SPATIAL ALGORITHMS FOR BLIND CHANNEL ESTIMATION

Y M S S Yapa

Dept. of Electrical and Computer Eng.,
National University of Singapore.

A Rahim Leyman[*]

$A^{\star}$STAR - Institute for Infocomm Research,
21 Heng Mui Keng Terrace, Singapore 119613.

ABSTRACT

Of the different information sources used by blind algorithms, explicit use of Finite Alphabet (FA) data is more recent than either the statistical data embedded at the source or the algebraic structure present in the channel. In channels that can be modeled using Finite Impulse Response (FIR) structures, the FA property results in the received vector set being *clustered* around theoretical centers. These centers are the result of the convolution of the channel matrix with a given transmitter symbol constellation. Defined as *spatial structure* in this paper, they provide sufficient information for blind estimation of channel coefficients. Here, we introduce two spatial tools, the *Primary* and *Secondary* Clustering Algorithms capable of processing the information structures described above. Then, using these two tools, we present the *Channel Estimation By Difference Set (CEDS)* algorithm for the estimating channel impulse response coefficients.

1. INTRODUCTION

Mobile communication has become one of the fastest growing technologies of the twenty first century. However, inherent properties of the wireless media place fundamental limitations on the capacity of such mobile systems. One of the main problems faced in wireless communication is Inter Symbol Interference (ISI). Traditionally, ISI has been compensated using adaptive equalizers with training data. However, recent demand for high bandwidth has made these algorithms obsolete with more efficient blind algorithms taking their place.

Most blind algorithms exploit two primary sources of embedded data for information: Statistical data via second [4] and fourth [2] order cumulants, and channel structure information via subspace algorithms such as the Cross Relational [5], Noise Subspace [6] and Least Squares Smoothing [3]. In addition to the two traditional sources, the finite alphabet of a transmitter encodes data into the wireless media [1]. The data structure is visible in the output vector set of a M-output platform, and bears semblance to a lattice described in M-space. Such, the data structures are named as *spatial structures* in this paper.

This paper is divided into three main sections. First, we will introduce the spatial structure and the mathematics used to model it. Secondly, we will formulate tools to handle the spatial structure. These tools form the core of spatial data processing our algorithm requires. Finally, we present our estimation algorithm, the *Channel Estimation by Difference Sets (CEDS)* and conclude this paper by providing an indepth analysis and discussion into its behavior.

*This work was supported under $A^{\star}$STAR Science and Engineering Research grant No: 0221060041

2. MATHEMATICAL PRELIMINARIES

2.1. Signal and Channel Model

Consider a Finite Impulse Response (FIR), Single Input M-Output (SIMO) channel of length L defined by its impulse response coefficients $\{h_{ij}\}$, where $i \in \{1, ..., M\}$ and $j \in \{0, ..., L\}$. The received baseband signal at the j^{th} receiver can then be described by,

$$x_j(n) \triangleq \sum_{l=0}^{L} h_{jl} s_{n-l} + w_j(n) \quad (1)$$

where s_n is the transmitted symbol at time index $t = nT$ and $w_j(n)$ is the noise component of the j^{th} channel at the same time index. T here is the symbol period. Stacking $x_j(n),\ j \in \{1, ..., M\}$ to form the vector,

$$\mathbf{x}(n) \quad \triangleq \quad \mathbf{H}\mathbf{s}_n + \mathbf{w}(n) \quad (2)$$

we obtain the mathematical model describing the structure of the received vector in a FIR, SIMO channel. In the above equation, $\mathbf{x}(n) \triangleq [x_1(n), ..., x_M(n)]'$, $\mathbf{w}(n) \triangleq [w_1(n), ..., w_M(n)]'$ and $\mathbf{s}_n \triangleq [s_n, ..., s_{n-L}]'$ describe the *received*, *noise* and transmitted *source* vectors respectively. The matrix $\mathbf{H} \triangleq [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_M]'$ denotes the impulse channel matrix with the j^{th} row defined as $\mathbf{h}_j \triangleq [h_{j0}, h_{j1}, ..., h_{jL}]'$.

2.2. Objective, Assumptions and Notations

The purpose of our algorithm is to recover the channel parameters $\mathbf{H}$ and L using spatial data available in $\mathbf{x}(n)$ under the following key assumptions:

a) The channel is stationary for the time duration needed to collect data for estimation.

b) The noise $\{w_i\}$ is zero mean, and statistically independent of the transmitted symbol sequence.

c) The transmitter symbols are independent and chosen from a finite alphabet. For the purpose of this paper $s_i \in \{1, -1\}$, and we define this alphabet, $C_T \triangleq \{1, -1\}$.

d) The channel matrix $\mathbf{H}$ is full column rank.

