A Decision-Aided Adaptive Equalizer with Simplified Implentation

Kuen-Tsair Lay, Hen-Wai Tsao, Lin-shan Lee

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan

Abstract

Digital transmission over telephone channel may suffer from intersymbol interference (ISI) because of the serious distortion in the available bandwidth. One approach to solve this problem is to use adaptive equalizers. In this paper, a new design for adaptive equalizers is proposed, in which the computational complexity involved is significantly less than that of conventional adaptive equalizers because almost all the multiplication operations involved in the latter are replaced by addition operations in the former. The key point is that by taking intermediate decisions from the received signal and using these decisions with finite possible values rather than the received signal samples to calculate the equalized signal and to adjust the equalizer tap coefficients, multiplications can effectively be replaced by additions.

I. INTRODUCTION

Digital transmission over telephone channel may suffer from intersymbol inter-ference due to the nonideal characteristics in the available bandwidth. For medium- or high-rate transmission, the restricted bandwidth and the undesired characteristics of frequency response result in severe distortion of the signal. Conventionally, equalization is used to deal with this kind distortion. Since the characteristics of a telephone channel can not be determined before it is used for transmission, no specific form of equalization can be devised in advance. As a consequence, adaptive schemes of equalizers must be taken into account (1,2,3,7). However, most conventional schemes of digital adaptive equalizers involve very large number of multiplication operations so that the computation complexity very often makes the real time implemention relatively expensive. This is why in this paper we propose a new method to perform the adaptive equalization. The special feature of the new method is that because some intermediate decisions of the data (1's or 0's) instead of the true value of the signals (real numbers) are used in the computation, many multiplication operations in traditional adaptive equalizers are replaced by additions, and simplified implementation is therefore possible.

II. System Model

A digital communication system can be represented by the function blocks in Fig.1. For simplicity, we will assume the modulation to be BPSK here, and similar results can be easily extended to other cases. The output of the datasonrce B1 is therefore +1 or -1 representing 1's and 0's, and the cascade of B2, B3, B4, B5, B6 in Fig.1 can be considered as a discrete-time linear system and can be described in terms of its unity response h_i's,

$$h_i = h(D_0 T + \tau + iT) \qquad (1)$$

where h(t) is the impulse response of the cascade of the blocks B2 to B5, and T is the bit duration, D_0 is a delay factor which makes h_0 the peak value of h_i's, and τ is the sampling time offset from t=0. The above model can be summarized in Fig.2. If the data source sends symbols x_n, then the output y_n of the sampler B6 is in the following equation, where v_n is the noise sample,

$$y_n = \sum_{j=-N}^{M} x_{n-j}\, h_j + v_n \qquad (2)$$

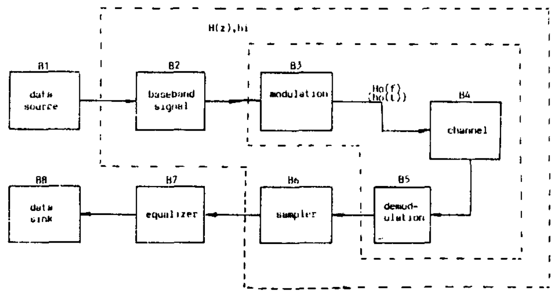

Fig.1 A Digital Communication System

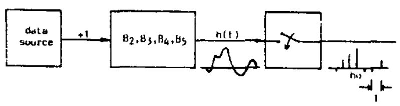

Fig.2. Unity response

where M and N are integers above which hi's are negligible. Therefore, a digital trans-mission system can be modeled as a digital filter represented by eq.(2) and shown in Fig.3.

