Sādhanā, Vol. 21, Part 1, February 1996, pp. 75–89. © Printed in India.

Wavelet packet based channel equalization

S GRACIAS[1] and V U REDDY[2]

[1]Motorola India Electronics Pvt. Ltd., No.1, St. Marks Road, Bangalore 560 001, India
[2]Electrical Communication Engineering Department, Indian Institute of Science, Bangalore 560 012, India

Abstract. Recently, considerable amount of attention is being given to the field of wavelets and wavelet packets. It has found numerous applications in signal representation, image compression and applied mathematics.

In this paper, we present a channel equalization method based on wavelet packets. The proposed equalizer structure is based on the fact that for sufficiently narrowband sequences, a non-ideal channel can be modelled as an attenuation and delay. If the data sequence is used to modulate a set of narrowband wavelet packets, then no equalization is required at the receiver end. The equalization problem reduces to that of determining the delay introduced by the channel for each of the wavelet packets. A minimum square variance algorithm for adaptively choosing the delay has been proposed. This algorithm has been shown to perform as desired analytically in a simple delay channel case. Simulations have been used to study its performance in the non-ideal channel's case and the results corroborate theoretical predictions.

Keywords. Channel equalization; wavelets and wavelet packets.

1. Introduction

Practical communication channels are noisy and band-limited. Hence, the received sequence is usually an attenuated, delayed and distorted version of the transmitted sequence (besides the noise introduced by the channel). When a stream of symbols is transmitted over the channel, the distortion results in interference between neighbouring symbols. This inter-symbol interference (ISI) is primarily due to the band-limited nature of the channel. A filter or signal processing algorithm, called an equalizer, is required at the receiver end to remove (or minimize) the effects of ISI. The parameters of the equalizer are adjusted on the basis of measurements of the channel characteristics (Proakis 1983; Qureshi 1985). These measurements could be made by initially transmitting a training sequence which is known to the receiver. Alternatively, in the blind equalization schemes (Benveniste & Goursat 1984), measurements made on the received sequence itself are used to estimate the channel characteristics.

The complexity of the equalizer is substantial for channels with severe ISI. To reduce the complexity of the receiver, the data symbols are used to modulate a narrow-band carrier which is then transmitted over the channel. Since a channel behaves like an ideal delay channel in a sufficiently narrow band, the narrow-band carrier suffers much less distortion thereby requiring reduced compensation and reduced equalizer complexity. However, a single narrow-band carrier would use only a fraction of the channel bandwidth available. This would mean transmitting the data at rates much lower than is possible. There are two complementary approaches to increase the data rate for a given bit error probability.

The first approach to increase the data rate is to use multi-level amplitude (M-ary) modulation of the carrier. The carrier takes M possible signal amplitudes, corresponding to $M = 2^k$ possible k-bit symbols. The increase in data rate, by a factor of M, is gained at the expense of increased signal power; the bandwidth utilization is still the same. Quadrature amplitude modulation is an efficient method to trade-off data rate against signal power.

The second approach to increase the data rate is to use multiple carriers, each occupying different regions of the channel bandwidth. Here, the increase in data rate is gained at the expense of greater bandwidth utilization. An orthonormal set of carriers would, in general, offer the best performance. A number of orthonormal sets have been suggested in the literature. The discrete multi-tone (DMT) (Chow 1992, ch. 2–4) system, for example, uses the Fourier basis sequences as the orthonormal set.

Recently, considerable amount of attention is being given to the field of wavelets and wavelet packets. Wavelet theory provides a unified framework for a number of signal processing techniques which have been independently developed. It has found numerous applications in signal representation, image compression and applied mathematics (Coombes *et al* 1989; IEEE 1992).

Whereas the Fourier basis sequences are all of equal bandwidth, wavelet packets are a generalization to the unequal bandwidth case. Here, we present a channel equalization method based on wavelet packets. The ability to select the bandwidth of the carriers could conceivably be used to improve the efficiency of the DMT system, though, of course, the DMT system has the advantage of having a number of fast algorithms for its implementation.

2. Problem statement

The problem of designing an equalizer and then adaptively choosing the equalizer parameters is a classic one. Recently, with the development of wavelets and renewed interest in multirate systems, a number of adaptive equalization algorithms using sub-band concepts have been proposed (Gilloire & Vetterli 1992; Shynk 1992; Sathe & Vaidyanathan 1993). These algorithms are based on splitting the output signal of the channel into sub-bands, applying standard adaptive equalization algorithms in each sub-band and then recombining the sub-bands to generate the equalized output. The sub-band scheme has greater computational efficiency than the full-band scheme. Furthermore, the convergence speed is improved as the adaptation step size can be matched to the energy distribution of the input signal in that band. However, if decimation is done close to the maximal rate in an attempt to reduce the number of computations, then the performance deteriorates.

Here, we approach the problem of channel equalization using wavelet packets in a different way. Any channel response can be sub-divided into a set of regions (possibly unequal), where its behaviour closely approximates the ideal delay channel. Since a wavelet packet is essentially a narrow-band sequence, a suitably designed packet would be essentially undistorted by passage through the channel. If a data bit is used to switch the polarity of the packet, then at the receiver the bit could be recovered by a simple matched filter-sampler combination.

Since wavelet packets can be designed with finite support, a data sequence could be transmitted over the channel by using time-delayed (shifted) versions of the same wavelet packet. If the delay between two successive wavelet packets is sufficient, the overlap between them is minimal, and at the receiver the data sequence can easily be identified and recovered. However, to increase the data rate, one would like to reduce the delay between the transmission of successive wavelet packets to a minimum. But, as one reduces the delay it becomes increasingly difficult to pick the correct sample at the matched filter output. This is because of the increased overlap between the wavelet packet and its shifted versions.

An important property of a wavelet packet is that it is orthogonal to an n_k-shifted version of itself (where n_k is the decimation factor associated with the wavelet packet in the kth sub-band). Thus, if we reduce the delay between successive wavelet packets to n_k, this property can be exploited by the receiver to recover the transmitted sequence, despite large amounts of overlap. Picking the correct sample is equivalent to estimating the delay introduced by the channel before decimating the matched filter output. Thus, if we use the data sequence to modulate a set of wavelet packets and its shifted versions, no equalization is required at the receiver end. A simple delay and matched filter combination followed by a decimator would suffice. The equalization problem reduces to that of determining the delay introduced by the channel, for each of the wavelet packets.

3. Proposed equalizer structure

The wavelet packet transform (IEEE 1992; Mathiarasan 1992) is a generalization of the Discrete Time Wavelet Transform (DTWT). The transform coefficients for the sequence $x(n)$ are given by

$$x_k(n) = \sum_{m=-\infty}^{+\infty} x(m)h_k(n_k n - m), k = 0, 1, \ldots, M-1, \tag{1}$$

where the n_k's are arbitrary positive integers which satisfy

$$\sum_{k=0}^{M-1} \frac{1}{n_k} = 1, \tag{2}$$

and the filters $h_k(n)$, $k = 0, 1, \ldots, M-1$ form the analysis bank of a non-uniform perfect reconstruction (PR) system. The decimation factor associated with the kth sub-band is n_k and the bandwidth of the kth sub-band is nominally π/n_k. The non-uniform filter bank can be generated by cascading uniform filter banks together in an arbitrary tree structure.

