Linear Prediction Methods for Blind Fractionally Spaced Equalization

Xiaohua Li and H. (Howard) Fan, Senior Member, IEEE

Abstract—In this paper, we describe adaptive methods for estimating FIR zero-forcing blind equalizers with arbitrary delay directly from the linear predictions of the observations. While most current methods require inversion or singular value decomposition (SVD) of the correlation matrix, our methods need only to solve two linear prediction problems. They can be implemented as RLS or LMS algorithms to recursively update the equalizer estimation. They are computationally efficient. The computational complexity in each recursion can be less than $15(LN)^2$ in the RLS case, where $LN$ equals the equalizer length, and $3L(LN)$ in the LMS case, where $L$ is the number of subchannels. Performance of the proposed methods and comparisons with existing approaches are shown by simulation to demonstrate their effectiveness.

Index Terms—Adaptive equalizers, blind equalization, intersymbol interference, linear prediction.

I. INTRODUCTION

ANY digital communication systems suffer from the problem of intersymbol interference (ISI), which may arise from the common phenomenon of multipath propagation, for example. To achieve reliable communication in these situations, channel equalization is necessary to eliminate ISI. Traditional equalization methods are based on training sequences or a priori knowledge of the channel [8], [10]. In many applications, for example, wireless communications, these approaches are often not suitable or not cost effective.

Blind equalization of transmission channels is important in many communication and signal processing applications because it does not require training sequences, nor does it require a priori channel knowledge. Instead, the known statistical properties of the transmitted signals are exploited to carry out the equalization at the receiver. Since blind equalizers do not require extra bandwidth for training, they have aroused much interest and resulted in great research activities.

Traditionally, higher (than second) order statistics of the symbol rate sampled channel outputs are used to estimate the channel and to calculate the equalizer. More recently, it has been shown that the second-order statistics contain sufficient information for the identification and equalization of FIR channels using cyclostationarity of the channel output with fractionally spaced equalizers [5]–[7]. Based on the seminal work in [5], many effective blind methods have been proposed for estimating the channel from the output-only second-order statistics. Each of these methods provides a blind estimation of the channel that can then be used to find the transmitted sequence. However, these algorithms usually require singular value decomposition (SVD) or eigenvalue decomposition (EVD) of the output correlation matrix. The computational burdens for SVD or EVD turn out to be a major obstacle to real-time implementation. In [15], approaches based on QR decomposition and ULV subspace tracking is proposed to reduce computations. Furthermore, it turns out that these SVD- or EVD-based algorithms are very sensitive to the channel length estimation or the rank estimation of the data correlation matrix. Our experience shows that the residual ISI or the residual output mean square error may be significantly higher if the rank estimation is off even by one. In a practically noisy environment, accurate rank determination may be difficult.

Another approach is to directly estimate a linear filter that can remove the ISI and/or suppress the additive noise without channel identification, such as the direct equalizer estimation method [1] or the linear prediction-based methods [3], [4]. The channel identification can also be calculated from this filter [2]. The direct estimation of the equalizer is computationally efficient and lends itself easily to the development of adaptive methods for tracking time-varying channels. In [2] it is shown that this approach is robust to overestimation of the channel order.

However, the algorithms in [2] and [3] calculate zero-forcing (ZF) equalizers with zero-delay only, and the result is not satisfactory. In many situations, a ZF equalizer with nonzero delay may give better results [9] because the noise enhancement of the ZF equalizer and the variance of the equalizer estimation may depend on the delay. A method for computing nonzero delay ZF equalizers was proposed in [4], which applies two stages of linear prediction where the input of the second stage is the output of the first one. The output of the first stage contains transient response, steady-state errors, as well as computation errors, which render the second stage not so reliable. This method also needs to know the channel tap with the largest magnitude, which is usually not known in practical situations.

Another linear prediction like method [on-line mutually referenced equalizers (MRE)] is presented in [18]. The linear prediction algorithm of [18] is used to make the MRE method recursive and computationally efficient. However, it can only use a linear constraint, and the first entry of this constraint has to be fixed to 1. This may not work for all situations and may not be the optimal one either. Hence, its linear prediction implementation may not achieve the optimal performance. In addition,
the linear prediction is performed on some modified data vectors, which we find is more sensitive to additive noise than other linear prediction algorithms performed directly on the sampled data vector. Furthermore, the large dimension of the data vector (larger than others by an order) makes it computationally much more complex and slower in convergence.

An approach for directly estimating nonzero delay ZF equalizer was also given in [1]. However, the algorithms in [1] and [2] need to compute explicitly the pseudoinverse of the correlation matrix. Because, in the fractional space, the correlation matrix of the sampled data (noiseless) is not full rank, there is a rank determination problem and the computation is not simple. An RLS-like recursive algorithm is developed in [1] to update this pseudoinverse recursively. However, because of the rank deficiency, it is very sensitive to the initialization and, thus, not reliable in real applications.

In this paper, we develop algorithms for estimating nonzero delay ZF equalizers based completely on linear prediction to avoid the problem of pseudoinverse of the correlation matrix. In fact, we need not to compute the correlation matrix at all. Our algorithms will be based on the sampled data vectors directly and require less computations compared with the MRE method. Based on the equalizer, we can also identify the channel.

