AN ADAPTIVE SEMI-BLIND ALGORITHM FOR CHANNEL IDENTIFICATION IN OFDM

TAREQ Y. AL-NAFFOURI[1], DIMITRIOS TOUMPAKARIS[1], AHMAD BAHAI[2], and AROGYASWAMI PAULRAJ[1]

[1]Electrical Engineering Department, Stanford University, CA 94305

[2]National Semiconductor, Santa Clara, CA 95052

ABSTRACT

OFDM is a convenient vehicle for high rate transmission. However, this requires an accurate estimation of the channel at the receiver side. This paper proposes an iterative/adaptive algorithm for semi-blind channel estimation. An initial channel estimate is obtained by relying on the artificial constraint of pilots. The algorithm subsequently switches to the blind mode powered by the natural constraints imposed by the sparsity of the channel and the redundant and finite alphabet nature of the data. It iterates between using the channel estimate to detect the data, and using the data estimate to further improve the channel estimate. The diagonal nature of the OFDM channel makes it possible to optimally detect the data with low complexity. The complexity of the algorithm is further reduced by performing channel estimation adaptively. The simulation results demonstrate the favorable behavior of the algorithm and the tradeoff that it provides between the number of pilots used and convergence speed.

1. INTRODUCTION

Lately, there has been increasing interest in OFDM as it combines the advantages of high achievable rates and easy implementation. This is reflected by the many standards that have considered and adopted OFDM including those for digital audio and video broadcasting, high speed modems over digital subscriber lines, and local area wireless broadband systems. [1]

OFDM divides the communication channel into independent subchannels by appending a cyclic prefix (CP) to the data block transmitted through the channel. This turns out to be a very convenient structure that lends itself to exploiting the various constraints imposed by the transmitter and the channel. Many techniques have been proposed in literature to estimate and equalize channels for OFDM transmission (see, e.g., [1, 2, 3, 4] and the references therein). In this paper, we propose a method for semi-blind channel and data recovery. Specifically, we use the natural channel constraints and those imposed by the transmitter to perform channel and data recovery within a few packets and to reduce the number of pilots that are eventually needed.

The paper is organized as follows. After introducing our notation, we describe the essential elements of OFDM transmission in Section 2. In Section 3, we study three channels associated with OFDM transmission and describe how they can be used for channel and/or data recovery. Our findings are subsequently organized in Section 4 into a self-contained semi-blind algorithm. Simulations are presented in Section 5.

1.1. Notation

We denote scalars with small-case letters, vectors with small-case boldface letters, and matrices with uppercase boldface letters. Caligraphic notation (e.g. $\boldsymbol{\mathcal{X}}$) is reserved for vectors in the frequency domain. When these variables become a function of time, the time index i appears between parentheses for scalars (e.g. $x(i)$) and as a subscript for vectors (e.g. $\boldsymbol{h}_i$). The notation $\hat{\boldsymbol{h}}$ is used to denote an estimate of $\boldsymbol{h}$.

Now consider a length-N vector $\boldsymbol{x}_i$. We deal with three derivatives associated with this vector. The first two are obtained by partitioning $\boldsymbol{x}_i$ into an upper (prefix) vector $\underline{\boldsymbol{x}}_i$ and a lower (usually longer) vector $\tilde{\boldsymbol{x}}_i$. The third derivative, $\overline{\boldsymbol{x}}_i$, is created by concatenating $\boldsymbol{x}_i$ with a copy of its prefix $\underline{\boldsymbol{x}}_i$. The interrelations among $\boldsymbol{x}_i$ and its derivatives are summarized by the following

$$\overline{\boldsymbol{x}}_i = \begin{bmatrix} \underline{\boldsymbol{x}}_i \\ \boldsymbol{x}_i \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{x}}_i \\ \tilde{\boldsymbol{x}}_i \\ \underline{\boldsymbol{x}}_i \end{bmatrix} \tag{1}$$

This notational convention will be extended to matrices as well. Thus, a matrix $\boldsymbol{Q}$ with N rows can be partitioned as

$$\boldsymbol{Q} = \begin{bmatrix} \underline{\boldsymbol{Q}} \\ \tilde{\boldsymbol{Q}} \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{Q}}_L \\ \tilde{\boldsymbol{Q}}_{N-L} \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{Q}}_{I_1} \\ \tilde{\boldsymbol{Q}}_{I_2} \end{bmatrix} \tag{2}$$

The subscripts stand for the indicator set of the rows in the partitioned matrices or for their number. They are omitted whenever they are understood.