Similarly, the inverse transform relation is given by

$$x(n) = \sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} x_k(m) f_k(n - n_k m). \tag{3}$$

The filters $f_k(n)$, $k = 0, 1, \ldots, M-1$ form the corresponding synthesis bank of the PR system.

For perfect reconstruction, the analysis and synthesis filters have to satisfy the following conditions,

$$f_k(n) = h_k(-n), \tag{4}$$

and

$$\sum_{n=-\infty}^{+\infty} f_k(n) f_m(n - n_{k,m} p) = \delta(k-m)\delta(p),\ k, m = 0, 1, \ldots, M-1, \tag{5}$$

where $n_{k,m} = \gcd(n_k, n_m)$.

The transform can also be interpreted as a projection of the sequence onto a set of orthonormal basis sequences $\eta_{km}(n)$, where

$$\eta_{km}(n) = f_k(n - n_k m),\ k = 0, 1, \ldots, M-1, \text{ and } m \in Z. \tag{6}$$

This orthonormal set of basis sequences is used as the "carrier" set for the data sequence. Before modulating these wave packets with data bits, we first split the data sequence into M sub-sequences. Since the bandwidth of the kth wavelet packet is inversely proportional to n_k, the bits allocated to the kth wavelet packet should also be inversely proportional to n_k. For example, in the uniform case, we could split the sequence $a(n)$ into blocks of length M and assign the kth element of every block to the kth sub-sequence.

The problem reduces to that of transmitting M sub-sequences $\{a_k(n)\}_{k=0}^{M-1}$ on the channel. Each of these sequences could be used to modulate a set of wave packets $\{\eta_{km}(n)\}_{m\in Z}$, to generate the transmitted sequence $t(n)$ as follows

$$t(n) = \sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} a_k(m)\eta_{km}(n). \tag{7}$$

Note from (7) that the mth bit of the kth sub-sequence of the data modulates (i.e., multiplies with the amplitude of the bit) the mth wave packet of the kth set.

Combining (6) with (7), we get

$$t(n) = \sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} a_k(m) f_k(n - n_k m), \tag{8}$$

or in the z domain

$$T(z) = \sum_{k=0}^{M-1} A_k(z^{n_k}) F_k(z) = \sum_{k=0}^{M-1} (A_k(z))_{\uparrow n_k} F_k(z). \tag{9}$$

That is, the modulation can be performed by passing the M sub-sequences through a bank of expanders followed by a synthesis bank as shown in figure 1.

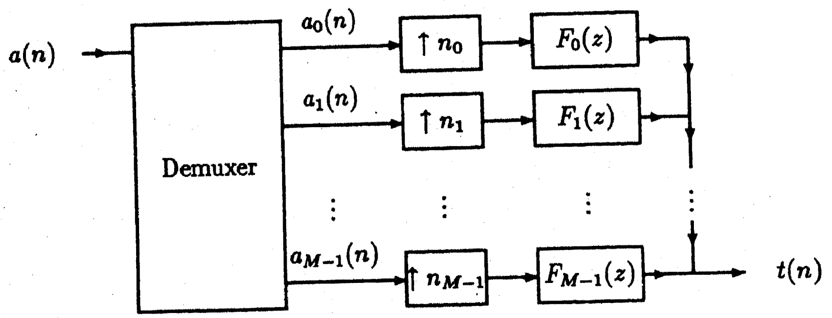


Figure 1. A wavelet packet based equalization scheme: Transmitter section.

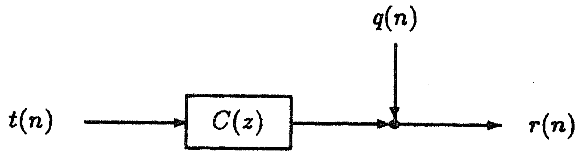


Figure 2. Noisy channel model.

The modulated signal $t(n)$ is then transmitted over a noisy channel (see figure 2). To demodulate the received signal $r(n)$, we first pass it through a bank of matched filters. From (4), the matched filters are simply the corresponding analysis filters. Before decimating the output of the matched filter by a factor n_i, we need to compensate for the delays experienced by the different wave packets (i.e., the different carriers). The receiver structure consisting of the matched filters, delays and decimators (see figure 3) performs the role of equalizer here. Figure 1 is called a transmultiplexer (Vaidyanathan 1993) as it converts a TDM signal into FDM, and vice-versa. We will now consider the problem of determining the delays at the receiver end.

4. Analysis of the equalization scheme

Consider the block diagram of figure 1. The z transform of the transmitted signal, $T(z)$, is given by

$$T(z) = \sum_{k=0}^{M-1} (A_k(z))_{\uparrow n_k} F_k(z) = \sum_{k=0}^{M-1} A_k(z^{n_k}) F_k(z). \tag{10}$$

The z transform of the received signal, $R(z)$, is given by

$$R(z) = T(z)C(z) + Q(z). \tag{11}$$

After equalization in the ith branch, we have

$$\hat{A}_i(z) = (R(z)H_i(z)z^{-\delta_i})_{\downarrow n_i}, 0 \le \delta_i \le n_i - 1. \tag{12}$$

Using (10) and (11)

$$\hat{A}_i(z) = \left(\sum_{k=0}^{M-1} A_k(z^{n_k}) F_k(z) C(z) H_i(z) z^{-\delta_i} + Q(z) H_i(z) z^{-\delta_i} \right)_{\downarrow n_i}. \tag{13}$$

Using (4), we get

$$\hat{A}_i(z) = \left(\sum_{k=0}^{M-1} A_k(z^{n_k}) H_k(z^{-1}) C(z) H_i(z) z^{-\delta_i} + Q(z) H_i(z) z^{-\delta_i} \right)_{\downarrow n_i} \tag{14}$$

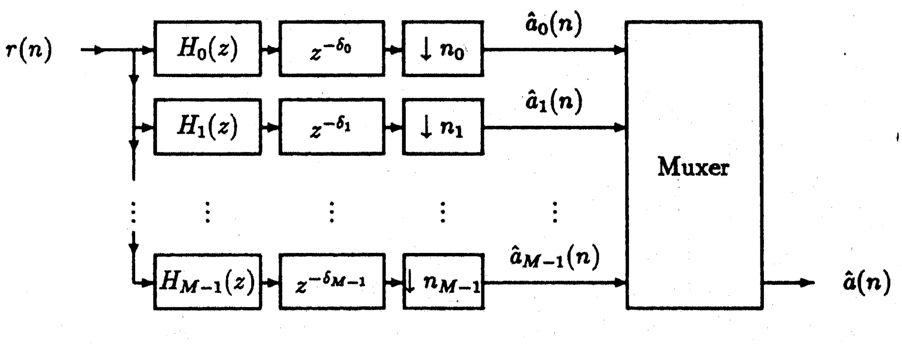


Figure 3. A wavelet packet based equalization scheme: Receiver section.

