Wavelet Based Channel Equalisation

Group Members:
Mohit Garg mohitgarg@iitb.ac.in 00D07015
Karthik Ramkumar karthikr@ee.iitb.ac.in 00007022

Abstract

Channel Equalisation is carried out to mitigate the effect of non-ideal channel behaviour in digital communication systems. The first equalisers were linear filters which slowly led to more complex but better performance non-linear methods. Wavelets have also been successfully applied to this domain. This document intends to give an overview of the use of wavelets to equalisation of non-ideal channels.

Index Terms

Wavelets, Channel Equalisation, Neural Networks, Multi-Resolution Analysis

I. Introduction

The impulse response of an ideal channel is the idealised *dirac delta* function *i.e.* it does not cause any distortion of the input signal. Practical channels being far from ideal, lead to distortion (Inter-Symbol Interference (ISI)) and require special techniques to prevent the performance of the communication system from degrading. Channel Equalisation is one such extensively used technique.

The aim of equalisation is to 'undo' the effect of the channel's non-ideal behaviour. The ideal channel equaliser is one which is the exact *inverse* of the impulse response of the channel. Since in practice, the channel response is not known beforehand, one has to take recourse to 'approximate' methods of channel equalisation. Most equalisers periodically update their parameters based on the channel conditions through the use of 'training sequences' sent by the transmitter (Adaptive Equalisation). This helps in estimating the current channel conditions.

Linear Equalisers

The earliest equalisers were linear in design. These are implemented using Linear Transversal Filters[1] which consists of a shift register bank and appropriate tap coefficients which are preset or adaptively updated using periodic training sequences. Linear Transversal Equalisers (LTEs) are simple to implement and have been effectively used in channels where the ISI is not severe (*e.g.* wired telephone channels).

Non Linear Equalisers

Many practical channels being non-linear, the deficiencies of LTEs were soon realised. This led to a lot of work on non-linear equalisation techniques. Non-linear Min. Mean Square Error (MMSE) estimators have been tried, Wiener Filtering approaches have also been looked at. The adaptive and machine learning nature of the equalisation operation has also led to many neural network based approaches like Multi-Layer Perceptions (MLP) and Radial Basis Functions (RBF) etc.

Wavelets have been used in both Linear and Non-Linear equalisation. Multi-Resolution Perfect Reconstruction Filter Banks have been investigated and analysed (*Linear*). Many researchers have also proposed Wavelet based Neural Networks (*Non-Linear*).

The rest of this report is organised as follows: Section II recapitulates some preliminary concepts in Multi-Resolution Analysis and Neural Networks, and in Section III, we discuss the broad categories of wavelet based equalisation techniques.

[1] We will be working in the discrete domain since any signal (input or intermediate) in a digital communication system can be represented by a discrete sequence.

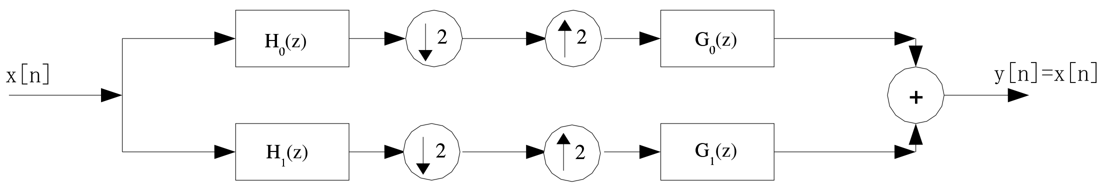

Fig. 1. One stage of a *dyadic*-Multiresolution Analysis Expansion

II. Background Theory

A. Channel Equalisation Basics

If a transmitted sequence $s[n]$ is passed through a dispersive channel, and the channel output $g[n]$ is corrupted by Additive White Gaussian Noise (AWGN) $\eta[n]$ (independent of $s[n]$. Thus, the received signal $x[n]$ can be written as (assuming the channel to be a Linear Shift Invariant (LSI) system):

$$x[n] = g[n] + \eta[n] = s[n] * h[n] + \eta[n] \tag{1}$$

where, $h[.]$ denotes the channel impulse response. Representing the channel by a FIR filter (finite channel memory), we can write

$$x[n] = \sum_{i=0}^{M} h[i]s[n-i] + \eta[n] \tag{2}$$

The output of the equaliser is the estimate of the transmitted data ($\hat{s}[n-d]$), using the observed samples ($x[n], x[n-1], ..., x[n-N+1]$) of the channel output. Here, d is called the *equaliser decision delay* and N is called the *order* of the equaliser. Both linear and non-linear estimates may be used thereby giving a host of equalisation algorithms.

