Chapter 7

A Modular Approach to Channel Equalization

The mobile cellular channel is generally considered to be Linear and Time-Variant (LTV). However, under limiting conditions, it can even be modeled as Linear and Time-Invariant (LTI). This approach is very convenient as it simplifies many problems associated with the mobile cellular communication systems design such as equalizer design (which is in effect a system identification problem). In this conventional approach of modeling the mobile cellular channel as LTI or LTV, we assume that the transmitter subsystem, including channel encoder and modulator, is linear. Then only, the notion of a linear time-invariant/variant channel holds good. But this is not the case often enough: Transmitter amplifiers are over driven to their nonlinear region to have better power efficiency or to increase the transmitted power so as to increase the SNR [86, 87]. One way to model the transmitter nonlinearity is to shift it to the channel. The resulting nonlinear channel can easily be modeled and compensated by incorporating a nonlinear channel equalizer, which can be the cascade of a non-linear preprocessor filter and an LTV equalizer. In such a case, the equalizer needs to tackle only the nonlinearities inherent to the channel. We propose such a scheme in this chapter. Simulations indicate that this modular approach of shifting the nonlinearity to the preprocessor from the inherently linear channel is indeed a good proposition.

7.1 Introduction

Mobile Cellular channels and the equalizer design for them, is a topic of intensive research activity. Most of the models used for mobile cellular channels assume a linear time-invariant/variant channel. Table 7.1 lists out typical discrete LTI channel transfer functions. It may be noted that the discrete transfer functions listed out in Table 7.1 are used in linear time-invariant channel modeling. The table considers only
upto second order channels. The impulse response coefficients mentioned in Table 7.1 will be time varying about their nominal values, in the case of LTV channels [14, 53].

Modeling of a LTI system is fairly simple. Consequently, equalization of a LTI channel is also simple: it merely involves finding the inverse system, such that the overall impulse response function of the cascade of channel and equalizer results in a delayed unit impulse, $\delta(n - d)$. In the case of LTV channels, equalization is more complex, as the channel parameters constantly change with respect to time. As mentioned in reference [56], even in the case of linear channel, a non linear equalizer is better due to the following reasons:

1. There exist several nonlinear equalizer techniques which are computationally efficient and provide good results in simulations.

2. The problem of equalization is inherently nonlinear and nonlinear equalizers converge faster.

3. In the presence of noise, and when the channel parameters are randomly varying, non-linear modeling gives better results.

<table>
<thead>
<tr>
<th>Channel</th>
<th>Transfer Function</th>
<th>Channel Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_1(z)$</td>
<td>$0.5 + 1.0z^{-1}$</td>
<td>nonminimum phase</td>
</tr>
<tr>
<td>$H_2(z)$</td>
<td>$0.6 + 0.8z^{-1}$</td>
<td>nonminimum phase</td>
</tr>
<tr>
<td>$H_3(z)$</td>
<td>$1.0 + 0.2z^{-1}$</td>
<td>minimum phase</td>
</tr>
<tr>
<td>$H_4(z)$</td>
<td>$0.2682 + 0.9296z^{-1} + 0.2682z^{-2}$</td>
<td>mixed phase</td>
</tr>
<tr>
<td>$H_5(z)$</td>
<td>$0.5 + 0.81z^{-1} + 0.31z^{-2}$</td>
<td>mixed phase</td>
</tr>
<tr>
<td>$H_6(z)$</td>
<td>$0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$</td>
<td>mixed phase</td>
</tr>
<tr>
<td>$H_7(z)$</td>
<td>$0.6963 + 0.6961z^{-1} + 0.1741z^{-2}$</td>
<td>mixed phase</td>
</tr>
</tbody>
</table>

Table 7.1: Transfer Functions of Linear Time-Invariant Channels

The use of large constellations provides bandwidth efficient modulation. Quadrature Amplitude Modulation (QAM) techniques have constellations, in which signal points are uniformly spread. Information is carried by both signal amplitude and phase; hence they are not constant envelopes. Thus, efficient nonlinear power amplifiers cannot be utilized in the transmitter, without equalization in the receiver.

