Nonlinear Blind Channel Equalization using A Modified Fuzzy *C*-Means

Soowhan Han†, Imgeun Lee††, and Witold Pedrycz†††

†Dept. of Multimedia Engineering, Dongeui University, Pusan, 614-714 Korea
†† Dept. of Visual Information Engineering, Dongeui University, Pusan, 614-714 Korea
††† Dept. of Electrical & Computer Engineering, University of Alberta, Edmonton, T6G 2G7 Canada

Summary
In this study, a nonlinear blind channel equalization is implemented by using a Modified Fuzzy *C*-Means (MFCM) algorithm. The proposed MFCM searches the optimal channel output states of a nonlinear channel from the received symbols, based on the Bayesian likelihood fitness function instead of a conventional Euclidean distance measure. In its searching procedure, all of the possible desired channel states are constructed by the combinations of estimated channel output states. The desired state with the maximum Bayesian fitness is selected and placed at the center of a Radial Basis Function (RBF) equalizer to reconstruct transmitted symbols. In the simulations, binary signals are generated at random with Gaussian noise. The performance of the proposed method is compared with that of a hybrid genetic algorithm (GA augment by simulated annealing (SA), GASA). It is shown that a relatively high accuracy and fast search speed has been achieved.

Key words:
Nonlinear blind channel equalization, Modified Fuzzy C-Means, RBF equalizer, Channel output states

1. Introduction

In digital communication systems, data symbols are transmitted at regular intervals. Time dispersion caused by non-ideal channel frequency response characteristics, or by multipath transmission, may create inter-symbol interference (ISI). This has become a limiting factor in many communication environments. Furthermore, the nonlinear character of ISI that often arises in high speed communication channels degrades the performance of the overall communication system [1]. To overcome this detrimental ISI effects and to achieve high-speed and reliable communication, we have to resort ourselves to nonlinear channel equalization.

The conventional approach to linear or nonlinear channel equalization requires an initial training period, with a known data sequence, to learn the channel characteristics. In contrast to standard equalization methods, the so-called blind (or self-recovering) equalization methods operate without a training sequence [2]. Because of its superiority, the blind equalization method has gained practical interest during the last few years. Most of the studies carried out so far are focused on linear channel equalization [3]-[5].

Only a few papers have dealt with nonlinear channel models. The blind estimation of Volterra kernels, which characterize nonlinear channels, was presented in [6], and a maximum likelihood (ML) method implemented via expectation-maximization (EM) was introduced in [7]. The Volterra approach suffers from enormous complexity. Furthermore the ML approach requires some prior knowledge of the nonlinear channel structure to estimate the channel parameters. The approaches with nonlinear structures such as multilayer perceptrons and piecewise linear networks, being trained to minimize some cost function, have been investigated in [8] and [9], respectively. However, in those methods, the structure and complexity of the nonlinear equalizer must be specified in advance. The support vector (SV) equalizer proposed by Santamaria et al. [10] can be a possible solution for both of linear and nonlinear blind channel equalization at the same time, but it still suffers from high computational cost of its iterative reweighted quadratic programming procedure. A unique approach to nonlinear channel blind equalization was offered by Lin et al. [11], in which they used the simplex GA method to estimate the optimal channel output states instead of estimating the channel parameters directly. The desired channel states were constructed from these estimated channel output states, and placed at the center of their RBF equalizer. With this method, the complex modeling of the nonlinear channel can be avoided. Recently this approach has been implemented with a hybrid genetic algorithm (that is genetic algorithm, GA merged with simulated annealing (SA); GASA) instead of the simplex GA. The resulting better performance in terms of speed and accuracy has been reported in [12]. However, for real-time use, the estimation accuracy and convergence speed in search of the optimal channel output states needs further improvement.

In this study, we propose a new modified Fuzzy *C*-Means (MFCM) algorithm to determine the optimal output states of a nonlinear channel. The FCM algorithm

Manuscript received July 5, 2007.
Manuscript revised July 25, 2007.

introduced in [13] and being widely used in pattern recognition, system modeling, and data analysis relies on the use of some distance function. Typically, this distance is viewed as the Euclidean one. In the proposed modifications, the construction stage for the possible data set of desired channel states by the elements of estimated channel output states and the selection stage by the Bayesian likelihood fitness function are added to the conventional FCM algorithm. These two additional stages make it possible to search for the optimal output states of a nonlinear blind channel. The MFCM shows the relatively high estimation accuracy combined with fast convergence speed. Its performance is compared with the one using the GASA. In the experiments, two different nonlinear channels are evaluated. The optimal output states of each of nonlinear channels are estimated using both MFCM and GASA. Using the estimated channel output states, the desired channel states are derived and placed at the center of a RBF equalizer to reconstruct transmitted symbols. The RBF equalizer is an identical structure with the optimal Bayesian equalizer, and its important role is to place the optimal centers at the desired channel states [14].

