Computational Aspects of Adaptive Radial Basis Function Equalizer Design

Sarat Kumar Patra, *Bernard Mulgrew*
Dept of Electrical Engineering, University of Edinburgh, Edinburgh, EH9 3JL, UK.
Tel: +44 131 6505655; fax: +44 131 650 6554
e-mail: skp@ee.ed.ac.uk

ABSTRACT

This paper investigates the computational aspects of radial basis function (RBF) equalizers. In an RBF implementation of the Bayesian equalizer the RBF centers are placed at equalizer channel states and the output layer weights are adjusted to +1/-1. Here we propose an RBF equalizer with scalar centers which can implement the Bayesian decision function. The proposed RBF equalizer provides lower computational complexity compared to the reported RBF equalizers and can efficiently employ subset center selection for computing the decision function resulting in a substantial reduction in computational complexity.

1 Introduction

The speed of transmission of information over a communication system is limited due to the effects of intersymbol interference (ISI) and additive noise. The process of removing these effects from the received signal to faithfully generate the transmitted information at the receiving end is called equalization. The structure of this communication system is depicted in Figure 1. The information symbol to be transmitted $s(k)$ is transmitted through a linear dispersive channel described by:

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

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 received signal vector $\mathbf{r}(k) = [r(k), r(k-1), ..., r(k-m+1)]$. Here m is the order of the equalizer. Normally a delay is associated with detection and hence the equalizer output is a delayed form of the transmitted sequence and can represented as $\hat{s}(k-d)$.

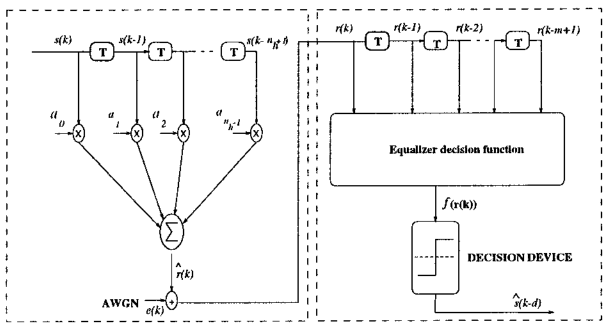

Figure 1: *A communication system*

2 Optimal Symbol Decision Equalizer

The general symbol decision equalizer depicted in Figure 1 is characterised by equalizer order m and delay d. The optimal decision function for this equalizer can be represented as [1]

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

Here σ_e^2 represents the channel noise variance, c_i^+ and c_i^- 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. Here it is assumed that the transmitted symbol $s(k)$ is binary taking the value from $+1/-1$. This equation can also be presented as :

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

Here n_s is the number of channel states equal to 2^{n_h+m-1}, p_i are the weights associated with each of the centers. p_i is +1 if $\mathbf{c}_i$ correspond to a positive channel state and -1 if it represents a negative channel state. It is also observed that each of the channel state vector has m components. We can represent any channel state c_i as $\mathbf{c}_i = [c_{i0}, c_{i1}, c_{i2}, ..., c_{i(m-1)}]$. Rewriting the

*On leave from Dept. of AE&IE, REC Rourkela, Orissa - 769 008, India squared norm in Eqn(3) as a summation and exploiting the properties of the exp function yields:

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

Here 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 (2) and (3) provide alternate realisations of the Bayesian decision function. In (3) the Euclidian distance between input vector $\mathbf{r}(k)$ and each of the channel state $\mathbf{c}_i$ 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. Alternately in (4), scalar distances are calculated, multiplied by $-1/(2\sigma_e^2)$ and exponential function evaluated. The products of exponential functions associated with particular channel states are linearly combined to provide the decision function. Both of these equations require the knowledge of channel states for estimating the decision function. How ever (4) could be preferred for actual implementation [2].

A similar argument can be applied to the normalised Bayesian equalizer of Cha and Kassam [3] which forms an estimate of the transmitted symbol themselves rather than a decision function. This, we represent as a normalised Bayesian equalizer with scalar states(NBEST) shown in eqn.5.

