Chapter 6

A Radial Basis Function Framework

In Chapters 3, 4, and 5, we have seen the realization of type–2 FAF, ANFIS, and CNFF based equalizers, respectively. This chapter is attributed to establish a common link between them—that they can be brought under the generic framework of the radial basis function neural network. The Radial Basis Function (RBF) based neural networks have been successfully used to solve many non-linear problems, including that of adaptive equalization. In this context, we present three different adaptive fuzzy/neuro-fuzzy channel equalizers that closely fit into the framework of RBF neural network based systems. We consider the type–2 Fuzzy Adaptive Filter (FAF) based channel equalizer along with a Compensatory Neuro-Fuzzy Filter (CNFF) and the one based on an Adaptive Network based Fuzzy Inference System (ANFIS) as applied to mobile cellular channels. We establish that the three implementations of adaptive equalizers do fit into the generic framework of Radial Basis Function (RBF) based systems.

6.1 Introduction

The theory of Radial Basis Functions (RBF) and their application to design channel equalizers are not new [49, 80, 81, 82, 83, 84]. Recently, there has been a new interest in the area, as evident from reference [19]. An important problem in data communications is that of channel equalization, that is, the removal of interference introduced by linear or non-linear message corrupting mechanisms, so that the originally transmitted symbols can be recovered correctly at the receiver [20]. Channel Equalization is an age old problem, ever since the advent of the telephone systems. Equalization is needed for Linear Time-Invariant (LTI) channels like the UTP cable or Coax, or the Linear Time Variant (LTV) channels like the radio channels used in mobile cellular telephony. The mobile cellular channel is time-variant due to multipath fading. The complexity of the equalizer increases as we move from the LTI channels to LTV channels. In the case of LTI channels like the UTP cable or Coax, the equalization is fairly less complex as the
problem is basically that of system identification of an LTI system, and then obtaining the inverse system impulse response. In the case of LTV channels, on the other hand, the problem of equalization is very complex due to the time-variant nature of the system itself. Noise introduced in the channel is yet another issue, which makes the problem more severe. In the case of mobile cellular channels, it is shown by several authors that the channel is linear and time-varying and have either a Ricean fading or Rayleigh fading characteristics [85]. The output SNR is affected by both Co-Channel Interference (CCI), which is present due to frequency re-use, and Adjacent Channel Interference (ACI), a contribution from the spectral leakage among frequencies used in adjacent channels in the cell [26].

The rest of the chapter is organized as follows: In Section 6.2, we review the principles of Radial Basis Functions. We discuss the implementation of the adaptive equalizers in Sections 6.3, 6.4, and 6.5. We make our observations and conclusions in Section 6.6.

6.2 Radial Basis Function Neural Networks

Originally, RBF neural networks are developed for data interpolation in multi-dimensional space [49]. Although the primary reason for using an equalizer on a communication channel has been to mitigate the effects of intersymbol interference, more recently it has been demonstrated that conventional equalizers can exploit the cyclostationary nature of the received signal and reduce the distortion due to both co-channel and adjacent channel interference. Radial basis function network can also be applied to this problem without the need to exploit the cyclostationary characteristics of the received signal. The structure of the RBF neural network is given in Figure 2.7. The use of the RBF has provided receivers with more controllable training characteristics than Multi-Layer Perceptron (MLP) receivers. However, the length of the training period is still too long for practical consideration. Blind equalization, in particular, is a demanding problem that currently receives a great deal of attention. While many techniques have been applied, RBF Bayesian methods have a unique contribution to play in this area, as they explicitly exploit the finite nature of the transmitted alphabet. This is unlike the techniques based on higher order statistics or the cyclostationary nature of the received signal, which are more complex. The RBF NN can accommodate the channel non-linearity by effectively combining a large number of Gaussian basis functions.
6.2.1 Review of Previous Work

The functional equivalence between RBF neural networks and a simplified class of fuzzy inference systems have been made in [84]. This functional equivalence enables us to apply what has been discovered (learning rule, representational power, etc.) for one of the models to the other, and vice versa. It is of interest to observe that two models stemming from different origins turn out to be functionally equivalent. Though these two models are motivated from different origins (RBF networks from physiology and fuzzy inference systems from cognitive science), they share common characteristics not only in their operations on data, but also in their learning process to achieve the desired mappings. We show that under some minor restrictions, they are functionally equivalent; the learning algorithms and the theorem on representational power for one model can be applied to the other, and vice versa [84].