The organization of this paper is as follows: Section 2 includes a brief introduction to the equalization of nonlinear channel using a RBF network; section 3 shows the relation between the desired channel states and the channel output states. In section 4, MFCM with a Bayesian fitness function is introduced. The simulation results, including comparisons with GASA and the conclusion, are provided in sections 5 and 6, respectively.

2. Equalization of Nonlinear Channel by a RBF Network

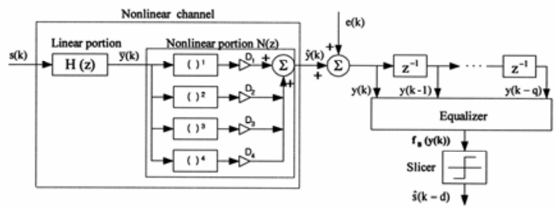


Fig. 1 The structure of a nonlinear channel equalization system.

A nonlinear channel equalization system is shown in Fig. 1. A digital sequence *s(k)* is transmitted through the nonlinear channel, which is composed of a linear portion described by *H(z)* and a nonlinear component *N(z)*, governed by the following expressions,

$$\bar{y}(k) = \sum_{i=0}^{p} h(i)s(k-i) \tag{1}$$

$$\hat{y}(k) = D_1\bar{y}(k) + D_2\bar{y}(k)^2 + D_3\bar{y}(k)^3 + D_4\bar{y}(k)^4 \tag{2}$$

where p is the channel order and D_i is the coefficient of the i^{th} nonlinear term. The transmitted symbol sequence *s(k)* is assumed to be an equiprobable and independent binary sequence taking values from $\{\pm 1\}$. We consider that the channel output is corrupted by an additive white Gaussian noise *e(k)*. Given this the channel observation *y(k)* can be written as

$$y(k) = \hat{y}(k) + e(k) \tag{3}$$

If q denotes the equalizer order (number of tap delay elements in the equalizer), then there exist $M = 2^{p+q+1}$ different input sequences

$$\boldsymbol{s(k)} = [s(k), s(k-1), \cdots, s(k-p-q)] \tag{4}$$

that may be received (where each component is either equal to 1 or –1). For a specific channel order and equalizer order, the number of input patterns that influence the equalizer is equal to M, and the input vector of equalizer without noise is

$$\boldsymbol{\hat{y}(k)} = [\hat{y}(k), \hat{y}(k-1), \cdots, \hat{y}(k-q)] \tag{5}$$

The noise-free observation vector $\boldsymbol{\hat{y}(k)}$ is referred to as the desired channel states, and can be partitioned into two sets, $\boldsymbol{Y}_{q,d}^{+1}$ and $\boldsymbol{Y}_{q,d}^{-1}$, as shown in (6) and (7), depending on the value of *s(k-d)*, where d is the desired time delay.

$$\boldsymbol{Y}_{q,d}^{+1} = \{\, \boldsymbol{\hat{y}(k)} \,|\, s(k-d) = +1 \} \tag{6}$$

$$\boldsymbol{Y}_{q,d}^{-1} = \{\, \boldsymbol{\hat{y}(k)} \,|\, s(k-d) = -1 \} \tag{7}$$

The task of the equalizer is to recover the transmitted symbols *s(k-d)* based on the observation vector ***y(k)***. Because of the additive white Gaussian noise, the observation vector ***y(k)*** is a random process having conditional Gaussian density functions centered at each of the desired channel states, and determining the value of *s(k-d)* becomes a decision problem. Therefore, Bayes decision theory [15] can be applied to derive the optimal solution for the equalizer. The solution forming the optimal Bayesian equalizer is given as follows

$$\boldsymbol{f_B}(\boldsymbol{y}(\boldsymbol{k})) = \sum_{i=1}^{n_s^{+1}} \exp\left(-\left\|\boldsymbol{y}(\boldsymbol{k}) - \boldsymbol{y}_i^{+1}\right\|^2 / 2\sigma_e^2\right) - \sum_{i=1}^{n_s^{-1}} \exp\left(-\left\|\boldsymbol{y}(\boldsymbol{k}) - \boldsymbol{y}_i^{-1}\right\|^2 / 2\sigma_e^2\right) \tag{8}$$