2. ESSENTIAL ELEMENTS OF OFDM

In OFDM, a data sequence $\{\mathcal{X}(i)\}$ is transmitted in packets $\boldsymbol{\mathcal{X}}_i$ of length N. Each packet undergoes an IDFT operation to produce the transform vector $\boldsymbol{x}_i$:

$$\boldsymbol{x}_i = \boldsymbol{Q}\boldsymbol{\mathcal{X}}_i \tag{3}$$

0-7803-7147-X/01/$10.00©2001 IEEE

where $\boldsymbol{Q}$ is the IDFT matrix

$$\boldsymbol{Q} = \left[e^{j\frac{2\pi}{N}lm}\right] \tag{4}$$

If the underlying sequence $\{x(i)\}$ is transmitted through a channel $\underline{\boldsymbol{h}}$ (which we take to be FIR of length $L+1$), it will be subject to intersymbol interference (ISI). To get around this, a guard band is inserted between any consecutive packets, $\boldsymbol{x}_{i-1}$ and $\boldsymbol{x}_i$. Specifically, to each packet $\boldsymbol{x}_i$, we append a cyclic prefix $\underline{\boldsymbol{x}}_i$ of length L as done in (1). This induces the sequence $\{\overline{x}(i)\}$ which in turn produces the sequence $\{\overline{y}(i)\}$ at the channel output. Motivated by the packet structure of the input, it is also convenient to deal with the output in the form of packets of length $M = N+L$, and further split each packet into a length-N packet $\boldsymbol{y}_i$ and a prefix associated with it $\underline{\boldsymbol{y}}_i$, i.e.

$$\overline{\boldsymbol{y}}_i = \begin{bmatrix} \underline{\boldsymbol{y}}_i \\ \boldsymbol{y}_i \end{bmatrix} \tag{5}$$

This is a natural way to partition the output because the prefix $\underline{\boldsymbol{y}}_i$ takes the burden of interference between $\overline{\boldsymbol{x}}_{i-1}$ and $\overline{\boldsymbol{x}}_i$, while the remaining part, $\boldsymbol{y}_i$, depends on the ith input packet $\overline{\boldsymbol{x}}_i$ only. These facts and more can be seen from the relationship

$$\begin{bmatrix} \boldsymbol{y}_{i-1} \\ \underline{\boldsymbol{y}}_i \\ \boldsymbol{y}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{n}_{i-1} \\ \underline{\boldsymbol{n}}_i \\ \boldsymbol{n}_i \end{bmatrix} + \tag{6}$$

$$\begin{bmatrix} \overline{\boldsymbol{H}} & \vdots & \boldsymbol{O}_{N\times L} \quad \boldsymbol{O}_{N\times N} \\ \cdots & & \cdots \\ \boldsymbol{O}_{L\times N} \quad \underline{\boldsymbol{H}}^B & \vdots & \underline{\boldsymbol{H}}^F \quad \boldsymbol{O}_{L\times N} \\ \cdots & & \cdots \\ \boldsymbol{O}_{N\times N} \quad \boldsymbol{O}_{N\times L} & \vdots & \overline{\boldsymbol{H}} \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{x}}_{i-1} \\ \tilde{\boldsymbol{x}}_{i-1} \\ \underline{\boldsymbol{x}}_{i-1} \\ \underline{\boldsymbol{x}}_i \\ \tilde{\boldsymbol{x}}_i \\ \underline{\boldsymbol{x}}_i \end{bmatrix}$$

where $\boldsymbol{n}$ is the output noise, which is assumed to be white Gaussian with variance σ_n^2. The matrices $\overline{\boldsymbol{H}}$, $\underline{\boldsymbol{H}}^B$, and $\underline{\boldsymbol{H}}^F$ are convolution (Toeplitz) matrices of proper sizes created from the vector $\underline{\boldsymbol{h}}$.