The output of an RBF network can be computed in two ways. For the simpler one, as shown in Figure 2.7, the output is the weighted sum of the function value associated with each receptive field:

\[ f(\bar{x}) = \sum_{i=1}^{N_r} f_i \cdot u_i = \sum_{i=1}^{N_r} f_i \cdot R_i(\bar{x}) \]  

(6.1)

where \( f_i \) is the function value, or strength, of the \( i^{th} \) receptive field.

The functional equivalence between an RBF network and a fuzzy inference system can be established if the following are true [84]:

1. The number of receptive field units is equal to the number of fuzzy if-then rules.
2. The output of each fuzzy if-then rule is composed of a constant.
3. The membership functions within each rule are chosen as Gaussian functions with the same variance.
4. The \( t\text{-norm} \) operator used to compute each rules firing strength is multiplication (or \( product \ t\text{-norm} \)).
5. Both the RBF neural network and the fuzzy inference system under consideration use the same method (i.e., either \( weighted \ average \) or \( weighted \ sum \)) to derive their overall outputs.

Motivation for the Unified Framework

It is apparent, by now, that if we can establish a \( unified \ framework \) for the adaptive equalizers based on neural networks and fuzzy logic, there will be several conveniences. There will be a synergical improvement in performance arising from combining
the best features of both. We will tackle one after another, all the three adaptive equalizers under consideration. Traditional adaptive algorithms for equalizers are based on the criterion of minimizing the mean square error between the desired filter output and the actual filter output, that is, these learning algorithms adjust the filter parameters to achieve a minimum of the criterion. Sheng Chen, et al. have investigated the application of a radial basis function network to digital communications channel equalization [80]. It is shown that the radial basis function network has an identical structure to the optimal Bayesian symbol-decision equalizer solution and, therefore, can be employed to implement the Bayesian equalizer. The training of a radial basis function network to realize the Bayesian equalization solution can be achieved efficiently using a simple and robust supervised clustering algorithm. This represents a radically new approach to the adaptive equalizer design.

6.3 Type-2 Fuzzy Adaptive Filter Equalizer

A channel with a more realistic equalizer order is used to study the performance of the RBF network under a variety of SNR’s. The channel transfer function was given by [12]:

$$H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}.$$  \hspace{1cm} (6.2)

Correct estimates of the channel order and the noise variance are assumed.

The type-2 FAF is realized using an unnormalized type-2 Takagi-Sugeno-Kang (TSK) fuzzy logic system [14, 53]. A clustering method is used to adaptively design the parameters of the FAF. The statistical signal processing-based approach (e.g., Bayesian decision rule) is based on a probability model (e.g., Gaussian distribution), whereas the FAF-based approach is model free. As noted in [59], a shortcoming to model-based statistical signal processing is the assumed probability model, for which model-based statistical signal processing results will be good if the data agrees with the model, but may not be so good if the data does not.

A type-2 TSK FLS is described by fuzzy IF-THEN rules which represent input-output relations of a system [14]. The type-2 TSK FLS has a rule base of $M$ rules, each having $p$ antecedents, where the $i$th rule, $R^i$, is expressed as:

$$R^i : IF \ x_1 \ is \ F^i_1 \ and \ x_2 \ is \ F^i_2 \ and \ ... \ and \ x_p \ is \ F^i_p$$

$$THEN \ y^i = c^i_0 + c^i_1 x_1 + c^i_2 x_2 + \ldots + c^i_p x_p,$$

in which $i = 1, 2, \ldots, M$; $c^i_j (j = 0, 1, 2, \ldots, p)$ are the consequent parameters; $y^i$ is the output of the $i$th IF-THEN rule; and, $F^i_k (k = 1, 2, \ldots, p)$ are type-2 fuzzy sets.
A Simplified Mathematical Formulation for FAF-II

Equation 6.1 gives the output of a radial basis function neural network (RBF NN) in terms of the strengths of the receptive fields. One of the conditions for RBF NN and the fuzzy adaptive filter to be equivalent is that the number of receptive fields \( N_r \) in RBF is equal to the number of fuzzy if-then rules \( (M) \) in FAF. Now the membership function for the \( k^{th} \) input is chosen as Gaussian with the variance \( \sigma_k^2 \) and mean, \( m_k \). It is also assumed that the product t-norm operator is used to compute the firing strength of each rule, and there are \( p \) inputs to the RBF NN and FAF. Then Equation 6.1 transforms to:

