Efficient Architecture for Bayesian Equalization Using Fuzzy Filters

Sarat Kumar Patra, *Student Member, IEEE,* and Bernard Mulgrew, *Member, IEEE*

***Abstract*—A normalized Bayesian solution is derived for digital communication channel equalization which uses estimates of scalar channel states. This equalizer is termed as a normalized Bayesian equalizer with scalar channel states (NBEST). The relationship between the NBEST and fuzzy equalizers is derived and computational aspects of fuzzy equalizers are investigated using different types of fuzzy basis functions. It is shown that the fuzzy equalizer in general demands much lower computational complexity than the optimum equalizer. Ways to further reduce the computation complexity of fuzzy equalizers is proposed and their performance evaluated. A novel scheme to select a subset of channel states close to the received vector, resulting in considerable reduction in the computational complexity, is also proposed. A fuzzy equalizer with this modified membership function is shown to perform close to the Bayesian equalizer.**

***Index Terms*—Bayesian equalizers, digital communication systems, equalizers, fuzzy systems.**

I. INTRODUCTION

THE SPEED OF transmission of information over a communication system is limited due to the effect of intersymbol interference (ISI) in the presence of noise. The process of removing the effects of ISI in the presence of noise to faithfully reconstruct the transmitted symbols is termed equalization. The structure of the communication system discussed here is presented in Fig. 1.

The information symbol to be transmitted $s(k)$ is transmitted through a linear dispersive channel described by

$$r(k) = \sum_{i=0}^{n_h-1} a_i s(k-i) + e(k). \tag{1}$$

Here n_h is the channel tap length with a_i being the individual taps and $e(k)$ refers to the additive white Gaussian noise (AWGN). The noise-free received sample of the channel is referred to as $\hat{r}(k)$ and the received samples are referred as $r(k)$. The equalizer reconstructs the transmitted symbols $s(k)$ by observing the noisy received signal vector $\mathbf{r}(k) = [r(k), r(k-1), \cdots, r(k-m+1)]^T \in R^m$, where m is the order of the equalizer. Normally a delay is associated with detection and the equalizer output is a delayed form of the transmitted sequence and can be represented as $\hat{s}(k-d)$. Here we assume that the communication system is binary in nature so that the transmitted symbol $s(k)$ is taken from $+1$ or -1 with equal probability. This makes the analysis easier and the same can be extended to any digital communication system in general.

The process of adaptive equalization can be classified into two categories, namely, sequence estimation and symbol decision equalization. The optimal solution for the sequence estimation equalizer is maximum-likelihood sequence estimation (MLSE) [1]. This equalizer can be implemented using the Viterbi algorithm [2]. The process of MLSE demands high computational complexity and requires knowledge of the channel. Generally the channel information is not available at the receiver end. For this reason a channel estimator is included in the MLSE equalization. The channel estimation can be a difficult process under certain circumstances, e.g., when the channel is associated with nonlinear terms. The symbol decision equalizers are relatively simple and computationally less complex than the MLSE. They in fact do not always require an explicit channel estimate. The optimal solution for a symbol equalizer can be formed from the Bayes probability theory [3] and is termed the Bayesian equalizer. The symbol decision equalizer can also be considered as an inverse filter [4] where an adaptive linear filter based on algorithms like least mean square (LMS) or recursive least square (RLS) are used. The adaptive filter here finds the channel inverse in the presence of noise providing a linear decision boundary. Generally the decision function demanded by an optimal equalizer is nonlinear in nature. The problem of equalization can be considered as a classification problem where the equalizer classifies the received signal vector to one of the signal constellations. With this perspective, the equalization can be treated as a nonlinear classification problem and for this reason the performance of the linear equalizers are far from optimal. This has led to the search for nonlinear equalizers that can provide a nonlinear decision function. Nonlinear equalizers using an artificial neural networks (ANN's) [5], [6], [7] and radial basis function (RBF) networks [5], [8], [9] have been successfully developed. The ANN equalizer provides a nonlinear decision function but the convergence of the network suffers due to its multimodal local minima. The RBF equalizers, on the other hand, provides localized functional behavior demanded by the optimal equalizer decision function but training of the centers is difficult. However, the orthogonal least square algorithm (OLMS) [10] or clustering [11] can be used to train the centers. Clustering in multidimensional

Manuscript received December 30, 1996; revised October 3, 1997. This paper was recommended by Associate Editor B. Leung.

S. K. Patra is with Signal and Systems Group, Electrical Engineering, University of Edinburgh, Edinburgh EH9 3JL, Scotland, U.K., on leave from the Department of Applied Electronics and Instrumentation Engineering, REC, Rourkela 769 008, Orissa , India.

B. Mulgrew is with Signal and Systems Group, Electrical Engineering, University of Edinburgh, Edinburgh EH9 3JL, Scotland, U.K.

Publisher Item Identifier S 1057-7130(98)05057-5.

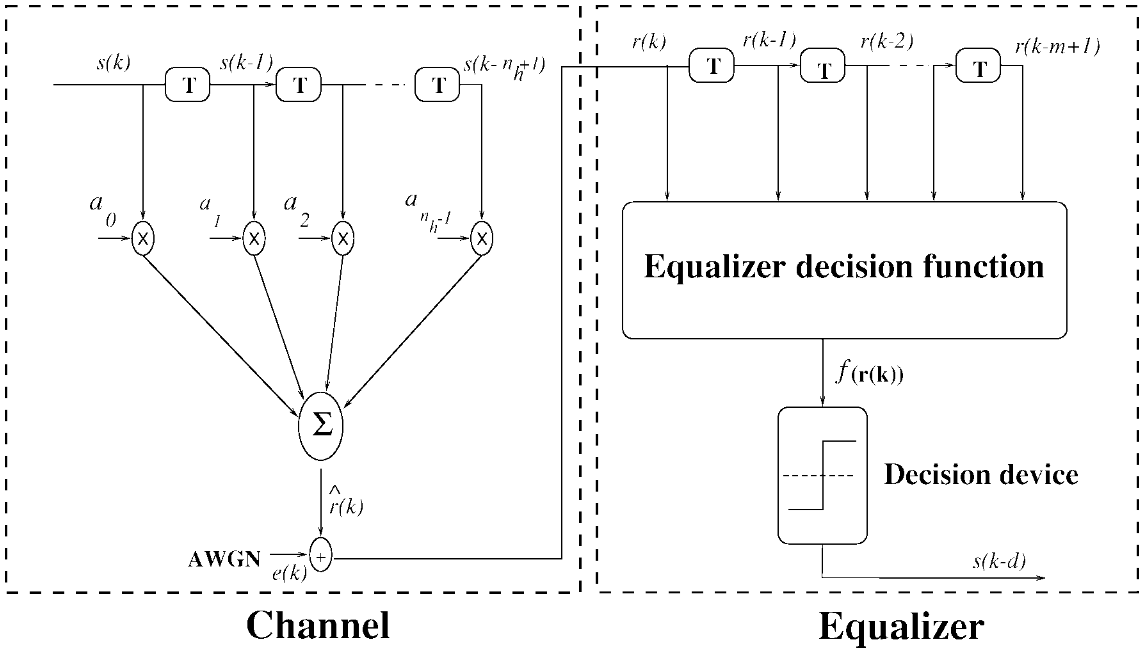


Fig. 1. A communication system.