Let $\mathbf{M}$ denote a Matrix. Then, in developing our paper, we shall use the notations, $\mathbf{M}'$ and $\mathbf{M}^{\dagger}$, to denote the matrix operators, transposition and inversion (the pseudo-inverse when the matrix is not square). Additionally, as described later, the notation $S_a \sim \mathbf{y}(a)$ will be used to link a state S_a to its spatial vector $\mathbf{y}(a)$.

2.3. The Spatial Structure

Consider the SIMO system described in (2). Under noiseless conditions, the received vector $\mathbf{x}(n)$ can be represented using its noiseless counterpart, $\mathbf{y}(n) = \mathbf{H}\mathbf{s}_n$. Then, under assumptions *(a)* and *(c)*, the vector set containing all received noiseless vectors,

$$Y \triangleq \left\{\mathbf{y}|\mathbf{y} = \mathbf{y}(i) \quad i \in \{1, ..., N\}\right\} \tag{3}$$

is finite with at most Z^L elements. In the equation, N is the number of received vectors and Z represents the number of symbols in the constellation C_T. Each element of Y, $\mathbf{y} \in Y$ describes a point in an M-dimensional space. Thus the set Y describes a lattice in M-dimensional space. This M-dimensional lattice can be thought as a state diagram, where each element represents a unique state. In this model, the output vector $\mathbf{y}(n)$ can then be seen transiting between the states in response to the input symbol s_n. This duality between the state diagram and its M-dimensional vector representation forms the basis of our spatial algorithms. In this paper, we associate a given state S_k to its respective M-dimensional *spatial vector* $\mathbf{y}(k)$ by,

$$S_k \sim \mathbf{y}(k) \tag{4}$$

The spatial vector $\mathbf{y}(k) = \mathbf{H}\mathbf{s}_k$, in turn consists of the two components, the channel matrix $\mathbf{H}$, and the *source vector segment* $\mathbf{s}_k = [s_k, ..., s_{k-L}]'$.

3. SPATIAL TOOLS

3.1. The Primary Clustering Algorithm (PCA)

The clustering algorithm used in our simulations is derived from the LBG algorithm Daneshgaran uses in [1]. However, instead of a single step approximation where all states are extracted from a single clustering iteration, our algorithm relies on a two-step approach that uses clustering in two subsequent steps defined by the thresholds D_1 and D_2. To begin deriving our algorithm, we shall first define $\tilde{Y}$ to be the set of extracted cluster vectors initially containing $C = 0$ elements. The clustering algorithm can then be described as follows:

i) Scan the received vectors sequentially, comparing the Euclidean squared distance, d of each received vector to the established C cluster centers.

$$d_{min} = \min_{m \in \{1,...,C\}} \sum_{i=1}^{M} [x_i(n) - \tilde{y}_i(m)]^2 \tag{5}$$

ii) If $d_{min} > D_1$, add the data vector as a new cluster center. Otherwise merge it to the closest center, m weighted by the number of points already merged.

iii) Sort the sub-clusters, $\tilde{\mathbf{y}} \in \tilde{Y}$ by the number of data points fused into each center.

iv) Beginning from the least populated sub-cluster, for each center, j compute distances l_{jk} to all other sub-clusters, $k \in \{1, ..., C\} \quad k \neq j$.

v) Find the closest center, k satisfying both $l_{jk} < D_2$ and $P_j + P_k < P_{MAX}$. Then, merge the centers j and k weighted by their populations.

The resultant set of vectors, $\tilde{Y}$ will be an approximate to the noiseless lattice structure Y. In this algorithm, P_i denotes the population of the sub-cluster i.

The distance thresholds D_1 and D_2 were empirically calculated using the clustering algorithm in an adaptive mode. In this step, Monte-Carlo iterations were carried out for each M-SNR pair, gradually increasing the threshold distance till the number of estimated centers converged around 2^{L+2} for D_1 in the first phase and to 2^{L+1} for D_2 in the second phase. Another threshold, $P_{MAX} = 0.8N/2^{L+1}$ is used in the second clustering phase to limit the number of vectors coalesced per cluster. This is 80% of the expected populations for each cluster, and was verified to be a good index using Monte Carlo iterations.

