NEAR EAST UNIVERSITY

GRADUATE SCHOOL OF APPLIED SCIENCES

A NEURO-FUZZY EQUALISER FOR CHANNEL DISTORTION

Tayseer A. M. Alshanableh

Ph.D. Thesis

Department of Electrical & Electronic Engineering

Nicosia – 2007

Tayseer A. M. Alshanableh: A Neuro-Fuzzy Equaliser for Channel Distortion

Approval of the Director of the Institute of Applied Sciences

Prof. Dr. İlkay Salihoğlu

We certify that this thesis is satisfactory for the award of the degree of Doctor of Philosophy in Electrical & Electronic Engineering

Examining Committee in Charge:

Prof. Dr. Okyay Kaynak,	**Chairman of Committee, UNESCO Chair on Mechatronics, E & E E. Dept., Boğaziçi University**
Prof. Dr. Fahreddin M. Sadıkoğlu,	**Dean of the Faculty of Engineering, NEU**
Assoc. Prof. Dr. Adnan Khashman,	**Chairman of the Electrical & Electronic Engineering Department, NEU**
Assoc. Prof. Dr. Rahib Abiyev,	**Deputy Chairman of the Computer Engineering Department, Supervisor, NEU**
Assist. Prof. Dr. Tayfun Nesimoğlu,	**Electrical & Electronic Engineering Department, Middle East Technical University-NCC**

I dedicate this work in memory of my late Mother...

ACKNOWLEDGEMENT

First of all I would like to express my special gratitude to my supervisor Assoc. Prof. Dr. Rahib Abiyev for his invaluable advice and guidance he provided to me during the course of this thesis.

I owe special thanks to Prof. Dr. Şenol Bektaş, the Vice-President of Near East University for his endless support throughout the years.

My thanks go to Prof. Dr. Fakhreddin M. Sadıkoğlu Vice-President and Dean of the Faculty of Engineering for his invaluable advice during the final stages of this work.

I would like to thank my parents who raised us twelve brothers and sisters, with great love and sacrifices.

Finally, I convey my love to my wife Filiz and my children; Muhammed, Nur, Yasemim, and Nazira for their warm support and patience throughout.

ABSTRACT

The main function of channel equalisation is to compensate distortion in a communication channel between a transmitter and a receiver. Designing an equaliser for a communication channel greatly improves the quality of signal transmission that leads to more efficient communication. In signal transmission, the presence of noise, intersymbol interference (ISI), and the time-varying characteristics of the channel requires the use of adaptive equalisers. Adaptive equalisers based on digital filtering, multilayer perceptron (MLP), radial basis functions (RBF), and fuzzy technology are widely used. However, MLP equalisers require long training time and are sensitive to the initial network parameters. The RBF equalisers are simple and require less time for training, but usually require a large number of nodes which increase the complexity of computation. In this thesis, the integration of neural networks and fuzzy technology is proposed, where a neuro-fuzzy system is considered for the equalisation of channel distortion. The construction of a fuzzy knowledge-based equaliser is a difficult problem in the design of an equaliser and time consuming. An effective way for the development of an equaliser's knowledge-base is the use of neural networks. The structure and design algorithms of the neuro-fuzzy equalisation system are presented. The use of neuro-fuzzy equaliser in digital signal transmission allows decreasing the training time of equaliser's parameters and decreasing the complexity of the network. According to the simulation results, the proposed Nonlinear Neuro-Fuzzy Network (NNFN) system provides more convergence rate and up to 10% improvement in the BER performance, in severely noisy channel conditions, compared to Adaptive Neuro-Fuzzy Inference System (ANFIS) and Feedforward Neural Network (FFNN) based systems.

CONTENTS

List of Abbreviations

ANFIS	Adaptive Neuro-Fuzzy Inference System
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate (Probability of error)
CCI	Co-Channel Interference
DCS	Digital Communication Systems
DFE	Decision Feedback Equaliser
DL	Discriminative Learning
DSP	Digital Signal Processing
EA	Evolutionary Algorithm
EKF	Extended Kalman Filter
FAF	Fuzzy Adaptive Filter
FFNN	Feedforward Neural Networks
FIR	Finite Impulse Response
FLS	Fuzzy Logic System
IIR	Infinite Impulse Response
IS	Importance Sampling
ISI	Intersymbol Interference
LMS	Least Mean Square
LSER	Least Symbol Error Rate
MAP	Maximum a-posterior Probabilities
MAPSD	MAP symbol–by–symbol Detector
MISO	Multiple-Input Single-Output
MLP	Multilayer Perceptron
MLSE	Maximum Likelihood Sequence Estimator
MLVA	Maximum Likelihood Viterbi Algorithm
MMSE	Minimum Mean Square Error
MSER	Minimum Symbol Error Rate
NFN	Neuro-Fuzzy Networks
NLMS	Normalized Least Mean Squares
NN	Neural Networks
NNFN	Nonlinear Neuro-Fuzzy Network
NSD	Neural Sequence Detector
RBF	Radial Basis Function
RLS	Recursive Least Squares
RLS	Recursive Least Squares
RNN	Recurrent Neural Networks
SCFNN	Self-Constructing Fuzzy Neural Network
SER	Symbol Error Rate
SNR	Signal-to-Noise Ratio
TE	Transversal Equaliser
TSK	Takagi–Sugeno–Kang
UKF	Unscented Kalman Filter

List of Figures

List of Tables

DECLARATION OF ORIGINALITY & CONTRIBUTION

The originality and contribution of the thesis include the followings:

- *Development of a neuro-fuzzy model for the equalisation of channel distortions,*

- *Based on gradient-descent algorithm, the mathematical model of the neuro-fuzzy equaliser was constructed,*

- *Simulation was performed by us using MATLAB files. Simulation, comparison and analysis of the results were carried out for the developed adaptive neuro-fuzzy equaliser,*

The routine used to do literature research work is an exception.

INTRODUCTION

Signals transmitted through a channel suffer from linear and nonlinear distortions. To eliminate these distortions, channel equalisation is needed. Channel equalisation is the process of compensating for the physical channel (amplitude and delay correction) between a transmitter and a receiver. It is an important area in communications as it can greatly improve the quality of transmission, which in turn leads to more efficient communication.

Whatever the physical medium used for transmission of information is, the transmitted signal is corrupted in a random manner by a variety of possible mechanisms, such as additive thermal noise generated by electronic devices, man–made noise and atmospheric noise. Interference from other users of the channel is another form of additive noise that often arises in both wireless and wire line communication systems. This interference is modelled as a random, additive white Gaussian noise (AWGN) at the output of the noise–free channel. The transmitted signal is subject to distortion due to these interferences and noise. Various equalisers have been applied to equalise these distortions and recover the original transmitted signal [1, 2].

In wireless communication channels, one of the main forms of signal degradation is the multipath propagation. Such signal distortion is characterised as a non-additive signal disturbance, which appears as time variation in the signal amplitude, usually called fading.

Signal distortions are usually characterised as random phenomena and are described in statistical terms. The effect of these signal distortions must be taken into account in the design of a communication system.

Another essential characteristic of the transmission of information through a channel is that the bandwidth allocated for the channel is often limited, resulting in a dispersion of power between neighbour symbols in the transmitted sequence. When digital signals are

transmitted through a communication channel one of the main problems that arises is due to multipath distortion is called the intersymbol interference (ISI).
The equalisation of channel distortion includes equalisation of channel noise and other interferences, such as ISI and co-channel interference. In other words, equalisation is the process of reversing the effect of multipath propagation, which has been considered as the most heavily, exploited area for adaptive filtering in digital communication systems [3].

Conventional methods for compensating channel distortion are based on introducing a linear equaliser to the output of the channel. Linear equalisers cannot reconstruct the transmitted signal when channels are nonlinear. When channel characteristics are stochastic and time-varying, adaptive equalisation based on digital filtering, multilayer perceptron (MLP), and radial basis functions (RBF) are used. MLP equalisers require long training time and are sensitive to the initial network parameters. The RBF equalisers are simple and require less training time, however, on the other hand usually require a large number of nodes which increase the complexity of computation. The performance of linear equalisers is limited due to their linear decision boundary, whereas, nonlinear equalisers provide good performance compared to linear equalisers due to their ability to form nonlinear decision boundaries. The performance of these equalisers is determined by the Bayesian equaliser, and decision feedback equaliser (DFE).

Nowadays neural networks and fuzzy technology are widely used for equalisation of channel distortions. Nonlinear adaptive filters based on neural network models have been used successfully for system identification and noise-cancellation in a wide class of applications [3]. There are number of research works, publications, which are devoted to fuzzy logic, genetic algorithms, neural computing etc. This allows the researchers to focus their investigations on artificial intelligence systems that make a shift nearer to soft computing [4].

The construction of equalisers on the basis of neural networks needs a certain time for learning parameters of the equaliser, while fuzzy technology is used to develop adaptive equalisers for nonlinear channels. In these equalisers human experts determine the fuzzy rules using input-output data pairs of the channel. These rules are used to construct the filter for nonlinear channels. The learning algorithms are applied to change parameters of the membership functions of the rules and develop equalisers. The use of such approach improves the adaptation speed.

In this thesis, neural networks and fuzzy technology are used for the development of a neuro-fuzzy equaliser for channel distortion.

The thesis consists of an introduction, four chapters, a conclusion, references and an appendix.

In chapter one, different methods of channel equalisation are reviewed. The state of application problem of neural and fuzzy technologies for channel equalisation is presented. The statement of research problem is given.

In the second chapter the structure of data transmission system, the functions of its main blocks are explained. The source of channel noise and interferences are given. The structure of adaptive neuro-fuzzy equalisation system for channel distortion is presented.

In chapter three the mathematical background of the construction of a neuro-fuzzy equaliser for channel distortion is presented. The structure of the neuro-fuzzy equaliser and its learning algorithm are described.

In chapter four the development of neuro-fuzzy equaliser for channel distortion is carried out. The simulation results of the neuro-fuzzy equaliser and the results of different types of adaptive equalisers are compared.

In the conclusion, the advantages of using neuro-fuzzy equalisation system are discussed. The results show that the use of the neuro-fuzzy equaliser ensures improved learning and BER performance conditions.

CHAPTER I
REVIEW ON CHANNEL EQUALISATION

1.1 Overview

Equalisation of channel distortions provides an accurate transmission of the input transmitted signals to a receiver. This is acquired by using efficient equalisation algorithms in signal transmission. In this chapter, understanding of the used methodologies in channel equalisation is considered. The application of different equalisation algorithms in digital signal transmission is analysed. The usage of neural networks, fuzzy and neuro-fuzzy technologies in adaptive channel equalisation is discussed.

1.2 The State of Application of Channel Equalisation

Channel equalisation includes the equalisation of linear and nonlinear distortions. These are ISI, co-channel interference, and noise. On one hand, linear equalisers are commonly used in receivers to compensate for linear channel distortion. On the other hand, nonlinear equalisers have the potential to compensate for both linear and nonlinear channel distortions. Different types of equalisers are applied for equalisation of channel distortions in order to recover transmitted signals at the receiver.

Equalisation can be divided into two types: sequence estimation, and symbol detection [2, 5]. The first one needs channel estimation, and it is computationally complex. In this thesis adaptive channel equalisation that realises symbol detection technique is considered. This is a classification problem in which the input baseband signal is mapped onto a feature space determined by the direct interpretation of a known training sequence. Here the aim is the separation of symbols in the output signal space whose optimal decision region boundaries are nonlinear. Recently, a nearest neighbour rule [6] is used to classify the distorted signal. In [7] a systematic feature space partitioning method is proposed to divide the entire feature space into two decision regions using a set of hyper-planes.

In general, all types of Digital Communication Systems (DCS's) are affected by ISI. For example, digital transmission over analogue telephone lines experiences ISI due to the limited bandwidth of the medium. Mobile radio channels are also affected by ISI resulting from multipath fading due to the relative motion between the transmitter and the receiver [8].

The ISI may cause errors when attempting to recover the data sequence. To make things worse, the channel characteristics that cause the distortion may vary considerably in time. Therefore, it is appropriate to assume that the channel, which is modelled as a linear system, is not known during the design of the receiver. In such a case the problem is to design a corrective system which, when cascaded with the front end of the receiver produces an output that, in some sense, corrects for the distortion caused by the channel and thus yields a replica of the transmitted signal. Since the distorting system is usually unknown, it is necessary for the corrective system to identify and continuously adapt to the, often, time–varying channel. Such a system is called an *adaptive equaliser*. The equalisation problem has received great attention in the literature and different solutions to this problem may be found [2].

In general the family of adaptive equalisers can be divided into *supervised equalisers* and *unsupervised equalisers*. For the identification of the unknown channel, it is often necessary, when possible to periodically excite the system with a known *training* or *pilot* signal interrupting the transmission of useful information. A replica of this pilot signal is available at the receiver and the receiver compares the response of the system with its input in order to update its parameters in some manner. Such equalisers are known as supervised equalisers. However, the constraints associated with some communication systems, such as digital television or digital radios do not provide the scope for the use of a training signal. In this situation the equaliser needs some form of unsupervised or *self recovery* method to update its parameters. These equalisers are called *blind equalisers*. After training, the equaliser is switched to *decision directed* mode, where the equaliser can update its parameters based on the actual detected data.

The process of supervised equalisation can be achieved broadly in two ways. These are *sequence estimation* and *symbol-by-symbol estimation or symbol detection*. The sequence estimator uses the sequence of received samples to recover the entire transmitted sequence of data symbols. The optimum sequence estimator is the maximum likelihood sequence estimator (MLSE) [9] and can be efficiently implemented based on a Viterbi trellis–the maximum likelihood Viterbi algorithm (MLVA) [10]. It is well known that the MLVA algorithm provides the best attainable equalisation performance. Since the MLSE requires that the entire data sequence has been received before the detection has been made, its theoretical performance can not be achieved in real–time systems where an arbitrary big decision lag cannot be tolerated.

The class of symbol–by–symbol equalisers, on the other hand, detect each transmitted symbol separately. In most cases, the decision of a symbol–by–symbol equaliser can be regarded as a function of a vector containing past received samples. This *decision function* is often restricted to be linear and the resulting equaliser is referred to as a *linear equaliser*. If there are no restrictions for the decision function, the equaliser is called a *nonlinear equaliser*. The optimum decision function is in general nonlinear and is given by the maximum a-posterior probabilities (MAP) criterion derived by Bayes's theory [11]. Hence, the optimum MAP symbol–by–symbol detector (MAPSD) is also called the *Bayesian equaliser* [12]. It has been shown in [13, 14] that the MAPSD provides a lower bit–error rate (BER) for a given lag than the MLSE. At high signal to noise ratios (SNR's), their performance is virtually indistinguishable. On the other hand, at low SNR the MLSE is inferior to the MAPSD.

Recent advances in signal processing techniques have provided a wide variety of nonlinear equalisers. These include Volterra series based equalisers [15], Mahalanobis distance equalisers [16], artificial neural networks, multilayer perceptrons (MLP), radial basis functions (RBF) network, fuzzy filters and fuzzy basis functions [17, 18, 19, 20]. The nonlinear equalisers, in general, treat equalisation as a pattern classification problem.

Another type of adaptive equalisers is based on linear system theory, such as decision feedback that improves the performance of the equaliser. The design of decision feedback equalisers (DFEs) that is based on the minimum mean square error (MMSE) principle is given in [3], where it uses least mean square algorithm for simple and effective adaptive implementation. It is well-known, however, that in certain situations the MMSE solution can be distinctly inferior to the optimal minimum symbol error rate (MSER) solution. In [3] the MSER design for multilevel pulse-amplitude modulation is considered. Block-data adaptive implementation of the theoretical MSER DFE solution is developed based on the classical Parzen window estimate of probability density function. Furthermore, a sample by sample adaptive MSER algorithm, called the least symbol error rate (LSER), is derived for adaptive equalisation application. The proposed LSER algorithm has a complexity that increases linearly with the equaliser length. Computer simulation is employed to evaluate the proposed alternative MSER design for equalisation application with multi-level signalling schemes.

A linear approach for the decision function of the symbol–by–symbol equaliser provides a computationally less complex *linear equaliser*, but at the expense of inferior performance. In order to design such linear equalisers, different optimisation criteria may be employed, such as minimum mean squared error (MMSE) or minimum amplitude distortion. The optimum, in the MMSE sense, linear equaliser is given by the Wiener equations [21], which require exact knowledge of the channel characteristics. In practice, however, the linear equaliser is a linear filter [22] trained with an adaptive algorithm like the least mean squares (LMS) or recursive least squares (RLS). These linear equalisers treat equalisation as inverse filtering and during the process of training they optimise a certain optimisation criterion such as MMSE.

A special category of equalisers is the class of decision feedback equalisers (DFE's). The DFE uses its past decisions in order to remove part of the distorting intersymbol interference from the received signal. The transfer function of a DFE is, in general, a non–linear function of the received signal, whatever its structure, due to the feedback operation. However, the operation of the DFE can be viewed as a function computed on the samples from the received signal and past detected symbols [18]. According to the nature of this function, the DFE may be classified as either linear or non–linear. In this thesis the term nonlinear equalisers is used exclusively for those equalisers that provide a nonlinear decision function based on received samples or the received samples along with previously detected samples.

Different approaches have been proposed for channel equalisation. Within the communications signal processing area adaptive filters are in common use as equalisers and echo cancellers tend to be of a simple finite impulse response (FIR) variety. Although infinite impulse response (IIR) filter types would be attractive from the view point of complexity, they are not generally used due to problems with speed of adaptation and stability [23]. These processors are generally adapted using least squares objective functions implemented by recursive least squares (RLS) or stochastic gradient algorithms such as Normalized Least Mean Squares (NLMS). Non-linear adaptive processors have been deployed in the application areas of both echo cancellation and equalisation. In the case of equalisation this has been done because it is often possible to deploy non-linear structures which use fewer observation samples than linear equalisers, thus introducing less noise [22].

Much work on non-linear equalisers has concentrated on linear in the parameter (LITP) models because they are easy to adapt using conventional algorithms. However, when substantial intersymbol interference is present then the complexity of these processors becomes excessive.

One of the conventional methods for compensation of channel distortion is based on introducing the linear equaliser (linear inverse filter to the channel frequency response) to the output of the channel. This design methodology is appropriate when the channel model is precisely known and characteristics of the channel are not time-varying. When a channel has time-varying characteristics adaptive equalisers are used. Classical approaches for adaptive equalisers design are based on the knowledge of the parametric channel model [24]. These are implemented by identifying the channel dynamic and then constructing an equaliser using the identified channel model. These processes require certain time to gathering statistical data about the channel and time consuming. One type of equalisers is based on increasing the number of equaliser taps and choosing the coefficients from different ranges of values according to the amplitude of distorted signals [25]. In this approach a large number of coefficients and switching thresholds are required.

The basic problem in channel equalisation is decreasing the bit error rate (BER) (or the probability of error) of the equaliser, which determines its performance. Channel distortions are mostly nonlinear; in this case, we need to use nonlinear channel equalisers in order to attain a lower BER, lower the mean squared error (MSE), and higher convergence rate than that of linear equalisers.

The performance of linear equalisers is limited due to their linear decision boundary, while nonlinear equalisers provide good performance compared to linear equalisers due to their ability to form nonlinear decision boundaries, where the performance of these equalisers is determined by the Bayesian equaliser. Decision feedback equaliser (DFE) adaptive filtering is now an integral part of most of the modern communication systems where it is involved with both channel equalisation and estimation techniques. In these systems, filters are generally fed with a short training sequence to which they have to adapt prior to receiving data. This training sequence is often multiplexed with the data, reducing the amount of data transmitted in each frame.

To maximise the efficiency of a system, training sequences need to be as short as possible requiring that adaptation occurs in as few iterations as possible. Also as the data rates of communication systems increase, the time available to complete a single iteration decreases. All of these factors place increasing demands on implemented algorithms, requiring fast digital signal processors with highly efficient optimised software.

In [26], an equaliser algorithm is presented which is suitable for the use with a differential detector operating in a time dispersive channel. The algorithm, derived from previous Bayesian coherent methods, is able to provide reliable performance even after differential detection. Results for the differential equaliser operating in a typical indoor wireless channel are presented and are shown to compare favourably with those of a coherent receiver, using decision feedback equalisation, in the presence of a frequency offset.

An importance sampling (IS) simulation method is presented for evaluating the lower-bound symbol error rate (SER) of the Bayesian DFE with M-PAM symbols, under the assumption of correct decision feedback [27]. By exploiting an asymptotic property of the Bayesian DFE, a design procedure is developed, which chooses appropriate bias vectors for the simulation density to ensure asymptotic efficiency (AE) of the IS simulation

In [27], the optimisation techniques for real-time adaptive algorithms based on Wiener and Kalman filter theory were developed. Two algorithms in particular were implemented on the TMS320C6201 evaluation module, these being the Least Mean Squares (LMS) and Recursive Least Squares (RLS). Benchmarking of the algorithms was performed allowing the evaluation of the maximum bit rate that can be supported in various situations. The two algorithms were also compared in other areas such as code size, ease of implementation, stability, reliability and data memory required for implementation.

An adaptive beam-forming technique is proposed based on directly minimizing the bit-error rate (BER) in [28, 29]. It is demonstrated that this minimum BER (MBER) approach utilises the antenna array elements more intelligently than the standard minimum mean square error (MMSE) approach. Consequently, MBER beam-forming is capable of providing significant performance gains in terms of a reduced BER over MMSE beam-forming.

Furthermore, a symbol-by-symbol adaptive implementation is considered, and a stochastic gradient algorithm, referred to as the least bit error rate, is derived. The proposed adaptive MBER beam-forming technique provides an extension to the existing work for adaptive MBER equalisation and multi-user detection.

An important application of signal processing is that of equalisation, which functions to compensate for the distortion undergone by a signal in its path between a transmitter and a receiver. In the past years there have been important advances in the field of equalisation that have brought, for instance, the wide development of mobile telephony. However, many equalisation systems are relatively basic. By improving the equalisation techniques mobile telephony operators could gain an increased capacity (number of telephones per cell) and call quality. In [30] the developing of algorithms for a class of nonlinear communication channels is considered. This is a difficult problem, since the field of nonlinear signal processing is relatively new. Our focus involves the use of genetic programming and associated optimisation techniques, an area in which shall be concentrated in the future, trying to improve the speed of these methods so that they can operate in real time.

Other equaliser type is based on increasing the number of the equaliser taps and choosing the coefficients from different ranges of values according to the amplitude of distorted signals. In this approach a large number of coefficients and switching thresholds are required.

Most of the described equalisers are based on linear system theory and they are efficiently used for equalisation of linear channels. The application of these equalisers to nonlinear channel does not provide the required BER characteristics; nowadays, neural networks and fuzzy technology are widely used for equalisation of nonlinear channel distortions.

1.3 State of Application of Neural Networks and Fuzzy Technologies for Channel Equalisation

1.3.1 Design of Neural Network Based Equalisers

Nonlinear equalisers have the potential to compensate for all nonlinear, linear, and additive channel distortion. Nonlinear adaptive filters based on neural network models have been used successfully for system identification and noise-cancellation in a wide class of applications. Different neural network structures such as Multilayer Perception (MLP), Radial-Basis Function Networks (RBF), and Recurrent Neural Networks (RNN) have been implemented to achieve these ideas. The construction of equalisers on the basis of neural network needs some time interval for learning the parameters of the equaliser.

Filtering is composed of two distinct estimation (computation) procedures. One is the estimation of the mapping (transformation) from the available samples, the other is the estimation of the output of the filter from the input by the realisation of this mapping. For a linear filter, it is not difficult to realise the mapping once the mapping is available.

For a nonlinear filter, the realisation of the mapping is not as easy as that for the linear filters. How to estimate and to realise effectively the mapping of nonlinear filters is the current research focus in this field [31].

The nonlinear mapping capability and the corresponding learning algorithm of the MLP network provide us with a new approach to attack the above problems of nonlinear filters. A general nonlinear filter is shown in Figure 1.1.

MLP networks comprise a large class of feedforward neural networks with one or more layers of neurons, called hidden neurons, between the input and output neurons. The key function of MLP networks is the implementation of a nonlinear input-output mapping of a general nature [31].

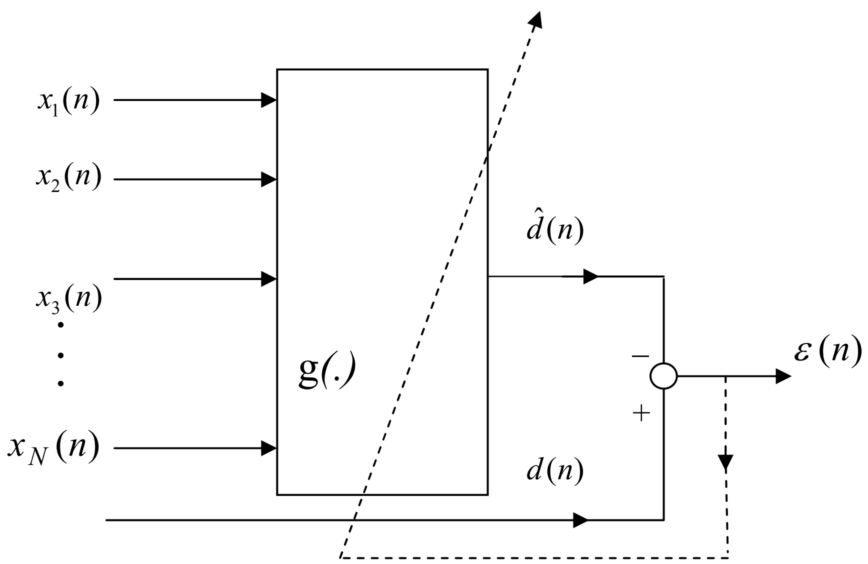


Figure 1.1 A general nonlinear filter. [31]

Here $x_i(n)$ are the input signals, $\hat{d}(n)$ is the filter output, $d(n)$ is the desired signal, and $\varepsilon(n)$ is the error.

