A METHOD OF TAP CLUSTERING IN AN ADAPTIVE FILTER FOR VIDEO GHOST CANCELLATION

A. Lish

Philips Laboratories
North American Philips Corporation
345 Scarborough Road
Briarcliff Manor, NY 10510

ABSTRACT

A video echo cancellation scheme that uses system identification techniques is presented, where the associated transversal filter has fewer multipliers than taps. This topology is made possible by exploiting the properties of the echo channel impulse response. First, optima are obtained by performing multiple system identifications with the available multipliers in various configurations along the transversal delay line. Then, a novel linear mapping is applied to reconstruct the real optimum of the filter, and the multipliers are redistributed where needed. The mapping reduces the redistribution problem to a simple geometric interpretation and permits a non-white reference to be used in system identification.

INTRODUCTION

In video transmission, multipath reception, caused by signal reflections off buildings or other terrestrial objects, causes distinct time shifted, sometimes smeared replicas or ghosts of the image to appear. Additionally, ghosting can be found in cable systems having poorly terminated connections. Since the late 1970s, researchers have been devising methods to cancel video ghosts [1]. This problem has proven to be a challenge both from the point of view of cancellation effectiveness as well as implementation efficiency.

In examining the impulse response of the ghost channel, we find that it is sparse because many (about 60%) of the response values over time are zero. If an inverse filter of the channel was constructed to cancel the echo function, it too can be made sparse, not requiring a multiplier at every tap in its transversal delay line.

We will present an approach to reduced-multiplier adaptive filters that differs from previously reported work [2]. If a transversal filter with a multiplier at each tap is set up to system-identify an echo channel, the tap weight pattern will mimick the channel impulse response in a least squares sense even if the order of the actual impulse response is higher than the filter. The optimal tap weight pattern for this filled-multiplier filter we will call the global optimum. Suppose that the multipliers at some taps are eliminated. If the resulting reduced multiplier-filter system-identifies the echo channel, the weights will assume a new optimum pattern. The new pattern can be related to the global optimum by a non-invertible linear mapping. However, if we combine a sufficient number of optima from different multiplier configurations into a single pseudo-optimum, an invertible mapping can be built between the pseudo-optimum and the global optimum. Consequently, with this mapping we can find our way to the global optimum with a reduced set of multipliers, and then redistribute the multipliers only where needed. In the redistribution, the multipliers will form local groups or clusters along the delay taps, hence, the redistribution will be called clustering.

We will first present the preliminaries to more specifically state the cluster problem. Next, we will discuss more formally the cluster formulation; this mainly involves a geometric description of the invertible optimum mapping, A_c. This mapping permits a non-white reference to be used in system identification. Special cases of A_c are included to help gain insight. Finally, some clustering algorithms based on A_c are summarized; simulation examples are also presented.

PRELIMINARIES

In a FIR adaptive filter the weights and signal are represented by R^M vectors $W^T=\{w_o \ldots w_{M-1}\}$ and $X^T = \{x(n) \ldots x(n-M+1)\}$, respectively. where M is the number of taps. The filter output may be then written by $y(n) = X^TW = W^TX$. When the filter weights are adjusted so that y(n) best approximates desired signal d(n) in the least squares sense, Weiner solution W^* results.

$$W^* = [E[XX^T]]^{-1} E[d(n)X] \quad (1)$$

$E[XX^T] = R$ and is known as the input auto correlation matrix.

The LMS gradient [3],

$$\hat{\nabla}(n) = -2\varepsilon(n)X(n)$$

may also be used, where $\varepsilon(n) = d(n) - X^TW$. Weights are iterated to reach W^* by following the gradient as shown below.

$$W(n+1) = W(n) - \mu\hat{\nabla}(n)$$

$$= W(n) + 2\mu\varepsilon(n)X(n) \quad (2)$$

The constant μ determines the rate of descent.

Considering adaptation, our problem is to distribute the multipliers only to the tap locations where they are needed and then adjust them to the correct value. This new filter structure is equivalent to a normal M tap FIR filter having M-N of the multipliers set to zero; this kind of tap weight pattern would be expected when identifying a non-smeared or smeared echo sequence that could be approximated by N active components. Suppose that the N multipliers are distributed in a known pattern that is different from the correct pattern of active taps required to replicate the echo. We can find out how the optimum associated with this known tap weight pattern is related to that of the global optimum of the filter having multipliers at all M taps. In fact, if we only know the optima of a sufficient number of different patterns containing N multipliers, we can combine this information to reconstruct what would be the global M tap optimum. The zero and non-zero taps can then be identified, leading to the correct multiplier distribution.