space is computationally complex and require long training sequences. This has led to the search for other nonlinear equalization techniques. A fuzzy equalizer based on a fuzzy adaptive filter was proposed in [12] and an equalizer based on a fuzzy system was proposed in [13]. These equalizers performed well but could not provide the Bayesian equalizer decision function. Additionally, the equalizer based on a fuzzy adaptive filter demanded high computational complexity. This paper investigates a new form of implementation of Bayesian equalizers using fuzzy filters.

The paper is outlined in six sections. Section II discusses the Bayesian equalizer and derives the process of its fuzzy implementation. Section III introduces the training schemes for fuzzy equalizers. Section IV discusses the advantages of implementing a fuzzy equalizer over the Bayesian equalizer. In Section V we present some of the simulation results, and Section VI provides the conclusions.

II. Bayesian Equalizer and its Fuzzy Implementation

The general symbol decision equalizer depicted in Fig. 1 is characterized by equalizer order m and delay d. The optimal decision function for this equalizer is derived from Baye's probability theory [3] and can be represented as [14]

$$f(\mathbf{r}(k)) = \sum_{i=1}^{n_s^+} \left(2\pi\sigma_e^2\right)^{-m/2} \exp\left(\frac{-\|\mathbf{r}(k)-\mathbf{c}_i^+\|^2}{2\sigma_e^2}\right) - \sum_{j=1}^{n_s^-} \left(2\pi\sigma_e^2\right)^{-m/2} \exp\left(\frac{-\|\mathbf{r}(k)-\mathbf{c}_j^-\|^2}{2\sigma_e^2}\right). \quad (2)$$

Here $\mathbf{r}(k)$ represents the equalizer input vector, σ_e^2 represents the channel noise variance, and $\mathbf{c}_i^+ \in R^m$ and $\mathbf{c}_j^- \in R^m$ are the positive and negative channel states, respectively. The terms n_s^+ and n_s^- are the number of positive and negative channel states respectively and they are equal. With the assumption of binary transmission, the sign of the decision function in (2) is sufficient to provide the decision and scaling terms can be ignored. With this, the decision function can be represented as

$$f(\mathbf{r}(k)) = \sum_{i=1}^{n_s} p_i \exp\left(\frac{-\|\mathbf{r}(k)-\mathbf{c}_i\|^2}{2\sigma_e^2}\right) \quad (3)$$

where n_s is the number of channel states, equal to 2^{n_h+m-1} with $n_s^+ = n_s^- = n_s/2$, and p_i are the weights associated with each of the centers. $p_i = 1$ if $\mathbf{c}_i \in n_s^+$ and $p_i = -1$ if $\mathbf{c}_i \in n_s^-$. It is also observed that each of the channel state vectors has m components which can be represented as $\mathbf{c}_i = [c_{i0}, c_{i1}, c_{i2}, \cdots, c_{i(m-1)}]^T \in R^m$. Rewriting the squared norm of (3) as a summation and exploiting the properties of the $\exp$ function yields

$$f(\mathbf{r}(k)) = \sum_{i=1}^{n_s} p_i \left\{ \prod_{l=0}^{m-1} \exp\left(-\frac{\|r(k-l)-c_{il}\|^2}{2\sigma_e^2}\right) \right\} \quad (4)$$

where c_{il} is the $(l+1)$th component of channel state vector $\mathbf{c}_i$, corresponding to the $(l+1)$th component of the input vector $\mathbf{r}(k)$.

Equations (3) and (4) provide alternative realizations of the Bayesian decision function. In (3) the Euclidian distance between input vector $\mathbf{r}(k)$ and each of the channel states $\mathbf{c}_i$ is first calculated. The result is then scaled by $-1/(2\sigma_e^2)$ and the exponential function is evaluated. These are linearly combined to provide the decision function. Alternatively in (4), scalar distances are calculated, scaled by $-1/(2\sigma_e^2)$ and the exponential function evaluated. The products of exponential functions associated with particular channel states are linearly combined to provide the decision function. Both of these functions require the knowledge of channel states for estimating the decision function. It was noted in [15] that (4) may be preferable to (3) for implementation. This approach is adopted here.

In line with the normalized radial basis function derived by Cha *et al.* [16], we can form a normalized Bayesian equalizer which forms an estimate of the transmitted symbols themselves rather than a decision function. This we represent as a normalized Bayesian equalizer with scalar channel states (NBEST). The NBEST provides a localized behavior and nonlocalized behaviors whereas the equalizer in (3) provides localized behavior

$$f(\mathbf{r}(k)) = \frac{\sum_{i=1}^{n_s} p_i \left\{ \prod_{l=0}^{m-1} \exp\left(-\frac{\|r(k-l) - c_{il}\|^2}{2\sigma_e^2} \right) \right\}}{\sum_{i=1}^{n_s} \left\{ \prod_{l=0}^{m-1} \exp\left(-\frac{\|r(k-l) - c_{il}\|^2}{2\sigma_e^2} \right) \right\}}. \tag{5}$$

The estimation of the decision function for Bayesian equalizer given by (3) and (5) needs the channel states. The channel states can be estimated during the training period. An insight into the equalizer decision function in (5) reveals that the equalizer contains n_s channel states each of m dimensions. The number of scalar channel states for any channel is $M = 2^{n_h}$. Each of the m components of n_s channel states are taken from the set of M scalar channel states. With this understanding, the equalizer decision function can be presented as

$$f(\mathbf{r}(k)) = \frac{\sum_{i=1}^{n_s} p_i \{ \prod_{l=0}^{m-1} \phi_l^{ij} \}}{\sum_{i=1}^{n_s} \{ \prod_{l=0}^{m-1} \phi_l^{ij} \}} \tag{6}$$

where ϕ_l^{ij} is a basis function of the form

$$\phi_l^{ij} = \exp\left[-\frac{1}{2} \left(\frac{r(k-l) - c_l^{ij}}{\sigma_e} \right)^2 \right]. \tag{7}$$

Here ϕ_l^{ij} is the basis function output generated from the scalar center c_l^{ij}, corresponding to the $j+1$ scalar center of $l+1$ element of $\mathbf{r}(k)$, where $0 \le l \le (m-1)$ and $0 \le j \le (M-1)$. i in (6) corresponds to the channel states number added for convenience, l and j are sufficient to specify the parameters for the equalizer. In (6), computation of $\prod_{l=0}^{m-1} \phi_l^{ij}$ is the same as the computation of $\exp(\frac{-\|\mathbf{r}(k) - \mathbf{c}_i\|^2}{2\sigma_e^2})$ in (3).