The organization of this paper is as follows. In Section II, we will formulate the problem and give a modified version for the MMSE equalizer algorithm of [1]. In Section III, we will develop linear prediction algorithms for channel equalization and identification. Some simulation results and comparison of our algorithms with some typical existing algorithms are presented in Section IV.

II. MULTICHANNEL LINEAR PREDICTION ERROR AND EQUALIZATION

A. Problem Formulation

Consider a continuous-time communication system

$$y(t) = \sum_{k=-\infty}^{\infty} s_k h_c(t - kT) + v_c(t)$$  (2.1)

where $s_k$ denotes the symbol emitted by the digital source at time $kT$ with $T$ being the symbol duration. $h_c(t)$ denotes the continuous-time channel, which is assumed to have finite support. $v_c(t)$ is additive noise that is assumed to be stationary as well as uncorrelated with $s_k$.

The corresponding fractionally spaced discrete time model can be obtained either by sampling the signal received on several sensors at the symbol duration $T$, or by oversampling the signal received on a single sensor, or by combining both techniques [6].

We assume the following throughout this paper.

i) The input sequence $s_k$ is stationary with zero mean, and $E[s_k s_{k-l}^*] = \sigma_k^2 \delta(k - l)$.

It is convenient to write the above equation as an equivalent discrete-time system

$$y(n) = \sum_{k=-\infty}^{\infty} s_k h(n - kL) + v(n) = x(n) + v(n).$$  (2.3)

The output $y(n)$ has periodically time-varying correlation with period $L$. In many cases, periodically correlated signals are conveniently represented by a vector stationary process. Define $y_k(n) = y(nL+i), h_k(n) = h(nL+i)$, and $v_k(n) = v(nL+i)$. The single-input single-output system of (2.3) has an equivalent single-input multiple-output description as

$$y_k(n) = \sum_{k=-\infty}^{\infty} s_k h_k(n - k) + v_k(n), \quad i = 0, 1, \ldots, L-1,$$  (2.4)

Letting

$$y(n) = \begin{bmatrix} y_0(n) \\ \vdots \\ y_{L-1}(n) \end{bmatrix}, \quad h(n) = \begin{bmatrix} h_0(n) \\ \vdots \\ h_{L-1}(n) \end{bmatrix}, \quad v(n) = \begin{bmatrix} v_0(n) \\ \vdots \\ v_{L-1}(n) \end{bmatrix},$$

we represent $y(n)$ in a vector form as

$$y(n) = \sum_{k=-\infty}^{\infty} s_k h_k(n - k) + v(n).$$  (2.5)

If $h(t)$ is an FIR of order $L_h$, i.e., $h(t)$ has support $t \in [0, L_h]$, then the subchannels $h_k(n)$ will be of order $L_h$. We suppose $h_0(0) \neq 0$. The system can be represented in the matrix form as

$$y_N(n) = \mathcal{H} s(n) + v_N(n).$$  (2.6)

where $\mathcal{H}$ is a $NL \times (N + L_h)$ block Toeplitz matrix, $s(n)$ is $(N + L_h) \times 1$, $v_N(n)$, and $y_N(n)$ are $NL \times 1$ vectors

$$\mathcal{H} = \begin{bmatrix} h(0) & h(1) & \cdots & h(L_h) & \cdots & 0 \\ 0 & h(0) & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & h(0) & \cdots & h(L_h) \end{bmatrix}, \quad s(n) = \begin{bmatrix} s_n \\ s_{n-1} \\ \vdots \\ s_{n-L_h} \\ \vdots \end{bmatrix}, \quad y_N(n) = \begin{bmatrix} y(n) \\ y(n-1) \\ \vdots \\ y(n-N+1) \end{bmatrix}, \quad v_N(n) = \begin{bmatrix} v(n) \\ v(n-1) \\ \vdots \\ v(n-N+1) \end{bmatrix}.$$  (2.7)

The model (2.6) can also be obtained by sampling signals from several sensors, where $L$ is now the number of sensors. For the $i$th sensor, $h_i(n), n = 0, \ldots, L_h$ are the subchannel coefficients [6].

We assume the following throughout this paper.

i) The input sequence $s_k$ is stationary with zero mean, and $E[s_k s_{k-l}^*] = \sigma_k^2 \delta(k - l)$. 

...
ii) The noise \( v(t) \) is stationary with zero mean and white with variance \( \sigma_v^2 \).

iii) \( s \) & \( v \) are uncorrelated.

### B. Zero-Forcing Equalizers

Consider the FIR linear equalizer shown in Fig. 1, where \( g_k(n) \) for \( i = 0, 1, \ldots, L - 1 \) is the order \( L_g \) equalizer of the \( i \)th subchannel. In the absence of noise, one natural choice is to require \( \hat{s}_k = s_{n-k} \) for some integer delay \( d \). This type of equalizer is known as zero-forcing (ZF) [9]. A ZF equalizer whose subchannels are order \( L_g \) is described by

\[
\sum_{i=0}^{L-1} \sum_{k=0}^{L_k} h_i(k) g_l^{(d)}(n-k) = \delta(n-d)
\]  

(2.9)

where the superscript \( (d) \) refers to the delay \( d \in [0, L_h + L_g] \).

Letting \( N = L_g + 1 \), then (2.9) can be written in the matrix form as

\[
H^T g_d = e_{d+1}
\]  

(2.10)

where \((\cdot)^T\) denotes transpose, and \( g_d \) is the \((L_g + 1) \times 1 \) vector of equalizer taps corresponding to delay \( d \), as shown at the bottom of the page. \( e_{d+1} \) is a \((L_h + L_g + 1) \times 1 \) vector with a 1 as the \((d+1)\)st element and zeros elsewhere.

