A NEURAL DEMODULATOR FOR AMPLITUDE SHIFT KEYING SIGNALS

Kenji NAKAYAMA **Kunihiko IMAI**

Dept. of Electrical and Computer Eng.
Faculty of Tech., Kanazawa Univ.
2-40-20, Kodatsuno, Kanazawa, 920 JAPAN
E-mail: nakayama@haspnn1.ec.t.kanazawa-u.ac.jp

ABSTRACT: A neural demodulator is proposed for amplitude shift keying (ASK) signal. It has several important features compared with conventional linear methods. First, necessary functions for ASK demodulation, including wide-band noise rejection, pule waveform shaping, and decoding, can be embodied in a single neural network. This means these functions are not separately designed but unified in a learning and organizing process. Second, these functions can be self-organized through the learning. Supervised learning algorithms, such as the backpropagation algorithm, can be applied for this purpose. Finally, both wide-band noise rejection and a very sharp waveform response can be simultaneously achieved. It is very difficult to be done by linear filtering. Computer simulation demonstrates efficiency of the proposed method.

I INTRODUCTION

Neural networks (NNs) have been effectively applied to signal processing and pattern recognition [1]-[5]. Features of NNs include self-organization, learning, nonlinear functions, and parallel implementation. How to utilize these features in each application is an important point.

Communication is also an interesting application field of NNs. Some nonlinear distortion can be compensated for by using NNs [3].

In this paper, demodulation problems are dealt with. In the demodulation process, undesirable signals and noises are rejected through filters. The extracted signal is transformed into its original or another desired waveform. Noise rejection filters usually distort a time response. If the original waveform is very sharp, like a pulse waveform, this distortion becomes fatal error.

In this paper, a neural demodulator for amplitude shift keying (ASK) signals is proposed. A multilayer neural network and the backpropagation algorithm [6] are employed. The purpose of this model is to achieve both wide-band noise rejection and a very sharp waveform response, which are difficult to be done by linear filters. Trained network structure, internal representation and an optimum activation function are discussed. Simulation results are also shown in order to examine efficiency of the proposed method.

II STRUCTURE OF NEURAL DEMODULATOR

Figure 1 shows the proposed neural demodulator. The received signal x(n) includes the ASK signal $x_A(n)$ and noise e(n) as follows:

$$x(n) = x_A(n) + e(n) \quad (1)$$

0-7803-1901-X/94 $4.00 ©1994 IEEE

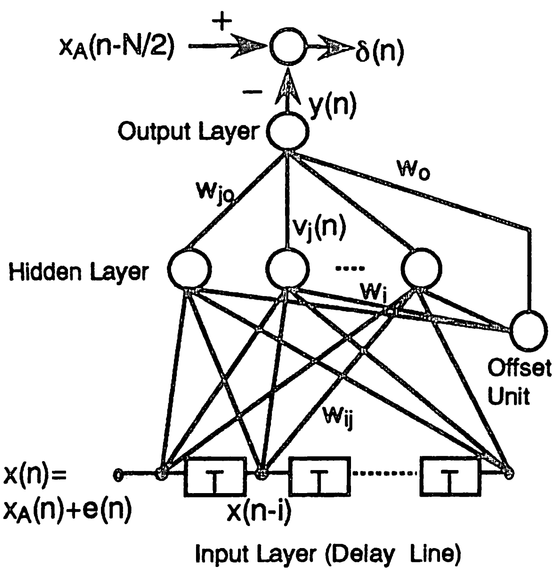

Fig.1 Proposed neural demodulator.

The input layer is composed of a delay line, including N-1 delay elements. T is a sampling period. The output of the ith delay element is denoted x(n-i). N samples of x(n), that is x(n-i), i=0 ~ N-1, are transmitted through connections in parallel. An offset unit is used, which always outputs 1 to the hidden units and to the output unit in order to adjust bias.

