Reduction of Noise over Wireless Communication

Shalini Gupta
Department of electronics & Communication Engg.
RAMA University, Mandhana , Kanpur
Shalini2009ec@gmail.com

***Abstract*:- An adaptive decision feedback equalizer is presented with a new adaptive algorithm. The algorithm uses sign normalized block based least mean square algorithm, and achieves a significant reduction of computational complexity. The proposed algorithm yields good bit error rate performance over a reasonable signal to noise ratio. In this scheme the incoming data is partitioned into non overlapping blocks and the filtering operation has been performed in frequency domain.**

Keywords—SISO Equalizer; Least Mean Square

I. INTRODUCTION

Physical channels used in transmission of digital signals can be rarely represented by a non-distorting channel model with additive noise as the only impairment. In practice the vast majority of channels are characterized by a limited bandwidth in which particular frequency components of transmitted signals are non-equally distorted (causing amplitude distortion) and non-equally delayed (causing delay distortion). These effects are the result of the physical properties of transmission medium and of the imperfect design of transmitter and receiver filters.

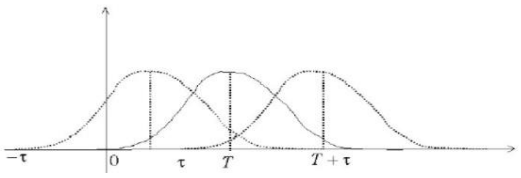

Figure 1: Intersymbol interference.

Period, channel responses to the subsequent data signals over-lap. The distortion caused by the resulting overlap of the received symbols called intersymbol interference (ISI). In many cases the channel impulse response spans even tens of signalling periods and the resulting ISI appears to be a major impairment introduced by the channel. Most high bit-rate communication systems suffer from intersymbol interference (ISI) in addition to noise during transmission over frequency selective channels. The ISI severely distorts signals transmitted through a mobile channel. This phenomenon is caused by multipath propagation and is the main factor that limits the performance of mobile communications. If the characteristics of the transmitting channel are time varying, ADFEs are often used to detect symbols and update the filter coefficients. However, in a number of applications, the computational complexity of the adaptive equalization part may be prohibitive. The excessive burden is mainly due to the long feedback part of the DFE, which is imposed by the nature of the problem. Conventional solutions use separate equalization and decoding functions to compensate for the ISI and noise, respectively. Recently, the concept of turbo equalization has been proposed, where the equalization and decoding functions are iteratively combined through the exchange of soft information between the constituent devices. In turbo equalization, a soft-input soft output (SISO) equalizer and a SISO decoder exchange soft information to jointly estimate the transmitted data.

Nyquist investigated the problem of specifying a received pulse so that no ISI occurs at the detector. He showed that the theoretical minimum bandwidth needed in order to detect R_s symbols, without ISI is $R_s/2$ Hz. This occurs when the system (consisting of the transmit filter, the channel, and the receive filter) transfer function $H_a(f)$ is made rectangular. Nyquist established that the basic pulse $h_a(t)$, with $h_a(t) \leftrightarrow H_a(f)$, needs to satisfy certain zero crossing condition in order to ensure zero ISI. If a_n, $n \in z$ denotes the transmitted sequence and τ denotes the symbol period. For baseband systems, the bandwidth required to detect $1/\tau$ such pulses (symbols) per second is equal to ½. However, in practice, it is almost impossible to design ideal Nyquist filters with rectangular transfer function. As a result, it is not possible to eliminate the ISI completely by the design of Nyquist pulses alone.

