Nonlinear Channel Equalization Using Fuzzy CMAC Neural Network

C. M. Kwan*, R. Xu*, L. Haynes*, J. D. Pryor **

* Intelligent Automation Inc.
2 Research Place
Suite 202
Rockville, MD 20850.
Tel. (301) - 590 - 3155
Fax. (301) - 590 - 9414
e-mail: ckwan@i-a-i.com

** United States Army
Space & Strategic Defense Command
Concepts and Applications Division
P.O. Box 1500
Huntsville, AL 35807-3801.

Keywords: nonlinear channel, equalization, Fuzzy CMAC

Abstract

It is well known that nonlinear distortion over a communication channel is now a significant factor hindering further increase in the attainable data rate in high-speed data transmission. Since the received signal over a nonlinear channel is a nonlinear function of the past values of the transmitted data pulses, it is not surprising that the linear equalizers do not work efficiently. We propose a new nonlinear equalizer that uses a new type of neural network called Fuzzy CMAC which combines the advantages of both fuzzy logic and CMAC (Cerebellar Model Arithmetic Computer) networks. The learning speed is an order of magnitude faster than conventional neural nets. Moreover, human expert knowledge in the form of linguistic rules can be easily incorporated into the scheme.

1. Introduction

In practical digital communications systems designed to transmit at high speed over band-limited channels, a succession of pulses transmitted through the channel at rates comparable to the channel bandwidth are smeared to the point that they are no longer distinguishable as well-defined pulses at the receiving terminal [1]. This is due to several factors. First, *linear amplitude and delay distortion* caused by the nonideal channel frequency response characteristic makes the symbols overlap. Also the delay distortion makes the zero crossings no longer periodically spaced. Second, signals transmitted through telephone channels are subject to other impairments such as nonlinear distortion, frequency offset, phase jitter, impulse noise, and thermal noise. *Nonlinear distortion* in telephone channels is due to the nonlinearities in amplifiers and compandors used in the telephone system. This type of distortion is very difficult to correct. A small *frequency offset* (< 5 Hz) results from the use of carrier equipment in the telephone channel. Such an offset cannot be tolerated in high-speed digital transmission systems which use synchronous phase-coherent demodulation. The offset can be compensated for by the carrier recovery loop in the demodulator. *Phase jitter* is a low--index frequency modiulation of the transmitted signal with the low frequency harmonics of the power line frequency (50 or 60 Hz). Phase jitter poses a serious problem in digital transmission of high rates. However, it can be tracked and compensated for at the demodulator. *Impulse noise* is an additive disturbance. It comes from the switching equipment in the telephone system. *Thermal noise* is also present at levels of 20 to 30 dB below the signal.

In this paper, we will deal with the signal distortion due to the linear and nonlinear channel characteristics and additive noises. Since frequency offset and phase jitter can be remedied by some other means, we will not discuss these impairments.

The popular approach to channel equalization is to model the channel as a linear finite-impulse-response (FIR) filter. Then an adaptive filter is used to cancel the distortion caused the channel. Various weight adjustment schemes such as LMS (Least Mean Square), RLS (recursive least-squares), and lattice algorithms are used to tune the filter parameters [1] [2]. However, these schemes can only handle linear distortions. It is well known that nonlinear distortion over a communication channel is now a significant factor hindering further increase in the attainable data rate in high-speed data transmission. Since the received signal over a nonlinear channel is a nonlinear function of the past values of the transmitted data pulses, it is not surprising that the linear equalizers do not work efficiently. Therefore, an efficient and effective equalizer should posses certain capability that can learn the nonlinear behavior of the channel characteristics. In [4] and [6], polynomial adaptive filters were developed for nonlinear channel equalization. In [5], multilayer perceptrons have been used as equalizers. In [3], fuzzy adaptive filters were used to cancel the nonlinear distortion.

Since nonlinear channels cover many kinds of nonlinear distortions, it is hard to single out one method that is dominant. Hence it is necessary to develop new nonlinear equalizers and this is the goal of this paper. We propose a new nonlinear equalizer that uses a new type of neural network called Fuzzy CMAC which combines the advantages of both fuzzy logic and CMAC (Cerebellar Model Arithmetic Computer) networks. The learning speed is an order of magnitude faster than conventional neural nets. Moreover, human expert knowledge in the form of linguistic rules can be easily incorporated into the scheme.

The paper is organized as follows. In section 2, we will introduce some backgrounds on Fuzzy CMAC. Then we will describe the channel equalization problem and our approach in section 3. Simulation results will be included. Finally, conclusions will be drawn in Section 4.

2. Fuzzy CMAC Neural Network

2.1 Comparison of a CMAC Network with a Fuzzy Logic Controller (FLC)

Two decades ago, a unique neural network model called a Cerebellar Model Arithmetic Computer (CMAC) was established by J. Albus [8] based on a model of the human memory and neuromuscular control system. The CMAC is a perceptron-like associative memory that performs nonlinear function mapping over a particular region of a function space. A CMAC network has the capability to learn an unknown nonlinear mapping by examples, and to reproduce multiple outputs in response to multiple inputs. Because of its table look-up mechanism, and its hash-code based mapping structure, CMACs are able to cope with high dimensional input/output applications without severely deteriorating their processing speed and performance.