Let connection weights from x(n-i) to the jth hidden unit be w_{ij}, and from the jth hidden unit to the output unit be w_{jo}. Network equations are expressed as follows:

$$u_j(n) = \sum_{i=0}^{N-1} w_{ij} x(n-i) + w_j \quad (2)$$

$$v_j(n) = f_H(u_j(n)) \quad (3)$$

$$net(n) = \sum_{j=0}^{M-1} w_{jo} v_j(n) + w_o \quad (4)$$

$$y(n) = f_O(net(n)) \quad (5)$$

w_j and w_o are connection weights from the offset unit to the jth hidden unit and the output unit, respectively. $f_H(\cdot)$ and $f_O(\cdot)$ are activation functions of the hidden units and the output unit, respectively.

The original ASK signal $x_A(n)$ is used as target. $x_A(n)$ is closely related to its neighborhood samples, that is $x_A(n-i)$, $-N/2 \leqq i \leqq -1$ and $1 \leqq i \leqq N/2$. Furthermore, the output y(n) is calculated using x(n-i), $0 \leqq i \leqq N-1$, as shown in Eqs.(2)-(5). Therefore, $x_A(n-N/2)$ is used as the target for y(n). The output error is evaluated by

$$\delta(n) = x_A(n-N/2) - y(n) \quad (6)$$

It should be pointed out that the proposed NN does not separate functions required, rather all functions are unified in a single NN.

III LEARNING ALGORITHM AND ACTIVATION FUNCTIONS

3.1 Learning Algorithms

Backpropagation algorithm is very powerful for multilayer neural networks [6]. It is also employed in our model. An important point of the learning is to automatically design necessary functions, including wide noise rejection and pulse waveform regeneration. Because these requirements are difficult to be satisfied simultaneously by linear signal processing. This point will be further investigated in **Sec.IV**.

3.2 Activation Functions

Another important point of neural network design is to optimize activation functions for each application. However, this issue still remain as an open question.

In this paper, we propose a valley function $f_{val}()$ given by Eq.(7).

$$f_{val}(x) = \frac{x^2}{1 + x^2} \quad (7)$$

$$f_{sig}(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (8)$$

Figure 2 shows this valley function,

which is used in the hidden layer. In the output unit, the sigmoid function given by Eq.(8) is used. Furthermore, for comparison, it is also used in the hidden layer.

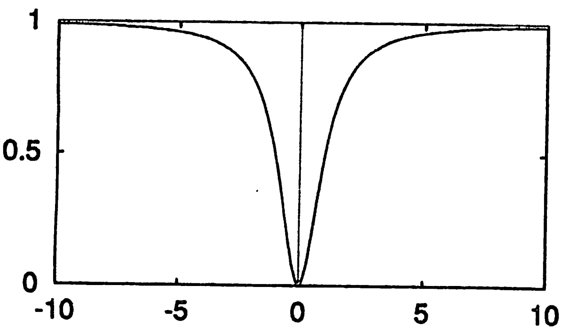

Fig.2 Valley activation function.

The purpose of using the valley function is to realize a full-wave rectifier. The input to the jth hidden unit $u_j(n)$ is a sum of products of the received signal samples x(n-i) and the connection weights w_{ij}. Thus, $u_j(n)$ can be filtered signal, in which the additional noise is removed. In other word, it can be a sinusoidal waveform. For this waveform, the valley function works as a full-wave rectifier.

However, the valley function is not exactly a full-wave rectifier. Because differential of the activation function is needed in the gradient algorithm. Furthermore, the noise can be completely rejected in $u_j(n)$. Finally, a single valley function cannot provide good performance as shown in **Sec. IV**. This means the hidden unit plays not only a rectifier but also pulse waveform generation. The latter requires fine phase adjusting among several hidden unit outputs. This can be done automatically.

By combining two sigmoid functions, a valley function can be formed as

$$-f_{sig}(x+x_0) + f_{sig}(x-x_0) + \theta \qquad (9)$$

θ is a positive constant. This means two hidden units, having $f_{sig}()$, are required to achieve the same performance as in using a single hidden unit, having $f_{val}()$.

