Super-Exponential Methods for Blind Deconvolution
Ofir Shalvi and Ehud Weinstein, Senior Member, IEEE

Abstract—A class of iterative methods for solving the blind deconvolution problem, i.e., for recovering the input of an unknown possibly nonminimum phase linear system by observation of its output is presented. These methods are universal in the sense that they do not impose any restrictions on the probability distribution of the input process provided that it is non-Gaussian. Furthermore, they do not require prior knowledge of the input distribution. These methods are computationally efficient, statistically stable (i.e., small error variance), and they converge to the desired solution regardless of initialization (no spurious local stationary points) at a very fast nearly super-exponential (exponential to the power) rate. The effects of finite length of the data, finite length of the equalizer and additive noise in the system on the attainable performance (intersymbol-interference) are analyzed. It is shown that in many cases of practical interest the performance of the proposed methods is far superior to linear prediction methods even for minimum phase systems. Recursive and sequential algorithms are also developed, which allow real-time implementation and adaptive equalization of time-varying systems.

Index Terms—Blind deconvolution, channel equalization, system identification, high-order cumulants/spectra.

I. INTRODUCTION

We consider the blind deconvolution problem in which we observe the output of an unknown possibly nonminimum phase linear system from which we want to recover its input using an adjustable linear filter (equalizer). This is a problem of considerable practical interest in diverse fields of engineering and applied sciences. For example, in digital communication the input process is a sequence of data symbols, where the unknown system represents the linear distortion caused by the transmission channel (e.g., a telephone line) between the information source and the receiver. In image restoration, the unobserved input represents the original image (scene) where the unknown system represents the blurring effects caused by the electronic or photographic medium.

It is well known that conventional linear prediction methods based on second-order statistics (i.e., correlation or power spectrum) are insufficient for solving the problem since they are incapable of identifying the phase of the unknown system's transfer function. For that reason, the problem cannot be solved when the input process is Gaussian. Consequently, numerous approaches for blind deconvolution based on high order moments/cumulants/spectra have been proposed in the recent literature. A partial list of references is given by [1]–[5], [7]–[11], [13]–[18], [20]–[25].

In this paper, we present a class of deconvolution algorithms that converge iteratively at a very fast nearly super-exponential (that is, exponential to the power) rate to the desired solution, regardless of initialization. As shown in [21], these methods can be linked to the deconvolution criteria proposed in [5], [10], [20], and [25]. For notational convenience, we shall concentrate on the one-dimensional deconvolution problem as appears in e.g., data communication. However, we note that the basic approach presented in this paper can also be applied to multidimensional deconvolution problems such as image deblurring.

II. PROBLEM FORMULATION

The basic problem of interest is illustrated in Fig. 1. We observe the output $y_t$ of an unknown discrete time-invariant system $\mathcal{H} = \{h_n\}$ with input $a_t$ being an unobserved realization (sample function) from a discrete-time stationary random process. We want to adjust the equalizer $C = \{c_n\}$ so that its output $z_t$ is equal to the input $a_t$ up to a constant delay and possibly a constant phase shift. Thus, if we denote by

$$s_n = h_n \circ c_n = \sum l h_{n-l}$$

the unit sample response of the unknown system combined (convolved) with the equalizer, then we want to set $c_n$ so that

$$s_n = e^{j\phi} \delta_{n-k} = \begin{cases} e^{j\phi}, & n = k, \\ 0, & n \neq k, \end{cases}$$

or, equivalently, that the frequency response of the combined system be

$$S(\omega) = e^{j(\phi - k\omega)}$$

where $k$ stands for the delay, and $\phi$ stands for the phase shift. We note that a constant delay is inherent because of the stationarity of the input process, and the constant phase delay.
shift is also unavoidable if the input distribution is invariant under rotation.