Using elementary curve fitting on results obtained above, we were able to derive empirical relationships for the two thresholds using the noise power, N_o and M. An interesting outcome of the modeling was the independence of the first threshold, $D_1 = N_o(2M + 5)$ from the channel length, L. This implies that the number of centers estimated by the first step of our clustering algorithm can also be used as a rough *blind estimator of the channel length*.

3.2. The Secondary Clustering Algorithm (SCA)

In addition to the primary need to separate the received data vectors into spatial clusters, we need an additional clustering tool that enables us to extract *vector families*. That is, given a vector family F_v of the vector $\mathbf{v}$ having a population of P_v,

$$F_v = \left\{\mathbf{f}_i|\mathbf{f}_i = \mathbf{v} + \mathbf{n}_i \quad i \in \{1, ..., P_v\}\right\} \tag{6}$$

we need to extract the estimates $\tilde{\mathbf{v}} \approx \mathbf{v}$ and $\tilde{P_v} \approx P_v$ from a vector set containing the vector family F_v in addition to other families. Here, the noise $\mathbf{n}_i$ is assumed to be zero mean. The algorithm used for this purpose is basically a derivative of our PCA. It is limited to the steps *(i)* to *(iii)*, with a variation in the derivation of D_1.

The threshold distance D_1 is empirically calculated using the deviation of the extracted vector population against the theoretical population. The theoretical population is extracted from a pilot output which is uncontaminated by noise. Two instances of the algorithm, one using noiseless data and the other in an adaptive form are run side by side across the entire M-SNR spectrum used in this paper. At each M-SNR, the adaptive algorithm iteratively increases the threshold distance D_1 till the extracted populations falls within an acceptable range of the theoretical populations. The expected values of D_1 across the twin indices of M and SNR are then tabulated to be used to separate and extract vector families.

4. CHANNEL ESTIMATION BY DIFFERENCE SETS (CEDS) ALGORITHM

In this section, we present an algorithm that estimates the channel parameters using purely spatial data. We will begin by introducing the mathematical structures that form the basis of the CEDS algorithm. To begin understanding the structures embedded in the spatial data we first need to declare the following definition.

Definition: Let $\mathbf{d}$ be the difference of two spatial vectors linked to the states S_b and S_c. Then, $\mathbf{d}$ is defined as an *elemental vector* of order p, $\mathbf{e}_p$, *if and only if* the spatial vectors generating the difference vector differ only in the p^{th} bit position of their respective *source vector segments*. That is if,

$$\begin{aligned} \mathbf{d} &\sim 0.5(S_b - S_c) \quad \textit{where} \\ S_b &\sim \mathbf{H}[a_L, ..., +a_p, ..., a_0]' \\ S_c &\sim \mathbf{H}[a_L, ..., -a_p, ..., a_0]' \end{aligned}$$

then,
$$\begin{aligned} \mathbf{e}_p &\triangleq \mathbf{d} \\ &= a_p[h_{1p}, h_{2p}, ..., h_{Mp}]' \end{aligned} \qquad (7)$$

where $\{a_i\} \in \{-1, +1\}$.

Using the above definition as an example, consider the state S_b. Changing the sign of any one symbol, a_p in its *source vector segment* results in forming another spatial vector corresponding to a different state, S_c. Moreover, the difference vector generated between the two states will be an *elemental vector* of order p as defined above. This structure creates the basis for our estimation algorithm. If difference vectors were to be calculated with respect to a given state S_b, then at least one vector will be an *elemental vector* of order p. Thus, for a channel of length $L+1$, $L+1$ unique *elemental vectors* exist. Consequently if *difference vectors* were to be generated for the complete set of states,

$$V \triangleq \{v | v = S_i \quad i \in \{1, ..., 2^{L+1}\}\} \qquad (8)$$

each of the 2^{L+1} *spatial vectors* associated with each state will contribute $L+1$ *elemental vectors*. In other words, 2^{L+1} copies of each of the unique $L+1$ *elemental vectors* will exist in the difference vector set,

$$D \triangleq \{\mathbf{d} | \mathbf{d} = 0.5(\mathbf{y}_i - \mathbf{y}_j) \quad \{i, j\} \in \{1, ..., 2^{L+1}\} \quad i \neq j\} \qquad (9)$$

where $\mathbf{y}_i \sim S_i$ and $\mathbf{d} \sim 0.5(S_i - S_j) \Rightarrow \mathbf{d} = 0.5(\mathbf{y}_i - \mathbf{y}_j)$.