A. Fuzzy Implementation

Wang and Mendel [12] proposed fuzzy LMS and fuzzy RLS filters and used them for nonlinear channel equalization. Subsequently, fuzzy filters of different structures were used for equalization [17]–[20] in a variety of applications. In our fuzzy implementation of the Bayesian equalizer we use the architecture of [12] which was used in conjunction with the RLS training algorithm. In this fuzzy filter, setting the membership function centers with scalar channel states, the spread parameter with the channel noise variance and generating the membership functions with (7), an equalizer with fuzzy filter can be represented as

$$f_k(\mathbf{r}(k)) = \frac{\sum_{i=1}^{n_c} \theta_i \{ \prod_{l=0}^{m-1} \psi_l^{ij} \}}{\sum_{i=1}^{n_c} \{ \prod_{l=0}^{m-1} \psi_l^{ij} \}} \tag{8}$$

where ψ_l^{ij} is the membership function generated from the scalar center c_l^{ij}, corresponding to the $(j+1)$th center of the $(l+1)$th input scalar. In this θ_i is a free design parameter of the filter which is adjusted during the training process. Here n_c corresponds to all possible combinations of the membership function taking one from each input scalar and $n_c = M^m$. The membership function is generated with (7) where ϕ is replaced by ψ. The superscript i is used for convenience and the terms l and j specify all the parameters of the function. The equalizer function (8) finds a weighted sum of the fuzzy basis functions (FBF's) [21]–[23] given by

$$f_{\mathrm{fbf}}(\mathbf{r}(k)) = \frac{\prod_{l=0}^{m-1} \psi_l^j}{\sum_{i=1}^{n_c} \{ \prod_{l=0}^{m-1} \psi_l^j \}}. \tag{9}$$

It may be noted that this FBF uses a singleton fuzzifier, product inference, center of gravity (COG) defuzzifier, and Gaussian membership function, and the filter in (8) forms the linear combination of these FBF's.

On observing the decision functions of NBEST (6) and the fuzzy equalizer (8), it can be seen that the NBEST has $n_s = 2^{n_h+m-1}$ terms and the fuzzy equalizer has $n_c = M^m$ terms. From this it is seen that the number of terms in NBEST is a subset of the terms in the fuzzy filter. If the channel states are known to the receiver, the corresponding weights of n_s terms of the fuzzy filter can be assigned $+1/-1$ depending on the values of p_i in NBEST and the remaining $n_c - n_s$ terms can be neglected to provide the optimum decision function. Hence the fuzzy equalizer decision function can be represented by (6) where only n_s fuzzy basis functions out of the available n_c functions are used. This reduces the computations involved with $n_c - n_s$ fuzzy basis functions and provides an optimum decision function. With this we can represent (6) as the fuzzy implementation of the Bayesian equalizer.

B. Fuzzy Equalizer Structure

The structure of the fuzzy equalizer is presented in Fig. 2. Here, the incoming signal sample is presented to the membership function generator. Each of the components of the membership function generator produces an output ϕ_l^j, characterized by its center c_l^j which are placed at the scalar channel states. Here l corresponds to the equalizer input number and j represents the fuzzy center at the scalar channel states. The membership functions are to be generated from each of the received scalars. The equalizer input vector is formed from the time-delayed samples of the reveived scalar. With this the membership function for input scalar $r(k-1)$ will be the delayed membership functions for input $r(k)$. This can be represented as

$$\phi_l^j(k) = \phi_{l-1}^j(k-1) \tag{10}$$

where $1 \le l \le (m-1)$ and $0 \le j \le M-1$.

The inference block of the equalizer has n_s units. Each of these units receives only one ϕ_l^j from each of the m inputs to the equalizer, and the combination of these is decided by the combination of the scalar channel states constituting the channel states. The output of the inference units are suitably weighted and added to provide a and b. The output of the