Two common measures of equalization performance (e.g., see [18]) are the maximum distortion (MD) defined by

$$\text{MD}(s) = \frac{\sum_n |s_n| - |s_{\text{max}}|}{|s_{\text{max}}|}$$

and the intersymbol-interference (ISI) defined by

$$\text{ISI}(s) = \frac{\sum_n |s_n|^2 - |s_{\text{max}}|^2}{|s_{\text{max}}|^2}$$

where $s_{\text{max}}$ is the component of $s_n$ having the maximal absolute value (the leading tap or cursor). Clearly, MD and ISI are zero if $s_n$ is of the form of (2), and a small value of MD/ISI indicates the proximity to the desired solution.

In the next section, we present a class of iterative algorithms for adjusting $s_n$ that converge monotonically at a very fast rate to the desired response regardless of initialization. We then show how to convert these virtual $s$-domain algorithms into realizable $c$-domain algorithms in which we only need to adjust the unit sample response (taps) $c_n$ of the equalizer.

III. THE ALGORITHM IN THE $s$-DOMAIN

Consider the following iterative procedure for adjusting the unit sample response coefficients (taps) of the combined system:

$$s'_n = s''_n(s'_n)^q$$

$$s''_n = \frac{1}{||s'||} s'_n,$$  \hspace{1cm} (4a) \hspace{1cm} (4b)

where $s_n$ are the tap values before the iteration, $s'_n$ are intermediate values, and $s''_n$ are the tap values after the iteration. We denote by $||s|| = \sqrt{s^*s} = \sqrt{\sum_n |s_n|^2}$ the norm of the (infinite dimensional) vector $s$ of $s_n$, and by $*$ and $+$ the conjugate and conjugate-transpose operations, respectively.

If we choose $p$ and $q$ to be nonnegative integers such that $p + q \geq 2$, then the transformation in (4a) followed by the normalization operation in (4b) causes the taps with smaller magnitudes to decrease more rapidly, forcing $s_n$ to converge very quickly to the desired response in which one tap (the leading tap) approaches one in magnitude, while all other taps approach zero. The effect of the algorithm is illustrated in Fig. 2 for the case $p = 2$ and $q = 1$. Suppose we start with $s_n = 0.8[n]$, which corresponds to an initial ISI of 3.56. After one iteration we obtain $s_n = 0.76(0.512)[n]$, which corresponds to ISI = 0.71. On the next iteration cycle we obtain $s_n = 0.98(0.13)[n]$, corresponding to ISI = 0.037, which is already quite close to the desired response.

Transforming (4a) and (4b) to the frequency domain, we obtain

$$S'(\omega) = \frac{1}{(2\pi)^{p+q-1}} S(\omega^p \cdots \omega^q) * \cdots * S(-\omega^p \cdots \omega^q)$$

$$= \frac{1}{(2\pi)^{p+q-1}} \int_{-\pi}^{\pi} \cdots \int_{-\pi}^{\pi} S(\omega - \sum_{i=1}^{p+q-1} \omega_i) S(\omega_1) \cdots S(\omega_{p-1}) S(-\omega_p \cdots \omega_{p+q-1}) d\omega_1 \cdots d\omega_{p+q-1}$$

and

$$S''(\omega) = \frac{S'(\omega)}{\left[ \frac{1}{2\pi} \int_{-\pi}^{\pi} |S'(\omega)|^2 d\omega \right]^{1/2}},$$  \hspace{1cm} (5a) \hspace{1cm} (5b)

where $S(\omega)$, $S'(\omega)$, and $S''(\omega)$ are the Fourier transforms of $s_n$, $s'_n$, and $s''_n$, respectively. To demonstrate the effect of the algorithm in the frequency domain, consider the case $p = 2$ and $q = 0$. Suppose the initial $S(\omega)$ has a narrow spectral null of width $\pi\epsilon$, at frequency $\omega_0$:

$$S(\omega) = \begin{cases} 0, & |\omega - \omega_0| \leq \pi \epsilon / 2, \\ 1, & |\omega - \omega_0| > \pi \epsilon / 2. \end{cases}$$