$$\hat{s}(k-d) = \operatorname{sgn}(\boldsymbol{f_B}(\boldsymbol{y}(\boldsymbol{k}))) = \begin{cases} +1, & \boldsymbol{f_B}(\boldsymbol{y}(\boldsymbol{k})) \geq 0 \\ -1, & \boldsymbol{f_B}(\boldsymbol{y}(\boldsymbol{k})) < 0 \end{cases} \tag{9}$$

where $\boldsymbol{y}_i^{+1}$ and $\boldsymbol{y}_i^{-1}$ are the desired channel states belonging to sets $\boldsymbol{Y}_{q,d}^{+1}$ and $\boldsymbol{Y}_{q,d}^{-1}$, respectively, and their numbers are denoted as n_s^{+1} and n_s^{-1}, and σ_e^2 is the noise variance. The desired channel states, $\boldsymbol{y}_i^{+1}$ and $\boldsymbol{y}_i^{-1}$, are derived by considering their relationship with the channel output states (as it will be explained in the next section). In this study, the optimal Bayesian decision probability (8) is implemented with the use of a RBF network. The structure of this network is shown in Fig. 2 [16], and its output is given as

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \omega_i \phi\left(\frac{\left\|\boldsymbol{x} - \boldsymbol{c}_i\right\|^2}{\rho_i}\right) \tag{10}$$

where n is the number of hidden units, c_i are the centers of the receptive fields, ρ_i is the width of the i^{th} units and ω_i is the corresponding weight. The RBF network is an ideal processing means to implement the optimal Bayesian equalizer when the nonlinear function ϕ is chosen as the exponential function $\phi(x) = e^{-x}$ and all of the widths are the same and equal to ρ, which is twice as large as the noise variance σ_e^2. For the case of equiprobable symbols, the RBF network can be simplified by setting half of the weights to 1 and the other half to -1. Thus the output of this RBF equalizer is same as the optimal Bayesian decision probability in (8).

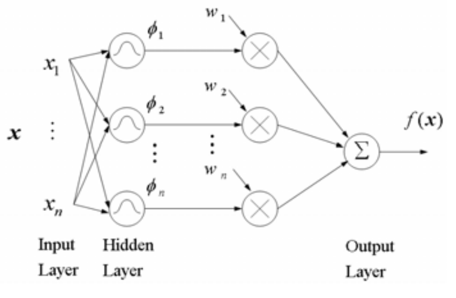


Fig. 2 The structure of a RBF network.

3. Desired Channel States and Channel Output States

The desired channel states, $\boldsymbol{y}_i^{+1}$ and $\boldsymbol{y}_i^{-1}$, are used as the centers of the hidden units in the RBF equalizer to reconstruct the transmitted symbols. If the channel order p=1 with $H(z) = 0.5 + 1.0z^{-1}$, the equalizer order q is equal to 1, the time delay d is also set to 1, and the nonlinear portion is described by $D_1 = 1, D_2 = 0.1, D_3 = 0.05, D_4 = 0.0$ (see Fig. 1), then the eight different channel states ($2^{p+q+1} = 8$) may be observed at the receiver in the noise-free case. Here the output of the equalizer should be $\hat{s}(k-1)$, as shown in Table 1. From this table, it can be seen that the desired channel states $[\hat{y}(k), \hat{y}(k-1)]$ can be constructed from the elements of the dataset, called "channel output states", $\{a_1, a_2, a_3, a_4\}$, where for this particular channel we have $a_1 = 1.89375$, $a_2 = -0.48125$, $a_3 = 0.53125$, $a_4 = -1.44375$. The length of dataset, $\tilde{n}$, is determined by the channel order, p, such as $2^{p+1} = 4$. In general, if q=1 and d=1, the desired channel states for $\boldsymbol{Y}_{1,1}^{+1}$ and $\boldsymbol{Y}_{1,1}^{-1}$ are (a_1,a_1), (a_1,a_2), (a_3,a_1), (a_3,a_2), and (a_2,a_3), (a_2,a_4), (a_4,a_3), (a_4,a_4), respectively. In the case of d=0, the channel states, (a_1,a_1), (a_1,a_2), (a_2,a_3), (a_2,a_4), belong to $\boldsymbol{Y}_{1,1}^{+1}$, and ($a_3$,$a_1$), ($a_3$,$a_2$), ($a_4$,$a_3$), ($a_4$,$a_4$) belong to $\boldsymbol{Y}_{1,1}^{-1}$. This relation is valid for the channel that has a one-to-one mapping between the channel inputs and outputs [11]. Thus the desired channel states can be derived from the channel output states if we assume p is known, and the main problem of blind equalization can be changed to focus on finding the optimal channel output states from the received patterns.