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

It is seen that the equalizer represented in eqn.(3) can be implemented with a radial basis function (RBF)[4] and the equalizer represented by eqn.(5) can be implemented as a combination of a set of fuzzy basis functions [5] with singleton fuzzifier, product inference, Gaussian membership function and centroid defuzzifier. In RBF implementation the RBF centers are placed at the equalizer channel states and the linear weights of the output layer are equated to p_i. Here we implement the decision function of (5) as a normalised radial basis function with scalar centers (NRBF_SC).

3 Channel States Estimation

From the previous section it has been observed that knowledge of the channel states is essential for evaluation of the optimum decision function for the equalizer. The channel state estimation needs the knowledge of the channel. But under most circumstances knowledge of the channel may not be available. Under these circumstances the channel states can be estimated during the training period when the transmitted symbols are known to the receiver. This can be achieved in two ways [1].

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

Table 1: *The channel states calculation*

- The channel model can be estimated using some algorithms like least mean square (LMS). With the knowledge of the channel its staight forward to calculate the channel states. This technique may fail if the channel is non-linear in nature.
- The channels states can be directly calculated based on some clustering algorithm[4]. The computational complexity of this could be very high if the order of the channel states is large and the convergence time for this is also large.

The channel states can be computed from the scalar channel states. The scalar channel states refers to the possible noise free received samples. The scalar states can be calculated by a clustering algorithm. Calculation of the scalar channel states is simple and computational complexity for this is independent of the order of the equalizer. These scalar states can be suitably combined to form the vector states[6]. Once the channel state vectors have been estimated finding the decision function of the equalizer is straightforward. We take an example to illustrate the relationship of scalar and vector channel states. Table 1 provides the channel state calculation for a equalizer of order $m = 2$, delay $d = 0$. The channel transfer function is $H_{ch}(z) = 0.5 + z^{-1}$. Here n_h is 2. Following observations are made from the channel state calculation:

- There are 2^{n_h+m-1}=8 vector channel states which can be represented as $[\hat{r}(k), \hat{r}(k-1)]$.
- There are 2^{n_h}=4 possible scalar channel states which correspond to each of the elements of $\hat{r}(k)$ or $\hat{r}(k-1)$.
- The weights p_i of the decision functions eqns.3 and 4 assume the value $+1$ or -1 for positive and negative states 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 state. The decision function can be obtained by suitable adjustment of the parameter p_i for the states that have changes from positive to negative states or vice-verse.

4 Normalised Radial Basis Function Equalizer with Scalar Centers

A NRBF_SC equalizer with scalar channel states represented by eqn.5 is shown in Figure 2. Here the incoming signal sample is presented to the basis function generator. Each of the component of the basis function generator produces an output S_{ij}, characterised by its scalar center c_{sj} which are placed at the scalar channel states. Here i corresponds to the equalizer input number and $i = 0$ for $r(k)$. The value of j ranges from 0 to $(M-1)$ where M is the number of the scalar center and $M = 2^{n_h}$. If some of the tap weights of the channel are same $M \leq 2^{n_h}$. Each of the component of the basis function generator compute the following function with the received signal $r(k)$ and its center.

$$S_{ij}(r(k)) = \exp\left(-\frac{\|r(k) - c_{sj}\|^2}{2\sigma^2}\right)$$

The basis function generator for the other $(m-1)$ inputs to the equalizer are not needed as their output would be the delayed output of the basis function generator for $r(k)$. The corresponding output for $r(k-1)$ which can be generated by delaying $S_{00}, S_{01}, ...S_{0(M-1)}$ can be represented as $S_{10}, S_{11}, ...S_{1(M-1)}$. Similarly the output for $r(k-m+1)$ would be $S_{(m-1)0}, S_{(m-1)1}, ...S_{(m-1)(M-1)}$. The product block of the NRBF_SC has n_s sub-blocks. Each of these sub-blocks receive only one of the S_{ij} corresponding to each of the m inputs to the equalizer. The specific element corresponding to each input is selected by the combination of the scalar centers which construct the specific vector states. The output of the product sub-blocks corresponding to the +ve channel states are added to provide a and the -ve channel state are added to provide b. The output of the NRBF_SC is computed by the function $(a-b)/(a+b)$. The output of the NRBF_SC passed through the sigmoid nonlinearity forms the detected sample. If it is not possible to find to which group of states a channel state belongs the weights p_i can be estimated by the LMS algorithm.