3. DECOMPOSING THE OFDM CHANNEL

Because of the redundant nature of the input, the convolution in (6) can be decomposed into two constituent convolution operations or subchannels. In the following we discuss both and demonstrate how they can be used for channel and/or data recovery.

3.1. Circular Convolution (Subchannel)

From (6), we have the subsystem of equations

$$\boldsymbol{y}_i = \overline{\boldsymbol{H}} \begin{bmatrix} \underline{\boldsymbol{x}}_i \\ \tilde{\boldsymbol{x}}_i \\ \underline{\boldsymbol{x}}_i \end{bmatrix} = \overline{\boldsymbol{H}}\,\overline{\boldsymbol{x}}_i + \boldsymbol{n}_i \tag{7}$$

This means that $\boldsymbol{y}_i$ is created solely from $\overline{\boldsymbol{x}}_i$ through convolution. Moreover, the existence of a cyclic prefix in $\overline{\boldsymbol{x}}_i$ renders this convolution cyclic:

$$\boxed{\boldsymbol{y}_i = \boldsymbol{h} \bullet \boldsymbol{x}_i + \boldsymbol{n}_i} \tag{8}$$

where $\boldsymbol{h}$ is a length-N zero-padded version of $\underline{\boldsymbol{h}}$

$$\boldsymbol{h} \triangleq \begin{bmatrix} \underline{\boldsymbol{h}} \\ \boldsymbol{O}_{(N-L-1)\times 1} \end{bmatrix} \tag{9}$$

It is best to describe this channel in the frequency domain where the the cyclic convolution (8) reduces to the element-by-element operation

$$\boxed{\boldsymbol{\mathcal{Y}}_i = \boldsymbol{\mathcal{H}} \odot \boldsymbol{\mathcal{X}}_i + \boldsymbol{\mathcal{N}}_i} \tag{10}$$

Here $\boldsymbol{\mathcal{H}}$, $\boldsymbol{\mathcal{X}}_i$, and $\boldsymbol{\mathcal{Y}}_i$ are the DFT's of $\boldsymbol{h}$, $\boldsymbol{x}_i$, and $\boldsymbol{y}_i$, respectively. In other words, we have

$$\boldsymbol{h} = \boldsymbol{Q}\boldsymbol{\mathcal{H}}, \quad \boldsymbol{x}_i = \boldsymbol{Q}\boldsymbol{\mathcal{X}}_i, \quad \text{and} \quad \boldsymbol{y}_i = \boldsymbol{Q}\boldsymbol{\mathcal{Y}}_i \tag{11}$$

From (9) and (11), we can show, using the unitary nature of $\boldsymbol{Q}$, that

$$\underline{\boldsymbol{Q}}^*_{L+1}\underline{\boldsymbol{h}} = \boldsymbol{\mathcal{H}} \tag{12}$$

This in turn enables us to write (10) in the *time-frequency form*

$$\boxed{\boldsymbol{\mathcal{Y}}_i = \operatorname{diag}(\boldsymbol{\mathcal{X}}_i)\,\underline{\boldsymbol{Q}}^*_{L+1}\underline{\boldsymbol{h}} + \boldsymbol{\mathcal{N}}_i} \tag{13}$$

3.1.1. *Mean-Square Estimation of Data*

The diagonal nature of the cyclic subchannel makes it possible to perform MMSE estimation with low complexity. Specifically, given the channel $\boldsymbol{\mathcal{H}}$ or an estimate of it, one can recover the input through an element-by-element operation:

$$\hat{\mathcal{X}}_i^{\text{MMSE}}(l) = \frac{\sqrt{\mathcal{E}_X}\mathcal{H}^*(l)}{\mathcal{E}_X|\mathcal{H}(l)|^2 + \sigma_n^2}\mathcal{Y}_i(l) \tag{14}$$