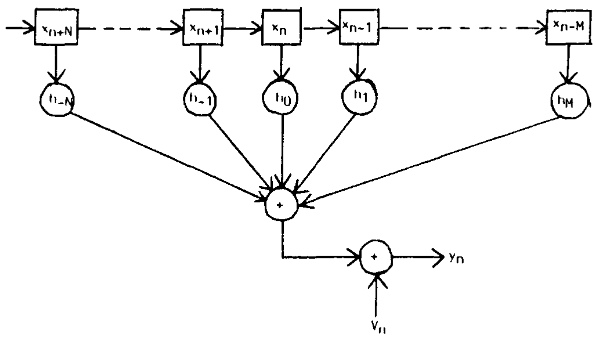

Fig.3. The System Model

III. The Operation of the Decision-Aided Adaptive Equalizer

CH2538-7/88/0000-1385 $1.00 © 1988 IEEE

Eq.(2) in the previous section can be rewritten as

$$y_n = h_0 x_n + \sum_{j=-N}^{M}{}' x_{n-j} h_j + v_n \qquad (3)$$

where the prime at the upper-right corner of $\sum$ means "excluding the term j=n". The third term on the right of eq.(3) is the noise term, which is usually negligible because of the high SNR (typically higher than 20 dB) for transmission over telephone channels. The second term on the right of eq.(3) is the ISI term, denoted by η,

$$\eta = \sum_{j=-N}^{M}{}' x_{n-j} h_j \qquad (4)$$

Our goal is to eliminate η. The prototype of the new decision-aided adoptive equalizer proposed in this paper is shown in Fig.4. In Fig.4, y_n is the demodulator output signal samples, u_n is the intermediate decision results (either immediate decisions or tentative decisions, as will be clear later), being either +1 or -1, c_i is the tap coefficient, z_n is the equalized signal. (N_c+1+M_c) is the order of the equalizer. Note that because all the u_n's are either +1 or -1, most of the multiplication operations can thus be replaced by additions. Later we will show that we can obtain a set of c_i's according to specific algorithms such that can be substracted from y_n on the condition that most decisions u_n's are correct.

The operation of this decision-aided adaptive equalizer is divided into 2 parts. First, a training process is required to obtain the proper c_i's. Then, the actual data can be sent. The training process is further divided into 2 phases; the taps are adjusted according to a sequence of two different algorithms. Summarizing, the equalizer operation is divided into 3 modes designated as TRAIN1, TRAIN2, and DATA, respectively.

IV. Tap Adjustment Algorithms

During training, what are transmitted from the transmitter is completely known in the receiving end. That is, x_i's in eq.(2) are known in advance. The effective structure of the decision-aided adaptive equalizer in TRAIN mode is shown in Fig.5. The symbols used in

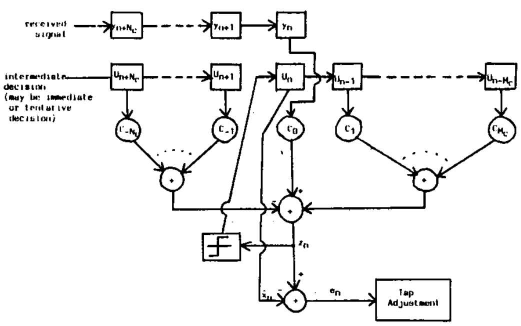

Fig.4 Prototype of decision-aided adaptive equalizer

Fig.5 are the same as in Fig.4; e_n is the error signal. Refer to Fig.5, we have the equalized signal in TRAIN1 and TRAIN2 respectively shown below:

$$\text{in TRAIN1:} \quad z_n = C_0 y_n \qquad (5)$$

$$\text{in TRAIN2:} \quad z_n = C_0 y_n - \sum_{j=-N_c}^{M_c} C_j X_{n-j} \qquad (6)$$

Note that to obtain equation (6), we make use of the fact that u_n is exactly equal to x_n. In both cases (i.e. TRAIN1 and TRAIN2) the error signal is:

$$e_n \equiv z_n - x_n \qquad (7)$$

The algorithm for tap adjustment in TRAIN1 is shown below:

1. Initially, set $C_0 \leftarrow 1$; $C_i \leftarrow 0$ for $i \neq 0$ (8)
2. $C_0(n) \leftarrow C_0(n-1) - \delta_1 \cdot e_n \cdot x_n$, $n=1,2,\ldots L_1$ (9)
3. $C_i(n) \leftarrow C_i(n-1) + \delta_1 \cdot e_n \cdot x_{n-i}$, $i \neq 0$, $n=1,2,\ldots L_1$ (10)

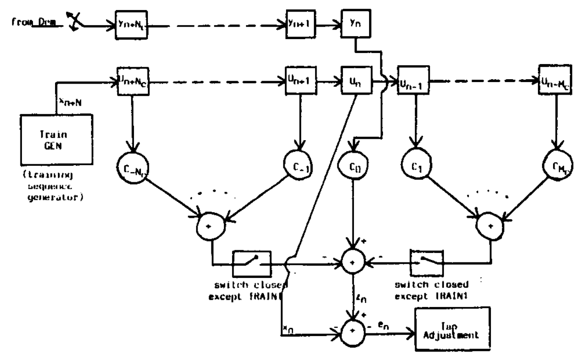

Fig.5. Effective structure of decision-aided adaptive equalizer in TRAIN

where δ_1 is the step size in TRAIN1; $C_i(n)$ is the coefficient c_i at intration n. L_1 is the number of iterations performed in TRAIN1.

The algorithm for tap adjustment in TRAIN2 is:

1. $C_0(n) \leftarrow C_0(n-1) - \delta_2 \cdot e_n \cdot x_n$, $n=L_1+1, L_1+2, \ldots, L_1+L_2$ (11)
2. $C_i(n) \leftarrow C_i(n-1) - \delta_2 \cdot e_n \cdot x_{n-i}$, $i=0$, $n=L_1+1, L_1+2, \ldots, L_1+L_2$ (12)

where δ_2 is the step size used in TRAIN2, L_2 is the number of iterations performed in TRAIN2. In the appendix, it is shown that at steady state (i.e. through large number of iterations) in TRAIN2, C_0, C_i's will converge to specific values,

$$\begin{cases} C_0 = 1/h_0 \\ C_i = h_i/h_0 \end{cases} \qquad (13)$$

and the equalized signal (z_n) will be very close to the desired signal (x_n).

In computer simulation we found that the tap coefficients change more quickly in TRAIN1 than in TRAIN2, but the resident excess mean square error is larger in the former. Conceptually, the operation of TRAIN1 is to quickly set up a starting point for adjustment in TRAIN2, while the operation of TRAIN2 is to finely adjust the tap coefficients to optimal values so that the equaliz-ed signal is very close to the actually desired signal. An example of the process of tap adjustment in training modes is shown in Fig.6.

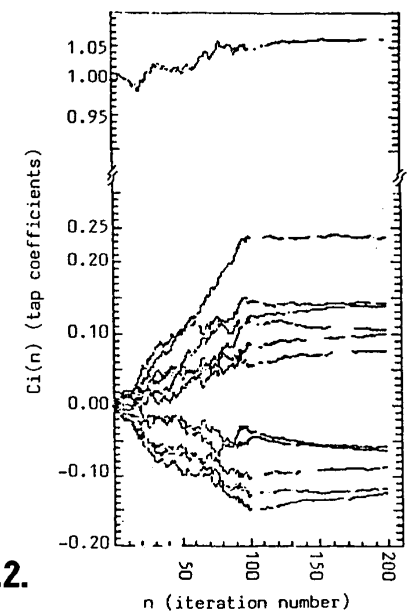

Fig.6. Tracing tap coefficients during TRAIN

Compared to conventional equalizers [4,5,6], this new scheme has two distinctive features. (1) It is u_n's, which are equal to x_n's during TRAIN mode, rather than y_n's that are used to calculate z_n. Since u_n's can have only 2 values (either +1 or -1), the operation $u_{n-i}\ c_n$ can be easily implemented using accumulation rather than multiplication. (2) Special tap adjustment algorithms are proposed, in which many multiplications for adjusting taps in conventional equalizers are similarly replaced by accumulations.