The minimum error entropy criterion was suggested in adaptive system training as an alternative to the mean-square-error (MSE) criterion, and it proved to produce better results in many tasks. A MLP scheme trained with this information theoretic criterion is applied to the problem of nonlinear channel equalisation [30]. In [30] a realistic nonlinear channel model, which is encountered when practical power amplifiers are used in the transmitter, where the bandwidth efficient 16-QAM scheme, which uses a dispersed constellation, is assumed. The nonlinearity of the MLP is dependent upon the discontinuity of the perceptron activation functions [32]. More nonlinearity exists for more discontinuous activation [33]. Ideally, a threshold function would be used to optimise the MLP structure. The Backpropagation algorithm, however, would not operate on such a structure so a sigmoid activation function with a small gradient is used. This activation function limits the nonlinearity and an optimal performance is not achieved.

An alternative network to the MLP for many applications of signal processing is the RBF network. Since MLP networks are sometimes plagued by long training time and may be trapped at bad local minima, RBF networks often provide a faster and more robust solution to the equalisation problem [3].

An RBF is a multidimensional function that depends on the distance between the input vector and a centre vector. RBF's provide a powerful tool for multidimensional approximation or fitting that essentially does not suffer from the problem of proliferation of the adjustable parameters as the dimensionality of the problem increases [34].

RBF network includes basis function that is viewed as the activation function in the hidden layer [35]. The most common basis function chosen is the Gaussian function. The RBF network and its complex equivalent (CRBF) have been found to be attractive.

In [36], comparison of the performances of MLP vs. RBF equalisers in terms of symbol error rate vs. SNR is given. It was shown that the combination of MLP-RBF equaliser outperforms MLP equalisers and RBF equalisers.

Most of the commonly used blind equalisation algorithms are based on the minimisation of a non-convex and nonlinear cost function and a neural network gives smaller residual error as compared to a linear structure. The efficiency of complex valued feed-forward neural networks for blind equalisation of linear and nonlinear communication channels has been confirmed by many studies. Two neural network models for blind equalisation of time-varying channels, for M-ary QAM and PSK signals are presented in [37]. The complex valued activation functions, suitable for these signal constellations in time-varying environment, are introduced and the learning algorithms based on the CMA cost function are derived. The improved performance of the proposed models is confirmed through computer simulation.

Constructing adaptive minimum bit error rate (MBER) neural network equalisers for binary signalling are considered in [31]. Motivated from a keruel density estimation of the BER, as a smooth function of training data, a stochastic gradient algorithm called the least bit error rate (LBER) is developed for adaptive nonlinear equalisers. This LBER algorithm is applied to adaptive training of a radial basis function (RBF) equaliser in a channel intersymbol interference (ISI) plus co-channel interference setting. A simulation study shows that the proposed algorithm has good convergence speed, and a small-size RBF equaliser trained by the LBER can closely approximate the performance of the optimal Bayesian equaliser. The results also demonstrate that the standard adaptive algorithm, the LMS, performs poorly for neural network equalisers because the MMSE is clearly suboptimal in the equalisation setting. The results also demonstrate that the standard adaptive algorithm, the LMS, performs poorly for neural network equalisers because the MMSE is clearly suboptimal in the equalisation setting [5].

The effectiveness of using an Evolutionary Algorithm (EA) for the equalisation of a non-minimum phase channel using a feedforward multilayer perceptron is given in [38]. The initialisation of the MLP decision regions, using a predefined shape that suits the equalisation problem, has been shown to considerably speed up the convergence of the algorithm, as well as improve the performance by increasing the likely hood of an "optimal" convergence result.

Conventional techniques utilising first and second order approximations of the error surface have been demonstrated to be ineffective in achieving an optimal solution in continuous simulations and have proved incapable of dealing with the more difficult non-minimum phase problems. Using an EA, a consistent near optimal solution is achieved.

In [39] DFEs based on two weighted neural networks are presented. It is shown that the choice of an innovative cost functional based on the Discriminative Learning (DL) technique, coupled with a fast training paradigm, can provide neural equalisers that outperform standard DFEs at a practical signal to the noise ratio (SNR). In particular, the novel Neural Sequence Detector (NSD) is introduced, which allows extending of the concepts of Viterbi-like sequence estimation to neural architectures. Resulted architectures are competitive with the Viterbi solution from cost-performance aspects, as demonstrated in experimental tests.

Recurrent neural networks (RNN) have feedback, small size (number of neurons), and high Bit Error Rate (BER) performance that make them attractive for high-speed adaptive equalisation of nonlinear channels with deep spectral nulls. RNN, in which each unit is connected to all other units, are the most general case of neural networks. RNN are highly non-linear dynamical systems that exhibit a rich and complex dynamical behaviour.

It is important to note that RNNs with the same structures can exhibit different dynamic behaviour as a result of using distinct training algorithms. Consequently, an RRN network is defined only when both its architecture and training algorithm are given. Several algorithms exist for the training of RNNs, the most widely known algorithm is Real Time Recurrent Learning (RTRL) algorithm. The RTRL algorithm is based on the minimisation of the MSE by a gradient descent procedure and is used to update the weights of the RNN during the training period. The small size of RNN equalisers makes them attractive for high speed channel equalisation when compared with the complexity associated with other neural equaliser structures [40, 79].

The structure of RNN based equaliser is given [79]. The RNN structure and its training algorithm are used to design equalisers for the equalisation of noise. The inputs of neural equaliser are the channel output signals. The output of the neural network is the recovered transmitted sequence of signals [1].

In [41] adaptive RNN based equaliser whose small size and high performance makes it suitable for high-speed channel equalisation is considered. The RNN based structure is proposed for both trained adaptation and blind equalisation. The performance of equaliser is evaluated via extensive simulations for variety of signal modulations and communication channel models. It is shown that the RNN equalisers have comparable performance with traditional linear filter based equalisers when the channel interferences are relatively mild, and that they outperform them by several orders of magnitude when either the channel's transfer function has spectral nulls or severe nonlinear distortion. In addition, the small size RNN equalisers, being generalized IIR filters and outperform multilayer perceptron equalisers of larger computational complexity in linear and non-linear channel equalisation cases.

In some communication systems the transmitted signal is contaminated by impulsive noise with a non-Gaussian distribution. Non-Gaussian noise causes significant performance degradation to communication receivers. In [41] a recurrent neural equaliser is applied to impulsive noise channels, for which the performance of neural network equalisers has never been evaluated. This application is motivated from the fact that the unscented Kalman filter (UKF), which is suited for training of the recurrent neural equaliser, provides a higher accuracy than the extended Kalman filter (EKF) in capturing the statistical characteristics for non-Gaussian random variables. The performance of the recurrent neural equaliser is evaluated for impulsive noise channels through Monte Carlo simulations. The results support the superiority of the UKF to the EKF in compensating the effect of non-Gaussian impulsive noise.

An adaptive decision feedback recurrent neural equaliser (DFRNE), which models a kind of an IIR structure, is proposed in [42]. Its performance is compared with the traditional linear and nonlinear equalisers with FIR structures for various communication channels. The small size and high performance of the DFRNE makes it suitable for high-speed channel equalisation.

An important problem in high density digital magnetic recording system is the removal of distortions introduced by linear or nonlinear message corrupting mechanisms in the reconstruction of the original symbols. Severe nonlinear distortions in high density digital magnetic recording systems can make it difficult for conventional equalisers to reconstruct the originally recorded symbols. In [43] a Decision Feedback Recurrent Neural Equaliser (DFRNE) with a simple structure, which can recover the original symbols correctly under severe nonlinear distortion, is described. By evaluating its performance through computer simulations for various channels, the DFRNE has comparable performance with traditional equalisers when the channel interferences are mild. And it outperforms them when the channel's transfer function has spectral nulls or when severe nonlinear distortion is present. In addition, the DFRNE, being essentially an IIR filter, is shown to outperform multilayer perceptron equalisers in linear and non-linear channel equalisation cases.

Recurrent neural networks (RNNs) have been successfully applied to communications channel equalisation because of their modelling capability for nonlinear dynamic systems [79]. Major problems of gradient-descent learning techniques commonly employed to train RNNs are slow convergence rates and long training sequences required for satisfactory performance. Decision-feedback equaliser using an RNN trained with Kalman filtering algorithms is presented in [44]. The main features of the proposed recurrent neural equalisers, using the extended Kalman filter (EKF) and unscented Kalman filter (UKF), are fast convergence and good performance using relatively short training symbols. Experimental results for various time-varying channels are presented to evaluate the performance of the proposed approaches over a conventional recurrent neural equaliser [44].

1.3.2. Channel Equalisation by Using Fuzzy Logic

One of the effective ways for the development of adaptive equalisers for nonlinear channels is the use of fuzzy technology in their development. These equalisers are nonlinear filters that are used for equalisation of variety of communication systems. In these equalisers, the fuzzy rules using input-output data pairs of the channel are determined. This type of adaptive equalisers can process numerical data and linguistic information in natural form. Fuzzy equaliser that includes fuzzy IF-THEN rules was proposed for nonlinear channel equalisation [45]. Human experts determine the fuzzy rules using input-output data pairs of the channel. These rules are used to construct the filter for nonlinear channel. The recursive least squares and least mean squares algorithms are applied to change parameters of the membership functions of rules and develop equalisers. The incorporation of linguistic and numerical information improves the adaptation speed and its bit error rate (BER).

In [45] it was indicated that a linear transversal filter requires a much larger training set to achieve the same error rate as it was achieved by fuzzy logic equaliser. The fuzzy logic equalisers are also proposed for quadratic amplitude modulation (QAM) constellation channel equalisation [46], and for implementation a Bayesian equaliser to eliminate co-channel interference [47, 48].

In [49], a new method to solve the channel equalisation problem using fuzzy logic is given. The membership functions are derived from the training data set and do not have to be pre-defined. A method for combining the outcomes of different rules is also proposed. The performance of the new method is compared with the transversal filter based equaliser. It is shown, using simulation that the fuzzy equaliser performs better in the presence of channel non-linearity.

The problem of channel equalisation in digital cellular radio (DCR) application is given in [39]. DCR systems are affected by co-channel interference (CCI), intersymbol interference (ISI) in presence of additive white Gaussian noise (AWGN). Here a fuzzy equaliser is proposed to equalise communication channels with such abnormalities. This equaliser performs close to the optimum Bayesian equaliser with a substantial reduction in computational complexity. The equaliser is trained with supervised and unsupervised scalar clustering techniques in sequence, and consists of a fuzzy equaliser with a pre-processor for CCI compensation. Simulation studies have demonstrated the performance of the proposed technique.

A fuzzy adaptive filter is constructed from a set of fuzzy IF-THEN rules which change adaptively to minimize some criterion function as new information becomes available [44]. A fuzzy adaptive filter uses a recursive least squares (RLS) adaptation algorithm.

The RLS fuzzy adaptive filter is constructed through the following four steps:

1) Define fuzzy sets in the filter input space $U \in R^n$ whose membership functions cover U; 2) Construct a set of fuzzy IF-THEN rules which either come from human experts or are determined during the adaptation procedure by matching input-output, data pairs; 3) Construct a filter based on the set of rules; and, 4) Update the free parameters of the filter using the RLS algorithm.

The most important advantage of the fuzzy adaptive filter is that linguistic information (in the form of fuzzy IF-THEN rules) and numerical information (in the form of input-output pairs**)** can be combined into the filter in a uniform fashion. Finally, this fuzzy adaptive filter is applied to nonlinear communication channel equalisation problems; the simulation results show that:

1) Without using any linguistic information, the RLS fuzzy adaptive filter is a well-performing nonlinear adaptive filter (similar to polynomial and neural-net adaptive filters); 2) By incorporating some linguistic description (in fuzzy terms) about the channel into the fuzzy adaptive filter, the adaptation speed is greatly improved; and, 3) The bit error rate of the fuzzy equaliser is very close to that **of** the optimal equaliser.

A new kind of adaptive filter called type-2 fuzzy adaptive filter (FAF) is proposed in [38]. This adaptive filter is realized by using an un-normalised type-2 Takagi–Sugeno–Kang (TSK) fuzzy logic system (FLS). The filter is applied to the equalisation of a nonlinear time-varying channel and it was demonstrated that it can implement the Bayesian equaliser for such a channel. The developed equaliser has a simple structure, and provides fast inference. A clustering method is used to adaptively design the parameters of the FAF. Two structures are used for the equaliser: transversal equaliser (TE) and decision feedback equaliser (DFE). A new decision tree structure is used to implement the decision feedback equaliser, in which each leaf of the tree is a type-2 FAF. This DFE vastly reduces computational complexity as compared to a TE. Simulation results show that equalisers based on type-2 FAFs perform much better than nearest neighbour classifiers (NNC) or equalisers based on type-1 FAFs [50].

In [50] type-2 fuzzy adaptive filter is used for overcoming time-varying co-channel interference (CCI). A clustering method is used to adaptively design the parameters of the FAF. The transversal equaliser and decision feedback equaliser structures are used to eliminate the CCI. Simulation results show that the equalisers based on type-2 FAFs perform better than the nearest neighbour classifiers or the equalisers based on type-1 FAFs when the number of co-channels is much larger than 1.

In [49] the channel equalisation using fuzzy logic is presented. Here membership functions are estimated from the training set and a method to estimate the delay of the communication channel is presented. The performance of the fuzzy equaliser is compared with the transversal filter equaliser. It is shown using simulations that the transversal filter requires a much larger training set to achieve the same error rate. Simulations results demonstrate that the performance of the fuzzy equaliser is better in the presence of channel nonlinearities.

Recently, fuzzy technology is used for the development of adaptive equalisers for nonlinear communication channels. They are nonlinear filters that are used for equalisation of variety of communication systems. In these equalisers, the fuzzy rules using input-output data pairs of the channel are determined. These rules are used to construct the filter for nonlinear channel. The recursive least squares (RLS) and the least mean squares (LMS) algorithms are applied to change parameters of the membership functions of rules and to develop equalisers [50]. The use of such approach improves the adaptation speed. In some cases the construction of fuzzy rules for equalisers is very difficult, and then one of the effective technologies for construction of equaliser's knowledge base is the use of neural networks. In this thesis integration of neural network and fuzzy technology is considered for equalisation of channel distortion. Neuro-fuzzy systems belong to a newly developed class of hybrid intelligent systems, which combine the main features of artificial neural networks with those of fuzzy logic [51]. Neither fuzzy reasoning systems nor neural networks are by themselves capable of solving problems involving at the same time both linguistic and numerical knowledge.

The design of a self-constructing fuzzy neural network (SCFNN)-based digital channel equaliser is proposed in this thesis. It is demonstrated that the SCFNN-based digital channel equaliser possesses the ability to recover the channel distortion effectively. The performance of SCFNN is compared with that of the adaptive-based-network fuzzy inference system (ANFIS) and the optimal Bayesian solution. Simulations were carried out in both real-valued and complex-valued nonlinear channels to demonstrate the flexibility of the proposed equaliser. The experimental results show that the performance of SCFNN can be close to that of the Bayesian optimal solution and ANFIS, while the hardware requirement of the trained SCFNN-based equaliser is much lower [52].

In some cases the construction of proper fuzzy rules for equalisers is difficult. In this case one of the effective technologies for construction of equaliser's knowledge base is the use of neural network. In this thesis, the adaptive channel equalisation by using recurrent neuro-fuzzy network is considered. The use of neuro-fuzzy technology allows using small number of parameters, fast and easy train equaliser. The equaliser based on neural networks doesn't need appropriate knowledge about channel dynamics. These equalisers give better results in bit error rate (BER), at the cost of computational strength.

1.4 The State of Research Problem

The presence of noise and the time-varying nature of channel need the usage of soft-computing elements, such as neuro-fuzzy technology, for the construction of an equaliser. In this thesis, adaptive equalisation, based on symbol detection on the output of the channel is considered. The equalisation considered is a geometric classification problem. The main objective is the separation of the received symbols in the output signal space. In this equalisation input base-band sequence of signals are mapped onto a feature space determined by the direct interpretation of a known training sequence, i.e. neuro-fuzzy rule is used to classify the distorted signal. The methodology used in equaliser's development is based on neural network and fuzzy theory.

Development of neuro-fuzzy system for equalisation of channel distortion includes the following steps:

- First, the analysis of the methodologies used for the equalisation of channel distortions and state of application problems of neural and fuzzy technologies for the development of an equaliser is considered.

- Second, the structure of data transmission and the operation structure of adaptive channel equalisation using neuro-fuzzy network is presented.

- Third, the mathematical model of the neuro-fuzzy network for the development of equalisation system for channel distortion is presented. The learning algorithm of neuro-fuzzy system is considered.

- Fourth, the development of the neuro-fuzzy equaliser for channel distortion is presented. The simulation results of the equaliser and a table of comparison of different equalisation techniques are presented.

1.5 Summary

Analysis of technologies used for equalisation of channel distortions demonstrates that one of the effective methodologies for the improvement of the efficiency of data transmission is the combination of neural networks and fuzzy logic. Neural network has self-learning characteristic that increases the accuracy of the data transmission, Whereas, Fuzzy logic allows to reduce the complexity of the data and to deal with uncertainty. In this chapter the state of art understanding of the used methodologies in channel equalisation is considered. The application of different equalisation algorithms in digital signal transmission is analysed. It is proposed to use combination neural network and fuzzy technologies in adaptive channel equalisation. The distortion of the received signal due to intersymbol interference was also highlighted and the need for adaptive equalisation was underlined. Finally, we discussed the different techniques employed for the problem of equalisation, with our attention based on nonlinear equalisers.

CHAPTER 2
STRUCTURE OF CHANNEL EQUALISATION

2.1. Overview

A communication system consists basically of three parts: transmitter, channel and receiver (Figure 2.1). A transmitter converts the electrical signal into a form that is suitable for transmission through the physical channel or transmission medium. The function of the receiver is to recover the message signal contained in the received signal. The communications channel is the physical medium that is used to send the signal from the transmitter to the receiver. A transmitted signal is distorted in the channel before it reaches the receiver.

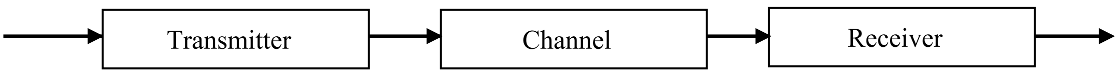


Figure 2.1 Basic Components of a Transmission System.

Digital communication differs from its analogue counterpart in that it can only transmit a finite number of waveforms. The information, transmitted as a stream of binary digits, is typically coded prior to transmission, i.e., redundant bits are added to the message to provide protection against transmission errors. In the same way, the information that leaves the receiver must be decoded before it can be used. A transmitted symbol is distorted by other transmitted symbols and also by noise that is defined as an unwanted signal. Noise itself is a signal that conveys information regarding the source of the noise. At the receiver the equalisation of channel is performed to neutralise the effect of distortion on the received signals. In this chapter the structure of data transmission system and the functions of its main components and equalisation of channel distortion will be discussed.

2.2. Architecture of Data Transmission Systems

A transmission channel is defined as the electrical medium between the source and the destination, the channel is characterised by its Loss/Attenuation, Bandwidth, Noise/Interference and Distortion. The receiver function is principally to reverse the modulation process of the transmitter in order to recover the message signal, attempting to compensate for any signal degradation introduced by the channel. This will normally involve amplification, filtering, demodulation and decoding, and in general is a more complex task than the transmitting process.

Advancement in digital signal processing (DSP) technology has made digital modulation more cost effective than analogue transmission systems. Digital modulation systems have more advantages comparing with the analogue ones; they provide more noise immunity, robustness to channel impairments, easier multiplexing of various forms of information, and greater security. In digital wireless communication systems, the modulating signal may be represented as a time sequence of symbols or pulses, where each symbol has m finite states. Each symbol represents n bits of information, where $n = \log_2 m$ bits/symbol [51].

Digital communication systems (DCSs) are designed to transmit the information generated by a source to one or more destinations in digital form. The architecture of a general DCS is presented in Figure 2.2. The data source constitutes the signal generation system that generates the information to be transmitted. Information sources may take a variety of different forms. They can be analogue, such as audio sources in radio broadcasting or video sources in TV broadcasting. In contrast, they can be digital such as binary data or ASCII characters generated by computers and storage devices (e.g. magnetic or optical disks). In modern digital communication systems all the information to be transmitted must be first converted into a sequence of digits. For non–digital sources, this is done through sampling and quantisation. Therefore, the information source can always be regarded as producing a stream of digits.

The stream of digital data, or information sequence, is then passed to the encoder. The purpose of the encoder is to introduce, in a controlled manner, some redundancy in the digital information sequence that can be used at the receiver to overcome the effects of noise and interference encountered during the transmission of the signal through the channel. This added redundancy actually serves to provide means for error detection and/or correction at the receiver end. Some of the typical coding schemes used are Gray codes, block codes (e.g., Hamming code and cyclic codes), convolution codes and turbo codes [1, 52].

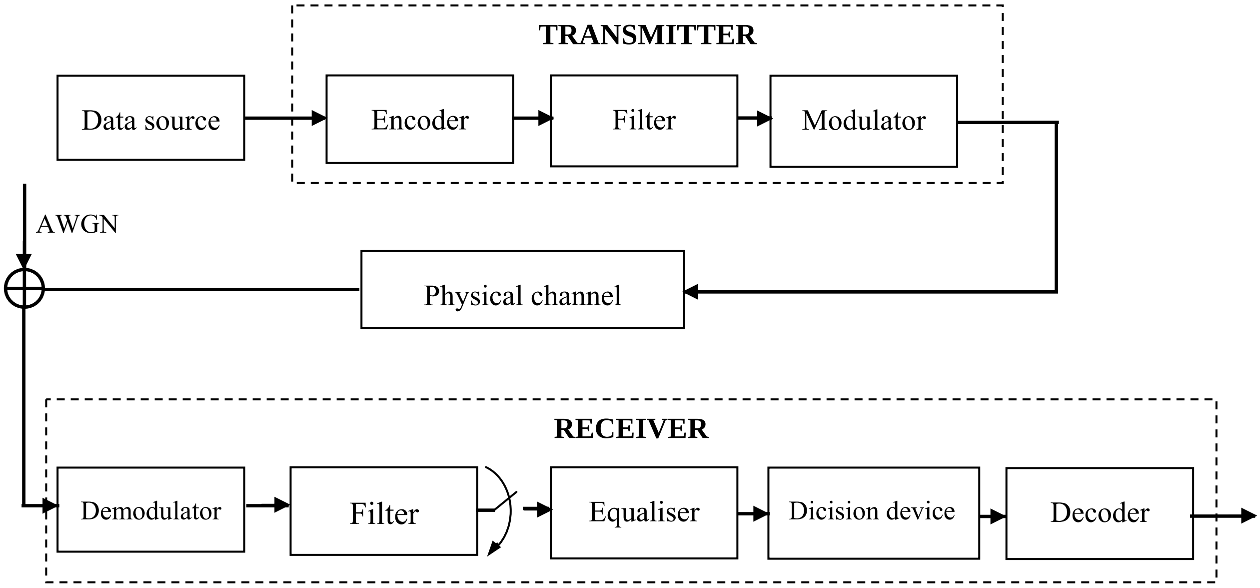


Figure 2.2. Architecture of a digital communication system. [31]

The information–bearing signals are usually transmitted by some type of carrier modulation. The channel over which the signal is transmitted is limited in bandwidth to an interval of frequencies centred about the carrier, as in double–sideband modulation, or adjacent to the carrier, as in single–sideband modulation. The bandwidth efficiency η_B describes the ability of a modulation scheme to accommodate data within a limited bandwidth (equation 2.1). There is a fundamental upper bound on achievable bandwidth efficiency. Shannon's channel coding theorem states that for an arbitrary small probability of error, the maximum possible bandwidth efficiency $\eta_{B\max}$ is limited by the noise in the channel, and is given by the channel capacity formula [53]. The Shannon's bound applies for additive white Gaussian noise is given by

$$\eta_{B\max} = \frac{C}{B} = \log_2(1+\frac{S}{N})$$

$$or \qquad \qquad ....(2.1)$$

$$C = B\log_2(1+\frac{S}{N})$$

Where C is the channel capacity (in bps), B is the channel bandwidth, S is the average power of the transmitted signal, and N is the power spectral density of the white

Gaussian noise, S/N is also called the signal-to-noise ratio. Shannon [54] also showed that if the transmission rate is less than the channel capacity, then it is possible to achieve reliable communication, with as small an error probability as desired. The efficient use of this restricted bandwidth is achieved through the choice of the encoding scheme and the design of the transmitter filter, also called the modulating filter.

The modulator, on the other hand, places the baseband signal over a high frequency carrier for transmission in the allocated spectrum using a modulation scheme. Some of the typical modulation schemes used in digital communication systems are amplitude shift keying (ASK), frequency shift keying (FSK), pulse amplitude modulation (PAM), phase shift keying (PSK), and quadrature phase shift keying (QAM) modulation.

The communication channels may be of various types. In wireless transmission, the channel may be the atmosphere (free space). On the other hand, telephone channels usually utilise a variety of physical media, such as wire lines, optical fibre cables, and wireless microwave links. Whatever the physical medium used for transmission of the information, the transmitted signal is corrupted in a random manner by a variety of possible mechanisms, such as additive thermal noise generated by electronic devices, man–made noise (e.g. automobile ignition noise) and atmospheric noise (e.g. electrical lightning discharges during thunderstorms) [55]. This interference is modelled as random, additive white Gaussian noise (AWGN) at the output of a noise–free channel. Another essential characteristic of the transmission of information through a channel is that the bandwidth allocated for the channel is often limited, resulting in the dispersion of power between neighbour symbols in the transmitted sequence. This distortion of the channel is called intersymbol interference (ISI).

At the receiver the signal is first demodulated to recover the transmitted signal in its baseband form. Then the demodulated signal that is processed by the receiver filter, also called receiver demodulating filter, should be ideally matched to the transmitter filter and channel impulse response. Normally the channel transfer function is not known to the receiver and may be non-stationary. For this reason the receiver is usually matched to the transmitter filter only.