IV SIMULATION AND DISCUSSIONS

4.1 Conditions of Simulation

The minimum pulse width is 50 msec, the carrier frequency is f_o=880Hz, and the sampling frequency is f_s=4kHz. Thus, the minimum pulse width includes 200 samples. Examples of the ASK signal and random noise are shown in Figs.3(a) and 3(b), respectively. The number of samples, applied to the network in parallel, is chosen to 200, which can cover the minimum pulse width.

The input signal samples, occupy from 0 to 100 sec, are used for

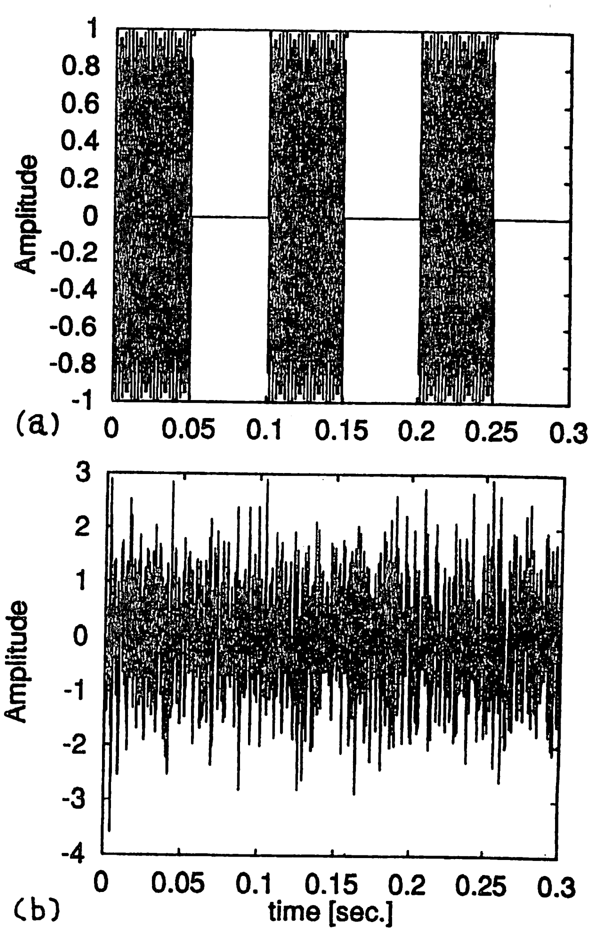

Fig.3 Examples of ASK signal (a) and white noise (b).

training. It includes 200 pulses with the minimum width and $4x10^5$ sampling points. After 100 sec, the input signal is used for evaluating performance of the trained and fixed neural demodulator.

4.2 Convergence Property and Generalization

Figure 4 shows learning curves. The curve A indicates the error squared $\delta^2(n)$ obtained using one hidden unit, having $f_{sig}()$. The curves B show the results of using two hidden units with $f_{sig}()$ and one hidden unit with $f_{val}()$. The curves C, D and E indicate three hidden units with $f_{sig}()$ and two and three hidden units with $f_{val}()$, respectively.

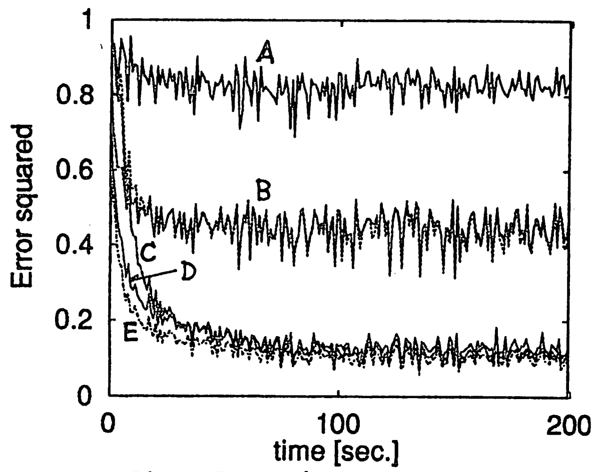

Fig.4 Learning curves.