V. Tap Adjustment During DATA Mode

In DATA mode, the actual data are transmitted. Now the data sequence from the transmitter is no longer known by the receiver in advance. Immediate decisions or tentative decisions thus must be made to aid the equalizer to make the final decision and to adjust the tap coefficients. This can be achieved by two stages as demonstrated in Figs.7-9 when one stage is not good enough. The term "immediate decision" represents the result of directly hard-limiting the received signal (i.e. y_n); in other words, immediate decision is the directly received information bit in the receiver without any equalization. The immediate decisions ξ_{0n} can then be used to calculate an tentatively equalized signal z_{n+N} and therefore the tentative decision ξ_{1n} as shown in Fig.7. The bit error rate for immediate decision (ξ_{0n}) is relatively high; but as long as it is on the order of 10^{-2} or less, most of the ISI terms can be substracted and the tentatively equalized signal z_{n+N} is usually closer to x_{n+N} than y_{n+N} (refer to Fig. 7). The bit error rate of the tentative decisions $_{1n}$ should therefore be significantly reduced as compared to the bit error rate of the immediate

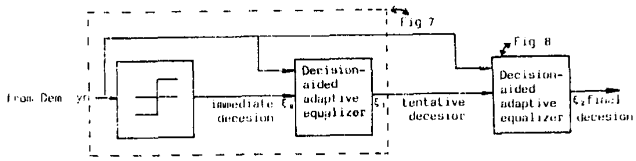

Fig.9. Decisions used in decision-aided adaptive equalizer.

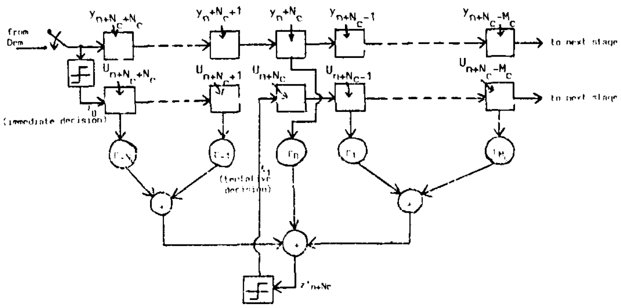

Fig.7. Making tentative decision ξ_{1n} based on immediate decision $_{0n}$.

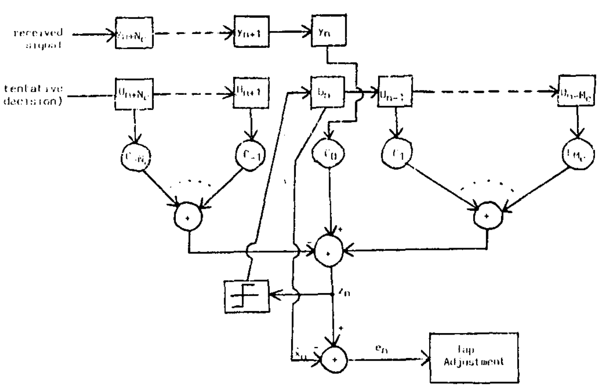

Fig.8. Making final decision ξ_{2n} based on tentative decisions ξ_{1n}

decision ξ_{0n}. When the bit error rate for $_{1n}$ is still too high, a second-stage equalization as shown in Fig.8 is required, in which the tentative decisions ξ_{1n} as obtained in the first stage in Fig.7 can be used to calculate the finally equalized signal z_n and the final decisions ξ_{2n}. The complete immediate-tentative-final decision scheme is summarized in Fig.9, which is a cascade of the two stages in Figs.7 and Fig. 8.