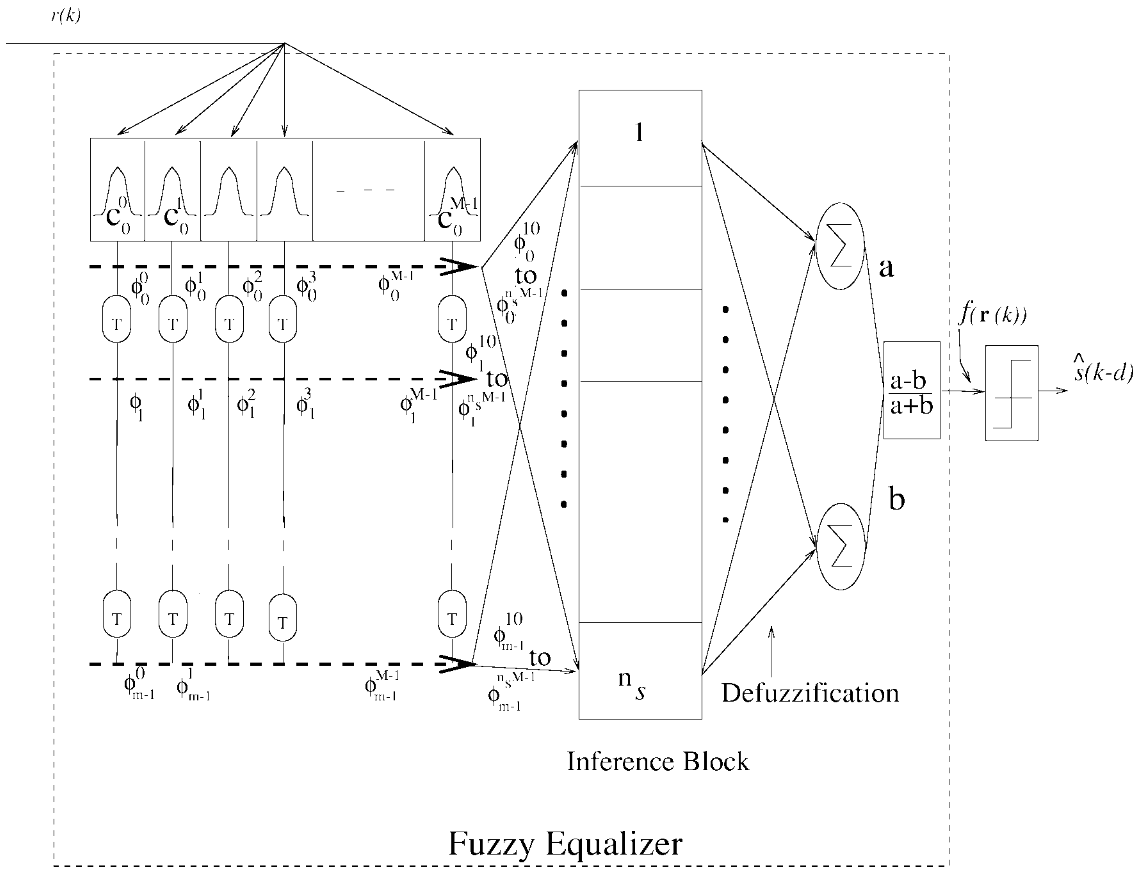


Fig. 2. Fuzzy implementation of Bayesian equalizer.

TABLE I
THE CHANNEL STATES CALCULATION FOR CHANNEL $0.5 + 1.0z^{-1}$ WITH $m = 2$, $d = 0$, $n_s = 8$ AND $M = 4$

No.	s(k)	s(k-1)	s(k-2)	$\hat{r}(k)$	$\hat{r}(k-1)$	
1	1	1	1	1.5	1.5	Positive
2	1	1	-1	1.5	-0.5	channel
3	1	-1	1	-0.5	0.5	states
4	1	-1	-1	-0.5	-1.5	
5	-1	1	1	0.5	1.5	Negative
6	-1	1	-1	0.5	-0.5	channel
7	-1	-1	1	-1.5	0.5	states
8	-1	-1	-1	-1.5	-1.5	

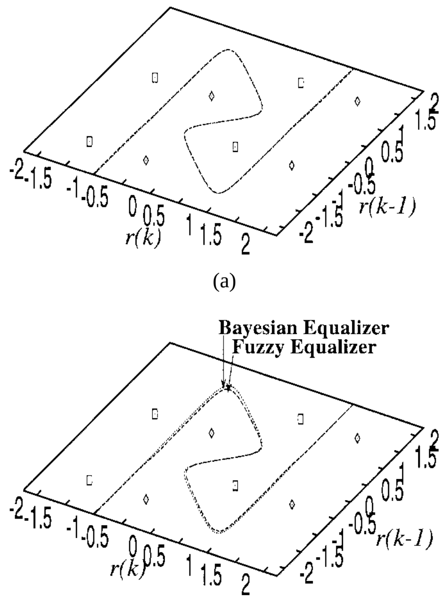
equalizer is computed by the equalizer function presented in (6) which is $(a - b)/(a + b)$. The output of the decision function passed through the sigmoid nonlinearity forms the detected sample. We consider an example to illustrate the working of this equalizer:

Example 1: We consider the channel $H_{\text{ch1}}(z) = 0.5 + z^{-1}$. The equalizer order is $m = 2$ and delay $d = 0$. The signal-to-noise ratio (SNR) is 8 dB. This provides $n_s = 8$ and $M = 4$. The channel states for this equalizer are presented in Table I. The channel states for this are $[\hat{r}(k), \hat{r}(k-1)]^T$. The scalar channel states constitute each of the possible $\hat{r}(k)$ or $\hat{r}(k-1)$. It is also seen that the m-dimensional n_s channel states take their components from the available M scalar channel states. The weights p_i of the equalizer decision function are $+1$ and -1 for positive and negative states, respectively.

For fuzzy implementation, the centers for membership function generators are placed at scalar channel states $+1.5$, $+0.5$, -0.5 and -1.5. The basis functions ϕ_1^0, ϕ_1^1, ϕ_1^2, ϕ_1^3 corresponding to $r(k-1)$, will be a delayed version of ϕ_0^0, ϕ_0^1, ϕ_0^2, ϕ_0^3 corresponding to $r(k)$. The inference block will consist of $n_s = 8$ subblocks. The products $\phi_0^0\phi_1^0$, $\phi_0^0\phi_1^2$, $\phi_0^2\phi_1^1$, $\phi_0^2\phi_1^3$ are added to provide a and $\phi_0^1\phi_1^0$, $\phi_0^1\phi_1^2$, $\phi_0^3\phi_1^1$, $\phi_0^3\phi_1^3$ are added to provide b. The calculation of the decision function next is straightforward.

The decision boundary of this equalizer is presented in Fig. 3. Here Fig. 3(a) presents the decision boundary of the fuzzy equalizer and the Bayesian equalizer when the channel states and noise statistics are known, where as in Fig. 3(b) the fuzzy equalizer uses the estimated channel states and noise statistics and the Bayesian equalizer uses the true channel parameters. This shows that the fuzzy equalizer is able to provide a near optimal decision boundary even at a low SNR of 8 dB.

The fuzzy equalizer developed here uses an FBF with product inference and COG defuzifier. Owing to the close relationship of this equalizer with the Bayesian equalizer, this equalizer can also be implemented with an RBF [9] with scalar centers [24]. However, use of a fuzzy system to implement this equalizer provides the possibity of using other forms of inference rules and defuzzification processes. This can provide some of the alternate forms of fuzzy implementation of the Bayesian equalizer.



(a)

(b)

Fig. 3. Fuzzy equalizer decision boundary for channel $0.5 + 1.0z^{-1}$, equalizer order $m = 2$, delay $d = 0$ with 8-dB SNR. (a) Actual states,. (b) Bayesian equalizer with actual states and fuzzy equalizer with estimated states.

1) Inference Rule: The fuzzy equalizer discussed above works with product inference. The output of each of the n_s inference rules are generated with the product rule. It is also seen from the membership function generator (7) that the membership for any input is $0 < \phi_l^j \leq 1$. Hence the output of any of the inference rules will be in the range $[0, 1]$ and will always be less than the smallest membership input to the rule. For this reason the product inference rule can be approximated by minimum inference rule. With this modification the equalizer decision function would be

$$f(\mathbf{r}(k)) = \frac{\sum_{i=1}^{n_s} p_i \{\min_{l=0}^{m-1} \phi_l^{ij}\}}{\sum_{i=1}^{n_s} \{\min_{l=0}^{m-1} \phi_l^{ij}\}}. \qquad (11)$$

Here $\min_{l=0}^{m-1}$ selects the minimum of the m inputs to each of the components of the inference block. With this, the computation of the products has been replaced by comparisons which are easy to implement in hardware.

