A NEURAL DEMODULATOR FOR QUADRATURE AMPLITUDE MODULATION SIGNALS

Katsuyoshi OHNISHI Kenji NAKAYAMA
Department of Electrical and Computer Eng., Faculty of Eng., Kanazawa Univ.
2-40-20, Kodatsuno, Kanazawa 920, Japan
E-mail nakayama@ec.t.kanazawa-u.ac.jp

ABSTRACT

A neural demodulator is proposed for quadrature amplitude modulation (QAM) signals. It has several important features compared with conventional linear methods. First, necessary functions for the QAM demodulation, including wide-band noise rejection, pulse waveform shaping, and decoding, can be embedded in a single neural network. This means that these functions are not separately designed but are unified in a learning process. Second, these functions can be self-organized through the learning. Supervised learning algorithms, such as the back-propagation 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.

1. 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 ,[6], [7]. 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 quadrature amplitude modulation (QAM) signals is proposed. A multi-layer neural network (MLNN) and the back-propagation algorithm [8] are employed. The purpose of this model is to achieve both wide-band noise rejection, a very sharp waveform response and code transformation, 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.

2. Quadrature Amplitude Modulation and Demodulation

Figure 1 shows a transmitter of the 16QAM. Binary codes (s_{11}, s_{12}) and (s_{21}, s_{22}) are transformed into 4-level signals, and are bandlimited by the roll-off filters. Furthermore, they are modulated by the quadrature carriers, and are composed into $x_{QAM}(n)$.

$x_{QAM}(n)$ is expressed by

$$\begin{aligned} x_{QAM}(n) &= A \sum_{m=-\infty}^{\infty} a_m g((n-m)T) \cos \omega_c nT \\ &\quad -A \sum_{m=-\infty}^{\infty} b_m g((n-m)T) \sin \omega_c nT \end{aligned} \quad (1)$$

where, a_m and b_m are the 4-level signals, and $g(nT)$ is the impulse response of the roll-off filter. A is a some gain factor. For convenience, nT is represented by only n in this paper.

0-7803-3210-5/96 $4.00©1996 IEEE

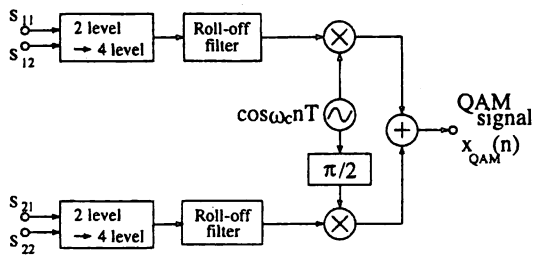

Fig. 1: Transmitter of 16QAM.

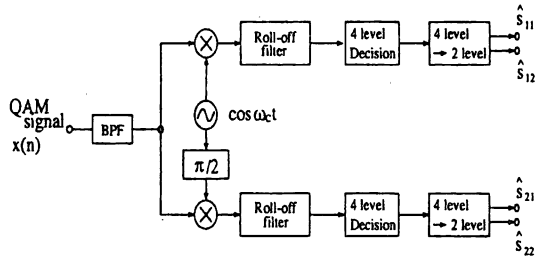

Fig. 2: Receiver of 16QAM.

The transmitted signal is demodulated at the receiver. A block diagram for the conventional demodulator model is illustrated in Fig.2. The noise included in the signal is rejected by passing a band-pass filter (BPF) at the front-end of the receiver. The signal is separated into two channels, the in-phase channel (I-channel) and the quadrature channel (Q-channel). The original signal is extracted from the coherent detected signal through a low-pass filter (Roll-off filter). The extracted multi-level code is transformed into the binary code.

3. Neural Demodulator for 16QAM

A block diagram for the proposed neural demodulator model is illustrated in Fig.3. The received signal $x(n)$ includes the QAM signal $x_{QAM}(n)$ and additive-white-Gaussian noise (AWGN) $noise(n)$ as follows:

$$x(n) = x_{QAM}(n) + noise(n) \tag{2}$$

This signal is applied to the neural demodulator directly, and are transformed into the binary codes.

