Gaussian Weighted CFCM for Blind Equalization of Linear/Nonlinear Channel

Soo-whan Han*

Abstract

The modification of conditional Fuzzy C-Means (CFCM) with Gaussian weights (CFCM_GW) is accomplished for blind equalization of channels in this paper. The proposed CFCM_GW can deal with both of linear and nonlinear channels, because it searches for the optimal desired states of an unknown channel in a direct manner, which is not dependent on the type of channel structure. In the search procedure of CFCM_GW, the Bayesian likelihood fitness function, the Gaussian weighted partition matrix and the conditional constraint are exploited. Especially, in contrast to the common Euclidean distance in conventional Fuzzy *C*-Means(FCM), the Gaussian weighted partition matrix and the conditional constraint in the proposed CFCM_GW make it more robust to the heavy noise communication environment. The selected channel states by CFCM_GW are always close to the optimal set of a channel even when the additive white Gaussian noise (AWGN) is heavily corrupted. These given channel states are utilized as the input of the Bayesian equalizer to reconstruct transmitted symbols. The simulation studies demonstrate that the performance of the proposed method is relatively superior to those of the existing conventional FCM based approaches in terms of accuracy and speed.

Keywords : Conditional Fuzzy C-Means, Gaussian weighted partition matrix, Blind Channel Equalization, Desired Channel States, Bayesian Equalizer

I. Introduction

Most of digital communication channels suffer from the inter-symbol-interference (ISI) due to non-ideal channel characteristics. The ISI will increase the symbol error rate at a receiver, sometimes preventing correct detection of a transmitted signal. The problem becomes more severe in the presence of additive white Gaussian noise (AWGN). Furthermore, the nonlinear character of ISI that often arises in high speed communication channels degrades the performance of the overall digital communication system [1]. As a result, channel equalizers are required to remove the channel distortion. Most of them take advantage of the use of known training sequences to adaptively extract channel information. The difficulty with this approach is that it consumes bandwidth. To mitigate this problem, blind-equalization algorithms have been proposed [2]-[4]. Here, instead of using training sequences, only an input signal and the knowledge of statistical properties of noise are required. The original transmitted message is recovered only from the received sequence that is corrupted by noise and ISI without any training sequence or a prior knowledge of the channel. However, because of inherent simplicity, most works for blind channel equalization deal with linear channels that are often inadequate for modeling channels which exhibit nontrivial nonlinearities [5]-[8]. This paucity does not mean blind nonlinear equalization methods exhibit less significance. Considering that nonlinear distortion exists in many communication systems such as high power amplifiers as well as high-density magnetic and optical storage channels, studying blind nonlinear system equalization methods comes with significant practical importance. Thus, in this paper, the blind equalization method, which could serve as a solution to both linear and nonlinear channels at the same time, is investigated.

Early works done on blind nonlinear channel

* Dept. of Multimedia Eng., Dongeui University
투고일자: 2013.3.27　수정완료일자: 2013.7.31
게재확정일자: 2013.8.1
※ This study was supported by Dongeui University Research Grant 2012AA199

equalization have focused on channel estimation by exploiting high order statistics (HOS) [9]-[11]. The resulting equalizers suffer from slow convergence and the fact that their optimization process could be easily trapped in local minima. The blind estimation of Volterra kernels, which characterize nonlinear channels, was presented in [12], while a maximum likelihood (ML) method implemented via expectation-maximization (EM) was introduced in [13]. Although these approaches seem to be applicable to nonlinear channels, the Volterra approach suffers from an enormous computational complexity required to construct a corresponding "inverse" Volterra filter, and the ML approach requires some prior knowledge of the nonlinear channel structure to estimate the channel parameters. The approach involving a nonlinear structure such as multilayer perceptrons being trained to minimize some cost function, has been investigated in [14]. 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. [15] can be viewed as a possible solution for both linear and nonlinear blind channel equalization, but it still suffers from high computational cost of its iterative reweighted quadratic programming procedure. The deterministic approach discussed in [16] based on second order statistic (SOS) has been successfully used to design blind equalizers of nonlinear channels, but its computational cost is also very high (requiring two matrix eigen-decompositions). Another SOS-based method provided by Raz and Van Veen [17] has limited practical application as it requires that every nonlinear sub-channel be linearizable by an FIR Volterra system. Furthermore, for this method, the sampling rate for the received signal has to be higher than the baud rate; otherwise a multi-sensor array must be utilized. This multi input multi output (MIMO) communication environment is usually required in the SOS based blind algorithms [18]. In addition, the signal to noise ratio (SNR) should be kept relatively high. A unique approach to blind channel equalization was offered by Lin and Yamashita [19]. In this method, they used the simplex Genetic Algorithm (GA) to estimate the optimal channel output states instead of estimating the channel parameters in a direct manner. The desired channel states of an unknown channel were constructed from these estimated channel output states, and placed at the center of their RBF equalizer. With this approach, the complex modeling of the nonlinear channel can be avoided and the method works well within a simple single input single output (SISO) communication environment. Additionally, this kind of approach can be applied to a linear channel as well, because it does not estimate the channel parameters but the channel output states directly, which is not dependent on the type of the channel structure. For the better performance in terms of speed and accuracy, this approach has been implemented with a hybrid genetic algorithm (that is genetic algorithm, GA merged with simulated annealing (SA); GASA) [20]. However, in general, the GA based algorithms may visibly suffer from their poor convergence properties. To overcome these weaknesses, FCM, one of the representative clustering algorithms which exhibits shorter processing time than the GA-based methods, has been investigated and the FCM-based algorithms such as Modified FCM (MFCM) [21] and MFCM with Gaussian weights (MFCM_GW) [22] have been successively developed. These FCM-based methods show the relatively faster convergence speed along with the reliable estimation accuracy in search of the optimal channel output states for blind channel equalization.