However, this represents the best *linear* estimate of $\mathcal{X}_i(l)$ given $\mathcal{Y}_i(l)$ which becomes truly optimal only when $\mathcal{X}_i(l)$ and $\mathcal{Y}_i(l)$ are jointly Gaussian. This is not the case for data transmission where $\mathcal{X}_i(l)$ is drawn from a finite alphabet. The diagonal nature of the channel actually enables us to pursue truly optimal estimation without compromising complexity. The optimum estimate of $\mathcal{X}_i(l)$ given $\mathcal{Y}_i(l)$ is given by the conditional expectation $E[\mathcal{X}_i(l)|\mathcal{Y}_i(l)]$ [5], which calls for calculating the conditional pdf $f[\mathcal{X}_i(l)|\mathcal{Y}_i(l)]$. Assuming that $\mathcal{X}_i(l)$ takes its values from the alphabet $A = \{A_1, A_2, \ldots, A_{|A|}\}$ with equal probability, it is straight forward to show that

$$f[\mathcal{X}_i(l)|\mathcal{Y}_i(l)] = \frac{e^{-\frac{|\mathcal{Y}_i(l)-\mathcal{H}(l)\mathcal{X}_i(l)|^2}{\sigma_n^2}}}{\sum_{j=1}^{|A|} e^{-\frac{|\mathcal{Y}_i(l)-\mathcal{H}(l)A_j|^2}{\sigma_n^2}}} \tag{15}$$

and subsequently that

$$\hat{\mathcal{X}}_i^{\mathsf{MMSE}}(l) = E\left[\mathcal{X}_i(l)|\mathcal{Y}_i(l)\right] = \frac{\sum_{j=1}^{|A|} A_j e^{-\frac{|\mathcal{Y}_i(l)-\mathcal{H}(l)A_j|^2}{\sigma_n^2}}}{\sum_{j=1}^{|A|} e^{-\frac{|\mathcal{Y}_i(l)-\mathcal{H}(l)A_j|^2}{\sigma_n^2}}} \quad (16)$$

The estimate gets eventually rounded to the nearest alphabet

$$\hat{\mathcal{X}}_i = \lfloor \hat{\mathcal{X}}_i^{\mathsf{MMSE}} \rfloor \quad (17)$$

3.1.2. Using Pilots for Initial Channel Estimation:

The circular convolution channel can be used for the dual purpose of channel estimation from known input. Although there are N elements in the frequency response $\mathcal{H}$, they have at most $L+1$ degrees of freedom controlled by the corresponding elements of $\underline{h}$ (see (12)). Thus, instead of identifying $\mathcal{H}$ from (10), we attempt to identify $\underline{h}$ from (13). To this end, let $I_p = \{i_1, i_2, \ldots, i_L\}$ denote the index set of the pilot bins. These pilots yield a subsystem of (13) in the IR $\underline{h}$

$$\begin{aligned}(\mathcal{Y}_i)_{I_p} &= \left(\mathrm{diag}\,(\mathcal{X}_i)\,\underline{Q}^*_{L+1}\right)_{I_p} \underline{h} + (\mathcal{N}_i)_{I_p} && (18)\\ &= A_{I_p}\underline{h} + (\mathcal{N}_i)_{I_p} && (19)\end{aligned}$$

where

$$A \triangleq \mathrm{diag}\,(\mathcal{X}_i)\,\underline{Q}^*_{L+1}$$

The subscript I_p in A_{I_p} is an indicator set of the rows of A_{I_p} as a submatrix of A. The system (19) can be solved uniquely in the LS sense when $L_p \geq L+1$. We can also reduce the overhead of pilots below $L+1$ and rely on the more natural channel and transmitter induced constraints [1] to identify the channel uniquely. In this case, however, the system of equations (19) becomes underdetermined and uniqueness can be restored by solving the (α-) *regularized* LS problem instead

$$\min_{\underline{h}}\ \alpha\|\underline{h}\|^2 + \|\mathcal{Y} - A\underline{h}\|_W^2 \quad (20)$$

where $W = \frac{1}{\sigma_n^2} I$ is the inverse covariance matrix of the noise. The solution of (20) is given by

$$\hat{\underline{h}} = \left(\alpha I + A_{I_p}^* W A_{I_p}\right)^{-1} A_{I_p}^* W (\mathcal{Y}_i)_{I_p} \quad (21)$$