A. Turbo Codes

Turbo coding has been used as an FEC technique for transmission over noisy channels. It has been used over the last decade in different wireless communication

systems such as 3G wireless system, Code Division Multiple Access (CDMA), deep space, and satellite communication systems [3, 7, 10]. This section serves as a background for turbo coding as a channel coding technique

Turbo codes, also known as Parallel Concatenated Convolutional Codes (PCCC), and serial concatenated codes concatenate two codes to achieve a good tradeoff between coding gain and decoding complexity [4, 6]. Serial Concatenated Convolutional Codes (SCCC) uses two codes in series separated by an interleaver as shown in figure [2]. In a turbo equalizer, it is known that different portions of a data block may require a large number of iterations between the equalizer and decoder to converge. This is similar in nature to the behavior observed in decoding of turbo codes. We exploit this property to elimination. We further reduce power in the system via early terming. We can achieve early termination by monitoring the soft information passed between the SISO devices typical channels show that the proposed complexity reduction techniques for linear turbo equalization can achieve significant energy savings.

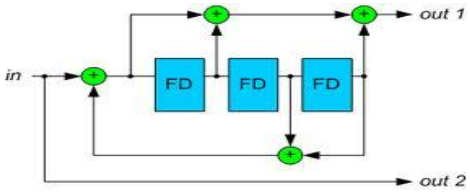

Figure 2.RSC encoder [10].

The primary reason for using a concatenated code is to achieve a low error rate with an overall decoding complexity that is lower than the one required for a single code with the same decoding performance. An interleaver is incorporated between the two codes to decor relate the received symbols that are affected by burst errors generated by the inner decoder. Turbo coding has been used as an FEC technique for transmission over noisy channels. It has been used over the last decade in different wireless communication systems such as 3G wireless system, Code Division Multiple Access (CDMA), deep space, and satellite communication systems. This chapter serves as a background for turbo coding as a channel coding technique.

B. Turbo Encoder

The general structure used in turbo encoders is shown in Figure [3], two component codes are used to code the same input bits, but an interleaver is placed between the encoders. Generally RSC codes are used as the component codes, but it is possible to achieve good performance using a structure like we shown with the aid of other component codes, such as for example block codes figure [3], Furthermore, it is also possible to employ more than two component codes. We concentrate entirely on the standard turbo encoder structure using two RSC codes. K=3 RSC code we have used as the component codes in most of our simulations. This code has generator polynomials 7 (for the feedback polynomial) and 5.

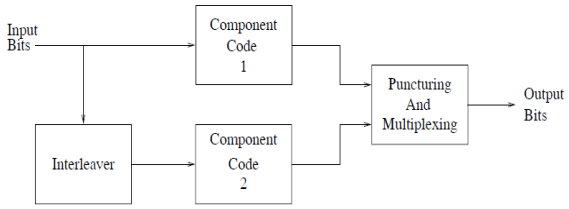

Figure 3. Turbo Encoder Schematic [10]

The outputs from the two component codes are then punctured and multiplexed. Usually both component RSC codes are half rate, giving one parity bit and one systematic bit output for every input bit. Then to give an overall coding rate of one-half, half the output bits from the two Encoders must be punctured. The arrangement that is often favored, and that we have used in our work, is to transmit all the systematic bits from the RSC encoder, and half the parity bits from each encoder. The systematic bits are rarely punctured, since this degrades the Performance of the code more dramatically than puncturing the parity bits. Therefore extremely good performance can be achieved with reasonable complexity by using very long interleaves [5,10]. However, for many important applications, such as speech transmission, extremely long frame lengths are not practical because of the delays. Therefore we have also investigated the use of turbo codes in conjunction with short frame lengths of the order of 100 bits. The generator matrix of a rate 1/2 constituent RSC code can be represented as:

$$g(D) = [1 \quad \frac{g1(D)}{G0(D)}] \qquad (10)$$

Where $G_O(D)$ and $g1(D)$ are respectively feedback and feed forward polynomials with degree m.