It is known that the Bayesian likelihood (BL), defined in (11), is maximized with the desired channel states derived from the optimal channel output states [17].

$$BL = \prod_{k=0}^{L-1} \max(f_B^{+1}(k), f_B^{-1}(k)) \tag{11}$$

where $f_B^{+1}(k) = \sum_{i=1}^{n_s^{+1}} \exp\left(-\left\|\boldsymbol{y}(\boldsymbol{k}) - \boldsymbol{y}_i^{+1}\right\|^2 / 2\sigma_e^2\right)$, $f_B^{-1}(k) = \sum_{i=1}^{n_s^{-1}} \exp\left(-\left\|\boldsymbol{y}(\boldsymbol{k}) - \boldsymbol{y}_i^{-1}\right\|^2 / 2\sigma_e^2\right)$ and L is the length of received sequences. Therefore, the BL is utilized as the fitness function (FF) of the proposed algorithm to find the optimal channel output states after taking the logarithm, which is shown in equation (12).

Table 1: The relation between desired channel states and channel output states

Nonlinear channel with $H(z)=0.5+1.0z^{-1}$, $D_1=1, D_2=0.1, D_3=0.05, D_4=0.0$, and d=1				
Transmitted symbols	Desired channel states			Output of equalizer
$s(k)$ $s(k-1)$ $s(k-2)$	$\hat{y}(k)$	$\hat{y}(k-1)$	By channel output states, $\{a_1, a_2, a_3, a_4\}$	$\hat{s}(k-1)$
1 1 1	1.89375	1.89375	(a_1, a_1)	1
1 1 –1	1.89375	–0.48125	(a_1, a_2)	1
–1 1 1	0.53125	1.89375	(a_3, a_1)	1
–1 1 –1	0.53125	–0.48125	(a_3, a_2)	1
1 –1 1	–0.48125	0.53125	(a_2, a_3)	–1
1 –1 –1	–0.48125	–1.44375	(a_2, a_4)	–1
–1 –1 1	–1.44375	0.53125	(a_4, a_3)	–1
–1 –1 –1	–1.44375	–1.44375	(a_4, a_4)	–1

$$FF = \sum_{k=0}^{L-1} \log(\max(f_B^{+1}(k), f_B^{-1}(k))) \qquad (12)$$

The optimal channel output states, which maximize the fitness function *FF*, cannot be obtained with the use of the conventional gradient-based methods, because the mathematical formulation between the channel output states and *FF* cannot be accomplished not knowing the channel structure [11]. For carrying out search of these optimal channel output states, a new modified version of FCM (MFCM) is developed, and its performance is compared with that of GASA introduced in [12].

4. A Modified Fuzzy C-Means Algorithm (MFCM)

In comparison with the standard version of the FCM, the proposed modification of the clustering algorithm comes with two additional stages. One of them concerns the construction stage of possible data set of desired channel states with the derived elements of channel output states. The other is the selection stage for the optimal desired channel states among them based on the Bayesian likelihood fitness function. For the channel shown in Table 1, the four elements of channel output states are required to construct the optimal desired channel states. If the candidates, $\{c_1, c_2, c_3, c_4\}$, for the elements of optimal channel output states $\{a_1, a_2, a_3, a_4\}$, are extracted from the centers of a conventional FCM algorithm, twelve (4!/2) different possible data set of desired channel states can be constructed by completing matching between $\{c_1, c_2, c_3, c_4\}$ and $\{a_1, a_2, a_3, a_4\}$. For the fast matching, the arrangements of $\{c_1, c_2, c_3, c_4\}$ are saved to the set *C* such as *C*(1)=1,2,3,4, *C*(2)=1,2,4,3, …,*C*(12)=3,2,1,4 before the search process starts. For example, *C*(2)=1,2,4,3 means the desired channel states is constructed with c_1 for a_1, c_2 for a_2, c_4 for a_3, and c_3 for a_4 in Table 1. At a next stage, a data set of desired channel states, which has a maximum Bayesian fitness value as described by (12), is selected. This data set is utilized as a center set used in the FCM algorithm. Subsequently the partition matrix *U* is updated and a new center set is sequentially derived with the use of this updated matrix *U*. The new four candidates for the elements of optimal output states are extracted from this new center set based on the relation presented in Table 1 (The eight centers in the new center set are treated as the desired channel states constructed by the elements of channel output states shown in Table 1. Thus each value of the new $\{c_1, c_2, c_3, c_4\}$ is replaced with it of the $\{a_1, a_2, a_3, a_4\}$ in the new center set, respectively). These steps are repeated until the Bayesian likelihood fitness function has not been changed or the maximum number of iteration has been reached. The proposed MFCM algorithm can be concisely described in the form of its pseudo-code, and Fig. 3 contains its flowchart.