The existence of the ZF equalizer \( g_d \) is proved in [1]–[4] as the following theorem.

**Theorem 1:** Assume that the subchannels \( \{h_i(n)\} \) have no common roots. An FIR ZF equalizer with subchannels of order \( L_g \) exists, provided \( L_h \geq L_g - 1 \).

We consider the noise-free case first. The correlation matrix of \( y_n(n) \) of (2.6) for \( N = L_g + 1 \) is

\[
R = E \left\{ y_N y_N^H \right\} = H H^H
\]  

(2.11)

where \((\cdot)^H\) stands for conjugate transpose, and \( R_s = E \{ s(n) s^H(n) \} = \sigma_s^2 I \). Therefore, we have

\[
R = \sigma_s^2 H H^H.
\]  

(2.12)

From (2.10), we have, for the zero-delay equalizer \((d = 0)\)

\[
g_0^T R = \sigma_s^2 [h^H(0) \ 0].
\]  

(2.13)

Note that \([h^H(0) \ 0]^H\) is the first column of matrix \( H \). Thus, the zero-delay equalizer can be estimated by solving (2.13)

\[
g_0^T = \sigma_s^2 [h^H(0) \ 0] R^+
\]  

(2.14)

where \((\cdot)^+\) denotes pseudoinverse.

Considering that an equalizer with nonzero delay may be more satisfactory than the zero-delay equalizer, it is proved in [1] that an equalizer \( g_d \) with delay \( d \) can be obtained from \( g_0 \) by

\[
g_d^T = g_0^T R_d R^+
\]  

(2.15)

where \( R_d = E \{ y_N(n-d) y_N^H(n) \} \). A delay \( d \) MMSE equalizer is also developed in [1] and turns out to be identical to (2.15) with noise-free \( R \) replaced by \( R \) with noise.

### C. Multichannel Linear Prediction Error

\( h(0) \) in (2.13) is an \( L \times 1 \) vector consisting of the first element of each subchannel and is not known \( a \) priori. A method to estimate \( h(0) \) is by multichannel linear prediction [2]–[4]. Consider the following linear prediction problem:

\[
e(n) = [I_L - P_{N-1} y_N(n)]
\]  

(2.16)

where the prediction error \( e(n) \) is an \( L \times 1 \) vector, \( P_{N-1} \) is an \( L \times L(N-1) \) matrix, and \( I_L \) is an \( L \times L \) identity matrix.

Minimizing the prediction error variance leads to the following optimization problem:

\[
\min_{P_{N-1}} f(P_{N-1})
\]  

(2.17)

**Theorem 2:** Suppose that \( P_{N-1} \) is the optimum solution to (2.17). Assuming the subchannels \( \{h_i(n)\} \) have no common roots and \( N \geq L_h \), then

\[
[I_L - P_{N-1}] H = h(0) [1 \ 0 \ \cdots \ 0]
\]  

(2.18)

\[
\min_{P_{N-1}} E \{ e(n) e^H(n) \} = \sigma_s^2 h(0) H^H(0).
\]  

(2.19)

**Proof:** Similar results are given in [4] without proof. Therefore, we present our proof. Since the subchannels have no common roots and \( N \geq L_h \), then \( H \) is column full rank [6]. From (2.12), the minimization problem becomes

\[
\min_{P_{N-1}} f(P_{N-1})
\]  

(2.20)

Define

\[
\begin{bmatrix}
H_1 \\
H_2
\end{bmatrix} = H
\]  

(2.21)

\[
g_d = \begin{bmatrix}
g_{(d)}(0) & \cdots & g_{(d)}(L_g) & \cdots & g_{(d)}(L_{L-1}(L_g))
\end{bmatrix}^T.
\]  

(2.22)
where \( \mathbf{H}_1 \) is \( L \times (N + L_h) \), and \( \mathbf{H}_2 \) is \( (N - 1) \times (N + L_h) \). Then, (2.20) becomes

\[
\begin{align*}
\mathbf{f}(\mathbf{P}_{N-1}) &= \text{tr}\left\{ (\mathbf{H}_1 - \mathbf{P}_{N-1}\mathbf{H}_2)(\mathbf{H}_1^H - \mathbf{H}_2^H \mathbf{P}_{N-1}^H) \right\} \\
&= \text{tr}\left\{ \mathbf{H}_1 \mathbf{H}_1^H - \mathbf{P}_{N-1}\mathbf{H}_1 \mathbf{H}_2^H - \mathbf{H}_1 \mathbf{H}_2^H \mathbf{P}_{N-1}^H + \mathbf{P}_{N-1}\mathbf{H}_2 \mathbf{H}_2^H \mathbf{P}_{N-1}^H \right\}.
\end{align*}
\]

Let \( \mathbf{p}_i \) be the \( i \)th row of \( \mathbf{P}_{N-1} \), \( \mathbf{a}_i \) be the \( i \)th column of \( \mathbf{H}_2 \mathbf{H}_2^H \), and then, let

\[
\frac{\partial \mathbf{f}(\mathbf{P}_{N-1})}{\partial \mathbf{p}_i} = -\mathbf{a}_i^H + \mathbf{p}_i \mathbf{H}_2 \mathbf{H}_2^H.
\]