Fuzzy set theory was initially proposed by L. Zadeh as a tool to model the imprecision that is inherent in human reasoning, especially when dealing with complexity. The fuzzy theory has seen its most widespread application in the area of control. Controllers using control laws specified with fuzzy set theory (or fuzzy logic) are known as fuzzy controllers. Such controllers are easier (relative to non-fuzzy controllers) to design, especially in cases where the laws are non-linear and the systems are complex.

A fuzzy logic controller consists of an input fuzzification module, a set of fuzzy control rules based on which the approximate reasoning technique is used to make a control decision, and an output defuzzification module.

Fuzzy logic controllers differ from classical math-model controllers in that they do not require an explicit mathematical model of how control outputs functionally depend on control inputs. Also fuzzy logic controllers allow designers to incorporate human knowledge into the control decision making process.

The advantages of a CMAC over a FLC are:

- There are very efficient learning laws to update the values of weights based on experience and examples.
- There is a random mapping mechanism to reduce the physical memory requirement for multiple input and high resolution applications.
- There exist efficient input encoding schemes for high dimensional input vectors.

The advantages of a FLC over a CMAC are:

- It is possible to interpret the implication of weight values using linguistic labels.
- The membership functions and the firing strengths contain additional information as to how close the input vector is to each linguistic variable. Therefore the number of input space partitions may be smaller to achieve the same generalization and output smoothness.
- The fuzzy rules can take a variety of forms while only numeric values can be associated with CMAC associative memory locations.
- There are many methods to construct a fuzzy control knowledge base, such as expert's experience and knowledge.

In the next section. we will propose a new neural network called Fuzzy CMAC which combines the advantages of both fuzzy logic and CMAC network.

2.2 Fuzzy CMAC Architecture

In this section, we present the novel concept of a Fuzzy CMAC neural network. This novel network was developed by Intelligent Automation Inc. (IAI). The idea is to combine the advantages of the FLC and the CMAC (Cerebellar Model Arithmetic Computer) and eliminate the respective disadvantages of the two. IAI has successfully applied this network to active vibration control, finger print analysis, and chaotic time-series prediction. Fig. 1 illustrates the architecture of the Fuzzy CMAC. The Fuzzy CMAC inherits the preferred features of arbitrary function approximation, learning, and parallel processing from the original CMAC neural network, and the capability of acquiring and incorporating human knowledge into a system and the capability of processing information based on fuzzy inference rules from fuzzy logic. The combination of a neural network and fuzzy logic yields an advanced intelligent system architecture.

At the input stage, the Fuzzy CMAC uses the fuzzification method of a FLC as its input encoding scheme. Fuzzy rules can be assigned to each associative memory cell. These rules may not necessarily have a crisp consequent part. The output generation uses a defuzzification approach which sums weighted outputs of the activated rules based on the firing strengths s_i. The overall mapping function of a Fuzzy CMAC can be formalized as

$$z(u) = \sum_{p=1}^{M} s_p w_p \tag{2.1}$$

where u $= [u_1 \quad u_2 \quad \ldots. \quad u_N]^T$ is the input vector. w_p , p = 1, 2, .., M, are the weights of the network. A4 = j_1 if N = 1 and $M = \sum_{i=2}^{N}(j_i - 1)\prod_{l=1}^{i-1} m_{l-1} + j_1$, for N > 1. $i = 1,2,\ldots,N$. m_i is the number of knot points on the i^{th} input dimension. The j_i^{th} knot point on the i^{th} input dimension is denoted as u_{i,j_i}, $j_i = 1,2,\ldots,m_i$. s_p , p = 1, 2, . .., M, are the firing strength of the fuzzy rules. They are calculated as

$$s_p = \mu_{1,j_1}(u_1)*\mu_{2,j_2}(u_2)*\ldots.*\mu_{N,j_N}(u_N) \qquad (2.2)$$

where * denotes the T-norm operator. There are many types of T-norms such as the min and product operators. Throughout this paper, we choose the T-norm as the product inference method since it is easy to implement. Hence, (2.2) can be written as

$$s_p = \mu_{1,j_1}(u_1)\mu_{2,j_2}(u_2)\ldots\mu_{N,j_N}(u_N) = \prod_{i=1}^{N}\mu_{i,j_i}(u_i) \qquad (2.3)$$

where $\mu_{i,j_i}(u_i)$ is the j_i^{th} membership function of the i^{th} input.