begin
  save arrangements of candidates, $\{c_1, c_2, c_3, c_4\}$*, to C*
  randomly initialize the candidates, $\{c_1, c_2, c_3, c_4\}$
  while (new fitness function–old fitness function) <Threshold
    for k=1 to C size
      map the arrangement of candidates, C[k] ,
        to $\{a_1, a_2, a_3, a_4\}$
      construct a set of desired channel states
        based on the relation shown in Table 1
      calculate its fitness function (FF[k]) by equation (12)

end
find a data set which has a maximum FF in k=1..C
update the membership matrix U by the data set utilized as a center set in the conventional FCM algorithm
derive a new center set by the U
extract the candidates, $\{c_1, c_2, c_3, c_4\}$, *from the new center set based on the relation shown in Table 1*
end
end

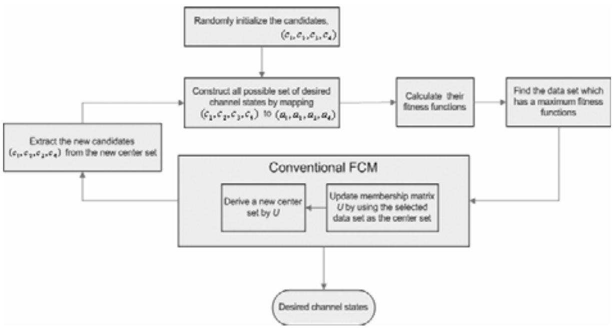


Fig. 3 The flowchart for MFCM.

In the search process carried out by the MFCM, a data set for the desired channel states which exhibits a maximum fitness value is always selected, and the candidates $\{c_1, c_2, c_3, c_4\}$ for the elements of channel output states are extracted from the data set by using the pre-established relation in Table 1. This means that the set of desired channel states produced by MFCM is always close to the optimal set and it has the same structure as shown in Table 1. Thus the centers of the first half in its output present the desired channel states for $\boldsymbol{Y}_{1,1}^{+1}$ and the rest present for $\boldsymbol{Y}_{1,1}^{-1}$, or reversely. In addition, in the pseudo-code, the MFCM checks all of the possible arrangements, *C*, to find the data set which has a maximum *FF* in *while*-loop. However, for the fast searching of the MFCM, it is not necessary to keep this work during the entire procedure. The derived new center set in the end of *while*-loop is treated as a data set for the desired channel states presented in Table 1 and each value of the new $\{c_1, c_2, c_3, c_4\}$ for next loop is replaced with each of the $\{a_1, a_2, a_3, a_4\}$ in the new center set, respectively (the new c_1 is replaced with a_1, the c_2 with a_2, the c_3 with a_3, and the c_4 with a_4). It means the new candidates, $\{c_1, c_2, c_3, c_4\}$, are always updated by using the arrangement *C*(1). Therefore, after the first couple of *while*-loops, the desired channel states constructed with the arrangement *C*(1) always has the maximum *FF* and the selected index *k* in the pseudo-code is quickly going to "1". This effect will be clearly shown in our experiments.

5. Experimental studies and performance assessments

To present the effectiveness of the proposed method, we consider blind equalization realized with GASA and MFCM. Two nonlinear channels in [11] and [18] are discussed. Channel 1 is shown in Table 1 while channel 2 is described as follows.

Channel 2:
$H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$,
$D_1 = 1, D_2 = 0.2, D_3 = 0.0, D_4 = 0.0$, and d=1

In channel 2, the channel order *p*, the equalizer order *q*, and the time delay *d* are 2, 1, 1, respectively. Thus the output of the equalizer should be $\hat{s}(k-1)$, and the sixteen desired channel states ($2^{p+q+1} = 16$) composed of the eight channel output states ($2^{p+1} = 8,\ a_1, a_2, a_3, \cdots, a_8$) may be observed at the receiver in the noise-free case. Those are shown in Table 2. The coefficients of channel 2 are symmetric, which means this channel has a linear phase characteristic. In this case, the number of observed channel output states becomes six instead of eight because a_2 and a_5, and a_4 and a_7 always have same values, 1.0219 and -0.7189 for this channel, respectively. However, in our simulations, each of all eight channel output states, $a_1, a_2, a_3, \cdots, a_8$, are searched and evaluated for more general cases. The parameters of the optimization environments for each of the algorithms are included in Table 3, and these are fixed for all experiments. The choice of these specific parameter values is not critical to the performance of GASA and MFCM. The fitness function described by (12) is utilized in both algorithms.

