Global Journal of Researches in Engineering: F
Electrical and Electronics Engineering
Volume 14 Issue 7 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 2249-4596 & Print ISSN: 0975-5861

Adaptive Filters

By Dr. Ziad Sobih & Prof. Martin
Northeastern University, Jordan

Abstract- We know the optimum and suboptimum receiver for ISI in the transmission through band-limited non ideal channels. The optimum employ maximum likelihood detection. The sub optimum employ linear equalizer.

In our design of the equalizer we assume that we know at the receiver the impulse response of the channel or the frequency response. In most communication channels this response is varying as a function of time. In this case we use an equalizer to adapt for this change. In this paper we will automatically adjust the equalizer coefficient to adapt for channel changes. We also analyze the performance characteristics including convergence rate and computational complexity.

GJRE-F Classification : FOR Code: 090699

Adaptive Filters

Dr. Ziad Sobih [α] & Prof. Martin [σ]

Abstract- We know the optimum and suboptimum receiver for ISI in the transmission through band-limited non ideal channels. The optimum employ maximum likelihood detection. The sub optimum employ linear equalizer.

In our design of the equalizer we assume that we know at the receiver the impulse response of the channel or the frequency response. In most communication channels this response is varying as a function of time. In this case we use an equalizer to adapt for this change.

In this paper we will automatically adjust the equalizer coefficient to adapt for channel changes. We also analyze the performance characteristics including convergence rate and computational complexity.

I. Adaptive Linear Equalizer

In the case of linear equalizer we will consider two methods to adjust the coefficient values ck. One method is to minimize the peak distortion at the output. The second method is to minimize the mean square error at the output. Now we will talk about each method.

II. The Zero-Forcing Algorithm

The peak distortion D(**c**) is minimized by selecting the appropriate **c**k. In general it is not easy computation. Except for the case D0 < 1. In this case D(**c**) is minimized by forcing qn = 0 and q0 = 1. We will use the zero forcing algorithm to achieve this condition.

The zero forcing algorithm is achieved by equating the cross correlation between the error and the desired sequence to zero. We have

$$\begin{aligned} E(\varepsilon_k I_{k-j}^*) &= E\left[(I_k - I_k)I_{k-j}^*\right] \\ &= E(I_k I_{k-j}^*) - E(\hat{I}_k I_{k-j}^*), \qquad j = -K, \ldots, K \end{aligned} \tag{1}$$

We assume that information symbols are uncorrelated. And that it is uncorrelated with the estimate

$$E(\varepsilon_k I_{k-j}^*) = \delta_{j0} - q_j, \qquad j = -K, \ldots, K \tag{2}$$

$$E(\varepsilon_k I_{k-j}^*) = 0, \qquad j = -K, \ldots, K \tag{3}$$

This mean that q0=1 and qn=0 for n not equal to zero.

When the channel response is not known the cross correlation in equation 1 can not be found. To circumvent this problem we send a known training sequence Ik to the receiver. This way we can estimate the cross correlation by using time averages. After we send this training sequence for samples length equal or more than equalizer length we can find the equalizer coefficient that satisfy equation 3.

A simple recursive algorithm

$$c_j^{(k+1)} = c_j^{(k)} + \Delta\varepsilon_k I_{k-j}^*, \qquad j = -K, \ldots, -1, 0, 1, \ldots, K \tag{4}$$

Were c j k is the j th coefficient at time k T, e k is the error at time k T, and delta is a scaling factor for rate adjustment. This is the zero forcing algorithm. The term e(k)I(k-j) is the cross correlation estimate of E (e(k) I(j-k)). This is accomplished by recursive first order difference equation which is a simple discrete time integrator. This is due to second fundamental theory of calculus.

Author α: *Northeastern University, Boston, MA, Jordan.*
e-mail: sobih84@gmail.com
Author σ: *MIT, Boston, MA, Jordan.*