3.2. Linear Convolution (Subchannel)

From (6), one can deduce the following relationship involving input and output prefixes

$$\underline{y}_i = \left[\ \underline{H}^B \quad \underline{H}^F\ \right] \begin{bmatrix} \underline{x}_{i-1} \\ \underline{x}_i \end{bmatrix} + \underline{n}_i \quad (22)$$

[1] The maximum delay spread is the only channel constraint that was used here. Additional sparsity constraints like the exact channel order and the location of the active (nonzero) taps can also be incorporated to improve convergence speed.

This shows that the underlying output prefix sequence $\{\underline{y}(i)\}$ can be obtained from the input prefix sequence $\{\underline{x}(i)\}$ through linear convolution with the channel, i.e.

$$\boxed{\underline{y}(i) = \underline{h}(i) * \underline{x}(i) + \underline{n}(i)} \quad (23)$$

This input/output relationship can be used for channel identification. Thus, given an estimate of the input, obtained, say, through frequency domain equalization, one can obtain a corresponding estimate of $\underline{x}_i$. This estimate, together with the output prefix, can be used to track channel variations (see [6]). However, since the channel and prefix lengths are of the same order, the identification algorithm requires a large number of prefixes (and hence packets) to converge. Convergence speed can be substantially increased if the whole packet $\overline{x}_i$ (including its cyclic prefix part) is used for channel estimation, as we will now explain.

3.3. Total (Linear) Convolution

The sequence $\{\overline{y}(i)\}$ at the channel output is related naturally to the input sequence $\{\overline{x}(i)\}$ through linear convolution with the channel

$$\overline{y}(i) = h(i) * \overline{x}(i) + \overline{n}(i) \quad (24)$$

Now, in line with the notation adopted here, define $\overline{h}$ to be another zero-padded version of h of length $N+L$ [2]

$$\overline{h} \triangleq \begin{bmatrix} \underline{h} \\ O_{(N-1)\times 1} \end{bmatrix} = \begin{bmatrix} h \\ O_{L\times 1} \end{bmatrix}$$

Zero padding does not affect linear convolution, and the output sequence $\{\overline{y}(i)\}$ is still related to $\{\overline{x}(i)\}$ though linear convolution with $\overline{h}$. Thus, the convolution (24) can be written in the more consistent form

$$\boxed{\overline{y}(i) = \overline{h}(i) * \overline{x}(i) + \overline{n}(i)} \quad (25)$$

Given the output sequence $\{\overline{y}(i)\}$ and the input sequence $\{\overline{x}(i)\}$ or an estimate of it, the total linear convolution can now be used for channel identification. This can be achieved adaptively, e.g. with the LMS algorithm:

$$\begin{aligned}\hat{\underline{h}}_{i+1} &= \hat{\underline{h}}_i + \mu e(i)\overline{x}_i^* && (26)\\ e(i) &= \overline{y}(i) - \overline{x}_i\hat{\underline{h}}_i && (27)\\ \overline{x}_i &= \left[\ \overline{x}(i) \ \cdots \ \overline{x}(i-L+1) \quad \overline{x}(i-L)\ \right] && (28)\end{aligned}$$

where $\hat{\underline{h}}_i$ is the estimate of $\underline{h}$ at time i. Note that the row vector $\overline{x}_i$ stands *not* for the packet transmitted at time i but rather for the shift regressor vector obtained from the time sequence $\{\overline{x}(i)\}$.

Remark: The boxed relationships (8), (10), (23) and (25) demonstrate how our notation blends smoothly with the nature of the OFDM problem, so much so that these relationships could be written almost by inspection.

[2] Recall that h, defined in (9), is too a zero-padded version of $\underline{h}$ of length-N.