A Fuzzy CMAC neural network combines the advantages listed for CMAC and FLC. One desirable feature the Fuzzy CMAC inherits from the CMAC model is that the receptive field of the sensing element has limited width. This means that there are only a small number of sensing elements to ‘be activated for any sensor reading. In conventional neural nets such as those based on multiple layer feedforward structure and backpropagation learning algorithms, all neurons are required to perform computation in order to compute the forward mapping or to perform learning. In Fuzzy CMACs, operations are localized and only a small subset of all the neurons need to be computed. Our initial comparisons of fuzzy CMACs with conventional neural nets show orders of magnitude increase in the speed of both function mapping and learning for typical problems on conventional computational hardware. Comparing the activating function of CMAC to the linguistic variables in the Fuzzy CMAC, one can view the activating function as the membership function of the Fuzzy CMAC input variables. Fig. 2 shows the comparison.

In terms of the control knowledge rule-base, the proposed Fuzzy CMAC differs from Albus’s CMAC [8] in that the weight values can be interpreted as knowledge rules through linguistic variables. This feature permits us to validate a learned Fuzzy CMAC in terms of the reasonableness of the learning results. This unique feature also provides a practical channel for knowledge acquisition. A knowledge base (a set of rules) can be established based on the learning results of a Fuzzy CMAC.

On the other hand, the Fuzzy CMAC distinguishes itself from Zadeh’s fuzzy controller in that it is able to build a learning control system starting with an empty knowledge base. The linguistic knowledge of human experts can be incorporated into these rules. Based on the initial knowledge base, a self learning algorithm is employed to modify the existing rules in order to improve the system performance.

2.3 Fuzzy CMAC Mapping Using B-Spline Membership Function

The fuzzy membership function can be chosen with considerable freedom, and this freedom can be used to optimize system performance. Membership functions should be computationally simple, flexible, and continuous in order to optimize the Fuzzy CMAC system design. We have compared many candidates for membership function, such as bell shape Gaussian functions, sinusoidal functions, etc., and we believe that the B-Spline function family possesses certain preferred properties that make them well suitable for the Fuzzy CMAC membership function. Let the j_i^{th} knot point on the i^{th} input dimension be denoted as $\mu_{i,j}$. 'Throughout this paper. we use the third order B-spline function. The firing strength of a fuzzy membership function $\mu(u_i)$ is the value of the function μ at the input u . For example, if u lies in the range of $\left[\mu_{i,j_i-1}(u), \mu_{i,j}(u)\right]$, then the firing strength of the j_i^{th} membership function will be

$$s_{i,j} = \frac{u - u_{i,j_i-2}}{u_{i,j_i} - u_{i,j_i-2}} \frac{u_{i,j_i} - u}{u_{i,j_i} - u_{i,j_i-1}} + \frac{u_{i,j_i+1} - u}{u_{i,j_i+1} - u_{i,j_i-1}} \cdot \frac{u - u_{i,j_i-1}}{u_{i,j_i} - u_{i,j_i-1}}. \tag{2.3}$$

The third order B-Spline membership functions have the following desirable properties:

(a) Positivity: $\mu_{i,j_i}(u_i) > 0$ for all $u_i \in \left[u_{i,j_i-3}, u_{i,j_i}\right]$.

(b) Compact support: $\mu_{i,j}(u_i) = 0$ for all u_i not belonging to $u_i \notin \left[u_{i,j-3}, u_{i,j}\right]$

(c) Normalization: $\sum \mu_{i,j}(u_i) = 1$ for any u_i.

(d) Derivatives exist and can be recursively calculated.

It should be noted that the positivity property is not that important in Fuzzy CMAC. However, the other three properties require more attention. The existence of derivatives is very important for many real-time control applications where the tracking errors are back-propagated through the Fuzzy CMAC. Compact support property means that? for a given input vector, only a small number of the membership functions will be fired and need to be computed. For our third order B-spline function, only 3 membership functions will be activated in each input dimension. Thus only a small number of weights need to be updated. In other words, our Fuzzy CMAC enjoys the localization property which represents a considerable amount of' computational savings when compared to a general feedforward backpropagation neural network which must calculate all the network weights whenever there is a new input vector. The smoothness of the B-spline functions also give the Fuzzy CMAC the generalization property which means similar inputs will give similar outputs. The normalization property will help in simplifying the computations. Since the output of a Fuzzy CMAC can be expressed as

$$z(u) = \frac{\sum_{j_N=1}^{m_N} \sum_{j_{N-1}=1}^{m_{N-1}} \cdots \sum_{j_1=1}^{m_1} \left[\prod_{i=1}^{N} \mu_{i,j_i}(u_i)\right] w(j_1, j_2, \ldots, j_N)}{\sum_{j_N=1}^{m_N} \sum_{j_{N-1}=1}^{m_{N-1}} \cdots \sum_{j_1=1}^{m_1} \prod_{i=1}^{N} \mu_{i,j_i}(u_i)}, \tag{2.5}$$

it seems from the above formula that the calculation of the numerator as well as the denominator of the *z(u)* both involve a summation of M (= $\prod_{i=1}^{N} m_i$) terms, each in turn requires N multiplications. By using the normalization property of the B-spline membership function, the calculation of the *z(u)* becomes much simpler. This fact is proven in the following theorem.