The output of the receiver filter is sampled at the symbol rate and the resulting discrete time signal is passed to the equaliser. The equaliser in the receiver removes the ISI distortion introduced due to the limited bandwidth of the channel. The decision device reconstructs the encoded transmitted binary sequence, based on the soft decisions made by the equaliser. Finally, the decoder performs the reverse operation of the encoder and reconstructs the sequence of transmitted information symbols.

2.3 Channel Characteristics

The transmitted signal is corrupted in a random manner by additive noise. Additive disturbances are thermal noise, man-made noise, and atmospheric noise. Interference from other users of the channel is another form of additive noise that often arises in both wireless and wire line communication systems.

In some radio communication channels, such as the ionospheric channel that is used for long-range, short-wave radio transmission, another form of signal degradation is multipath propagation. Such signal distortion is characterised as a non-additive signal disturbance, which manifests itself as time variations in the signal amplitude, usually called fading [56].

Both additive and non-additive signal distortions are usually characterised as random phenomena and described in statistical terms. The effect of these signal distortions must be taken into account in the design of the communication system.

A communications channel may be described in terms of its characteristic properties. These channel characteristics include bandwidth (how much information can a channel accommodate), quality (how reliably can the information be correctly conveyed across the channel), and whether the channel is dedicated (to a single source) or shared (by multiple sources).

Obviously a higher bandwidth in a channel allows more information to be conveyed per unit of time. High bandwidths mean that more users can share the channel, depending on their means of accessing it. High bandwidths also allow more demanding applications (such as graphics) to be supported for each user of the channel.

Reliability of communication is obviously important. A low quality channel is prone to distorting the messages it conveys; a high quality channel preserves the integrity of the messages it conveys. Depending on the quality of the channel in use between communicating entities, the probability of the destination correctly receiving the message from the source might be either very high or very low. If the message is received incorrectly it needs to be retransmitted.

If the probability of receiving a message correctly across a channel is too low, the system (source, channel, message, and destination) must include mechanisms which overcome the errors introduced by the low quality channel. Otherwise no useful communication is possible over that channel. These mechanisms are embodied in the communication protocols employed by the corresponding entities.

The effective bandwidth describes what an application experiences and depends on the quality of service (QOS) provided by the channel. For example, modems scale back their transmission speed based largely on their perception of channel quality in order to optimally use the transmission medium.

In general, shared and reliable channels are more resource efficient than those which enjoy neither of these characteristics. Shared channels enjoy greater efficiency than dedicated ones because most data communication is burst in nature, with long idle periods punctuated by brief message transmissions. Reliable channels are more efficient than unreliable ones because retransmissions are not required as often (because there are fewer transmission-induced errors).

2.4 Channel Distortions

On propagating through a channel, signals are shaped and distorted by the frequency response and the attenuating channel characteristics. There are two main manifestations of channel distortions: magnitude distortion and phase distortion. In addition, in radio communication, we have the multipath effect, in which the transmitted signal may take several different routes to the receiver, with the effect that multiple versions of the signal with different delay and attenuation arrive at the receiver.

The common types of channel distortion are:

- Frequency-dependent phase shifts, giving rise to differential group delay
- Gain variations with frequency caused by the channel filtering effect
- Gain variations with time as experienced in a radio/infra red channel
- Frequency offsets between transmitter and receiver due to Doppler Shift or local oscillator errors.

Distortion can be introduced within the transmitter, the receiver and the channel. In some cases it can be corrected using channel equalisers, and gain and frequency control systems. Unlike noise and interference, distortion disappears when the signal is turned off.

2.4.1 Multipath Propagation

A transmitted signal is probably subject to reflections from buildings, mountains or other reflectors. This leads to the received signal becoming distorted or even temporarily suppressed. This is called multipath channel. Associated with each path are a propagation delay and an attenuation factor. Both the propagation delays and the attenuation factors vary slowly with time as a result of changes in the channel environment.

The amplitude attenuation factor varies slowly and must change sufficiently to cause a significant change in the received signal. Since the carrier frequency is quite high, small delay differences give rise to large phase changes. Multipath propagation channel that is embodied in the received signal results in signal fading. The frequency selective, fast fading phenomenon is primarily a result of the time variations in the phases. That is, the randomly time-variant phases associated with the vectors result on occasion in the vectors adding destructively.

When this occurs, the resulting received signal is very small, or practically zero. At other times, the vectors add constructively, making the received signal large. Thus, the amplitude variations in the received signal, so-called signal fading, are due to the time-varying multipath characteristics of the channel.

If the transmitter, receiver, or reflectors are moving within a multipath environment, the path lengths will vary with time and so the relative phases between signals will also vary with the position of the users. The result is that the receiver experiences a combined signal with fluctuating amplitude and phase as a function of time [57].

In mobile radio propagation, fading is used to describe the rapid fluctuations of the amplitudes, phases, or multipath delays of a radio signal over a short period of time or travel distance, so that large-scale path loss effect may be ignored. Fading is caused by interference between two or more versions of the transmitted signal which arrive at the receiver at slightly different times. These waves, called multipath waves, combine at the receiver antenna to give a resultant signal which can vary widely in amplitude and phase, depending on the distribution of the intensity and relative propagation time of the waves and the bandwidth of the transmitted signal [51].

Multipath in the radio channel creates small-scale fading effects. The three most important effects are:

- Rapid changes in signal strength over a small travel distance or time interval
- Random frequency modulation due to varying Doppler shifts on different multipath signals
- Time dispersion (echoes) caused by multipath propagation delays.

In wireless channels, the high data rates inevitably give rise to severe frequency selectivity and multipath interference. Multipath channels pose many challenging signal processing problems for designers of high-performance receivers. To overcome channel selectivity transmission techniques, such as code-division multiple access (CDMA) and orthogonal frequency-division multiplexing (OFDM), are used. At the same time, however, the scattering-rich environment engendered by the presence of multiple propagation paths leads to tremendous potential diversity gains that can be exploited through the use of suitable transmission/reception strategies such as multiple-input multiple-output (MIMO) techniques [58].

2.4.2 Intersymbol Interference

The basic problem is that of digital transmission through a dispersive medium (such as a mobile radio channel which) introduces distortion due to multipath effects. This means that data symbols interfere with each other.

Let us consider what happens when pulsed information is transmitted over an analog channel such as a phone line or airwaves. Even though the original signal is a discrete time sequence (or a reasonable approximation), the received signal is a continuous time signal. Heuristically, one can consider that the channel acts as an analog low-pass filter, thereby spreading or smearing the shape of the impulse train into a continuous signal whose peaks relate to the amplitudes of the original pulses. Mathematically, the operation can be described as a convolution of the pulse sequence by a continuous time channel response. The operation starts with the convolution integral:

$$x(k) = h(k) * s(k) = \int_{-\infty}^{\infty} h(\tau)s(k-\tau)d\tau = \int_{-\infty}^{\infty} s(\tau)h(k-\tau)d\tau \tag{2.2}$$

where $x(k)$ is the received signal, $h(k)$ is the channel impulse response, and $s(k)$ is the input signal. The second half of the equation above is a result of the fact that convolution is a commutative operation.

Component $s(k)$ is the input pulse train, which consists of periodically transmitted impulses of varying amplitudes. Therefore,

$$\begin{aligned} s(k) &= 0 \qquad for\, k \neq nT \\ s(k) &= S_n \qquad for\, k = nT \end{aligned} \tag{2.3}$$

where T represents the symbol period. This means that the only significant values of the variable of integration in the above integral are those for which $k = nT$. Any other value of k amounts to multiplication by 0. Therefore $x(k)$ can be written as

$$x(k) = \sum_{n=-\infty}^{\infty} s_n h(k - nT) \tag{2.4}$$

This representation of $x(k)$ more closely resembles the convolution sum, however, that it still describes a continuous time system. It shows that the received signal consists of the sum of many scaled and shifted continuous time system impulse responses. The impulse responses are scaled by the amplitudes of the transmitted pulses of $x(k)$.

In equation 2.4, the first term is the component of $x(k)$ due to the N^{th} symbol. It is multiplied by the centre tap of the channel-impulse response. The other product terms in the summation are ISI terms. The input pulses in the neighbourhood of the N^{th} symbol are scaled by the appropriate samples in the tails of the channel-impulse response.

2.4.3 Noise

The Gaussian process has been always the dominant noise model in communications and signal processing literature, mainly because of the central limit theorem. In addition, the Gaussian assumption often leads to analytically tractable solutions [1].

Unfortunately, in many communication channels, the observation noise exhibits Gaussian, as well as impulsive characteristics. The sources of impulsive noise may be either natural, or man made. It may include atmospheric noise or ambient acoustic noise. It might come from relay contacts, electromagnetic devices, electronic apparatus, or transportation systems, switching transients, and accidental hits in telephone lines [2].

Most of the systems are optimised under the Gaussian assumption and their performance is significantly degraded by the occurrence of impulsive noise [4, 5]. That is, more realistic statistical models must be used. Impulsive noise is more likely to exhibit sharp spikes or occasional bursts of outlying observations than one would expect from normally distributed signals.

Noise may be defined as any unwanted signal that interferes with the communication, measurement or processing of an information-bearing signal. Noise is characterised as random, unpredictable electrical signals from natural sources. Noise is present in various degrees in almost all environments. For example, in a digital cellular mobile telephone system, there may be several variety of noise that could degrade the quality of communication, such as acoustic background noise, thermal noise, electromagnetic radio-frequency noise, co-channel interference, radio-channel distortion, echo and processing noise. Noise can cause transmission errors and may even disrupt a communication process; hence noise processing is an important part of modern telecommunication and signal processing systems. The success of a noise processing method depends on its ability to characterise and model the noise process, and to use the noise characteristics advantageously to differentiate the signal from the noise.

Depending on its source, a noise can be classified into a number of categories, indicating the broad physical nature of the noise, as follows:

a. Acoustic noise: emanates from moving, vibrating, or colliding sources and is the most familiar type of noise present in various degrees in everyday environments. Acoustic noise is generated by such sources as moving cars, air-conditioners, computer fans, traffic, people talking in the background, wind, rain, etc.

b. Electromagnetic noise: present at all frequencies and in particular at the radio frequencies. All electric devices, such as radio and television transmitters and receivers, generate electromagnetic noise.

c. Electrostatic noise: generated by the presence of a voltage with or without current flow. Fluorescent lighting is one of the most common sources of electrostatic noise.

d. Channel distortions, echo, and fading: due to non-ideal characteristics of communication channels. Radio channels, such as those at microwave frequencies used by cellular mobile phone operators, are particularly sensitive to the propagation characteristics of the channel environment.

e. Processing noise: the noise that results from the digital/analog processing of signals, e.g. quantisation noise in digital coding of speech or image signals, or lost data packets in digital data communication systems.

Depending on its frequency or time characteristics, a noise process can be classified into one of several categories as follows:

a. Narrowband noise: a noise process with a narrow bandwidth such as a 50/60 Hz from the electricity supply.

b. White noise: purely random noise that has a flat power spectrum. White noise theoretically contains all frequencies in equal intensity.

c. Band-limited white noise: a noise with a flat spectrum and a limited bandwidth that usually covers the limited spectrum of the device or the signal of interest.

d. Coloured noise: non-white noise or any wideband noise whose spectrum has a non-flat shape; examples are pink noise, brown noise and autoregressive noise.

e. Impulsive noise: consists of short-duration pulses of random amplitude and random duration.

f. Transient noise pulses: consists of relatively long duration noise pulses.

For convenience, most research works assume noise to fall predominantly into the class of Additive White Gaussian Noise (AWGN) which does indeed form an adequate classification for most cases. However, this is a general simplification of the whole noise issue.

In order to use in the design of communication systems, it is convenient to construct mathematical models that reflect the most important characteristics of the transmission channels. Then, the mathematical model for the channel is used in the design of the channel encoder and modulator at the transmitter and the demodulator and channel decoder at the receiver. A brief description of channel models that are frequently used to characterise many of the physical channels that are encountered in practice is given below:

2.4.3.1 The Additive Noise Channel

The simplest mathematical model for a communication channel is the additive noise channel, illustrated in Figure 2.3. In this model, the transmitted signal $x(k)$ is corrupted by an additive random noise process $n(k)$. Physically, the additive noise process may arise from electronic components and amplifiers at the receiver of the communication system, or from interference encountered in transmission as in the case of radio signal transmission.

If the noise is introduced primarily by electronic components and amplifiers at the receiver, it may be characterised as thermal noise. This type of noise is characterised statistically as a Gaussian noise process. Hence, the resulting mathematical model applies to a broad class of physical communication channels, and because of its mathematical tractability this is the predominant channel model used in the channel is usually called the additive Gaussian noise channel.

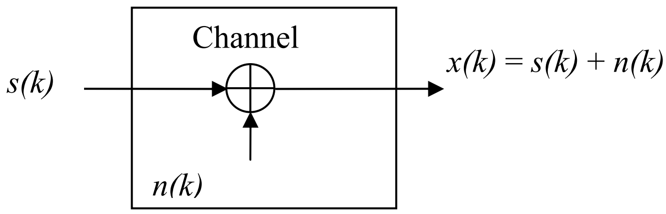


Figure 2.3 The additive Gaussian noise channel [59]

2.4.3.2 The Linear Filter Channel

In some physical channels such as wire line telephone channels, filters are used to ensure that the transmitted signals do not exceed specified bandwidth limitations and thus do not interfere with one another. Such channels are generally characterised mathematically as linear filter Channels with additive noise, Figure 2.4. Hence, if the channel input is the signal *s(k)* the channel output is the signal

$$x(k) = s(k) * h(k) + n(k) = \int_{-\infty}^{+\infty} h(\tau) s(k-\tau) d\tau + n(k) \qquad (2.5)$$

where $h(\tau)$ is the impulse response of the linear filter and * denotes convolution.

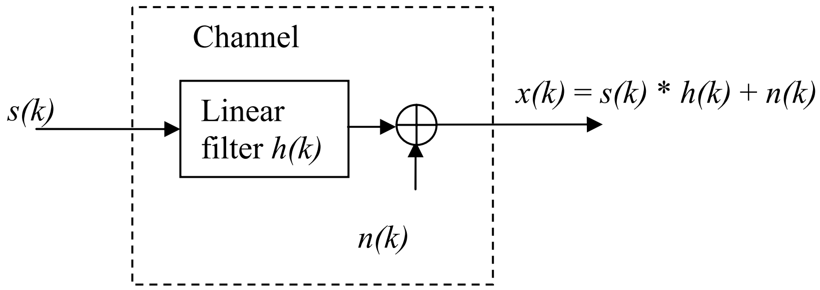


Figure 2.4 The linear filter channel with additive noise [59]

When the signal undergoes attenuation in transmission through the channel, the received signal is

$$x(k) = \alpha\, s(k) + n(k) \qquad \ldots\ldots(2.6)$$

where α represents the attenuation factor.

2.4.3.3 The Linear Time-Variant Filter Channel

Physical channels such as underwater acoustic channels and ionospheric radio channels which result in time-variant multipath propagation of the transmitted signal may be characterised mathematically as time-variant linear filters. Such linear filters are characterised by a time-variant channel impulse response $h(\tau;k)$, where $h(\tau;k)$ is the response of the channel at time k due to an impulse applied at time k-τ. Thus, τ represents the elapsed-time variable. The linear time-variant filter channel with additive noise is illustrated Figure 2.5. For an input signal *s(k)*, the channel output signal is

$$x(k) = s(k) * h(\tau;k) + n(k) = \int_{-\infty}^{+\infty} h(\tau;k)s(k-\tau)d\tau + n(t) \tag{2.7}$$

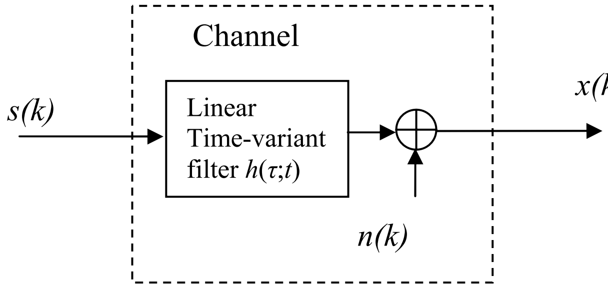


Figure 2.5 Linear time-variant filter channel with additive noise [61]

A good model for multipath signal propagation through physical channels, such as the ionosphere and mobile cellular radio channels, is a special case of (2.7) in which the time-variant impulse response has the form

$$h(\tau;k) = \sum_{n=1}^{L} a_n(k)\delta(k-\tau_k) \tag{2.8}$$

where the $\{a_n(k)\}$ represents the possibly time-variant attenuation factors for the L multipath propagation paths. If (2.8) is substituted into (2.7), the received signal has the form

$$h(\tau;k) = \sum_{n=1}^{L} a_n(k)\delta(k-\tau_k) + n(k) \tag{2.9}$$

Hence, the received signal consists of L multipath components, where each component is attenuated by $\{a_n(k)\}$ and delayed by $\{\tau_n\}$.

The three mathematical models described above adequately characterise a large majority of physical channels encountered in practice. These three channel models are used for the analysis and design of communication systems [59].

2.5. Structure of Channel Equalisation System

As shown above, in digital communications, channels are affected by various linear, nonlinear, and additive distortions. Various equalisers have been applied to equalize these distortions and recover the original transmitted signal. Linear equalisers could not reconstruct the transmitted signal when channels have significant non-linear distortion. Since non-linear distortion is often encountered on time-variant channels, linear equalisers do not perform well in such kind of channels. When a channel has time-varying characteristics and the channel model is not precisely known, adaptive equalisation is applied. Neural networks are widely used for the equalisation of nonlinear channel distortion [60, 61, 62, 63, 65, 67]. One class of adaptive equalisers is based on multilayer perceptron (MLP) and radial basis functions (RBF) [60-66]. The MLP equalisers require long time for training and are sensitive to the initial choice of network parameters [60, 64, 65]. The RBF equalisers are simple and require less time for training, but usually require a large number of centers, which increase the complexity of computation [61, 62, 66]. Another effective way for the development of adaptive equalisers for nonlinear channels is the use of fuzzy technology. This type of adaptive equalisers can process numerical data and linguistic information in natural form [37, 45, 46, 49, 50]. Human experts determine fuzzy IF-THEN rules using input-output data pairs of the channel. These rules are used to construct the filter for the nonlinear channel. In these systems the incorporation of linguistic and numerical information improves the adaptation speed and the bit error rate (BER) performance [45].

Sometimes the construction of proper fuzzy rules for equalisers is difficult. One of the effective technologies for the construction of equaliser's knowledge base is the use of neural networks. Much effort has been devoted to the development and improvement of fuzzy neural network models. The structures of most of neuro-fuzzy systems mainly implement the TSK-type or Mamdani-type fuzzy reasoning mechanisms. Adaptive neuro-fuzzy inference system (ANFIS) implements TSK-type fuzzy system [70], where the consequent parts include linear functions. This fuzzy system can describe the considered problem by means of combination of linear functions.

In this thesis, the structure of the neuro-fuzzy network based equalisation system has been proposed. The neuro-fuzzy network is used for equalisation of nonlinear channel distortion. The neuro-fuzzy network allows short training time of the equaliser and gives better results in terms of bit error rate (BER), at the cost of computational strength.

Neuro-fuzzy network based equalisers are nonlinear adaptive equalisers. They are called adaptive for the reason that they are capable of self adjustment, where these equalisers can change in accordance to their input signals, where they have the ability to update their coefficients. The adaptive equaliser requires two signals:

- the input signal $s(k)$

- the reference (or desired) input $s_d(k)$

An adaptive equaliser is used to compensate for the distortion caused by the transmission medium, and its operation involves a training mode followed by a tracking mode. The equaliser is trained by transmitting a known test data sequence. A synchronised version of the test signal is generated in the receiver, meaning the adaptive equaliser is now supplied with a desired response. The equaliser output is subtracted from this desired response to give an estimation error. This estimation error is used to adaptively adjust the coefficients of the equaliser to their optimum values. When the training is completed, the adaptive equaliser tracks possible time variations in channel characteristics during transmission. It does this by using a receiver estimate of the transmitted sequence as a desired response. The receiver estimate is obtained by applying the equaliser output to a decision device.

In Figure 2.6 the proposed structure of channel equalisation system is given. $s(k)$ is binary input signals that are to be transmitted through the channel. Input signal are distorted by noise $n(k)$ in the channel. In particular case the noise is the additive Gaussian noise. The channel may be non-linear, but the input-output symbol sequence map is assumed to be unambiguous. In modern interference-limited cellular telephony systems, the main error source is the Intersymbol Interference (ISI), rather than the thermal noise. The ISI consists in the spreading of symbol information through subsequent signal samples, and is the main problem in the relatively high SNR, typical of most existing transmission systems.

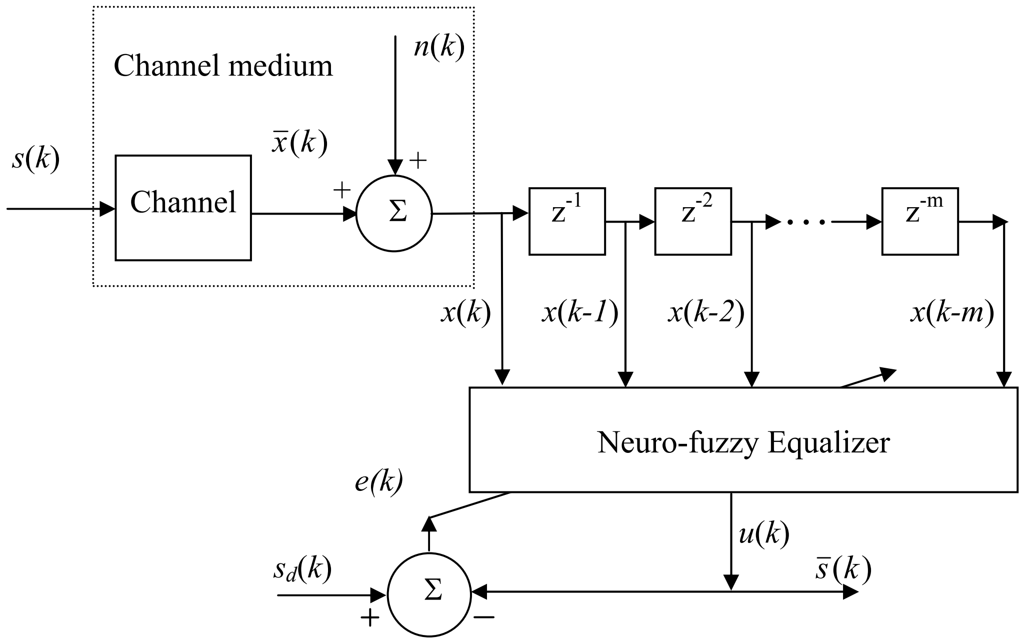


Figure 2.6 Structure of a neuro-fuzzy equalisation system

At the receiver, the equaliser is applied in order to compensate for channel distortion. The purpose of the equaliser is to estimate $s(k)$, minimizing the combined effects of ISI and noise. In particular, in this thesis the neuro-fuzzy equalisation system is proposed. It makes use of a set of delayed input samples as an input signal. Neuro-Fuzzy Networks (NFNs) can be successfully applied to the adaptive equalisation of digital nonlinear communication channels. NFNs are able to yield significant performance when little information is available about the channel model. This fact can be explained by the very general assumptions made on the mapping from the received signal to the output symbol space that recast the demodulation problem as a classification task.

2.6 Summary

The discussion of data transmission system, its main components, their functions, channel characteristics has been considered in this chapter. Interferences, Noises and their types have been described. As a corollary, the effect of each type of noise and how they present on the system has been learned.

Also in this chapter the equalisation problem of channel distortion is introduced. An adaptive equaliser is used to compensate for the distortion caused by the transmission medium, and its operation involves a training mode followed by a tracking mode. The equaliser is trained by transmitting a known test data sequence. A synchronised version of the test signal is generated in the receiver, meaning the adaptive equaliser is now supplied with a desired response. The equaliser output is subtracted from this desired response to give an estimation error. This estimation error is used to adaptively adjust the coefficients of the equaliser to their optimum values. When the training is completed, the adaptive equaliser tracks possible time variations in channel characteristics during transmission. It does this by using a receiver estimate of the transmitted sequence as a desired response. The receiver estimate is obtained by applying the equaliser output to a decision device.

CHAPTER 3

MATHEMATICAL BACKGROUND OF A NEURO-FUZZY EQUALISER

3.1 Overview

Neural networks and fuzzy systems have established their reputation as alternative approaches to signal processing. Both have certain advantages over conventional methods, especially when vague data or prior knowledge is involved. However, their applicability suffered from several weaknesses of the individual models. Therefore, combinations of neural networks with fuzzy systems have been proposed, where both models complement each other. These neural fuzzy or neuro-fuzzy systems overcome some of the individual weaknesses and offer some appealing features.

Since 1990s, there have been large research efforts aimed at synthesizing fuzzy logic with neural networks. This combination of neural networks and fuzzy logic seems natural because the two approaches generally attack the design of "intelligent" systems from different angles. Neural networks provide algorithms for learning, classification, and optimization, whereas fuzzy logic deals with reasoning on a higher (semantic or linguistic) level. Consequently, the two technologies complement each other. By integrating neural networks with fuzzy logic, it is possible to bring the low-level computational power and learning of neural networks into fuzzy logic systems. The synergism of integrating neural networks with fuzzy logic systems into a functional system with low-level learning, high-level thinking, and reasoning transforms the burden of the tedious design problems of the fuzzy logic decision systems to the training/learning of connectionist neural networks.

In this chapter a brief description of neural networks and fuzzy logic will be given. The structure and operations algorithms of neuro-fuzzy system that is used for channel equalisation will be described.

3.2. Neuro-Fuzzy System

There are many ways to combine neural networks and fuzzy logic. Neural networks provide algorithms for numeric classification, optimization, and associative storage and recall. Working at the semantic level, fuzzy logic provides processing of inexact or approximate data. By incorporating fuzzy logic techniques into a neural network, we can obtain more flexibility. Fuzzy neural networks provide greater representation power, have higher processing speeds, and are more robust than conventional neural networks. Fuzzy neural networks are in fact "fuzzified" neural networks. The integration of neural networks and fuzzy logic allows decreasing the nodes of network and train the network in a shorter time.