After training, the tap coefficients are close to the optimal values (in the sense of meeting eq.(13)). If the channel is time varying during data transmission, the tap adjustment can continue in DATA mode (as shown in Fig.8). The tap adjustment in DATA is the same as that in TRAIN2 except that decision x_n is used instead of the actual transmitted symbol $\hat{x}_n$. The algorithm for tap adjustment in DATA is:

1. $c_0(n) \leftarrow c_0(n-1)\ -\delta_3 \cdot e_n \cdot \hat{x}_n$ (14)
2. $c_i(n) \leftarrow c_i(n-1) + \delta_3 \cdot e_n \cdot \hat{x}_{n-i}\ ,\ i \neq 0$ (15)

where δ_3 is the step size in DATA mode.

Since the bit error rate for equalized decision (x_n) is usually very small, i.e., $\hat{x}_n = x_n$ for most of the time, there is little degradation in the performance.

VI. Performance Evaluation

Computer simulations have been used to calculate the BER (bit error rate) for some BPSK examples. We will see that the coefficient convergence is slower for decision-aided adaptive equalizers. This is the price which has to be paid for the reduced computation complexity. That is, a longer period of training may be required, but once it is trained, the low comput-ation complexity allows it to equalize very quickly and efficiently.

For transmission over telephone channel, SNR is usually high (typically above 20dB), therefore we neglect the noise effect. BER0 is the BER for decision if no equalization is used at all, i.e., BER0 is the BER for immediate decision. BER1 is the BER for tentative decision, and BER2 is for final decision.

Since the channel is binary symmetrical,

$$BER0 = Prob(y_n<0|x_n=+1) \quad (16)$$

Given $x_n=+1$, y_n depends on $\{x_{n-i},\ i=-N,\ldots -1,1,\ldots M\}$ (refer to eq.(3)). For some combinations of x_{n-i}'s, y_n will be less than 0, which makes an error. The number of combinations of x_{n-i}'s which make y_n less than 0 divided by the number of all possible combinations is the BER0 (recall that we neglect noise). Similar to eq(16), we have eq.(17) and eq.(18), where z_n' is the equalized signal for tentative decision and z_n is the equalized signal for final decision.

$$BER1 = Prob(z_n' < 0 \mid x_n = +1) \quad (17)$$

$$BER2 = Prob(z_n < 0 \mid x_n = +1) \quad (18)$$

Table.1 BER0, BER1, BER2 and L_0 for some examples.

Unity response	taps at the end of TRAIN1	taps at the end of TRAIN2
h(-8) = 0.003	* δ_2= 0.01	* δ_2= 0.01
h(-7) = -0.012	* L = 100	* L = 100
h(-6) = -0.012		
h(-5) = -0.068	c(-5) = -0.048	c(-5) = -0.064
h(-4) = 0.090	c(-4) = 0.111	c(-4) = 0.104
h(-3) = -0.109	c(-3) = -0.146	c(-3) = -0.125
h(-2) = -0.110	c(-2) = -0.124	c(-2) = -0.117
h(-1) = 0.219	c(-1) = 0.232	c(-1) = 0.237
h(0) = 0.943	c(0) = 1.067	c(0) = 1.069
h(1) = 0.143	c(1) = 0.124	c(1) = 0.141
h(2) = 0.132	c(2) = 0.143	c(2) = 0.143
h(3) = 0.084	c(3) = 0.052	c(3) = 0.076
h(4) = 0.099	c(4) = 0.081	c(4) = 0.099
h(5) = -0.070	c(5) = -0.033	c(5) = -0.057
h(6) = -0.070	c(6) = -0.103	c(6) = -0.088
h(7) = -0.030		
h(8) = 0.001		

