Gaussian Weighted MFCM for Nonlinear Blind Channel Equalization

Soowhan Han† and Sungdae Park††

†Dept. of Multimedia Engineering, Dongeui University, Pusan, 614-714 Korea
†† Dept. of Digital Culture Content Engineering, Dongeui University, Pusan, 614-714 Korea

Summary
In this study, a modified Fuzzy *C*-Means algorithm with Gaussian weights (MFCM_GW) is presented for the problem of nonlinear blind channel equalization. The proposed algorithm searches for the optimal channel output states of a nonlinear channel based on received symbols. In contrast to conventional Euclidean distance in Fuzzy *C*-Means (FCM), the use of the Bayesian likelihood fitness function and the Gaussian weighted partiton matrix is exploited in this method. In the search procedure, all of the possible desired channel states are constructed by considering the combinations of estimated channel output states. The desired state characterized by the maximal value of the Bayesian fitness is selected and updated by using the Gaussian weights. After this procedure, the final desired state is placed at the center of a Radial Basis Function (RBF) equalizer to reconstruct transmitted symbols. The performance of the proposed method is compared with those of a simplex genetic algorithm(GA), a hybrid genetic algorithm (GA merged with simulated annealing (SA):GASA), and a previously developed version of MFCM. In particular, the relatively high accuracy and fast search speed of the method are observed.
Key words:
Modified Fuzzy *C*-Means, Gaussian Weighted Partition Matrix, Bayesian Likelihood, Nonlinear Blind Channel Equalizer,

1. Introduction

Time dispersion caused by non-ideal channel frequency response characteristics, or by multipath transmission, may create inter-symbol interference (ISI) in digital communication systems. 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, nonlinear channel equalization is necessary.

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]-[4].

Only a few papers have dealt with nonlinear channel models. The blind estimation of Volterra kernels, which characterize nonlinear channels, was presented in [5] while a maximum likelihood (ML) method implemented via expectation-maximization (EM) was introduced in [6]. In spite of their advantages, these methods are not free from limitations. The Volterra approach suffers from enormous computational complexity. Furthermore the ML approach requires some prior knowledge of the nonlinear channel structure to estimate the channel parameters. The approach with a nonlinear structure such as multilayer perceptrons, being trained to minimize some cost function, has been investigated in [7]. However, in this method, the structure and the complexity of the nonlinear equalizer must be specified in advance. The support vector (SV) equalizer proposed by Santamaria et al. [8] 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. [9], 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 a genetic algorithm, GA merged with simulated annealing (SA); GASA) [10] and a modified Fuzzy *C*-Means (MFCM) algorithm [11] instead of the simplex GA. The resulting better performances in terms of accuracy and speed have been reported. However, the estimation accuracy and the convergence speed in search of the optimal channel output states needs further improvement for the heavy noise environments such as real-time use.

In this study, a new modified Fuzzy *C*-Means algorithm

Manuscript received May 5, 2008
Manuscript revised May 20, 2008

with Gaussian weights (MFCM_GW) to determine the optimal output states of a nonlinear channel is presented. In the proposed algorithm, the Gaussian weighted partition matrix is developed and applied to the previous version of MFCM [11] for the reduction of noise effect. Thus, even the received symbols are corrupted by a heavy noise, the MFCM_GW can estimate the optimal output states with the relatively high accuracy and fast convergence speed. Its performance is compared with those of a simplex GA, a GASA and a MFCM. In the experiments, the optimal output states are estimated by each of four search algorithms. 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 [12].

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, the proposed MFCM_GW is introduced. Simulation results including comparisons with three other algorithms are provided in Section 5. Conclusions are presented in Section 6.

2. Equalization of Nonlinear Channel using a RBF Network

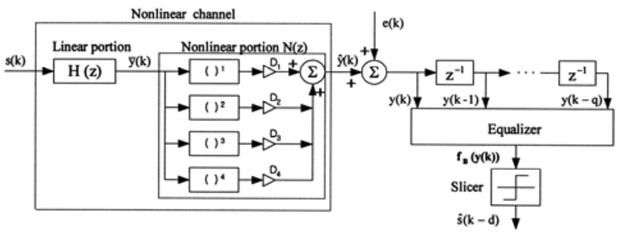


Fig. 1 The structure of a nonlinear channel equalization system.

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 a two-valued set $\{\pm 1\}$. It is assumed 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

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

The noise-free observation vector $\hat{\boldsymbol{y}}(\boldsymbol{k})$ 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.