3.3 Fuzzy Inference Systems

3.3.1. Architecture of Fuzzy Inference Systems

Fuzzy inference systems are also known as fuzzy-rule-based systems. A fuzzy inference system is composed of the following functional blocks (see Figure.3.1);

- A fuzzification inference which transform the crisp inputs into degrees to match with linguistic values;

- A rule base containing a number of fuzzy if-then rules;

- A database which defines the membership functions of the fuzzy sets used in the fuzzy rules;

- A decision-making unit which performs the inference operations on the rules;

- A defuzzification inference which transform the fuzzy results of the inference into crisp output.

The following steps demonstrate the inference operations upon fuzzy if-then rules:

1. Compare the input variable with the membership functions on the premise part to obtain the membership values (or compatibility measures) of each linguistic label. (This step is often called fuzzification).

2. Combine (through a specific *T-norm* operator, usually multiplication or min.) the membership values on the premise part to get firing strength (weight) of each rule.

3. Generate the qualified consequent (either fuzzy or crisp) of each rule depending on the firing strength.

4. Aggregate the qualified consequent to produce a crisp output. (This step is called defuzzification.)

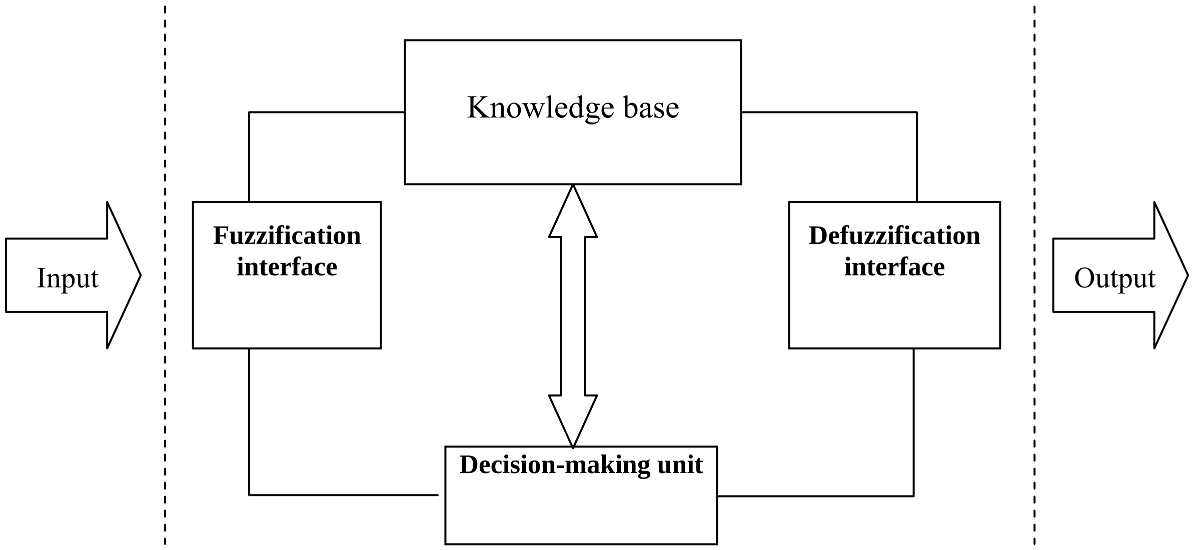


Figure 3.1 Structure of fuzzy inference system

3.3.2. Rule Base Fuzzy IF-THEN Rules

One of the basic blocks of a fuzzy system is the fuzzy knowledge base that includes a set of fuzzy If-THEN rules. Fuzzy if-then rules or fuzzy conditional statements are expressions of the form

$$\text{If } u \text{ is } A \text{ Then } y \text{ is } B \tag{3.1}$$

Here u and y are input and output linguistic variables. A and B are labels of the fuzzy sets characterised by appropriate membership functions. A is the premise and B is the consequent parts of the fuzzy rule.

If-Then rules can be represented in many forms. Its simple form is Single Input Single Output (SISO) that is given by formula (3.1). Other forms are Multi-Input Single-Output (MISO) and Multi-Input Multi-Output (MIMO), given by formulas (3.2) and (3.3) respectively.

$$\text{If } u_1 \text{ is } A_1^j \text{ and } u_2 \text{ is } A_2^k \text{ and ,….., and } u_n \text{ is } A_n^l \text{ Then } y_q \text{ is } B_q^p \tag{3.2}$$

$$\text{If } u_1 \text{ is } A_1^j \text{ and } u_2 \text{ is } A_2^k \text{ and ,….., and } u_n \text{ is } A_n^l \text{ Then } y_1 \text{ is } B_1^r \text{ and } y_2 \text{ is } B_2^s \tag{3.3}$$

Fuzzy values A and B are described by the membership functions. Depending on the problem, the forms of membership functions can be different. Figure 3.2 shows some of the mostly used typical shapes of membership functions.

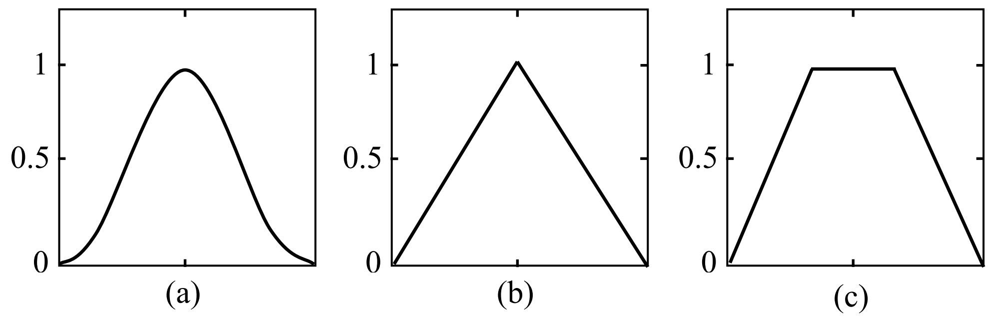


Figure 3.2 Examples of membership functions (a) bell, (b) triangular, (c) trapezoidal

A common example of a function that produces a bell curve is based on the exponential function,

$$\mu(x) = \exp\left[\frac{-(x - x_0)^2}{2\sigma^2}\right] \tag{3.4}$$

Here x is the independent variable on the universe, x_0 is the position of the peak relative to the universe, and σ is the standard deviation.

Triangle and trapezoidal membership functions are described by expressions (3.5) and (3.6), correspondingly

$$\mu(x) = \begin{cases} 1 - \dfrac{\bar{x} - x}{\bar{x} - x_l}, & x_l < x < \bar{x} \\ 1 - \dfrac{x - \bar{x}}{x_r - \bar{x}}, & \bar{x} < x < x_r \end{cases} \tag{3.5}$$

$$\mu(x) = \begin{cases} 1 - \dfrac{\bar{x}_l - x}{\bar{x}_l - x_l}, & x_l < x < \bar{x}_l \\ 1, & \bar{x}_l < x < \bar{x}_r \\ 1 - \dfrac{x - \bar{x}_r}{x_r - \bar{x}_r}, & \bar{x}_r < x < x_r \end{cases} \tag{3.6}$$

When the consequent part of the fuzzy rules are just a mathematical function of the input variables, then such type of rules are called Takagi and Sugeno fuzzy rules. The format of the method is:

$$\text{if } A1(x_1),\ A2(x_2),\ \ldots,\ An(x_n) \text{ then } Y = f(x_1, x_2, \ldots, x_n) \tag{3.7}$$

The antecedent (premise) part is fuzzy. The function *f* in the consequent part is usually a simple mathematical function, linear or quadratic:

$$f = a_0 + a_1 \times x_1 + a_2 \times x_2 + \ldots + a_n \times x_n \tag{3.8}$$

Fuzzy if-then rules have been used extensively in modelling. Due to the qualifiers on the premise parts, each fuzzy if-then rule can be viewed as a local description of the system under consideration. Fuzzy if-then rules form a core part of the fuzzy inference system.

Fuzzification: Fuzzy sets are used to quantify the information in the rule-base, and the inference mechanism operates on fuzzy systems to produce fuzzy sets. The fuzzy system converts its numeric inputs $u_i \in U_i$ into fuzzy sets (the process is called "fuzzification") so that they can be used by the fuzzy system.

Let U_i^* denotes the set of all possible fuzzy sets that can be defined on U_i. Given $u_i \in U_i$, fuzzification transforms u_i to a fuzzy set denoted by A_i^{fuzz}, defined on the universe of discourse U_i. This transformation is produced by the fuzzification operator F defined by

$$F: U_i => U_i*$$

where

$$F(u_i) = A_i^{fuzz},$$

Quite often "singleton fuzzification" is used, which produces a fuzzy set $A_i^{fuzz} \in U_i^*$ with a membership function defined by

$$\mu_{A_i^{fuzz}}(x) = \begin{cases} 1 & \mathrm{x = u_i} \\ 0 & \text{otherwise} \end{cases}$$

Any fuzzy set with this form for its membership function is called a "singleton." Basically, the singleton fuzzy set is a different representation for the number u_i. Singleton fuzzification is generally used in implementations since, without the presence of noise, we are absolutely certain that u_i takes on its measured value (and no other value), and since it provides certain savings in the computations needed to implement a fuzzy system. For example, "Gaussian fuzzification," which involves forming bell-shaped membership functions about input points, or triangular fuzzification, which uses triangles [68].

Generally the fuzzification process is the act of obtaining a value of an input variable (e.g., $e(t)$) and finding the numeric values of the membership function(s) that are defined for that variable.

3.3.4 Inference Mechanism

The inference mechanism has two basic tasks:

I. Determining the extent to which each rule is relevant to the current situation as characterised by the inputs u_i, $i = 1, 2, ..., n$ (this task called "matching");

II. Drawing conclusions using the current inputs u_i and the information in the rule-base (we call this task an "inference step").

Let A_1^j x A_2^k x . . . x A_n^l be the fuzzy set representing the premise of the i^{th} rule. There are then two basic steps to matching.

Step 1: Combine Inputs with Rule Premises: The first step in matching involves finding fuzzy sets $\overline{A}_1^j$, $\overline{A}_2^k$, . . . , $\overline{A}_n^l$, with membership functions

$$\mu_{\overline{A}_1^j}(u_1) = \mu_{A_1^j}(u_1) * \mu_{\overline{A}_1^{fuz}}(u_1)$$

$$\mu_{\overline{A}_2^k}(u_2) = \mu_{A_2^k}(u_2) * \mu_{\overline{A}_2^{fuz}}(u_2)$$

.

.

$$\mu_{\overline{A}_n^j}(u_n) = \mu_{A_n^j}(u_n) * \mu_{\overline{A}_n^{fuz}}(u_n)$$

(for all j, k, . . . , l) that combine the fuzzy sets from fuzzification with the fuzzy sets used in each of the terms in the premises of the rules. If singleton fuzzification is used, then each of these fuzzy sets is a singleton that is scaled by the premise membership function

(e.g. $\mu_{\overline{A}_n^j}(\overline{u}_n) = \mu_{A_n^j}(u_n)$ for $\overline{u}_1 = u_1$ and $\mu_{\overline{A}_n^j}(\overline{u}_n) = 0$ for $\overline{u}_1 \neq u_1$).

That is, with singleton fuzzification we have $\mu_{\bar{A}_n^{fuz}}(u_i) = 1$, for all i = l, 2, ..., n for the given u_i inputs so that

$$\mu_{\bar{A}_1^j}(u_1) = \mu_{A_1^j}(u_1)$$

$$\mu_{\bar{A}_2^k}(u_2) = \mu_{A_2^k}(u_2)$$

$$\cdot$$

$$\mu_{\bar{A}_n^l}(u_n) = \mu_{A_n^l}(u_n)$$

Step 2: Determine Which Rules Are On: In the second step, we form membership values $\mu_i(u_1,\ u_2,\ \dots\ ,\ u_n)$ for the i^{th} rule's premise that represent the certainty that each rule premise holds for the given inputs. Define

$$\mu_i(u_1,\ u_2,\ \dots\ ,\ u_n) = \mu_{\bar{A}_1^j}(u_1)\mu_{\bar{A}_2^k}(u_2)\dots\mu_{A_n^l}(u_n)$$

which is simply a function of the inputs $u_i, \mu_i(u_1,\ u_2,\ \dots\ ,\ u_n)$ represents the certainty that the premise of rule i matches the input information when singleton fuzzification is used. This $\mu_i(u_1,\ u_2,\ \dots\ ,\ u_n)$ is simply a multidimensional certainty surface. It represents the certainty of a premise of a rule and thereby represents the degree to which a particular rule holds for a given set of inputs. The inference step determines the implied fuzzy set. Next, the inference step is taken by computing, for the i^{th} rule, the "implied fuzzy set" B_q with membership function

$$\mu_{\bar{B}_q^i}(y_q) = \mu_\iota(u_1, u_2, \dots, u_n) * \mu_{B_q^p}(y_q) \qquad (3.9)$$

The implied fuzzy set $\bar{B}_q^\iota$ specifies the certainty level that the output should be a specific crisp output y_{q} within the universe of discourse y^q, taking into consideration only rule I.

After the inference step the defuzzification is applied to aggregate the conclusions of all the rules that are represented by the implied fuzzy sets.

Defuzzification Methods: There are many defuzzification methods that can be used in fuzzy inference system [69]. In the Centre of Gravity (COG) method the crisp output value *u* is the abscissa under the centre of gravity of the fuzzy set,

$$u = \frac{\sum_i \mu\,(x_i)x_i}{\sum_i \mu\,(x_i)} \tag{3.10}$$

Here x_i is a running point in a discrete universe, and $\mu(x_i)$ is its membership value in the membership function. The expression can be interpreted as the weighted average of the elements in the support set.

The Centre of gravity method for singletons has the following form

$$u = \frac{\sum_i \mu(s_i)s_i}{\sum_i \mu(s_i)} \tag{3.11}$$

Here s_i is the position of singleton *i* in the universe, and $\mu\ (s_i)$ is equal to the firing strength α_i of rule *i*. This method has a relatively good computational complexity and *u* is differentiable with respect to the singletons s_i, which is useful in neuro-fuzzy systems.

Another more used method is the Center of Average (COA). A crisp output y_q^{Crisp} is chosen using the centers of each of the output membership functions and the maximum certainty of each of the conclusions represented with the implied fuzzy sets, and is given by

$$y_q^{Crisp} = \frac{\sum_{i=1}^{R} b_i^q \sup_{yq}\{\mu\, B_q^i(y_q)\}}{\sum_{i=1}^{R} \sup_{yq}\{\mu\, B_q^i(y_q)\}} \tag{3.12}$$

where "*sup*" denotes the "supermum" (i.e., the least upper bound which can often be thought of as maximum value). Hence, $\sup_x\{\mu(x)\}$ can be simply thought of as the highest value of $\mu(x)$.

The inference mechanisms on different type of fuzzy system [70, 71] are graphically given in Figure 3.3. Depending on the types of fuzzy reasoning and fuzzy if-then rules employed, most fuzzy inference systems can be classified into three types.

In Type 1 fuzzy systems the overall output is the weighted average of each rule's crisp output introduced by rule's firing strength and the output membership functions.

In Type 2 systems the overall fuzzy output is derived by applying 'max' operation to the qualified fuzzy outputs (each of which is equal to the minimum of firing strength and the output membership function of each rule).

The above mentioned defuzzification algorithm (for example, centroid of an area) can be used to choose the final crisp output based on the overall fuzzy output.

Type 3 is Takagi and Sugeno's fuzzy if-then rules, where the output of each rule is a linear combination of input variables plus a constant term, and the output is the weighted average of each rule's output.

Fig.3.3 utilizes a two-rule two-input fuzzy inference system to show different types of fuzzy rules and fuzzy reasoning mentioned above.

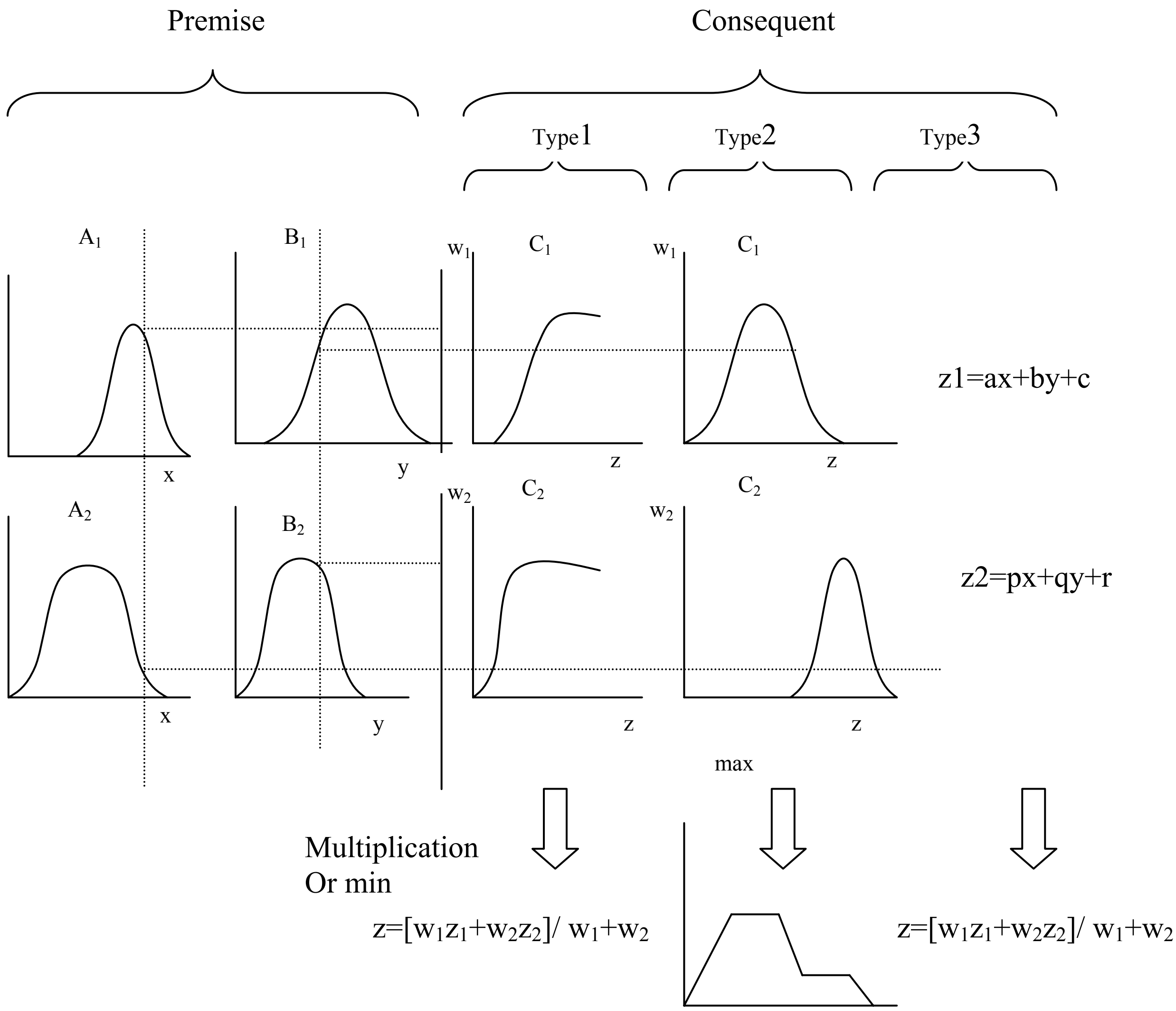


Figure 3.3 Types of fuzzy reasoning mechanisms

3.4 The Artificial Neural Networks

The basic unit of neural networks, the artificial neurons, simulates the four basic functions of natural neurons. Artificial neurons are much simpler than the biological neuron;

Figure 3.4 shows the basics of an artificial neuron.

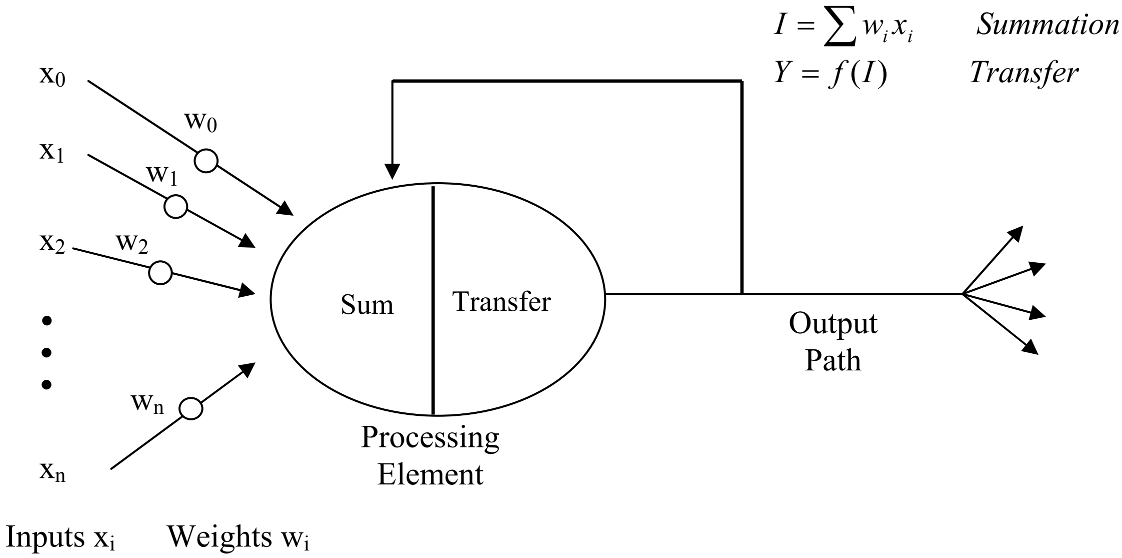


Figure 3.4 Artificial Neuron

Note that various inputs to the network are represented by the mathematical symbol *x*(*n*). Each of these inputs are multiplied by a connection weight, these weights are represented by *w*(*n*). In the simplest case, these products are simply summed, fed through a transfer function to generate a result, and then output.

The output of the artificial neuron shown in Figure 3.4 is calculated as

$$y_i = f(\sum_{j=1}^{n} w_{ij} x_j - \theta_i) \tag{3.13}$$

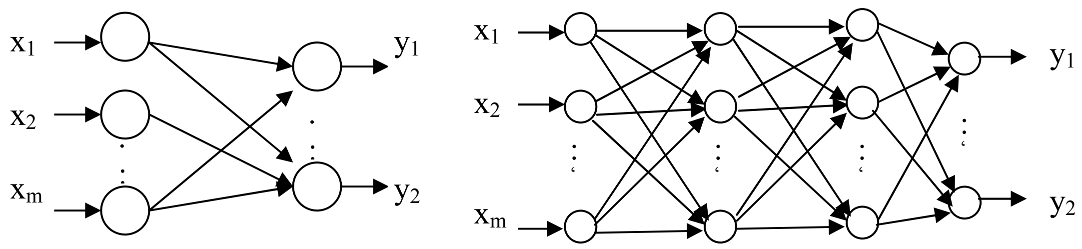
x_i is the input, y_j is the output of the neuron, w_{ij} is the weight coefficients, θ_i is the bias, f is the activation function.

The activation function can be either linear or nonlinear [72]. A nonlinear sigmoid function is often used as the activation function (3.14).

$$y_j = \frac{1}{1 + \exp[-(\sum_{j=1}^{n} w_{ij} x_i - \theta_j)]} \tag{3.14}$$

Neural networks consist of a set of neurons in layer(s). The neurons are interconnected by weighted connections. Neurons in the network are called processing elements (PE's) that simply multiplies an input by a set of weights, and nonlinearly transforms the result into an output value. The power of neural computation comes from the massive interconnection among the PE's.

Neural networks can be classified as non-recurrent (feedforward), recurrent and full 2connected networks [72]. Feedforward neural network structures may be a single layer or a multilayer (Figure 3.5 (a), (b)).



(a) (b)

Figure 3.5 (a) a single layered, (b) a simple multilayer neural network

3.4.1. Neural Networks Learning, Backpropagation Training Algorithm

Neural networks are sometimes called machine learning algorithms, because changing of its connection weights (training) causes the network to learn the solution to a problem. The strength of connection between the neurons is stored as a weight-value for the specific connection. The system learns new knowledge by adjusting these connection weights. The learning ability of a neural network is determined by its architecture and by the algorithmic method chosen for training.

The training algorithms used for neural networks can be classified into supervised, unsupervised and reinforcement learning [72]. The most frequently used fast training algorithm is supervised algorithm Backpropagation.

In supervised algorithms the information about errors is filtered back through the system and is used to adjust the connection weights between the layers, thus improving performance. The Backpropagation algorithm is the most widely used supervised training algorithm for multilayer feedforward networks (Figure 3.6).

Multilayer feedforward networks normally consist of three, four or more layers. There is always one input layer and one output layer and usually one or more hidden layers although in some classification problems two hidden layers may be necessary. The input layer neurons are not sigmoid unit and they are used for distributing input signals. Once the neurons for the hidden layer are computed, their activations are then fed to the next layer, until all the activations finally reach the output layer, in which each output layer neuron is associated with a specific classification category.

In a fully connected multilayer feedforward network, each neuron in one layer is connected by a weight to every neuron in the previous layer. A bias is also associated with each of these weighted sums. Thus in computing the value of each neuron in the hidden and output layers one must first take the sum of the weighted sums and the bias and then apply *f*(sum) (the sigmoid function) to calculate the neuron's activation.

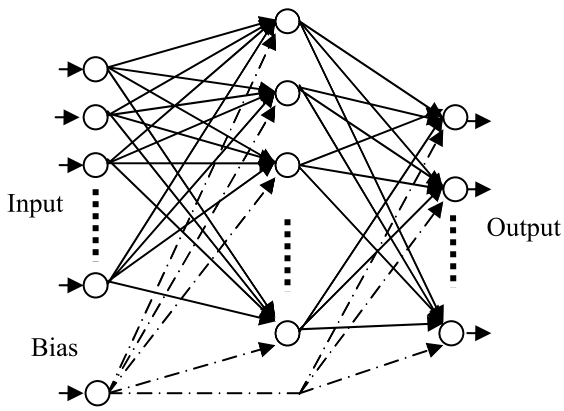


Figure 3.6 Multilayer feedforward network

Let us consider the training processes of three layer neural network. The feedforward phase in network can be described through three steps: input (*I*), hidden (*H*), and output layer (*O*).