In the first 100 sec period, the neural network is trained using x(n). In the rest 100 sec, the trained and fixed NN is examined using the new coming signal x(n). From Fig.4, it can be confirmed that the error does not increase after 100 sec. This guarantees generalization for untrained data. Furthermore, an error rate in detecting '1' and '0' in this interval is zero, that is 100% accuracy.

4.3 Activation Functions

From these results, it can be concluded that the valley function is more useful than the sigmoid one. Three hidden units are sufficient in this problem. As discussed in **Sec.3.2**, two sigmoid functions are required in order to achieve the same performance as in using a single valley function.

4.4 Output Signal and Error

Figure 5 shows the output y(n) with a solid line, the error δ (n) squared with a dotted line and the target $x_A(n-N/2)$ with a dashed line. From this figure, the neural network can outputs very sharp waveform. A single linear filter, designed to reject wide-band noise, cannot produce such a sharp response. Because a high-Q linear filter usually causes waveform distortion in the time domain.

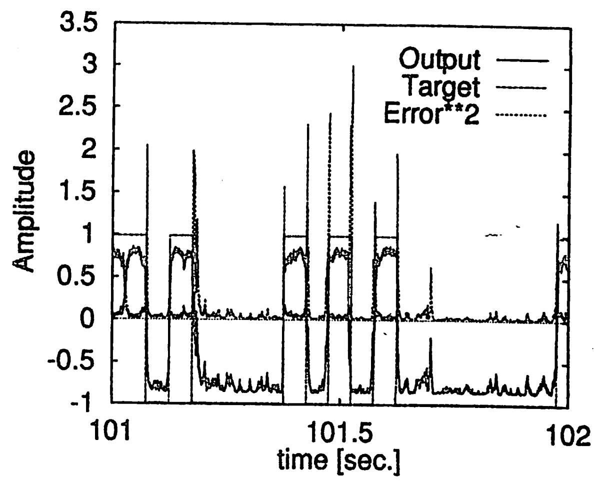

Fig.5 Output y(n)-solid line, target $x_A(n-N/2)$-dashed line and error δ (n)-dotted line.

4.5 Connection Weights

Figure 6 shows the connection weights from the input layer to two hidden units (a1) and (a2), and from the hidden layer to the output unit (b). Transformation from the hidden layer to the output is only summation.

Amplitude of Fourier transform for the connection weights (a1) in Fig.6 is shown in Fig.7. It has peak amplitude