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

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

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. The determination of the value of *s(k-d)* becomes a decision problem. Therefore Bayes decision theory [13] can be applied here to derive the optimal solution for the equalizer. The solution forming the optimal Bayesian equalizer is given as follows [14]

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

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

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 number of elements in these sets are denoted by n_s^{+1} and n_s^{-1}. Furthermore 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 [12], and its output is given as

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

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 a suitable processing structure 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 of the receptive fields 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 a half of the weights to 1 and the other half to -1. Thus the output of this RBF equalizer is the same as the optimal Bayesian decision probability in (8).

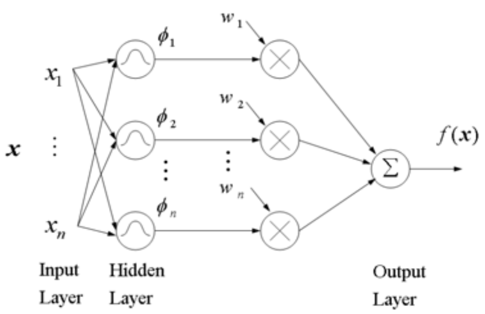


Fig. 2 The structure of a RBF network.

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 is taken as 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 $D_1 = 1, D_2 = 0.1, D_3 = -0.2, D_4 = 0.0$ (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 $a_1 = 1.0500, a_2 = -0.4500, a_3 = 0.5000, a_4 = -0.6000$. 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 [9]. 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), given by (11), is maximized with the desired channel states derived from the optimal channel output states [14].

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

where $f_B^{+1}(k) = \sum_{i=1}^{n_s^{+1}} \exp\left(-\|\boldsymbol{y(k)} - \boldsymbol{y}_i^{+1}\|^2 / 2\sigma_e^2\right)$, $f_B^{-1}(k) = \sum_{i=1}^{n_s^{-1}} \exp\left(-\|\boldsymbol{y(k)} - \boldsymbol{y}_i^{-1}\|^2 / 2\sigma_e^2\right)$ 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. Being more specific, the fitness function is taken as the logarithm of the BL, that is

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

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

Nonlinear channel with $H(z) = 0.5 + 1.0z^{-1}$, $D_1 = 1, D_2 = 0.1, D_3 = -0.2, D_4 = 0.0$, 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	1.0500	1.0500	(a_1, a_1)	1
1 1 −1	1.0500	-0.4500	(a_1, a_2)	1
−1 1 1	0.5000	1.0500	(a_3, a_1)	1
−1 1 −1	0.5000	-0.4500	(a_3, a_2)	1
1 −1 1	-0.4500	0.5000	(a_2, a_3)	−1
1 −1 −1	-0.4500	-0.6000	(a_2, a_4)	−1
−1 −1 1	-0.6000	0.5000	(a_4, a_3)	−1
−1 −1 −1	-0.6000	-0.6000	(a_4, a_4)	−1

The optimal channel output states, which maximize the fitness function *FF*, cannot be obtained with the use of the conventional gradient-based methods given the fact that the channel structure is not known in advance [9]. For carrying out search of these optimal channel output states, the proposed MFCM_GW is utilized in this study, and its performance is compared with those of three other search algorithms introduced in [9]-[11].

4. A Modified Fuzzy *C*-Means with Gaussian Weights (MFCM_GW)

The previous version of MFCM introduced in [11] comes with two additional stages in comparison with the standard Fuzzy *C*-Means [15]. 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 (or randomly initialized at first), 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 $\{a_1, a_2, a_3, a_4\}$. To facilitate fast matching, the arrangements of $\{c_1, c_2, c_3, c_4\}$ are saved to the mapping 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 are 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 the 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 $\boldsymbol{U}$ is updated and a new center set is sequentially derived with the use of this updated matrix $\boldsymbol{U}$. 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 those 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. More details about MFCM can be found in [11].

However, the performance of MFCM is easily affected by a heavy noise, because its partition matrix $\boldsymbol{U}$ and center set are updated based on Euclidean distance measure in the standard FCM algorithm. As mentioned in section 2, the received symbol $\boldsymbol{y(k)}$ is a random process having conditional Gaussian density functions centered at each of the desired channel states because of the additive white Gaussian noise. Thus to avoid this noise effect, the proposed MFCM_GW utilizes the Gaussian density function to derive the membership matrix $\boldsymbol{U}$ and a new center set such as shown in equation (13) and (14). It has a more robust characteristic to the noise than MFCM does, and it will be clearly shown in the experiments.