To simplify the notation, we define

$$S_{ki}(z) \triangleq H_k(z^{-1})H_i(z), \tag{15}$$

$$D_{ki}(z) \triangleq S_{ki}(z)C(z). \tag{16}$$

This gives

$$\hat{A}_i(z) = \sum_{k=0}^{M-1} (A_k(z^{n_k})D_{ki}(z)z^{-\delta_i})_{\downarrow n_i} + (Q(z)H_i(z)z^{-\delta_i})_{\downarrow n_i} \tag{17}$$

or in the time domain

$$\hat{a}_i(n) = \sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} a_k(m) d_{ki}(n_i n - n_k m - \delta_i) + \sum_{m=-\infty}^{+\infty} h_i(m) q(n_i n - m - \delta_i). \tag{18}$$

In the noiseless case, (18) reduces to

$$\hat{a}_i(n) = \sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} a_k(m) d_{ki}(n_i n - n_k m - \delta_i). \tag{19}$$

Thus, the output in the ith branch is not a delayed version of the input even in the noiseless case. This is due to the interference between samples of the same branch signal as well as the interference across branches. We should choose δ_i such that this interference is minimized and the output $\hat{a}_i(n)$ is mapped to a delayed version of $a_i(n)$. The minimum square variance (MSV) algorithm developed below meets this objective. We will motivate this algorithm for the noiseless case.

5. Motivation of minimum square variance algorithm

Consider (19). If $d_{ki}(n_i n - n_k m - \delta_i) = 0$ for $k \neq i$, the interference across branches will be zero. Further, if $d_{ii}(n_i n - n_i m - \delta_i)$ is a delta sequence, the interference between samples of the same branch will be zero. Now the question we ask is the following. Will an appropriate choice for δ_i force the above mentioned conditions on the $d_{ki}(\cdot)$? To see this, we first investigate the properties of $d_{ki}(\cdot)$.

We begin by noting that, using Euclid's identity, we can make the substitution $n_i n - n_k m = n_{k,i} p$, where p is some arbitrary integer and $n_{k,i} = \gcd(n_k, n_i)$. Thus, instead of investigating the properties of $d_{ki}(n_i n - n_k m - \delta_i)$, we look at the properties of $d_{ki}(n_{k,i} m - \delta_i)$. Using Parseval's theorem and the fact that the wavelet packets and their shifted versions form an orthonormal set, it can be shown that (Gracias 1994)

$$\sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} d_{ki}^2(n_{k,i} m - \delta_i) = \frac{1}{2\pi} \int_0^{2\pi} |C(e^{j\theta}) H_i(e^{j\theta})|^2 \, d\theta. \tag{20}$$

We note from (20) that $\sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} d_{ki}^2(n_{k,i} m - \delta_i)$ is independent of δ_i and is equal to the channel energy in the portion specified by the ith branch.

Now, squaring the LHS of (20) gives

$$\left(\sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} d_{ki}^2(n_{k,i} m - \delta_i) \right)^2 = \sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} d_{ki}^4(n_{k,i} m - \delta_i) + \text{positive cross-terms}. \tag{21}$$

Since each term in the RHS of (21) is positive, we get the following inequality

$$\sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} d_{ki}^4(n_{k,i}m - \delta_i) \leq \left(\sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} d_{ki}^2(n_{k,i}m - \delta_i) \right)^2 . \tag{22}$$

The equality in the above equation holds only if all the positive cross terms are zero. This can happen only if exactly one of the terms (i.e., $d_{ki}(\cdot)$) of the sum is non-zero (the trivial case of the equality when all the terms are identically zero is not permissible since the RHS of (20) is guaranteed to be positive). If the wave packets are chosen with a small overlap in the frequency domain, then $H_k(z^{-1})H_i(z), i \neq k$ will be close to zero. This implies from (16) that $d_{ki}(n), i \neq k$ will be close to zero. Thus the above equality holds only if $d_{ii}(n_{k,i}m - \delta_i)$ is a delta sequence and $d_{ki}(n_{k,i}m - \delta_i), i \neq k$ is identically zero. If $d_{ii}(n_i m - \delta_i)$ is a delta sequence, then the output sequence is a delayed version of the input sequence, i.e., there is no ISI.

Note from (20) that the RHS of (22) is independent of δ_i. This suggests that if we choose δ_i to maximize the LHS of (22), then the sequence $d_{ki}(n_{k,i}m - \delta_i)$ will approximately assume the properties mentioned above, thereby minimizing the ISI.

In practice, the channel response is unknown, and hence, $d_{ki}(\cdot)$ is unknown. Thus, we have to make the appropriate choice of δ_i based on the output signal $\hat{a}_i(n)$ and its statistics.

6. Minimum square variance algorithm

In the previous sections, we have shown that to minimize the ISI, we should choose δ_i such that $\sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} d_{ki}^4(n_{k,i}m - \delta_i)$ is maximized. Since $\sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} d_{ki}^2(n_{k,i}m - \delta_i)$ is independent of δ_i (see (20)), we can rewrite

$$\max \left(\sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} d_{ki}^4(n_{k,i}m - \delta_i) \right) \text{ w.r.t. } \delta_i \Leftrightarrow$$

$$\min \left(\left(\sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} d_{ki}^2(n_{k,i}m - \delta_i) \right)^2 - \sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} d_{ki}^4(n_{k,i}m - \delta_i) \right) \text{ w.r.t. } \delta_i . \tag{23}$$

From the statistics of $\hat{a}_i(n)$, we can show that

$$\operatorname{var}[\hat{a}_i{}^2(n)] = 2\alpha^4 \left(\left(\sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} d_{ki}^2(n_{k,i}m - \delta_i) \right)^2 - \sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} d_{ki}^4(n_{k,i}m - \delta_i) \right) . \tag{24}$$

From (24) and (23), we get the following relation

$$\text{maximize} \left(\sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} d_{ki}^4(n_{k,i}m - \delta_i + 1) \right) \text{ w.r.t. } \delta_i \Leftrightarrow$$

$$\text{minimize } (\operatorname{var}[\hat{a}_i^2(n)]) \text{ w.r.t. } \delta_i . \tag{25}$$

Thus if we choose δ_i such that $\operatorname{var}[\hat{a}_i^2(n)]$ is minimized, then the ISI will be minimized. We call the algorithm which performs this minimization as the minimum square variance