B. Multi-Resolution Analysis

A m-band Perfect Reconstruction Filter Bank (PRFB) is a collection of filters for *analysing* (breaking up) an incoming signal into different parts through a collection of analysis filters, and then *synthesising* it back through a set of synthesis filters. Needless to say, there is a direct relation between the analysis and synthesis filters. Repeating such m-band PRFB's on each branch of one system leads us to the idea of *Multi-Resolution Analysis (MRA)*.

The simplest and the most studied PRFB structure is the 2-band PRFB (*dyadic* MRA). One stage (building block) of the dyadic-MRA expansion is shown in Fig. 1. For the output to be an exact replica of the input signal (except for scaling and delay – to take care of *non-causality*), the transfer functions of the filters must satisfy the following constraints:

$$\begin{aligned} H_0(z)G_0(z) + H_1(z)G_1(z) &= C_0 z^{-D} \\ H_0(-z)G_0(z) + H_1(-z)G_1(z) &= 0 \end{aligned}$$

Hence, we are left with two degrees of freedom to choose the filters. Hence, we get different MRA expansions. An important property of ideal-PRFB filters is the *power complementarity* property *i.e.*

$$|H_0(e^{j\omega})|^2 + |H_0(e^{j(\omega+\pi)})|^2 = constant \tag{3}$$

C. Neural Networks

Neural Networks are generic structures modeled on the neurons in the brain (hence the name 'neural'). The brain performs its activities using densely interconnected structures of large number of neurons. All learning is also believed to be carried out through interconnections of these neurons.

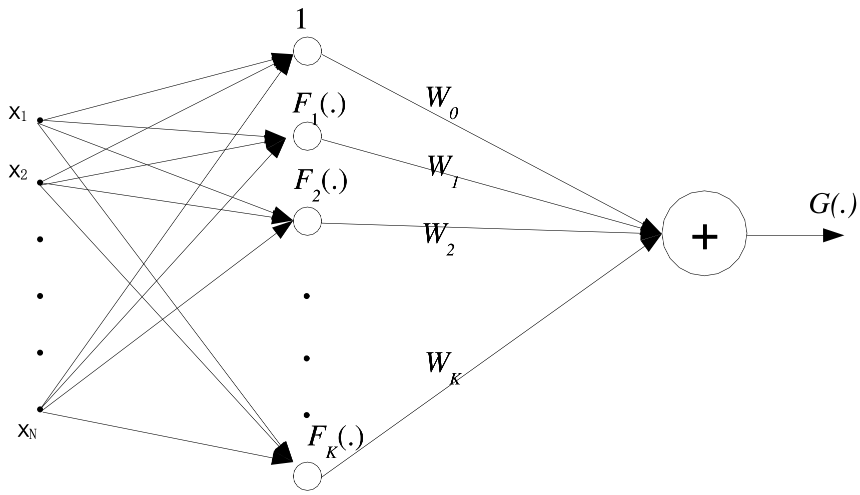

Fig. 2. A single-hidden layer Neural Network Structure. In a Wavelet Neural Network, the functions $F_j(.)$ are dilates and translates of a mother wavelet.

Artificial Neural Networks (ANNs) are systems consisting of a large number of elements densely interconnected with each connection having a specified weight. These systems can be 'trained' to adapt themselves (change their parameters) w.r.t. the desired input-output characteristics. Neural Networks find a lot of use in Artificial Intelligence applications and situations where not much is known about the underlying process generating the input data. Example applications include stock value forecasting, pattern recognition, weather forecasting etc.

Wavelet Neural Networks (WNNs) contain wavelets as some of the units of the network and participate in the learning process thereby adaptively changing themselves to the need of the desired input-output characteristics. Fig. 2 is an example of a WNN. The output equation can be written as Eq. 13.

III. Application

The aim of equalisation is to estimate the unknown symbol in the corrupted incoming signal. The main cause of corruption is the ISI which is due to the finite bandwidth of the channel. Thus, intuitively, one can see the importance of simultaneous resolution in both time and frequency. Hence the interest in wavelet based techniques. Most equalisation approaches operate in the time domain and hence any Fourier transform based analysis will not simultaneously resolve in both time and frequency. Wavelets give us the flexibility to represent the incoming signal in terms of both temporally and spectrally 'compact' basis elements thereby giving us a more powerful tool for recovering the signals.