However, vectors that are not *elemental vectors*, i.e. vectors generated by states differing by more than one bit in their source vector segments will be less populous. For a difference vector resulting from q bit differences in the *source vector* segments, the maximum number of identical vectors that can be created is upper bound by

$$N_q = 2^{L-q+2} \qquad (10)$$

The *elemental vector* families in D will be more populous, and this provides the key to their identification and consequent extraction by clustering algorithms. Of the *elemental vectors*, (7) indicates that they are in fact channel coefficients. More precisely, they are columns of $\mathbf{H}$. Thus, the extracted vector set would be essentially the channel matrix, albeit having sign and permutation ambiguities.

The ambiguities results from not knowing the time order of the channel vectors extracted. Sign and permutation ambiguities can be resolved later using time data in a post processing step. Let the matrix thus extracted be denoted by $\tilde{\mathbf{H}}$. We can now summarize our algorithm as follows:

i) Use the PCA to extract an estimate of Y, to $\hat{Y}$ from the input data vectors $\mathbf{x}(n) \quad n \in \{1, ..., N\}$

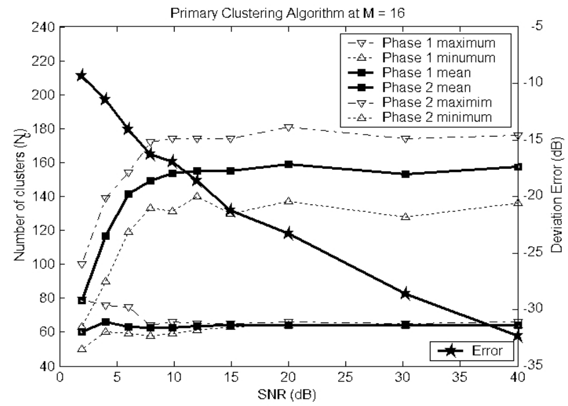

Fig. 1. Behavior of the PCA

ii) Generate the difference vector set, D from the estimated vector set $\hat{Y}$ as shown in (9).

iii) Use population relationships to extract elemental vectors by applying SCA to D.

iv) Using time data, correct sign and permutation ambiguities.

5. RESULTS AND DISCUSSION

The channel model we used in our simulations was a stochastic SIMO model, with impulse parameters modeled as zero mean Gaussian processes having unit variances. Channel coefficients and noise are assumed identically and independently distributed, and in this simulation noise was modeled as a zero mean Gaussian process. For the reference system, a channel length of $L = 6$ was selected with $M = 16$ receivers, and the results obtained using a data set of $N = 2000$ samples per iteration. Finally the results obtained were then averaged over 30 Monte-Carlo iterations.

In Fig. 1, we illustrate the behavior of the PCA algorithm. There, we plot the maxima, minima and average of the clusters extracted in the first and second phase from a Monte Carlo set of 50 iterations. The graph shows that the average clusters output by the PCA algorithm is relatively stable. However, the deviation error

$$E_{DEV} = \sum_{c=1}^{\min[N_C, 2^{L+1}]} \min_{j \in \{1, ..., 2^{L+1}\}} |\mathbf{y}_c - \mathbf{y}_j| \qquad (11)$$

deteriorates with SNR. From an alternate point of view, the PCA algorithm can be seen as an estimator of the channel length. This is illustrated in Fig. 2. In the figure, the solid and dotted lines outline the maxima and minima of the channel estimates. It is evident from the figure that the channel length estimate begins to deteriorate in successively higher SNRs as the length increases. Estimation then becomes problematic when the minima of a channel length crosses over the maxima of the next lower length.

The behavior of the CEDS algorithm with respect to the number of multipaths, M and data set size, N is illustrated in Fig. 3 and 4 respectively. They indicate that