Theorem 1

Because of the normalization property of B-spline function, the denominator in (2.1) always takes the value 1, e.g.

$$\sum_{j_N=1}^{m_N} \sum_{j_{N-1}=1}^{m_{N-1}} \cdots \sum_{j_1=1}^{m_1} \sum_{i=1}^{N} \mu_{i,j_i}(u_i) = 1 \tag{2.6}$$

Proof: See [7].

2.4 Fuzzy CMAC Universal Approximation Theorem

We have proven the following universal approximation theorem for the Fuzzy CMAC neural network. The theorem essentially guarantees that the Fuzzy CMAC neural network can uniformly approximate any nonlinear continuous function over a specific compact region to any degree of accuracy. The universal approximation theorem provides us with justification for applying the Fuzzy CMAC to almost any nonlinear system modeling problem.

Theorem 2

For any given real continuous function *g(u)* on a compact set $U \subset R^N$, and arbitrary $\varepsilon > 0$, there exists a Fuzzy CMAC neural network *z(u)* in the form of (2.1) such that

$$\sup|z(u) - g(u)| < \varepsilon \qquad \text{for} \qquad \forall u \subset U \tag{2.7}$$

where $u = [u_1 \quad u_2 \quad \ldots \quad u_N]^T$ is the input vector; $w(j_1, j_2, \ldots, j_N)$ are the weights of the Fuzzy CMAC; and $\mu_{i\ j_i}$ are the membership function values.

Proof: See [7].

2.5 Learning Algorithm for the Fuzzy CMAC

The conventional CMAC learning process is designed so that the correction initiated by an output error is evenly distributed among all weights that contribute to the output, regardless of their true proportion of contribution to the output. In the Fuzzy CMAC, a more intelligent method is adopted in which the correction of the output error is distributed to the weights in proportion to their contribution. The weight updating algorithm is described below.

Given a training pair (u, z_d) where u = $[u_1, u_2, \ldots, u_N]^T$ is input vector and z_d is the desired output, update weights and knot points associated with fired fuzzy rules, such that the error between z and z_d is reduced. Suppose there is only one output and many inputs. The Fuzzy CMAC mapping can be expressed in terms of the product of firing strength and weights

$$z = \sum_{p=1}^{M} s_p w_p \ . \tag{2.8}$$

The learning algorithm is targeted to reduce error between the desired output z_d and the actual output of Fuzzy CMAC by adjusting the values of weights

$$E = \frac{1}{2}(z_d - z)^2 = \frac{1}{2}e^2 \ . \tag{2.9}$$

The adjustment of the weights is based on the following rules

$$\Delta w_p = -\beta e s_p , \qquad p= 1, 2,..., M \tag{2.10}$$

where β is the learning rate. These learning rules differ from the original CMAC learning rule proposed by Albus in that the adjustment of a weight is proportional to its contribution (measured by its firing strength) to the output signal based on which the error is generated. The conventional CMAC distributes the output error evenly among all the contributing weights.

3. Nonlinear Channel Equalization Using Fuzzy CMAC

A typical digital communication system is shown in Fig. 3 [3]. The channel includes the effects of the transmitter filter, the transmission medium, the receiver matched filter, and other components. The transmitted data sequence $s(k)$ is assumed to be an independent sequence taking values from $\{-1, 1\}$ with equal probability. The inputs to the equalizer. $x(k), x(k-1), \ldots, x(k-n+1)$ are the channel outputs corrupted by an additive noise $\eta(k)$. The task of the equalizer at the sampling instant k is to produce an estimate of the transmitted symbol $s(k-d)$ using the information contained in $x(k)$, $x(k-1), \ldots, x(k-n+1)$. The integers n, d are known as the order and lag of the equalizer, respectively.

The equalization problem can be formulated as follows. Similar to [5], we define

$$P_{n,d}(1) = \{\underline{\hat{x}}(k) \in R^n \mid s(k-d) = 1\},$$

$$P_{n,d}(-1) = \{\underline{\hat{x}}(k) \in R^n \mid s(k-d) = -1\} \tag{3.1}$$

where

$$\underline{\hat{x}}(k) = \left[\hat{x}(k), \hat{x}(k-1), \ldots, \hat{x}(k-n+1)\right]^T.$$