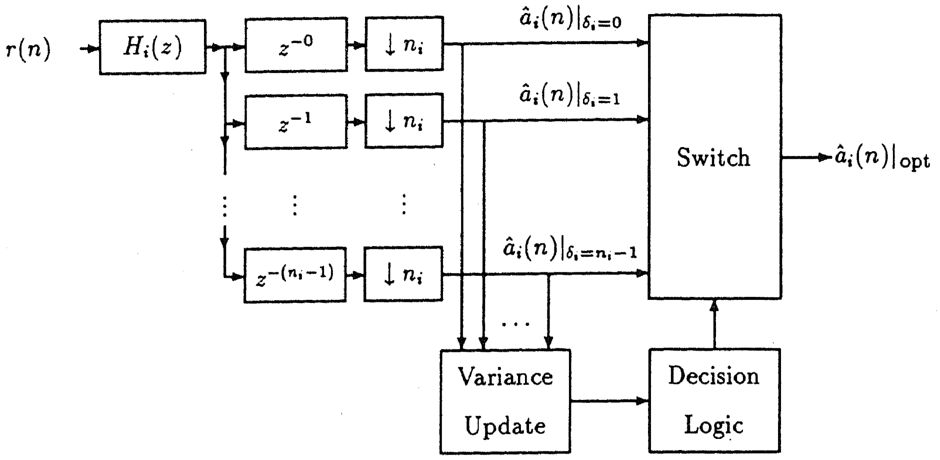

$r(n)$
$H_i(z)$
z^{-0}
z^{-1}
$z^{-(n_i-1)}$
$\downarrow n_i$
$\hat{a}_i(n)|_{\delta_i=0}$
$\hat{a}_i(n)|_{\delta_i=1}$
$\hat{a}_i(n)|_{\delta_i=n_i-1}$
Switch
$\hat{a}_i(n)|_{\mathrm{opt}}$
Variance Update
Decision Logic

Figure 4. Block diagram of the MSV algorithm in the i-th branch.

(MSV) algorithm. The block diagram of the MSV algorithm in the ith branch is given in figure 4.

The output of the ith receiver filter is decomposed into its n_i polyphase components. At every n_ith instant, the sample variance of each of these components is updated. The polyphase component with the smallest sample variance is declared as the output of the branch. We can re-state the algorithm formally as follows:

For each branch at the receiver,

- split the filter output into n_i length blocks,
- setup n_i registers to hold the sample variances,
- initialize these registers to zero,
- use the kth sample of the block to update the variance in the kth register,
- declare the kth sample of the block as the desired received output of the branch, if the value of the kth register is minimum.

7. Performance of the MSV algorithm in a simple delay channel case

We will now explore how the MSV algorithm performs in the simple case of a noiseless delay channel. Assuming,

$$C(z) = z^{-\gamma} \tag{26}$$

and using (16), we get

$$D_{ki}(z) = z^{-\gamma} S_{ki}(z). \tag{27}$$

Substituting (27) into (17) with $Q(z) = 0$ gives

$$\hat{A}_i(z) = \sum_{k=0}^{M-1} (A_k(z^{n_k}) S_{ki}(z) z^{-\gamma-\delta_i})_{\downarrow n_i}. \tag{28}$$

Suppose we choose δ_i such that,

$$-\gamma - \delta_i = -pn_i \tag{29}$$

where p is some arbitrary integer (note that this is always possible as $0 \leq \delta_i \leq n_i - 1$). Using the Noble identities, (28) reduces to

$$\hat{A}_i(z) = \sum_{k=0}^{M-1} (A_k(z^{n_k/n_{k,i}}) z^{-pn_i/n_{k,i}} (S_{ki}(z))_{\downarrow n_{k,i}})_{\downarrow n_i/n_{k,i}}. \tag{30}$$

Now

$$(S_{ki}(z))_{\downarrow n_{k,i}} = (H_k(z^{-1})H_i(z))_{\downarrow n_{k,i}}. \tag{31}$$

The LHS of the above equation is just a rewriting of the orthonormality condition in the z-domain. Thus, we have

$$(S_{ki}(z))_{\downarrow n_{k,i}} = \delta(k, i). \tag{32}$$

Using (32) in (30), we have

$$\begin{aligned}\hat{A}_i(z) &= \sum_{k=0}^{M-1} (A_k(z^{n_k/n_{k,i}}) z^{-pn_i/n_{k,i}} \delta(k, i))_{\downarrow n_i/n_{k,i}} \\ &= (A_i(z^{n_i/n_{i,i}}) z^{-pn_i/n_{i,i}})_{\downarrow n_i/n_{i,i}} \\ &= A_i(z) z^{-p}.\end{aligned} \tag{33}$$

Thus, if we choose δ_i according to (29), then the output is an undistorted version of the input.

To see what the MSV algorithm gives, substitute (27) in (24). This gives

$$\begin{aligned}\operatorname{var}[\hat{a}_i{}^2(n)] = 2\alpha^4 \Bigg(&\left(\sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} s_{ki}^2 (n_{k,i}m - \gamma - \delta_i) \right)^2 \\ &- \sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} s_{ki}^4 (n_{k,i}m - \gamma - \delta_i) \Bigg).\end{aligned} \tag{34}$$

If δ_i satisfies (29), then

$$\begin{aligned}\operatorname{var}[\hat{a}_i{}^2(n)] = 2\alpha^4 \Bigg(&\left(\sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} s_{ki}^2 (n_{k,i}m - n_i p) \right)^2 \\ &- \sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} s_{ki}^4 (n_{k,i}m - n_i p) \Bigg).\end{aligned} \tag{35}$$

Making the substitution $l = m - n_i/n_{k,i} p$ in (35), we get

$$\operatorname{var}[\hat{a}_i{}^2(n)] = 2\alpha^4 \left(\left(\sum_{k=0}^{M-1} \sum_{l=-\infty}^{+\infty} s_{ki}^2 (n_{k,i}l) \right)^2 - \sum_{k=0}^{M-1} \sum_{l=-\infty}^{+\infty} s_{ki}^4 (n_{k,i}l) \right). \tag{36}$$

Now, using (15) and (5),

$$\begin{aligned}s_{ki}(n_{k,i}l) &= \sum_{n=-\infty}^{+\infty} f_k(n) f_i(n - n_{k,i}l) \\ &= \delta(k - i)\delta(l), k, i = 0, 1, \ldots, M - 1.\end{aligned} \tag{37}$$

From (37), we obtain

$$\begin{aligned}\mathrm{var}[\hat{a}_i{}^2(n)] &= 2\alpha^4\left(\left(\sum_{k=0}^{M-1}\sum_{l=-\infty}^{+\infty}(\delta(k-i)\delta(l))^2\right)^2\right.\\ &\qquad\left.-\sum_{k=0}^{M-1}\sum_{l=-\infty}^{+\infty}(\delta(k-i)\delta(l))^4\right)\\ &= 2\alpha^4(1-1)=0. \end{aligned} \tag{38}$$

Thus, for δ_i satisfying (29), the variance of $a_i^2(n)$ is identically zero. For all other values of δ_i, the variance would be non-zero as (37) can no longer be applied. We can therefore conclude that for a simple delay channel (with no noise), the MSV algorithm will yield the proper value of the delay.