CLUSTER FORMULATION

To start, suppose that the FIR filter has M taps and N multipliers where N<M. The global optimum, $W^* = \{w^*_o \ldots w^*_{M-1}\}^T$, resides in R^M tap weight

CH2868-8/90/0000-3170$1.00 © 1990 IEEE

space. Also $W^* \varepsilon E^{No}$ where $E^{No} \subset R^M$ is the subspace spanned by the non-zero weights. E^{No} is defined when the multipliers are distributed to the correct tap pattern needed to identify the echo. For an arbitrary tap pattern spanned by E^{Ni}, there is optimum $W_{si}^* \varepsilon E^{Ni}$. With $E^{Ni} \subset R^M$ and i = 1,..,T, we put $\bigcap_i E^{Ni} = \phi$ and $\bigcup_i E^{Ni} = R^M$. We can now create an observed optimum W_c^* by direct sum as shown below.

$$W_c^* = \bigoplus_{i=1}^{T} W_{si}^* \tag{3}$$

Since W_c^* is in R^M, we can find a invertible mapping A_c:W^* --> W_c^*

$$W_c^* = A_c \, W^* \tag{4}$$

where A_c is an MXM matrix.

Before presenting the A_c equations, we illustrate our description with a 2 tap example having a single multiplier as displayed in Figure 1 (In this case M=2 and N=1). The single multiplier is first connected to the left delay tap (Fig. 1a) and is optimized to setting w_1'. When connected to the right delay tap (Fig. 1b), the optimum is found to be w_2'. After combining these coordinates to form W_c^* (Fig. 1c) it is seen that this is a point different from global optima, W^*, but can be related to W_c^* through A_c.

Using the concepts behind (3) and (4) and a derivation too lengthy to be presented here, the general M tap case is given by

$$W_c^* = \sum_{i=1}^{T} W_{si}^* = \sum_{i=1}^{T} \left[Z_i^{-1} \left[Z_i S_i R S_i Z_i^T \right]^{-1\square} Z_i^{T^{-1}} \right] R W^* \tag{5}$$

$$= A_c W^*,$$

where we define the switching matrix S = diag{s_K}, s_K = 1,0. s_K = 1 corresponds to taps having multipliers, where s_K = 0 is the converse.

$$[\text{Note: } \sum_{K=1}^{M} s_K = N]$$

Additionally, permutation matrix, Z, is defined as shown:

$$Z = [z_{i,K}] = \begin{cases} 1 \text{ if } i = s_K \left[\sum_{i=1}^{K} s_i \right] + (1-s_K) \left[\sum_{i=1}^{M} s_i + \sum_{i=1}^{K} (1-s_i) \right] \\ 0 \text{ otherwise} \end{cases} \qquad K=1, M \tag{6}$$

Note that for a given S matrix denoted by S_i in (5), there is a corresponding Z matrix Z_i. In the term $[Z_i S_i R S_i Z^T_i]$ we have created a kind of subspace auto correlation matrix. Pre-multiplying R by $Z_i S_i$ nulls rows corresponding to multiplierless taps and moves the remaining active rows to the top of the matrix; post-multiplying R by $S_i Z^T_i$ nulls and moves remaining columns to the left. The resultant matrix is an NxN sub-matrix nested in a MxM matrix of '0' elements. The operator $-1^{\square}$ inverts the NxN nested sub-matrix and places the results back in the same position the original sub-matrix occupied within the full MxM matrix. Additionally, to meet the conditions fulfilling (2) and (3), the switching pattern defined by {S_1, .. ,S_T] must follow the constraint

$$\sum_{i=1}^{T} S_i = I .$$

Now that mapping A_c has been described, it is easy to see how a cluster can be formed. Referring to the 2 tap example of Figure 2, we see that W_c^* is constructed in the same manner as in Figure 1. Both components of W_c^* are non-zero. If we were to use (5) and then find A_c^{-1}, the global optimum, W^*, can be calculated. Notice that W^* has a single non-zero component, hence, only one multiplier is needed in this simple cancellation filter. Thus, clustering in the 2 tap filter is achieved when the single multiplier is assigned to the first tap. In larger filters having M taps and N multipliers, the simple case described can be generalized, as will be shown in the 'Algorithms and Examples' section.