In the experiments, 10 independent simulations for each of two channels with five different noise levels (SNR=5,10,15,20 and 25db) are performed with 1,000 randomly generated transmitted symbols and the results are averaged. The MFCM and GASA have been implemented in a batch mode in facilitate comparative analysis. With this regard, we determine the normalized root mean squared errors (NRMSE)

$$\text{NRMSE}=\frac{1}{\|\boldsymbol{a}\|}\sqrt{\frac{1}{m}\sum_{i=1}^{m}\|\boldsymbol{a}-\hat{\boldsymbol{a}}_i\|^2} \qquad (13)$$

where $\boldsymbol{a}$ is the dataset of optimal channel output states, $\hat{\boldsymbol{a}}_i$ is the dataset of estimated channel output states, and *m* is the number of experiments performed (*m*=10).

Table 2: The desired channel states and channel output states in channel 2

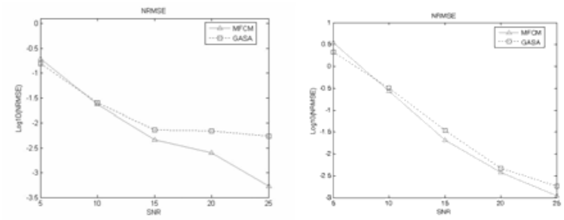
Nonlinear channel with $H(z)=0.3482+0.8704z^{-1}+0.3482z^{-2}$, $D_1=1, D_2=0.2, D_3=0.0, D_4=0.0$, and d=1				
Transmitted symbols	Desired channel states			Output of equalizer
$s(k)$ $s(k-1)$ $s(k-2)$ $s(k-3)$	$\hat{y}(k)$	$\hat{y}(k-1)$	By channel output states, $\{a_1, a_2, \cdots, a_8\}$	$\hat{s}(k-1)$
1 1 1 1	2.0578	2.0578	(a_1, a_1)	1
1 1 1 -1	2.0578	1.0219	(a_1, a_2)	1
1 1 -1 1	1.0219	-0.1679	(a_2, a_3)	1
1 1 -1 -1	1.0219	-0.7189	(a_2, a_4)	1
-1 1 1 1	1.0219	2.0578	(a_5, a_1)	1
-1 1 1 -1	1.0219	1.0219	(a_5, a_2)	1
-1 1 -1 1	0.1801	-0.1679	(a_6, a_3)	1
-1 1 -1 -1	0.1801	-0.7189	(a_6, a_4)	1
1 -1 1 1	-0.1679	1.0219	(a_3, a_5)	–1
1 -1 1 -1	-0.1679	0.1801	(a_3, a_6)	–1
1 -1 -1 1	-0.7189	-0.7189	(a_4, a_7)	–1
1 -1 -1 -1	-0.7189	-1.0758	(a_4, a_8)	–1
-1 -1 1 1	-0.7189	1.0219	(a_7, a_5)	–1
-1 -1 1 -1	-0.7189	0.1801	(a_7, a_6)	–1
-1 -1 -1 1	-1.0758	-0.7189	(a_8, a_7)	–1
-1 -1 -1 -1	-1.0758	-1.0758	(a_8, a_8)	–1

Table 3: Parameters of the optimization environments

GASA	Population size	50
	Maximum number of generation	100
	Crossover rate	0.8
	Mutation rate	0.1
	Random initial temperature	[0, 1]
	Cooling rate	0.99
MFCM	Maximum number of iteration	100
	Minimum amount of improvement	10^{-5}
	Exponent for the matrix U	2
	Random initial output states	[-1 1]

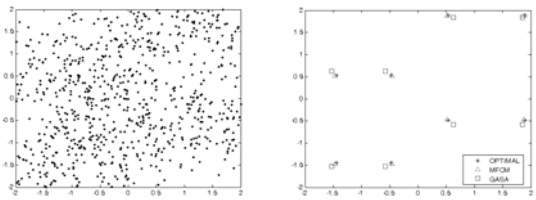
As shown in Fig. 4, the proposed MFCM comes with the lower NRMSE for both channels even the differences are not significant when dealing with the high order channel such as channel 2. Each sample of 1,000 received symbols under 5db SNR for both channels and their desired channel states constructed from the estimated channel output states by MFCM and GASA is shown in Fig. 5. In addition, we compared the search time of the algorithms. As mentioned in the end of Section 4, for the fast convergence speed of MFCM, it is not necessary to construct all of the possible data set of desired channel states in *while*-loop during the entire searching procedure, because the new candidates $\{c_1, c_2, c_3, c_4\}$ for next loop are always updated by using the matching arrangement *C*(1). The selected index *k* for the maximum *FF* is not changed after the first couple *while*-loops, and it is quickly going to "1". Each sample of variations of index *k* and the fitness function during the searching procedure for channel 1 and 2 is shown in Fig. 6. Thus the *for*-loop in the pseudo-code of MFCM, which is the construction stadge of MFCM, is skipped if the index *k* has not changed during the last 5 epochs in our experiments. The search times for MFCM and GASA are included in Table 4; Notably, the proposed

MFCM offers much higher search speed for both channels and this could be attributed to its simple structure. Finally, we investigated the bit error rates (BER) when using the RBF equalizer; refer to Table 5. It becomes apparent that the BER with the estimated channel output states realized by the MFCM is almost same as the one with the optimal output states for both channels.