Figure 4 shows the proposed multi-layer 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 i th delay element is denoted $x_l(n-i)$. N samples of $x(n)$, that is $x_l(n-i), i = 0 \sim N-1$, are transmitted through the connections in parallel. A bias unit is used, which always outputs 1 to the hidden units and the output units in order to adjust bias.

Let the connection weight from the i th delay element to the j th hidden unit be w_{lji}, and from the j th hidden unit to the k th output unit be w_{lkj}. Network equations are expressed as follows:

$$u_{lj}(n) = \sum_{i=0}^{N-1} w_{lji} x_l(n-i) + w_{lj} \tag{3}$$

$$v_{lj}(n) = f_H(u_{lj}(n)) \tag{4}$$

$$net_{lk}(n) = \sum_{j=0}^{M-1} w_{lkj} v_{lj}(n) + w_{lk} \tag{5}$$

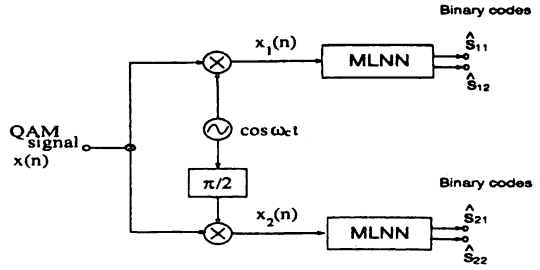

Fig. 3: Neural demodulator for 16QAM.

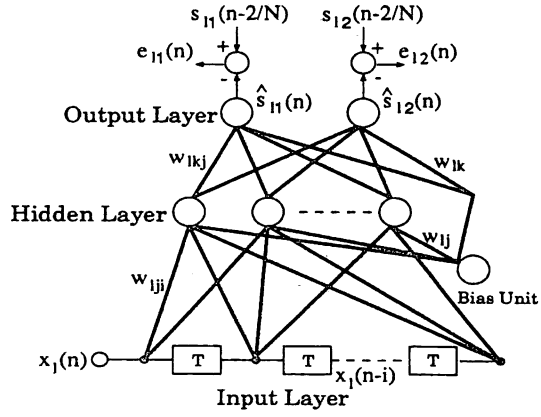

Fig. 4: Neural multi-layer demodulator for 16QAM signals. $l = 1, 2$.

$$\hat{s}_{lk}(n) = f_{Ok}(net_{lk}(n)) \quad (6)$$

w_{lj} and w_{lk} are the connection weights from the bias unit to the j th hidden unit and the k th output unit, respectively. $f_H(\cdot)$ and $f_{Ok}(\cdot)$ are activation functions of the hidden units and the k th output unit, respectively.

The original binary data in each channel, that is $s_{11}(n)$, $s_{12}(n)$, $s_{21}(n)$ and $s_{22}(n)$, are used as targets. Since these data are modulated in generating the QAM signal, a sample of $x_{QAM}(n)$ is closely related to its neighborhood samples, that is $x_{QAM}(n-i)$, $-N/2 \leq i \leq -1$ and $+1 \leq i \leq +N/2$. Furthermore, $\hat{s}_{lk}(n)$ is calculated using $x_l(n-i), 0 \leq i \leq N-1$, as shown in Eqs.(2)-(7). $x_l(n)$ is given by

$$x_l(n) = x(n)cos(\omega_c \frac{n}{f_s} + \phi_l) \qquad \phi_1 = 0, \phi_2 = \pi/2 \quad (7)$$

Therefore, $s_{lk}(n-N/2), l = 1,2, k = 1,2$, are used as the targets for $\hat{s}_{lk}(n)$. The output error is evaluated by

$$e_{lk}(n) = s_{lk}(n - N/2) - \hat{s}_{lk}(n) \quad (8)$$

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

4. Learning Algorithm and Activation Functions

4.1. Learning Algorithm

The Back-propagation algorithm is very powerful for the MLNNs. It is also employed in our model. An important point of the learning is to automatically design necessary functions, including wideband noise rejection and pulse waveform regeneration. Because these requirements are difficult to be satisfied simultaneously by linear signal processing.