$$U_{ik}^{(m+1)} = \frac{\exp(-\left\| \boldsymbol{y}(\boldsymbol{k}) - \boldsymbol{y}_i^{(m)} \right\|^2 / 2\sigma_e^2)}{\sum_{j=1}^{n_s} \exp(-\left\| \boldsymbol{y}(\boldsymbol{k}) - \boldsymbol{y}_j^{(m)} \right\|^2 / 2\sigma_e^2)} \quad (13)$$

$$\boldsymbol{y}_i^{(m+1)} = \sum_{k=0}^{L-1} U_{ik}^{(m+1)} \boldsymbol{y}(\boldsymbol{k}) \quad (14)$$

where $\boldsymbol{y}_i^{(m)}$ is the estimated center set of MFCM_GW at the m^{th} iteration, utilized as the desired channel states shown in Table 1, and n_s is the total number of center vectors(n_s =8 for the channel in Table 1). The proposed MFCM_GW algorithm can be concisely described in its flowchart shown in Fig. 3.

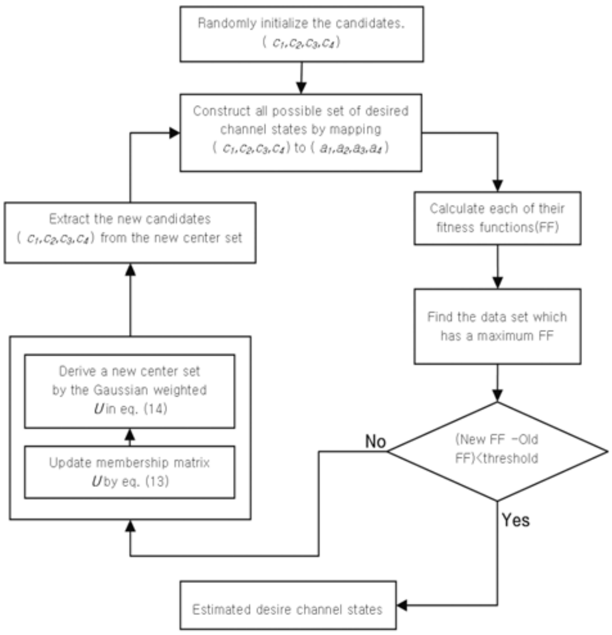


Fig. 3 The flowchart for MFCM_GW.

5. Experimental Studies

In this section, the nonlinear blind equalizations realized with the use of the simplex GA, GASA, MFCM and MFCM_GW are taken into account to demostrate the effectiveness of the proposed method. Two nonlinear channels in [9] and [16] with different channel order are discussed. Channel 1(channel order=1) is shown in Table 1 while Channel 2 is described as follows.

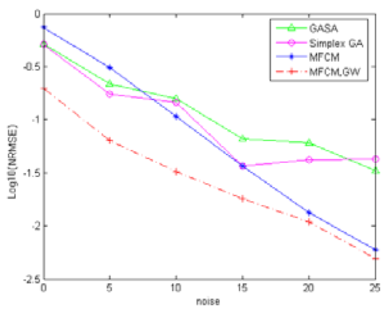
Channel 2:
$H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$,
$D_1 = 1, D_2 = 0.2, D_3 = 0.0, D_4 = 0.0$, and d=1

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. The desired channel states, (a_1,a_1), (a_1,a_2), (a_2,a_3), (a_2,a_4), (a_5,a_1), (a_5,a_2), (a_6,a_3), (a_6,a_4), belong to $\boldsymbol{Y}_{1,1}^{+1}$, and ($a_3$,$a_5$), ($a_3$,$a_6$), ($a_4$,$a_7$), ($a_4$,$a_8$), ($a_7$,$a_5$), ($a_7$,$a_6$), ($a_8$,$a_7$), ($a_8$,$a_8$) belong to $\boldsymbol{Y}_{1,1}^{-1}$, where $a_1, a_2, a_3, \cdots, a_8$ are 2.0578, 1.0219, -0.1679, -0.7189, 1.0219, 0.1801, -0.7189 and -1.0758, respectively. These sixteen desired channel states for channel 2 are summarized in [11].

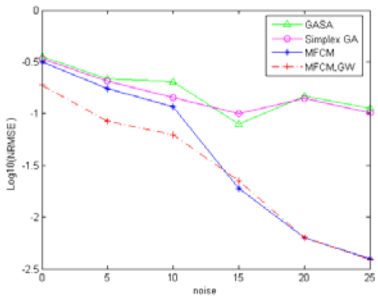
In the experiments, 10 independent simulations for each of two channels with six different noise levels (SNR=0,5,10,15,20 and 25db) are performed with 1,000 randomly generated transmitted symbols. Afterwards the obtained results are averaged. The four search algorithms, simplex GA, GASA, MFCM and MFCM_GW, have been implemented in a batch mode to facilitate comparative analysis. With this regard, we determine the normalized root mean squared errors (NRMSE)