BER0 = 4.944 E-3
BER1 = 6.957 E-5
BER2 = 9.608 E-7
L_0 = 70

Table 1.1

Unity response	taps at the end of TRAIN1	taps at the end of TRAIN2
h(-8) = 0.010	* δ_1= 0.01	* δ_2= 0.01
h(-7) = -0.011	* L = 100	* L = 100
h(-6) = -0.037		
h(-5) = -0.076	c(-5) = -0.070	c(-5) = -0.072
h(-4) = 0.098	c(-4) = 0.122	c(-4) = 0.099
h(-3) = -0.121	c(-3) = -0.151	c(-3) = -0.123
h(-2) = -0.113	c(-2) = -0.136	c(-2) = -0.111
h(-1) = 0.219	c(-1) = 0.217	c(-1) = 0.209
h(0) = 1.102	c(0) = 0.907	c(0) = 0.915
h(1) = 0.213	c(1) = 0.176	c(1) = 0.183
h(2) = 0.137	c(2) = 0.140	c(2) = 0.130
h(3) = 0.112	c(3) = 0.079	c(3) = 0.092
h(4) = 0.099	c(4) = 0.086	c(4) = 0.092
h(5) = -0.070	c(5) = -0.055	c(5) = -0.058
h(6) = -0.050	c(6) = -0.078	c(6) = -0.059
h(7) = -0.073		
h(8) = 0.001		

BER0 = 3.525 E-3
BER1 = 2.453 E-5
BER2 = 1.662 E-7
L_0 = 45

Table 1.2

unity response	taps at the end of TRAIN1	taps at the end of TRAIN2
h(-8) = 0.010	* δ_1= 0.01	* δ_2 = 0.01
h(-7) = -0.010	* L = 100	* L = 100
h(-6) = -0.037		
h(-5) = -0.076	c(-5) = -0.062	c(-5) = -0.075
h(-4) = 0.089	c(-4) = 0.103	c(-4) = 0.100
h(-3) = -0.110	c(-3) = -0.145	c(-3) = -0.122
h(-2) = -0.111	c(-2) = -0.128	c(-2) = -0.116
h(-1) = 0.202	c(-1) = 0.211	c(-1) = 0.216
h(0) = 0.984	c(0) = 1.033	c(0) = 1.026
h(1) = 0.201	c(1) = 0.181	c(1) = 0.195
h(2) = 0.137	c(2) = 0.138	c(2) = 0.142
h(3) = 0.101	c(3) = 0.065	c(3) = 0.091
h(4) = 0.095	c(4) = 0.079	c(4) = 0.092
h(5) = -0.037	c(5) = -0.004	c(5) = -0.025
h(6) = -0.050	c(6) = -0.090	c(6) = -0.065
h(7) = -0.007		
h(8) = 0.001		

BER0 = 4.791 E-3
BER1 = 1.174 E-6
BER2 = 7.044 E-14
L_0 = 55

Table 1.3

To compare the speed of convergence, define L_0 to be the number of iterations required for the conventional linear transversal equalizer to have the BER below some given threshold (in the examples here, less than 10^{-10}). Some typical results for many examples simulated are shown in Table 1.

In the examples, L_1 = 100, L_2 = 100; hence the iterations required for TRAIN is 200. Compared to L_0 which are 70, 45, 55, respectively, we see that the convergence is apparently slower in the decision-aided adaptive equalizer. This is intuitively correct because in the new method we are not adjusting the coefficients according to the true gradient of the errors; instead, we are making use of some kind of "modified" or "approximated" gradient. Therefore the fastest converging can not be achieved. There is a trade-off between the computational complexity and the converging speed. From the values for BER0, BER1, BER2 we see that bit error rate is drastically reduced after each stage of equalization. Also, if a BER signiticantly lower than that of 2 (i.e. BER2) is needed, a third stage of equalization can be exploited to furthur reduce the bit error rate.

VI. Conclusion

We have proposed a new structure for adaptive equalization in which with the aid of the immediate and tentative decisions many multiplication operations required in conventional adaptive equalizers are replaced by addition operations. This decision-aided adaptive equalizer is therefore much more easier to implement. All the discussions here are based on BPSK modulation, but they can be easily extended to other modulation schemes. For example, there is only one thing to be modified for QPSK —— complex values instead of real quantities in signal and tap coefficients must be used.