Two broad approaches have been reported in the literature we surveyed:

- *Multi-Resolution Analysis (MRA)* based equalisers use the MRA filter banks to emphasise those parts of the signal that need to be detected. This approach is linear in nature.
- The *Wavelet Neural Network (WNN)* based equalisers we looked at, used neural networks with dilates and translates of a mother wavelet as *activation functions*. This is inherently a non-linear approach.

A. Multi-Resolution Analysis based Equalisers

A dyadic Perfect Reconstruction Filter Bank (PRFB) is shown in Fig. 1. At each intermediate step of the filter bank, we get output sequences corresponding to the frequency content (in a given frequency band determined the preceeding filters) of a particular time duration (determined by the duration and shape of the impulse response of the preceeding filters) of the input signal.

In Fig. 1, each branch gain is unity and hence the input signal is perfectly reconstructed at the output. In order to emphasise particular frequency components of the incoming signal in a particular duration of time, one can add some gain terms in these branches so as to minimise the error in the reconstructed symbol and the actual transmitted symbol. This is the idea behind using MRA filter banks as channel equalisers. There are two variables in a PRFB: The *branch gains* and the *filter bank filters* themselves. We looked at a paper each in both these directions.

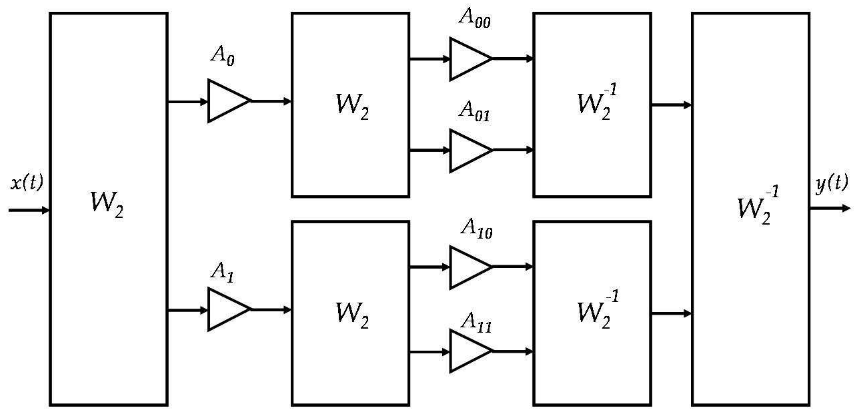

Fig. 3. A *dyadic*-MRA expansion via a Perfect Reconstruction Filter Bank with branch gains

Adding Gain terms in each branch

Cariolaro and Favalli, in [1], use a two bank PRFB structure and apply gains to each of the intermediate branches. The filters in the filter bank are taken from a standard PRFB structure. (Fig. 3)

They consider the input (discrete time) sequence (output of the channel) $x[.]$ as a corrupted version of the actual transmitted sequence $s[.]$ with some mean squared error. The branch gains are then computed so as to minimise the Mean Squared Error (MSE) between the output of one stage of the filter bank structure and the actual transmitted sequence. The aim is to make the MSE between the output and the actual less then that between the input and the actual. Hence, using more and more MRA stages at each branch, one can arbitrarily reduce the error to zero with probability one (statistical optimality).

Consider a single stage of a PRFB with branch gains as shown in Fig. 3, we get the following reconstruction equation:

$$Y(z) = [A_0H_0(z)G_0(z) + A_1H_1(z)G_1(z)]X(z) + [A_0H_0(-z)G_0(z) + A_1H_1(-z)G_1(z)]X(-z) \tag{4}$$

From this equation, the authors determine the values of the gains to minimise the error between the actual transmitted signal and the processed output signal. They next define the following parameters with reference to Fig. **??**.