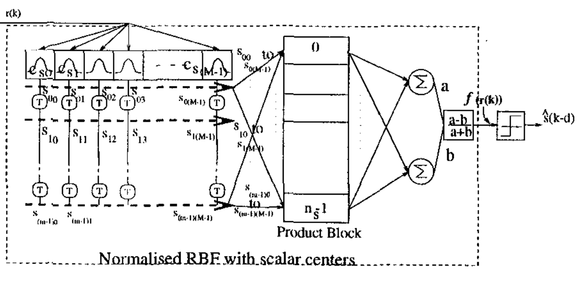

Figure 2: *NRBF_SC Equalizer*

We consider the previously discussed example to illustrate the working of this equalizer. Here again the equalizer order is assumed to be $m = 2$ for simplicity. As per Table 1 the scalar centers for the basis function block are placed at +1.5, +0.5, -0.5 and -1.5 which correspond to the values of $c_{s0}, c_{s1}, c_{s2}, c_{s3}$. The basis function output for the input $r(k-1)$, will be $S_{10}, S_{11}, S_{12}, S_{13}$ and these can be generated by delaying the functions $S_{00}, S_{01}, S_{02}, S_{03}$ corresponding to $r(k)$. The multiplier block will con-

Equalizer Type	Add/ Sub	Mul	Div.	e^{-x}
RBF	$2mn_s$	$(m-1)n_s$	n_s	n_s
NRBF_SC	$M + n_s$	$M + (m-1)n_s$	$M + 1$	M
NRBF_SC (Sub-Set)	$2^m + 2$	$(m-1)2^m$	3	2

Table 2: *Computational Complexity Comparison*

sist of $n_s = 8$ sub-blocks. The sub-blocks 1 to 8 compute the products $S_{00}S_{10}$, $S_{00}S_{12}$, $S_{02}S_{11}$, $S_{02}S_{13}$, $S_{01}S_{10}$, $S_{01}S_{12}$, $S_{03}S_{11}$, $S_{03}S_{13}$ respectively. The products $S_{00}S_{10}$, $S_{00}S_{12}$, $S_{02}S_{11}$, $S_{02}S_{13}$ are added to provide a and $S_{01}S_{10}$, $S_{01}S_{12}$, $S_{03}S_{11}$, $S_{03}S_{13}$ are added to provide b. The calculation of the decision function is straight-forward. In the next section we discuss the advantages of the NRBF_SC equalizer over the conventional RBF equaliser.

5 Advantages of Normalised Radial Basis Function with Scalar Centers

We have seen in the previous sections that the NRBF_SC provides the same decision function as the RBF implementation of Bayesian equaliser. But it can be seen from the decision functions that NRBF_SC can provide the same decision function with less computational complexity per sample calculation. The computations demanded by the two equalisers is presented in Table 2. These can be summarised as under.

- The number of addition in NRBF_SC is $M + n_s$ compared to $2mn_s$ in a RBF.
- The number of multiplications in NRBF_SC are slightly higher than the RBF but this is compensated by the reduction in the number of divisions.
- The number of $\exp(-x)$ evaluations is reduced considerably in the NRBF_SC compared to the RBF.
- The parameter n_s is related to the equalizer order in an exponential term but M is independent of the equalizer order. Hence with the increase in the equalizer order the reduction in computational complexity for NRBF_SC over the RBF equalizer can be exponentially related.
- It is straight forward to employ subset center selection with scalar centers compared to the equalizer with vector centers. This is described in details in the next sub-section.

5.1 Subset Center Selection

On observing the decision function of the RBF implementation of Bayesian equalizer in eqn.(3) it is seen that the contribution of RBF centers to the decision function is inversely related to its distance from the input vector. If the centers nearer to the received signal vector can be found the computation involved in calculating the decision function can be greatly reduced without noticeable drop in the performance criterion, by neglecting the RBF centers far from the received vector. In a multidimensional signal space it is not efficient to find the nearby channel states and the scheme may not provide sufficient reduction in the computations involved. On the other hand, the decision function provided by NRBF_SC by selecting a subset of the total centers will not be difficult as the centers are in one-dimensional space. In this case if we select only 2 scalar centers that are nearest to the received samples $r(k)$ the reduction in the computational requirement can be enormous with small degradation in the performance. In the next section we have shown by simulations that the performance degradation for the equalizer would be small compared to the reduction in the computational requirement. The computation required by subset center NRBF_SC equalizer to compute each sample is also presented in Table 2. From this it is evident that there is a substantial reduction in the computational complexity for the NRBF_SC with a sub-set center selection.