7.1 *Computational issues*

Suppose the data sequence is arriving at a rate B (with respect to some system of units). At the transmitter end, the sequence is split into M sub-sequences. The ith sub-sequence will be at the rate B/n_i. The sub-sequences are passed through an n_i-fold expander. Thus, the filtering operations have to be performed at the rate B. Since the bandwidth of the ith filter is inversely proportional to i, we can assume that its length is n_iL. Thus, the computation rate in each branch is approximately n_iLB. Thus, the total computation rate at the transmitter end is approximately $\sum_{i=0}^{M-1} n_iLB$. However, if we use the polyphase representation to implement the n_i-fold expander and filter cascade, then we can reduce the rate in each branch by a factor n_i (Vaidyanathan 1993). This makes the computation rate for the transmitter section approximately MLB.

The computational rate at the transmitter can be further reduced if the filter bank is implemented using a tree structure. For example, consider the uniform case, with $M = 2^p$. Instead of implementing the filter bank as a set of M-fold expander-filter combinations, we could implement it as a cascade of p stages of 2-fold expander-filter combinations. The filter lengths would be ML and $2L$ respectively. This would reduce the computation rate to $2pLB$, which compares favorably to FFT-based schemes.

The sequences are then combined and transmitted over the channel at the rate B. As in the case of the transmitter the total computation rate for demodulation is approximately $\sum_{i=0}^{M-1} n_iLB$. The polyphase representation cannot be used here as all the polyphase components of the received signal are required to make the decision.

Since $\sum_{k=0}^{M-1}\sum_{m=-\infty}^{+\infty} d_{ki}^2(n_{k,i}m - \delta_i)$ is independent of δ_i, minimizing the variance of $\hat{a}_i^2(n)$ is equivalent to maximizing the fourth moment of $\hat{a}_i(n)$. Updating of the sample fourth moment requires 4 multiplications and 1 addition operation. If we consider only the multiplications, the computation rate is $4MB$ since there are n_i polyphase components in each branch, and these components are arriving at a rate of B/n_i. Thus the total computation rate at the receiver is $(4M + \sum_{i=0}^{M-1} n_iL)B$. Once the algorithm converges, only the polyphase component corresponding to the selected delay has to be computed. This brings down the computation rate at the receiver to MLB.

The equalization scheme has been proposed for a stream of bits. At the transmitter, we map the bits 0 and 1 to the levels $+\alpha$ and $-\alpha$, respectively, to generate the input to the equalizer. Similarly, at the receiver the received signal (after equalization) has to be mapped back to a bit stream. This can be done using an appropriate threshold.

The equalizer scheme is based on the fact that the channel can be approximated by a simple delay in a sufficiently narrow band. For the M branch (uniform case) equalizer, each wavelet packet has a bandwidth of π/M. Clearly, if we increase M, the approximation gets better. However, we have seen that the computation complexity per sample is $0(M^2)$. Thus, increasing M imposes a heavy computational burden on the system.

The design of appropriate wavelet packets is equivalent to the design of a non-uniform PR filter bank. This can be accomplished by cascading appropriate (uniform) paraunitary filter banks. A design technique for such banks, based on cosine modulation (Koilpillai & Vaidyanathan 1992), requires the desired length of the filters and a cost function (to be minimized) as design parameters. Recall from the previous sections that the inter-branch interference in the ith branch is small if $d_{ki}(n_{ki}n - \delta_i)$, $i \neq k$ is close to zero. In the absence of any information about the channel, we could use the cost function $\sum_{n=-\infty}^{+\infty} s_{ki}^2(n)$ for the design of the wave packets. This will ensure that $d_{ki}(n)$, $i \neq k$ is small (see (16)), and hence, the inter-branch interference using the designed wave packets is small.

8. Simulations

In the simulations, we considered the uniform case, i.e., $n_i = M, i = 0, 1, \ldots, M-1$. The input was a random sequence taking values +1 and -1 with equal probability. We first consider the case of a simple delay channel, $C(z) = z^{-1}$. The received signal for the two-branch equalizer with compensating delays of both zero and one is shown in figure 5. It is clear from the figure that the output depends critically on the delay chosen. A wrong choice of delay would result in a wrong decoding of the received signal. This clearly illustrates the problem of picking the correct sample at the output of the matched filter. The delay computed by the MSV algorithm for this case is shown in figure 6.

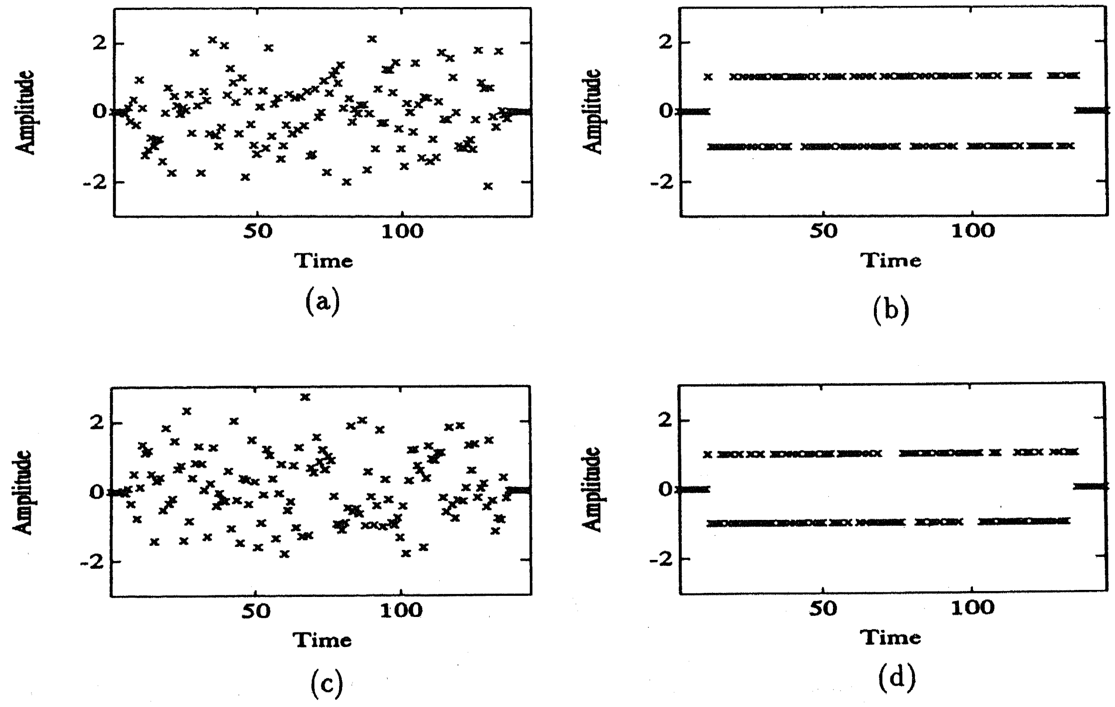


Figure 5. Received signals for a simple delay channel with a two-branch equalizer with $\delta_i = 0$ and 1. **(a)** $i = 0, \delta_i = 0$. **(b)** $i = 0, \delta_i = 1$. **(c)** $i = 1, \delta_i = 0$. **(d)** $i = 1, \delta_i = 1$.