Special cases of A_c:

1. Suppose R = {r_{ij}} = diag {λ }= {Λ} (pseudo random noise is an approximation).

Following the same construction as (5),

$$W_c^* = \sum_{i=1}^{T} W_{si}^* = [\sum_{i=1}^{T} S_i \,] W^* = I W^* \tag{7}$$

If R = Λ, then any switching pattern corresponding to conditions forming (5) (i.e., non-ambiguously spanning full tap space R^M) yield a W_c^* matching W^*, no mapping necessary.

2. This case is trivial but should be mentioned. Suppose the set {S_i} consists of a single I matrix. Substituting this in (5) with some calculation yields

$$W_c^* = I W^* \tag{8}$$

3. This case is the opposite of case 2). Instead of having I, corresponding to one multiplier per tap, suppose there is only a single multiplier available for M taps. To model this, let {S_1 = diag{1,0,0,..,0}, s_2 = diag{0,1,0,..,0}, .., S_M = diag{0,0,..,0,1}}. Naturally

$$\sum_{i=1}^{T} S_i = I.$$

After some manipulation using (5), we get

$$W_c^* = \frac{1}{r_{ii}} R \, W^* \tag{9}$$

where r_{ii} is a main diagonal element of R.

Note: Switching patterns will vary between cases 1) and 2). Likewise it is likely that corresponding A_c matrices will have properties varying between those of I and $1/r_{ii}$ R. More on this will be discussed.

Algorithms and Examples

Combing:

A switching pattern that meets the conditions needed to obtain a valid W_c^* usable in (5) is the comb. The algorithms to be discussed all use this pattern. We will describe a 3 to 1 comb, but this approach is not restricted to a 3 to 1 ratio. Assuming that the transversal filter has M taps, let us say that we find (for example) the worst case echo requires that only N multipliers are needed in this structure at any given time. Let M = 3N. To form the comb, distribute the N weights evenly over the 3N taps, leaving a gap of 2 taps between multipliers. The Comb can be advanced. More specifically:

- Distribute N weights evenly over the FIRST of every Third 3N taps; LMS Converge† and Store Weight Values forming W_{s1}.
- Distribute N weights evenly over the SECOND of every Third 3N taps; LMS Converge† and Store Weight Values forming W_{s2}.
- Distribute N weights evenly over the THIRD of every Third 3N taps; LMS Converge† and Store Weight Values forming W_{s3}.
- Combine Stored Weight Values forming W_c^*.

In the sequence described, the comb advances position only after each W_{si} has converged to its final value. A useful variation would be to advance the comb after each LMS cycle of (2) thus updating W_c^* every 3 cycles. Intermediate values of W_c^* provided with this "Cyclic Comb" may be desirable in giving continuous subjective improvements to the picture as final convergence is reached.

1. Clustering with $R = \Lambda$

   a) Find W_c^* either by Advancing or Cyclic comb.

   b) Select N non-zero†† values of weights; assign the N multipliers to related locations and set to corresponding weight values.

2. Clustering with $R \neq \Lambda$

This algorithm uses steps a) and b) from above. However, this intermediate step must be added between a) and b).

   a') Calculate $W^* = A_c^{-1} W_c$. W^* is used in step b).

Since A_c^{-1} is a function of R and $\{S_1,..,S_T\}$, it may be pre-calculated and stored. It is best to find a reference signal and switching pattern (i.e. perhaps a different comb ratio) to lead to a very simple well behaved A_c^{-1} structure, as this will facilitate multiplication later.

3. Coarse-Fine Clustering Projecting Forward through A_c

   a) Find W_c^* either by Advancing or Cyclic Comb.

   b) Select N maximum magnitude elements of W_c^*; assign the N multipliers to related locations and set to corresponding Weight Values (considered as initial condition).

   c) With the Weights set as in b), perform final LMS convergence. This fine tunes the weights to match W^*, thus compensating for sidelobe distortion (as mentioned below).