However, in the heavy noise-corrupted communication channels, the performances of MFCM and MFCM_GW are no longer superior to GA-related approaches even though the MFCM_GW shows the best estimation accuracy among them [21][22]. For real-time use, the search algorithm should be robust to intensive noise communication environments. It results in considering the use of some different types of modified version of FCM-based algorithm. A suitable modification comes in the form of a so-called Conditional FCM, or CFCM for brief. The CFCM was first introduced in [23], and successfully applied to a channel equalization problem [24][25]. In CFCM, the conditioning aspect of the clustering mechanism is introduced by taking the conditioning variable defined over the corresponding patterns. More specifically, the conditioning variable describes a level of involvement of incoming input pattern in the constructed clusters. It can be helpful to reduce the influence of heavy noise-corrupted sequences in the underlying clustering procedure. Its application to blind channel equalization problem is shown in [25] and the high performance experimental results have been achieved. In addition, the use of the Gaussian weighted partition matrix instead of Euclidean measure in MFCM_GW proves highly effective to search the optimal channel states from the heavy noise-corrupted received sequences [22]. It causes that the received symbol in a communication channel has a random process having conditional Gaussian density functions centered at each of the desired channel states because of the AWGN. Thus, in this paper, the CFCM with Gaussian weights (CFCM_GW) is developed and utilized for blind channel equalization under a heavy noise communication environment. It is accomplished by the use of Bayesian likelihood fitness function and the involvement of the relation between desired channel states and channel output states. The final clustered units of the CFCM_GW with this modification represent the desired states of the unknown channel and are utilized to compute the decision probability of Bayesian equalizer for blind equalization. Its performance is compared with the previously developed MFCM and MFCM_GW in [21] and [22], respectively. In the experiments, both of linear and nonlinear channels with the heavy noise (SNR=0, 2.5, 5, 7.5, 10db) are evaluated.

II. Bayesian Solution for Linear/Nonlinear Channel Equalization

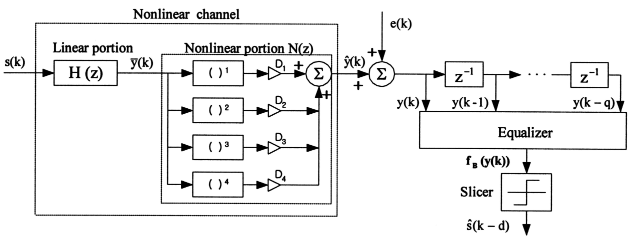


Fig. 1. An overall structure of a nonlinear channel equalization system

The channel equalization system discussed here is depicted in Fig. 1. A digital information sequence *s(k)* is transmitted through the 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 $h(\cdot)$ is the time-domain of *H(z)*, *p* is the channel order and D_i stands for the coefficient of the i^{th} nonlinear term in *N(z)*. This nonlinearity in a channel can be due to nonlinearities associated with nonlinear devices used in the transmitter and the receiver. The transmitted symbol sequence, *s(k)*, is assumed to constitute an equiprobable and independent binary sequence taking values from a two-valued set $\{\pm 1\}$. The channel output, $\hat{y}(k)$, is assumed to be corrupted by the AWGN, *e(k)*. Given this, the channel observation, *y(k)*, can be expressed as

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

If q denotes the equalizer order (viz. a number of tap delay elements in the equalizer), then there exist $M = 2^{p+q+1}$ different input sequences that may be received (where each component in (4) is either equal to 1 or -1).

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

For a specific channel order and equalizer order, these M input patterns influence the input vector of equalizer, which is shown in (5) for a noise-free case.

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

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.

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

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

In case of a linear channel (D_1*=1,* D_2*=0,* D_3*=0 and* D_4*=0*), $\hat{\boldsymbol{y}}(\boldsymbol{k})$ in (3), (5), (6) and (7) is just replaced with $\boldsymbol{y(k)}$ in (1). The task of the equalizer is to recover the transmitted symbols, *s(k-d)*, based on the observation vector, ***y(k)***. Because of the AWGN, the observation vector is a random process having conditional Gaussian density functions centered at each of the desired channel states, $\hat{\boldsymbol{y}}(\boldsymbol{k})$. The determination of the value of *s(k-d)* becomes a decision problem. Bayes decision theory [26] provides the optimal solution to the general decision problem. It is applied here and the optimal decision function for the Bayesian equalizer can be represented as follows, see [27][28]

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

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

where y_i^{+1} and y_i^{-1} are the desired channel states belonging to sets $Y_{q,d}^{+1}$ and $Y_{q,d}^{-1}$, 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 optimal equalizer solution in (8) depends on the desired channel states. In other words, the solution of blind channel equalization crucially depends on how to find the desired channel states, y_i^{+1} and y_i^{-1}, only from the observation vector *y(k)*. In this study, the new search algorithm, called "CFCM_GW", is derived and investigated in search of the optimal output states of an unknown channel, and its desired channel states are configured with the searched channel output states. The construction of desired channel states by using the relation with channel output states will be explained in the next section. The optimal Bayesian decision probability in (8) is used to derive the fitness function of proposed CFCM_GW, and also utilized as an equalizer, along with (9), for the reconstruction of the transmitted symbols.

III. Desired Channel State by Channel Output State

In the previous section, it has been observed that the knowledge of the desired channel states is essential for the evaluation of the optimal decision function in the Bayesian equalizer. The estimation of channel states requires the knowledge of the channel. However, under most circumstances, it may not be available. Additionally, the estimation of channels for nonlinear channels is very difficult in a direct manner. Thus, in the proposed algorithm, the estimation of desired channel states is accomplished by using the scalar channel states called "channel output states". The determination of these channel output states is simple and its computational complexity is independent from the equalizer order. Once the desired channel states have been constructed by using the estimated channel output states, finding the decision function of the Bayesian equalizer is straightforward.

