoul National University, in 1974 and in 1976, respectively, and the Ph. D. degree from Iowa State University, Ames, Iowa, in 1983, all in electrical engineering. He was the recipient of the 1984 Guillemin-Cauer Prize Paper Award of the IEEE Circuits and Systems Society (co-recipient with A. N. Michel and R. K. Miller) with the paper “Stability Analysis of Interconnected Systems Using Computer Generated Lyapunov Functions”. Prof. Nam has been with the Department of Electrical and Computer Engineering at Kangwon National University, Chunchon, Korea, since 1983. His research interests include nonlinear systems analysis, neuro-fuzzy networks, wavelet transform applications, and communication networks for automation.

Seok Won Lee
He was born in 1969. He received the B. S. and M. S. degrees in control and instrumentation engineering from Kangwon National University, in 1995 and in 1997, respectively. He is currently a Ph.D. candidate at Kangwon National University. His research interests include soft modem, wavelet transform applications such as fingerprint recognition, MPEG audio, TCP/IP and mobile robot.