Therefore

\[
\frac{\partial \mathbf{f}(\mathbf{P}_{N-1})}{\partial \mathbf{P}_{N-1}} = -\mathbf{H}_1 \mathbf{H}_1^H + \mathbf{P}_{N-1} \mathbf{H}_2 \mathbf{H}_2^H.
\]

Let the above derivative equal \( \mathbf{0} \), and then, let

\[
(\mathbf{H}_1 - \mathbf{P}_{N-1}\mathbf{H}_2)\mathbf{H}_2^H = \mathbf{0}.
\]

From (2.7), we find that the first column of \( \mathbf{H}_2 \) is zero, and the other columns consist of a submatrix of full column rank under the assumptions [6]. Hence, except for the first column, the other columns of matrix \( (\mathbf{H}_1 - \mathbf{P}_{N-1}\mathbf{H}_2) \) are all zero. Therefore, we get (2.18).

Equation (2.19) can be deduced from (2.18) easily because

\[
\begin{align*}
\min_{\mathbf{P}_{N-1}} E\{e(n)^2/n(n)\} &= \sigma_e^2\mathbf{I}_L - \mathbf{P}_{N-1} \\
&\times \mathbf{H}^H[\mathbf{I}_L - \mathbf{P}_{N-1}].
\end{align*}
\]

From (2.19), we know that any column of the matrix \( \min_{\mathbf{P}_{N-1}} E\{e(n)^2/n(n)\} \) is proportional to \( \mathbf{h}(0) \). Hence, \( \mathbf{h}(0) \) can be estimated from the prediction error by (2.19). We can use it in computing the MMSE equalizer according to (2.14) and (2.15). We will denote it as modified MMSE equalizer.

III. LINEAR PREDICTION METHODS FOR BLIND EQUALIZATION

A. Nonzero Delay Equalizer and Linear Prediction

We have developed the modified MMSE equalizer by linear prediction error and correlation matrices. The modified MMSE equalizer and the existing algorithms in [1] and [2] require the explicit calculation of the pseudoinverse of the correlation matrix \( \mathbf{R} \). Since \( \mathbf{R} \) may be rank deficient (under high SNR), the inversion is very computationally complex, and rank determination plays too critical a role.

The recursive pseudoinverse matrix updating method is proposed in [1]. Because of the rank deficiency problem of \( \mathbf{R} \), this approach may not be reliable. Our simulations demonstrate that the estimation of the eigenvalues of the noise subspace sensitively depend on the initialization. Furthermore, under finite precision implementations, the inversion of noise eigenvalues may result in large error. Therefore, this approach may suffer numerical problems, and the equalization result may degrade greatly. Some sets of data have to be used in [1] to make an off-line estimation of the initialization values. This procedure is not computationally simple, nor is it convenient in blind equalization of time-varying channels.

Although a cyclic LMS algorithm is presented in [1], which alleviates this problem to some extent, however, the LMS part can only obtain the zero-forcing equalizer with zero delay. In order to compute the equalizers with nonzero delay, we again have to compute explicitly the pseudoinverse of the correlation matrix. The problem of [1] remains. Therefore, it is better to get rid of the dependence of the solution on explicit computation of the correlation matrix and its pseudoinverse.

Our goal is to develop algorithms for estimating equalizers completely based on linear prediction. By recursive linear prediction, we do not have to compute explicitly the correlation matrix and its pseudoinverse, although the adaptive algorithms converge finally to the optimal solution represented theoretically by the pseudoinverse matrix.

A zero-delay ZF equalizer based completely on linear prediction is presented in [2]–[4]. From (2.12), (2.13), and (2.18), we know that

\[
\mathbf{g}_0^T = \mathbf{h}^H(0)[\mathbf{I}_L - \mathbf{P}_{N-1}]
\]

is a zero-delay ZF equalizer.

We now show that the \( d \)-delay equalizer \( \mathbf{g}_d \) in (2.15) can also be estimated by linear prediction. Consider the following linear prediction problem:

\[
\begin{align*}
\min_{\mathbf{P}_N} &\text{tr}\left\{ E\{\mathbf{J}(n)\mathbf{J}^H(n)\} \right\} \\
&\text{subject to } \mathbf{J}(n) = \mathbf{y}_N(n-d) - \mathbf{P}_N \mathbf{y}_N(n) \quad (3.2)
\end{align*}
\]

where \( \mathbf{J}(n) \) is an \( LN \times 1 \) prediction error vector, and \( \mathbf{P}_N \) is an \( LN \times LN \) projection matrix. We have the following theorem.

**Theorem 3:** If \( \mathbf{P}_N \) is the optimal solution to (3.2), then the \( d \)-delay ZF equalizer \( \mathbf{g}_d \) can be computed from the zero-delay ZF equalizer \( \mathbf{g}_0 \) by

\[
\mathbf{g}_d^T = \mathbf{g}_0^T \mathbf{P}_N. \quad (3.3)
\]

**Proof:** The optimal \( \mathbf{P}_N \) is obtained by minimizing

\[
\begin{align*}
&\text{tr}\{ E\{\mathbf{J}(n)\mathbf{J}^H(n)\} \} \\
&= \text{tr}\{ E\{[\mathbf{y}_N(n-d) - \mathbf{P}_N \mathbf{y}_N(n)]\mathbf{J}^H(n)\} \} \quad (3.4)
\end{align*}
\]