C. Log Likelihood Ratio (LLR)

The concept of LLRs was shown by Robertson to simplify the passing of information from one component decoder to the other in the iterative decoding of turbo codes, and so is now widely used in the turbo coding literature. The LLR of a data bit (U_K) is denoted as $L(U_K)$ and is depend to be merely the log of the ratio of the probabilities of the bit taking its two possible values i.e.

$$L(U_K) = \ln \frac{P(U_K = +1)}{P(U_K = -1)} \qquad (2)$$

The two possible values for the bit U_K are taken to be +1 and −1, rather than 1 and 0. This definition of the two values of a binary variable makes no conceptual difference, but slightly simples the mathematics in the derivations which follow. The sign of the LLR $L(U_K)$ of a bit (U_K) will indicate whether the bit is more likely to be +1 or−1, and the magnitude of the LLR gives an indication of how likely it is that the sign of the LLR gives the correct value of (U_K). The bracketed term in this equation does not depend on probability that (U_K) = +1 or −1, and so it can be treated as a constant in certain applications, such as where we use this equation in the derivation of the MAP algorithm. If we assume that the transmitted bit (X_K) k = ±1 has been sent over a Gaussian or fading channel using BPSK modulation, then we can write for the probability of the matched output (Y_K) that:

$$P(Y_K, X_K = +1) = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-E_B}{2\sigma^2} (Y_K - a)^2 \quad (3)$$

Where E_b is the transmitted energy per bit, σ2 is the noise variance and a is the fading amplitude (we have a = 1 for non-fading AWGN channels). Similarly, we have:

$$P(Y_K, X_K = -1) = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-E_B}{2\sigma^2} (Y_K + a)^2 \quad (4)$$

Therefore, when we use BPSK over a (possibly fading) Gaussian channel, we can rewrite Equation then:

$$L_C = 4a \frac{E_B}{2\sigma^2} \qquad (5)$$

Is defined as the channel reliability value, and depends only on the signal-to-noise ratio (SNR) and fading amplitude of the channel. Hence, for BPSK or QAM over a (possibly fading) Gaussian channel, The conditional LLR $L(Y_K, X_K)$, which is referred to as the soft output of the channel, is simply the matched filter output Y_K multiplied by the channel reliability value Lc. Having introduced LLRs, to describe the operation of the MAP algorithm, which is one of the possible soft-in soft-out component decoders that can be used in an iterative turbo decoder.

II. LINEAR EQUALIZER

In digital communications, a turbo equalizer is a type of receiver used to receive a message corrupted by a communication channel with intersymbol interference (ISI) [2]. It approaches the performance of a maximum a posteriori (MAP) receiver via iterative message passing between a soft-in soft-out (SISO) equalizer and a SISO decoder. It is closely related to turbo codes, as a turbo equalizer may be considered a turbo decoder if the channel is viewed as a convolutional code.

The problem is better approached digitally by employing a digital filter at the receiver which approximates the inverse of the equivalent discrete time model of the channel. Note that the n-th receive symbol in the absence of noise, $r_n = (T + n\tau)$.

With $h_a(t) \leftrightarrow H_a(f)$, the ISI can then be eliminated by using a filter with transfer function $C(z) \approx 1/(z)$ at the receiver. Such a filter, popularly called zero forcing equalizer, is shown in Figure [5], where n_k denotes the additive channel noise.

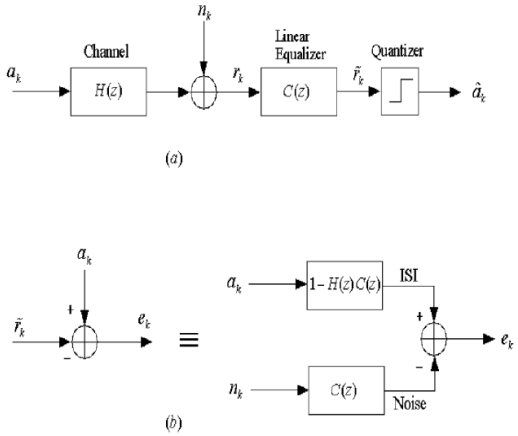

Figure 5 Basic block diagram of linear turbo equalizer

Denoted by e_k the error in the equalized pre-decision output $\tilde{r}_k . e_k = a_k - \tilde{r}_k$. Figure [5]b showed an equivalent structure, showing the contributions to the error signal, from the ISI and the filtered noise.