- Input layer (*I*): The output of the input layer is equal to the input of the hidden layer as shown in Figure 3.6.

$$Output_I = Input_H$$

- Hidden Layer (*H*): The input of the hidden layer is equal to the sum of multiplying all the input layer outputs by the corresponding weights that connect the two layers as shown in Figure 3.6.

$$Input_H = \sum_i wieght_{IHi} * output_I$$

The output of the hidden layer is the result of the sigmoid transfer function of the hidden layer input.

$$Output_H = \frac{1}{1 + e^{-Input_H}}$$

- Output layer (*O*): The input of the Output layer is equal to the sum of multiplying all the hidden layer outputs by the corresponding weights that connect the two layers as shown in Figure 3.6.

$$Input_O = \sum_j wieght_{HOj} * output_H$$

The output of the output layer is the result of the sigmoid transfer function of the output layer input.

$$Output_O = \frac{1}{1 + e^{-Input_O}}$$

After the feedforward phase the output of the output layer ($Output_O$), is compared with the target value of the neural network (Target), the result is the network error ($Error_O$).

$$Error_o = Target\text{-}Output_o$$

The purpose of the Backpropagation training is to minimise the error of all training patterns by adjusting the weight values, the new value of the hidden-output layer weight is updated according to the following equation.

$$Nwieght_{HO} = Owieght_{HO} + \eta * (error_O * Output_O * (1 - Output_O)) * Ouput_H$$

Where $Nwieght_{HO}$ denotes the new hidden-output layer weights, $Owieght_{HO}$ denotes the old hidden-output layer weight, η is the learning rate. The new weight of the hidden-input layer weight is also updated according to the following equation [72].

$$Nwieght_{IH} = Owieght_{IH} + \eta * (error_H * Output_H * (1 - Output_H)) * Ouput_I$$

Where $Nwieght_{IH}$ denotes the new hidden-input layer weights, $Owieght_{IH}$ denotes the old hidden-input layer weight.

The following algorithm summarizes the Backpropagation training.

1. Perform the forward-propagation phase for an input pattern and calculate the output error.

2. Change all weight values of each weight matrix using the formula.

Weight (*old*)+*learning rate***output error***output* (*neuron i*)*(*1-output* (*neuron i*))**output* (*neuron i-1*).

3. Go to step 1.

4. The algorithm ends, if all output patterns match their target pattern.

Backpropagation doesn’t require large memory space, easy to implement, the error level is usually accepted and calculated quickly.

3.5. Neuro-Fuzzy Network Models

There are many ways to synthesis neuro-fuzzy network models. Neural networks provide algorithms for numeric classification, optimization, and associative storage and recall. Working at the semantic level, fuzzy logic provides the processing inexact or approximate data. By incorporating fuzzy logic techniques into a neural network, we can obtain more flexibility. Fuzzy neural networks provide greater representation power, have higher processing speeds, and are more robust than conventional neural networks. Fuzzy neural networks are in fact "fuzzified" neural networks.

The integration of Fuzzy logic and neural network has different approaches [73]: Input-Output Approach, Preprocess/postprocess Approach, and Hybrid System Approach.

One of the ways to combine neural networks with fuzzy logic is to design a hybrid system wherein some processing stages are implemented with neural networks and some with a fuzzy inference system. An example of such a system would be a tree classifier in which classification at some node can be carried out with a fuzzy inference system and classification at some other node can be performed using a neural network. The main advantage of such a hybrid system is that when the classification is based on experts' rules we can use the fuzzy inference system, and when the classification is based on training samples we can use a neural network.

Hybrid systems are defined in many different ways. In a simple way, hybrid systems are those composed by more than one intelligent system. Hybrid systems are expected to be more powerful due to the combining advantages of different intelligent techniques. The most popular hybrid systems are: Sequential hybrid systems and incorporated hybrid systems. Incorporated hybrid systems represent the greatest degree of integration. The first system contains the second one or vice-versa. An example is a Neuro-Fuzzy system, where a Fuzzy inference system is implemented using a Neural Network Structure. In this system neural networks are used to implement a fuzzy inference system.

A fuzzy inference system consists of three components. First, a rule base contains a selection of fuzzy rules. Second, a database defines the membership functions used in the rules and, finally, a reasoning mechanism carries out the inference procedure on the rules and given facts.

Jang and Sun [74] presented an adaptive network model for a fuzzy inference system called adaptive network-based fuzzy inference systems (ANFISs). The ANFIS model is a generic model, and neural networks and fuzzy inference systems can be considered as special instances of an adaptive network when proper node functions are assigned [75].

In this thesis the development of hybrid neuro-fuzzy system that implements fuzzy inference mechanism in neural network structure including nonlinear function for channel equalisation is considered.

3.5.1 Nonlinear Neuro-Fuzzy Network

3.5.1.1 Structure of Nonlinear Neuro-Fuzzy Network

The kernel of a fuzzy inference system is the fuzzy knowledge base. In a fuzzy knowledge base, the information that consists of input-output data points of the system is interpreted into linguistic interpretable fuzzy rules. In [75] a training procedure with variable system structure approach for fuzzy inference system is presented. In [76] using α-level procedure the training of fuzzy neural network is carried out and the developed system is applied for control of technological processes. The structures of most of neuro-fuzzy systems mainly implement the TSK-type or Mamdani-type fuzzy reasoning mechanisms. Adaptive neuro-fuzzy inference system (ANFIS) implements TSK-type fuzzy system, in which the consequent parts of ANFIS include linear functions. This neuro-fuzzy system can describe the considered problem by means of combination of linear functions. Sometimes these fuzzy systems need more rules, during modelling complex nonlinear processes in order to obtain the desired accuracy. Increasing the number of the rules leads to increasing the number of neurons in the hidden layer of the network.

To improve the computational power of the neuro-fuzzy system, we use nonlinear functions in the consequent part of each rule. Based on these rules, the structure of the nonlinear neuro-fuzzy network (NNFN) has been proposed. Because of these nonlinear functions, NNFN network has more computational power, and, it can describe nonlinear processes with the desired accuracy. In this thesis, the NNFN is used for equalisation of nonlinear channel distortion. The NNFN network allows in better convergence rate and gives better BER results, at the cost of computational strength [77].

In this thesis, the fuzzy rules that have IF-THEN form and constructed by using nonlinear quadratic functions are used. The use of a nonlinear function allows increasing the computational power of neuro-fuzzy system [78]. They have the following form.

If x_1 is A_{j1} and x_2 is A_{j2} and…and x_m is A_{jm} Then

$$y_j = \sum_{i=1}^{m} (w1_{ij} x_i^2 + w2_{ij} x_i) + b_j \tag{3.15}$$

Here $x_1, x_2, \ldots, x_m$ are input variables, y_j $(j=1,..,n)$ are output variables which are nonlinear quadratic functions, A_{ji} is a membership function for i-th rule of the j-th input defined as a Gaussian membership function. $w1_{ij}$, $w2_{ij}$ and b_j $(i=1,..m, j=1,\ldots,n)$ are parameters of the network.

The fuzzy model that is described by IF-THEN rules can be obtained by modifying parameters of the conclusion and premise parts of the rules. In this thesis, a gradient-descent method is used to train the parameters of the rules in the neuro-fuzzy network structure.

Using fuzzy rules in equation (3.15), the structure of the NNFN is proposed (Fig.3.7). The NNFN includes seven layers.

In the first layer the number of nodes is equal to the number of input signals. These nodes are used for distributing input signals.

In the second layer each node corresponds to one linguistic term. For each input signal entering to the system the membership degree to which the input value belongs to a fuzzy set is calculated. To describe linguistic terms the Gaussian membership function is used.

$$\mu 1_j(x_i) = e^{-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}}, \quad i=1..m, \quad j=1..J \tag{3.16}$$

Here m is number of input signals, J the is number of linguistic terms assigned for external input signals x_i, c_{ij} and σ_{ij} are centre and width of the Gaussian membership functions of the j-th term of i-th input variable, respectively. $\mu 1_j(x_i)$ is the membership function of i-th input variable for j-th term.

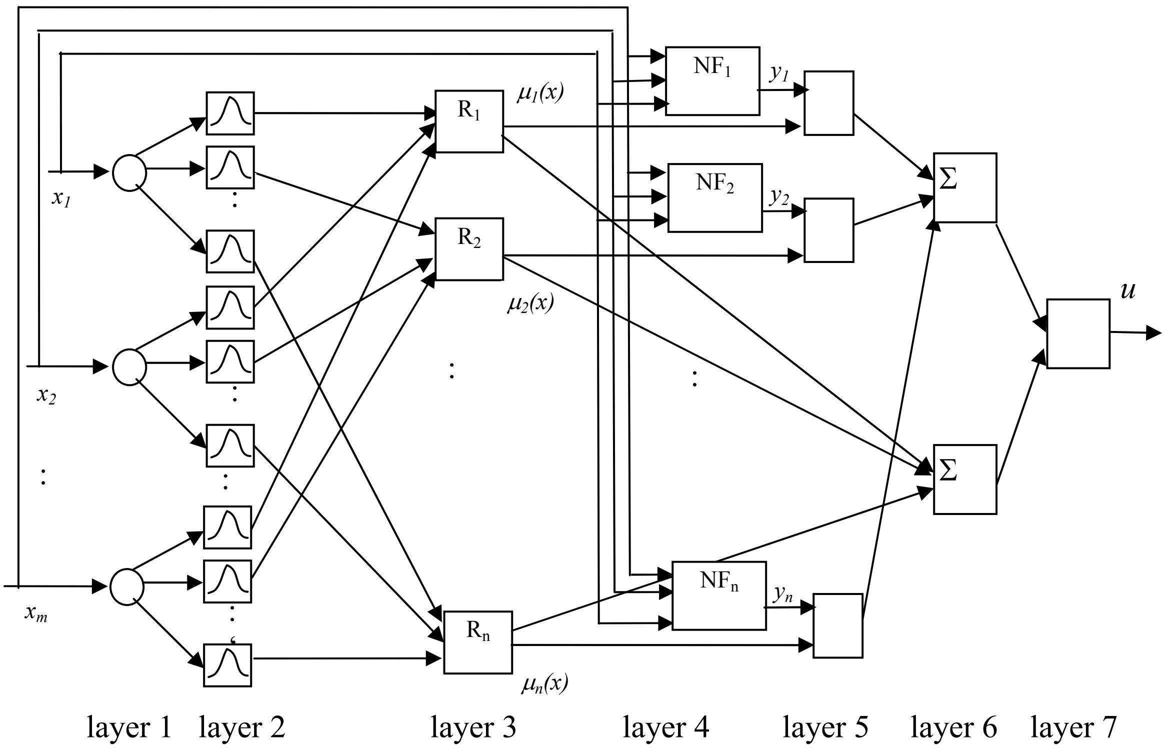


Figure 3.7 The NNFN architecture

In the third layer, the number of nodes corresponds to the number of the rules (R_1, R_2,…,R_n). Each node represents one fuzzy rule. To calculate the values of output signals, the AND (min) operation is used. In formula (3.17), Π is the min operation

$$\mu_l(x) = \prod_j \mu 1_j(x_i),\ l=1,..,n,\ j=1,..,J \tag{3.17}$$

The fourth layer is the consequent layer. It includes *n* Nonlinear Functions (NF) that are denoted by NF_1, NF_2,…,NF_n. The outputs of each nonlinear function in Fig.3.7 are calculated by using the following equation (3.15-3.17).

$$y_j = \sum_{i=1}^{m} (w1_{ij} x_i^2 + w2_{ij} x_i) + b_j, \quad j = 1,...,n \tag{3.18}$$

In the fifth layer, the output signals of third layer $\mu_l(x)$ are multiplied with the output signals of the nonlinear functions.

In the sixth and the seventh layers, defuzzification is made to calculate the output of the entire network.

$$u = \sum_{l=1}^{n} \mu_l(x) y_l \Big/ \sum_{l=1}^{n} \mu_l(x) \tag{3.19}$$

Here y_l is the outputs of fourth layer that are nonlinear quadratic functions u is the output of whole network. After calculating the output signal of the NNFN, the training of the network starts.

3.5.1.2 Learning of the Nonlinear Neuro-Fuzzy Network

Training includes the adjustment of the parameter values of membership functions c_{ij} and σ_{ij} ($i=1,..,m,\ j=1,..,n$) in the second layer (premise part) and parameter values of the nonlinear quadratic functions $w1_{ij}$, $w2_{ij}$, b_j ($i=1,..,m,\ j=1,..,n$) in the fourth layer (consequent part). At first step, on the output of network the value of error is calculated.

$$E=\frac{1}{2}\sum_{i=1}^{O}(u_i^d-u_i)^2 \qquad (3.20)$$

Here O is number of output signals of network (in the given case O=1), u_i^d and u_i are the desired and current output values of the network, respectively. The parameters $w1_{ij}$, $w2_{ij}$, b_j $(i=1,..,m,\ j=1,..,n)$ and c_{ij} and σ_{ij} $(i=1,..,m,\ j=1,..,n)$ are adjusted using the following formulas.

$$w1_{ij}(t+1)=w1_{ij}(t)+\gamma\frac{\partial E}{\partial w1_j}+\lambda(w1_{ij}(t)-w1_{ij}(t-1)) \qquad (3.21)$$

$$w2_{ij}(t+1)=w2_{ij}(t)+\gamma\frac{\partial E}{\partial w2_{ij}}+\lambda(w2_{ij}(t)-w2_{ij}(t-1)) \qquad (3.22)$$

$$b_j(t+1)=b_j(t)+\gamma\frac{\partial E}{\partial b_j}+\lambda(b_j(t)-b_j(t-1)) \qquad (3.23)$$

$$c_{ij}(t+1)=c_{ij}(t)+\gamma\frac{\partial E}{\partial c_{ij}}; \qquad (3.24)$$

$$\sigma_{ij}(t+1)=\sigma_{ij}(t)+\gamma\frac{\partial E}{\partial \sigma_{ij}}$$

Here γ is the learning rate, λ is the momentum, m is number of input signals of the network (input neurons) and n is the number of rules (hidden neurons), $i=1,..,m$, $j=1,..,n$.

The values of derivatives in (3.21-3.22) are determined by the following formulas.

$$\frac{\partial E}{\partial w1_{ij}}=(u(t)-u^d(t))\cdot\frac{\mu_l}{\sum_{l=1}^{n}\mu_l}\cdot x_i^2;$$

$$\frac{\partial E}{\partial w2_{ij}}=(u(t)-u^d(t))\cdot\frac{\mu_l}{\sum_{l=1}^{n}\mu_l}\cdot x_i^2$$

$$\frac{\partial E}{\partial b_j}=u(t)-u^d(t))\cdot\frac{\mu_l}{\sum_{l=1}^{n}\mu_l} \qquad (3.25)$$

The derivatives in (3.24) are determined by the following formulas.

$$\frac{\partial E}{\partial c_{ij}} = \sum_j \frac{\partial \mathrm{E}}{\partial u} \frac{\partial u}{\partial \mu_l} \frac{\partial \mu_l}{\partial c_{ij}} \quad , \frac{\partial E}{\partial \sigma_{ij}} = \sum_j \frac{\partial \mathrm{E}}{\partial u} \frac{\partial u}{\partial \mu_l} \frac{\partial \mu_l}{\partial \sigma_{ij}} \tag{3.26}$$

Here

$$\frac{\partial E}{\partial u} = u(t) - u^d(t), \quad \frac{\partial u}{\partial \mu_l} = \frac{y_l - u}{\sum_{l=1}^{L} \mu_l}, \quad i = 1,..,m, j = 1,..,n, l = 1,..,n \tag{3.27}$$

$$\frac{\partial \mu_l(x_j)}{\partial c_{ji}} = \begin{cases} \mu_l(x_j) \dfrac{2(x_j - c_{ji})}{\sigma_{ji}^2} & \textit{if } j \text{ node is connected to rule node } l \\ 0, \quad \text{otherwise} \end{cases} \tag{3.28}$$

$$\frac{\partial \mu_l(x_j)}{\partial \sigma_{ji}} = \begin{cases} \mu_l(x_j) \dfrac{2(x_j - c_{ji})^2}{\sigma_{ji}^3} & \textit{if } j \text{ node is connected to rule node } l \\ 0, \quad \text{otherwise} \end{cases} \tag{3.29}$$

Taking into account the formulas (3.26) and (3.29) in (3.21)-(3.24) the learning of the parameters of the NNFN is carried out.

3.6. Summary

The Neural network and Fuzzy systems are complementary rather than competitive. Fuzzy logic offer a tool to process inaccurate and approximate information, as well as mechanism for implementing rules. Fuzzy systems mainly based on knowledge of experts, or generated form sample data points. This knowledge is often formulated through fuzzy rule-based to fuzzy sets. Fuzzy sets allow partial memberships. NN has learning, generation abilities. NN provide algorithms for classification and optimization, and they work at numerical level. Neural fuzzy systems combine both features, provide more flexibility, faster, and more robust than NN alone. There are number of ways to combine NF systems. In this chapter architecture, and the operation principle of fuzzy system, NN and integration of these technologies called neuro-fuzzy system have been considered. The learning algorithm neuro-fuzzy system that uses back-propagation algorithm is given.

CHAPTER 4
DEVELOPMENT OF A NEURO-FUZZY EQUALISER FOR CHANNEL DISTORTION

4.1. Overview

In this chapter the development of a neuro-fuzzy equaliser for channel distortion is given. The structure of the neuro-fuzzy equaliser is presented. The neuro-fuzzy equaliser is applied for equalisation of linear and nonlinear channel distortion. Using MATLAB package program the computer simulation of the neuro-fuzzy system for channel distortions has been performed. The learning results of the neuro-fuzzy equalisation system are described. The BER versus SNR for different noise levels is constructed. The comparison results of the neuro-fuzzy equalisation system with the other adaptive equalisation techniques are presented.

4.2. Development of a Neuro-Fuzzy Equaliser

The structure of equalisation system is shown in Figure 4.1. The random binary input signals $s(k)$ are transmitted through the communication channel. Channel medium includes the effects of the transmitter filter, transmission medium, receiver filter and other components. Input signals can be distorted by noise, and intersymbol interference. Intersymbol interference is mainly responsible for linear distortions, while nonlinear distortions are introduced through converters, propagation environment, etc. Channel output signals are filtered and entered to the equaliser, for equalisation of distortions.

During the equaliser design, the equaliser current output signals are compared with the input signals transmitted through the channel. In case of the presence of error the learning of the neuro-fuzzy equaliser starts. Learning includes the adjusting of the parameters values of the equaliser by using formulas (3.21), (3.22), (3.23) and (3.24) [79, 82, 83]. Learning is continued until, for all input-output pairs, the value of the error would be an acceptable minimum value.

During simulation the transmitted signals $s(k)$ are known input samples with an equal probability of –1 and 1. These signals are corrupted by additive noise $n(k)$. The corrupted signals are inputs for the equaliser. In channel equalisation, the problem is the classification of the coming input signals of the equaliser onto feature space which is divided into two decision regions. A correct decision of the equaliser occurs if $\bar{s}(k) = s(k)$. Here $s(k)$ is channel input, $\bar{s}(k)$ is the decision output of the equaliser. Based on the values of the transmitted signal $s(k)$ (i.e., ±1) the channel state can be partitioned into two classes R^+ and R^-. Here $R^+=\{x(k) \mid s(k)=1\}$ and $R^- =\{x(k) \mid s(k)= -1\}$. In [60] and [61] it was shown that the output of the optimal equaliser can be mathematically characterised as

$$\bar{s}(k) = \mathrm{sgn}(f(X(k)) = \begin{cases} 1, & f(X(k)) \geq 0 \\ -1, & f(X(k)) \leq 0 \end{cases}$$

where $X(k)=\{x(k), x(k\text{-}1), x(k\text{-}2),\ldots,x(k\text{-}m)\}$, $f: R^n \rightarrow \{-1,1\}$.

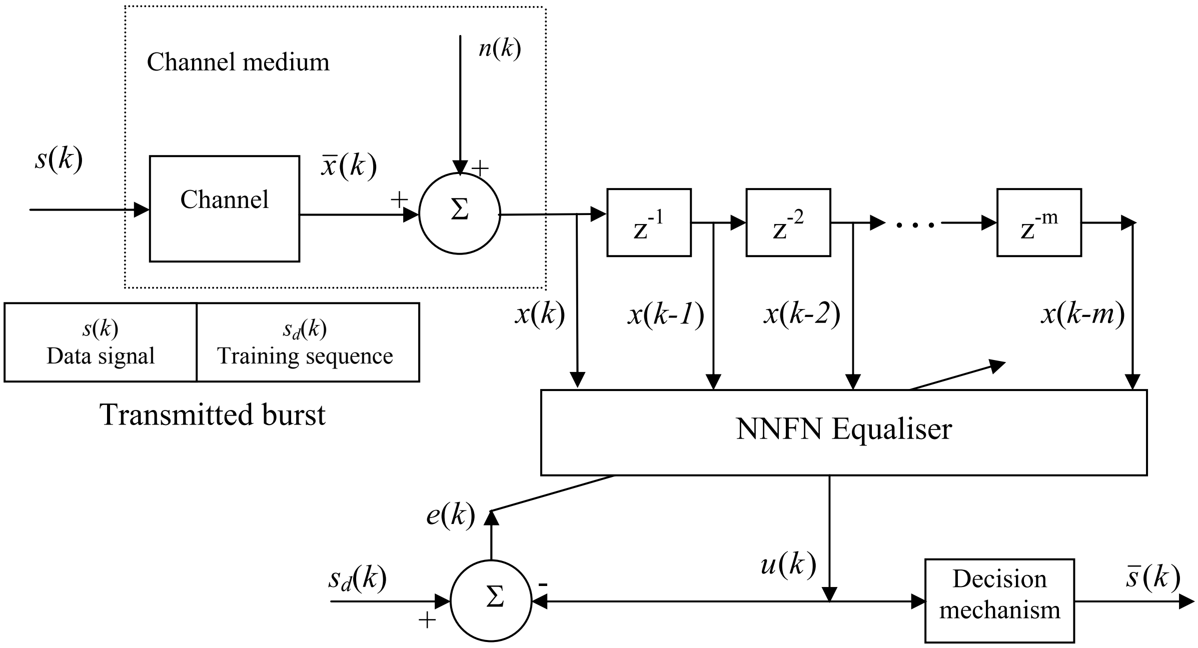


Figure 4.1 The structure of the equalisation system

The NNFN structure and its training algorithm are used to design the equaliser. Simulations have been carried out for the equalisation of linear and nonlinear channels.

Figure 4.2 illustrates the structure of the neuro-fuzzy equaliser for channel distortion.