After performing one iteration cycle we obtain (see Fig. 3):

$$S''(\omega) = \begin{cases} 1 + \frac{\pi \epsilon}{2}, & \omega = -2\omega_0 + o(\epsilon^2), \\ 1 + o(\epsilon^2), & |\omega - 2\omega_0| \leq \pi \epsilon, \\ 1 + o(\epsilon^2), & |\omega - 2\omega_0| > \pi \epsilon. \end{cases}$$

If we regard the difference between $S(\omega)$ and the desired response in (3) as “disturbance,” then a single iteration of the algorithm causes the width of the “disturbance” to be multiplied by a factor of 2, the location of the peak of the “disturbance” is multiplied by a factor of 2, and the peak value is multiplied by a factor of $\epsilon/2$ which essentially eliminates the-
spectral null. With arbitrary $p$ and $q$, we obtain similar effects: the width of the “disturbance” is multiplied by a factor of $(p + q)$, and the peak value is multiplied by a factor smaller than $e^{p+q}$. 

To analyze the convergence behavior of the algorithm, we note that by (4a) and (4b),

$$
\left| \frac{s_n'}{s_{n_0}'} \right| = \left| \frac{s_n}{s_{n_0}} \right|^{p+q}
$$

(6)

Since $p + q > 2$, then

$$
|s_{n_0}| \geq |s_{n_1}| \geq |s_{n_2}| \geq \cdots \Rightarrow |s_{n_0}'| \geq |s_{n_1}'| \geq |s_{n_2}'| \geq \cdots
$$

(7)

That is, the indexes $n_0, n_1, n_2, \ldots$ of the ordered a components are preserved throughout the iterations. In particular, the index of the leading tap is preserved along the iterations. Thus,

$$
\text{MD}(s') = \sum_{n \neq n_0} \left| \frac{s_n'}{s_{n_0}'} \right|^p = \sum_{n \neq n_0} \left| \frac{s_n}{s_{n_0}} \right|^{p+q}
$$

$$
\leq \text{MD}(s)^{p+q}
$$

(8)

and

$$
\text{ISI}(s') = \sum_{n \neq n_0} \left| \frac{s_n'}{s_{n_0}'} \right|^2 = \sum_{n \neq n_0} \left| \frac{s_n}{s_{n_0}} \right|^{2(p+q)}
$$

$$
\leq \left( \sum_{n \neq n_0} \left| \frac{s_n}{s_{n_0}} \right|^2 \right)^{(p+q)} = \text{ISI}(s)^{p+q}
$$

(9)

Performing (8) and (9) iteratively we obtain, respectively,

$$
\text{MD}(s^{(l)}) \leq \text{MD}(s^{(0)})^{[p+q]^l}
$$

(10)

$$
\text{ISI}(s^{(l)}) \leq \text{ISI}(s^{(0)})^{[p+q]^l}
$$

(11)

where $s^{(0)} = (\cdots s_{-1}^{(0)}, s_0^{(0)}, s_1^{(0)}, \cdots)^T$ denotes the initial value of $s$, and $s^{(l)} = (\cdots s_{-1}^{(l)}, s_0^{(l)}, s_1^{(l)}, \cdots)^T$ denotes the value of $s$ after $l$ iterations of the algorithm. These inequalities indicate that $\text{MD}(s^{(0)})$ and $\text{ISI}(s^{(0)})$ converge to zero at a very fast at least super-exponential (that is, exponential to the power) rate to zero, provided that $\text{MD}(s^{(0)})$ and $\text{ISI}(s^{(0)})$ are smaller than 1. Since by (4b) $\|s^{(0)}\| = 1$, then $s^{(l)}$ must converge to a vector having only one nonzero component whose magnitude equals one, i.e., to the desired response.