\[
\begin{align*}
\tilde{f}(\tilde{x}) &= \sum_{i=1}^{N_r} f_i \times w_{1} = \sum_{i=1}^{N_r} f_i R_i(\tilde{x}) \\
&= \sum_{i=1}^{M} y_i \times \exp \left[ -\frac{1}{2} \left( \frac{x_1 - m_1}{\sigma_1} \right)^2 \right] \times \exp \left[ -\frac{1}{2} \left( \frac{x_2 - m_2}{\sigma_2} \right)^2 \right] \ldots \times \prod_{k=1}^{p} \exp \left[ -\frac{1}{2} \left( \frac{x_k - m_k}{\sigma_k} \right)^2 \right] \\
&= \sum_{i=1}^{M} y_i \prod_{k=1}^{p} \exp \left[ -\frac{1}{2} \left( \frac{x_k - m_k}{\sigma_k} \right)^2 \right] = y
\end{align*}
\]

(6.3)

It can be observed that (6.3) is identical to the output formula for a Radial Basis Function (RBF) network when Gaussian membership functions are used as the RBFs. This kind of RBF network has been applied to Bayesian equalization.

6.4 Compensatory Neuro-Fuzzy Filter Equalizer

The large computational complexity associated with the Viterbi algorithm and the poor performance of the linear equalizers has led to the development of symbol-by-symbol equalizers using the Maximum a Posteriori probability (MAP) principle-Bayesian equalizers. The compensatory fuzzy reasoning method is used in adaptive fuzzy operations that can make the fuzzy logic system more adaptive and effective. Besides, the Pseudo-Gaussian (PG) membership function can provide the compensatory neuro-fuzzy filter higher flexibility and it can approach the optimized result more accurately. An on-line learning algorithm, which consists of the structure learning and the parameter learning, is proposed. The structure learning is based on the similarity measure of asymmetry Gaussian membership functions and the parameter learning is based on the supervised gradient decent method. We apply the proposed CNFF for co-channel interference suppression (CCI) and additive with Gaussian noise (AWGN) filtering. Computer simulation results show that the bit error rate of the CNFF is close to
The Bayesian equalizers have been approximated using nonlinear signal processing techniques like Artificial Neural Networks (ANN) [66, 67], Radial Basis Functions (RBF) [80], recurrent neural networks, and fuzzy filters [8, 54]. These new techniques provide advantages of both good performance and low computational cost. Fuzzy filters are nonlinear filters that incorporate linguistic information in the form of IF-THEN fuzzy rules. Fuzzy filters have been used for equalization due to their success in the related area of pattern classification [53]. Wang and Mendel [54] have presented Fuzzy Basis Functions (FBF) for channel equalization. Lin and Juang [70] have developed the Artificial Neuro Fuzzy Filters (ANFF) and use it for equalization and noise reduction. This ANFF constructs its rule base in a dynamic way with the training samples. Patra and Mulgrew [71] derived the close relationship between the fuzzy equalizers and the equalizer based on Maximum a Posteriori probability principle (MAP). Liang and Mendel [53] have developed type-2 fuzzy adaptive filters and demonstrated that it can implement the Bayesian equalizer.

The Compensatory Neuro-Fuzzy Filter (CNFF), which can be constructed by learning from training examples, can be contrasted with the traditional fuzzy logic control systems in their network structure and learning ability. The CNFF is a four-layer structure (see Figure 5.1). Nodes at layer one are input nodes (linguistic nodes) which represent input linguistic variables. Layer four is the output layer. Nodes at layer two are term nodes which act as members of the respective linguistic variable. Each node at layer three is a compensatory rule node, which first explored systematically by Takagi and Sugeno, has found numerous practical applications in control, prediction and inference [24]. However, there are some basic aspects of this approach which represents one fuzzy logic rule. Thus all the layer-three nodes form a fuzzy rule base [20]. Besides, the compensatory fuzzy reasoning method is used in adaptive fuzzy operations that can make the fuzzy logic system more adaptive and effective.

The compensatory operation can map the pessimistic input $x_1$ and the optimistic input $x_2$ to make the relatively compromised decision for the situation between the worst case and the best case. For example, $c(x_1, x_2) = x_1^{\gamma} x_2^{1-\gamma}$, where $\gamma \in [0, 1]$ is called the compensatory degree [20]. Many researchers have used the compensatory operation to fuzzy systems successfully.

