OPTIMAL TRAINING SEQUENCES FOR TDMA SYSTEMS

Markus Rupp
Bell-Labs, Lucent Technologies, Wireless Research Lab.
Zadelstede 1-10, 3431 JZ Nieuwegein, THE NETHERLANDS
Fax: +31 30609 7498, Tel: +31 30609 7493
e-mail:rupp@lucent.com

ABSTRACT

This paper deals with the problem of finding *optimal* training sequences for adaptive equalizers in TDMA systems. Such sequences give the equalizer a good initial value, are required for symbol-timing recovery and can provide a small but robust amount of information by utilizing a small set of codes with strong discrimination power. While every single task has been dealt with in literature, the combination of all three of them is a new problem that requires smart search strategies to explore a huge space of possibilities. On the example of a local wireless loop design, it is shown how the search problem can be reduced.

1 Introduction

Depending on the task for which a training sequence is required, various requirements need to be satisfied for optimal performance. So is the training sequence for an equalizer optimal if the equalizer learns rapidly. On the other hand, learning behavior of an adaptive algorithm depends on the algorithm in combination with the training signals. A Least-Squares (LS) algorithm offers largely independence on the training sequence but due to its numerical sensitivity, it is complicated to implement efficiently and robustly, even in its low complexity recursive forms. For a simple gradient-type algorithm (LMS), it is known from literature that for long update periods fastest training is obtained for a statistically white sequence (see, for example [1]). This is a statistical measure and can hardly be applied to a single training sequence of typically short duration. Other quality measures are required here and will be provided in Section 2.

Good symbol timing recovery (STR) is necessary to find the optimal sampling time and slot synchronization in a digital receiver. If this is achieved on a training sequence, it requires an a-periodic time-acf with a strong peak and small side-lobes which is not necessarily identical with the best training property as will be shown in Section 3.

Finally, good discrimination requires having a set of training sequences with strong distinction; optimal seems a set of orthogonal sequences. Orthogonality is, however, not necessarily allowing for good a-periodic time-acfs, limiting STR and/or training performance. We, furthermore, restrict ourselves to (complex-valued) binary training sequences since it is believed that they are well suited for chip design and lead to simple implementations.

2 Criteria for Channel Estimation and Equalizer Training

In classical approaches (see Haykin[1]) it is believed that best learning performance for gradient-type algorithms is obtained when the input sequence is *statistically white*. While this is a helpful statement in describing the learning performance of such algorithms on (very) long training runs, when applying a short training sequence, one can hardly speak of a statistically white property. The term statistically white thus needs to be redefined, or, even better, the condition for rapid learning needs to be redefined since only a very short sequence (much shorter than the data package) is of interest.

When writing the LMS recursion over several updates a particularly simple structure occurs when the consecutive regression vectors are orthogonal, i.e, $\mathbf{x}_k^H\mathbf{x}_{k-1}$. If we define the autocorrelation $r_x(k,l) = |\mathbf{x}_k^H\mathbf{x}_{k+l}|^2$, it turns out that the algorithm learns faster, the more $r_x(k,l) = 0$ for growing lag l (see also [2] for more details). Of particular interest are so-called cyclic sequences for which $r_x(k,l) = 0$ for $l = 1..M-1$ when M is the dimension of the vector. Such sequences span a space of size M and thus allow perfect identification up to this order. Real-valued binary sequences do not seem to exist with this property. In [2] a table is shown that lists cyclic sequences for arbitrary length M utilizing simple alphabets.

These sequences are cyclic; thus can be repeated. Note that the a-periodic time-acf of these sequences have very large side-lobes.

Updating by such a sequence leads to

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \frac{\alpha}{M}\mathbf{x}_k \left[d(k) - \mathbf{x}_k^H \mathbf{w}_k\right] \tag{1}$$

which can be rewritten into terms of weight error vectors $\mathbf{v}_k = \mathbf{w}_k - \mathbf{w}$ and noise $n(k) = d(k) - \mathbf{x}_k^H \mathbf{w}_k$:

$$\mathbf{v}_{k+1} = \left[\mathbf{I} - \frac{\alpha}{M}\mathbf{x}_k \mathbf{x}_k^H\right] \mathbf{v}_k - \frac{\alpha}{M}\mathbf{x}_k n(k) \tag{2}$$

After M steps of operation, the result is

$$\mathbf{v}_{M+1} = \left[\mathbf{I} - \frac{\alpha}{M}\sum_{l=1}^{M} \mathbf{x}_l \mathbf{x}_l^H\right] \mathbf{v}_1 - \frac{\alpha}{M}\sum_{l=1}^{M} n(l)\mathbf{x}_l \tag{3}$$

which can be reformulated with help of the matrix $\mathbf{X} = [\mathbf{x}_1, ..\mathbf{x}_M]$ and the noise vector $\mathbf{n}_1 = [n(1), .., n(M)]$ to

$$\mathbf{v}_{M+1} = \left[\mathbf{I} - \frac{\alpha}{M}\mathbf{X}\mathbf{X}^H\right] \mathbf{v}_1 - \frac{\alpha}{M}\mathbf{X}\mathbf{n}_1 \tag{4}$$