4.2. Activation Functions

Another important point of the neural network design is to optimize activation functions for each application. In this paper, we use a sigmoid function given by Eq.(9) in the hidden layer.

$$f_H(x) = \frac{1 + e^{-x}}{1 - e^{-x}} \quad (9)$$

In the output layer, the received multi-level waveforms should be transformed into two-level waveforms, that is binary code. Four-level signals are transformed into 2-bit binary signals. For this purpose, different kinds of activation functions are employed for the most significant bit (MSB) and the least significant bit (LSB). They are given by

$$\text{MSB: } f_{o1}(x) = \frac{1}{1 - e^{-\beta x}} \quad (10)$$

$$\text{LSB: } f_{o2}(x) = f_{o1}(x + \text{TH}) - f_{o1}(x) + f_{o1}(x - \text{TH}) \quad (11)$$

These activation functions are shown in Fig.5, where the binary codes are also shown. Using these nonlinear functions, the multi-level signals can be linearly transformed into the binary codes.

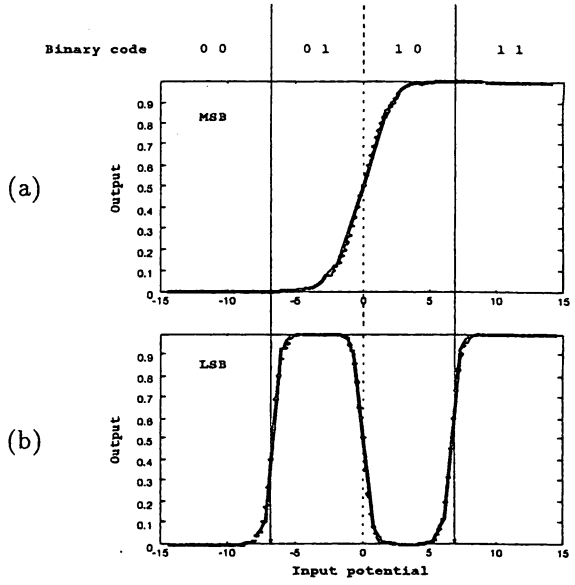

Fig. 5: Activation function of output unit.

5. Simulation and Discussions

5.1. Conditions of Simulation

The baseband pulse signals of each channel have the amplitude of $\pm 3, \pm 1$, and the minimum pulse width is $T = 1$ sec. The carrier frequency is $f_c = 20$ Hz, and the sampling frequency is $f_s = 200$ Hz. These frequencies are normalized for convenience. The number of samples, applied to the network in parallel, is chosen to 200, which can cover the minimum pulse width. In each MLNN, the numbers of the input units and the hidden units are 200 and 16,

respectively. The number of the output units is two corresponding to $\hat{s}_{l1}$ and $\hat{s}_{l2}$. Examples of the QAM signals and AWGN are shown in Fig.6 and Fig.7, respectively.

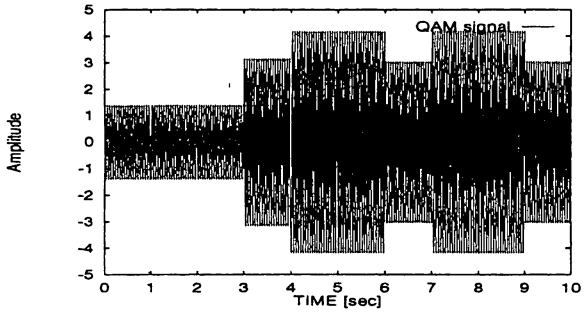

Fig. 6: Example of QAM signal.