4. SEMI-BLIND ALGORITHM FOR CHANNEL ESTIMATION

The above developments can be organized into a semi-blind algorithm for channel estimation

- **A priori information**
  - Noise variance σ_n^2 and pilot locations (indexed by I_p)
- **Initial channel estimation**
  - Use pilots to find the LS estimate of the channel, $\hat{\underline{h}}$ (21). Set $W = \frac{1}{\sigma_n^2} I$
- **Detection, estimation, and iterative refinement of estimates** For each incoming packet $\mathcal{X}_i$, perform
  - MMSE detection:
    - Find frequency response estimate $\hat{\mathcal{H}}$ (12)
    - Perform MMSE detection $\hat{\mathcal{X}}_0^{\text{MMSE}} \ldots \hat{\mathcal{X}}_i^{\text{MMSE}}$ (linear (14) or optimal (16))
    - Round to the nearest alphabet point $\hat{\mathcal{X}}_0 \ldots \hat{\mathcal{X}}_i$ (17)
    - Obtain time domain blocks $\hat{x}_0 \ldots \hat{x}_i$ (11)
  - Adaptive channel estimation
    - Employ the LMS to identify channel adaptively (26)-(28)
  - Iterative refinement
    - Repeat as long as estimates change

5. SIMULATIONS

We consider OFDM PAM transmission with packet length $N = 128$ and cyclic prefix length $L = 15$. The channel is FIR of length 16 (which requires 16 pilots to identify in the noiseless case). Pilots are transmitted in the first packet only, and their number is varied between 4 and 16 (i.e. between 25% and 100% of what is required for identifiability in the noiseless case). The SNR is fixed at 11 dB. In the algorithm, adaptation is performed over 5 packets with 4 iterations per packet for a total of 20 iterations. We use three setups to demonstrate the various aspects of our algorithm, plotting in each the mean-square error $\|\hat{\underline{h}}_i - \underline{h}\|^2$ vs the number of iterations.

Linear vs. optimum MMSE estimation: We begin by demonstrating the advantage of employing optimum vs. linear estimation on the performance of our algorithm. We use only 8 pilots and take the input to be 4-PAM. The two methods are compared in figure 1. As expected, optimal estimation exhibits lower mean-square error.

Adaptation using the whole packet vs CP only: We proceed by showing the advantage of using the whole packet for channel identification over using only part of it (e.g. using the CP part as done in [6]). The input here is taken to be 2-PAM.

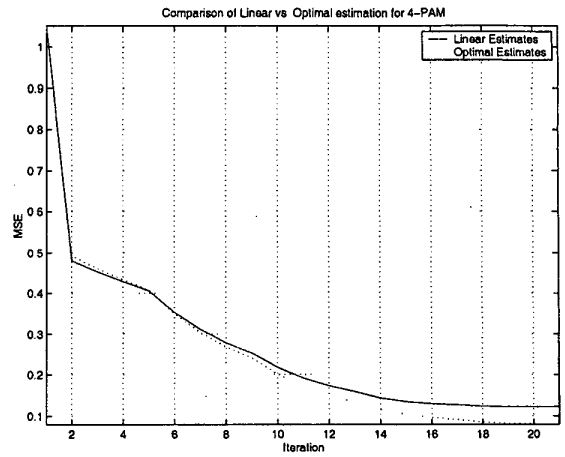

Figure 1: Optimal vs. linear MMSE estimation for 4-PAM and 8 pilots

The two methods are compared in Fig. 2 where estimation is initialized with 8 pilots. Fig 3 compares the two methods when 16 pilots are used.

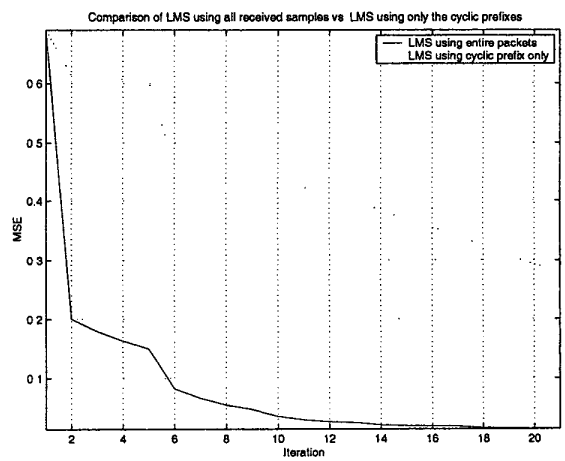

Figure 2: Adaptation using the whole packet vs. CP only for 2-PAM and 8 pilots

Tradeoff between number of pilots and convergence speed: Fig. 4 exhibits the trade off between the number of pilots and convergence speed. Increasing the number of pilots improves the convergence speed and reduces the steady-state estimation error.