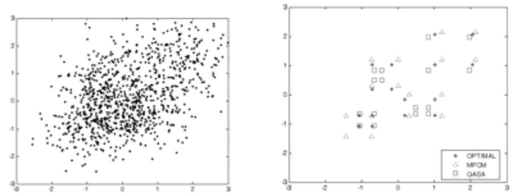


(a) channel 1 (b) channel 2

Fig. 4 NRMSE for the MFCM and GASA.

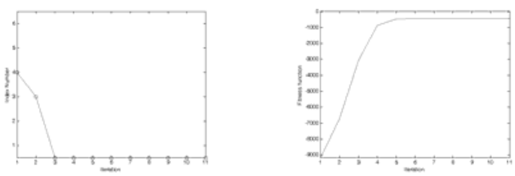


(a) channel 1

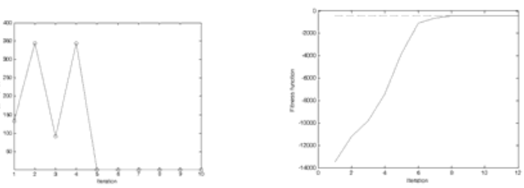


(b) channel 2

Fig. 5 Each sample of received symbols for both channels and their desired channel states produced by MFCM and GASA.



(a) channel 1: index variation(left) and its fitness function(right)



(b) channel 2: index variation(left) and its fitness function(right)

Fig. 6 Each sample of variations of index *k* during the searching procedure in MFCM.

Table 4: The averaged search time(in sec) for MFCM and GASA (Simulation environment : Pentium4 2.6Ghz, 512M Memory, code written in Matlab 6.5)

Channel	SNR	GASA	MFCM
Channel 1	5db	70.1922	0.3188
	10db	68.8266	0.2984
	15db	68.6516	0.2781
	20db	69.0344	0.3469
	25db	69.2734	0.3812
Channel 2	5db	205.0953	14.6094
	10db	200.5969	11.2969
	15db	200.8891	11.9375
	20db	203.2812	12.4188
	25db	189.3297	13.2281

Table 5: Averaged BER(no. of errors/no. of transmitted symbols).

Channel	SNR	with optimal states	GASA	MFCM
Channel 1	5db	0.0799	0.0822	0.0810
	10db	0.0128	0.0128	0.0127
	15db	0	0.0001	0.0001
	20db	0	0	0
	25db	0	0	0
Channel 2	5db	0.1161	0.1172	0.1186
	10db	0.0481	0.0493	0.0489
	15db	0.0106	0.0106	0.0106
	20db	0.0013	0.0013	0.0013
	25db	0.0007	0.0007	0.0007

6. Conclusion

In this paper, we have introduced a new modified fuzzy *c*-means clustering algorithm for nonlinear blind channel equalization. In this approach, the highly demanding modeling of an unknown nonlinear channel becomes unnecessary as the construction of the desired channel states is accomplished directly on a basis of the estimated channel output states. It has been shown that the proposed MFCM with the Bayesian likelihood treated as the fitness function offers better performance in comparison to the solution provided by the GASA approach. In particular, MFCM successively estimates the channel output states with relatively high speed and substantial accuracy. Therefore an RBF equalizer, based on MFCM, can constitute a viable solution for various problems of nonlinear blind channel equalization. Our future research pursuits are oriented towards the use of the MFCM under more complex optimization environments, such as those encountered when dealing with channels of high dimensionality and equalizers of higher order. Additionally, the way to speed up for the searching procedure of MFCM should be considered.

References

[1] Biglieri, E., Gersho, A., Gitlin, R. D., and Lim, T. L., " Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission," IEEE J. Selected Areas Commun., SAC-2(5), pp.765-777, 1984.

[2] Proakis, J. G., Digital Communications. Fourth Edition, McGraw-Hill, New York, 2001.

[3] Cao, X. R. and Zhu, J., "Blind equalization with a linear feedforward neural network," Proc. 5th European Symp. on Artificial Neural Networks, pp.267-272, Bruges, Belgium, 1997.