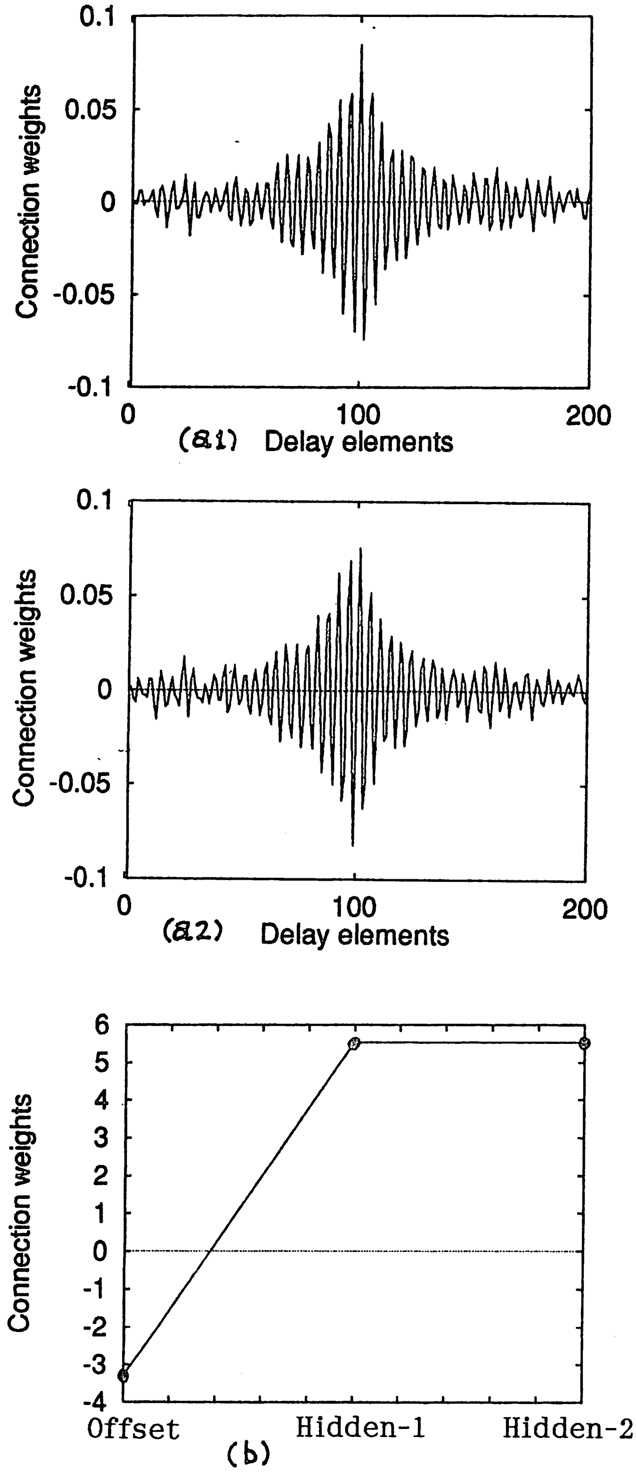

Fig.6 Connection weights from input layer to hidden layer (a1) and (a2), and from hidden layer to output unit (b).

at the signal frequency f_o=880Hz. This frequency response can be easily expected based on property of linear filtering. However, two sets of connection weights cooperate with each other to suppress wide-band noise, at the same time, to produce very sharp time response. For this purpose, fine adjustment of phase responses is required. The NN can do this optimization automatically.

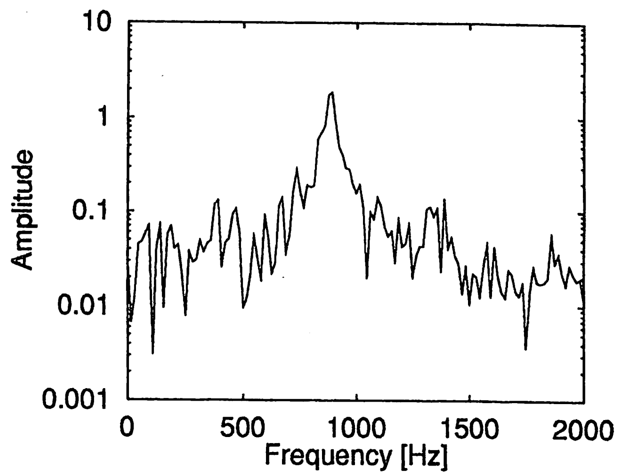

Fig.7 Amplitude of Fourier transform of connection weights shown in Fig.6(a1).

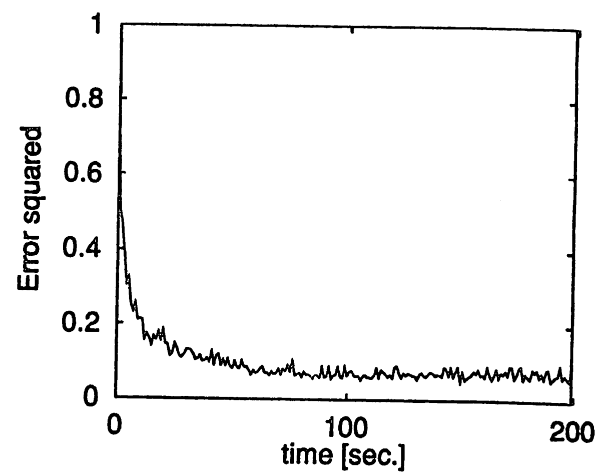

Fig.8 Learning curve for sinusoidal noise. Three hidden units, having valley function.