This approach is very useful for an A_c matrix having a dominant main diagonal and off diagonal elements that produce negligible sidelobes when a single basis element of W^* is mapped to W_c^*. This method is also computationally simpler than algorithm 2) for non-sparse A_c^{-1} and can work where A_c^{-1} is too ill behaved for 2).

In our test, a sin t/t reference signal was chosen for its smooth spectral characteristics (any frequency null would make tuning at that point indeterminate). To match video bandwidth, the sin t/t roll-off was set at 5 MHz. Sampling was done at 3x bandwidth with a 3 to 1 comb switching pattern. Ac was calculated using (5) and found to be practically indistinguishable from $(1/r_{ii})R$, (9) [case 3]. To show the utility of the mapping, a single echo that can be described as having $W^* = \{0,0,..,1,..,0,0\}$ was processed with a 3 to 1 Advancing Comb (with LMS as described in this section under 'Combing') to obtain W_c^*. W^* was also mapped through A_c to find W_c^* according to (5). Figure 3 shows the results of the Advancing Comb while Figure 4 that of the mapping.

In Figure 5, a video image is put through a multiple smeared ghosting function and de-ghosted with algorithm 3).

CONCLUSIONS

We have presented an approach to echo cancellation using an adaptive transversal filter having less multipliers than taps. The basis for the approach is a new invertible mapping, A_c, relating optima from different multiplier configurations to the global optimum of a full multiplier filter. Future work on matching the best A_c matricies to the right algorithms, as well as finding new usable mappings can prove fruitful. Additionally, the effectiveness of the approach should be studied with a large variety of echo channels.

ACKNOWLEDGEMENTS

Thanks to Steve Herman for providing a sample of his simulations for this paper, and to Indur Mandhyan for encouraging advice. I especially thank the Lord for his love and patience with the author.

BIBLIOGRAPHY

[1.] Ciciora, W., et at., "Tutorial on Ghost Cancelling Television System", IEEE Trans. on Consumer Electronics; (February 1979).

[2.] Murata et al., "Practical TV Ghost Canceller Using a 2-Stage CCD Transversal Filter", Journal of Television, Vol. 40, No. 11, (1986), (Japan).

[3.] Widrow, B. McCool, J., et al., "Stationary and Non-stationary Learning Characteristics of the LMS Adaptive Filter", Proc. of IEEE, (August 1976).

[4.] Frost, O.L., "An Algorithm for Linearly Constrained Adaptive Array Processing", Proc. of IEEE, (August 1972).

† This convergence is an example of constrained adaptation [4].

†† Some components of Wc* would only be zero under ideal conditions with perfect convergence. The correct objective could be achieved by selecting the N maximum magnitude weight values.

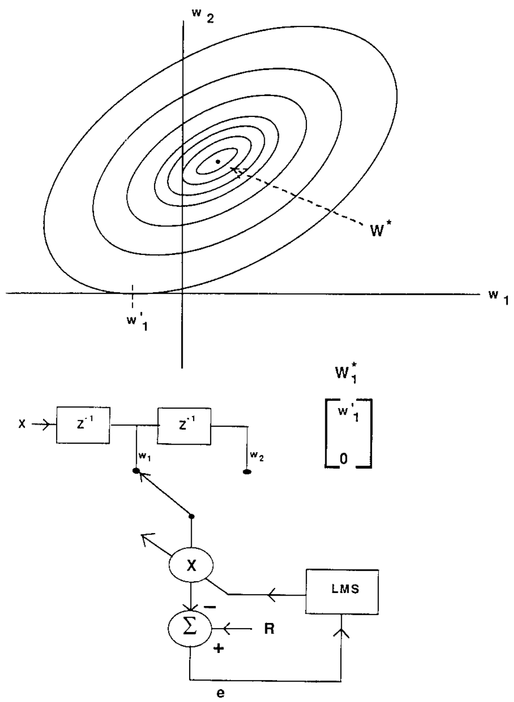


Figure 1a. 2 to 1 Cluster Mapping: w_1'