The rest of the chapter is organized as follows: In Section 7.2, we introduce the non-linear channel model, which is most suitable for the mobile cellular channels. Then in Section 7.3, we consider contemporary non-linear equalizers based on RBF, Multi-Level Perceptrons (MLP), and Fuzzy Adaptive Filters (FAF) used for them. In Section
7.4, we introduce the modular approach. Simulation results for the proposed model are given in Section 7.5. We conclude the Chapter in Section 7.6.

### 7.2 Non-Linear Channel Models

Practical power amplifiers introduce nonlinear distortion in the amplitude and the phase of the transmitted signal. The simple nonlinear model, described by Saleh, is widely used in developing methods to equalize nonlinear channels [30]. This model formulates the amplitude and phase distortion due to a nonlinear amplifier in the transmitter, using two simple two-parameter formulae. Input signal to the nonlinear channel can be written as:

\[
s(t) = a(t) \cos[\omega_c t + \phi(t)]
\]

Here, \( \omega_c \) is the carrier frequency, \( a(t) \) is the modulated amplitude, and \( \phi(t) \) is the modulated phase. The amplitude and phase distortion are functions of the amplitude of the input signal, which are denoted by \( A[a(t)] \) and \( \Phi[a(t)] \) respectively. The output signal after the nonlinear channel is given by:

\[
r(t) = A[a(t)] \cos\{\omega_c t + \phi(t) + \Phi[a(t)]\}
\]

The model describes the distortions \( A[a(t)] \) and \( \Phi[a(t)] \) by the following functions [30]:

\[
A[x] = \frac{\alpha_n x}{1 + 3 \alpha_n x^2}
\]

\[
\Phi[x] = \frac{\alpha_c x^2}{1 + \beta_c x^2}
\]

Now we have:

\[
s_n(t) = \sum_{-\infty}^{\infty} a_n \exp(j \theta_n) p(t - nT)
\]

Here the \( n^{th} \) symbol interval is given by the amplitude and phase \( a_n \) and \( \theta_n \), \( T \) is the symbol interval, and \( p(t) \) is the pulse waveform with duration \( T \). The received signal, in complex baseband representation, is composed of the signal distorted by the nonlinear channel and a complex Gaussian noise with uncorrelated real and imaginary parts.

Linear equalizers that employ training sequences are often based on adaptive Finite Impulse Response (FIR) filters. They are easy to implement and track linear distortion in the channel fairly well provided that enough taps are used (using 50–100 taps is common). Some linear equalizers, such as a zero-forcing equalizer, may amplify channel noise [42]. As an alternative, nonlinear equalizers have the potential to compensate for all the three sources of channel distortion. A common nonlinear equalizer is the Decision-Feedback Equalizer (DFE).
7.3 Non-Linear Channel Equalizers

In this section we consider important nonlinear channel equalizers, which are contemporary, before we consider the modular approach.

7.3.1 Non-Linear Equalizers Based on RBF NN

We now investigate various non-linear equalizers based on Artificial Neural Networks, that are commonly used for equalization of cellular mobile channels.

Multi-layer feed-forward neural networks and Radial Basis Function (RBF) networks have been proposed recently to utilize the non-linearity in channel equalization problem. This is because Artificial Neural Networks (ANN) can easily perform non-linear classifications and function associations.

A signal suffers from nonlinear, linear, and additive distortion when transmitted through a channel. Linear equalizers are commonly used in receivers to compensate for linear channel distortion. It is well known that nonlinear equalizers have the potential to compensate for all the three sources of channel distortion, viz. channel nonlinearity, linear distortion, and additive noise (AWGN) \[50\]. Several authors have shown that nonlinear feedforward equalizers based on either Multi-Layer Perceptron (MLP) or Radial Basis Function (RBF) neural networks can outperform linear equalizers \[50\]. A reduced complexity neural network equalizer can be made by cascading an MLP and a RBF network. In simulation, the new MLP-RBF cascade equalizer outperformed MLP equalizers and RBF equalizers \[50\].