- *input error* $e[n] = x[n] - s[n]$
- *output error* $u[n] = y[n] - s[n]$
- *intermediate errors*

$$\begin{aligned}
e_0[n] &= x_0[n] - s_0[n] \\
e_1[n] &= x_1[n] - s_10[n] \\
u_0[n] &= y_0[n] - s_0[n] \\
&= A_0x_0[n] - s_0[n] \\
&= (A_0 - 1)s_0[n] + A_0e_0[n] \\
u_1[n] &= x_1[n] - s_1[n] \\
&= A_1x_1[n] - s_1[n] \\
&= (A_1 - 1)[n] + A_1e_1[n]
\end{aligned}$$

Thus, minimising $u[.]$ implies minimising it as a function of A_0 and $A - 1$. They then exploit the orthogonality of the PRFB filters and write the following for real signals. This helps in breaking the MMSE problem into two separate minimisations:

$$f_u(A_0, A_1) = f_0(A_0) + f_1(A_1) \tag{5}$$

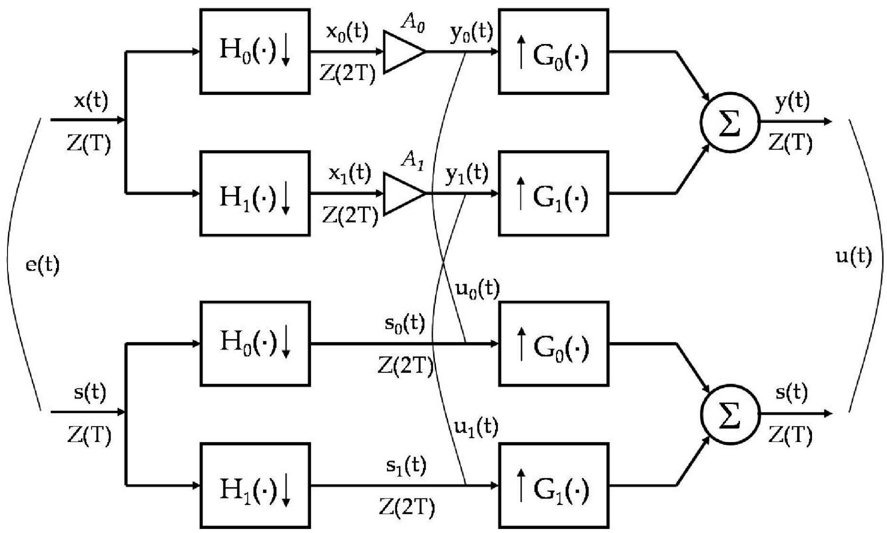

Fig. 4. Error terms for evaluation of the branch gains.

Now, for ideal-PRFB, we can write:

$$E_u = E_{u_0} + E_{u_1} = f_0(A_0) + f_1(A_1) \tag{6}$$

where $E_x = E(x[n]x^*[n])$ (mean instantaneous power in the signal). Since the system is linear, hence using the orthogonality of the error and the observed signal, we get the following values for the optimal branch gains:

$$A_0 = \frac{E(s_0[n]x_0^*[n])}{E_x} \tag{7}$$

$$A_1 = \frac{E(s_1[n]x_1^*[n])}{E_x} \tag{8}$$

Thus, the branch gains can be determined from the training sequence and may be periodically updated as per the need. (In order to estimate the statistical means $E(.)$, one may have to assume ergodicity of the underlying random processes and hence approximate the statistical means by the corresponding time averages.)

In order to evaluate the performance of the system and to clearly see the importance of the Multi-Resolution structure, the authors define following parameters:

- *Error-to-Signal Ratio (ESR)*: At the output Γ

$$\Gamma = \frac{E_u}{E_s} \tag{9}$$

  and, at the input R

$$R = \frac{E_e}{E_s} \tag{10}$$

- *Quality Factor I* (Note the $0 \leq I < 1$ is desired)

$$I = \frac{\Gamma}{R} = \frac{E_u}{E_e} \tag{11}$$

Thus, as we go to higher and higher resolutions by considering more and more stages at each branch of the 'branch gain MRA' structure, we get the following iterative equations (see [1] for derivation) (the *power complementarity* plays an important role in these derivations)

$$\Gamma = \alpha_0 I_0 R_0 + \alpha_1 I_1 R_1 = IR \tag{12}$$

where, for $i = 0, 1$

$$\begin{aligned}\alpha_i &= \frac{E_{s_i}}{E_s}\\ R_i &= \frac{E_{e_i}}{E_{s_i}}\\ \Gamma_i &= \frac{E_{u_i}}{E_{s_i}}\end{aligned}$$

Thus, Eq. 12 can be applied iteratively for more MRA stages thereby reducing the ESR (Γ) arbitrarily to zero with probability one (statistical optimality).