Note that $\underline{\hat{x}}(k)$ is the noise-free output of the channel, and $P_{n,d}$ (1) and $P_{n,d}$ (-1) represent the two sets of possible channel noise-free output vectors $\underline{\hat{x}}(k)$ that can be produced from sequences of the channel inputs containing $s(k-d) = 1$ and $s(k-d) = -1$, respectively. The equalizer can be characterized by the function

$$g_k : R^n \to \{-1, 1\} \tag{3.2}$$

with

$$\hat{s}(k-d) = g_k(\underline{x}(k))$$

where

$$\underline{x}(k) = \left[x(k), x(k-1), \ldots, x(k-n+1)\right]^T$$

is the observed channel output. Let $p_1\left[\underline{x}(k) \mid \underline{\hat{x}}(k) \in P_{n,d}(1)\right]$ and $p_{-1}\left[\underline{x}(k) \mid \underline{\hat{x}}(k) \in P_{n,d}(-1)\right]$ be the conditional probability density functions of x(k) given $\underline{\hat{x}}(k) \in P_{n,d}(1)$ and $\underline{\hat{x}}(k) \in P_{n,d}(-1)$, respectively It was shown in [5] that the equalizer which is defined by

$$f_{opt}(\underline{x}(k)) = \mathrm{sgn}[p_1(\underline{x}(k) \mid \underline{\hat{x}}(k) \in P_{n,d}(1)) - p_{-1}(\underline{x}(k) \mid \underline{\hat{x}}(k) \in P_{n,d}(-1))] \tag{3.3}$$

achieves the minimum bit error rate for the given order n and lag d, where $\mathrm{sgn}(y) = 1\ (-1)$ if $y \ge 0\ (y < 0)$. If the noise $\eta(k)$ is zero-mean and Gaussian with covariance matrix

$$Q = E[(\eta(k), \ldots, \eta(k-n+1))(\eta(k), \ldots, \eta(k-n+1))^T], \tag{3.4}$$

then

$$p_1(\underline{x}(k) \mid \underline{\hat{x}}(k) \in P_{n,d}(1)) - p_{-1}(\underline{x}(k) \mid \underline{\hat{x}}(k) \in P_{n,d}(-1))$$

$$= \sum_{\underline{\hat{x}}_+ \in P_{n,d}(1)} \exp\left[-\frac{1}{2}(\underline{x}(k) - \underline{\hat{x}}_+)^T Q^{-1}(\underline{x}(k) - \underline{\hat{x}}_+)\right] - \sum_{\underline{\hat{x}}_- \in P_{n,d}(-1)} \exp\left[-\frac{1}{2}(\underline{x}(k) - \underline{\hat{x}}_-)^T Q^{-1}(\underline{x}(k) - \underline{\hat{x}}_-)\right].$$

Let us consider a nonlinear channel described by the following discrete model

$$\hat{x}(k) = s(k) + 0.5s(k-1) - 0.9[s(k) + 0.5s(k-1)]^3 \tag{3.5}$$

and the white Gaussian noise $\eta(k)$ with $E(\eta^2(k)) = 0.2$. The optimal decision boundary for this case is shown in Fig. 4. The elements of the sets $P_{2,0}$ (1) and $P_{2,0}$ (-1) are illustrated in Fig. 4 by the "o" and "*" respectively. It can be seen that the boundary is very nonlinear. Linear equalization techniques will not perform well for such a nonlinear boundary.

Here we propose to use Fuzzy CMAC as a nonlinear equalizer. The procedure consists of two steps.

*Step 1: Generation **of** training **data***

For Fuzzy CMAC to be an efficient nonlinear equalizer, a thorough training of various possible cases is necessary. From Fig. 4, it can be seen there are 8 possible noise free elements. We generate 10,000 data pairs around the 8 noise free elements by adding Gaussian noise to them. The data sets are shown in Fig. 5. The desired outputs of the network are either 1 or -1. The Fuzzy CMAC equalizer has two inputs and one output. The training of the network is done by evaluating the error and the error is used to adjust the weights in the Fuzzy CMAC. We used 10 membership functions for each input of the equalizer.

Step 2: Performance test

To test the performance of the nonlinear equalizer, we generate another 100,000 data pairs with various noise amplitudes. The output of the Fuzzy CMAC is compared with the desired values to check whether the decision is right or wrong. Due to the generalization property of the Fuzzy CMAC, the network can make good decision about untrained cases. The overall performance is shown in Fig. 6, which shows the bit error rate versus the signal-to-noise ratio.

4. Conclusion

A new nonlinear equalizer using Fuzzy CMAC neural network is proposed to perform equalization for nonlinear channels. The training speed of the network is an order of magnitude faster. Moreover, the human expert knowledge can be incorporated into the equalizer design. Simulation results show that the performance is similar to existing approaches.

References

[1] J. Proakis, *Digital Communications,* McGraw-Hill Book Co., New York, 1989.

[2] C.F.N. Cowan, "Performance Comparisons of Finite Linear Adaptive Filters," *IEE Proc.*, vol. 134, pt. F, no.3, pp. 211-215, 1987.

[3] L. Wang, *Adaptive Fuzzy Systems and Control,* Prentice-Hall, New Jersey, 1994

[4] E. Biglieri, A. Gersho, R.D. Gitlin, and T.L. Lim, "Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission," *IEEE J. on Selected Areas in Communications,* no.5, pp. 765-777, 1984.