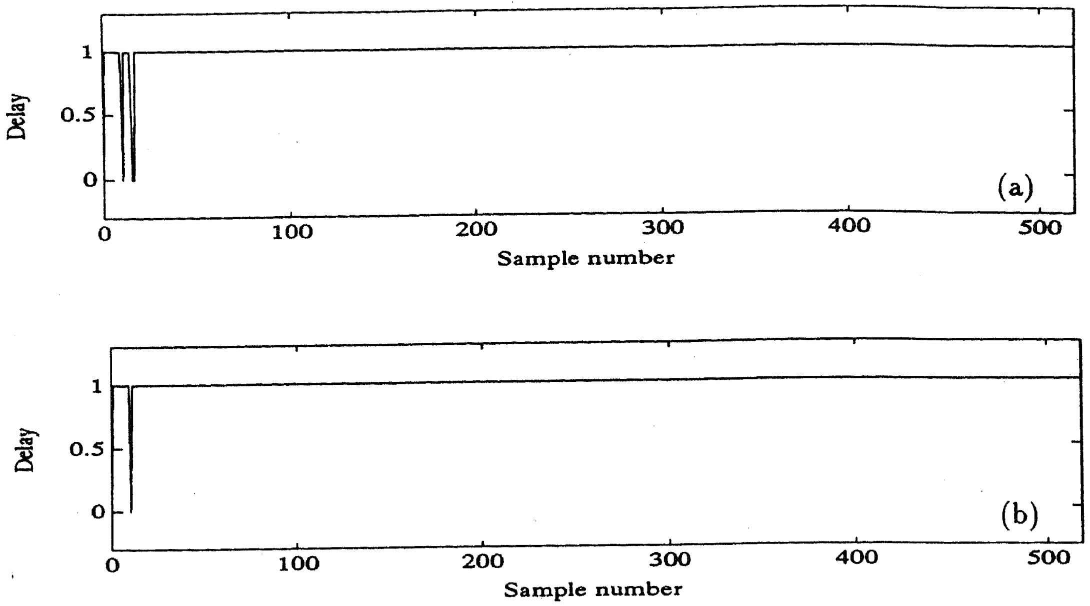


Figure 6. Delay computed by the MSV algorithm for the two-branch equalizer for a simple delay channel in the noise-less case. (**a**) $i = 0$. (**b**) $i = 1$.

To study the effect of noise, we consider the simple delay channel with zero mean white Gaussian noise for the two-branch case. The delay computed by the MSV algorithm for the noisy case with SNR's of 10dB and 5dB are shown in figure 7. We note that the algorithm takes longer time to converge, but the converged value of the delay is unaffected by noise. The convergence of the fractional error for the noiseless case and for the noisy case with SNR's of 10dB and 5dB is shown in figure 8. Note that the noise affects the decoding to a bit stream and hence the steady-state error.

To test the wavelet packet based equalization scheme and the MSV algorithm, simulations were carried out using the three channels (denoted by A, B and C, respectively) with impulse responses as shown in figure 9.

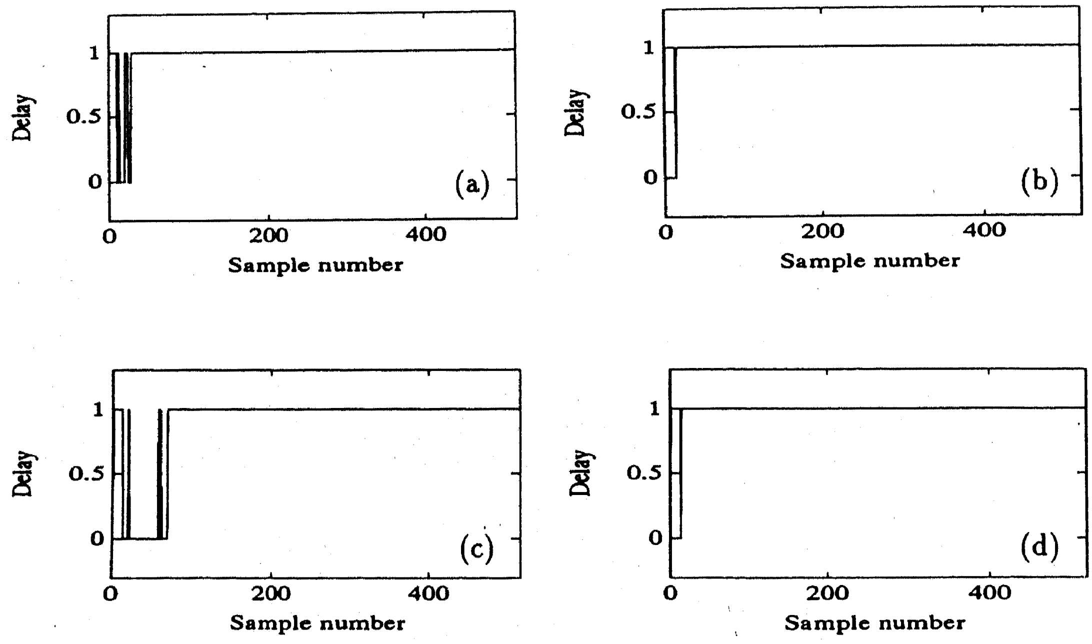


Figure 7. Delay computed by the MSV algorithm for the two-branch equalizer for a simple delay channel in the noisy case. (**a**) $i = 0$, (**b**) $i = 1$ (SNR $= 10$ dB). (**c**) $i = 0$, (**d**) $i = 1$ (SNR $= 5$ dB).