A. Least mean squares (LMS) Adaptive algorithm

In practice however, the channel characteristics is often not known and is found to vary with time [13]. For example, in case of circuit-switched telephone network, channel characteristics varies widely from call to call due to different switching paths while in the context of mobile communications, the channel characteristics changes significantly, within a call duration particularly because of the relative movements of the nearby reflectors (towers, buildings etc.) and also during handoff. The equalizer coefficients in such cases need to adapt themselves continuously to track the channel variations.

.

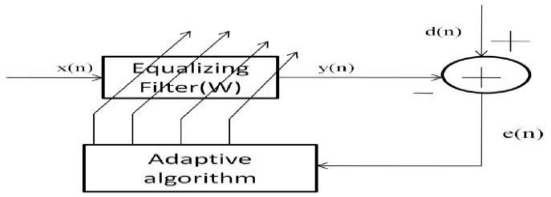

Figure 6 LMS adaptive algorithm system

LMS algorithms are a class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal (difference between the desired and the actual signal). It is a stochastic gradient descent method in that the filter is only adapted based on the error at the current time. It was invented in 1960 by Stanford University professor Bernard Widrow Ted Hoff

Consider a Method of Steepest Descent

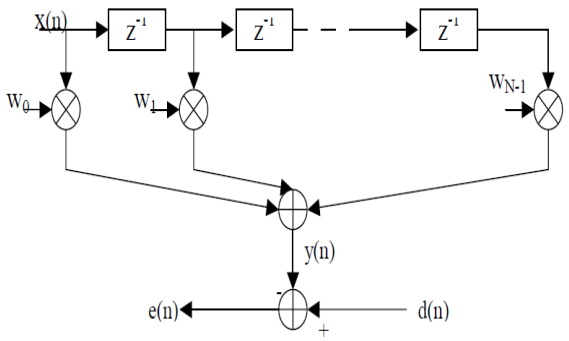

Figure 7 transversal wiener filter

- Assuming that all the signals involved are real-valued signals
- Signal input $\overline{w} = [w_0, w_1, w_2 \ldots\ldots\ldots w_{N-1}]^T$
- Tap-weight vector $\bar{x}(n) = [x(n), x(n-1,)x(n-2)\ldots\ldots\ldots x(n-N+1)]^T$
- filter output $y_n = \overline{w}^T . \bar{x}(n)$
- error signal $e(n) = d(n) - y(n)$
- performance function $\xi = E[e^2(n)]$

Where auto correlation matrix of the filter input is

$$R = E[\bar{x}(n)x^{-T}(n)]\cdot$$

And $\bar{p} = E[\bar{x}(n)d(n)]$ cross-correlation vector between $\bar{x}(n)$ and *d(n).*

The channel characteristics are often not known and are found to vary with time for regenerating any signal that we want to receive. As a example, in case of circuit-switched telephone network, channel characteristics varies widely from call to call due to different switching paths while in the context of mobile communications, the channel characteristics changes significantly, within a call duration particularly because of the relative movements of the nearby reflectors (towers, buildings etc.) and also during handoff. The equalizer coefficients in such cases need to adapt themselves continuously to track the channel variations LMS algorithms are a class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal (difference between the desired and the actual signal). As the steepest descent to find filter weights *W(n),* which minimize a cost function. And cost function defined by

$$C(n) = E\{e(n)^2\} \qquad (7)$$

Where *e(n)* is the error at the current sample *'n'* and *E{.}* denotes the expected value. This cost function *(n)* is the mean square error, and it is minimized by the LMS.