2) Defuzzification Process: The output layer of the fuzzy equalizer [see (6) and (8)] finds a weighted sum of the inference rules and normalizes this with the inference output. The weights associated with the inference rules are $+1/-1$. It is seen that the rule nearest to the input vector would provide the maximum output, and the contribution from the remaining rule will be minimal. These characteristics of the decision function can be utilized by replacing the COG defuzzifier with a maximun defuzzifier. This defuzzifier can be combined either with product inference or with the minimum inference. With this, the equalizer decision function can be represented as

$$f(\mathbf{r}(k)) = \frac{p_{\max} \max_{i=1}^{n_s} \{\prod_{l=0}^{m-1} \phi_l^{ij}\}}{\max_{i=1}^{n_s} \{\prod_{l=0}^{m-1} \phi_l^{ij}\}} \qquad (12)$$

$$f(\mathbf{r}(k)) = \frac{p_{\max} \max_{i=1}^{n_s} \{\min_{l=0}^{m-1} \phi_l^{ij}\}}{\max_{i=1}^{n_s} \{\min_{l=0}^{m-1} \phi_l^{ij}\}}. \qquad (13)$$

Here $\max_{i=1}^{n_s}$ corresponds to the maximum of the available n_s inferences and $p_{\max}$ is the weight associated with the maximum inference. With this decision function, (12) and (13) use maximum defuzzification, where the output of the equalizer is based on the maximum of the n_s inference rules and the weight associated with it. The equalizer (12) uses product inference where as (13) uses the minimum inference rule. In both of these defuzzification processes, computation of weighted sum of the inference is replaced by a comparison operation.

With the above analysis, a variety of fuzzy equalizers to approximate the Bayesian decision function can be designed. These equalizers can provide alternative equalizer architectures with a reduction in computational complexity.

III. Fuzzy Equalizer Training

The fuzzy equalizer developed here can be trained in two steps.

A. Step 1: Channel State Estimation

The estimation of the decision function using the fuzzy equalizer given by (6) and (8) needs the scalar channel states and their combination which forms the channel states. Estimation of these requires the channel information which in most cases is not available. However, these can be estimated during the training period and can be achieved in the following ways [14].

- The channel model can be identified using some algorithms like the LMS. With the knowledge of the channel it is straightforward to calculate the scalar channel states and their combination which forms the channel states. This technique may fail if the channel suffers from nonlinear distortion.
- The scalar channel states can be computed with scalar supervised clustering. This in conjuction with the training signal can provide the scalar states combinations that form the channel states [11]. The number of scalar channel states depend only on channel order and hence will demand a smaller training sequence compared to vector channel state estimation. Fig. 4 presents the learning curve for scalar channel state estimation for channel $H_{\text{ch2}}(z) = 0.5 + 0.81z^{-1} + 0.31z^{-2}$. Here the channel state estimation has been averaged over 20 experimants. From the training curves it is seen that the scalar channel states converge to the desired states in around 30 iterations.

B. Step 2: Equalizer Weight Update

Once the scalar channel states have been estimated, the fuzzy rules can be formed. Next the equalizer is constructed with weights of the inference rules assigned to $+1/-1$ depending on whether the rule belongs to a positive or negative channel state. The channel states and the noise statistics estimation can involve some error. In order to compensate for this the weights associated with the rules can be fine tuned with a gradient descent algorithm and a training signal. This step

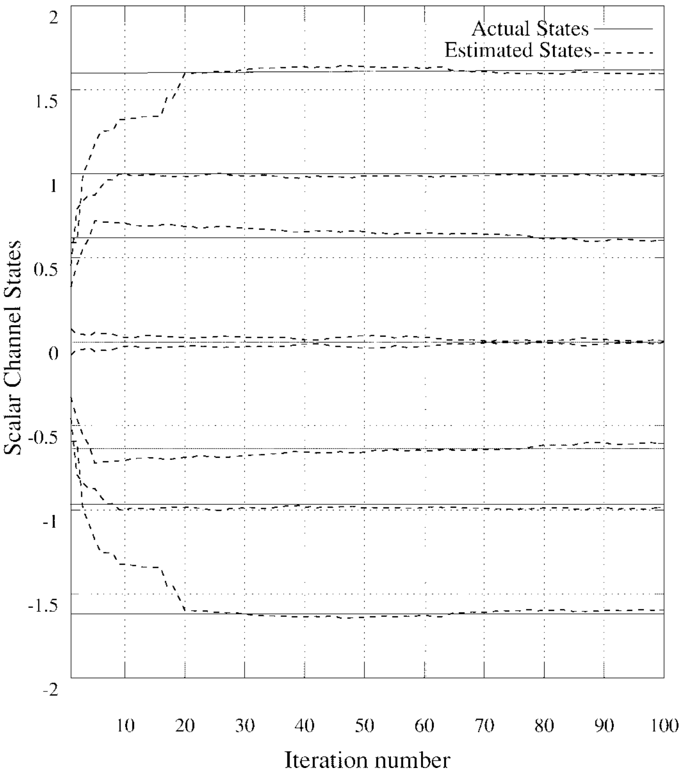


Fig. 4. Scalar channel states training curve for channel $0.5 + 0.81z^{-1} + 0.31z^{-2}$, the actual channel states ± 1.62, ± 1.0, ± 0.62, and ± 0.00.

would take only a few samples as the initial weight assignment is very close to the final values.

IV. Advantages of the Fuzzy Equalizer

We have seen that the fuzzy implementation of NBEST provides the Bayesian equalizer decision function. An insight into the Bayesian decision function (4) and the fuzzy implementation of NBEST shows some of the advantages of fuzzy implementation of a Bayesian equalizer. These advantages are summarized below.

A. Computational Complexity

After training is complete, the equalizer parameters are fixed and the actual detection of transmitted symbols starts. The computational requirements of a fuzzy equalizer and NBEST are the same. The computations required for estimating each of the samples with the Bayesian equalizer and its RBF implementation, NBEST and the fuzzy equalizer are listed in Table II. The second part of the table provides the typical computational requirements for an equalizer with $m = 4$, $n_h = 3$, and $M = 8$. From this table, the following inferences can be arrived at with regard to the computational advantages of fuzzy implementation of Bayesian equalizer.

- Fuzzy implementation of the Bayesian equalizer provides a significant reduction in addition, division and $\exp(x)$ evaluations.
- The time shift property of the membership function generation provides a considerable reduction in evaluation of $\exp(x)$ functions and division.