Communication channel equalization using RBF neural networks are reported in several literatures \[81, 82\]. The economical network structure ensured by the Minimum Resource Allocation Network (MRAN) algorithm which uses on-line learning, has the capability to grow and prune the RBF network’s hidden neurons \[81\]. Compared to earlier methods, the MRAN algorithm does not have to estimate the channel order first, and fix the model parameters. The superiority of this method over existing methods is that, separate channel order estimation is not necessary. The algorithm uses an Extended Kalman Filter (EKF) to determine the weight and width of each of the nodes. This is different from previous studies, where the width values have to be set to an estimate of the noise variance of the received data. To test the algorithm for non-linear channels, the following non-linear channel \[80\] was chosen:

\[
y(t) = r(t) + 0.2x^2(t) - 0.1x^3(t) + e(t) \tag{7.6}
\]

\[
H(z) = \frac{X(z)}{S(z)} = 0.3482 + 0.8701z^{-1} + 0.3482z^{-2} \tag{7.7}
\]
The linear component $H(z)$, of the channel can be modeled as a FIR filter. The equalizer order is chosen as $n_l = 4$. In this example, the order of linear part of the channel impulse response $n_b = 2$. Thus, there will be 64 desired states for the channel output, $(2^{n_b} + n_l = 64)$. The decision delay was set to one ($\tau = 1$). The MRAN algorithm is used to train the equalizer with 500 data samples at an SNR of 20dB.

The MRAN algorithm using Radial Basis Function Neural Networks is seen to be well suited for channel equalization problems. Its ability to build up a network, based on certain parameters is seen to have an advantage over other methods, as it can be used for on-line training of the data for equalization. The algorithm's performance is evaluated by using it to build up an equalization network for two channels (linear and non-linear). The resulting networks are then tested by comparing their bit error rate performance to that of the Bayesian equalizer. The results show that the networks obtained, are comparable in performance to Bayesian equalizers when suitable training parameters are selected.

![Figure 7.1: Conventional Equalizer Scheme](image)

![Figure 7.2: Linear Equalizer with Preprocessor Filter—Proposed Scheme](image)
CHAPTER 7. A MODULAR APPROACH TO CHANNEL EQUALIZATION

Figure 7.3: Simulation Results of Preprocessor Scheme with ANFIS Prefilter, showing the time domain responses. The input signal \( x(t) \) is a Gaussian pulse train, the output of the channel is given by \( y(t) = x(t) + 0.2 x^2(t) - 0.1 x^3(t) + \eta(t) \). Note that waveforms (a) and (d) are close to each other than (a) and (c).

### 7.3.2 Non-Linear Equalizers Based on Multi-Layer Perceptrons

The idea of using Multi-Layer Perceptrons (MLP) has existed in the literature with successful examples of improved performance over linear equalizers. The MLP equalizer consists of two MLPs operating in parallel. One of them, MLP1, is trained to learn the mapping from the amplitude of the transmitted symbol, \(|S|\), to the amplitude of the received signal, \(|R|\), where \( S \) and \( R \) are phasors, obtained from the signals by integrating over one symbol duration and scaling down by the symbol duration. Assigning the input-output variables in this manner also helps the MLP to avoid modeling the noise in the received signal. The other, MLP2, is trained to learn the mapping from \(|R|\) to the phase shift introduced by the nonlinear channel, where the desired output is given by the phase difference \( \angle R - \angle S \) between the received and transmitted symbols [83].

The two MLPs are trained, both with a single hidden layer with 6 neurons and
CHAPTER 7. A MODULAR APPROACH TO CHANNEL EQUALIZATION

A linear output neuron using the entropy minimization algorithm. The training set consisted of 360 symbols. The variance of the discrete-time noise is adjusted to achieve a predetermined Signal-to-Noise Ratio (SNR) at the equalizer input. SNR here represents the ratio of average bit energy to noise Power Spectral Density (PSD). For each SNR value MLPs are trained and tested independently. In training the MLPs, steepest ascent for information potential is used. A dynamic step size, whose value increases when the update yields a better performance, and decreases when the performance degrades, is utilized. It is observed that the weights of MLPs converged to the optimal solution in about 20–30 iterations, for all SNR values, with an initial step size of 1. It is observed that these MLPs converged in 100 iterations starting with the same step size. Upon completion of the training process, the equalizers are tested for Bit Error Rate (BER) using appropriate noise levels and sufficiently long test bit sequences [83].