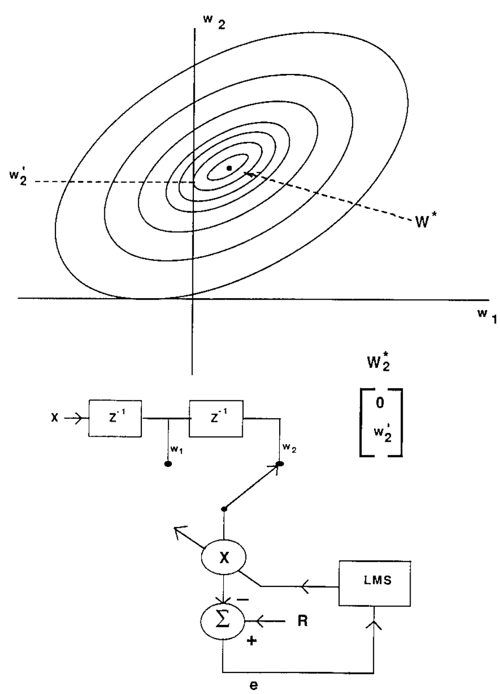


Figure 1b. 2 to 1 Cluster Mapping: w_2'

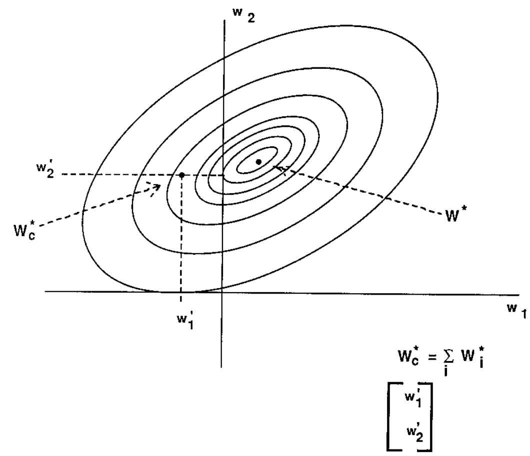


Figure 1c. 2 to 1 Cluster Mapping: W_3^*

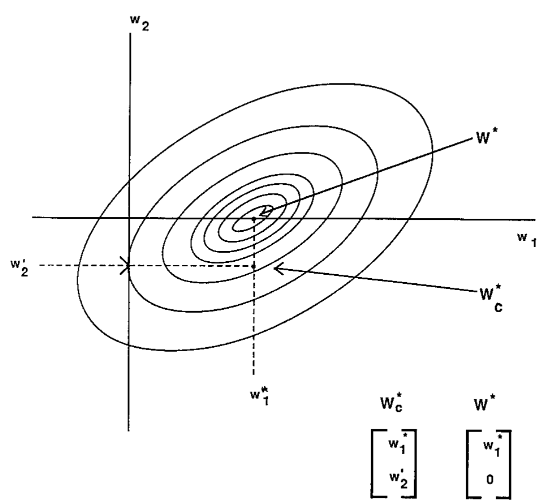


Figure 2. A Cluster Case.

Figure 3.

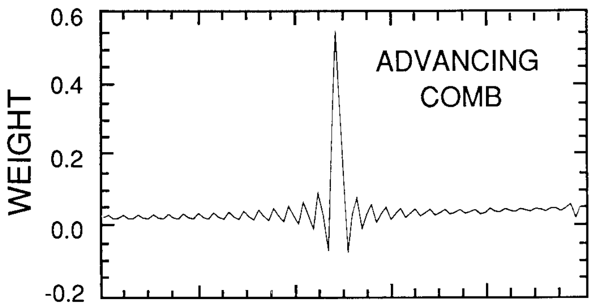
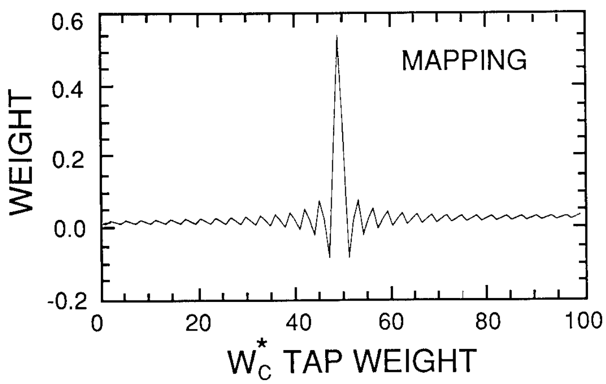


Figure 4.



Original

Ghosted

De-Ghosted

Figure 5. Video Deghosting Example.