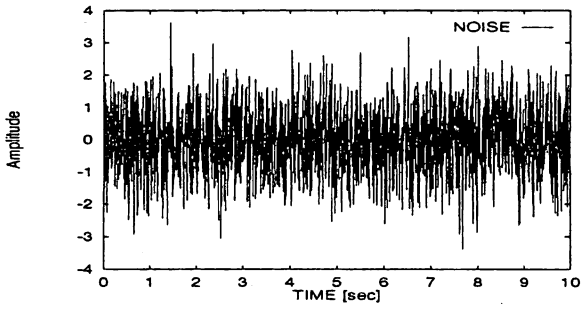

Fig. 7: Example of white Gaussian noise.

Simulation was done using 16QAM signal including AWGN. Furthermore, a phase difference between carriers used in the transmitter and in the receiver is also taken into account.

The MLNN is trained during some interval, where 7×10^7 signal samples, that is 3.5×10^5 pulses are included. After this interval, the training is stopped, and the network is fixed. Different data sequence, that is test data, are applied to this network for evaluating generalization.

5.2. Convergence Property and Generalization

Figure 8 shows a learning curve. The AWGN is included, and a signal-to-noise ratio (SNR) is 7 dB.

AWGN Environment

The neural demodulator is evaluated by using the QAM signal with different data sequence from the training signal and with the AWGN, SNR = 7 dB. Figure 9 shows the output $\hat{s}_{11}(n)$.

From this figure, the neural demodulator can output very sharp waveforms. A linear filter, designed to reject the wide-band noise, cannot produce such a sharp response. Because a high-Q linear filter usually causes waveform distortion in the time domain.

Phase Error Environment

In the coherent detector, the phase difference between the carriers in the transmitter and the receiver may causes the output waveform distortion. In the conventional demodulator, the phase difference $\delta\theta$ causes the following amplitude distortion.

$$S = \frac{a_i}{2} cos(\delta\theta) - \frac{b_i}{2} sin(\delta\theta) \qquad (12)$$

Where, S is signal after passing a filter. a_i and b_i are baseband signals of I-channel and Q-channel, respectively. $\delta\theta$ is phase error. When $\delta\theta = 10$ degrees, the amplitude, which ideally takes 1 or 0, may change by 0.35. Therefore, the noise margin is decreased to 0.15, while the ideal noise margin is 0.5.

Figure 10 shows the output $\hat{s}_{11}(n)$ with the phase error $\delta\theta = 10$ degrees. The output waveform is still very sharp. The waveform distortion is about 0.13 at most in the middle part of the pulse waveform as shown in Fig.10. Thus, the neural demodulator is robust for the phase error of the carriers.

5.3. Connection Weights

Figures 11(a) and 11(b) show the examples of the connection weights from the input layer to the hidden units. Amplitude of the Fourier transform for these connection weights are shown in Figs.12(a) and 12(b), respectively. From these figures, the connection weights from the input layer to the hidden unit are similar to an impulse response of low pass filters (LPFs). Both sets of the connection weights have the amplitude response of LPF. Because, the frequency component of $x_{QAM}(n)$ is shifted to DC and 40 Hz by multiplying $x_{QAM}(n)$ by $cos(\omega_c t)$ or $sin(\omega_c t)$. The DC component is used to regenerate pulse waveforms. Thus, the DC component should be passed, and the 40 Hz component and the AWGN should be suppressed. Such characteristics are achieved through the supervised learning.

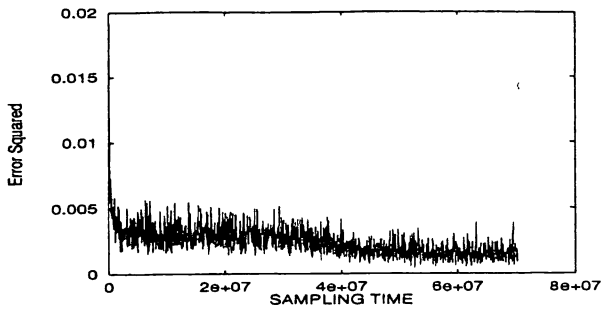

Fig. 8: Learning curve for 16QAM with white Gaussian noise. Signal-to-noise ratio is 7 dB.