Nonlinear channel equalization is a technique used to combat some imperfect phenomenon in high-speed channel [20]. The transmission input signal $s(k)$ is a sequence of statistically independent random binary symbols taking values $s(k) \in \{-1, 1\}$. The equalizer uses an input receiver signal vector $e(k) \in \mathcal{R}^m$, the $m$ dimensional space,
then the channel function can be described as:

\[ \hat{r}(k) = f[s(k), s(k-1), \ldots, s(k-N)] \]  

(6.4)

In general, \( f \) is a nonlinear function of the past transmitted signal, and the channels change slowly but significantly over time, so a nonlinear channel equalizer with adaptation ability is needed. At the receiving end, the observed signal \( x(k) \) is the channel output \( \hat{r}(k) \) corrupted by additive noise \( e(k) \), that is: \( x(k) = \hat{r}(k) + e(k) \). The noise source \( e(k) \) is assumed to be zero mean white Gaussian with a variance of \( \sigma^2_e \). The task of the equalizer is to reconstruct the transmitted signal \( s(k-d) \) from the observed information sequence \( x(k), x(k-1), \ldots, x(k-N+1) \) (where \( d \) and \( N \) denoted the lag and order, respectively) such that greater speed and higher reliability can be achieved.

### A Mathematical Formulation of CNFF

In a communication channel with AWGN (with zero mean and variance \( \sigma^2 \)), but no CCI, the decision function of Bayesian equalizer is

\[
f(x(k)) = \sum_{i=1}^{n_s} \prod_{l=1}^{n-1} f_i \exp\left(-\frac{1}{2} \frac{|x(k-l) - \hat{r}(k-l)|^2}{\sigma_i^2}\right)
\]

\[
= \sum_{i=1}^{n_s} f_i \prod_{l=1}^{n-1} \exp\left(-\frac{1}{2} \frac{|x(k-l) - \hat{r}(k-l)|^2}{\sigma_i^2}\right)
\]

(6.5)

where \( f_i \) equals either +1 or -1 as determined by the channel state category. It is clear from (6.5) that the CNFF output is similar to that obtained from an RBF NN, as given by Equation (6.1). With CCI, the numerator of equation (6.5) gets modified as:

\[
y = f(x(k)) = \sum_{i=1}^{n_s} f_i \sum_{m=1}^{n_s} \prod_{l=1}^{n-1} \exp\left(-\frac{1}{2} \frac{|x(k-l) - \hat{r}_m(k-l)|^2}{\sigma_e^2 + \sigma_{co}^2}\right)
\]

(6.6)

where \( \hat{r}_m(k-l) \) is the \( l^{th} \) element of the \( m^{th} \) co-channel state.

### 6.5 ANFIS Based Channel Equalizer

System modeling based on conventional mathematical tools (e.g., differential equations) is not well suited for dealing with ill-defined and uncertain systems. By contrast, a fuzzy inference system employing fuzzy if–then rules can model the qualitative aspects of human knowledge and reasoning processes without employing precise quantitative analyses. Fuzzy if–then rules or fuzzy conditional statements are expressions of the form \( IF A THEN B \), where A and B are labels of fuzzy sets, characterized by
appropriate membership functions. Due to their concise form, fuzzy if-then rules are often employed to capture the imprecise modes of reasoning that play a central role in the human ability to make decisions in an environment of uncertainty and imprecision. Another form of fuzzy if-then rule, proposed by Takagi and Sugeno, has fuzzy sets involved only in the premise (or antecedent) part. However the consequent part is described by a nonfuzzy equation of the input variable. Both types of fuzzy if-then rules have been used extensively in both modeling and control. Through the use of linguistic labels and membership functions, a fuzzy if-then rule can easily capture the spirit of a rule of thumb used by humans [24]. The steps of fuzzy reasoning (inference operations upon fuzzy if-then rules) performed by fuzzy inference systems are described in Section 4.2 of Chapter 4.

A Mathematical Formulation of ANFIS Equalizer

We make use of a type-3 ANFIS with Gaussian membership functions for the channel depicted in Figure 4.4. The estimate of the channel output in this case can be expressed as:

\[
y = \sum_{i=1}^{n_s} w_i f_i = \sum_{i=1}^{n_s} w_i f_i \sum_{i=1}^{n_s} w_i
\]

\[
= \sum_{i=1}^{n_s} f_i \times \exp \left[ -\frac{1}{2} \left( \frac{r_i - m_{z_i}}{\sigma_{z_i}} \right)^2 \right] \times \exp \left[ -\frac{1}{2} \left( \frac{x_{co1}}{\sigma_{co1}} \right)^2 \right] \times \exp \left[ -\frac{1}{2} \left( \frac{x_{co2}}{\sigma_{co2}} \right)^2 \right] \cdots \sum_{i=1}^{n_s} w_i
\]