[5] S. Chen, G.J. Gibson, C.F.N. Cowan, and P.M. Grant, "Adaptive equalization of finite nonlinear channels using multilayer perceptrons," *Signal Processing,* vol. 20, pp. 107-119, 1990.

[6] D.D. Falconer, "Adaptive equalization of channel nonlinearities in QAM data transmission systems," *The Bell System Technical Journal,* vol. 57, no. 7, pp. 2589-2611, 1978.

[7] Z. Geng, H. Xu, and L.S. Haynes, "Fuzzy CMAC Neural Network Application to Flight Vehicle Control," *Phase I report to U.S. Army,* November, 1994.

[8] J. Albus, "A new approach to manipulator control: The cerebellar model articulation controller (CMAC)," *Trans. ASME, Journal of Dynamic Systems, Measurement and Control,* vol. 97, pp. 220-227, 1975.

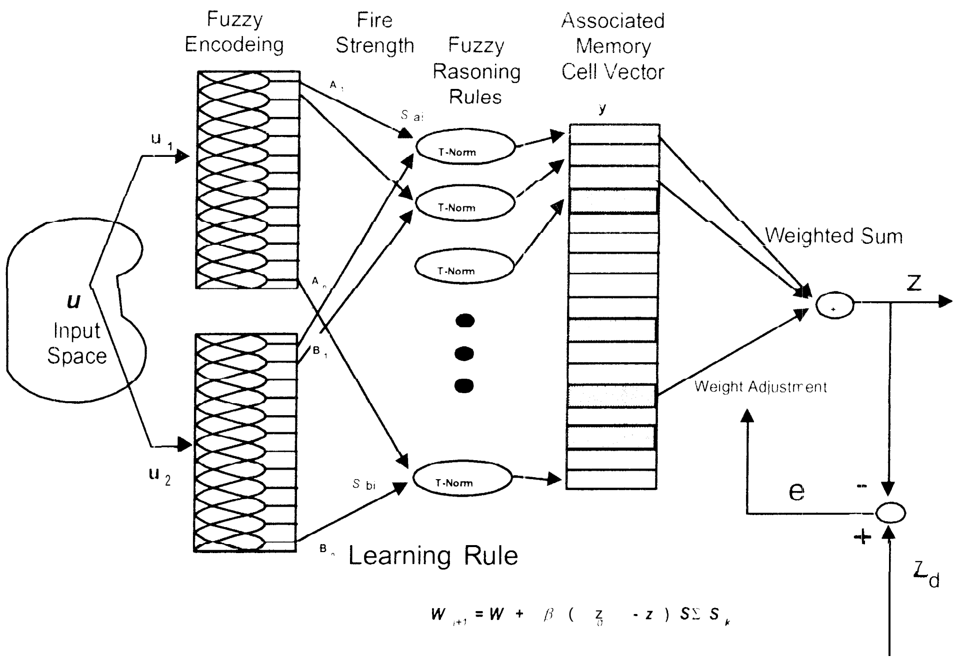

Fig. 1 Architecture of Fuzzy CMAC Neural Network

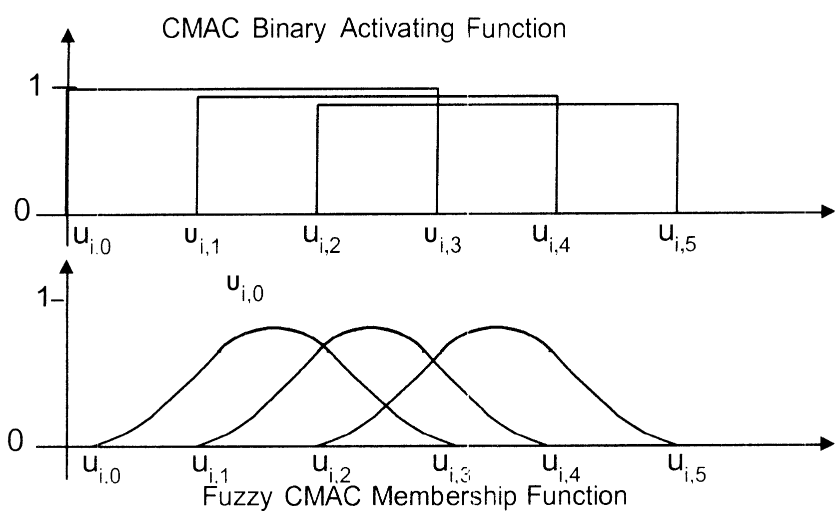

Fig. 2 Comparison of CMAC Binary Activating Function with Fuzzy CMAC Membership Function