An interesting flexibility that this approach offers is to trade-off implementation complexity for output accuracy. The more the MRA stages in the equaliser, the lower the output error but higher will be the complexity of the implementation. Thus, this advantage can be used to design equalisers which are flexible enough to be able to change their output ESR as per the needs of the application.

Changing the Filter Bank Filters

Tsatsanis and Giannakis, in [2], apply the MRA PRFB structure to the equalisation problem and adaptively change the filter coefficients themselves. The authors go on to derive a MMSE equaliser for a channel varying over relatively fast time scales.

The main idea behind the approach adopted by the authors is to model the channel as a linear FIR system (finite channel memory) and expand the time-varying coefficients onto a wavelet basis (with fixed coefficients). The choice of the basis elements minimises the MSE of the equaliser output w.r.t. training sequence. The mathematics is somewhat involved and we will not present it here. One can refer to the paper for details.

B. Wavelet Neural Networks

Wavelet Neural Networks (WNNs) that we looked at, consist of dilates and translates of a mother wavelet forming the *activation functions* of the Neural Network. All the papers that we looked at used a single hidden layer structure and adaptive weights for each wavelet output.

Fig. 2 shows a prototype WNN with activation functions given by the translates and dilates of the mother wavelet $\Psi(.)$. This wavelet is a function of the input vector $\mathbf{x} \in \mathcal{R}^N$ to $\mathcal{R}$. Thus writing the output of this WNN as a function of the input, we get

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{K} w_i \Psi[D_i(\mathbf{x} - \mathbf{t_i})] \tag{13}$$

where, $D_i = diag(\mathbf{d_i}), 1 \leq i \leq K$ are the diagonal dilation matrices with $\mathbf{d_i} \in \mathcal{R}^N$'s as the dilation vectors, and $\mathbf{t_i} \in \mathcal{R}^N$ are the translation vectors. The choice of the mother wavelet is crucial to the performance of the system.

The papers which we looked at, took the multi-dimensional wavelet $\Psi(.) : \mathcal{R}^N \rightarrow \mathcal{R}$ as the direct product of N scalar wavelets. Thus, $\Psi(\mathbf{x}) = \psi(x_1)\psi(x_2)...\psi(x_N)$, where $\mathbf{x} = (x_1, x_2, ..., x_N)$.

The idea is to approximate the inverse of the channel response through a WNN based filter. The 'goodness' of the approximation depends on the criteria over which the optimisation is done. The papers that we looked at, assume some mother wavelet and differ in the either the parameters that they adaptively update (based on periodic training sequences) or the criterion for optimising the performance of the algorithm.

- Jiang *et al*, in [3], use the Minimum probability of error as the criterion for determining the WNN parameters. The parameters over which the adaptive optimisation is carried out are: dilation, d_i, & translation parameters, t_i, of the mother wavelet and the weights w_i, in the combining branches in the neural network.
- He & He [4] use the MMSE criterion and optimise over d_i, t_i and w_i adaptively.
- Chang & Yeh [5], use the MMSE optimiser and vary the weights, w_i, adaptively.

WNNs are non-linear filters and hence, are much more powerful than linear MRA based wavelet equalisers. The performance comparisons reported in the above papers show that WNN based equalisers perform better than many other classes of linear and non-linear equalisers – at least for the channels considered.

ACKNOWLEDGMENT

The authors would like to thank Prof. V. M. Gadre for solving all our doubts during the course of the lectures, and giving us the opportunity to explore this interesting application of Wavelets.

REFERENCES

[1] D. Cariolaro and L. Favalli, *Recovery of ISI channels using multiresolution wavelet equalization*, IEEE International Conference on Communications 2002, 28 April-2 May 2002, Vol. 1, Pages:74-78.

[2] M. K. Tsatsanis and G. B. Giannakis, *Time-varying channel equalization using multiresolution analysis*, Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis, 4-6 Oct. 1992, Pages:447-450.

[3] M. Jiang *et al*, *Wavelet neural networks for adaptive equalization*, 2002 6th International Conference on Signal Processing, Vol. 2, 26-30 Aug. 2002, Pages:1251-1254.

[4] S. He and Z. He, *Blind equalization of nonlinear communication channels using recurrent wavelet neural networks*, 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 4, 21-24 April 1997, Pages:3305-3308.

[5] P-R. Chang and B-F. Yeh, *Nonlinear communication channel equalization using wavelet neural networks*, 1994 IEEE International Conference on Neural Networks, Vol. 6, 27 June-2 July 1994, Pages:3605-3610.