The relationship of desired channel states and channel output states is illustrated by using the sample channel in Table 1. 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 $D_1 = 1, D_2 = 0.0, D_3 = -0.9, D_4 = 0.0$ (see Fig. 1), then the eight different channel states ($2^{p+q+1} = 8$) may be observed at the receiver in a 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 $[\hat{y}(k), \hat{y}(k-1)]$ are composed of the elements of the channel output states, $\{a_1, a_2, a_3, a_4\}$, where for this particular channel, $a_1 = 1.5375, a_2 = 0.3875, a_3 = -0.3875$ and $a_4 = -1.5375$ are observed. The length of dataset, $\tilde{n}$, is determined by the channel order, p, such as $2^{p+1} = 4$. It is independent from the equalizer order. In general, if q=1 and d=1, the desired channel states for $Y_{1,1}^{+1}$ and $Y_{1,1}^{-1}$ 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. A change in the decision delay only changes some of the positive states to negative states and equal number of negative states to positive states. For example, in 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 $Y_{1,1}^{+1}$, and (a_3,a_1), (a_3,a_2), (a_4,a_3), (a_4,a_4) belong to $Y_{1,1}^{-1}$. This relation is always valid for the channel that has a one-to-one mapping between the channel inputs and outputs [19]. Thus, the desired channel states can be derived from the channel output states if the channel order, p, is assumed to be known, and the main problem of blind equalization can be changed to focus on the determination of the optimal channel output states from the received patterns.

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

Nonlinear channel with $H(z) = 0.5 + 1.0z^{-1}$, $D_1 = 1, D_2 = 0.0, D_3 = -0.9, D_4 = 0.0$ and d=1						
Transmitted symbols			Desired channel states			Equalizer output
$s(k)$	$s(k-1)$	$s(k-2)$	$\hat{y}(k)$	$\hat{y}(k-1)$	By channel output states, $\{a_1, a_2, a_3, a_4\}$	$\hat{s}(k-1)$
1	1	1	1.5375	1.5375	(a_1, a_1)	1
1	1	−1	1.5375	0.3875	(a_1, a_2)	1
−1	1	1	−0.3875	1.5375	(a_3, a_1)	1
−1	1	−1	−0.3875	0.3875	(a_3, a_2)	1
1	−1	1	0.3875	−0.3875	(a_2, a_3)	−1
1	−1	−1	0.3875	−1.5375	(a_2, a_4)	−1
−1	−1	1	−1.5375	−0.3875	(a_4, a_3)	−1
−1	−1	−1	−1.5375	−1.5375	(a_4, a_4)	−1

IV. Construction of Fitness Function

In order to find the optimal channel states, the Bayesian likelihood (*BL*) is considered. Since the Bayesian decision variable is a probability density variable, similar to the conventional likelihood, the *BL* can be defined by (10). It is known that the *BL* in (10) is always maximized with respect to the desired channel states derived from the optimal channel output states [29].

$$BL = \prod_{k=0}^{L-1} \max(f_B^{+1}(k), f_B^{-1}(k)) \tag{10}$$

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

$f_B^{-1}(k) = \sum_{i=1}^{n_s^{-1}} \exp(-\|\boldsymbol{y}(\boldsymbol{k}) - \boldsymbol{y}_i^{-1}\|^2 / 2\sigma_e^2)$ 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 (*FF*) is taken as the logarithm of the *BL*, that is

$$FF = \sum_{k=0}^{L-1} \log(\max(f_B^{+1}(k), f_B^{-1}(k))) \tag{11}$$

The determination of the maximum *FF* is not possible without the knowledge of channel structure [19]. In addition, from the relation between *FF* and channel output states shown in Fig. 2 (where several local maxima exist), it cannot be easily solved by conventional gradient-based methods. That is one of the reasons a clustering algorithm is considered as a way to find the maximum *FF*.

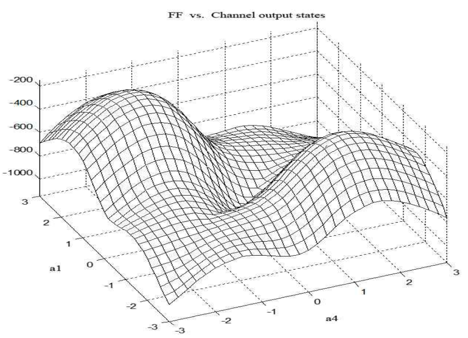


Fig. 2. *FF* vs. channel output states (a_1 and a_4) for the channel in Table1 (a_2 and a_3 are set to their optimal values (0.3875 and -0.3875), respectively)

V. CFCM with Gaussian Weights for Optimal Channel States

Before introducing the proposed CFCM_GW to be used to search the optimal channel states for blind channel equalization, the previously developed version of MFCM presented in [21] should be investigated. This is also justified by the fact that these two algorithms exhibit the same structure.

In comparison with the standard version of the Fuzzy C-Means (FCM) presented in [30], the MFCM comes with two additional stages. One of them concerns the construction stage of all possible data set of desired channel states with the estimated 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 shown by (11). For the channel shown in Table 1, the four elements ($2^{p+1} = 4$) of channel output states, $\{a_1, a_2, a_3, a_4\}$, are required to construct the optimal desired channel states. If the candidates for these elements, $\{c_1, c_2, c_3, c_4\}$, are randomly initialized, twelve(4!/2) different possible data sets 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 as a certain mapping set C such that $C(1)=1,2,3,4$, $C(2)=1,2,4,3$, ⋯, $C(12)=3,2,1,4$ before the search process starts. For example, the notation $C(2)=1,2,4,3$ means that the set of 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. The desired channel states for this set are described as $\boldsymbol{y}_{i\_C(2)}$ ($\boldsymbol{y}_{i\_C(2)}^{+1}$ and $\boldsymbol{y}_{i\_C(2)}^{-1}$ for sets $r_{1,1}$ and $r_{1,1}^{*}$, respectively), and its fitness function in (11) is presented by $FF(2)$. As mentioned at the beginning of Section 4, if the set of desired channel states by a combination $C(2)$ is optimal, it has a maximum value [29]. Thus at the next stage, a data set of desired channel states, which has a maximum Bayesian fitness value, is selected as shown below

$$[index\_j,\ max\_FF] = \max(FF(1), FF(2), \cdots, FF(12)) \tag{12}$$

This data set ($\boldsymbol{y}_{i\_C(index\_j)}$), which is the set of desired channel states configured by the selected $C(index\_j)$, is utilized as a center set in the conventional FCM algorithm. Subsequently the partition matrix U is updated and a new center set, y_i, is sequentially derived with the use of this updated matrix U. These are expressed as

$$U_{ik}^{(m+1)} = \frac{1}{\sum_{l=1}^{n_s}\left(\frac{\left\| y(k) - y_{i\_C(index\_j)}^{(m)} \right\|}{\left\| y(k) - y_{l\_C(index\_j)}^{(m)} \right\|}\right)^2} \qquad (13)$$