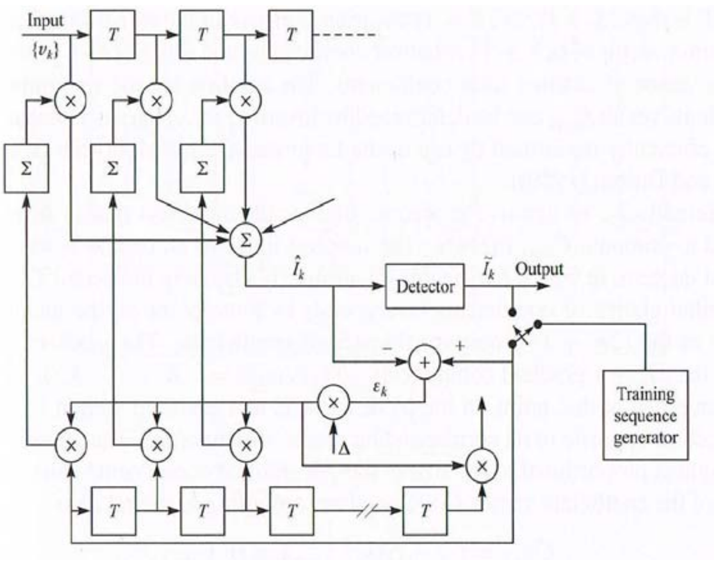

Figure 1

When we use a training sequence we are in the training mode. We do this until the coefficient of the equalizer converge to the optimal. At this point the decision of the output is sufficiently reliable. Now we go to the adaptive mode were we use the estimated output to continue adjusting the coefficients. In this case the cross correlation equation to update c is

$$c_i^{(k+1)} = c_i^{(k)} + \Delta \tilde{\varepsilon}_k \tilde{I}^*_{k-i} \quad (5)$$

This is similar to least mean square LMS algorithm. We will talk about MSE in detail next.

III. The lms Algorithm

First we calculate the error. We find that to minimize the error C has to satisfy the set of linear equations.

$$\Gamma C = \xi \quad (6)$$

Were T is the covariance metric of the input v k. C is the quantizer coefficients and E is the input times d the desired response.

An iterative algorithm can be used that avoid metric inversion to calculate C opt. the simplest is the method of steepest decent. In which we choose any C to start and it will converge by time to C opt. our start C will correspond to some point on the MSE surface curve. The gradient G is computed at this point. Using this gradient we move to the next estimate closer to the minimum.

$$C_{k+1} = C_k - \Delta G_k, \qquad k = 0, 1, 2, \ldots \quad (7)$$

The gradient is

$$G_k = \frac{1}{2}\frac{dJ}{dC_k} = \Gamma C_k - \xi = -E\left(\varepsilon_k V_k^*\right) \quad (8)$$

Were Ck is the coefficients at the k iteration. E k is the error at k iteration. V is the vector of the input. Delta is a small positive number that will ensure convergence of the iterative process. If the minimum MSE is reached at some k then Gk=0 so that Ck stay constant. This method is slow but it is good to explain things.

The basic difficulty of this method is the lack of knowledge of the gradient vector G. G depend on the covariance metric T and the vector E of cross correlation. These depend on the coefficient of the equivalent discrete channel model in which the receiver do not know. To overcome this difficulty an estimate of G is used. The coefficient can be found using this estimate as

$$\hat{C}_{k+1} = \hat{C}_k - \Delta \hat{G}_k \quad (9)$$

Were we have an estimate of the gradient and an estimate of the coefficients

We find that

$$\hat{G}_k = -\varepsilon_k V_k^* \quad (10)$$

The equation is

$$\hat{C}_{k+1} = \hat{C}_k + \Delta \varepsilon_k V_k^* \quad (11)$$

This is the basic LMS algorithm for recursively adjusting the weight coefficients described by Widrow (1966). It is figure 2

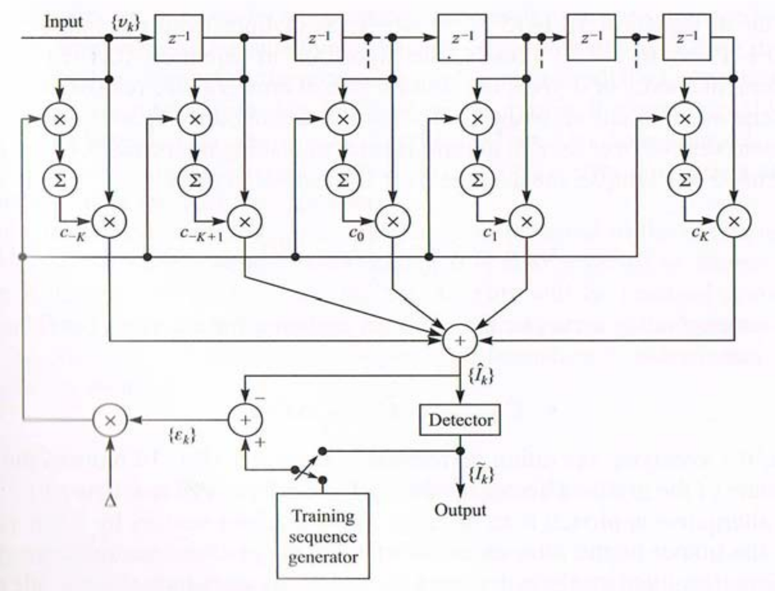

Figure 2

Equation 11 has been used in commercial adaptive equalizers that are used in high speed modems. Three versions are used using only the sign information.