$$\text{NRMSE} = \frac{1}{\|\boldsymbol{a}\|}\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{a} - \hat{\boldsymbol{a}}_i\right\|^2} \quad (15)$$

where $\boldsymbol{a}$ is the dataset of optimal channel output states, $\hat{\boldsymbol{a}}_i$ is the dataset of estimated channel output states, and N is the number of experiments (N=10). As shown in Fig. 4, the proposed MFCM_GW comes with the lowest NRMSE for both two channels, and the performance differences are more severe under the high noise levels such as SNR=0,5,10db. It is caused by the fact that the MFCM_GW uses the Gaussian weights shown in equation (13) and (14) to reduce the noise interference as mentioned in section 4. A sample of 1,000 received symbols under 5db SNR for channel 1, and its desired channel states constructed from the estimated channel output states by each of four search algorithms are shown in Fig. 5.

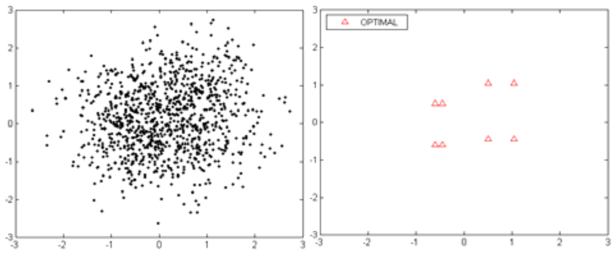


(a) channel 1

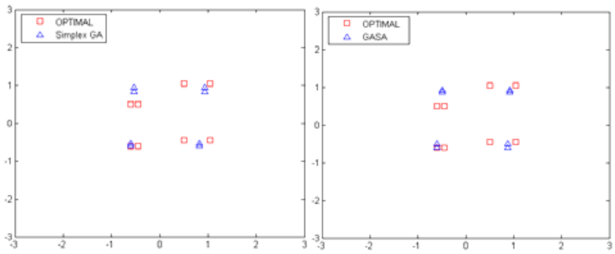


(b) channel 2

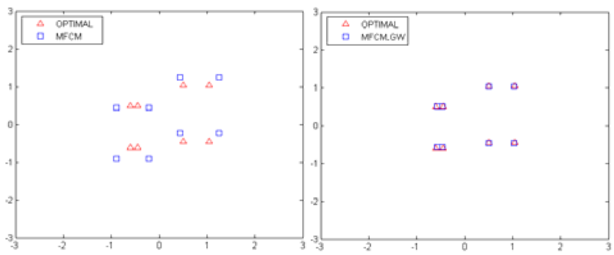
Fig. 4. NRMSE for channel 1 and 2.



(a) 1000 received symbols under 5db SNR (b) optimal desired channel states



(c) estimated by simplex GA (d) estimated by GASA



(e) estimated by MFCM (f) estimated by MFCM_GW

Fig. 5. A sample of received symbols for channel 1 and its desired channel states

In addition, we compared the search time of the algorithms. The search times for each of four algorithms are included in Table 2; notably, the MFCM and MFCM_GW offer much higher search speed for both channels and this could be attributed to their simple structures. The basic architecture of MFCM_GW is shared with the one of MFCM introduced in [11]. However, the search speed for the proposed MFCM_GW is much faster where the noise level is going up (SNR=0,5,10db) as shown in the performance of NRMSE. Finally, we investigated the bit error rates (BER) when using the RBF equalizer; refer to Table 3. It becomes apparent that the BER with the estimated channel output states realized by the MFCM_GW is almost the same as the one with the optimal output states for both channels.

Table 2. The averaged search time(in sec) for each of four algorithms.
(Simulation environment : Pentium4 2.8Ghz, 2G RAM, code written in Matlab 7.1)

Channel	SNR	Simplex GA	GASA	MFCM	MFCM_GW
Channel 1	0db	40.8609	41.2141	0.5391	0.3188
	5db	41.9953	42.2703	0.3016	0.2594
	10db	42.2578	42.1203	0.1750	0.1688
	15db	42.0000	42.1953	0.1532	0.1609
	20db	42.3781	42.3860	0.1516	0.1531
	25db	42.0125	41.8531	0.1547	0.1781
Channel 2	0db	56.8953	56.7391	2.9844	2.6000
	5db	59.2610	59.6953	3.3172	1.3187
	10db	59.9703	59.7000	2.0844	0.9750
	15db	60.6172	59.8954	1.6891	0.7063
	20db	60.2391	60.0985	3.4547	0.8922
	25db	58.3188	58.0703	1.6969	1.4969