Appendix

Here we will show that at the end of TRAIN2 mode, the tap coefficients should converge to a specific set of values, which make the equalized signal (z_n) equal to (more precisely, very close to) the desired signal (x_n). Substituting eq.(2) into eqs.(6) and (7) and noting that $x_nx_n = 1$, we have

$$e_n=(h_0c_0(n-1)-1)x_n+\sum_{j=-N_c}^{M_c}{}' (h_jc_0(n-1)-c_j(n-1))x_{n-j} + \sum_{\substack{-N\le j<-N_c \\ M\ge j>M_c}} h_jc_0(n-1)x_{n-j} + v_nc_0(n-1) \quad (A.1)$$

Substituting eq.(A.1) into eq.(11),

$$c_0(n) = c_0(n-1) - \delta_2[(h_0c_0(n-1)-1)x_n + \sum_{j=-N_c}^{M_c}{}' (h_jc_0(n-1)-c_j(n-1))x_{n-j} + \sum_{\substack{-N\le j<-N_c \\ M\ge j>M_c}} h_jc_0(n-1)\, x_{n-j} + v_nc_0(n-1)]x_n \quad (A.2)$$

Define

$$\psi_0(n) = -\delta_2 \sum_{j=-N_c}^{M_c}{}' (h_jc_0(n-1)-c_j(n-1))x_{n-j}x_{n-} - \delta_2 \sum_{\substack{-N\le j<-N_c \\ M\ge j>M_c}} h_j\, c_0(n-1)x_{n-j}\, x_n + \delta_2\, v_n\cdot c_0(n-1)x_n \quad (A.3)$$

which is a disturbing term for adjusting c_0 at intration n. $\psi_0(n)$ is a random variable, with mean and variance as follows, where σ_n is the variance of noise

$$\text{mean}\ (\psi_0(n)) = 0$$

variance

$$(\psi_0(n)) = \delta_2^2[\sum_{j=-N_c}^{M_c}{}' (h_j c_0(n-1) - c_j(n-1))^2$$

$$+ \sum_{\substack{-N\le j<-N_c \\ M\ge j>M_c}} (h_j c_0(n-1))^2 + \sigma_n^2] \qquad (A.4)$$

At steady state,

$$c_0(n) = c_0(n-1) \triangleq c_0 \qquad (A.5)$$

$$c_j(n) = c_j(n-1) = c_j$$

Substituting eq(A.5) into eq.(A.2) and neglecting the disturbing term $\psi_0(n)$, we have

$$c_0 = c_0 + \delta_2 (1 - h_0 c_0)$$

therefore, $c_0 = 1/h_0$ (A.6)

Similarily, we can find c_j's at steady state. Substituting eq.(A.1) into eq.(13) and noting that $x_{n-j} x_{n-j} = 1$,

$$c_i(n) =$$

$$c_i(n-1) + \delta_2(h_i c_0(n-1) - c_i(n-1)) + \delta_2\{(h_0 c_0(n-1) - 1)x_n x_{n-i}$$

$$+ \sum_{\substack{j=-N_c \\ j\ne i}}^{M_c}{}' (h_j c_0(n-1) - c_j(n-1)) x_{n-j} x_{n-i}$$

$$+ \sum_{\substack{-N\le j<-N_c \\ M\ge j>M_c}} h_j c_0(n-1) x_{n-j} x_{n-i} + v_n c_0(n-1) x_{n-i}\} \qquad (A.7)$$