To prove convergence to the desired response under weaker conditions, we invoke (6) and (7) once again:

$$
\text{MD}(s') = \sum_{n \neq n_0} \left| \frac{s_n'}{s_{n_0}'} \right| = \sum_{n \neq n_0} \left| \frac{s_n}{s_{n_0}} \right|^{p+q}
$$

$$
\leq \frac{s_{n_1}}{s_{n_0}} \left| \frac{s_{n_1}}{s_{n_0}} \right|^{p+q-1} \cdot \sum_{n \neq n_0} \left| \frac{s_n}{s_{n_0}} \right|^{p+q-1} \cdot \text{MD}(s)
$$

(12)

Performing (12) iteratively,

$$
\text{MD}(s^{(l)}) \leq \left( \prod_{i=0}^{l-1} \left| \frac{s_{n_{i+1}}}{s_{n_i}} \right| \right)^{p+q-1} \cdot \text{MD}(s^{(0)}).
$$

(13)

By (6),

$$
\left| \frac{s_{n_{i+1}}}{s_{n_i}} \right| = \left| \frac{s_{n_{i+1}}}{s_{n_i}} \right|^{p+q}
$$

(14)

Thus, substituting (14) into (13) and following straightforward algebraic manipulations

$$
\text{MD}(s^{(l)}) \leq \prod_{i=0}^{l-1} \left| \frac{s_{n_{i+1}}}{s_{n_i}} \right|^{p+q}
$$

(15)

and

$$
\text{ISI}(s^{(l)}) \leq \prod_{i=0}^{l-1} \left| \frac{s_{n_{i+1}}}{s_{n_i}} \right|^{p+q}
$$

(16)

In a similar way, we obtain

$$
\text{MD}(s^{(l)}) \leq \prod_{i=0}^{l-1} \left| \frac{s_{n_{i+1}}}{s_{n_i}} \right|^{p+q}
$$

(17)

$$
\text{ISI}(s^{(l)}) \leq \prod_{i=0}^{l-1} \left| \frac{s_{n_{i+1}}}{s_{n_i}} \right|^{p+q}
$$

(18)

Since $\text{MD}(s^{(0)}) \geq \Gamma_0$ and $\text{ISI}(s^{(0)}) \geq \Gamma_0^2$, then the bounds in (15) and (16) are tighter than the bounds in (10) and (11), respectively. Furthermore, the convergence of these bounds to zero (and thus to the desired response) only requires the weaker condition that $\Gamma_0 < 1$, i.e., that the leading tap of the initial response is unique.

In the case where the leading tap of the initial response is not unique, i.e., there are $M > 1$ distinct taps that exactly achieve the maximal absolute value, it is easy to verify that the algorithm converges to a vector $s$ having $M$ nonzero components of equal magnitude $1/\sqrt{M}$. However, these convergence points are unstable equilibria, any small deviation from which is sufficient to ensure convergence to the desired response. For example, in case $p + q = 3$, if $\Gamma_0 = 0.7$,
then by (16) we need \( l = 2 \) iterations to improve the initial ISI by a factor of at least 100. If \( \Gamma_0 = 0.999 \), which represents a very small deviation from the indicated anomaly, we need \( l \approx 7 \) iterations for a factor 100 improvement in the initial ISI.

IV. THE ALGORITHM IN THE C-DOMAIN

The \( s \)-domain algorithm presented in the previous section is stated in terms of the taps \( s_n \) of the combined system which, of course, are unavailable for adjustment. In this section we show how to implement the algorithm in the \( c \)-domain using only the taps \( c_n \) of the equalizer and some statistical cumulants of the observed data. Since the equalizer being used is of finite length, the \( s \)-domain algorithm in (4a), (4b) with the associated super-exponential convergence rate can only be approximated. Since we use empirical cumulants, obtained by sample averages over the available data, instead of the exact cumulants, it causes random errors, and the presence of additive noise can only add to these errors. All these effects must be taken into consideration.

The organization of this main section is as follows. In Section IV-A, we define cumulants and present some of their properties that will be used in the sequel. In Section IV-B, we present the \( c \)-domain algorithm subject to the finite-length restriction (still with the exact cumulants). In Section IV-C, we present a realizable algorithm obtained by replacing the unavailable cumulants by their empirical estimates, and analyze its statistical stability (error variance) for a few special cases. In Section IV-D, we briefly address the effect of additive noise in the system. In Section IV-E, we present simulation results to confirm the analysis. In Section IV-F, we develop sequential and adaptive approximations of the algorithm.