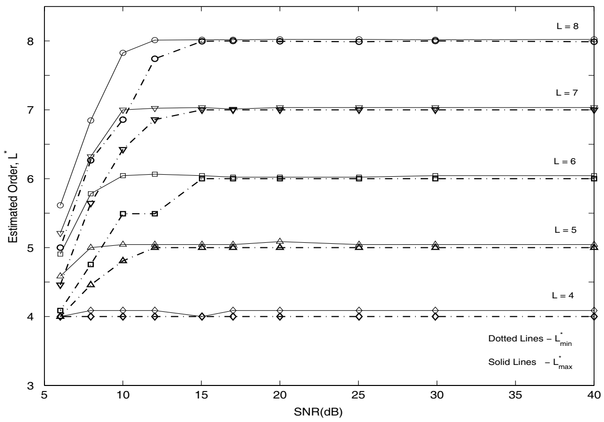

Fig. 2. Channel length estimation using PCA

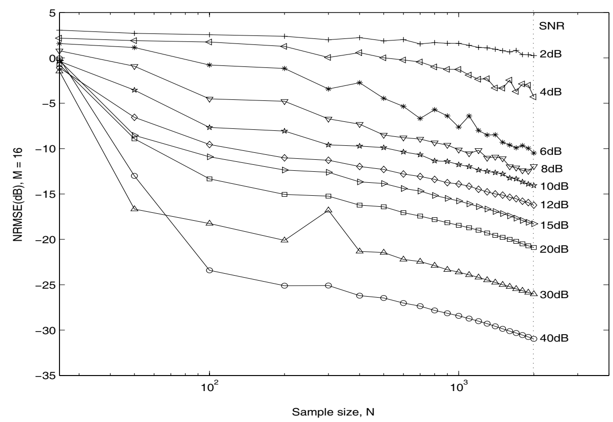

Fig. 4. Behavior of the CEDS against N

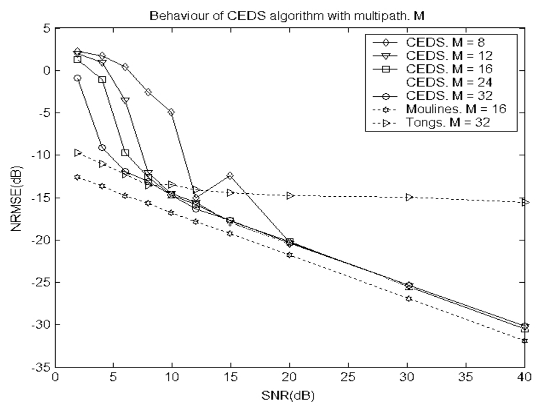

Fig. 3. Behavior of the CEDS against M

the CEDS algorithm performs acceptably in SNR's below 15 dB. However, in noisy environments its performance deteriorates. The effect noise has on the CEDS can be seen to decrease as the number of multipaths used for estimation increases. This is because then, spatial separation of the clustered centers increases and the clustering algorithms are able to coalesce clusters more successfully. Also shown in the figure are Moulines [6] and Tongs [4] algorithms. Although Moulines algorithm surpasses the CEDS in performance with about a 3dB gain, it should be kept in mind that the results of the CEDS is dependant on both the PCA and SCA algorithms. Better spatial tools may enable the CEDS to outperform [6]. Tong's algorithm outlines the behavior expected from statistical algorithms. It outperforms the CEDS in low SNRs but lags behind it in high SNRs where the CEDS can take advantage of the finite convergence property built into its deterministic nature.

6. CONCLUSION

A methodology for spatial processing leading to blind channel identification is presented in this paper. The Primary Clustering Algorithm generating both an estimate of the channel length and the noiseless spatial structure forms the core of our algorithm. Such, knowledge of the channel length is not required provided the operating SNR is known.

Additionally, it should be noted that assumption *(d)* is not used in the CEDS algorithm. Instead, it is used to recover the extracted channel matrix from sign and permutation errors. New studies into this area may result in better algorithms without such limitations.

7. REFERENCES

[1] F. Daneshgaran and M. Laddomada, "Multiscale LBG Clustering for SIMO Identification", *Proc. IEEE Int. Conf. on Comm.*,New York, NY ,pp. 84−88, May 2002.

[2] D. Godard, "Self-recovering equalization and carrier-tracking in two-dimensional data communication systems," *IEEE Trans. Commun.*, Vol. 28, pp. 1867-1875, Nov. 1980.

[3] G. Xu, H. Liu, L. Tong, and T. Kailath, "A least-squares approach to blind channel identification," *IEEE Trans. Signal Processing,* Vol. 43, pp. 2982-2993, Dec. 1995.

[4] L. Tong, G. Xu, and T. Kailath, "Blind identification and equalization based on second order statistics: A time domain approach," *IEEE Trans. Inform. Theory*, vol. 40, pp. 340-349, Mar. 1994.

[5] H. Liu, G. Xu, and L. Tong, "A deterministic approach to blind equalization", in *Conf. Rec. 1994 IEEE ICASSP Conf. on Acoust., Speech, and Signal Proc.* Vol.4, pp 581-584, April 1994.

[6] E. Moulines, P. Duhamel, J. F. Cardoso, and S. Mayrargue, "Subspace-methods for the blind identification of multichannel FIR filters," *IEEE Trans. on Acoust., Speech, and Signal Proc.,* Vol: 43, Issue: 2, pp. 516 - 525, Feb. 1995