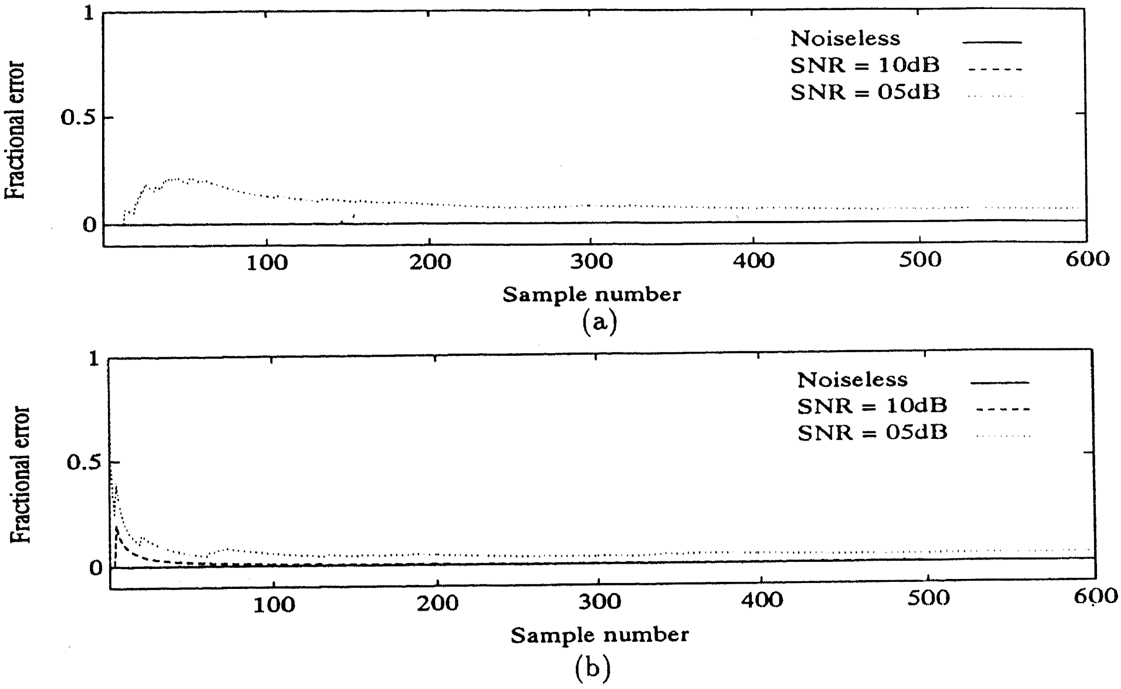


Figure 8. Fraction of bits in error for a delay channel for a two-branch equalizer with additive noise. (**a**) $i = 0$. (**b**) $i = 1$.

The fractional error (the fraction of bits in error, i.e., the number of bits in error upto the nth instant divided by the total number of bits received upto that instant) at the receiver output is shown in figure 10 for a typical two-branch equalizer in the case of the three channels A, B and C. Note that the steady state error is minimum in channel C and maximum in channel B. This implies that channel B causes maximum ISI. This is evident

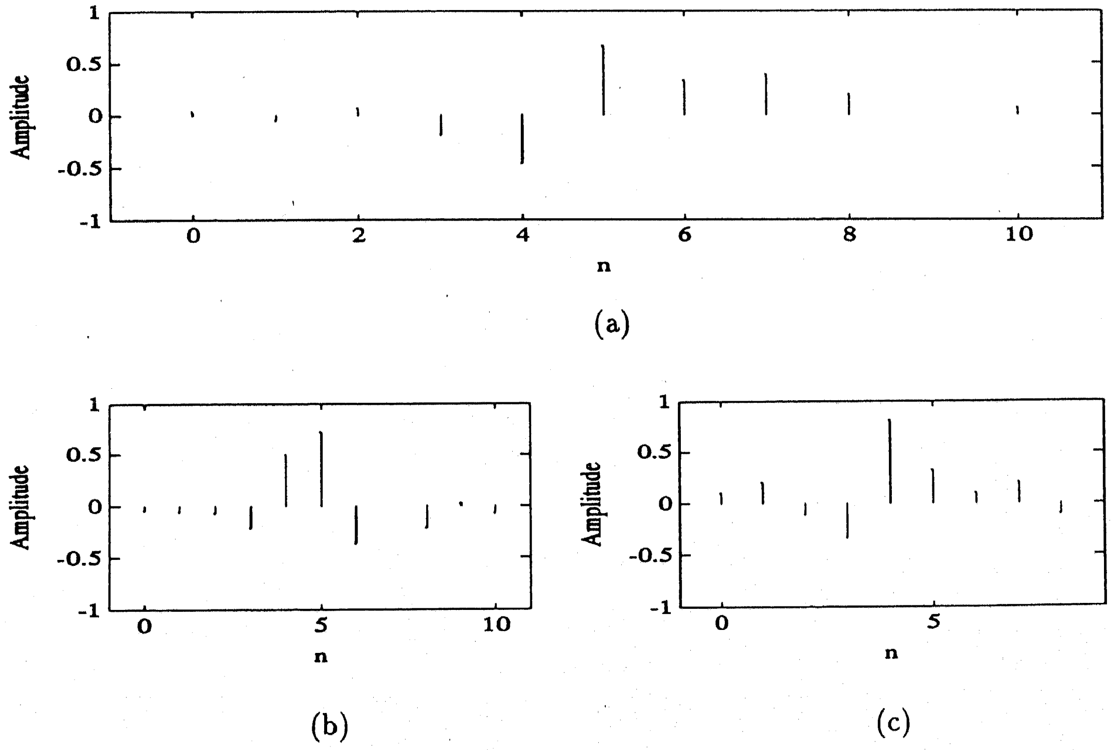
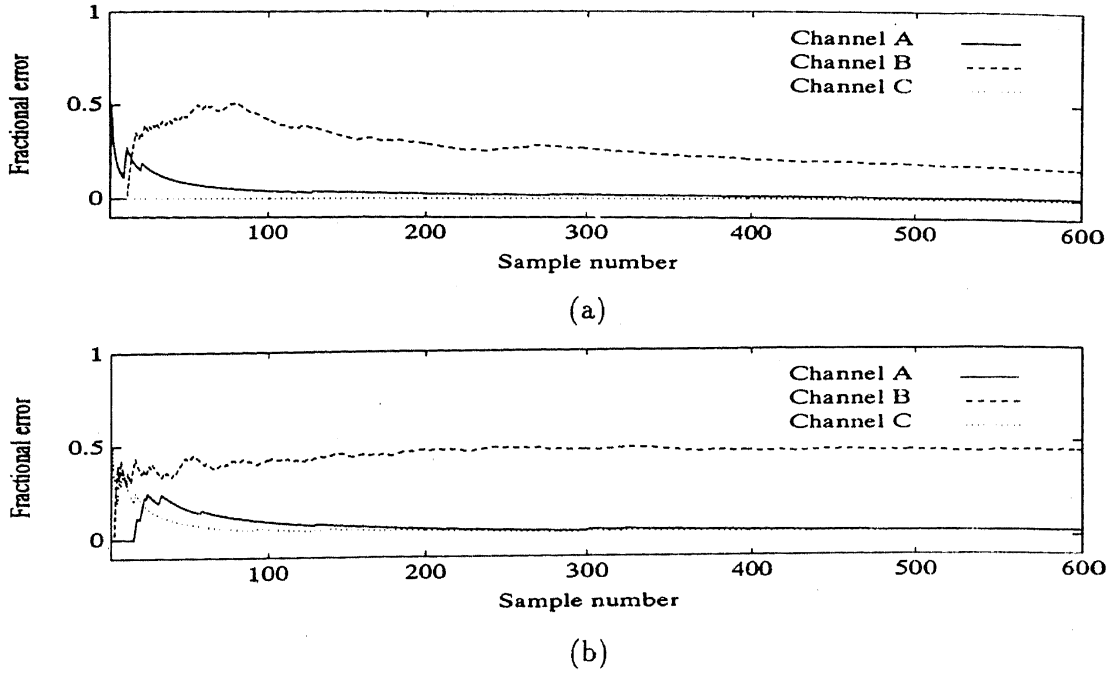


Figure 9. Impulse responses of three typical channels. (**a**) Channel A. (**b**) Channel B. (**c**) Channel C.