A. Mathematical Preliminaries

Let \( x_1, x_2, \ldots, x_n \) be a set of real/complex random variables possessing the joint characteristic function:

\[
\phi(\omega) = E \left\{ e^{j\omega^T x} \right\} = E \left\{ e^{j\sum_{i=1}^n \omega_i x_i} \right\}
\]

where \( j = \sqrt{-1} \) and \( E \{ \cdot \} \) stands for the expectation operation.

The joint cumulant of \( x_{n_1}, x_{n_2}, \ldots, x_{n_m} \), \( n_i \in \{1, 2, \ldots, n\} \) is defined by

\[
\text{cum}(x_{n_1}; x_{n_2}; \ldots; x_{n_m}) = (-j)^m \frac{\partial^m \ln \phi(\omega)}{\partial \omega_{n_1} \cdots \partial \omega_{n_m}} \bigg|_{\omega = 0},
\]

where \( \ln(\cdot) \) stands for the natural logarithm. An alternative definition of cumulants in terms of moments is given in ([3, ch. 2, definition 2.3.1]). Thus, if \( x_1, x_2 \) are zero-mean random variables then

\[
\begin{align*}
\text{cum}(x_1) &= 0 \\
\text{cum}(x_1; x_2) &= E\{x_1; x_2\} \\
\text{cum}(x_1; x_2; x_3) &= E\{x_1; x_2; x_3\} \\
\text{cum}(x_1; x_2; x_3; x_4) &= E\{x_1; x_2; x_3; x_4\} \\
\end{align*}
\]

and so on.

For notational convenience, let

\[
\text{cum}(x_1; x_2; \ldots; x_i; \cdots) = \text{cum}(x_1; p; \cdots).
\]

\[\text{p terms}\]

The following properties can easily be verified (see [3]).

P1) Linearity: \( \text{cum}(\sum_{i} a_i x_i; \cdots) = \sum_{i} a_i \text{cum}(x_i; \cdots) \)

P2) If \( x_{n_1}, x_{n_2}, \ldots, x_{n_m} \) can be divided into two statistically independent subsets, then their joint cumulant is zero.

P3) If \( x_{n_1}, x_{n_2}, \ldots, x_{n_m} \) are jointly Gaussian, then their joint cumulant is zero whenever \( \sigma > 2 \).

B. Algorithm Implementation Using Finite-Length Equalizers

In this subsection, we show how to express the \( s \)-domain algorithm (4a), (4b) in terms of the equalizer taps and some data cumulants. We shall make the following assumptions:

1) The input \( a_t, t = 1, 2, \ldots \) is a sequence of real/complex independent identically distributed (i.i.d.) continuous/discrete type random variables. We assume the existence of the following nonzero cumulants of \( a_t \):

\[
\text{cum}(a_t; a_t^*; \cdots) = \text{var}(a_t) > 0,
\]

\[
\text{cum}(a_t; a_t^*; \cdots; q + 1) \neq 0.
\]

where \( p \) and \( q \) are the parameters appearing in (4a). Note that since \( p + q \geq 2 \) then by P3) \( a_t \) must be non-Gaussian.

Comment: It is sufficient to assume that \( a_t, t = 1, 2, \ldots \) are statistically independent random variables. However, for the sake of simplicity, we further assume that they are also identically distributed.

2) The unknown system \( \mathcal{H} = \{ h_n \} \) is a stable, possibly nonminimum phase, linear time-invariant filter whose inverse (which may be noncausal) \( \mathcal{H}^{-1} = \{ h_n^{-1} \} \) exists.

3) The equalizer \( C = \{ c_n \}_{n=L_0}^{L_1} \) is a tap-delay line of finite-length \( L = L_2 - L_1 + 1 \).