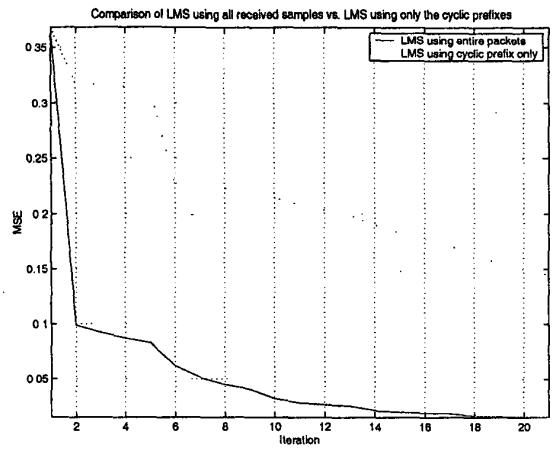

Figure 3: Adaptation using the whole packet vs. CP only for 2-PAM and 16 pilots

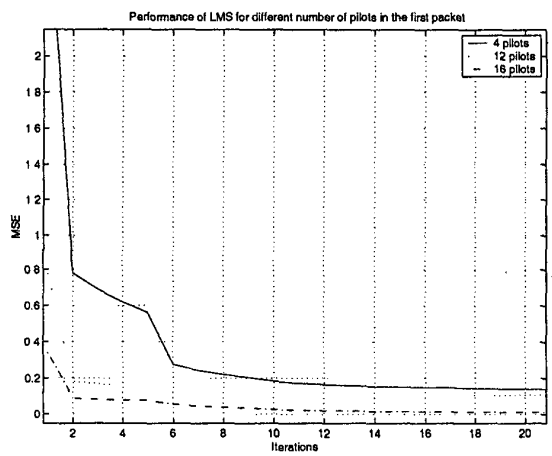

Figure 4: Comparison of convergence speed for different number of pilots in the first packet for 2-PAM input

6. CONCLUSION

In this paper, we introduced a semi-blind algorithm for channel identification in OFDM. The algorithm uses a few pilots to initialize the estimation process and subsequently utilizes the natural constraints imposed by the channel and the transmitter to identify the channel completely. The iterative nature of the algorithm enhances the quality of the estimates, and its adaptive nature serves to reduce the overall complexity. By incorporating some reliability information about the channel and data estimates, one could further improve the robustness and speed of the proposed algorithm.

7. REFERENCES

[1] Z. Wang and G. B. Giannakis, "Wireless multicarrier communications: Where Fourier meets Shannon," *IEEE Signal Process. Mag.*, vol. 17, No. 3, pp. 29–48, May 2000.

[2] R. W. Heath Jr. and G. B. Giannakis, "Exploiting input cyclostationarity for blind channel identification in OFDM systems," *IEEE Trans. Signal Process.*, vol 47, pp. 774–780, Mar. 1999.

[3] B. Muquet, M. de Courville, P. Duhamel, and V. Buzenae, "A subspace based blind and semi-blind channel identification method for OFDM systems," *Proc. IEEE Workshop for Signal Process. Advances in Wireless Commun.*, Annapolis, MD, pp. 243–246. May 1999.

[4] S. Zhou and G. B. Giannakis, " Finite-alphabet based channel estimation for OFDM and related multi-Carrier systems," *IEEE Trans. Commun.*, vol. 49, no. 8, pp. 1402–1414, Aug. 2001.

[5] T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, 2000.

[6] X. Wang and R. Liu, "Adaptive channel estimation using cyclic prefix in multicarrier modulation system," *IEEE Commun. Lett.*, vol. 3, no. 10, pp. 291–293, Oct. 1999.