TABLE II
COMPUTATIONAL COMPLEXITY COMPARISON FOR BAYESIAN EQUALIZERS, NBEST, AND FUZZY EQUALIZERS AND FOR EQUALIZERS WITH $m = 4$, $n_h = 3$, $n_s = 64$, AND $M = 8$

Equalizer Type	**Add/ Sub**	**Mul**	**Div.**	e^{-x}
Bayesian(RBF)	$2mn_s$	mn_s	n_s	n_s
NBEST	$M + n_s$	$M + mn_s$	$M + 1$	M
FUZZY	$M + n_s$	$M + mn_s$	$M + 1$	M
Bayesian (RBF)	512	256	64	64
NBEST	72	264	9	8
FUZZY	72	264	9	8

TABLE III
COMPUTATIONAL COMPLEXITY COMPARISON FOR DIFFERENT FUZZY EQUALIZERS AND FOR EQUALIZERS WITH $m = 4$, $n_h = 3$, $n_s = 64$ AND $M = 8$

Inf Type	**Defuzz Type**	**Add/ Sub**	**Mul**	**Div.**	e^{-x}	**Compare**
Prod	COG	$M + n_s$	$M + mn_s$	$M + 1$	M	
Min.	COG	$M + n_s$	$M + n_s$	$M + 1$	M	$(m - 1)n_s$
Prod	Max.	M	$M + (m - 1)n_s + 1$	$M + 1$	M	n_s
Min.	Max.	M	$M + 1$	$M + 1$	M	mn_s
Prod	COG	72	264	9	8	
Min.	COG	72	72	9	8	192
Prod	Max.	8	201	9	8	64
Min.	Max.	8	9	9	8	256

- Evaluation of $\exp$ and division in a Bayesian equalizer are related to n_s which in turn is exponentially related to the equalizer order but in the fuzzy equalizer they are related to M which is independent of the equalizer order. Hence, with the increase in the equalizer order the reduction in computational complexity for fuzzy equalizer over the Bayesian equalizer can be exponentially related.
- Introduction of the minimum inference rule and maximum defuzzification replaces the product computation by a comparison operation which is very easy to implement and fast to process in real time implementation. The computations involved for estimation of each symbol with this modification is presented in Table III. The second part of the table provides the typical figure for an equalizer with $m = 4$, $n_h = 3$, and $M = 8$. From this it is seen that using minimum inference or maximum defuzzification replaces the product computation by comparison operation considerably. These provide an alternate approximation to the Bayesian decision function evaluation with a reduction in the computational complexity.

In this paper we have compared the computational complexity of a fuzzy equalizer with a Bayesian equalizer which can be implemented with an RBF. The Bayesian equalizer provides the optimum performance for symbol spaced equalizers providing the upperbound for bit error rate (BER) performance for any symbol spaced equalizers. The computational complexity adavantages and disadvantages of Bayesian equalizer against MLSE and linear equalizers are discussed in [25] and [14].

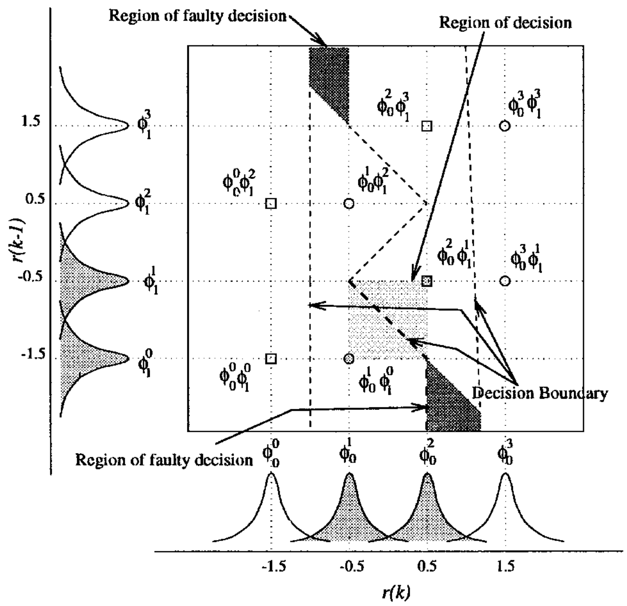


Fig. 5. Decision boundary for membership function modification for channel $0.5 + 1.0z^{-1}$, equalizer order $m = 2$, delay $d = 0$.

B. Subset State Selection

In the Bayesian equalizer (3), the equalizer decision function is based on a weighted sum of n_s basis functions centered at the channel states. From the decision function it can be seen that the contribution of a channel state to the decision function is inversely related to its distance from the input vector. Under this circumstance, if a set of channel states near the input vector can be found, the equalizer decision function can be approximated with this subset of the available n_s channel states. Chng [26] proposed a process of selecting a subset of available channel states to approximate the Bayesian decision function with a smaller number of channel states. In a fuzzy implementation of the Bayesian equalizer it is very easy to employ subset state selection to reduce the number of inference rules, which reduces the computation involved. This involves modification of the membership function. In general, all M membership functions corresponding to a input provide nonzero output irrespective of the input scalar. If an input is far from a scalar center, the membership function from that center will be neglegible and can be neglected. Considering this, it may be enough to use only the two nearest centers from the observed received scalars for membership function calculation and the membership function contribution from other centers can be neglected. This provides only two nonzero membership function out of the available M functions for each input. This would generate only two nonzero inferences against 2^{m+n_h-1}. Using some simple checks to determine these rules, the decision function can be computed. We illustrate this with an example.

Example 2: We consider the problem presented earlier. The equalizer has eight channel states constructed from four scalar channel states. The equalizer decision making capability under this circumstance is presented in Fig. 5. Here the positive channel states are shown as ◯ and negative channel states are shown as □. The membership function generation for $r(k)$ and $r(k-1)$ are shown along the sides. Consider an input vector $[0.0, -0.75]^T$. This input vector provides nonzero membership functions for ϕ_0^1, ϕ_0^2 and ϕ_1^0, ϕ_1^1. These, when translated with inference rules with channel states intotwo dimensions, would provide only two nonzero inference rules corresponding to the channel states $[-0.5, -1.5]^T$ and $[0.5, -0.5]^T$. The region of space that will be covered by the membership functions is shown shaded in the figure. The decision function for this input region is a straight line equidistant from both centers in the space covered by the membership functions. With a change in the input vector, different sets of inference rules corresponding to channel states will be selected, providing a combined decision boundary, as shown. All these individual decision boundaries join to provide a nonlinear decision boundary. The region in which the equalizer is unable to approximate the decision region is also shown in the figure.