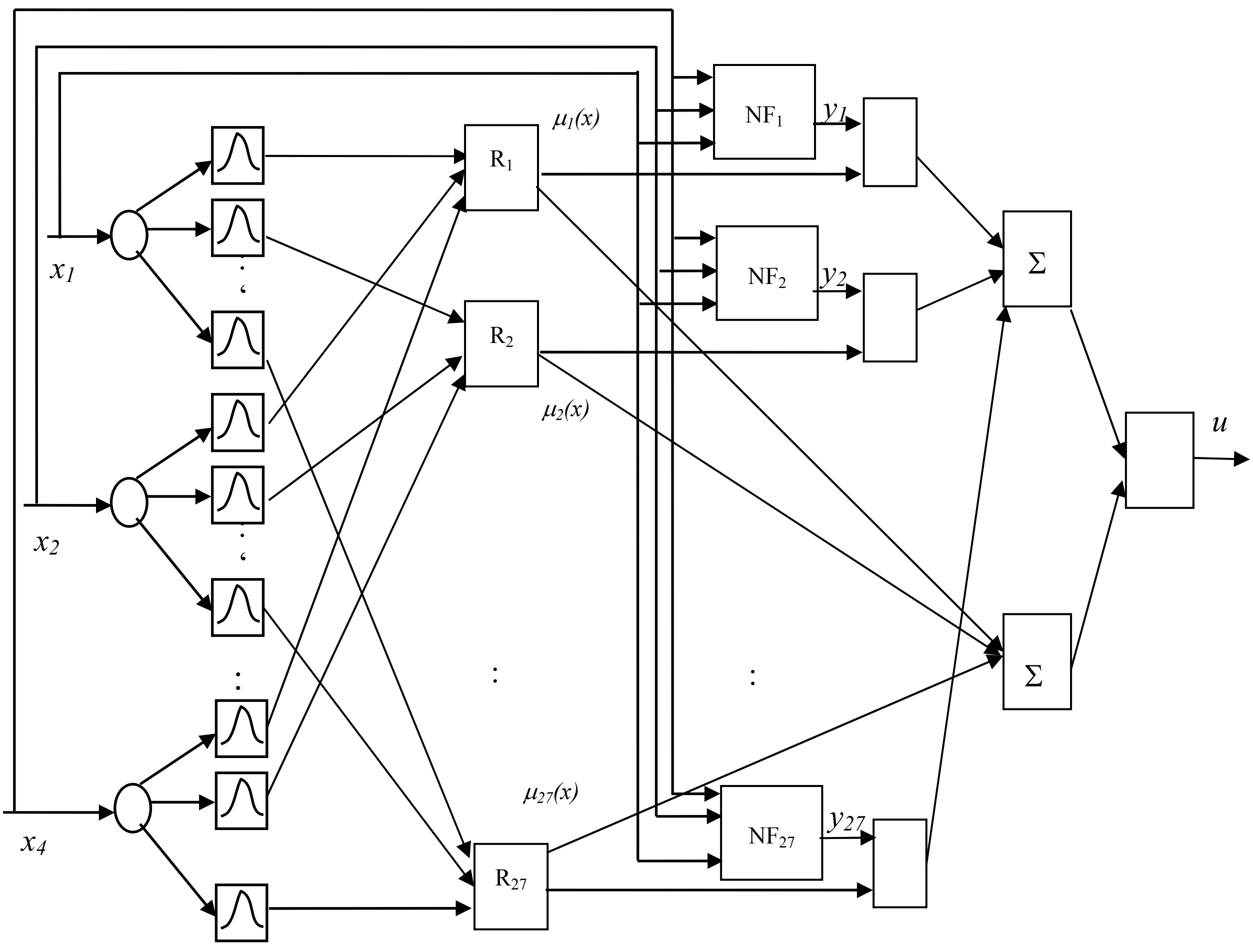


Figure 4.2 The structure of NNFN based equaliser

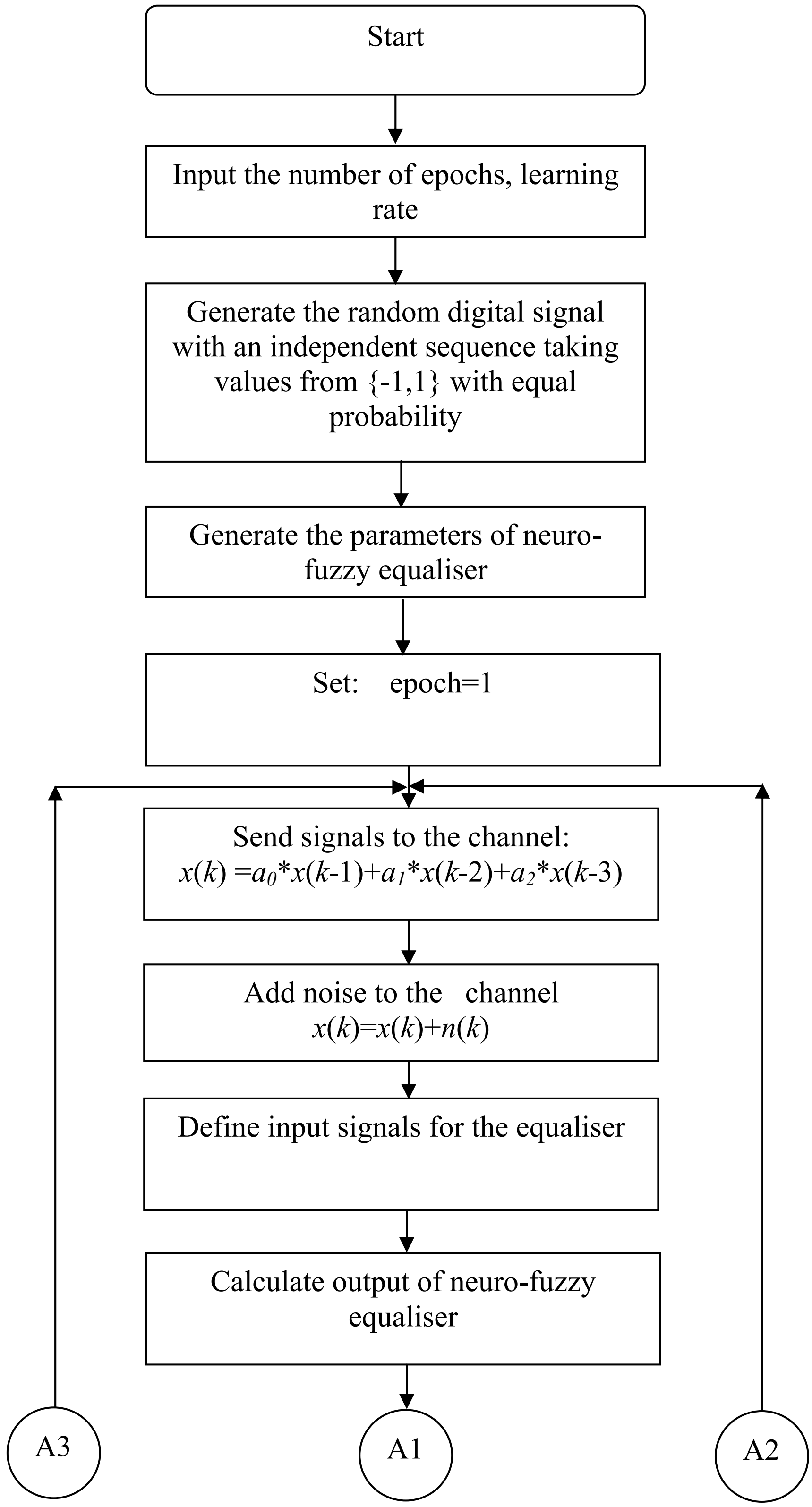
Here the network input signals $x(k)$ are the channel output signals applied to the network at time k, $x_i(k-i)$ ($i=1..4$), the number of neurons in the input layer is equal to 27, the number of hidden neurons (rules) is equal to 27, and u is the output signal of the network.

This neuro-fuzzy structure is used for equalisation of different channels. The operation algorithms of the neuro-fuzzy equaliser are given in chapter 3.

4.3 Flowchart Diagram of the Neuro-Fuzzy Equalisation System

The block diagram of the neuro-fuzzy equalisation system is given in Figure 4.3. The block scheme of realisation of the neuro-fuzzy equalisation system includes the following steps.

- Enter the number of epochs, learning rate
- Generate random digital input signal for channel with an independent sequence taking values from {-1, 1} with equal probability
- Generate the parameters of the neuro-fuzzy equaliser. Enter the number of neurons in input, hidden, and output layers
- Set epoch number to 1
- Select input signal and send to the channel
- Add additive noise to the channel and calculate the output of the channel
- Define the input signals for equaliser and send them to the equaliser
- Calculate the equaliser output.
- Calculate the error of the equaliser output
- Test the value of error. If error is less than an acceptable small value then take the next value of binary input signals and send it to the channel
- If error is more than an acceptable small value then using the learning algorithm, train the parameters of equaliser
- Take the next value of binary input signals and send it to the channel input
- Test the number of epochs. If it is more than the given number of epoch value stop the training process
- If epoch number is less than given number of epoch value, increment the current epoch value and send first binary input signal to the channel.


Start
Input the number of epochs, learning rate
Generate the random digital signal with an independent sequence taking values from {-1,1} with equal probability
Generate the parameters of neuro-fuzzy equaliser
Set: epoch=1
Send signals to the channel:
$x(k) = a_0*x(k-1)+a_1*x(k-2)+a_2*x(k-3)$
Add noise to the channel
$x(k)=x(k)+n(k)$
Define input signals for the equaliser
Calculate output of neuro-fuzzy equaliser
A3
A1
A2

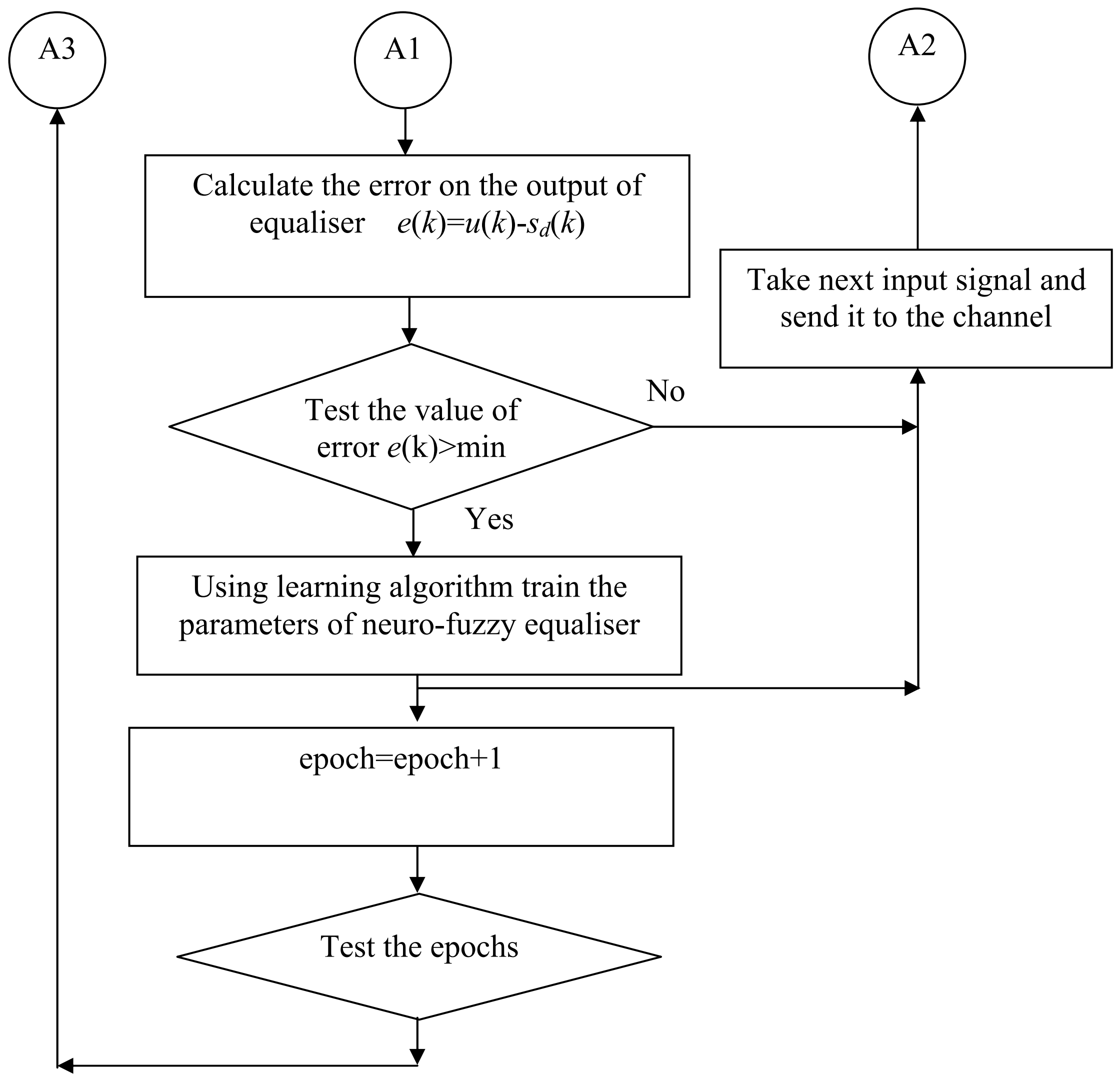


Figure 4.3 Flowchart diagram of neuro-fuzzy equalisation system

4.4 Simulation of the Neuro-Fuzzy Equalisation System

In this thesis the NNFN structure and its training algorithm are used to design the equaliser. Simulations have been carried out for the equalisation of linear and nonlinear channels.

In the first simulation, we use the following non-minimum-phase channel model [11,43, 50].

$$x(k) = a_1(k)s(k) + a_2(k)s(k\text{-}1) + a_3(k)s(k\text{-}2) + n(k) \tag{4.1}$$

where $a_1(k) = 0.3482$, $a_2(k) = 0.8704$ and $a_3(k) = 0.3482$, and $n(k)$ is the additive noise. This type of channel is encountered in real communication systems. During the equaliser design, the sequence of transmitted signals is given to the channel input. 200 symbols are used for training and 10^3 data signals for testing.

Figure 4.4 demonstrates the transmitted binary signals over the linear channel. They are assumed to be an independent sequence taking values from {-1,1} with equal probability. On the output of the channel the additive Gaussian noise $n(k)$ is added to the transmitted signal.

Figure 4.5 shows the received signal, which is the input signal for the equaliser.

Figure 4.6 demonstrates the state of noisy channel, where the noise variation is 0.5. In the output of the equalisation system, the deviation of the original transmitted signal from the current equaliser output is determined. This deviation or error $e(k)$ is used to adjust the network parameters. Training is continued until the value of the error for all training sequence of signals is acceptably low.

$$e(k) = u(k) - s_d(k)$$

where $e(k)$ is the error, $u(k)$ is the equaliser output signal and $s_d(k)$ is the desired signal.

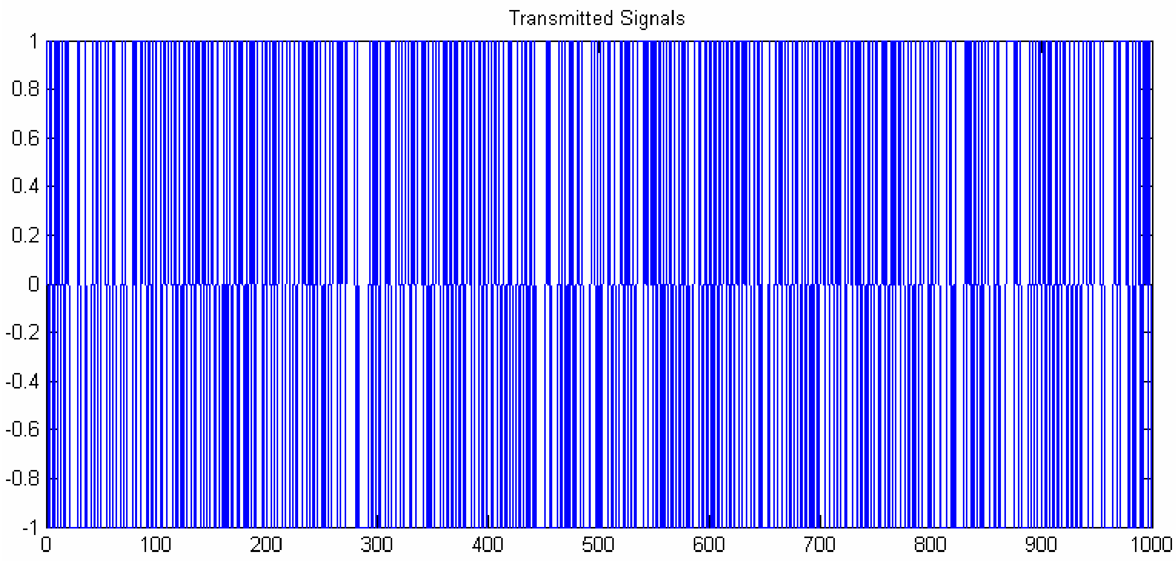


Figure 4.4 Transmitted binary signals

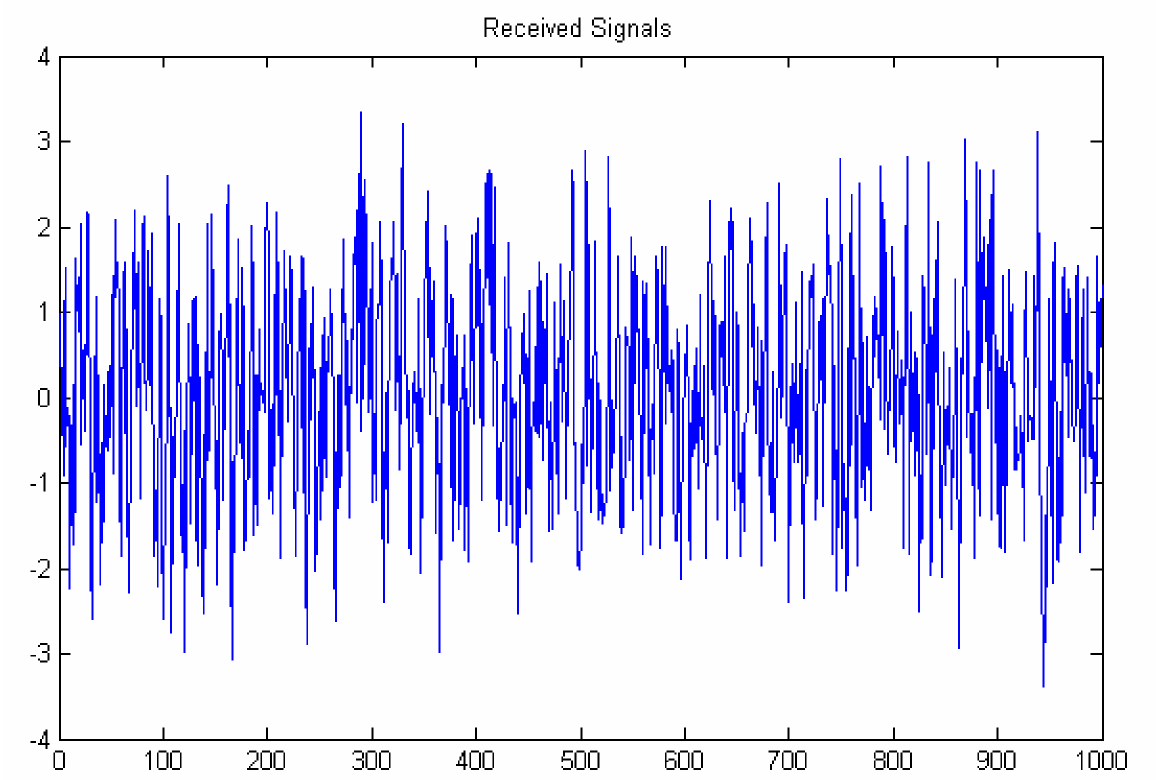


Figure 4.5 Channel output with additive noise

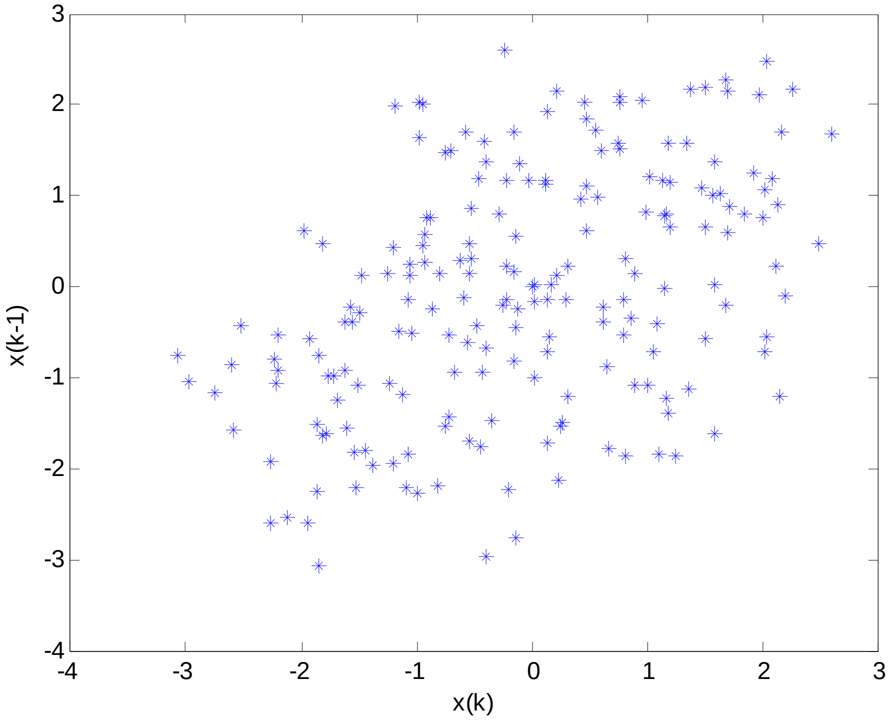


Figure 4.6 Channel state with noise

During simulation, the input signals for the equaliser are the outputs of the channel *x(k), x(k-1), x(k-2), x(k-3)*. Using NNFN, ANFIS [75], and feedforward neural networks (FFNN) the computer simulation of the equalisation system has been performed. During simulation, we used 27 rules (hidden neurons) in the NNFN, 36 hidden neurons in the FFNN and 36 rules (hidden neurons) in the ANFIS based equaliser. The learning of equalisers has been carried out for 3000 iterations. Figure 4.7 demonstrates the curve that describes the learning progress of NNFN equalisation system. During learning the parameter values of NNFN equaliser have been determined. Figure 4.8 demonstrates the state of the channel after equalisation of channel distortion by using NNFN equaliser.

Table 4.1 shows the BER performance of the non-minimum phase channel before and after equalisation. Here the results are obtained when the equaliser was trained without noise.

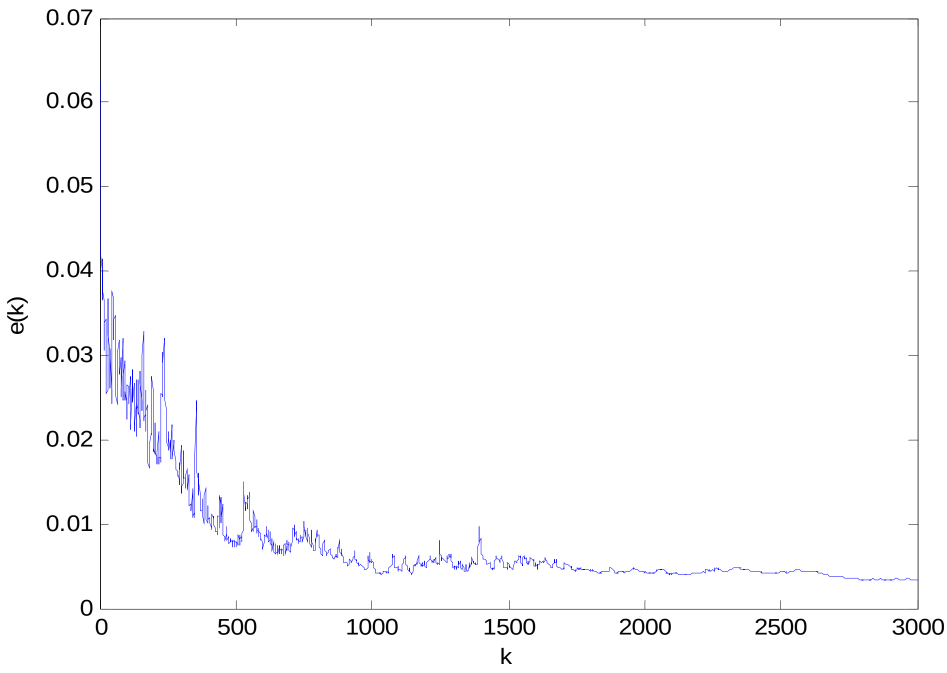


Figure 4.7 Convergence curve

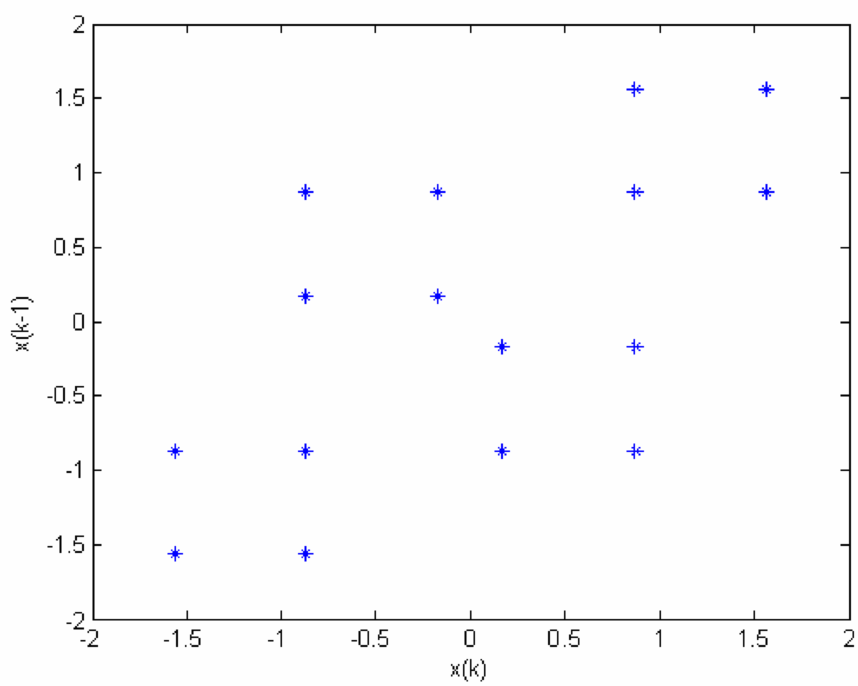


Figure 4.8 Channel state after equalisation

4.5 BER Performance

An equaliser's performance is evaluated by its probability of error. This probability predicts the BER. To calculate the BER the equaliser is tested with a statistically independent random of 10^3 channel samples. An error value, $e_i(k)$ is generated for a range of noise variances

The BER can be plotted against the channel noise to compare the equaliser performances. The BER is calculated as:

$$BER = \log_{10}\left(\sum_{i=1}^{n} \frac{e_i(k)}{n}\right) dB \tag{4.2}$$

where $e_i(k)$ is the error, and n (n=10^3) is the number of samples.

The channel noise is measured as a signal-to-noise ratio (SNR):

$$SNR = 10\log_{10}\left(\sigma_s^2 / \sigma_n^2\right) dB \tag{4.3}$$

where σ_s^2 and σ_n^2 are the signal and noise variances, respectively [11].

Table 4.1 BER performance of channel model (4.1) before and after equalisation

SNR	BER Before equalisation	BER After equalisation
20.041251	-1.045757	-2.120149
10.470649	-0.463442	-1.832105
6.967405	-0.414539	-1.658254
5.225684	-0.390406	-1.556800
4.324438	-0.387216	-1.458367
3.366142	-0.368556	-1.165436
2.292490	-0.358526	-1.143211
1.191605	-0.342944	-1.098761
0.799823	-0.340084	-0.515700

Using the same initial conditions the computer simulation of equalisation systems by using ANFIS [77], and FFNN have been performed. During simulation, 36 hidden neurons in the FFNN and 36 rules (hidden neurons) in the ANFIS based equaliser. The learning of equalisers has been carried out for 3000 iterations. After simulation the performance characteristics (BER versus SNR) for all equalisers have been determined. Bit Error Rate (BER) versus Signal-Noise Ratio (SNR) characteristics have been obtained for different noise levels. Figure 4.9 shows the performance of equalisers based on NNFN, ANFIS, FFNN and DFE. In Fig. 4.9 the solid line is the performance of the NNFN based equaliser, the dashed line is the performance of the equaliser based on ANFIS and the dash-dotted line is the performance of FFNN based equaliser, the dashed line with squares is the performance of the DFE. As shown in Figure 4.9, at the area of low SNR (high level of noise) the performance of NNFN based equaliser is better than other equalisers.