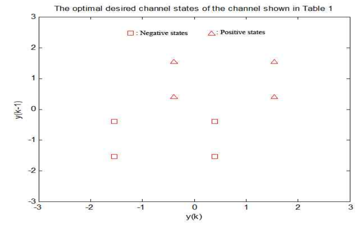
$$y_i^{(m+1)} = \frac{\sum_{k=0}^{L-1}\left(U_{ik}^{(m+1)}\right)^2 y(k)}{\sum_{k=0}^{L-1}\left(U_{ik}^{(m+1)}\right)^2} \qquad (14)$$

where $y_i^{(m+1)}$ is the estimated center set at the $(m+1)^{th}$ iteration and n_s is the total number of center vectors (n_s=8 for the channel in Table 1). In the next, the new four candidates for the elements of optimal output states are extracted from this new center set, $y_i^{(m+1)}$, based on the relation presented in Table 1. The eight centers in the new center set, $y_i^{(m+1)}$, are treated as the desired channel states constructed by the elements of channel output states, $\{a_1, a_2, a_3, a_4\}$, shown in Table 1, and thus each value of the new $\{c_1, c_2, c_3, c_4\}$ is replaced with each one of the $\{a_1, a_2, a_3, a_4\}$ in the new center set as in (15), respectively.

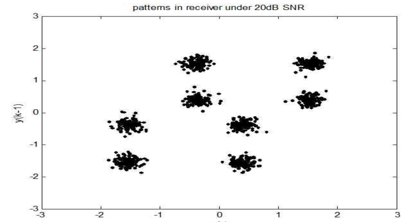
$$c_r^{(m+1)} = a_r \text{ in } y_i^{(m+1)} \text{ where } r=1,2,3,4 \qquad (15)$$

With this new set of candidates, the steps are repeated again until the Bayesian likelihood fitness function is not changed or the maximum number of iteration has been achieved. More details about MFCM can be found in [21].

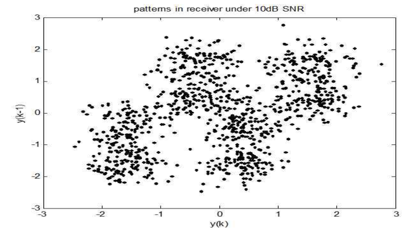
The MFCM illustrated above showed the better performance than the existing hybrid GA algorithm in terms of speed and estimation accuracy, however, at low SNRs, the differences of accuracy for both algorithms are not significant [21]. As mentioned in Section 2, the received symbol, $y(k)$, is a random process having conditional Gaussian density functions centered at each of the desired channel states because of the use of the AWGN. Thus, under low SNRs, the noise variance σ_e^2 is high and the received patterns are quite scattered such as shown in Fig. 3, which makes it difficult for the MFCM to estimate their correct centers.



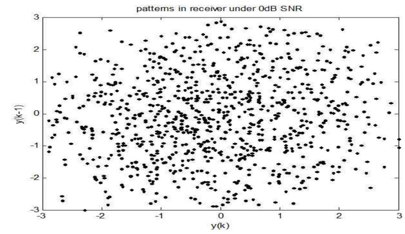
(a) optimal desired states(noise-free)



(b) received patterns under 20dB SNR



(c) received patterns under 10dB SNR



(d) received patterns under 0dB SNR

Fig. 3. The optimal desired states (noise-free) for the channel shown in Table 1(a), the received patterns under 20 dB SNR(b), 10dB SNR(c) and 0dB SNR(d).

This weakness of MFCM to significant level of noise can be overcome by applying the different weights to each of received patterns, which depend on their distances to the constructed clusters. To be more specific, the closer the received pattern to the clusters, the higher weight is attached and consequently more influential it becomes in the clustering process. This can be accomplished by using the clustering procedure of the CFCM. In the CFCM, the conditioning aspect of the clustering mechanism is introduced by taking into consideration the conditioning variable assuming values, $f_1, f_2, \ldots, f_k$ on the corresponding patterns [23]. Here

f_k taking values in the unit interval describes a level of involvement of received symbol, *y(k)*. For example, if $f_i = 0$, the i^{th} received pattern is regarded as meaningless in the clustering procedure and the calculations of the resulting prototypes are not affected by this element. Subsequently, the calculations of the partition matrix U do not take this into consideration. On the other hand, the pattern for which $f_i = 1$ contributes to the clustering process to the highest extent. The membership degree in CFCM is described as follows

$$U_{ik}^{(m+1)} = \frac{f_k}{\sum_{l=1}^{n_s}\left(\frac{\left\|\boldsymbol{y}(\boldsymbol{k}) - \boldsymbol{y}_{i\_C(index\_j)}^{(m)}\right\|}{\left\|\boldsymbol{y}(\boldsymbol{k}) - \boldsymbol{y}_{l\_C(index\_j)}^{(m)}\right\|}\right)^2} \qquad (16)$$