$$c_{(k+1)j} = c_{kj} + \Delta\,\mathrm{csgn}(\varepsilon_k)v^*_{k-j}, \qquad j = -K,\ldots,-1,0,1,\ldots,K$$

$$c_{(k+1)j} = c_{kj} + \Delta\varepsilon_k\,\mathrm{csgn}\left(v^*_{k-j}\right), \qquad j = -K,\ldots,-1,0,1,\ldots,K$$

$$c_{(k+1)j} = c_{kj} + \Delta\,\mathrm{csgn}(\varepsilon_k)\,\mathrm{csgn}\left(v^*_{k-j}\right), \quad j = -K,\ldots,-1,0,1,\ldots,K$$

where csgn(x) is defined as (12-13-14-15)

$$\mathrm{csgn}(x) = \begin{cases} 1+j & [\mathrm{Re}(x) > 0, \mathrm{Im}(x) > 0] \\ 1-j & [\mathrm{Re}(x) > 0, \mathrm{Im}(x) < 0] \\ -1+j & [\mathrm{Re}(x) < 0, \mathrm{Im}(x) > 0] \\ -1-j & [\mathrm{Re}(x) < 0, \mathrm{Im}(x) < 0] \end{cases}$$

Equation 14 is the most easy to implement. It has the slowest convergence relative to the others.

Several other LMS algorithms are obtained by averaging the gradient vector over several iteration points before we adjust c.

$$\bar{\hat{G}}_{mN} = -\frac{1}{N}\sum_{n=0}^{N-1}\varepsilon_{mN+n}V^*_{mN+n} \tag{16}$$

now the equation is

$$\hat{C}_{(k+1)N} = \hat{C}_{kN} - \Delta\bar{\hat{G}}_{kN} \tag{17}$$

this reduce the noise

A good approach is to low pass filter the noisy gradient vector to have a good estimate. A simple way to do that is

$$\bar{\hat{G}}_k = w\bar{\hat{G}}_{k-1} + (1-w)\hat{G}_k, \qquad \bar{\hat{G}}(0) = \hat{G}(0) \tag{18}$$

Where 0 < w < 1 determine the band width of the low pass filter. If w is close to unity the filter bandwidth is small. On the other hand if w is small the filter has a large band width. With this filtered quantity the LMS equation is

$$\hat{C}_{k+1} = \hat{C}_k - \Delta\bar{\hat{G}}_k \tag{19}$$

In the training mode we have the detected sequence and the transmitted sequence at the receiver to adjust the weight of the coefficients of the equalizer. After training we can start the adaptive mode. The length of the training sequence has to be equal or more to the number of coefficients.

The training sequence is usually used to be periodic. The period is N=2k+1. In this case the gradient is averaged over the period length as in equation 17. The weights adjustment can be by making a decision on the received data and use this decision to calculate the error. As long as the receiver operate at low error rate the convergence algorithm will be ok. The second algorithm is to compare the received data with the transmitted data assuming it is provided by a probe initially for training.

If the channel response changes, the error will change. This mean that the weights will change according to equation 11. We will also have a change if the information sequence or the noise statistics change. Thus, the equalizer is adaptive.

a) Convergence Properties of the LMS Algorithm

The convergence properties given in equation 11 is governed by the step size parameter. We will choice the step size parameter to ensure convergence of the steepest descent algorithm equation 7 using the exact value of the gradient.

We have

$$\begin{aligned} \boldsymbol{C}_{k+1} &= \boldsymbol{C}_k - \Delta \boldsymbol{G}_k \\ &= (\boldsymbol{I} - \Delta \boldsymbol{\Gamma})\boldsymbol{C}_k + \Delta \boldsymbol{\xi} \end{aligned} \tag{20}$$

This can be modeled by the closed loop control system in figure 3. The autocorrelation matrix in 20 is coupled. In order to solve the equations in 20 we have to decouple this matrix. This is also done to find the convergence property. The appropriate linear transformation to decouple this matrix is

$$\boldsymbol{\Gamma} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^H \tag{21}$$

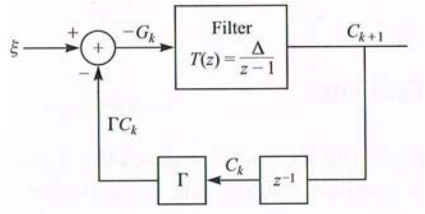

Figure 3

This form has a matrix with diagonal equal to the Eigen values of the system.