Where μ is the step-size parameter and controls the convergence characteristics of the LMS algorithm; *e*(n)* is the mean square error between the output *y(n)* and the reference signal which is given by:

$$e^2(n) = [d^*(n) - w^h x(n)]^2 \quad (8)$$

Applying steepest descent means to take the partial derivatives with respect to the individual entries of the filter coefficient (weight) vector.

In the method of steepest descent the problem is the computation involved in finding the values r and R matrices in real time. The LMS algorithm on the other hand simplifies this by using the instantaneous values of covariance matrices r and R instead of their actual values i.e.:

$$R(n) = x(n)x^{h}(n)$$
$$r(n) = d^{*}(n)x(n)$$

Now, the cost function we need to take a step in the opposite direction of ∇C_n to express that in mathematical terms:

$$W(n+1) = W(n) - \frac{\mu}{2} E\{X(n)e^{*}(n)\} \quad (9)$$

Where $\mu/2$ is the step size (adaptation constant). That means we have found a sequential update algorithm which minimizes the cost function. Unfortunately, this algorithm is not realizable until we know $E\{X(n)e^{*}(n)\}$ .For most systems the expectation function $E\{X(n)e^{*}(n)\}$ must be approximated:

$$E\{X(n)e^{*}(n)\} = X(n)e^{*}(n) \quad (10)$$

The LMS algorithm is initiated with an arbitrary value *W(0)* for the weight vector at *n=0*. The successive corrections of the weight vector eventually leads to the minimum value of the mean squared error. Hence update algorithm follows as:

$$W(n+1) = W(n) + \mu X(n)e^{*}(n) \quad (11)$$

Therefore the weight update can be given by the following equation.

III. LMS (LEAST MEAN SQUARE) ARCHITECTURE FOR HARDWARE IMPLEMENTATION

A digital logic that computed for the SISO encoder with minimum ISI value, and converge the given value in original form. In block we shown mix a error signal $e^{*}(n)$, with the input signal $x(n)$, and (BTOS) signal. After update the filter coefficient with the help of step size in terms of LMS algorithm, logic that we use for the convergence:

$$W^{f}(n+1) = W^{f}(n) + \mu_a x(n)e^{*}(n)$$
$$W^{f}(n+1) = W^{b}(n) + \mu_b v(n)e^{*}(n)$$

IV. RESULTS

D. MATLAB simulation results

We are using DFE for simulation. For updating the filter coefficient we are using least mean square algorithm. Firstly we provide the random signal around 1000 sample of the data, as input signal to set the training mood. Once it gets primary input sample, After providing a staring tab vector, with the help of other component, step size, input signal. The next coefficient of filter got auto updated by the help of LMS algorithm easily.

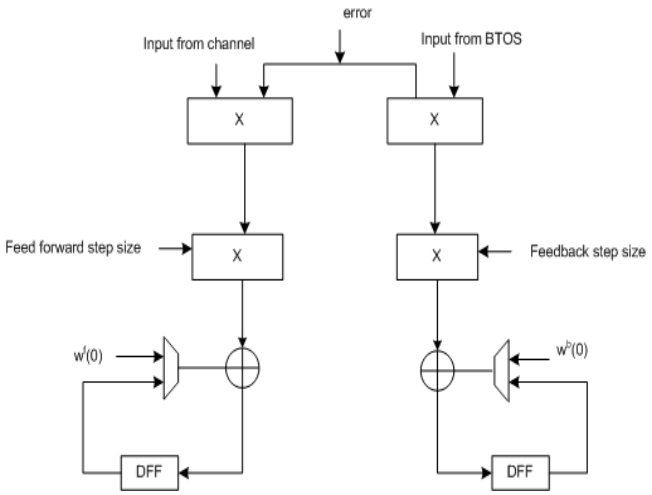

Figure 9. Least Mean Square Architecture

Decision Feedback Equalizer has designed using the RSC encoder for 4-QAM modulation scheme SNR =10. When DFE in training mode, from the results it is observed that Mean Square Error of the error output has convergence after 50 samples of data and the power of estimated symbol values varying between 0 and 1.