For the application to search the optimal channel states from the noise-corrupted received patterns, the new conditional constraint f_k in (16) should contain the distance information of each of received patterns, and it has a high value if the corresponding pattern is closely located at the estimated center. It is known that the Bayesian likelihood (*BL*) by (10) is always maximized with respect to the optimal desired channel states and utilized as the fitness function of proposed algorithm by (11). In addition, for the calculation of *BL*, if a received pattern is located near the optimal desired channel states, $\boldsymbol{y}_i^{+1}$ or $\boldsymbol{y}_i^{-1}$, this pattern produces a higher value of $f_B^{+1}(k)$ or $f_B^{-1}(k)$ in (10), and the *BL* becomes to be larger. Therefore each component of *BL* for the received patterns is utilized as the conditional constraint f_k (after normalization). The computational details are described as follows

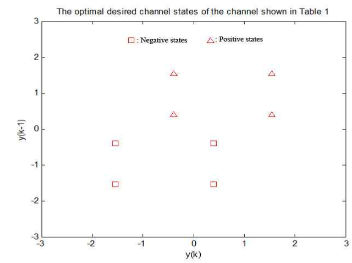
$$nf_k = \max(f_B^{+1}(k), f_B^{-1}(k)) \qquad (17)$$

$$f_k = nf_k / \max(nf_k) \qquad (18)$$

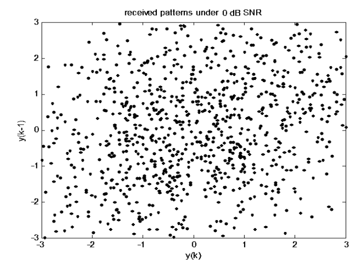
However, the partition matrix U in (16) is updated based on Euclidean distance measure even though it has the conditional constraint f_k which represents the Gaussian probability density of received pattern. It causes the clustering process with (16) is easily affected by the heavy AWGN. In the previous work of MFCM_GW [22], the use of the Gaussian weighted partition matrix instead of Euclidean measure was highly effective to the heavy noise. Therefore, in the proposed algorithm, to reduce the noise effect, the Gaussian density function along with the conditional constraint f_k is utilized as shown in (19) to update the partition matrix U, and a new center set $\boldsymbol{y}_i$ is sequentially derived by (20). The effectiveness of the proposed conditional constraint f_k and the Gaussian weighted partition matrix U under a heavy noise environment is demonstrated in Fig. 4.

$$U_{ik}^{(m+1)} = \frac{f_k \times exp(-\left\|\boldsymbol{y(k)} - \boldsymbol{y}_{i\_C(index\_j)}^{(m)}\right\|^2 / 2\sigma_e^2)}{\sum_{l=1}^{n_s} exp(-\left\|\boldsymbol{y(k)} - \boldsymbol{y}_{l\_C(index\_j)}^{(m)}\right\|^2 / 2\sigma_e^2)} \qquad (19)$$

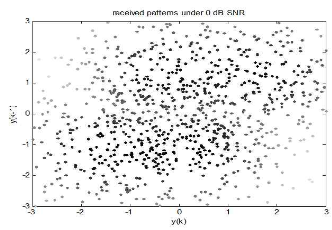
$$\boldsymbol{y}_i^{(m+1)} = \sum_{k=0}^{L-1} U_{ik}^{(m+1)}\, \boldsymbol{y(k)} \qquad (20)$$



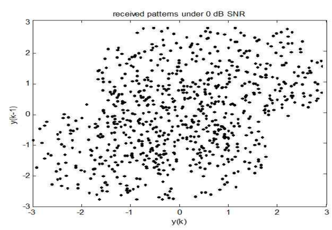
(a) optimal desired channel states



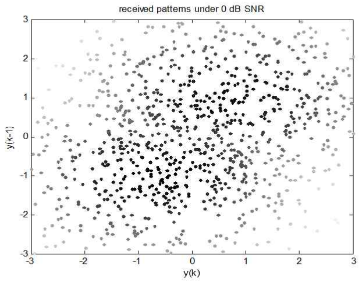
(b) received patterns under 0 dB SNR



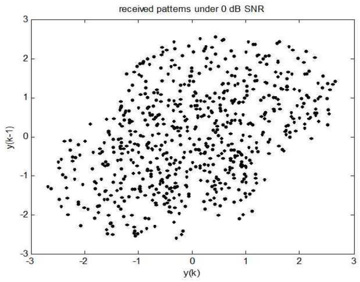
(c) patterns by f_k : 1(black) ↔ 0(white)



(d) patterns only for $f_k > 0.5$



(e) patterns by f_k : 1(black) ↔ 0(white)



(f) patterns only for $f_k > 0.5$

Fig. 4. The optimal desired states for the channel shown in Table 1(a), the received patterns under 0 dB SNR(b), and received patterns displayed by their conditional constraint f_k without Gaussian weights (c)(d), and with Gaussian weights (proposed CFCM_GW) (e)(f).

The values of the conditional constraint f_k after 10 epochs of clustering procedure without (U by (16)) and with Gaussian weights (U by (19)) are displayed in Fig. 4. In both cases, they are relatively very low for noisy patterns (close to "0" indicated by the bright color in Fig. 4(c) and (e)). On the other hand, the received patterns located near the optimal channel states are more weighted by the conditional constraint f_k (close to "1", black color in Fig. 4(c) and (e)) and generated a higher contribution to the clustering procedure. In addition, the patterns with the high values of f_k (black color) in Fig. 4(c) and (d) are spread wider than in Fig. 4(e) and (f). In other words, in Fig. 4(e) and (f), these patterns are more densely located near the optimal centers. It means that, by the clustering procedure with Gaussian weights, the closer located patterns near the optimal states have the more relatively high values of f_k. Therefore, in the proposed algorithm, the Gaussian weighted partition matrix along with the conditional constraint f_k, shown in (19), is exploited instead of (13) or (16) and a new center set y_i is derived by (20). The resulting estimation accuracy is increased even with low SNRs as it will be shown in the next section. The proposed search algorithm is summarized in the following pseudo-code.