Letting the derivative of (3.4) with respect to \( \mathbf{P}_N^H \) equal zero, we get, in a similar procedure as the proof of Theorem 2

\[
\begin{align*}
\frac{\partial \text{tr}\{ E\{\mathbf{J}(n)\mathbf{J}^H(n)\} \}}{\partial \mathbf{P}_N} &= E\{ -\mathbf{y}_N(n-d)\mathbf{y}_N^H(n) + \mathbf{P}_N \mathbf{y}_N(n)\mathbf{y}_N^H(n) \} \\
&= \mathbf{0}.
\end{align*}
\]

Therefore

\[
\mathbf{P}_N \mathbf{R} - \mathbf{R}_d = \mathbf{0}, \quad (3.5)
\]

Comparing (3.5) with (2.15), we get (3.3).

B. RLS Equalizer

One major advantage of the linear prediction approach is the ability to develop computationally efficient and reliable adaptive algorithms for estimating the equalizers. The adaptive al-
Algorithms are also useful for tracking time-varying channels. Although the methods of [1] and [18] can all be implemented as adaptive algorithms, [1] requires the sensitive correlation matrix pseudoinverse computations, whereas [18] may not converge to the optimal solution. Both of them are computationally more complex and converge more slowly than what we can achieve using the conventional linear prediction of (3.1) and (3.2).

In order to achieve fast convergence, we can use the recursive least-squares (RLS) algorithm to update the linear prediction. Two linear prediction problems are involved in estimating the nonzero delay equalizers. The first is (2.16); we are required to compute the prediction filter \( \mathbf{P}_{N-1} \) and to estimate the prediction error \( \varepsilon(n) \). A zero-delay equalizer \( \mathbf{h}(0) \) is obtained in (3.1), considering that \( \varepsilon(n) \) is proportional to \( \mathbf{h}(0) \). Then, the second linear prediction problem (3.2) is computed to find \( \mathbf{P}_N \). Then, the \( d \)-delay equalizer \( \mathbf{g}_d \) is obtained [see (3.3)]. The algorithm is listed here.

- Initialize the algorithm by setting
  \[ Q_1(0) = \delta_1^{-1} \mathbf{I}_{L(N-1)}, \quad \delta_1 = \text{small positive constant} \]
  \[ \mathbf{P}_{N-1}(0) = 0 \]
  \[ Q_2(0) = \delta_2^{-1} \mathbf{I}_{LN}, \quad \delta_2 = \text{small positive constant} \]
  \[ \mathbf{P}_N(0) = 0 \]
  \[ \mathbf{F}(0) = 0. \]

- For each instant of time \( n = 1, 2, \ldots \), compute
  1) The first linear prediction problem
  \[ \mathbf{y}(n) \triangleq \begin{bmatrix} \mathbf{y}_{N,1}^H(n) \\ \mathbf{y}_{N,2}^H(n) \end{bmatrix} \]
  \[ \mathbf{K}_1(n) = \frac{\lambda^{-1} \mathbf{Q}_2(n-1) \mathbf{y}_{N,2}(n)}{1 + \lambda^{-1} \mathbf{y}_{N,2}^H(n) \mathbf{Q}_2(n-1) \mathbf{y}_{N,2}(n)} \]
  \[ \varepsilon(n) = \mathbf{y}_{N,1}(n) - \mathbf{P}_{N-1}(n-1) \mathbf{y}_{N,2}(n) \]
  \[ \mathbf{P}_{N-1}(n) = \mathbf{P}_{N-1}(n-1) + \varepsilon(n) \mathbf{K}_1^H(n) \]
  \[ \mathbf{Q}_1(n) = \lambda^{-1} \mathbf{Q}_2(n-1) - \lambda^{-1} \mathbf{K}_1(n) \mathbf{y}_{N,2}^H(n) \mathbf{Q}_2(n-1). \]
  2) Compute \( \mathbf{g}_0 \)
  \[ \mathbf{F}(n) = \lambda \mathbf{F}(n-1) + \varepsilon(n) e(n) e(n)^H \]
  \[ \mathbf{g}_0^T = \mathbf{f}(n) [\mathbf{I}_L - \mathbf{P}_{N-1}(n)] \]
  where \( \mathbf{f}(n) \) is the column of \( \mathbf{F}(n) \) with the largest norm.
  3) Compute the second linear prediction problem
  \[ \mathbf{K}_2(n) = \frac{\lambda^{-1} \mathbf{Q}_2(n-1) \mathbf{y}(n)}{1 + \lambda^{-1} \mathbf{y}_{N,2}^H(n) \mathbf{Q}_2(n-1) \mathbf{y}_{N,2}(n)} \]
  \[ \mathbf{J}(n) = \mathbf{y}(n) - \mathbf{P}(n-1) \mathbf{y}(n) \]
  \[ \mathbf{P}_N(n) = \mathbf{P}_N(n-1) + \mathbf{J}(n) \mathbf{K}_2^H(n) \]
  \[ \mathbf{Q}_2(n) = \lambda^{-1} \mathbf{Q}_2(n-1) - \lambda^{-1} \mathbf{K}_2(n) \mathbf{y}_{N,2}^H(n) \mathbf{Q}_2(n-1). \]
  4) Compute the \( d \)-delay ZF equalizer
  \[ \mathbf{g}_d^T = \mathbf{g}_d^T \mathbf{P}_N(n). \]

The computational complexity of the above algorithm is approximately \( 15L(N)^2 \). The term \( \lambda (0 \leq \lambda \leq 1) \) is a “forgetting” factor included to reduce the influence of past values on the statistics and, thereby, allow the algorithm to track time variations in the channel. It affects the convergence as well as the accuracy of the algorithm. The tradeoff between tracking and convergence dictates \( \lambda \) in the interval \([0.98, 1]\). For simplicity, the same \( \lambda \) is shared in the algorithm.