We put 21 into 20 and we have

$$\boldsymbol{C}_{k+1}^o = (\boldsymbol{I} - \Delta \boldsymbol{\Lambda})\boldsymbol{C}_k^o + \Delta \boldsymbol{\xi}^o \tag{22}$$

Now the system is decupled and the solution is

$$\boldsymbol{C}_{k+1}^o = (\boldsymbol{I} - \Delta \boldsymbol{\Lambda})\boldsymbol{C}_k^o \tag{23}$$

This mean that the system will converge if

$$|1 - \Delta\lambda_k| < 1, \qquad k = -K, \ldots, -1, 0, 1, \ldots, K \tag{24}$$

The inequality that insure convergence is

$$0 < \Delta < \frac{2}{\lambda_{\max}} \tag{25}$$

We have an upper bound on the max Eigen value

$$\begin{aligned} \lambda_{\max} &< \sum_{k=-K}^{K} \lambda_k = \operatorname{tr} \boldsymbol{\Gamma} = (2K+1)\Gamma_{kk} \\ &= (2K+1)(x_0 + N_0) \end{aligned} \tag{26}$$

The convergence is fast if (1-deltaΛ) is small. This mean that the pole is far from the unit circle. As we can see from the equations λ max/λ min determine the convergence rate.

IV. Excess MSE Due to Noisy Gradient Estimates

The receiver when adjusting the coefficients use a noisy estimate of the gradient vector. the noise result in random fluctuation from the optimal value. So we have j = j min +j delta were j delta is the variance of noise.

The total MSE at the output is

$$J = J_{\min} + (\boldsymbol{C} - \boldsymbol{C}_{\text{opt}})^H \boldsymbol{\Gamma} (\boldsymbol{C} - \boldsymbol{C}_{\text{opt}}) \tag{27}$$

Were C opt is the optimum value of C. simplifying we get

$$J = J_{\min} + \sum_{k=-K}^{K} \lambda_k E|c_k^o - c_{k\,\text{opt}}^o|^2 \tag{28}$$

We find j delta as

$$J_\Delta = \sum_{k=-K}^{K} \lambda_k E|c_k^o - c_{k\,\text{opt}}^o|^2 \tag{29}$$

It has been shown that

$$J_\Delta = \Delta^2 J_{\min} \sum_{k=-K}^{K} \frac{\lambda_k^2}{1 - (1 - \Delta\lambda_k)^2} \tag{30}$$

Simplifying

$$\begin{aligned} J_\Delta &\approx \tfrac{1}{2}\Delta J_{\min} \sum_{k=-K}^{K} \lambda_k \\ &\approx \tfrac{1}{2}\Delta J_{\min} \operatorname{tr} \boldsymbol{\Gamma} \\ &\approx \tfrac{1}{2}\Delta(2K+1) J_{\min}(x_0 + N_0) \end{aligned} \tag{31}$$

Were x0+N0 is the received signal plus noise

Delta should be

$$\Delta < \frac{2}{(2K+1)(x_0+N_0)} \qquad (32)$$

For example

$$\Delta = \frac{0.2}{(2K+1)(x_0+N_0)} \qquad (33)$$

The degradation in the output SNR of the equalizer due to excess MSE is less than 1 dB.

The analysis of the excess error is for assuming that C have converged to optimum.

The convergence has been studied by many researchers.