Define

$$\psi_i(n) = \delta_2 \cdot (h_0 c_0(n-1) - 1) x_n x_{n-i} + \sum_{\substack{j=-N_c \\ j\ne i}}^{M_c}{}' (h_j c_0(n-1)$$

$$- c_j(n-1)) x_{n-j} x_{n-i}) + \sum_{\substack{-N\le j<-N_c \\ M\ge j>M_c}} h_j c_0(n-1) x_{n-j} x_{n-i}$$

$$+ v_n c_0(n-1) x_{n-i}\} \qquad (A.8)$$

which is a disturbing term for adjusting c_i at iteration n. $\psi_i(n)$ is a random variable with mean and variance shown as follows

mean $(\psi_i(n)) = 0$

variance $(\psi_i(n)) =$

$$\delta_2^2 \{(h_0 c_0(n-1) - 1)^2 + \sum_{\substack{j=-N_c \\ j\ne i}}^{M_c}{}' (h_j c_0(n-1) - c_j(n-1))^2$$

$$+ \sum_{\substack{-N\le j<-N_c \\ M\ge j>M_c}} (h_j c_0(n-1))^2 + \delta_n^2 \cdot c_0^2(n-1)\} \qquad (A.9)$$

Substituting eq.(A.5) into eq.(A.7) and reglecting the disturbing term $\psi_i(n)$, we have

$$c_i = c_i + \delta_2 \cdot (h_i c_0 - c_i) \qquad (A.10)$$

therefore, $c_i = h_i \cdot c_0$

Substitating eq.(A.6) into eq.(A.10), we have

$$c_i = h_i / h_0 \qquad (A.11)$$

we therefore have eq.(13),

$$c_0 = 1/h_0 \qquad (14)$$

$$c_i = h_i / h_0$$

We can now show that once eq.(13) is achieved, we have the equalized signal very close to the desired signal. Recite eq.(2) and ref to Fig. 3,

$$y_n = \sum_{j=-N}^{M} x_{n-j} h_j + v_n \qquad (2)$$

we thus have

$$z_n = y_n \cdot c_0 - \sum_{j=-N_c}^{M_c}{}' c_j x_{n-j}$$

$$= (\sum_{j=-N}^{M} h_j x_{n-j} + v_n) \cdot c_0 - \sum_{j=-N_c}^{M_c}{}' c_j x_{n-h=j}$$

$$= c_0 h_0 x_n + \sum_{j=-N_c}^{M_c}{}' (h_j c_0 - c_j) x_{n-j}$$

$$+ \sum_{\substack{-N\le j<-N_c \\ M\ge j>M_c}} h_j x_{n-j} c_0 + v_n \qquad (A.12)$$

In eq.(A.12), v_n is the noise term; the third term is the uncancelled ISI term (due to the insufficient stages of equalizer). The noise term can be neglected when SNR is large enough. If N_c and M_c are large enough, the uncancelled ISI term can also be neglected. Noting that from eq.(13) we have $c_0 h_0 = 1$ and $h_j = c_j$, eq.(A.12) can be rewritten as:

$$z_n = c_0 h_0 x_n + \sum_{j=-N_c}^{M_c}{}' (h_j c_0 - c_j) x_{n-j} = x_n \qquad (A.13)$$

therefore the equalized signal is approximately equal to the desired signal.

Reference

[1] Qureshi, S.U.H., "Adaptive Equalization," Proc. IEEE Vol.73, No.9, pp.1349-1387 (1985)

[2] Gersho, A. and Lim, T.L., "Adaptive Cancellation of Inter- symbol Interference for Data Transmission," BSTJ, Vol.60, No.11, pp.1997-2021 (1981)

[3] Lucky, R.W. and Rudin, H.R., "An Automatic Equalizer for General-Purpose Communication Channels," BSTJ, Vol.46, No.9, pp.2179-2208 (1967).

[4] Ferrara, E. R., "Fast Implementation of LMS Adaptive Filters," IEEE Tran. on ASSP, Vol.28, No.4, pp.474-475 (1980).

[5] Widrow, B., McCool, J. and Ball, M., "The Complex LMS Algorithm," Proc. IEEE pp.719-720 (1975).

[6] Giordano, A.A. and Hsu, F.M., Least Square Estimation with Applications to Digital Signal Processing, Chap. 6 (1985).

[7] Proakis, J.G. "Adaptive nonlinear filtering techniques for data transmission," IEEE Symp. on Adaptive Processes, Decision and Control, pp.XV.2.1-5 (1970).