Table 3. Averaged BER(%)(no. of errors/no. of transmitted symbols) for channel 1 and 2

Channel	SNR	with optimal states	Simplex GA	GASA	MFCM	MFCM _GW
Channel 1	0db	27.94	28.10	28.01	28.50	28.19
	5db	15.18	15.21	15.39	15.70	15.21
	10db	4.52	4.98	4.71	4.69	4.49
	15db	0.20	0.24	0.25	0.23	0.23
	20db	0.00	0.00	0.00	0.00	0.00
	25db	0.00	0.00	0.00	0.00	0.00
Channel 2	0db	21.03	21.19	21.65	21.51	21.49
	5db	11.93	12.24	12.49	11.98	11.91
	10db	4.11	4.68	4.87	4.65	4.86
	15db	1.01	1.57	1.34	1.00	1.00
	20db	0.09	2.42	2.28	0.09	0.09
	25db	0.00	1.22	1.13	0.00	0.00

6. Conclusion

In this paper, a new modified Fuzzy *C*-Means clustering algorithm with Gaussian weights for nonlinear channel blind equalization has been introduced. 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_GW offers better performance in comparison with the solutions provided by the simplex GA, GASA, and the previous version of MFCM approach. In particular, MFCM_GW successfully estimates the channel output states with relatively high speed and substantial accuracy even when the received symbols are significantly corrupted by a heavy noise. Therefore an RBF equalizer, based on MFCM_GW, 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_GW under heavy noise communication environments.

References

[1] E. Biglieri, A. Gersho, R.D. Gitlin and T.L. Lim, "Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission," IEEE J. Selected Areas Commun. SAC-2(5), pp.765-777, 1984.
[2] J.G. Proakis, Digital Communications, Fourth Edition, McGraw-Hill , New York, 2001.
[3] E. Serpedin and G.B. Giannakis, "Blind channel identification and equalization with modulation-induced cyclostationarity," IEEE Trans. Signal Processing, vol.46, pp.1930-1944, 1998.
[4] Y. Fang, W.S. Chow and K.T. Ng, "Linear neural network based blind equalization," Signal Processing, vol.76, pp 37-42, 1999.
[5] T. Stathaki and A. Scohyers, "A constrained optimization approach to the blind estimation of Volterra kernels," Proc. IEEE Int. Conf. on ASSP 3, pp.2373-2376, 1997.
[6] G.K. Kaleh and R. Vallet, "Joint parameter estimation and symbol detection for linear or nonlinear unknown channels," IEEE Trans. Commun., vol.42, pp.2406-2413, 1994.
[7] D. Erdogmus, D. Rende, J.C. Principe and T.F. Wong, "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.
[8] I. Santamaria, C. Pantaleon, L. Vielva and J. Ibanez, "Blind Equalization of Constant Modulus Signals Using Support Vector Machines," IEEE Trans. Signal Processing, vol.52, pp.1773-1782, 2004.
[9] H. Lin and K. Yamashita, "Hybrid simplex genetic algorithm for blind equalization using RBF networks," Mathematics and Computers in Simulation, vol.59, pp.293-304, 2002.
[10] S. Han, W. Pedrycz and C. Han, "Nonlinear Channel Blind Equalization Using Hybrid Genetic Algorithm with Simulated Annealing," Mathematical and Computer Modeling, vol.41, pp.697-709, 2005.
[11] S. Han, I. Lee and W. Pedrycz, "Nonlinear Blind Channel Equalization using A Modified Fuzzy *C*-Means," International Journal of Computer Science and Network Security, vol.7, pp.200-207, 2007.
[12] S. Chen, B. Mulgrew and P.M. Grant, "A Clustering Technique for Digital Communications Channel Equalization Using Radial Basis Function Networks," IEEE Trans. Neural Networks, vol.4, pp.570-579, 1993.
[13] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[14] H. Lin and K. Yamashita, "Blind equalization using parallel Bayesian decision feedback equalizer," Mathematics and Computers in Simulation , vol.56, pp.247-257, 2001.
[15] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
[16] S.K. patra and B. Mulgrew, "Fuzzy techniques for adaptive nonlinear equalization," Signal Process, 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 signal & image processing, pattern recognition and neural networks. He is a member of IEEE, KIMICS, KMS, KISPS and KFLISS.

Sungdae Park received B.S., M.S. and Ph.D. degree in Multimedia Eng., Dong-eui University, Korea, in 2002, 2005 and 2008, respectively. Since Feb. of 2008, he has joined the Dept. of Digital Content Eng., Dong-eui University, where he is currently a fulltime lecture. His research interests are digital design & nonlinear editing, neural networks, pattern recognition and image processing.