This is an example were we have 11 coefficients. We start with all of them are zero. We see that all of them will converge to the right value of the channel. We use MS algorithm. The Matlab code is given. As we can see the error converge to zero as the number of iteration increase. We also add noise to the input to the equalizer.

```
echo on
N=500;
K=5;
isi=[.05 -.063 .088 -.126 -.25 .9047 .25 0 .126 .038 .088];
sigma=.01;
delta=.09;
num_of_realizations=1000;
mse_av=zeros(1,N-2*K);
for j=1:num_of_realizations,
   for i=1:N,
      if (rand<.5),
         info(i)=-1;
      else
         info(i)=1;
      end;
      echo off;
   end;
   if (j==1);echo on ; end


end
noise =awgn(info,10);

   d_c=[ 0 0 0 0 0 0 0 0 0 0 0 ];

   for k=1:(N-10),
      y_kk=[noise(k:k+10)];
      out=y_kk*isi'
     z_k=d_c*y_kk.';
      e_k= out-z_k
      d_c = d_c + e_k * .115 *[ noise(k:k+10) ]
      mse(k) =e_k^2;
      echo off;
   end;
  if (j==1);echo on ;end
   mse_av=mse_av+mse;
   echo off
  echo on
mse_av=mse_av/num_of_realizations
```

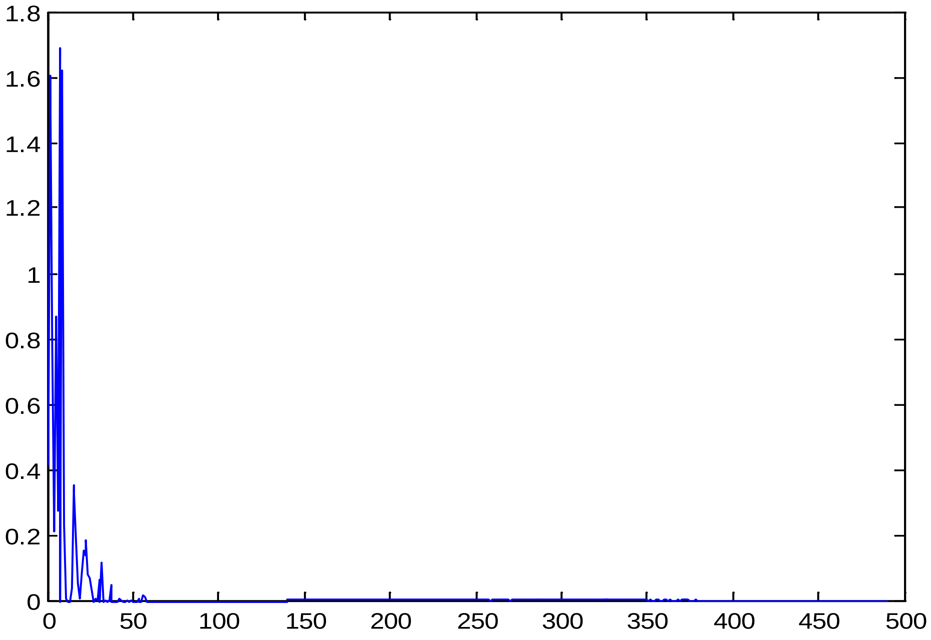

Figure 4

The choice of delta is very critical in the LMS algorithm. To be practical we design the most significant bits for an estimate of the coefficient and the least significant bits for delta so that the LMS algorithm find its way exactly to C opt.

It is important to mention that the LMS algorithm is able to track slowly time varying system. In other words j min is a function of time and the MSE surface is moving with time index n. The LMS method try to track j min but always behind it. This mean that we have another error which is the lag error. The total error is

$$J_{\text{total}} = J_{\min}(n) + J_{\Delta} + J_l \qquad (34)$$

Where j l is the lag error.

Now we try to plot j delta and j lag as a function of delta. They will behave as in the figure below. J delta decrease and j lag increase as a function of delta. The total will have a minimum which is the optimum choice for delta.

If the time varying system is fast the j lag will dominate the error. In this case the LMS will not be appropriate and we must use recursive least squares algorithms.

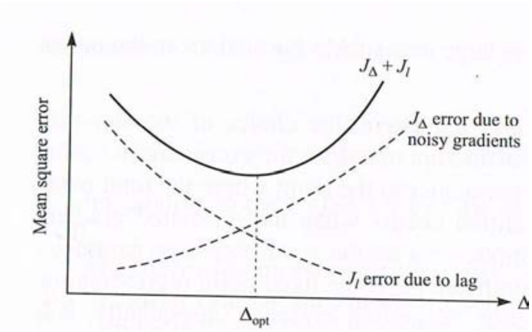

Figure 5

V. Accelerating the Initial Convergence rate in the LMS Algorithm

As we see the convergence rate is controlled by delta. It is also strongly related to the spectral characteristics of the channel. By that I mean the ratio of lambda max to lambda min. if this ratio is close to one the convergence will be fast. If not the convergence will be slow. In this case we say that the channel has poor spectral characteristics.