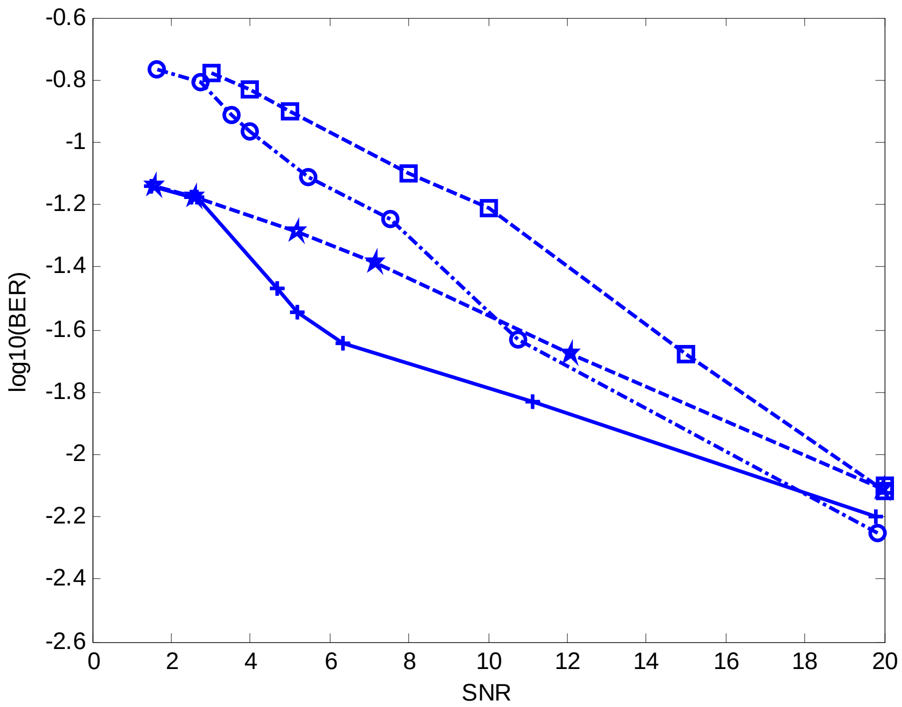


Figure 4.9 BER performance of the NNFN (solid line with ‘+’), ANFIS (dashed line with ‘o’), FFNN based equalisers (dash-doted line with ‘*’) and DFE equaliser (dashed line with squares)

4.6 Simulation of the Neuro-Fuzzy Equalisation System for Time-Varying Channel

In the second simulation, the neuro-fuzzy equalisation system for time-varying channel has been considered. During simulation, the following nonlinear channel model was used [11, 43, 50]

$$x(k) = a_1(k)s(k) + a_2(k)s(k\text{-}1) \text{ -}0.9 \cdot (a_1(k)s(k) + a_2(k)s(k\text{-}1))^3 + n(k) \tag{4.4}$$

Where $x(k)$ is the output of the channel, $s(k\text{-}1)$ is the time delay introduced by the channel, and $a_1(k)$ and $a_2(k)$ are time varying channel coefficients with initial values $a_1(0) = 1$ and $a_2(0) = 0.5$. These channel coefficients are generated by using a second-order Markov model with 3rd order nonlinearity in the presence of AWGN filtered by a second-order Butterworth low-pass filter with normalised cut-off frequency 0.1 [20, 76]. The coloured Gaussian sequences which were used as time- varying coefficients a_i are generated with a standard deviation of 0.1. The curves representing the time variation of the channel coefficients are illustrated in Figure 4.10. The first 200 symbols are used for training, 10^3 data signals are used for testing. Table 4.1 illustrates the BER comparison of the channel after training with and without noise.

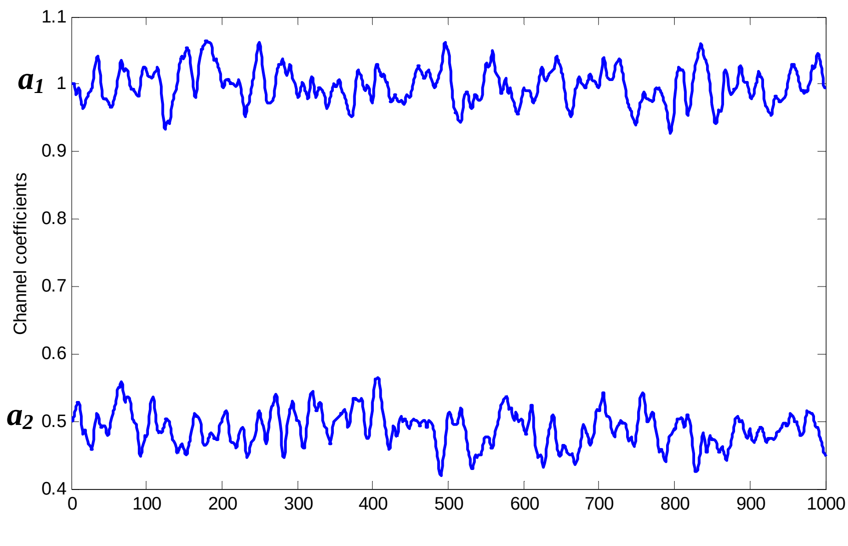


Figure 4.10 Time-varying channel coefficients

The time-varying impulse response $h(k)$ of the channel model (4.4) is given by

$$h(k) = \sum_{i=1}^{2} a_i(k)\delta(k-(i-1)) - 0.9\left[\sum_{i=1}^{2} a_i(k)\delta(k-(i-1))\right]^3 \qquad (4.5)$$

where $\delta(k)$ is the unit impulse.

The simulations are performed using NNFN, ANFIS and FFNN, 36 neurons are used in the hidden layer of each network. The transmitted signals were assumed to be an independent sequence taking values between {-1,1} with equal probability. On the output of the channel the additive Gaussian noise *n(k)* is added to the transmitted signal. Figure 4.11 shows the received signal. This signal is the input signal for the equaliser.

The channel states are plotted in Figure 4.12 and 4.13. Figure 4.12 demonstrates the state of the noisy channel, while Figure 4.13 demonstrates the state of noise free channel. Noise variation is taken as 0.5. In the output of the equalisation system, the deviation of original transmitted signal from the current equaliser output is determined. This error *e(k)* is used to adjust network parameters. In Figure 4.14 the convergence curve of the NNFN equaliser for 3000 learning iterations is given. Figure 4.15 illustrates the BER performance of the equalisers for the channel (equation 4.4), averaged over 10 independent trials. Tables 4.2 illustrate the BER performance of the NNFN equaliser of the time-varying channel before and after equalisation. Here the results are obtained when the equaliser was trained without noise.

Table 4.2 BER performance of channel model (4.4) before and after equalisation

SNR	**BER Before equalisation**	**BER After equalisation**
19.810266	-1.346787	-2.949794
11.117741	-0.515700	-2.230969
6.344230	-0.478862	-1.310223
5.176330	-0.437707	-0.967986
4.666310	-0.444906	-0.665439
2.644919	-0.414539	-0.621434
2.528533	-0.399027	-0.571083
1.520657	-0.401209	-0.550997

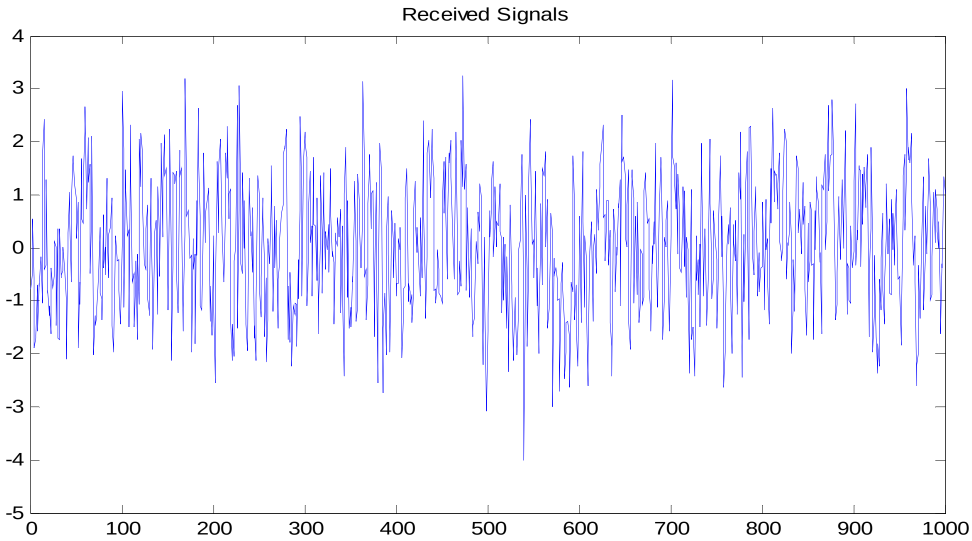


Figure 4.11 Received signal

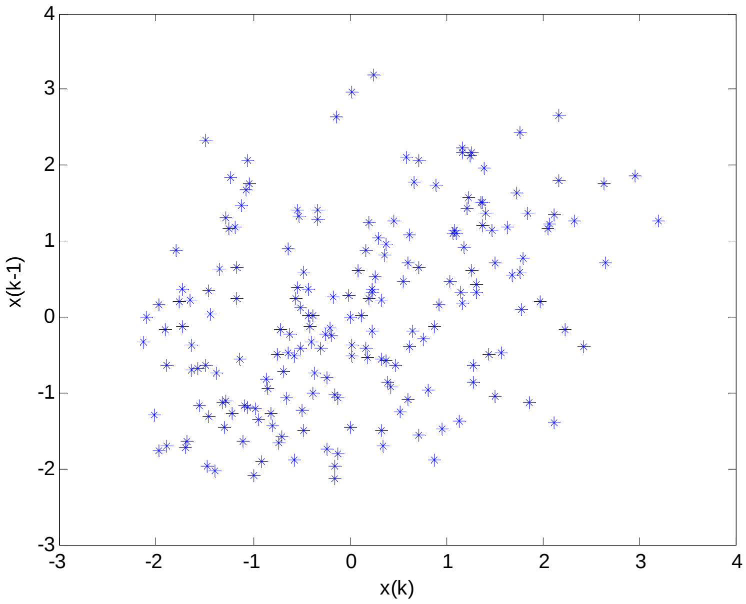


Figure 4.12 Channel state with additive noise

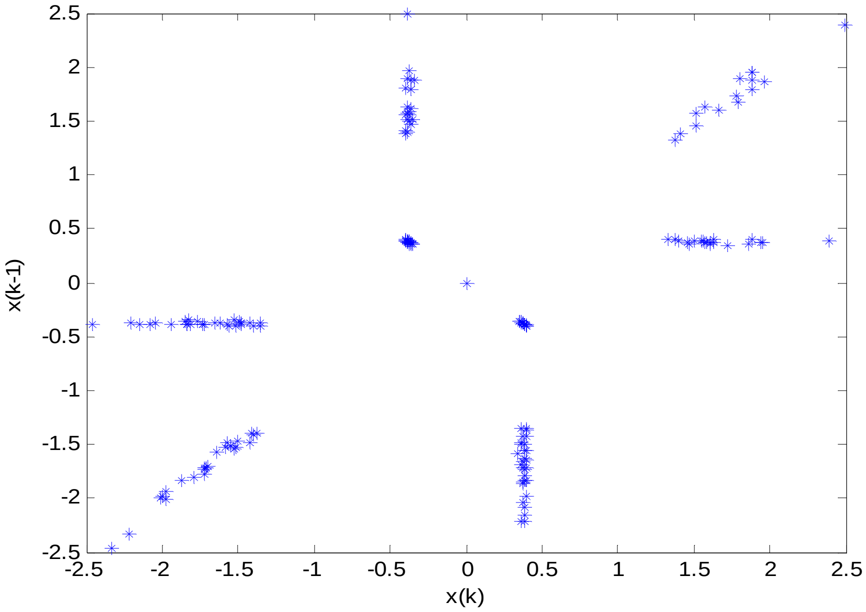


Figure 4.13 Channel state without noise

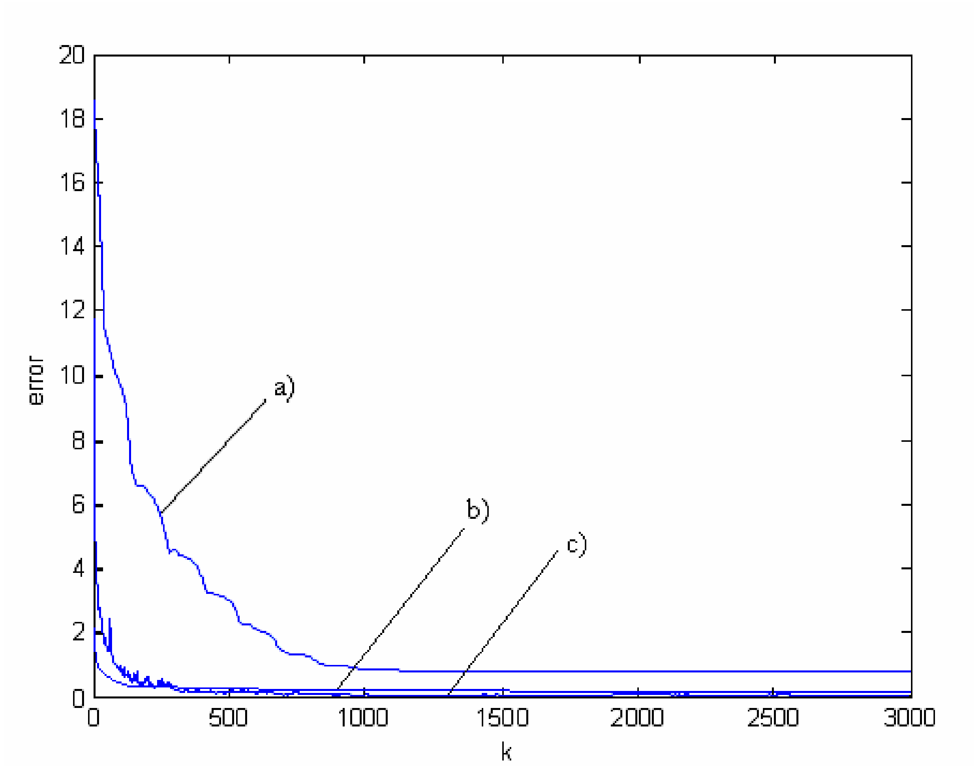


Figure 4.14 Convergence curve. a) FFNN, b) ANFIS, c) NNFN (channel model 4.4)

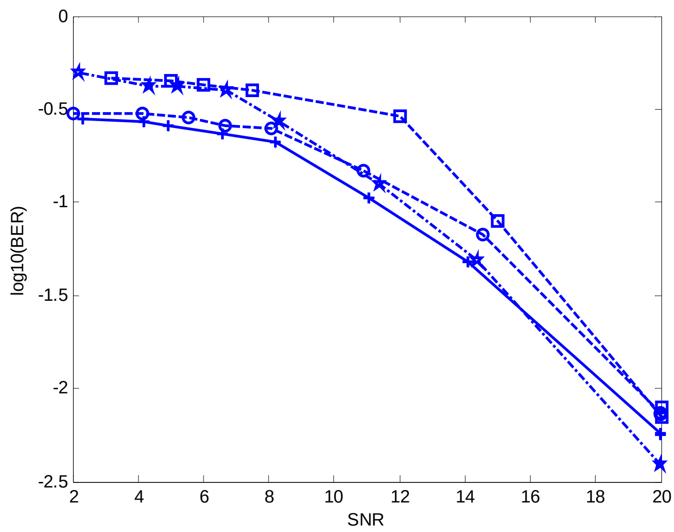


Figure 4.15 BER performance of the NNFN (solid line with '+'), ANFIS (dashed line with 'o'), FFNN based equalisers (dash-doted line with '*') and DFE equaliser (dashed line with squares). (channel model 4.4)

Figure 4.15 illustrates that the BER performance of NNFN based equaliser is better than other equalisers. The obtained result satisfies the efficiency of application of NNFN technology in channel equalisation.

In Table 4.3 the comparative analysis of the NNFN, ANFIS and FFNN based equalisers show that the NNFN based equalisation system provides better BER performance than the other two systems.

Table 4.3 Comparison of BER performance for channel model (4.4)

SNR	BER Performance		
	FFNN	**ANFIS**	**NNFN**
19.99	-2.4097940	-2.121999	-2.230969
14.085229	-1.307608	-1.161011	-1.310223
11.058583	-0.880794	-0.820406	-0.967986
8.192512	-0.550842	-0.595593	-0.665439
6.586104	-0.380790	-0.574955	-0.621434
4.944612	-0.364108	-0.537707	-0.571083
4.171610	-0.361219	-0.513716	-0.550997
2.313622	-0.298708	-0.510050	-0.549772

4.7 Conclusion

There are several techniques available to combat the multipath propagation effect. Some of these techniques are: frequency diversity and space diversity. These signal diversity techniques where used in analogue radio and have been adapted to digital systems that undergo highly selective interference. However, these techniques require a bandwidth overhead that is not available in most systems, and can provide sufficient equalisation for specific channels (minimum phase) [80].

Analysis of signal transmission systems demonstrates that, the presence of noise, intersymbol interference (ISI), and the time-varying characteristics of the channel require the use of adaptive equalisers for channel equalisation. Adaptive equalisers based on MLP networks are sometimes plagued by long training time and may be trapped at local minima, while RBF based equalisers have shortcomings related to long training time, a large number of hidden nodes, where the network structure size can increase exponentially as the problem difficulty increases [81, 82, 83, 84], in order to obtain a desired BER characteristic of equalisation system.

Integrated adaptive neuro-fuzzy system with nonlinear fuzzy inference model for equalisation of channel distortion is proposed in this thesis. The architecture of the nonlinear neuro-fuzzy system (NNFN) for the nonlinear time-varying channel was developed. The effect of multipath propagation that causes intersymbol interferences the presence of AWGN has been analyzed. The architecture and design algorithms of the NNFN were presented.

Based on gradient-descent learning algorithm the software was developed for the NNFN equalisation.

Simulations were performed using MATLAB software package for NNFN, ANFIS and FFNN.

Comparative analysis of the convergence rate demonstrated that the use of NNFN based equaliser allows decreasing training time of equaliser's parameters and decreasing the complexity of the network.

According to the simulation results, the proposed NNFN system provides higher convergence rate and improvement in the BER between -0.0334 dB to -0.0699 dB. This means that, on an average, the NNFN system provides 6-10% better BER performance in severely noisy channel conditions (SNR= 2-8 dB) compared to ANFIS and FFNN based systems.

4.8 Future work

As shown in the thesis, the high level of noise significantly causes distortion in the transmitted signal when the channel has nonlinear and time-varying characteristics. To eliminate this distortion an adaptive neuro-fuzzy system with sufficient number of fuzzy rules is needed. Type-1 fuzzy sets have limited capabilities to directly handle data uncertainties connected with the time-varying nature of the channel. These time-varying channel characteristics are interpreted as uncertainties in its coefficients. To handle these uncertainties and to improve the BER performance of the equaliser with possible less number of fuzzy rules the neuro-fuzzy system based on Type-2 fuzzy sets may be considered in the future.

References

[1] Proakis J., “Digital Communications,” New York, McGraw-Hill (1995).

[2] Qureshi S. U. H., “Adaptive equalization,” Proceedings of the IEEE, vol. 73, pp. 1349–1387, September 1985.

[3] Chen S., Hanzo L., Mulgrew B., “Decision feedback equalisation using multiple-hyperplane partitioning for detecting ISI-corrupted M-ary PAM signals,” IEEE Tans. Communications, May 2001.

[4] Zadeh L.A., “Fuzzy sets and applications: selected papers by L. A. Zadeh, John Wiley & Sons, New York, 1987.

[5] Jose-Revuelta, L. M. S., Cid-Sueiro J., “A neuro-evolutionary framework for Bayesian blind equalization in digital communications,” Signal Processing, vol. 83, pp. 325–338, 2003.

[6] Forney J. G. D., “Maximum–likelihood sequence estimation of digital sequences in the presence of intersymbol interference,” IEEE Transactions on Information Theory, vol. 18, pp. 363–378, May 1972.

[7] Forney J. G.D. , “The Viterbi algorithm,” Proceedings of the IEEE, vol. 61, pp. 268–278, March 1973.

[8] Savazzi P., Favalli L., Costamagna E., Mecocci A., “ A supoptimal approach to channel equalisation based on the nearest neighbour rule,” IEEE J. Selected Areas Commun., vol. 16, pp. 1640-1648, Dec. 1998.

[9] Kim Y., Moon J., “Delay-constrained asymptotically optimal detection using signal-space partitioning,” ICC'98 Proc., Atlanta, USA, 1998.

[10] Duda R. O., Hart P. E., “Pattern Classification and Scene Analysis,” John Wiley and Sons, 1973.

[11] Chen S., Mulgrew B., McLaughlin S., “Adaptive Bayesian equalizer with decision feedback,” IEEE Transactions on Signal Processing, vol. 41, pp. 2918–2927, September 1993.

[12] Giridhar K., Shynk J. J., Mathur A., Chari S., Gooch R. P., “Nonlinear techniques for the joint estimation of cochannel signals,” IEEE Transactions on Communications, vol. 45, pp. 473–484, April 1997.

[13] Iltis R. A., Shynk J. J., Giridhar K., “Bayesian algorithms for blind equalisation,” IEEE Transactions on Communications, vol. 42, pp. 1019–1032, Feb/Mar/Apr 1994.

[14] Steel R. (Ed), "Mobile Radio Communication," Pentec Press, London, 1992.

[15] Mulgrew B., Grant P. M., Thompson J. S., "Digital Signal Processing: Concepts and Applications," Houndmills, Basingstoke, U.K.: Macmillan, 1st ed., 1999.

[16] Haykin S., "Adaptive Filter Theory," Prentice–Hall-Information and systems science series, 3rd ed., New Jersey,1997.

[17] Falconer D. D., "Adaptive Equalization of Channel Nonlinearites in QAM Data Transmission Systems", Bell System Technical Journal, vol.27, no.7, 1978.

[18] Mulgrew B., "Nonlinear signal processing for adaptive equalisation and multi–user detection," in Proceedings of the European Signal Processing Conference, EUSIPCO, (Island of Rhodes, Greece), pp. 537–544, 8-11 September 1998.

[19] Regalia P.A. Adaptive IIR Filtering in Signal Processing and Control. Dekker, New York, 1995.

[20] Cowan C.F.N., Semnani S., "Time-variant equalization using novel nonlinear adaptive structure," Int.J.Adaptive Contr. Signal Processing, vol.12,no.2, pp.195-206, 1998.

[21] Tanner R., Cruickshank D. G. M., Hassel–Sweatman C. Z. W., Mulgrew B., "Receivers for nonlinearly separable scenarios in DS-CDMA," Electronics Letters, vol. 33, pp. 2103–2105, December 1997.

[22] Theodoridis S., Cowan C., Callender C., See C., "Schemes for equalisation of communication channels with nonlinear impairments," IEE Proc. Commun., vol. 142, pp. 165–171, June 1995.

[23] Mulgrew B., "Applying radial basis functions," IEEE Sig. Proc. Magazine, pp. 50–65, March 1996.

[24] Cid-Sueiro J., Figueiras-Vidal A. R., "Channel equalization with neural networks," Digital Signal Processing in Telecommuniocations - European Project COST#229 Technical Contributions (A. R. Figueiras-Vidal, ed.), pp. 257–312, London, U.K.: Springer– Verlag, 1996.

[25] Patra S., Mulgrew B., Grant P., "Subset centre selection with fuzzy implementated radial basis function equaliser design," CSDSP, 1998.

[26] Chen S., "Importance sampling simulation for evaluating the lower-bound BER of the Bayesian DFE," IEEE Tans. Communications, 2001.

[27] Dahnoun N., Hart M., Umbarila F., "Software optimisation techniques for real-time applied adaptive filtering using the TMS320C6201 digital signal processor,"

http://dsp.fh-mannheim.de/sip/ss01/thema4.html., University of Bristol, Texas Instruments

[28] Tanizaki, Hisashi & Mariano, Roberto S., "Nonlinear and non-Gaussian state-space modelling with Monte Carlo simulations," Journal of Econometrics, Elsevier, vol. 83(1-2), pages 263-290,1998.

[29] Tanizaki, Hisashi & Mariano, Roberto S, "Prediction, Filtering and Smoothing in Non-linear and Non-normal Cases Using Monte Carlo Integration," Journal of Applied Econometrics, John Wiley & Sons, Ltd., vol. 9(2), pages 163-79, 1994.

[30] Erdogmus D., Rende D., Jose C. Principe, Tan, F. W. "Nonlinear channel equalization using multiplayer perceptrons with information-theoretic criterion." In Proc. Of 2001 IEEE Signal Processing Society Workshop, 2001, pp.443-451.

[31] Luo F. L., Unbehauen R., "Applied neural networks for signal processing", Cambridge University Press, 1998.

[32] Hush D., Horne B., "Progress in supervised neural networks," IEEE signal processing magazine, vol.10, no.1, pp. 8-39, June 1993.

[33] Hush D., Salas J., Horne B., "Error surface for multi-layer perceptrons," IEEE Transactions on Systems, Man. and Cybernetics, vol.22, no.5, September 1992.

[34] Pengij H., Wen-Ming CA02, Pei-Liang QJ., "The Application of two weighted Neural Network for channel equalization problem", Proceedings of the IEEE 6th CAS Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication, Vol. 2, pp397-400, May 2004.

[35] Broomhead D.S., Lowe D., "Multivariable functional interpolation and adaptive networks," Complex Systems, Vol.2, pp.321-355, 1988.

[36] Luo F. L. and Unbehauen R., "A minor subspace analysis algorithm," IEEE Trans. On Neural Networks, vol. 8, pp. 1149-1155, 1998.

[37] Sarwal P., Srinath M.D., "A fuzzy logic system for channel equalization," IEEE Trans. Fuzzy System, Vol.3, pp.246-249, 1995.