After using the 4-QAM modulation for SNR=10,

We can show in the Figure [10], initially the MSE is high then after around 50 sample, the MSE get settled. By the simulation results are shown, its clear soft input and the soft output get called by each other with minimum MSE, and ISI effect.

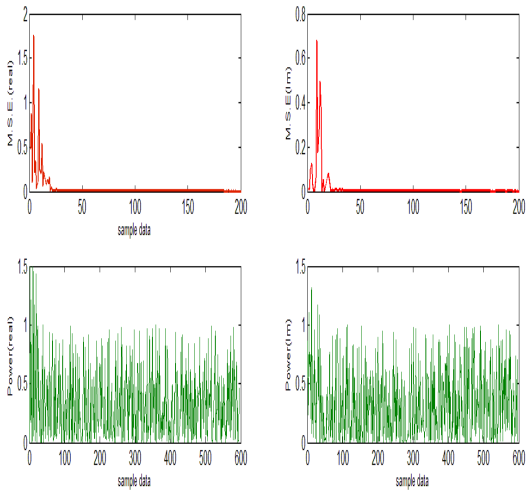

Figure 10. MATLAB Simulation results of DFE in training mode

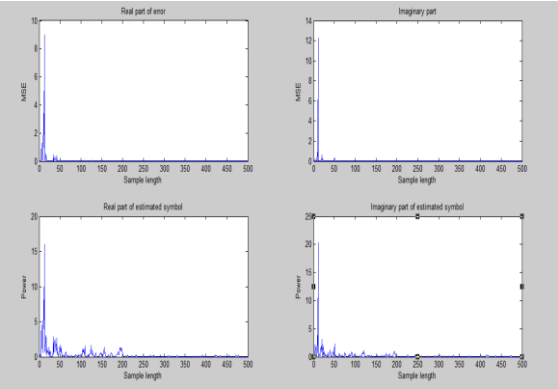

Figure 11. MATLAB Simulation results of DFE in tracking mode

V. CONCLUSION

SISO equalizer has been implemented using Least mean square (LMS) algorithm. DFE (decision feedback equalizer) has been implemented in MATLAB and it has been observed that MSE (mean square error) converges to zero. We explore architectural approaches for energy-efficient linear turbo equalization by eliminating unnecessary

REFERENCES

[1] R G Sung, vldt , ―VLSI design for Turbo Equalizer. Berkeley,2012 .

[2] Ch. S Kumar, D. Madhavi, K .V Reddy, "Adaptive Decision feedback Equalizer", International Conference on computational intelligence and computing research (ICCIC), 28-29 Dec 2010.

[3] Seok-Jun Lee, Andrew C. Singer, Member and Naresh R. Shanbhag, "Analysis of Linear Turbo Equalizer via EXIT Chart", Global Telecommunications Conference IEEE, 2003.

[4] C. E. Shannon, "A mathematical theory of communications-I & II," Bell Syst. Tech. J.,vol.27, pp. 37A. S. Barbulescu, S. S. Pietrobon, "Interleaver design for turbo codes," Electronic Letters, vol. 30, pp. 2107–2108, Dec. 1994.

[5] B. Vucetic and J. Yuan, "Turbo Codes: Principles and Applications", Kluwer Academic Publishers, 2000.

[6] J.G. Proakis, "Digital Communications", 3rd ed., McGraw-Hill,1995

[7] Qureshi, S. U. H., \Adaptive Equalization,Proc. IEEE, vol. 73, no. 9, pp. 1349- 1387, Sept. 1985.

[8] S.J LEE N.R.Shanbhag and Andrew c. Singer , Energy Efficient VLSI Architecture for Linear turbo Equalizer Journal of VLSI Signal Processing 39, 49–62, 2005