\[
= \sum_{i=1}^{n_s} f_i \times \exp \left[ -\frac{1}{2} \left( \frac{z_i - m_{z_i}}{\sigma_{z_i}} \right)^2 \right] \times \prod_{k=1}^{p} \exp \left[ -\frac{1}{2} \left( \frac{x_{k} - x_{co_k}}{\sigma_{co_k}} \right)^2 \right]
\]  

where \( w_i = \exp \left[ -\frac{1}{2} \left( \frac{z_i - m_{z_i}}{\sigma_{z_i}} \right)^2 \right], \) \( i = 1, 2, \ldots, n_s \). Note that \( \overline{w}_i = \frac{w_i}{\sum w_i} \) is the normalized weight of \( f_i \). The equation (6.7) shows that the ANFIS equalizer too can be brought under the generic framework of RBF network. It may be noted that the ANFIS accommodates non-linearity by a convex combination of linear partitions.

Simulations

Two comparative plots obtained after simulations, are given in Figures 6.1 and 6.2. Figure 6.1 is obtained based on 100 Monte-Carlo (MC) simulations. The inputs
to the equalizers have both CCI and AWGN components, apart from the channel component. However, the standard deviation of the CCI signal is fixed at 0.18. In the first simulation, the variance of the AWGN is changed in accordance with the relation:

\[ \sigma_n = \frac{\sigma_s}{10^{\frac{SNR}{10}}} \]

(6.8)

where \( SNR = 15:0.5:20 \text{dB} \), and \( \sigma_n \) and \( \sigma_s \) are variances of AWGN and the signal (the signal being a random binary waveform) respectively. Note that variances of all co-channels are set to the same value. Identical random signal inputs are given to a Radial Basis Function Neural Network (RBF NN) based equalizer and an ANFIS-27 based equalizer, which has two inputs and 7 membership functions. The spread factor of the
CHAPTER 6. A RADIAL BASIS FUNCTION FRAMEWORK

Figure 6.2: Performance of RBF NN and ANFIS-25: (a) Mean BER at output of the equalizer versus SNR in dBs, (b) Variance of BER versus SNR in dBs, (c) Standard Deviation of BER versus SNR in dBs, and (d) Standard Deviation of BER versus Standard Deviation of AWGN.

RBF NN was chosen as 1. In Figure 6.1(a), the output mean BER is plotted against SNR of the channel output. In Figure 6.1(b) and (c), variance and standard deviation of BER are plotted against SNR for the same simulation setup. In Figure 6.1(d), standard deviation of BER is plotted against standard deviation of AWGN. In Figure 6.2(a) to (d), the results of simulations on the same RBF NN (with spread 0.2) based equalizer and an ANFIS-25 based equalizer, under identical input conditions are given. The slightly superior performance of ANFIS-27 compared to ANFIS-25 based equalizer as regards the mean BER performance is attributed to the more number of rules (49) for the former, as against 25 for the latter. Therefore more precise system approximation is achieved by ANFIS-27. But this is achieved at the cost of more time for convergence.
6.6 Conclusion

We have shown that all the three Neuro–Fuzzy Equalizers discussed in previous sections, fall into the generic framework of RBF Neural Networks. Simulation results also indicate that the response of the RBF NN based equalizer is comparable to that of ANFIS based equalizer. However, as it is shown by Figures 6.1 and 6.2, when the SNR is low, both ANFIS–25 and ANFIS–27 based equalizers slightly outperform the RBF NN, as far as the mean of BER is concerned. This is due to the more complex structure of the ANFIS. When the SNR is above \(18.5\, \text{dB}\), both types of equalizers perform identically. Again, for low values of standard deviation of AWGN, the performance of RBF NN equalizer is more or less the same, as that of the ANFIS–25/ANFIS–27 equalizer, with respect to the standard deviation of BER. But, at high values of standard deviation of AWGN, ANFIS equalizer performance is better. In the case of ANFIS–25 versus RBF NN, the performances are almost identical, as is evident from Figure 6.2(a) and (b). This shows that when the number of nodes is low, (here it is 75), performance of ANFIS is identical to that of RBF NN.

The advantages in bringing in all the three equalizers, viz. FAF-II, ANFIS, and CNFF based equalizers into the generic framework of Radial Basis Function are the following:

1. The concept of Radial Basis Function Neural Network (RBF NN) is well known and hence optimization of the equalizer parameters is easy.

2. It is of great interest to investigate the performances of all the three, and arrive at a particular solution which is most suited for a particular application scenario.

3. It is easier to arrive at the optimal, Bayesian equalizer solution, if we can bring in the generic framework.