4.6 Rejection of Periodic Noise

Sinusoidal noise is added to the ASK signal. When the noise frequency is close to the signal frequency, a high-Q BPF is usually needed. This situation was also simulated using the proposed neural demodulator.

Figure 8 shows the learning curve obtained by using three hidden units, having the valley function. The noise frequency is 900Hz, which is very close to the signal frequency 880Hz. The error at 100 sec is almost the same as in Fig.4. Thus, the neural demodulator can suppress periodic noise, whose frequency locates very close the signal frequency.

4.7 Effects of Using Linear Bandpass Filter

Another way to reject noise is to use a bandpass filter (BPF) just before the NN. This method was also simulated for comparison. Figure 9 shows a learning curve. Three hidden units, having the valley function are used. As a result, the BPF method cannot provide good performance like the proposed NN. The reasons are explained in the following.

First, the BPF output still includes the noise, whose spectrum locates near by the signal. Even though the wide-band noise is suppressed, the narrow-band noise can remain, which has high correlation with the signal. This noise degrades waveform shaping.

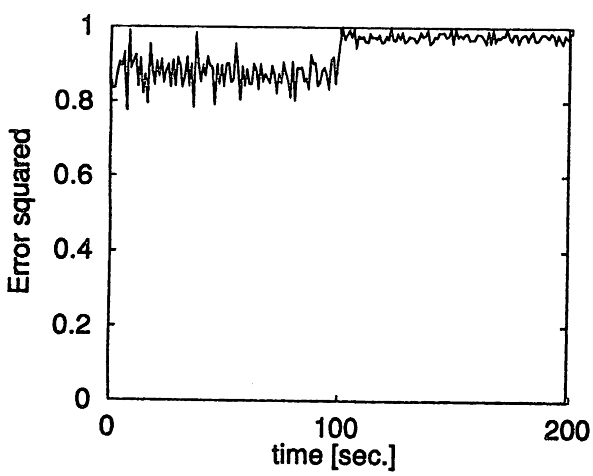

Fig.9 Learning curve for using BPF and three hidden units, having valley function.

On the other hand, the noise can be well reduced by using a very high-Q BPF. In this case, however, the time response, that is a pulse waveform response, is greatly distorted.

V CONCLUSION

The neural demodulator for ASK signals has been proposed. Necessary functions, including wide-band noise rejection, pulse waveform regeneration and decoding, can be embodied in the single NN. These functions are self-organized through the learning. The activation function has been proposed, which can play a role of rectifier.

Computer simulation shows efficiency of the proposed method. White noise and sinusoidal noise, having almost the same level as the signal power, can be rejected. At the same time, very sharp pulse waveform can be generated. Accuracy of demodulation was 100% in the interval from 100 sec to 200 sec. Conventional methods, using a BPF, cannot provide good performance like the NN.

REFERENCES

[1]A.Maren, C.Harston and R.Pap, Handbook of Neural Computing Applications, Academic Press Inc., 1990.

[2]D.R.Hush and B.G.Hrne, "Progress in supervised neural networks", IEEE Signal Processing Magazine, Jan.1993.

[3]T.Kohonen et al, Proc. IJCNN'90, San Diego, vol.I pp.223-228, June 1990.

[4]K.Hara and K.Nakayama, Proc., WCNN'93 Portland, vol.IV, pp.675-678, July 1993.

[5]K.Hara and K.Nakayama, Proc. IJCNN'93 Nagoya, pp.601-604, Oct. 1993.

[6]D.E.Rumelhart and J.L.McClelland, Parallel Distributed Processing, MIT Press, 1986.