The corresponding weight vector error after M steps is now

$$E[|\mathbf{v}_{M+1}|^2] = (1-\alpha)^2 E[|\mathbf{v}_1|^2] + \alpha^2 \sigma_n^2 \tag{5}$$

thus the optimal step-size

$$\alpha_{opt} = \frac{E[|\mathbf{v}_1|^2]}{E[|\mathbf{v}_1|^2] + \sigma_n^2} \, . \tag{6}$$

Initially, it can be expected that $E[|\mathbf{v}_1|^2]$ is much larger than $\sigma_n^2$and thus initially $\alpha \approx 1$. This allows optimizing the step-size sequence on a per-block update basis. It can be shown that the optimal sequence is $\alpha_{opt}(k) = \frac{1}{k}$ and the corresponding weight error vector variance decreases

$$E[|\mathbf{v}_{nM}|^2] = \frac{\sigma_n^2}{n} \, . \tag{7}$$

It is of interest, whether to keep the same training sequence and repeat only the update sequence (thus, making the training sequence to appear longer) or to repeat the training sequence several times on transmission so that new noise components are added (thus, really making the training sequence longer). For both cases we obtain after n blocks of M updates:

$$\begin{aligned} E[|\mathbf{v}_{nM}|^2] &= \prod_{l=1}^{n}(1-\alpha_l)^2 E[|\mathbf{v}_1|^2] \\ &+ \sigma_n^2 \begin{cases} \left[\sum_{l=1}^{n}\prod_{k=l+1}^{n}\alpha_l(1-\alpha_k)^2\right]^2 & \text{keep} \\ \sum_{l=1}^{n}\left[\prod_{k=l+1}^{n}\alpha_l(1-\alpha_k)^2\right]^2 & \text{rep.} \end{cases} \end{aligned} \tag{8}$$

where *keep* and *rep.*(=repeat) are indicating the two options. For the optimal step-size sequence the terms can be solved explicitly and one obtains σ_n^2 for *keep* and σ_n^2/n for *repeat*. In other words for the *keep* mode, more updates will not improve the accuracy any further. The result for the *keep* mode can be obtained in one step with step-size $\alpha = 1$.

This is different for a (truly-) longer training sequence. The improvement is proportional to $1/n$. Note that for the repeat mode under cyclic sequences, the LMS achieves the same estimation quality as an LS-estimation. With growing length $N = nM$ of the training sequence, the MSE improves proportionally. A further advantage is that the same LMS circuitry can be reused for channel tracking in the decision directed mode.

3 Criteria for Symbol Timing Recovery

A common STR-technique is applying a matched filter. This technique works well if the channel's impulse response displays one clear peak, carrying most of the symbol's energy. In this case, a sequence that has an a-periodic time-acf of impulse shape results after matched filtering with the channel impulse response and thus the strong peak can be detected. A sequence that offers a zero autocorrelation for arbitrary time lags> 0 does not exist in (real or complex-valued) binary form. Well-known M-sequences[3] with maximum side-lobe one are typically very long and thus not appropriate for equalizer sequence. Barker codes[4] seem to be a perfect solution since they offer very small side-lobes (max equals 1), however, only very few Barker sequences (of maximum length $M = 13$) are known. The number can be increased by two measures: (a) combining real-valued Barker sequences to complex-valued ones and (b) allowing larger side-lobes than one, *generalized Barker Codes* can be defined. For example, for side-lobes two and three there exist solutions for all N.

In order to test their identification quality, the code for $N = 13$,

$$\mathbf{t}_1 \triangleq [1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1]$$

was selected. The MSE (relative to M/N, optimal value is one) for LS is listed in Table1. Although for most cases ($M > 2$) the quality is close to optimal, none is perfectly orthogonal allowing for reduced complexity. Repeating the sequence to $N = 26$ helps only little.

4 Criteria for Discrimination

Very robust (but small amounts of) information can be transmitted in the training word itself by selecting a set of training words rather than a single one. Transmitted information can be channel quality information, like the SNR, a change in the access scheme or its parameters.