TABLE IV
COMPUTATIONAL COMPLEXITY COMPARISON FOR FUZZY EQUALIZERS WITH MODIFIED MEMBERSHIP FUNCTION GENERATION FOR SUBSET STATE SELECTION AND FOR EQUALIZERS WITH $m = 4$, $n_h = 3$, $n_s = 64$, AND $M = 8$

Inf Type	Defuzz Type	Add/ Sub	Mul	Div.	e^{-x}	Compare
Prod	COG	$M+2$	$M+2m$	3	2	
Min.	COG	$M+2$	$M+2$	3	2	$< (m-1)2^m$
Prod	Max.	M	$M+2(m-1)+1$	3	2	$< 2^m$
Min.	Max.	M	$M+1$	3	2	$< m2^m$
Prod	COG	10	16	3	2	
Min.	COG	10	10	3	2	< 48
Prod	Max.	8	15	3	2	< 16
Min.	Max.	8	9	3	2	< 64

This form of modification of the membership function can reduce the computational complexity of the equalizer considerably. The computation involved per sample calculation with this form of membership function is presented in Table IV.

This modification of the membership function provides a natural method for selecting a subset of the available channel states reducing the computational complexity. However, if the scalar channel states are very closely spaced, this process of selecting only two membership functions may not provide good performance. Under this circumstance, however, more than two nonzero membership functions of the input vector may be used. With the increase in the number of membership functions, the number of nonzero inference rules will increase, providing a better performance at the cost of higher computation. However, if a subset of all the available scalar channel states are selected, the numbers of valid fuzzy rules will be less than n_s. This provides a way of trading performance for complexity within the equalizers.

V. RESULTS AND DISCUSSION

Fuzzy equalizers developed in the previous sections were evaluated with extensive simulations. To study the performance of the fuzzy equalizers, BER performance was evaluated for different fuzzy equalizers and compared with the Bayesian equalizer performance. The equalizer performance

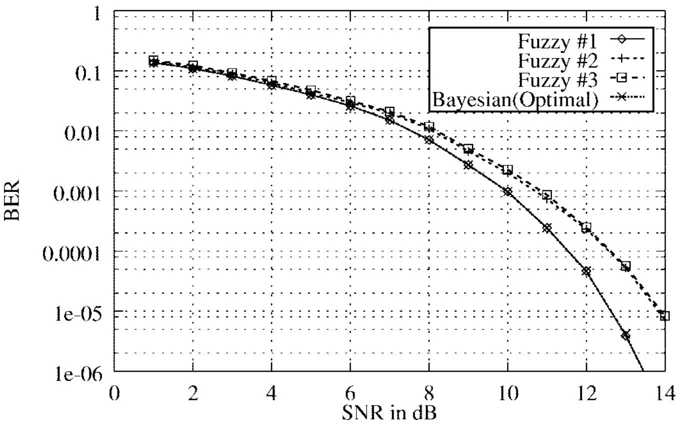


Fig. 6. Fuzzy equalizer BER performance with different types of fuzzy systems for channel $H_{\mathrm{ch}3}(z) = 0.407 - 0.815z^{-1} - 0.407z^{-2}$, order $m = 5$, delay $d = 3$, centers at ± 1.629, ± 0.815, and ± 0.001. Fuzzy-1: Product Inference and COG defuzzifier; Fuzzy-2: minimum inference with COG defuzzifier; Fuzzy-3: minimum inference with maximum defuzzifier.

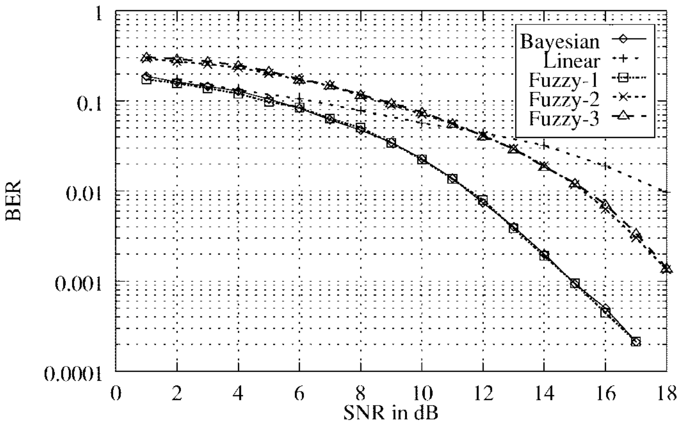


Fig. 7. Fuzzy equalizer BER performance with estimated channel states and noise variance for channel $H_{\mathrm{ch}4}(z) = 0.3482 + 0.8704z^{-1} + 0.3483z^{-2}$, order $m = 4$, delay $d = 1$. Fuzzy-1: fuzzy equalizer with product Inference and COG defuzzifier; Fuzzy-2: limiting membership function with product inference and COG defuzzifier; Fuzzy-3: limiting membership function with minimum inference and maximum defuzzifier.

with different types of inference rule and membership function was studied. The channel considered for this is $H_{\mathrm{ch}3}(z) = 0.407 - 0.815z^{-1} - 0.407z^{-2}$. The BER performance of three types of fuzzy equalizers were compared with the Bayesian equalizer. The equalizer performance was evaluated with knowledge of the channel states and noise statistics. The scalar centers of the equalizer were placed at ± 1.629, ± 0.815 and ± 0.001. The BER performance of the equalizers is presented in Fig. 6. Fuzzy-1 refers to the fuzzy equalizer (6) with product inference and COG defuzzifier, Fuzzy-2 refers to the equalizer (11) operating with minimum inference and COG defuzzifier and the Fuzzy-3 (11) works with minimum inference and maximum defuzzifier. From their performance curves, it is seen that all the fuzzy equalizers perform close to the Bayesian equalizer. The fuzzy equalizers operating with the minimum inference and maximum defuzzifier also perform close to the Bayesian equalizer with a large reduction in the computational complexity. This validates our assumption of the minimum inference rule and maximum defuzzification process.