In many cases, it has been observed [9] that selecting \( d \approx (L_0 + L - 2)/2 \) results in good equalizer performance. A recursive algorithm to find the best delay \( d \) is discussed in [1], which is also applicable in our case.

The computation of the above algorithm can be further reduced. The first prediction problem is performed on the past data vector \( \mathbf{y}_{N,2}(n) \) with dimension \( L(N-1) \times 1 \) to find a \( L \times L(N-1) \) matrix \( \mathbf{P}_{N-1}(n) \), whereas the second prediction is performed on \( \mathbf{y}(n) \) with dimension \( LN \times 1 \). Note that the last \( L(N-1) \) entries of \( \mathbf{y}(n) \) are \( \mathbf{y}_{N,2}(n) \). Therefore, we can do the first prediction problem on \( \mathbf{y}(n-L) \) to find a \( L \times LN \) matrix \( \mathbf{P}_{N-1}(n) \) and then simply drop the last \( L \) columns to compute \( \mathbf{g}_0 \). Because the farthest past values have the least effects on the recent data, the norm of the last \( L \) columns of \( \mathbf{P}_{N-1}(n) \) is much less than other entries. Therefore, this simplification is reasonable. However, \( \mathbf{y}(n-L) \) was used earlier in the second linear prediction \( L \) iterations. Therefore, the two linear prediction problems are simplified to (almost) one with the first one sharing \( \mathbf{K}_2(n-L) \) and \( \mathbf{Q}_2(n-L) \) of the second one. The computational complexity is reduced to about \( O(LN)^2 \). We call it the simplified RLS algorithm.

C. LMS Equalizer

The above linear prediction problems can also be computed by an LMS algorithm. In a straightforward manner, the first one can be updated by

\[ \varepsilon(n) = \mathbf{y}_{N,1}(n) - \mathbf{P}_{N-1}(n) \mathbf{y}_{N,2}(n) \]
\[ \mathbf{P}_{N-1}(n) = \mathbf{P}_{N-1}(n-1) + \mu \varepsilon(n) \mathbf{y}_{N,2}^H(n). \]

The second one is updated by

\[ \mathbf{J}(n) = \mathbf{y}(n) - \mathbf{P}(n) \mathbf{y}(n) \]
\[ \mathbf{P}_N(n) = \mathbf{P}_N(n-1) + \mu \mathbf{J}(n) \mathbf{y}_{N,2}^H(n). \]

The above LMS algorithm has computation of about \( 5LN^2 \). The main computation comes from the second multichannel linear prediction. However, we can further reduce the computation.

Left-multiplying the vector \( \mathbf{h}_N^H(0) [\mathbf{I}_L - \mathbf{P}_{N-1}] \) to (3.2), we get

\[ \mathbf{j}(n) \triangleq \mathbf{h}_N^H(0) [\mathbf{I}_L - \mathbf{P}_{N-1}] \mathbf{y}(n-d) \]
\[ - \mathbf{h}_N^H(0) [\mathbf{I}_L - \mathbf{P}_{N-1}] \mathbf{y}_N(n). \]

Recall from (3.1) and (3.3) that

\[ \mathbf{g}_d^T = \mathbf{h}_N^H(0) [\mathbf{I}_L - \mathbf{P}_{N-1}] \mathbf{P}_N. \]

In addition, from (2.6) and (2.18), we have for the noiseless case

\[ s_{n-d} = \mathbf{h}_N^H(0) [\mathbf{I}_L - \mathbf{P}_{N-1}] \mathbf{y}(n-d). \]

Therefore, (3.8) is equivalent to

\[ s_{n-d} = \mathbf{h}_N^H(0) [\mathbf{I}_L - \mathbf{P}_{N-1}] \mathbf{y}_N(n-d). \]
Note that (3.11) is only an alternative way of expressing \( s_{n-d} \) but may not give a satisfactory estimate of \( s_{n-d} \) since it is only a zero-delay equalizer operated onto the delayed data. Hence, (3.12) is necessary to improve the equalization. Equations (3.11) and (3.12) can thus be used to replace the second linear prediction (3.7).

Equations (3.11) and (3.12) can be adaptively optimized using an LMS algorithm. First, we apply (3.6) to estimate \( \mathbf{P}_{N-1} \) and the prediction error \( e(n) \), and then, an estimation of \( h(0) \) can be obtained by (2.19). After calculating the estimation of \( s_{n-d} \) by (3.11), we update (3.12) by an LMS algorithm to minimize \( E\{[j(n)]^2\} \) or simply \( |j(n)|^2 \)

\[
j(n) = s_{n-d} - \mathbf{g}_d^T(n) \mathbf{y}_N(n)
\]

\[
\mathbf{g}_d(n) = \mathbf{g}_d(n-1) + \mu_2 \mathbf{y}_N(n) j(n)
\]

(3.13)

where \((\cdot)^*\) stands for complex conjugate. The computation complexity of this algorithm is about \(3L(L+N)\).