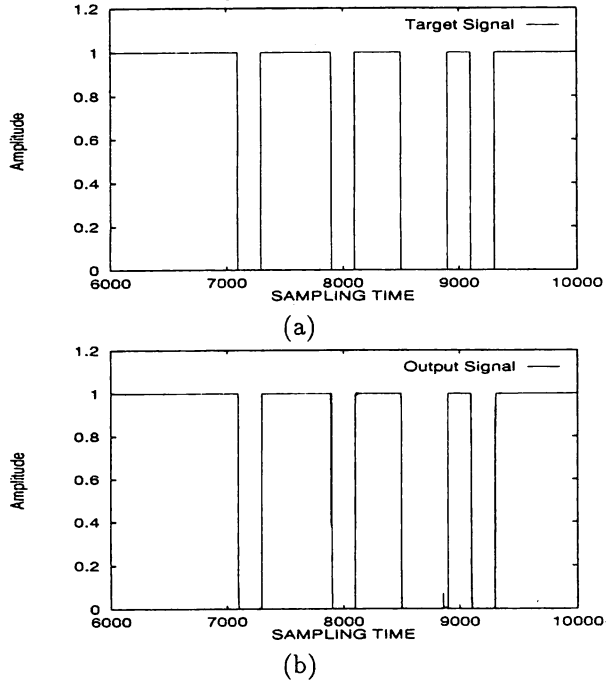

Fig. 9: Example of (a) target signal $s_{11}(n)$ and (b) output signal $\hat{s}_{11}(n)$. SNR = 7 dB.

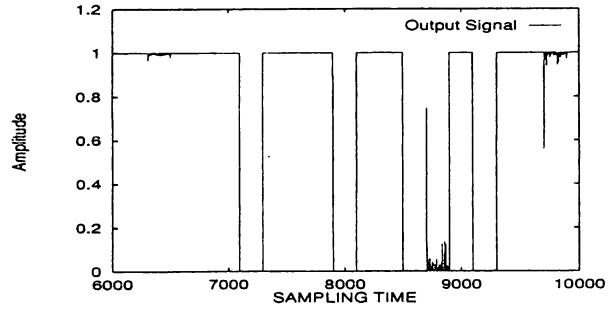

Fig. 10: Example of output signal $\hat{s}_{11}(n)$. Phase error is 10 degrees.

Furthermore, the following should be pointed out. Using 16 hidden units mean to use 16 nonlinear filters. Each filter can suppress the noise and the unnecessary component. At the same time, they cooperate with each other in pulse waveform shaping. In the case of linear signal processing, a single linear filter is usually used. Furthermore, using several linear filters in parallel cannot provide any benefit, because the composition of them is equivalent to a single filter. Thus, its performance is limited, compared with the neural version.

6. Conclusions

The neural demodulator for the 16QAM signals has been proposed. Necessary functions, including wideband noise rejection, pulse waveform regeneration and decoding can be embedded in the single NLNN. These functions are self-organized through the learning. The activations have been optimized.

Computer simulation shows efficiency of the proposed method. AWGN, up to SNR = 7 dB, can be well rejected. The effect of the phase error of the carriers up to 10 degrees is also suppressed. Furthermore, the very sharp pulse waveform can be generated. Accuracy of demodulation was very high. Conventional methods, using a linear filter, cannot provide good performance like the proposed MLNN.

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.Horne, "Progress in supervised neural networks," IEEE Signal Processing Magazine, Jan.1993.

[3] T.Kohonen et al, "Combining linear Equalization and self-organizing adaptation in dynamic discrete-signal detection," Proc, IJCNN'90, San Diego, vol.I, pp.223-228, June 1990.