6 Simulation Results

In order to demonstrate the performance of the NRBF_SC equalizer proposed in this paper following simulations were carried out. In all the tests $s(k)$ was an equiprobable random binay number taking the value from $+1/-1$. The channel considered for the simulation studies has a transfer function $H_{ch} = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$. The equalizer order and the detection delay were chosen to be 2 and 0 respectively. The bit error rate (BER) was chosen as the performance criteria in all cases. The results of the simulations are presented in Figure 3. Here the Bayesian equalizer was simulated with knowledge of the channel and the channel states. The weights p_i of eqn.3 were taken as +1/-1 as per the channel states. The NRBF_SC equalizer was assumed to have the proper knowledge of the channel and the scalar channel states and also the combinations of the scalar states which provide the vector states. The weights of the NRBF_SC equalizer were trained by LMS algorithm with the initial weights being 0. In the third case the NRBF_SC with subset of its centers was considered for simulation. Here channel knowledge was not available and the scalar channel states were generated from the received sample during the training period. Additionally the weights p_i were also trained by LMS algorithm. For all inputs only the two nearest scalar centers of the NRBF_SC to the received sample were considered to be active and the contribution from all the other centers were neglected. From the simulation curves it is seen that the performance of the proposed equalizers are close to performance of the Bayesian equalizers. The small performance degradations can be attributed to the limitations of the LMS algorithm used for weight update. The performance degradation was nearly 1dB at 20dB SNR.

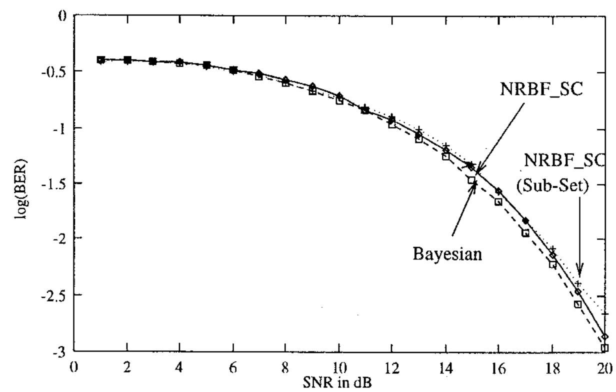

Figure 3: *Performance of different equalizers*

7 Conclusion

Here we have proposed a normalised RBF equalizer with scalar centers which provides a advantage over the conventional RBF equalizer with vector centers in terms of computational complexity. We have also shown how this NRBF_SC can be used with a subset of its centers. We have demonstrated that the performance of this equalizers is close to the Bayesian equalizer with vector channel states.

References

[1] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive bayesian equalizer with decision feedback," *IEEE Transactions on Signal Processing*, vol. 41, pp. 2918–2927, September 1993.

[2] T. Poggio, "Network for approximation and learning," *Proceedings of the IEEE*, vol. 78, pp. 1481–1497, sep 1990.

[3] I. Cha and S. A. Kassam, "Interference cancellation using radial basis function network," *Signal Processing (Eurasip)*, vol. 147, pp. 247–268, December 1995.

[4] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communication channel equalization using radial-basis-function networks," *IEEE Transactions on Neural Networks*, vol. 4, pp. 570–579, July 1993.

[5] L.-X. Wang and J. M. Mendel, "Fuzzy basis function, universal approximation and orthogonal least-square learning," *IEEE Transactions on Neural Networks*, vol. 3, pp. 807–814, September 1992.

[6] S. Chen, S. M. Laughlin, B. Mulgrew, and P. M. Grant, "Bayesian decision feedback equaliser for overcoming co-channel interference," *IEE Proceedings - Communication*, vol. 143, pp. 219–225, August 1996.