Since our formulation is linear prediction and only second-order statistics are involved, the algorithm will surely converge to its global minimum. However, as with any other LMS algorithms, initialization and choice of step size \( \mu_1 \) and \( \mu_2 \) play critical roles in the speed of convergence and the steady-state performance.

**D. Blind Channel Identification**

Once the equalizer is available, we can use it to perform channel identification. From (2.5) and recalling that the channel has finite support, we get

\[
\mathbf{y}(n) = [h(0) \ \cdots \ h(L_h)] + \mathbf{v}(n).
\]

(3.14)

For all \( k = 0, \ldots, L_h \)

\[
E\{\mathbf{y}(n)s_{n-k}^*\} = E\left\{[h(0) \ \cdots \ h(L_h)] \times \begin{bmatrix}s_n \\ \vdots \\ s_{n-L_h}\end{bmatrix} s_{n-k}^* + \mathbf{v}(n) s_{n-k}^* \right\} = \sigma_\delta^2 \mathbf{h}(k).
\]

(3.15)

Substituting \( s_{n-k} = \mathbf{g}_d \mathbf{y}_N(n-k+d) \) into (3.15), we have

\[
\mathbf{h}(k) = \frac{1}{\sigma_\delta^2} E\{\mathbf{y}(n) \mathbf{g}_d^* \mathbf{y}_N^*(n-k+d)\}.
\]

(3.16)

Then

\[
[h(0) \ \cdots \ h(L_h)] = \frac{1}{\sigma_\delta^2} E\{\mathbf{y}(n) \mathbf{g}_d^* \mathbf{y}_N^*(n+d) \ \cdots \mathbf{y}_N^*(n-L_h+d)\}.
\]

(3.17)

Letting \( \mathbf{H} = [h(0) \ \cdots \ h(L_h)] \), the channel can be estimated recursively by the estimator

\[
\mathbf{H}(n) = \lambda \mathbf{H}(n-1) + \mathbf{y}(n) \mathbf{g}_d^* \times [\mathbf{y}_N^*(n+d) \ \cdots \ \mathbf{y}_N^*(n-L_h+d)].
\]

(3.18)

The effectiveness of this channel identification approach depends on the successfulness of the equalizer. Since it is computationally very simple, it can be efficiently added into the algorithms described in the previous sections for estimating the channel coefficients.

**IV. SIMULATION RESULTS**

In this section, we use simulations to examine the performance of the equalization methods described in the previous sections. In addition, we compare the performance of the proposed methods with some existing second- and higher order methods for channel equalization and identification. As a performance measure, we estimate the residual ISI over 100 Monte Carlo runs. Let the “overall” channel impulse response be

\[
f(n) = \sum_{i=0}^{L_d} \sum_{j=0}^{L_d} g_i(j) h_i(n-j).
\]

(4.1)

The residual ISI is defined as

\[
\text{ISI} = \frac{\sum_n |f(n)|^2}{\max_n |f(n)|^2}.
\]

(4.2)

The mean-square error (MSE) of symbol estimation is defined as in [1]

\[
\text{MSE} = E\{|\mathbf{s}_n - \mathbf{s}_n|^2\}.
\]

(4.3)

For all simulations, the signal-to-noise ratio (SNR) is defined to be at the input to the equalizer

\[
\text{SNR} = \frac{E\{|x(n)|^2\}}{E\{|e(n)|^2\}}.
\]

(4.4)

For each experiment, we have used an i.i.d. input sequence drawn from a 16-QAM constellation. The noise is drawn from a white Gaussian distribution at varying SNR’s.

1) Experiment 1—Performance of Proposed Algorithms: We first consider the performance of the RLS equalizer, simplified RLS equalizer, and the LMS equalizer [see (3.6) and (3.13)]. The channel is drawn from the matrix at the bottom of the page [6]. The number of subchannels is \( L = 4 \). Let the equalizer order be \( L_d = 4 \), and let the delay be \( d = 4 \).

Fig. 2(a) shows the learning curves for ISI of the RLS equalizer under SNR \( 15 \) dB, \( 25 \) dB, and \( 35 \) dB, respectively. It shows that the RLS equalizer converges and achieves sufficiently low ISI after as few as 300 symbols. Fig. 2(b) depicts the
learning curves for ISI of the simplified RLS equalizer and LMS equalizer across 2000 symbols at 25 dB. The simplified RLS version is a little bit worse than the RLS equalizer but is much better than the LMS one. However, after a sufficient number of iterations, both of them will achieve satisfactorily low ISI. Note that the LMS equalizer is randomly initialized before running. \( \mu_1 \) and \( \mu_2 \) are 0.005. Fig. 3 shows the received constellation and the equalized constellation at SNR = 25 dB for 500 symbols. Clearly, the blind equalizer has opened the eye.

2) Experiment 2—Comparison with Existing Algorithms: In this experiment, we compare the performance of the RLS and LMS equalizer in Section III-A with existing blind fractionally spaced equalization techniques. The channel used is an empirically measured digital microwave channel with a duration spanning eight symbols; see Fig. 4. It is derived by linear decimation of the FFT of the “full-length” impulse response. See [16] for more details. The SNR was 25 dB, and the equalizer is causal and of order \( L_d = 7 \). We select delay \( d = 4 \) for our prediction-based RLS and LMS algorithms.