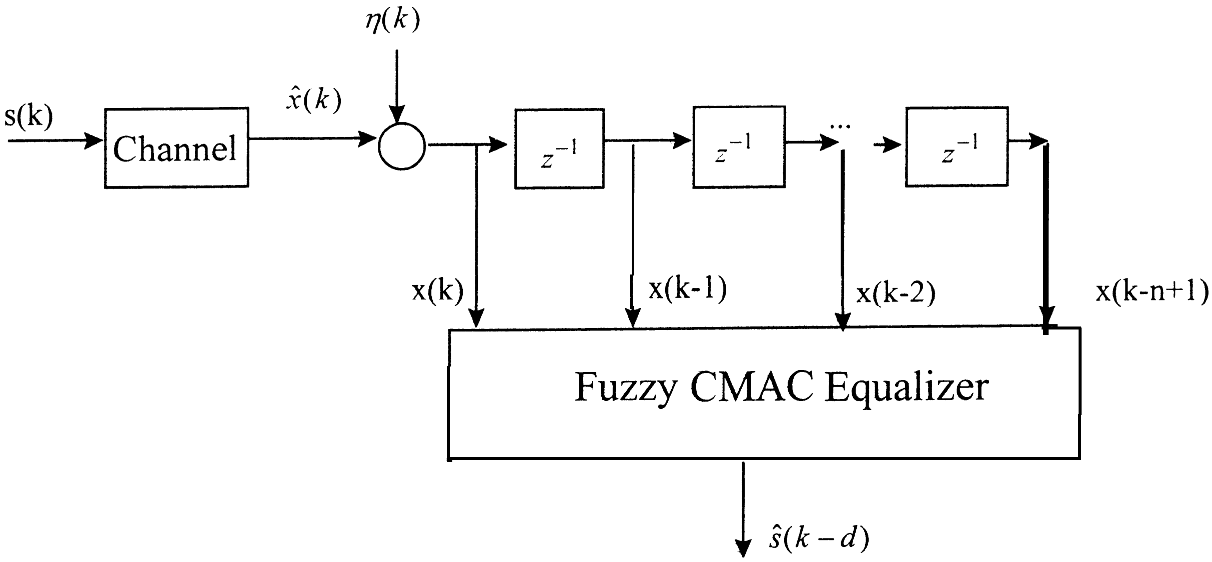

Fig. 3 Data transmission system.

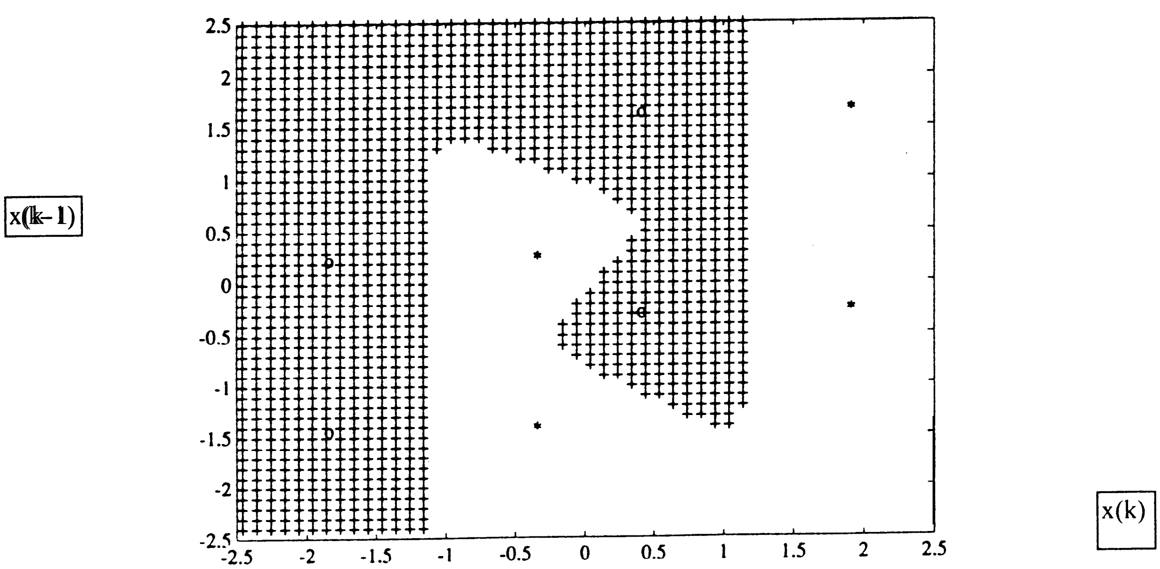

Fig. 4 Optimal decision region for channel (3.5).