Researchers investigated this topic from all angels. One simple way is to start with large delta and reduce it to the optimum as time pass by.

To accelerate initial convergence methods have been studied in 1971 and 1977. One method is to replace delta by the weighting matrix W. the form is

$$\begin{aligned} \hat{C}_{k+1} &= \hat{C}_k - W\hat{G}_k \\ &= \hat{C}_k + W(\Gamma\hat{C} - \xi) \\ &= \hat{C}_k + We_kV_k^* \end{aligned} \qquad (35)$$

Where W ideally is the inverse of the autocorrelation matrix of the input data. The autocorrelation can be estimated and the inverse can be found.

If the training sequence is periodic, W can be seen as a single row and the block diagram of the system will be as in figure 6.

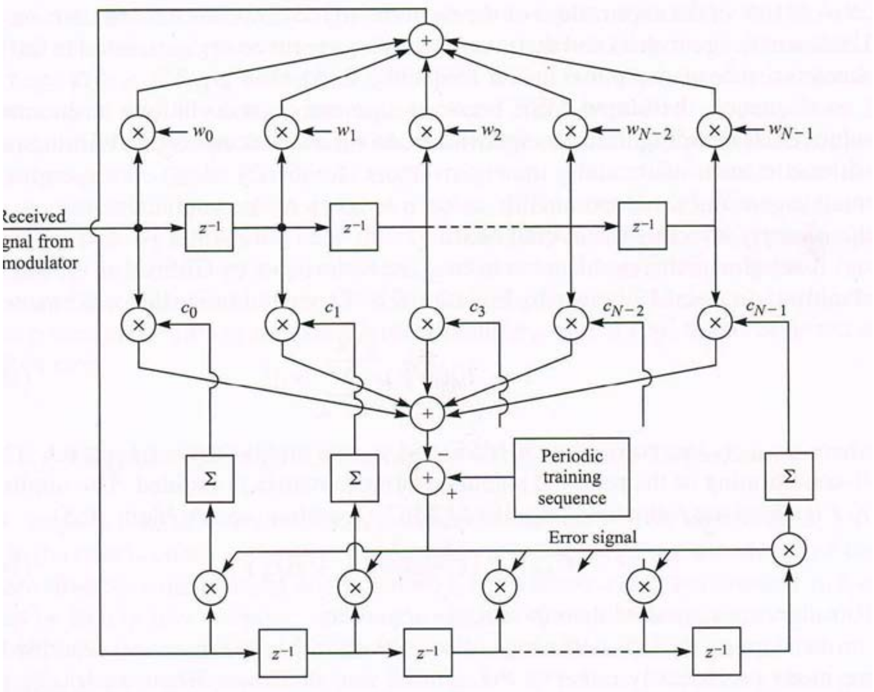

Figure 6

We can estimate the weights from the received signal. The basic steps are:

First we collect one period of N samples in the equalizer delay line. Then we compute N-point DFT (Rn). Then we compute the power spectrum (Rn^2) and we add to it N times the estimate of the noise variance. Then we compute the inverse DFT. This yields wn in figure 6. To adjust the taps we have

$$c_j^{(k+1)} = c_j^{(k)} - e_j \sum_{m=0}^{N-1} w_k v^*_{k-j-m}, \qquad j = 0, 1, \ldots, N-1 \quad (36)$$

VI. Conclusion

As in the case of linear adaptive equalizer the coefficient of decision feedback equalizer can be adjusted recursively. This is based on minimization of MSE.

(37)

We should note that this will converge and for the expected value we can use (epsilon V).

(38)

This is the LMS algorithm. First we use a training sequence then we switch to dissection mode.

References Références Referencias

1. A. Papoulis, Probability, Random Variables, and Stochastic Process, 2002.
2. J. G. Proakis, Digital Communications, 2001.
3. R. J. Schilling, Engineering Analysis, 1988.
4. H. L. Van Trees, Detection, Estimation, and Modulation Theory,1968.
5. J. G, Proakis, Inroduction to Digital Signal Processing, 1988.
6. C. Chen, Linear System Theory and Design, 1984
7. S. Haykin, Communication System, 1983.
8. T. H. Glisson, Introduction to System Analysis, 1985.
9. Martin Schetzen, Airborne Doppler Radar, 2006.
10. Martin Schetzen, The Volterra & Wiener Theories of Nonlinear Systems, 2006.
11. Martin Schetzen, Discrete System using Matlab, 2004.
12. Arvin Grabel, Microelectronics, 1987.

This page is intentionally left blank