(a)

(b)

Figure 10. Fraction of bits in error for a typical two-branch equalizer for different channels. (**a**) $i = 0$. (**b**) $i = 1$.

from the impulse response of channel B, where two coefficients have nearly the same amplitude.

The effect of increasing the number of branches is shown in figure 11, which gives the plot of the fractional error in the zeroth branch for equalizers with two, three and five branches, respectively. The steady-state fractional error is lowest in the five-branch case. This is because in the five-branch case, each wavelet packet occupies a comparatively smaller bandwidth and hence is relatively undistorted.

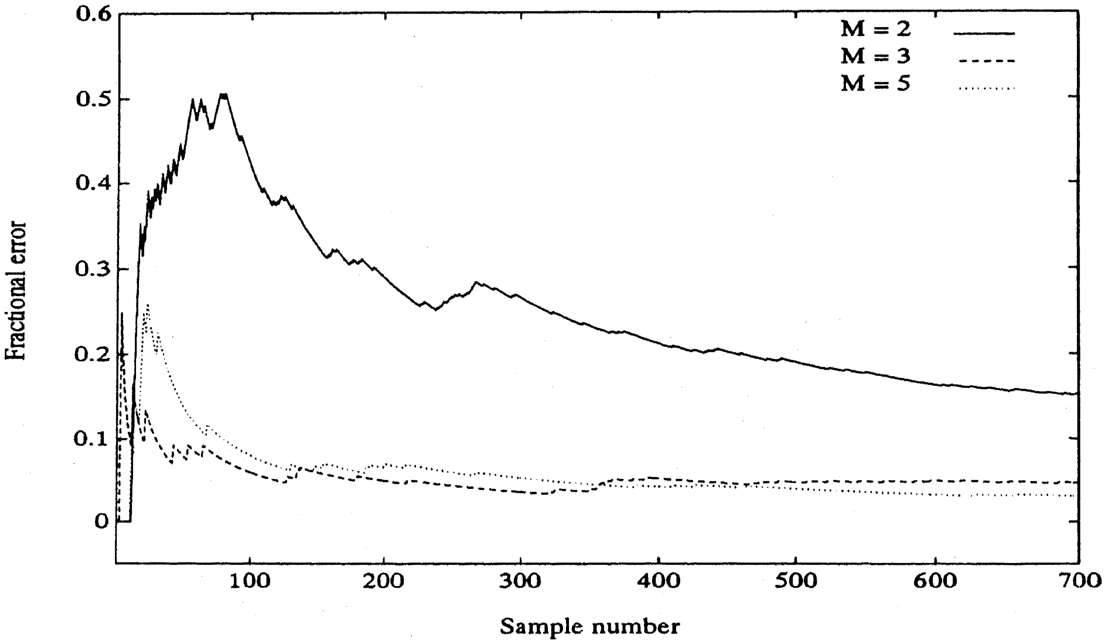


Figure 11. Fraction of bits in error for the zeroth branch of an M-branch equalizer.

9. Conclusions

The paper addresses the problem of channel equalization using wavelet packets. Wavelet packets were introduced as a natural generalization of the orthonormal basis sequences used in Fourier analysis, namely, the sinusoids. The proposed equalizer scheme was based on the fact that wavelet packets, being narrow-band, suffer only attenuation and delay when passed through a non-ideal channel.

The equalizer structure, comprising a wavelet packet modulator, a compensating delay and a matched-filter demodulator was shown to be easily implementable in terms of filters and other multirate components. An algorithm to choose the delay values adaptively for each wavelet packet was motivated using the inherent multirate and orthonormal properties of the wavelet packet set. The algorithm (called the MSV algorithm) uses the variance of the square of the received sequence to choose that value of the compensating delay which minimizes the ISI.

Simulations were carried out to test the equalizer structure and the MSV algorithm. The two-branch equalizer was tested for different channel models and also for various noise levels. The algorithm converges rapidly, and the steady state fractional error (which asymptotically becomes the probability of error) is low except for cases where the channel-generated interference is pronounced. It is demonstrated that such cases can be tackled by increasing the number of branches in the equalizer, i.e., by appropriately selecting the wavelet packets.

References

Benveniste A, Goursat M 1984 Blind equalizers. *IEEE Trans. Commun.* COM-32: 871–883

Chow J S 1992 *Finite-length equalization for multi-carrier transmission systems.* Ph D thesis, Dept. of Elec. Eng., Stanford University, California

Coombes J M, Grossmann A, Tchamitchian P (eds) 1989 *Wavelets: Time-frequency methods and phase space* (Berlin: Springer-Verlag); *Proc. Intl. Conf.* (Marseille, France, 1987)

Gilloire A, Vetterli M 1992 Adaptive filtering in sub-bands with critical sampling: analysis, experiments, and applications to acoustic echo cancellation. *IEEE Trans. Signal Process.* 40: 1862–1875

Gracias S 1994 *Channel equalization using wave packets.* Master's thesis, Dept. Elec. Commun. Eng., Indian Institute of Science, Bangalore

IEEE 1992 Special issue on Wavelet Transforms and Multiresolution Signal Analysis. *IEEE Trans. Inf. Theory* IT-38

Koilpillai R D, Vaidyanathan P P 1992 Cosine-modulated fir filter banks satisfying perfect reconstruction. *IEEE Trans. Signal Process.* 40: 770–783

Mathiarasan S 1992 *Subspace splitting algorithm for wave packet generation and its application in channel modelling.* M.Sc.(Eng.) thesis, Dept. Elec. Commun. Eng., Indian Institute of Science, Bangalore

Proakis J 1983 *Digital communications* (New York: McGraw Hill)

Qureshi S U H 1985 Adaptive equalization. *Proc. IEEE.* 73: 1349–1387

Sathe V P, Vaidyanathan P P 1993 Effects of multirate systems on the statistical properties of random signals. *IEEE Trans. Signal Process.* 41: 131–146

Shynk J J 1992 Frequency-domain and multirate adaptive filtering. *IEEE Signal Process. Mag.* 9: 14–37

Vaidyanathan P P 1993 *Multirate systems and filter banks* (Englewood Cliffs, NJ: Prentice Hall)