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 j=1 to C size
      map the arrangement of candidates, C[j], 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[j]) by equation (11)
    end
    find a data set which has a maximum FF in j=1..C size
      : equation (12)
    find the conditional constraint for the selected data set
      : equations (17) & (18)
    update the membership matrix U *by the data set*
      utilized as a center set : equation (19)
    derive a new center set by the U*: equation (20)*
    extract the candidates, $\{c_1, c_2, c_3, c_4\}$*, from the new center set*
      based on the relation shown in Table 1 and C(1)
      : equation (15)
end

In the proposed search algorithm, all possible sets of desired channel states are constructed with the candidates by using the structure shown in Table 1 and a data set which exhibits a maximum fitness value is always selected. Therefore, the set of desired channel states produced by the proposed CFCM_GW is always close to the optimal set, and its first half presents the desired channel states for $r_{1,1}$ and the rest presents for $r_{1,1}^*$, or reversely. In addition, as the fast searching procedures of MFCM in [21] and MFCM_GW in [22], the proposed CFCM_GW does not need to check all of the possible arrangements, $C(1), C(2), \cdots, C(12)$, to find the data set which has a maximum *FF* after the first couple of *while*-loop. It is because the new candidates, $\{c_1, c_2, c_3, c_4\}$, are extracted by using the arrangement $C(1)$ as shown in (15) at the end of *while*-loop and thus the set of desired channel states constructed by $C(1)$ always has the maximum *FF* after a couple of clustering epochs. Therefore, in the experiments, for the fast searching of proposed CFCM_GW, the *for*-loop in the pseudo-code is skipped if the selected *index_j* has not been changed during the last 5 epochs. From this moment, the set of desired channel states only by *C(1)* is constructed with the new candidates and utilized for a further process.

VI. Experimental Results and Performance Assessment

In this section, the proposed CFCM_GW is compared and evaluated vis-a-vis the previously developed algorithms which also estimate the optimal channel states of unknown channel to solve the problem of blind equalization. As mentioned in the introduction, the MFCM and the MFCM_GW presented in [21] and [22], respectively, showed better performance than the simplex GA [19] and the GASA [20] in terms of speed and accuracy. To demonstrate the effectiveness of the method, blind equalizations realized with the use of the MFCM, the MFCM_GW and the proposed CFCM_GW are considered in the experiments. Three channels including one linear model are discussed. Channel 1, shown in Table 1, and channel 2 stand for a nonlinear model where channel 3 concerns a linear model (here the nonlinear terms in channel have been removed). These channels were discussed in [19]-[22]. The detailed description of the channels is presented below.

Channel 2 (nonlinear):

$H(z) = 0.5 + 1.0z^{-1}$,

$D_1 = 1, D_2 = 0.1, D_3 = -0.2, D_4 = 0$, and d=1

Channel 3 (linear):

$H(z) = 0.5 + 1.0z^{-1}$,

$D_1 = 1, D_2 = 0, D_3 = 0, D_4 = 0$, and d=1

In the experiments, 10 independent simulations for each of three channels with five different noise levels (SNR=0, 2.5, 5, 7.5, and 10dB) were performed with 1,000 randomly generated transmitted symbols (L=1000). Afterwards, the obtained results were averaged. The MFCM, the MFCM_GW and the proposed CFCM_GW have been implemented in a batch mode to facilitate comparative analysis. In addition, they are evaluated with the use of same parameters shown in Table 2, and these are fixed for all experiments. The choice of the specific parameter values is not critical to the performance of MFCM, MFCM_GW and CFCM_GW as well. The fitness function described by (11) is utilized in all three algorithms. With this regard, the normalized root mean squared errors (NRMSE) is determined in the form

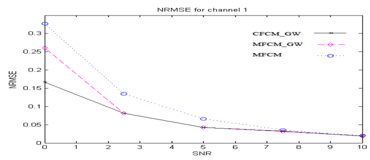
$$\text{NRMSE}= \frac{1}{\|\boldsymbol{a}\|}\sqrt{\frac{1}{N}\sum_{\boldsymbol{i}=1}^{N}\|\boldsymbol{a}-\hat{\boldsymbol{a}}_{\boldsymbol{i}}\|^2} \qquad (21)$$

where $\boldsymbol{a}$ is the data set of optimal channel output states, $\hat{\boldsymbol{a}}_{\boldsymbol{i}}$ is the data set of estimated channel output states in the i^{th} simulation, and N is the total number of independent simulations (N=10).

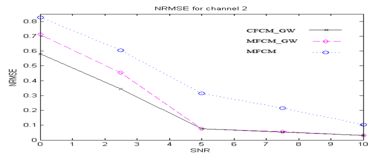
Table 2. Parameters used in the algorithms.

	MFCM	MFCM_GW	Proposed CFCM_GW
Maximum number of iteration	100	100	100
Threshold for *FF* variation	10^{-3}	10^{-3}	10^{-3}
Exponent for the partition matrix U	2	1	1
Random initial channel output states	[-0.5 0.5]	[-0.5 0.5]	[-0.5 0.5]

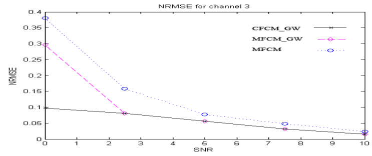
The values of NRMSEs after 10 independent simulations for each of three channels are averaged and illustrated in Fig. 5. The proposed CFCM_GW comes with lower NRMSE for all three channels, and the performance differences are more severe in higher noise levels (0dB and 2.5dB). It is caused by the fact that the CFCM_GW uses the conditional constraint f_k with Gaussian weights shown in (19) and (20) to reduce the noise interference as mentioned in section 5. As shown in Fig. 4(e), with a low SNR, the received patterns are widely distributed and the values of conditional constraint for each of them are quite different depending on their distances to the estimated centers. The patterns, which are more distant from their centers, have lower conditional constraints and are less weighted on the clustering procedure. On the contrary, the received patterns, which are located near their centers, are applied with higher weights and they become more influential in the clustering process. In addition, the use of Gaussian probability instead of simple Euclidean distance measure in the partition matrix U by (19) makes the difference of those patterns more seriously. This is why the proposed CFCM_GW is highly effective to find the optimal channel states when the received patterns are heavily corrupted by noise. A sample of 1,000 received symbols under 0dB SNR for channel 2 and its desired channel states constructed from the estimated channel output states by the MFCM, the MFCM_GW and the proposed CFCM_GW are illustrated in Fig. 6.