In the second phase of the experiment, the equalizer was trained with both training phases. The channel used for this experiment was $H_{\mathrm{ch}4}(z) = 0.3482 + 0.8704z^{-1} + 0.3483z^{-2}$. First the scalar channel states and the channel noise statistics were determined with supervised clustering. The clustering was based on 200 training samples and the result was averaged over 50 experiments. From the estimated scalar channel states, the equalizer membership functions were generated and the fuzzy inference rules were created. Next the weights associated with the inferences were fine tuned with the LMS algorithm to compensate for the effect of the error in channel states estimation. This constituted 500 training samples. The BER performances of the equalizers are presented in Fig. 7. It is seen from the results that the equalizers with the estimated channel states performed close to the optimal Bayesian equalizer. Here the Fuzzy-1 equalizer works with product inference and COG defuzzifier (6). The Fuzzy-2 and Fuzzy-3 employ modified membership function generation discussed in the previous section. Fuzzy-2 works with product inference with COG defuzzifier and Fuzzy-3 works with minimum inference and maximum defuzzifier. In both of these equalizers, only two of the membership functions for scalar channel states closest to input scalar $r(k)$ were considered. From the performance curves it is seen that the fuzzy equalizer with all channel states (6) performs close to Bayesian equalizer. When a subset of the channel states is selected by changing the membership function generator, there is a performance degradation. This performance degradation is around 3 dB at 10^{-3} BER. The performance of the substate state equalizer is slightly inferior to the linear equalizer when the SNR is below 11 dB but much better than the linear equalizer above 11-dB SNR. The performance drop at low SNR condition is due to the fact that at low SNR conditions the Gaussian spread parameter is large and the membership functions from the scalar channel states far from the input scalar would have more contribution on inference rule compared to high SNR condition when the spread parameter is small.

VI. Conclusion

The relationship between the optimal Bayesian equalizer and the fuzzy equalizer has been demonstrated. This relationship reduces the computational complexity of the latter and leads to fast training algorithms. Further computational advantages have been achieved by modification to the inference rule and defuzzification techniques. The relationship between the two networks has also led to an elegant scheme for state selection which provides its own computational advantages.

References

[1] G. D. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 363–378, May 1972.
[2] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1971.
[3] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[4] S. U. H. Quereshi, "Adaptive equalization," in *Proc. IEEE*, vol. 73, pp. 1349–1387, Sept. 1985.
[5] S. Haykin, *Neural Networks—A Comprehensive Foundation*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[6] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Adaptive equalization of finite nonlinear channels using multilayer perceptron," *Signal Processing (EURASIP)*, vol. 20, pp. 107–119, 1990.
[7] M. Meyer and G. Pfeiffer, "Multilayer perceptron based decision feedback equalisers for channels with intersymbol interferance," *Proc. Inst Elect. Eng.*, vol. 140, pp. 420–424, Dec. 1993.
[8] J. Moody and C. J. Darken, "Fast learning in networks for locally tuned processing units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989.
[9] B. Mulgrew, "Appling radial basis functions," *IEEE Signal Processing Mag.*, vol. 13, pp. 50–65, Mar. 1996.
[10] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithms for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, pp. 302–309, Mar. 1991.
[11] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communication channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 570–579, July 1993.
[12] L.-X. Wang and J. M. Mendel, "Fuzzy adaptive filters, with application to nonlinear channel equalization," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 161–170, Aug. 1993.
[13] P. Sarwal and M. D. Srinath, "A fuzzy logic system for channel equalization," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 246–249, May 1995.
[14] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive Bayesian equalizer with decision feedback," *IEEE Trans. Signal Processing*, vol. 41, pp. 2918–2927, Sept. 1993.
[15] T. Poggio and F. Girosi, "Network for approximation of learning," *Proc. IEEE*, vol. 78, pp. 1481–1497, Sept. 1990.
[16] I. Cha and S. A. Kassam, "Interference cancellation using radial basis function network," *Signal Processing (EURASIP)*, vol. 47, pp. 247–268, Dec. 1995.
[17] K. Y. Lee, "Complex fuzzy adaptive filter with LMS algorithm," *IEEE Trans. Signal Processing*, vol. 44, pp. 424–429, Feb. 1996.
[18] ——, "Complex RLS fuzzy adaptive filter and its application to channel equalization," *Electron. Lett.*, vol. 30, pp. 1572–1574, Sept. 1994.
[19] ——, "Fuzzy adaptive decision feedback equaliser," *Electron. Lett.*, vol. 30, pp. 749–751, May 1994.
[20] D. G. Oh, J. Y. Choi, and C. W. Lee, "Adaptive nonlinear equalizer with reduced computational complexity," *Signal Processing (EURASIP)*, vol. 47, pp. 307–317, Dec. 1995.
[21] L.-X. Wang and J. M. Mendel, "Fuzzy basis function, universal approximation and orthogonal least square learning," *IEEE Trans. Neural Networks*, vol. 3, pp. 807–814, Sept. 1992.
[22] J. S. R. Jang and C. T. Sun, "Functional equivalance between radial basis function networks and fuzzy inference systems," *IEEE Trans. Neural Networks*, vol. 4, pp. 156–159, Jan. 1993.
[23] H. M. Kim and J. M. Mendel, "Fuzzy basis functions: Comparisons with other basis functions," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 158–167, May 1995.
[24] S. K. Patra and B. Mulgrew, "Computational aspects for adaptive radial basis function equalizer design," in *Proc. IEEE Int. Symp. on Circuits and Systems*, Hong Kong, June 1997, vol. 1, pp. 521–524.
[25] S. Chen, S. McLaughlin, B. Mulgrew, and P. M. Grant, "Adaptive Bayesian decision feedback equalizer for dispersive mobile radio channels," *IEEE Trans. Commun.*, vol. 43, pp. 1937–1946, May 1995.
[26] E.-S. Chng, H. Yang, and W. Skarbek, "Reduced complexity implementation of Bayesian equalizer using local RBF network for channel equalization problem," *Electron. Lett.*, vol. 32, pp. 17–19, Jan. 1996.

Sarat Kumar Patra (S'98) received the B.Sc. (Eng.) and M.Sc. (Eng.) degrees from UCE Burla, Sambalpur University, India, and REC Rourkela, India, in 1986 and 1991 respectively. He is currently working toward the Ph.D. degree at the University of Edinburgh, Scotland, U.K.

Since 1989 he has been with the Department of Applied Electronics & Instrumentation Engineering, REC Rourkela. His research interests are in adaptive signal processing, fuzzy systems, and neural networks and their application to communication and signal processing.

Mr. Patra is a life member of the Indian Society of Technical Education, India, and the Institution of Electronic and Telecommunication Engineers, India.

Bernard Mulgrew (M'88) received the Ph.D. degree in 1987 from the University of Edinburgh, Scotland, U.K.

In 1979 he joined Radar Systems Department at Ferranti Defence Systems, Edindurgh. From 1983 to 1986, he was a Research Associate in the Department of Electrical Engineering at the University of Edinburgh, studying the performance and design of adaptive filter algorithms. He is currently a reader in that department, teaching courses on signals and systems and signal processing. His research interests are in adaptive signal processing and estimation theory and their application to radar and communication systems. He is a co-author of two books on signal processing.

Dr. Mulgrew is a member of the Institution of Electrical Engineering (U.K.), EURASIP, and the Audio Engineering Society.