Figs. 5(a) and 6(a) show the ISI and MSE for our prediction-based RLS of Section III-B, super-exponential algorithms (SEA) of CMA [17], linear-prediction based algorithms of Slock et al. [4] and Meraim et al. [2], the RLS algorithm of Giannakis et al. [1], and the indirect (subspace) method of Moulines et al. [6]. We see that our prediction-based RLS algorithm has the lowest ISI and MSE and the fastest “convergence.” The computations of our prediction-based RLS algorithm, SEA, Slock, and Giannakis algorithms are all of the order of \( O((LN)^2) \). The Moulines and Meraim algorithms need EVD and thus have more intensive computation.

Figs. 5(b) and 6(b) compare the ISI and MSE performance of four LMS-type algorithms:
- prediction-based LMS [see (3.6) and (3.13)];
- CMA 2-2 [11];
- cyclic LMS of [1];
- LMS MRE of [18].

In order to compare both the convergence and the steady-state error, they were initialized to approximately similar state. Different step sizes are used for these algorithms. The step size for
where $\alpha(t, 0.45)$ is a raised roll-off cosine pulse with the roll-off factor 0.45, and $W(t)$ is a rectangular truncation window spanning $[-0.85T, 5.14T]$. This channel is similar to that of [15]. There are $L = 4$ subchannels. The channel coefficients are shown in Fig. 7(a), whereas the subchannel zeros are shown in Fig. 7(b). We find that there are near common zeros among all subchannels.

The performance comparison of the four algorithms is shown in Fig. 8. It shows that our LMS prediction algorithm still performs better than the others, whereas the subspace algorithm and the cyclic LMS algorithm have poor performance because of the rank determination problem with the correlation matrix.

V. CONCLUSION

Using only linear prediction of the output data, we have provided algorithms for direct calculation of fractionally spaced ZF equalizers with an arbitrary delay and then for channel identification. Although these schemes are based on second-order statistics, they do not require explicit computation of the correlation matrix or its inverse. Instead, only two multichannel linear prediction problems are involved. They can be efficiently implemented as the RLS or LMS adaptive algorithms. The computation can be less than $15(LN)^2$ in each updating recursion in the RLS case and $3LN^2$ in the LMS case. Hence, the computation is low. Simulation shows they converge faster and have lower ISI or MSE than many other algorithms.

It is demonstrated in [2] that direct equalization is robust against channel order overestimation. Hence, our algorithms are less sensitive to order determination, provided we set the equalizer order equal to or larger than the channel order. At this point, our algorithms are similar in computation to the traditional ZF equalizer algorithms such as those of [11] and [17]. However, our algorithms are based on second-order statistics only; therefore, they have faster and guaranteed convergence and better performance in presence of noise.

Simulation results show that our algorithms have good performance in channel equalization. Considering they are computationally efficient and reliable, they are good candidates for real-time application.

Note that the assumption $h(0) \neq 0$ is not essential. If $h(0) = 0$ but $h(1) \neq 0$, our methods will estimate $h(1)$ instead of $h(0)$, and the equalization will still work in much the same way. Some problems may arise when $h(0) \neq 0$ but will have small entries. This is because large relative errors may result in estimating these small entries. If only some entries of $h(0)$ are small, it only results in slightly higher final MSE or ISI values. However, if all entries of $h(0)$ are small, all prediction-based algorithms may not work properly. This aspect deserves further research.

REFERENCES


Xiaohua Li received both the B.S. and M.S. degrees in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1992 and 1995, respectively. Since August 1996, he has been pursuing the Ph.D. degree in electrical engineering at the University of Cincinnati, Cincinnati, OH, where he is a Graduate Research Assistant.

From 1995 to 1996, he was with Shanghai Jiao Tong University and Shanghai Medical Instruments Corporation. His work focused on digital signal processor-based digital hearing aids and multi-channel cochlear implants. His current research includes adaptive and array signal processing, blind channel identification and equalizations for wireless communications, and single and multiuser detection for code division multiple access systems.

Mr. Li was awarded the Excellent Graduates of Shanghai and Chinese Instrument Society Award.

H. (Howard) Fan (SM’90) received the B.S. degree from Guizhou University, Guiyang, China, in 1976 and the M.S. and Ph.D. degrees from the University of Illinois, Urbana, in 1982 and 1985, respectively, all in electrical engineering.

From 1977 to 1978, he worked as a Research Engineer with the Provincial Standard Laboratory and Bureau of Guizhou Province, Guiyang. In 1978, he joined the Graduate School of the University of Science and Technology of China, Hefei, and then transferred to the University of Illinois, where he was a Teaching and Research Assistant from 1982 to 1985. He joined the Department of Electrical and Computer Engineering, University of Cincinnati, Cincinnati, OH, in 1985 as an Assistant Professor and is now a Professor. His research interests are in the general fields of systems and signal processing, in particular, signal representation and reconstruction, adaptive signal processing, signal processing for communications, and system identification. He spent six months in 1994 with the Systems and Control Group, Uppsala University, Uppsala, Sweden, as a Visiting Researcher.

Dr. Fan received the first ECE Departmental Distinguished Progress in Teaching Excellence Award from the University of Cincinnati in 1987. He served as an Associate Editor of the *IEEE TRANSACTIONS ON SIGNAL PROCESSING* from 1991 to 1994. He is a member of Tau Beta Pi and Phi Kappa Phi.