(a) for channel 1

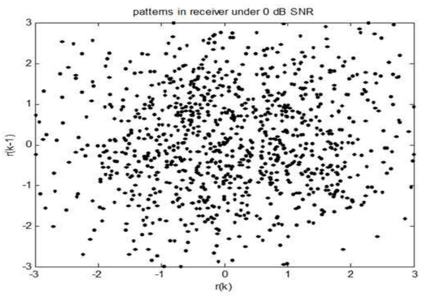


(b) for channel 2

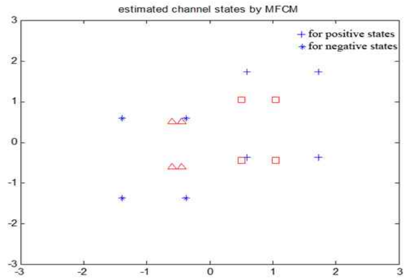


(c) for channel 3

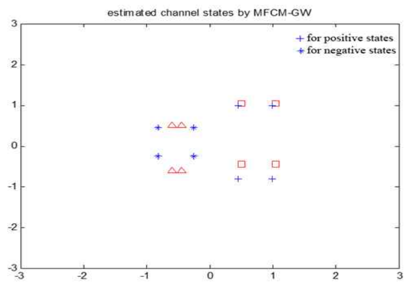
Fig. 5. NRMSE of the MFCM, the MFCM_GW and the proposed CFCM_GW.



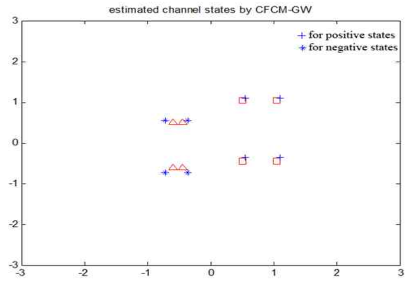
(a) received patterns under 0db SNR



(b) estimated channel states by MFCM



(c) estimated channel states by MFCM_GW



(d) estimated channel states by CFCM_GW

Fig. 6. A sample of received symbols under 0dB SNR for channel 2(a) and its eight desired channel states estimated by the MFCM(b), the MFCM_GW(c) and the proposed CFCM_GW(d) (including optimal positive(□) and negative(△) states).

In addition, the search times of all three algorithms are evaluated and included in Table 3. As the fast searching procedures of MFCM in [21] and MFCM_GW in [22], the proposed CFCM_GW uses the fast searching as well, which means the *for*-loop in the pseudo-code is skipped if the selected *index_j* has not been changed during the last 5 epochs. The overall search time by the proposed CFCM_GW, especially with 0dB SNR, is slightly slower because of the calculation time for the conditional constraint and the Gaussian weighted partition matrix. However, their difference is not significant where the proposed CFCM_GW provides much better performance in terms of NRMSE under 0dB SNR. Additionally, some of search times for the CFCM_GW under high SNRs, are faster, and it is caused by the use of the conditional constraint and the Gaussian weighted partition matrix which reduces the number of convergence epochs. Finally, the bit error rates (BER) when using the Bayesian equalizer is investigated and shown in Table 4. It becomes apparent that the BER with the estimated channel output states realized by the proposed CFCM_GW is close enough to the one with the optimal output states for all three channels. However, especially for low SNRs, the performance of CFCM_GW does not dominate in terms of BER as much as it does in terms of NRMSE. It is resulted from the fact that the Bayesian decision function shown in (8) is affected by heavy noise (high value of noise variance σ_e^2) even though the desired channel states are estimated well with high accuracy by the proposed CFCM_GW. For further improvement of BER, the decision function (or mechanism) of the equalizer should be investigated in near future.

Table 3. The averaged search time(in sec) for MFCM, MFCM_GW and proposed CFCM_GW (Matlab 7.0 run on Intel core 2).

Channel	SNR	MFCM	MFCM_GW	CFCM_GW
Channel 1	0.0dB	0.237	0.331	0.331
	2.5dB	0.231	0.237	0.212
	5.0dB	0.200	0.187	0.193
	7.5dB	0.181	0.150	0.150
	10dB	0.168	0.137	0.131
	avr. time	0.2034	0.2084	0.2034
Channel 2	0.0dB	0.256	0.293	0.343
	2.5dB	0.243	0.300	0.281
	5.0dB	0.256	0.200	0.218
	7.5dB	0.206	0.187	0.193
	10dB	0.193	0.193	0.168
	avr. time	0.2308	0.2346	0.2406
Channel 3	0.0dB	0.231	0.315	0.329
	2.5dB	0.231	0.231	0.225
	5.0dB	0.200	0.187	0.187
	7.5dB	0.187	0.150	0.150
	10dB	0.175	0.162	0.144
	avr. time	0.2048	0.2090	0.2070

Table 4. Averaged BER(%) (no. of errors/no. of transmitted symbols).