[4] K.Hara and K.Nakayama, "High resolution of multi-frequencies using multilayer networks trained by back-propagation algorithm," Proc. WCNN'93 Portland, vol.IV, pp.675-678, July 1993.

[5] K.Hara and K.Nakayama, "Classification of multi-frequency signals with random noise using multilayer neural networks," Proc. IJCNN'93 Nagoya, pp.601-604, Oct 1993.

[6] K.Nakayama and K.Imai, "A neural demodulator for amplitude shift keying signal," Proc.

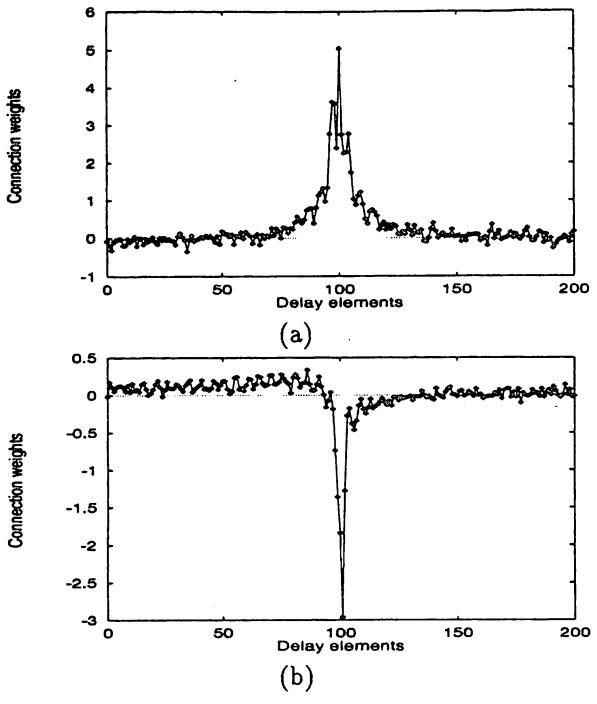

Fig. 11: Examples of connection weights from input layer to hidden layer.

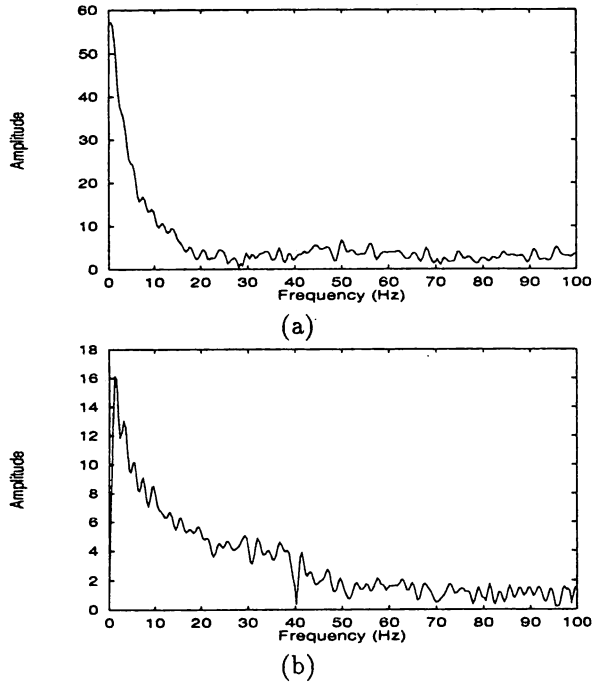

Fig. 12: Examples of amplitude of the Fourier transform of connection weights shown in Figs.11(a) and 11(b).

ICANN'94, Sorrento, Italy, pp.1017-1020, May 1994.

[7] D.P.Bouras, P.T.Mathiopoulos and D.Makrakis, "Neural-net based receiver structures for single- and multi-amplitude signals in interference channels," Proc. IEEE workshop on Neural Networks for Signal Processing IV, pp. 535-544, 1994.

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