The combined system response subject to the finite-length restriction is

\[
s_n = \sum_{l=L_1}^{L_2} c_l h_{n-l}.
\]

In a vector form

\[
s = H e.
\]

where \( s \) is the possibly infinite vector taps of the combined system

\[
s = (s_1, s_2, \ldots, s_n) \trans .
\]

\( e \) is the \( L \times 1 \) vector equalizer taps

\[
e = (e_{L_1}, e_{L_1+1}, \ldots, e_{L_2}) \trans 
\]

and \( H \) is the matrix of \( L \) columns and possibly infinite number of rows, whose elements are

\[
H_{ij} = h_{i-j}, \quad -\infty < i < \infty, \quad L_1 \leq j \leq L_2.
\]
In the deconvolution problem, we want to adjust $c$ so that $s = Hc$ is equal to the vector $\delta^{(k)}$ whose $n$th element is $\delta_{n-k}$ for some fixed $k$ (for simplicity, we shall ignore the constant phase shift). However, since $c$ is of finite length, we may only require that $c$ is chosen to minimize the distance (norm) between $Hc$ and $\delta^{(k)}$. Assuming that $H$ is known, this is a standard linear least squares problem whose solution is

$$
\min_{c} \|Hc - \delta^{(k)}\|^2 \Rightarrow c = (H^T H)^{-1} H^T \delta^{(k)}
$$

$$
\Rightarrow s = Hc = H(H^T H)^{-1} H^T \delta^{(k)}, \quad (21)
$$

where we note that by assumption 2) $H$ is full rank so that $H^T H$ is invertible for any $L$. This is the optimal solution in the least squares sense, and also in the sense of minimizing the mean-square restoration error since

$$
E\{[z_t - a_{t-k}]^2\} = E\{\sum_n (s_n - \delta^{(k)}_n) a_{t-n}\}
$$

$$
= \sum_n \sum_m (s_n - \delta^{(k)}_n) (s_m - \delta^{(k)}_m) E\{a_{t-n} a_{t-m}\}
$$

$$
= E\{|a_t|^2\} \|s - \delta^{(k)}\|^2 = E\{|a_t|^2\} \|Hc - \delta^{(k)}\|^2.
$$

No equalizer, with or without a training sequence, can do better than the solution in (21) even if $H$ is known a priori. As $L$ increases, the matrix product $H(H^T H)^{-1} H^T$ approaches the identity matrix $I$ (more precisely, the $k$th column of $H(H^T H)^{-1} H^T$ approaches $\delta^{(k)}$ in which case the solution in (21) closely approaches the desired solution.

Having this result in mind, we now turn to the algorithm specified by (4a), (4b). According to (4a), we want to set $c'$ so that $s' = Hc'$ is equal to the vector $g$ whose components are

$$
g_n = s_p^n (s^*_n)^q. \quad (22)
$$

However, since the number of equalizer taps available for adjustment is finite, we shall only require that $c'$ is chosen to minimize the distance (norm) between $Hc'$ and $g$:

$$
\min_{c'} \|Hc' - g\|^2 \Rightarrow c' = (H^T H)^{-1} H^T g. \quad (23a)
$$

The normalization operation in (4b) is equivalent to

$$
c'' = \frac{1}{\sqrt{c'^T H^T H c'}} c'. \quad (23b)
$$

Projecting the $c$-domain algorithm in (23a), (23b) back into the $s$-domain, we obtain

$$
s' = H(H^T H)^{-1} H^T g \quad (24a)
$$

$$
s'' = \frac{1}{\|s'\|} s'. \quad (24b)
$$

It is not difficult to prove that the point of convergence of the algorithm subject to the finite-length restriction is given approximately by

$$
\tilde{s}^{(k)} = \frac{1}{\sqrt{(\delta^{(k)})^T H(H^T H)^{-1} H^T \delta^{(k)}}} H(H^T H)^{-1} H^T \delta^{(k)}
$$

(25)

(just substitute $s = \tilde{s}^{(k)}$ into (24a), (24b) and show that to a first-order approximation $s'' = \tilde{s}^{(k)}$ up to a constant phase shift), and the ISI at that point is

$$
ISI(\tilde{s}^{(k)}) = \frac{\|\tilde{s}^{(k)}\|^2 - ([h_{s}(k)]^T \tilde{s}^{(k)})^2}{([h_{s}(k)]^T \tilde{s}^{(k)})^2}
$$

$$
= \frac{1 - (\delta^{(k)})^T H(H^T H)^{-1} H^T \delta^{(k)}}{(\delta^{(k)})^T H(H^T H)^{-1} H^T \delta^{(k)}}. \quad (26)
$$