Some remarkable properties of the proposed equalizer are its computational simplicity, due to the small size of MLPs that can achieve good performance, efficient ex-
7.3.3 Non-Linear Equalizers Based on Fuzzy Adaptive Filters

The most commonly used recent fuzzy models are type-1 Fuzzy Adaptive Filter (FAF-I) as propounded by S.K.Patra and Bernard Mulgrew [71], and an improved version by Qilian Liang and Jerry M. Mendel discussed in [20]. A still different approach is to use the Adaptive Network based Fuzzy Inference System (ANFIS) [24]. There are some very recent innovations in Blind Channel Equalization Using Predictive RBF neural Networks [19]. J.S.R. Jang has established the functional equivalence between Fuzzy Inference Systems and RBF neural networks [84].
7.4 A Modular Approach for Non-Linear Channel Equalizer

As shown in Figure 7.1, conventional linear as well as non-linear channel equalizers are cascaded to a linear time-variant (or non-linear time-variant) channel, to combat the Inter-Symbol-Interference (ISI). In the case of a linear transmitter stage followed by a Linear Time-Variant (LTV) channel, an LTV or NLTV channel equalizer would suffice. As mentioned in section 7.2, the proposed system model incorporating a preprocessor filter that takes care of the channel non-linearities arising due to transmitter subsystem is shown in Figure 7.2. The proposed paradigm is based on a divide and conquer rule. To realize the preprocessor filter, we used an ANFIS–27 with the following parameters:

Number of Rules=49 (131 nodes); Membership Function Type: Gaussian; and Number of Epochs=20.
Figure 7.7: The signal $x(t)$ is a Gaussian pulse train, the output is given by $y(t) = x(t) + 0.2x^2(t) - 0.1x^3(t) + \eta(t)$. Note that waveforms (a) and (d) are close to each other than (a) and (c).

As a second method, we also use a RBF NN with spread 1, as the preprocessor filter. The results of the simulations are discussed in the following section.

### 7.5 Simulation Results

For the simulations, we use the input–output relation given in Equation 7.6. In the first simulation, a Gaussian Pulse train is used as the signal, $x(t)$. The standard deviation of the AWGN at the channel is taken as 0.2. The input signal, input to the preprocessor, the output of the preprocessor filter, and the output of the Linear Equalizer are given in Figure 7.3(a) to (d). The corresponding spectra of the respective signals are shown in Figure 7.4(a) to (d). In the second simulation, we use a Frequency-Hopping carrier as the signal, $x(t)$. The results are given in Figure 7.5 and 7.6. It is evident from the plots that the ANFIS based prefilter is effective in eliminating the higher order non-
linearities. Rest assured, the Linear Time-Varying Channel Equalizer may be able to estimate the input sequence more closely.

In an entirely different set of simulations, we used a Radial Basis Function Neural Network (RBF NN) to act as the preprocessor filter. Simulation results for identical input vectors are as shown in Figure 7.7(a) to (d) and 7.8. It can be seen from the Figures 7.7 and 7.8, that performance is almost comparable. Similarly, for a Frequency-Hopping carrier, the response of the RBF NN based prefilter and the output of the Linear Equalizer are given in Figure 7.9. The corresponding Spectra of signals are given in Figure 7.10.
CHAPTER 7. A MODULAR APPROACH TO CHANNEL EQUALIZATION 116

Figure 7.9: Simulation Results: RBF NN Prefilter with FH Carrier. Note that waveforms (a) and (d) are close to each other than (a) and (c).

7.6 Conclusion

We have shown that preprocessor filters based on the Adaptive Network based Fuzzy Inference System (ANFIS) and Radial Basis Function Neural Networks (RBF NN) are effective tools in combating nonlinearities introduced by the transmitter subsystem and the channel. Even though we have used ANFIS and RBF NN to realize the preprocessor filter, any suitable adaptive filter structure described in previous sections, and the generic FIR structures can also be used. Also, it is necessary to process the output of prefilter in the equalizer, using any of the methods discussed in previous sections. We can conclude that the merits for going for a preprocessor filter are:

1. The preprocessor filter can take care of the nonlinearities introduced by the channel and the transmitter. Since, transmitter power amplifier is preceding the channel, the channel nonlinearity is to be taken care of first.
2. With the preprocessor filter, the role of the equalizer is getting simplified, as the former removes the nonlinearities introduced by the channel and the transmitter Power Amplifier (PA). Thus the combination of the preprocessor filter and equalizer has the merits of simplicity in design as well as an improvement in performance.