[4] Serpedin, E. and Giannakis, G. B., "Blind channel identification and equalization with modulation-induced cyclostationarity," IEEE Trans. Siganl Processing, vol.46, pp.1930-1944, 1998.

[5] Fang, Y., Chow, W. S., and Ng, K. T., "Linear neural network based blind equalization," Signal Processing, vol.76, pp.37-42, 1999.

[6] Stathaki, T. and Scohyers, A., "A constrained optimization approach to the blind estimation of Volterra kernels," Proc. IEEE Int. Conf. on ASSP 3, pp.2373-2376, 1997.

[7] Kaleh, G. K. and Vallet, R., "Joint parameter estimation and symbol detection for linear or nonlinear unknown channels," IEEE Trans. Commun., vol.42, pp.2406-2413, 1994.

[8] Erdogmus, D., Rende, D., Principe, J.C., and Wong, T. F., "Nonlinear channel equalization using multilayer perceptrons with information theoretic criterion," Proc. Of IEEE workshop Neural Networks and Signal Processing, pp.443-451, MA, U.S.A., 2001.

[9] Liu, X. and Adaly, T., "Canonical piecewise linear network for nonlinear filtering and its application to blind equalization," Signal Processing, vol.61, pp.145-155, 1997.

[10] Santamaria, I., Pantaleon, C., Vielva, L., and Ibanez, J., "Blind Equalization of Constant Modulus Signals Using Support Vector Machines," IEEE Trans. Signal Processing, vol.52, pp.1773-1782, 2004.

[11] Lin, H. and Yamashita, K., "Hybrid simplex genetic algorithm for blind equalization using RBF networks," Mathematics and Computers in Simulation, vol.59, pp.293-304, 2002.

[12] Han, S., Pedrycz, W., and Han, C., "Nonlinear Channel Blind Equalization Using Hybrid Genetic Algorithm with Simulated Annealing," Mathematical and Computer Modeling, vol.41, pp.697-709, 2005.

[13] Bezdek, J.C., Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York, 1981.

[14] Chen, S., Mulgrew, B., and Grant, P. M., "A Clustering Technique for Digital Communications Channel Equalization Using Radial Basis Function Networks," IEEE Trans. Neural Networks, vol.4, pp.570-579, 1993.

[15] Duda, R. O. and Hart, P. E., Pattern Classification and Scene Analysis, Wiley, NewYork, 1973.

[16] Lee, J., Beach, C., and Tepedelenlioglu, N., "Channel Equalization using Radial Basis Function Network," ICASSP vol.3, pp.1719-1722, Atlanta, Georgia, U.S.A., 1996.

[17] Lin, H. and Yamashita, K., "Blind equalization using parallel Bayesian decision feedback equalizer," Mathematics and Computers in Simulation, vol.56, pp.247-257, 2001.

[18] Patra, S.K. and Mulgrew, B., "Fuzzy techniques for adaptive nonlinear equalization," Signal Processing, vol.80, pp.985-1000, 2000.

Soowhan Han received B.S. degree in electronics, Yonsei University, Korea, in 1986, and M.S. and Ph.D. degree in Electrical & Computer Eng., Florida Institute of Technology, U.S.A. in 1990 and 1993, respectively. From 1994 to 1996, he was an assistant professor of the Dept. of Computer Eng., Kwandong University, Korea. In 1997, he joined the Dept. of Multimedia Eng., Dongeui University, Korea, where he is currently a professor. His major interests of research include digital dignal & image processing, pattern recognition and neural networks. He is a member of IEEE, KIMICS, KMS, KISPS and KFLISS.

Imgeun Lee received B.S., M.S. and Ph.D. degree in electronics engineering, Yonsei University, Korea, in 1991, 1993 and 1998, respectively. From 1998 to 2001, he worked for Daewoo Electronics, Korea, developing digital television and broadcasting equipments. He is a professor at the dept. of Visual Information Eng., Dongeui University, Korea, since 2002. His research area is including digital image processing, computer vision and digital broadcasting.

Witold Pedrycz (M'88, SM'94, F'99) is a Professor and Canada Research Chair (CRC) in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. He is also with the Systems Research Institute of the Polish Academy of Sciences, Warsaw, Poland. His active research pursuits are in Computational Intelligence, human-centric computing, fuzzy modeling, knowledge discovery and data mining, fuzzy control including fuzzy controllers, pattern recognition, knowledge-based neural networks, relational computation, bioinformatics, and Software Engineering. He has published numerous papers in this area. He is also an author of 11 research monographs covering various aspects of Computational Intelligence and Software Engineering.