Assuming perfect synchronization, a good criteria for a set of M training words, $\mathbf{t}_1, .., \mathbf{t}_M$, each of length N is be to be orthogonal:

$$[\mathbf{t}_1; \mathbf{t}_2; ...; \mathbf{t}_M]\, [\mathbf{t}_1; \mathbf{t}_2; ...; \mathbf{t}_M]^H = N\, \mathbf{I}_M . \tag{9}$$

$\mathbf{I}_M$ denoting the identity matrix of dimension M. However, such words usually do not show good STR properties. Therefore, a better approach, taking (small)

M	$N=13$	$N=26$
2	1.00	1.002
3	1.02	1.01
4	1.04	1.01
5	1.20	1.03
6	1.22	1.06
7	15.00	1.07
8		1.10
9		1.10
10		1.11
11		1.11
12		1.10
13		1.08

Table 1: Identification $(\mathrm{tr}([\mathbf{X}^H\mathbf{X}]^{-1})/(M/(N-M+1)))$ quality applying the Barker sequence once ($N=13$) and twice ($N=26$).

symbol-timing imperfection into account, is allowing for relatively small side-lobes in the acf while limiting the cross-correlation between the codes. More mathematically,

$$\begin{aligned} D(\mathbf{t}_i(k),\mathbf{t}_i(k+l)) &> c_3; \mathrm{for}|l|\le l_1 \\ &\le c_1; \mathrm{for}|l|> l_1 \\ D(\mathbf{t}_i(k),\mathbf{t}_j(k+l)) &\le c_1; \mathrm{for}|l|\le l_2 \\ &\le c_2; \mathrm{for}|l|> l_2 \end{aligned}$$

with $0\le c_1\le c_2 \ll c_3 \le N$. Here, D names a discrimination measure, for example the L_1-norm of the difference. For binary values, the two cases $\pm\mathbf{t}_i$ need to be looked at since a non-coherent detector cannot discriminate between them. The area for which STR-uncertainty exists, is specified by the parameters l_1, l_2. For $c_2=l_1=l_2=0$ (9) is obtained again. Searching for codes with such restrictive criteria can become tedious. It is thus better to define a quality measure, for example by weighting the auto- and cross-correlation:

$$Q_{ij}=\sum_l a_l D(\mathbf{t}_i(k),\mathbf{t}_i(k+l)) + \sum_l b_l D(\mathbf{t}_i(k)\mathbf{t}_j(k+l))\,.$$

Since only a small area for the time lag is of interest, the search can be considerably improved. Summing up over all candidates $\mathbf{t}_i$ and $\mathbf{t}_j$, a measure is obtained. The candidate set with the smallest measure gives the best results. This procedure has the advantage that it provides at least one set that has been shown best.

5 Combining Constraints

Combining two or even all three of the above-mentioned properties without loosing desired properties is far away from being a simple task. Concatenating three sequences that are optimal for each task requires too long sequences and the data payload would become too high. As shown before, having a Barker sequence with almost white statistical properties is not the optimum sequence for fastest learning nor is there a set of Barker sequences with good discrimination properties. For the FWL system, five different modulation schemes are to distinguish: QPSK,8PSK,16QAM,32QAM and 64QAM. Note that for a given training length N, a space of $(2^N)^5$ binary sequence is to search through and measure their quality on a set of typical channels. Assuming $N=21$, if this task takes 5min. for each training sequence on a 233MHz Pentium, it would take 3.8×10^{26} CPU-years. Note also that if the length is not given, one also has to search through various values of N.

While QPSK gives good results for small training sequences, the higher modulation schemes require more training. Training with two Barker sequences ($N=26$ as indicated in Table 1) showed acceptable results. This, however, requires that a set with four members of Barker(-like) sequences must exist with good discrimination power. Unfortunately, this is not the case. Also, by using shorter ($N=21$) training, the same SER could be obtained.

It was thus decided to run the training for QPSK on a Barker code of length $N=13$ $(=\mathbf{t}_1)$ and to extend the other modulation schemes with eight more training symbols. This leaves the search space at 1.8×10^{20} CPU-years. The first task of the receiver then is to discriminate between this Barker code and another sequence of length $N=13$. Such a sequence must also have good STR properties because the STR is done now on these two sequences. There are 88 sequences with maximum side-lobe two. Among those 88 sequences, the 12 that showed best discrimination towards the Barker sequence were selected. The discrimination power was measured at 0dB SNR for QPSK and automatic STR, (i.e. the starting point of the sequence was not known to the receiver) under 500 channels. This small set of candidates was run on equalizer training to select the best. The result of this is the following sequence with side-lobe two:

$$\mathbf{t}_{11} \triangleq [1,-1,1,1,1,1,-1,1,-1,-1,1,1,-1]\,. \qquad (10)$$

The search took only 2CPU hours for this step.

The problem of STR has now been de-coupled from the remaining problem. Only a set of four sequences ($\mathbf{d}_1-\mathbf{d}_4$, see Table 2) needs to be found to discriminate to the four higher modulation schemes. Note that the Walsh-Hadamard sequences are not a good solution here. First, they only provide perfect discrimination at the correct timing; secondly, they are sequences of strong periodicities offering not much training improvement, which is very important for the higher modulation schemes.