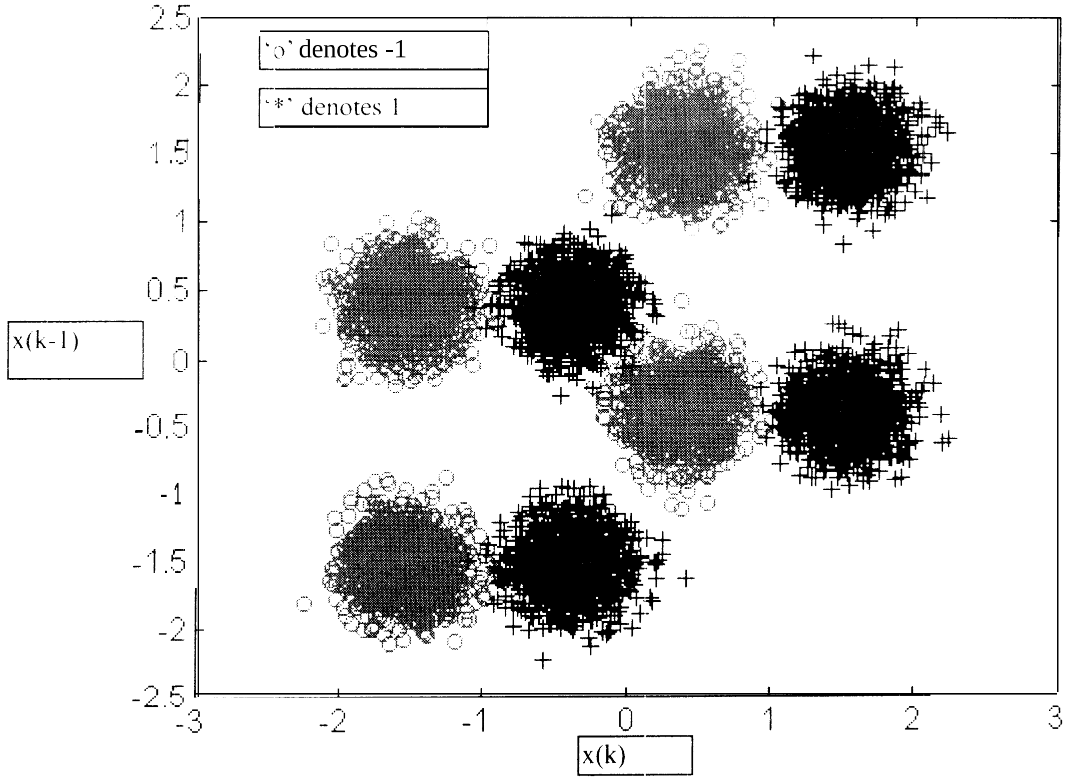

Fig. 5 Training data.

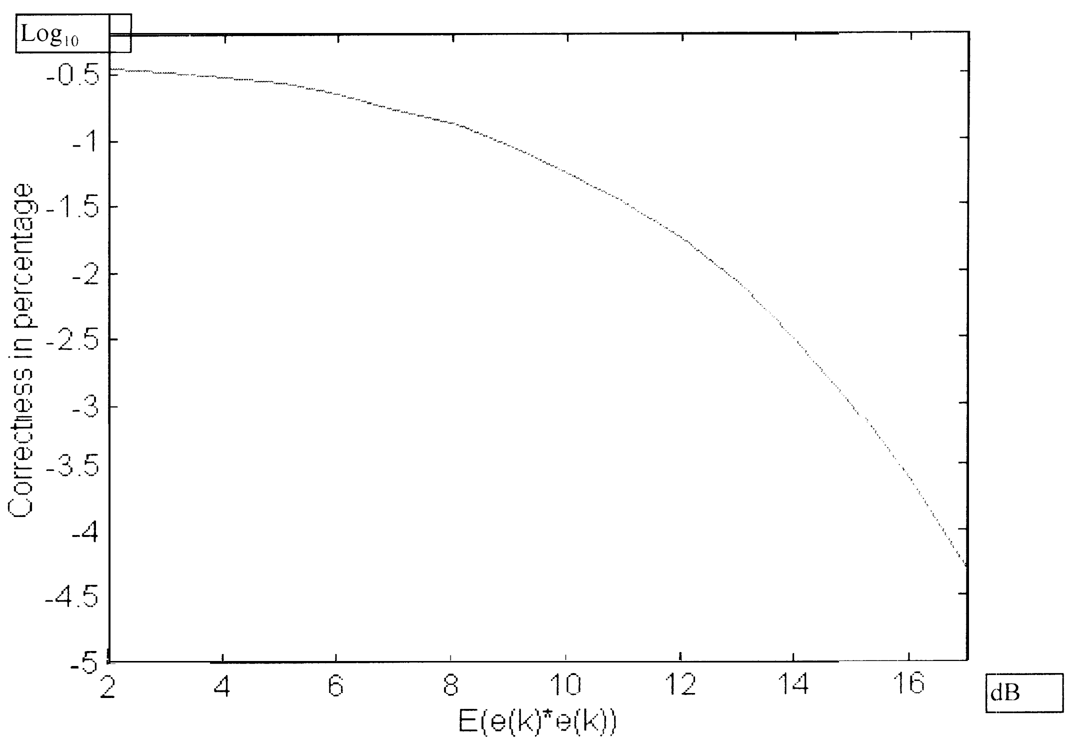

Fig. 6 Bit error rate versus signal to noise ratio.