[38] Pandey R., "Fast Blind Equalization Using Complex-Valued MLP," Neural Processing Letters Volume 21, Issue 3, pp. 215 – 225, June 2005.

[39] Sweeney F., Power P., Cown CFN., "Analysis of evolutionary techniques for channel equalisation," Proc. Irish Signals & Systems Cont. Dublin, pp. 545-552, June 1998.

[40] Biao L., Brian L.E., “Channel equalisation by feedforward neural network,”Proceedings of the IEEE-ISCAS’99, May 30- June 02, 1999.

[41] Kechriotis G., Zervas E., Manolakos E. S., “Using RNN for adaptive communication channel equalisation,” IEEE Trans. On Neural Networks, vol. 5, issue 2, pp. 267-278, 1994.

[42] Choi J., Bouchard M., Yeap T. H., “Decision feedback recurrent neural equalization with fast convergence rate,” IEEE Trans. On Neural Networks, vol. 16, issue 3, May 2005.

[43] Ong S., You C., Choi S., Hong D., “A Decision Feedback Recurrent Neural Equalizer as an Infinite Impulse Response Filter,” IEEE Transaction on Signal Processing, Vol. 45, No. 11, November 1997.

[44] Choi J., Lima de C. A.C., Haykin S., “Kalman Filter-Trained Recurrent Neural Equalizers for Time-Varying Channels,” IEEE Transactions on Communications, Vol. 53, No. 3, 2005.

[45] Wang L., Mendel J. M., “Fuzzy adaptive filters, with application to Nonlinear Channel Equalization,” IEEE Transaction on Fuzzy Systems, Vol.1, No.3, 1993.

[46] Sarwal P., Srinath M.D., “A fuzzy logic system for channel equalization,” IEEE Trans. Fuzzy System, Vol. 3, pp.246-249, 1995.

[47] Lee K.Y., “Complex fuzzy adaptive filters with LMS algorithm,” IEEE Transaction on Signal Processing, Vol.44, pp.424-429 1996.

[48] Patra S.K., Mulgrew B., “Efficient architecture for Bayesian equalization using fuzzy filters,” IEEE Transaction on Circuit and Systems II, vol. 45, pp.812-820, 1998.

[49] Patra S.K., Mulgrew B., “Fuzzy implementation of Bayesian equalizer in the presence of intersymbol and co-channel interference,” Proc. Inst. Elect. Eng. Commun., Vol. 145, pp.323-330, 1998.

[50] Liang Q., Mendel J. M., “Equalisation of nonlinear time-varying channels using type-2 Fuzzy adaptive filters,” IEEE Transaction on Fuzzy Systems, Vol.8, No.5, October 2000.

[51] Rappaport, T. S., “Wireless communications, principles and practice,” 2nd Ed., Prentice Hall PTR, NJ, USA, 2002.

[52] Berrou, C., Glavieux, A., “Near optimum error correcting coding and decoding: Turbo–Codes,” IEEE Transactions on Communications, vol. 44, pp. 1261–1271, October 1996.

[53] Shannon, C.E., “A mathematical theory of communication,” Bell Systems Technical Journal, vol. 27, pp. 379–423, July 1948.
[54] Shannon, C.E., “A mathematical theory of communication,” Bell Systems Technical Journal, vol. 27, pp. 623–656, October 1948.
[55] Proakis, J. G., Salehi, M., “Communication system engineering,” 2nd Ed., Prentice Hall, NJ, USA, 2002.
[56] Bateman A., “Digital communications, design for the real world,” Addison-Wesley, England, 1999.
[57] Haykin S., “Communication systems,” 4th Ed.John Wiley, Upper Saddle River, USA, 2000.
[58] Biglieri E., McLane P. J., Kam P.Y., ”Special issue on modulation, coding and signal processing for wireless communications,” IEEE Transactions on Wireless Communications, vol.12, WC-M April, 2005.
[59] Proakis J. G., “Digital Communications,” 3rd Ed., McGraw-Hill, 1995.
[60] Chen S., Gibson G.J., Cowan C.F.N., Grant P.M. “Adaptive equalization of finite non-linear channels using multiplayer perceptrons”. Signal Process,20,(2), pp. 107-119, 1990.
[61] Chen S., Gibson G.J., Cowan C.F.N., Grant P.M., “Reconstruction of binary signals using an adaptive radial-basis function equalizer,” Signal Processing, 22,(1), pp. 77-93, 1991.
[62] Chen S., Mclaughlin S., Mulgrew B., “Complex valued radial based function network,” Part II: “Application to digital communications channel equalization”, Signal Processing, 36, pp. 175-188, 1994.
[63] Haykin S., “Neural Networks: A Comprehensive Foundation,” 2nd Ed. Upper Saddle River, NJ: Prentice-Hall, 1999.
[64] Peng M., Nikias C.L., Proakis J.G., “Adaptive Equalization for PAM and QAM Signals with Neural Networks,” in Proc. Of 25th Asilomar Conf. On Signals, Systems & Computers, vol.1 pp. 496-500, 1991.
[65] Peng M., Nikias C.L., Proakis J., “Adaptive equalization with neural networks: new multiplayer perceptron structure and their evaluation,” Proc.IEEE Int. Conf.Acoust., Speech, Signal Proc., vol.II,(San Francisco,CA), pp. 301-304, 1992.

[66] Lee J.S., Beach C.D., Tepedelenlioglu N., "Channel equalization using radial basis function neural network," Proc.IEEE Int. Conf.Acoust., Speech, Signal Proc., 1996, vol.III, (Atlanta, GA), pp. 1719-1722, 1996.

[67] Zhe Chen, Antonio C. de C.Lima," A new neural equalizer for Decision-Feedback Equalization," IEEE Workshop on Machine Learning for signal processing, 2004.

[68] Siu S., Chia-Lu, Chien-Min Lee," TSK-based decision feedback equalization using an evolutionary algorithm applied to QAM Communication Systems," IEEE Transactions on Circuits and Systems, vol.52, No.9, 2005.

[69] Jang J.S.R., Sun C.-T., Mizutani E., "Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence," Prentice-Hall, NJ, 1997.

[70] Klir G.J., Yuan B., "Fuzzy sets and fuzzy logic, theory and applications," Prentice-Hall PTR, Upper Saddle River, NJ, USA, 1995.

[71] Nauck D., Kruse R., NEFCLASS, "A neuro-fuzzy approach for the classification of data. in K. M.George, Janice H. Carrol, Ed Deaton, Dave Oppenheim, and Jim Hightower, ACM Symposium on Applied Computing, Nashville, pp. 461-465. ACM Press, NewYork, 26-28 February 1995.

[72] Bose N. K., Liang P., "Neural networks fundamentals with graph, algorithms, and applications," McGraw-Hill Int. Ed., Singapore, 1996.

[73] Gupta, M. M, "Fuzzy neural network: Theory and application," Proceedings of SPIE, vol 2353, pp.300-325, 1994

[74] Jang, J.S.R.,Sun, C.-T., Mizutani. E., "Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence, Prentice-Hall, NJ, (1997).

[75] Shing, J., Jang R., "ANFIS: Adaptive-Network-Based Fuzzy Inference System," IEEE Transactions on Systems, Man and Cybernetics, Vol.23, No.3, pp. 665-683, May/June 1993.

[76] Önder M., Efe, Kaynak O., "On stabilization of Gradient-Based Training Strategies for Computationally Intelligent Systems". IEEE Transactions on Fuzzy Systems, Vol.8, No.5, pp. 564-575, 2000.

[77] Abiyev, R.H., Aliev, R.A., Aliev, R.R., "The synthesis of fuzzy control system with neural networks based tuner". News of Academy of Sciences, Tech. Cybernetics N2, pp. 192-197, 1994.

[78] Abiyev R., Mamedov F., Alshanableh T., “Neuro-fuzzy system for channel noise equalization,” International Conference on Artificial Intelligence.IC-AI’04, Las Vegas, Nevada, USA, June 21-24, 2004.

[79] Mamedov F., Abiyev R., Alshanableh T., “Recurrent Neural Network for Equalization of Channel Distortion,” Electrical, Electronics and Computer Engineering Symposium NEU-CEE2004. Nicaosia, TRNC, Turkey, pp.256-260, March 11-13, 2004.

[80] Siller C., “Multipath propagation,” IEEE communications magazine, vol.22, no.2, pp.6-15, Feb. 1984.

[81] Eng-Siong C., Yang H., Skarbek W., “Reduced complexity implementation of Bayesian equaliser using local RBF network for channel equalisation problem,” IEEE Electronics Letters, vol.32, no.1, pp.17-19, Jan. 1996.

[82] Abiyev, R., Mamedov, F., Alshanableh, T., “Equalization of channel distortions by using nonlinear neuro-fuzzy Network,” ISNN (2), Lecture Notes in Computer Science, Vol. 4492, pp. 241-250, Springer, 2007.

[83] Abiyev, R. H., Alshanableh, T., “Neuro-fuzzy network for adaptive channel equalization,” 5th Mexican International Conference on Artificial Intelligence. IEEE CS. MICAI 2006. Apizaco, Mexico, November 13-17, 2006.

[84] Abiyev, R., Mamedov, F., Alshanableh, T., “Nonlinear neuro-fuzzy network for channel equalization. Advances in Soft Computing 41, Springer-Verlag, Berlin Heidelberg, pp. 327-336, July, 2007.

List of Publications

1. Abiyev, R., Mamedov, F., Alshanableh, T., “Equalization of channel distortions by using nonlinear neuro-fuzzy Network,” ISNN (2), Lecture Notes in Computer Science, Vol. 4492, pp. 241-250, Springer, July, 2007.
2. Abiyev, R., Mamedov, F., Alshanableh, T., “Nonlinear neuro-fuzzy network for channel equalization. Advances in Soft Computing 41, Springer-Verlag, Berlin Heidelberg, pp. 327-336, July, 2007.
3. Abiyev, R. H., Alshanableh, T., “Neuro-fuzzy network for adaptive channel equalization,” 5th Mexican International Conference on Artificial Intelligence. IEEE CS. MICAI 2006. Apizaco, Mexico November 13-17, 2006
4. Abiyev, R., Mamedov, F., Alshanableh, T., “Neuro-fuzzy system for channel noise equalization,” International Multi-Conference in Computer Science & Computer Engineering. International Conference on Artificial Intelligence.IC-AI’04, Las Vegas, Nevada, USA, June 21-24, 2004.
5. Abiyev, R., Mamedov, F., Alshanableh, T., “Adaptive equalization of channel distortion using recurrent neuro-fuzzy network,” International Journal of Computational Intelligence. ICCI2004. Nicaosia, TRNC, Turkey, pp.154-157, May 27-29, 2004.
6. Mamedov, F., Abiyev, R., Alshanableh, T., “Recurrent neural network for equalization of channel distortion,” Electrical, Electronics and Computer Engineering Symposium NEU-CEE2004. Nicaosia, TRNC, Turkey, pp.256-260, March 11-13, 2004.
7. Mamedov, F., Abiyev, R., Alshanableh, T., “Petrol ve gaz boru hatlarınde sinir ağı kullanarak yenim denetimi yapan sinyal işletim sistemi,” IEEE Sinyal İşleme ve Uygulamları Kurultayı, SIU’2001, Doğu Akdeniz Üniversitesi, Gazimağusa, KKTC, pp.468-471, 25-27 Nisan 2001.
8. Mamedov, F., Abiyev, R., Alshanableh, T., “Signal processing system using radial-basis neural network for inspection of pipelines corrosion,” Pakistan Journal of Applied Science. Volume 1, N:2, pp.117-118, April, 2001.

Appendix

MATLAB Files

```
function [y,signal_var]=...
    chanel_model(x,r1,fk)
d=0;  chan_mod=1;
for k=1:fk
%   linear channels
     if chan_mod==1
         if k==1
             yc(k)=0.3482*x(k-d);
         elseif k==2
             yc(k)=0.3482*x(k-d)+0.8704*x(k-1-d);
         end
         if k>2
             yc(k)=0.3482*x(k-d)+0.8704*x(k-1-d)+0.3482*x(k-2-d);
         end;
     end
     if chan_mod==2
         if k==1
             yc(k)=0;
         end
         if k>1
             yc(k)=0.5*x(k)+x(k-1);
         end
     end
     if chan_mod==3
         % nonlinear channels
         yc(k)=0.3482*x(k-d)+0.8704*x(k-1-d)+0.3482*x(k-2-d)-
0.7*(0.3482*x(k-d)+0.8704*x(k-1-d)+0.3482*x(k-2-d))^3;

     end
    y(k)=yc(k)+r1(k);
end
mean_yc=sum(yc)/fk;
signal_var=sum((yc-mean_yc).^2)/fk;
```

```
function [c,o,w1,w2,b,xin,y,r1]=...
chanel_nl(N1,N2,M,c,o,w1,w2,b)
N1=4; N2=27; M=1;
fname=['fchanel' int2str(N2) '.dat']
maxl=2;
global xin y r1 fk
n=1000;     %test  all signals
fk=200;     % training signals
clc
pause
disp('0- Parameters Init  ')
disp('1- Learning         ')
disp('2- Reading          ')
disp('3- Saving           ')
disp('4- Without learning ')
disp('5- Noise ')
disp('6- Exit')
num=input('Enter number:');

switch num
     case 0, regim=0;
           xin=2*round(rand(n,1))-1
           replay=input('Generate new random noise  y/n :','s');
            noise_var=0.5;
             if replay=='y'
                 [r1,sigma]=noise(n,noise_var);
             else
                 r1(1:n)=0;
             end

%              aa1=0.3482;  aa2=0.8704;  aa3=0.3482;
%              [y,signal_var]=chanel_model(xin,r1,n);
%       yc=y'-r1;

%      [a1,a2,a3]=chanel_coef(aa1,aa2,aa3,n); pause
       aa1=1; aa2=0.5; aa3=0;fpt = fopen('fchan_coefl000.dat','r');
[a1]=fscanf(fpt,'%f \n',[n]);
[a2]=fscanf(fpt,'%f\n',[n]);[a3]=fscanf(fpt,'%f \n',[n]);
       a3=0;  fclose(fpt);
[y,signal_var]=chanel_model_timevar(xin,r1,n,a1,a2,a3);

              SNR = 10*log10(signal_var/noise_var);
              sprintf('signal_var=%f  noise_var=%f  SNR=%f',
signal_var,noise_var,SNR)
             plot(xin)
             title('Transmitted Signals');
             pause
             plot(y)
             title('Received Signals');
             pause
             norm=minmax(y)
             delta=(abs(norm(1))+abs(norm(2)))/(N2-1);
             c(1:N1,1:N2)=0;  N22=ceil(N2/2);
                ii=0; kc(1:N22)=0;
                for j=1:N22
                    kc(j)=delta*(j-1)/j;
                    c(1:N1,j)=0-ii*kc(j);
                    o(1:N1,j)=0.1+0.01*j; %kc(j)
                    if(j>1)
```

```
                        c(1:N1,N22+j-1)=ii*kc(j);
                        o(1:N1,N22+j-1)=0.1+0.01*j;
                    end
                    ii=ii+1;
                end
                if(mod(N2,2)==0)
                    c(1:N1,N2)=c(1:N1,N2-1)+delta;
                    o(1:N1,N2)=0.2;
                end

            o(1:N1,N2)=0.2; o(1:N1,N22)=0.2;
            w1=0.02*rand(N2,N1)
            w2=0.02*rand(N2,N1)
            b=0.02*rand(N2,1);
            pause
            [c,o,w1,w2,b,xin,y,r1]=chanel_nl(N1,N2,M,c,o,w1,w2,b);
    case 1, regim=1;

[c,o,w1,w2,b]=chaneld_nl(N1,N2,M,c,o,w1,w2,b,n,fk,xin,y,r1,regim)
            pause
             [c,o,w1,w2,b,xin,y,r1]=chanel_nl(N1,N2,M,c,o,w1,w2,b)
    case 2, fpt = fopen(fname,'r');
          [c]=fscanf(fpt,'%f \n',[N1,N2]);
          [o]=fscanf(fpt,'%f \n',[N1,N2]);
          [w1]=fscanf(fpt,'%f \n',[N2,N1]);
          [w2]=fscanf(fpt,'%f \n',[N2,N1]);
          [b]=fscanf(fpt,'%f \n',[N2]);
          pause
        [c,o,w1,w2,b,xin,y,r1]=chanel_nl(N1, N2,M, c,o,w1,w2,b);
    case 3, fpt = fopen(fname,'w');
        fprintf(fpt,'%f %f %f \n',c,o,w1,w2,b)
        fclose(fpt);
        pause
        [c,o,w1,w2,b,xin,y,r1]=chanel_nl(N1, N2,M, c,o,w1,w2,b);

    case 4, regim=4;
            chaneld_nl(N1, N2,M,c,o,w1,w2,b,n,fk,xin,y,r1,regim)
             [c,o,w1,w2,b,xin,y,r1]=chanel_nl(N1,N2,M,c,o,w1,w2,b)
    case 5,
    case 6,
    otherwise disp('unknown')
end
```

```
function
[c,o,w1,w2,b]=chaneld_dfenl(N1,N2,M,c,o,w1,w2,b,n,fk,xin,y,r1,regim)
global xin r1 fk
nc=0.00;
epoch=3000;
ku=1;
if regim==1
    a=input('Enter new learning rate:');
end
for t=N1:fk
    Data(t,:)=[y(t) y(t-1) y(t-2) y(t-3) xin(t)]; % y(t-4)
    Data1(t,:)=[y(t) y(t-1) y(t-2) y(t-3) ]; % y(t-4)
    if(t>N1)
        z1(tN1)=y(t);     z2(tN1)=y(t-1);
    end
end
Data
ind1 = find(xin(:,1) > 0);
ind2 = find(xin(:,1) < 0);
for t=2:fk
    zz1(t)=y(t);     zz2(t)=y(t-1);
end
%plot(zz1(ind1), zz2(ind1), 'x', zz1(ind2), zz2(ind2), 'o');
plot(z2,z1,'*');
xlabel('x(k)'); ylabel('x(k-1)');
pause

if regim==1

[c,o,w1,w2,b,a]=nefuz_train2_nl(regim,N1,N2,M,Data,fk,epoch,a,c,o,w1,w
2,b);
end

%Test
[row,col]=size(Data);
for t=N1:n
    Data2(t,:)=[y(t) y(t-1) y(t-2) y(t-3) xin(t)];
    for i=1:col-1
            x(i)=Data2(t,i);
        end
    out(t)=Data2(t,col);
    [ys,summin,minm,m,ym,net]=nefuz_nl(N1,N2,M,x,c,o,w1,w2,b);
    tyst(t)=ys;
    if(regim==4)||(regim==1)
        if(ys>0)
            tys(t)=1;
            tys1(t)=1;
        else
            tys(t)=-1;
            tys1(t)=0;
        end
    end
    if(out(t)>0)
        out1(t)=1;
    else
        out1(t)=0;
    end
end;
plot(out)
hold
```

```
plot(tyst,'--r')
pause
sprintf('TEST before decision device')
ert=out-tyst;  yc=y'-r1;
[SNR,BER]=snr_ber(yc,r1,ert,n);
r2(1:n)=0;
[ytest]=chanel_model(tyst,r2,n);
for t=N1:n
   if(t>N1)
       zz1(tN1)=ytest(t);     zz2(tN1)=ytest(t-1);
   end
end
plot(zz1,zz2,'*')
xlabel('x(k)'); ylabel('x(k-1)');
pause

sprintf('TEST after decision dev. with (-1,1) and (0,1)')
plot(out)
hold
plot(tys,'--r')
er=out-tys;
[SNR,BER]=snr_ber(yc,r1,er,n);
er1=out1-tys1;
[SNR,BER]=snr_ber(yc,r1,er1,n);
if(regim==4)||(regim==1)
     [num,berr]=biterr(out1,tys1);
 end
 sprintf('num=%f   BERnl()=%f    SNRnl()=%f',num,berr,SNR)
pause

plot(er)
pause

r2(1:n)=0;
[ytest]=chanel_model(tys,r2,n);
for t=N1:n
   if(t>N1)
       zz1(tN1)=ytest(t);     zz2(tN1)=ytest(t-1);
   end
end
plot(zz1,zz2,'*')
xlabel('x(k)'); ylabel('x(k-1)');
pause
```

```
function  [ys,summin,minm,m,ym,net]=nefuz_nl(N1,N2,M,x,c,o,w1,w2,b);
m(1:N1,1:N2)=0; minm(1:N2)=0;
for i=1:N1
    for j=1:N2
        m(i,j)=exp(-(((x(i)-c(i,j))/o(i,j))^2));
    end
end
summin=0;
for j=1:N2
    minm(j)=10;
    for i=1:N1
        if m(i,j)<=0.02
            m(i,j)=0;
        end
        if m(i,j)~=0
            if minm(j)>m(i,j)
                minm(j)=m(i,j);
            end
        end
    end
    if minm(j)>=10
        minm(j)=0;
    end
end
summin=sum(minm);
% j is number of nl members   % ii is number of rules
for j=1:N2
    net(j)=0;
    for i=1:N1
        net(j)=net(j)+x(i)*x(i)*w1(j,i)+x(i)*w2(j,i);
    end
    net(j)=net(j)+b(j);
end
for i=1:M
    ym(i)=0;
    for j=1:N2
        ym(i)=ym(i)+net(j)*minm(j);
    end
    if summin==0
        ys(i)=0;
    else
        ys(i)=ym(i)/summin;
    end
end
```

```
function
[cfind,ofind,wfind1,wfind2,bfind,a]=nefuz_train2_dfenl(regim,N1,N2,M,D
ata,fk,epoch,a,c,o,w1,w2,b)
nc=0.00;  pi=3.14;
ku=1;
w1o=w1; w2o=w2; bo=b;
[row,col]=size(Data);
epoc1=1;  t_err(epoc1)=0;
time_begin = cputime;
while epoc1<=epoch

    t_er(epoc1)=0;
    for t=N1:fk
        for i=1:col-1
            x(i)=Data(t,i);
        end
        out(t)=Data(t,col);
        [ys,summin,minm,m,ym,net]=nefuz_nl(N1,N2,M,x,c,o,w1,w2,b);
        er=out(t)-ys;
        if((summin~=0)&&(abs(er)>0.0001))

[c,o,w1,w2,b,w1o,w2o,bo]=trainf_nl(N1,N2,M,x,er,minm,summin,m,ys,a,c,o
,w1,w2,b,ym,net,w1o,w2o,bo);
        end
        t_er(epoc1)=t_er(epoc1)+(er*er);
    end
    ser=t_er(epoc1);
    if epoc1==1
        ser0=ser;
    end
    decay=(ser0-ser)/ser0;
    if(decay<=0)
        decay0=-1; decay1=-1;
    end
     if((mod(epoc1,5)==0))
         if(decay0>0)
             a=a*1.01;
         end
         if(decay0<0)
             a=a/1.011;
         end
         decay1=decay0;
         decay0=1;
     end
    ser0=ser;
   sprintf('%i   ser=%f  decay=%f   a=%f  summin=%f',epoc1,
ser,decay,a,summin)
    if epoc1==1
        serfind=ser;
    end
    if ser<serfind
        serfind=ser;   cfind=c;  ofind=o; wfind1=w1; wfind2=w2;
bfind=b;
    end
    epoc1=epoc1+1;
end
time_end = cputime-time_begin
t_er=t_er/sqrt(200);
plot(t_er);
xlabel('k'); ylabel('error');
```

```
pause

fp=fopen('frmse.dat','w');
fprintf(fp,'%f \n',t_er);
fclose(fp);
```

```
function [y,signal_var]=...
    chanel_model_timevar(x,r1,fk,a1,a2,a3)
d=0; chan_mod=2;
if chan_mod==1
    for k=1:fk
        %   linear channels
        if k==1
            yc(k)=a1(k)*x(k-d);
        elseif k==2
            yc(k)=a1(k)*x(k-d)+a2(k)*x(k-1-d);
        end
        if k>2
            yc(k)=a1(k)*x(k-d)+a2(k)*x(k-1-d)+a3(k)*x(k-2-d);
        end;
%      y(k)=0.5*x(k)+x(k-1);
       y(k)=yc(k)+r1(k);
   end
end
if chan_mod==2
    for k=2:fk
        % nonlinear Time varrying channel channels
        yc(k)=a1(k)*x(k-d)+a2(k)*x(k-1-d)-0.9*(a1(k)*x(k-d)+a2(k)*x(k-
1-d))^3;
        y(k)=yc(k)+r1(k);
    end
end
mean_yc=sum(yc)/fk;
signal_var=sum((yc-mean_yc).^2)/fk;
```

```
function
[c,o,w1,w2,b,w1o,w2o,bo]=trainf_nl(N1,N2,M,x,er,minm,summin,m,yy,a,c,o
,w1,w2,b,ym,net,w1o,w2o,bo);
a2=0.0;   %a2=0.2500;
w2o1=w2;
w1o1=w1;
bo1=b;
for j=1:N2
    for i=1:N1
        w1(j,i)=w1(j,i)+a*er*x(i)*x(i)*minm(j)/summin+a2*(w1(j,i)-
w1o(j,i));
        w2(j,i)=w2(j,i)+a*er*x(i)*minm(j)/summin+a2*(w2(j,i)-
w2o(j,i));
    end
end

for j=1:N2
    b(j)=b(j)+a*er*minm(j)/summin+a2*(b(j)-bo(j));
end
sec=1;
if sec==1
 for j=1:N2
    kw(j)=(net(j)-yy)/summin;    %using defuzification
 end

 for i=1:N1
     for j=1:N2
       c(i,j)=c(i,j)+a*er*kw(j)*m(i,j)*2*(x(i)-c(i,j))/(o(i,j)^2);
 %       o(i,j)=o(i,j)+a*er*kw(j)*m(i,j)*2*((x(i)-
c(i,j))^2)/(o(i,j)^3);
 end
 end
end

 w2o=w2o1;
 w1o=w1o1;
 bo=bo1;
```

Neural Network Parameters	
Number of input neurons	4
Number of hidden neurons	27
Number of output neurons	1
Learning rate	0.025
Momentum	0.2
Error	0.0001
Epochs	3000