For a length-8 binary code, there exists $\binom{256!}{252!\times 4!} = 256\times255\times254\times253/24 = 174\ 10^6$

Modulation	Part 1	Part 2
QPSK	$\mathbf{t}_1(N=13)$	—
8PSK	$\mathbf{t}_{11}(N=13)$	$\mathbf{d}_1(N=8)$
16QAM	$\mathbf{t}_{11}(N=13)$	$\mathbf{d}_2(N=8)$
32QAM	$\mathbf{t}_{11}(N=13)$	$\mathbf{d}_3(N=8)$
64QAM	$\mathbf{t}_{11}(N=13)$	$\mathbf{d}_4(N=8)$

Table 2: *Partitioning of the Training Sequences.*

8PSK	$\mathbf{d}_1 = [$	1	-1	-1	-1	1	1	-1	-1]
16QAM	$\mathbf{d}_2 = [$	1	1	1	1	-1	-1	1	-1]
32QAM	$\mathbf{d}_3 = [$	-1	1	-1	1	-1	-1	-1	1]
64QAM	$\mathbf{d}_4 = [$	-1	-1	1	-1	1	1	1	1]

Table 3: *The best four codes* $\mathbf{d}_1 - \mathbf{d}_4$.

possibilities for sets with four members. If a check on 500 channels takes 5CPU minutes, the full search procedure takes $1,663$ CPU years! A useful pre-selection seems is to check the cross-correlation functions of the sets since this takes only a fraction of the time for computing the false decisions. However, if it takes only 0.1% of the time, the search still takes 1.6 CPU-years! A more practical method is thus to run the training of all the 256 possible training words over the set of 500 channels. This requires one CPU-day. From the best training words, one selects a subset with four members that have suitable discrimination power.

Surprisingly, simulations showed that depending on what modulation scheme is used, different training sequences are optimal. The six best training sequences for 8PSK at 17dB SNR, 16QAM at 20dB SNR and 64QAM at 25dB SNR were completely disjunct. This is not considered being a problem since one training sequence from each subset can be selected. The discrimination was tested at 0dB SNR. The so obtained set with best discrimination is listed in Table 3. The search for the 8-symbol set was done based on the minimum SER. A corresponding procedure based on minimum BER showed similar results. Since a different labeling could change the results considerably, this condition was not considered any further.

Two methods for discrimination are possible:

coh. The sequence $\mathbf{t}_{11}$ is taken into account as part of a 21 bit long word. The advantage is that up to 8 bits can be different (non-coherent only four!), the disadvantage that the majority of the bits is identical and thus will provide less discrimination power. The best discrimination possible (minimum) at perfect timing is thus $(13+8)$ versus $(13-8)=16$. However, only one can have all eight bits different. For seven differences it is 21-7=14. The selected set (defined further ahead) has only $(13+8)-(13-4)=12$ and $21-13=8$ due to its training properties.

n.-coh. The sequence $\mathbf{t}_{11}$ is not taken into the discrimination process but only the STR derived from it. Now the maximum difference between the bits can only be four due to the phase uncertainty. Thus, now the maximum discrimination power is eight versus zero, thus $8-0=8$. If only three differences occur, the discrimination is $8-(5-3)=6$, thus smaller than in the coherent case. For the selected set, we obtain 8 and $8-4=4$.

Assuming perfect timing, the set results in the following distances for the non-coherent case:

$$D_n = \begin{bmatrix} 0 & 4 & 4 & 8 \\ 4 & 0 & 8 & 4 \\ 4 & 8 & 0 & 4 \\ 8 & 4 & 4 & 0 \end{bmatrix},$$

while the coherent case results in

$$D_c = \begin{bmatrix} 0 & 12 & 12 & 8 \\ 12 & 0 & 8 & 12 \\ 12 & 8 & 0 & 12 \\ 8 & 12 & 12 & 0 \end{bmatrix}.$$

Simulation showed that the coherent method is about two times better, i.e., at 0dB SNR the error count was two times less than for the non-coherent method. Assuming equal error probability, the discrimination numbers let expect a three dB improvement.

6 Conclusions

In conclusion, rather than exploring a space of $(2^{21})^5 = 4\times 10^{31}$ possible training sequence sets in order to find very particular properties in training, STR and discrimination, the problem was split in smaller tasks that could be handled by a Pentium processor in acceptable time. Thus, the overall search time was considerably shortened.

References

[1] Simon Haykin, *Adaptive Filter Theory*, 3. edition, Prentice Hall 1996.

[2] M.Rupp, " Fast implementation of the LMS algorithm," Proc. Eusipco 2000.

[3] J.D. Gibson, *The Mobile Communication Handbook,* IEEE Press, 1996.

[4] R.H.Barker, *Group Synchronizing of Binary Digital Systems,* Communication Theory, Acad. Press NY, 1953.