Channel	SNR	with optima l states	MFCM	MFCM_GW	CFCM_GW
Channel 1	0.0dB	19.8	21.4	21.6	20.1
	2.5dB	15.3	15.3	15.4	15.4
	5.0dB	10.7	10.7	10.6	10.6
	7.5dB	6.69	6.79	6.77	6.80
	10dB	2.77	2.78	2.75	2.75
Channel 2	0.0dB	27.5	30.0	34.0	32.1
	2.5dB	22.6	26.1	26.4	24.7
	5.0dB	16.0	17.1	16.0	16.0
	7.5dB	9.45	9.83	9.41	9.43
	10dB	4.93	5.14	4.87	4.87
Channel 3	0.0dB	19.3	21.5	22.1	19.2
	2.5dB	13.6	13.7	13.6	13.7
	5.0dB	8.95	9.06	9.09	9.08
	7.5dB	4.52	4.57	4.57	4.57
	10dB	1.79	1.75	1.77	1.76

VII. Conclusion

A modification of FCM (called CFCM_GW) aimed at the estimation of desired channel states of an unknown digital communication channel is provided for blind equalization, and successfully evaluated with both of linear and nonlinear channels. By taking this kind of approach, the highly demanding modeling task of an unknown channel becomes unnecessary as the construction of the desired channel states is accomplished directly on the basis of the estimated channel output states. It has been shown that the proposed CFCM_GW offers better performance in comparison to the solution provided by the previously developed algorithms (MFCM and MFCM_GW). In particular, because of the conditional constraint and the Gaussian weighted partition matrix, the proposed CFCM_GW can estimate the channel output states with substantial accuracy and speed even when the received symbols are significantly corrupted by heavy noise. Therefore, the CFCM_GW can possibly constitute a search algorithm of optimal channel states for the various problems of blind channel equalization. For future works, this algorithm could be evaluated with wider range of communication environments including higher order channels. Furthermore, as mentioned at the end of last section, the decision function of the Bayesian equalizer could be investigated for the improvement of BER in low SNRs. The research for more powerful search algorithms (easier and faster to compute or implement and more robust to heavy noise) will also be continued.

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] Z. Ding and L. Ye, *Blind Equalization and Identification*, Marcel Dekker, New York, 2001.

[3] H. Gazzah and K. A. Meraim, "Blind ZF equalization with controlled delay robust to order over estimation," *Signal Processing*, vol. 83, pp.1505-1518, 2003.

[4] J. Zhu, XR Cao and RW. Liu, "A blind fractionally spaced equalizer using higher order statistics," *IEEE Trans. Circuits Syst. II: Analog Digital Signal Process*, vol. 46, pp.755-764, 1999.

[5] 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.

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

[7] Yun Ye and Saman S. Abeysekera, "Efficient blind estimation and equalization of non-minimum phase communication channels via the use of a zero forcing equalizer," *Signal Processing*, vol. 86, pp.1019-1034, 2006.

[8] R. Tadisetti and P.R. Kumar, "Blind Channel Equalization Using CMA Algorithm," *International*

Conference on Advances in Recent Technologies in Communication and Computing, pp.680-683, 2009.
[9] S. Mo and B. Shafai, "Blind equalization using higher order cumulants and neural networks," *IEEE trans. Signal Processing*, vol. 42, pp.3209-3217, 1994.
[10] X. H. Dai, "CMA-based nonlinear blind equaliser modelled by a two-layer feedforward neural network," *IEE Proceedings:Communications*, vol. 148, pp.243-248, 2001.
[11] R. Pandey and J. K. Gautam, "Blind equalization of nonlinear communication channels using modified FLANN," *IETE Journal of Research*, vol. 48, pp.127-132, 2002.
[12] T. Stathaki and A. Scohyers, "A constrained optimization approach to the blind estimation of Volterra kernels," *Proc. IEEE Int. Conf. on ASSP*, vol. 3, pp.2373-2376, 1997.
[13] 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.
[14] 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.
[15] 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.
[16] R. Lopez-Valcarce and S. Dasgupta, "Blind equalization of nonlinear channels from second-order statistics," *IEEE Transactions on Signal Processing*, vol. 49, pp.3084-3097, 2001.
[17] G. Raz and B. Van Veen, "Blind equalization and identification of nonlinear and IIR systems-a least squares approach," *IEEE Trans. Signal Processing*, vol. 48, pp.192-200, 2000.
[18] S. Panda, S.M. Rout, M. Singh, U. Mishra, S.K. Routray and S.P. Panigrahi, "Blind channel equalization and phase recovery of transmitted signals using Autocorrelation Matching method," *Journal of Computing*, vol. 2, pp.145-152, 2011.
[19] 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.
[20] 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.
[21] S. Han, I. Lee and W. Pedrycz, "Modified fuzzy c-means and Bayesian equalizer for nonlinear blind channel," *Applied Soft Computing*, vol. 9, pp.1090-1096, 2009.
[22] S. Han and S. Park, "Blind linear/nonlinear equalization for heavy noise-corrupted channels," *International Journal of KIMICS*, vol. 7, pp.383-391, 2009.
[23] W. Pedrycz, "Conditional Fuzzy Clustering in the Design of Radial Basis Function Neural Networks," *IEEE Trans. Neural Networks*, vol. 9, pp.601-612, 1998.
[24] K. Yoon, K. Kwak and S. Kim, "Nonlinear channel equalization using fuzzy clustering adaptive neuro-fuzzy filter," *Journal of Korea Electronics Engineers Society*, vol. 38, pp.35-41, 2001.
[25] S. Han, "Blind channel equalization using conditional fuzzy c-means," *Journal of Korea Multimedia Society*, vol. 14, pp.965-980, 2011.
[26] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
[27] S. Chen, B. Mulgrew and S. McLaughlin, "Adaptive Bayesian equalizer with decision feedback," *IEEE Trans. Signal Processing*, vol. 41, pp.2918-2927, 1993.
[28] S. Chen, B. Mulgrew and M. Grant, "A clustering technique for digital communication channel equalization using radial basis function network," *IEEE Trans. Neural Networks*, vol. 4, pp.570-579, 1993.
[29] H. Lin and K. Yamashita, "Blind equalization using parallel Bayesian decision feedback equalizer," *Mathematics and Computers in Simulation*, vol. 56, pp.247-257, 2001.
[30] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

Soo-whan Han (Member) received the B.S. in electronics from Yonsei University in 1986, and the M.S. and the Ph.D. degree from Forida Tech. in 1990 and 1993, respectively.

He is currently a professor at the Dept. of Multimedia Engineering in Dongeui University, Korea. His research interests include the areas of Digital Signal & Image Processing, Pattern Recognition, and Fuzzy Logic & Neural Networks.