We see that $\tilde{s}^{(k)}$ coincides with (21) up to a gain factor which is very close to unity. Thus, by iteratively performing the algorithm in (23a), (23b) we converge to the optimal attainable solution subject to the finite-length restriction. The residual ISI given by (26) is close to zero for sufficiently large $L$. There is a weak dependence on the index $k$ of the leading tap which can be exploited through initialization, since the index of the leading tap is preserved throughout the iterations.

The algorithm in (23a), (23b) is still implicit since $H$ and $g$ are unknown. Our next step is to convert this algorithm into a realizable algorithm expressed in terms of joint cumulants of the input and the output of the equalizer. We start off from

$$
y_t = a_t h_t = \sum_k h_k a_{t-k}. \quad (27)
$$

Then,

$$
y_{t-n} = \sum_k h_k a_{t-n-k} = \sum_k h_{k-n} a_{t-k}. \quad (28)
$$

Invoking properties P1) and P2),

$$
\text{cum}(y_{t-n}; y_{t-m}) = \text{cum}\left(\sum_k h_{k-n} a_{t-k}; \sum_k h_{k-m} a_{t-k}\right)
$$

$$
= \sum_k \sum_k h_{k-n} h_{k-m} \text{cum}(a_{t-k}; a_{t-k})
$$

$$
= \text{cum}(a_t; a^*_t) \sum_k h_{k-n} h_{k-m}
$$

$$
= \text{cum}(a_t; a^*_t) (H^T H)_{mn}. \quad (27)
$$

Invoking P1) once again,

$$
\text{cum}(z_t; p; z^*_t; q; y^*_t-n)
$$

$$
= \text{cum}(z_t; p; z^*_t; q; \sum_k h_{k-n} a^*_t-k)
$$

$$
= \sum_k h^*_t-k \text{cum}(z_t; p; z^*_t; q; a^*_t-k), \quad (28)
$$

where

$$
\text{cum}(z_t; p; z^*_t; q; a^*_t-k)
$$
\[ \text{cum}(z_t; z_{t-1}; \ldots; z_{t-k}; a_t^*; \ldots; a_{t-k}^*) \]
\[ = \sum_{l_1} \ldots \sum_{l_p} \sum_{m_1} \ldots \sum_{m_q} s_{l_1} \ldots s_{l_p} s_{m_1} \ldots s_{m_q} \mu_{l_1-1} \ldots \mu_{l_p-1} \mu_{m_1-1} \ldots \mu_{m_q-1} \cdot \text{cum}(a_t; \ldots; a_{t-k}; \ldots; a_{t-k}^*) \] 
\[ = \sum_{l_1} \ldots \sum_{l_p} \sum_{m_1} \ldots \sum_{m_q} s_{l_1} \ldots s_{l_p} s_{m_1} \ldots s_{m_q} \mu_{l_1-1} \ldots \mu_{l_p-1} \mu_{m_1-1} \ldots \mu_{m_q-1} \cdot \text{cum}(a_t-1; \ldots; a_{t-l}; \ldots; a_{t-m}; \ldots; a_{t-k}^*) \] 
\[ \text{(29)} \]

Since \( a_t, t = 1, 2, \ldots \), are i.i.d. random variables, then by P2,
\[ \text{cum}(a_t-1; \ldots; a_{t-l}; \ldots; a_{t-m}; \ldots; a_{t-k}^*) = \begin{cases} \text{cum}(a_t; p; a_t^*; q+1), & l_1 = \ldots = l_p = m_1 = \ldots = m_q = k, \\ 0, & \text{otherwise.} \end{cases} \] 
\[ \text{(30)} \]

Substituting (30) into (29),
\[ \text{cum}(z_t; z_{t-1}; queue q+1) = \text{cum}(a_t; p; a_t^*; q+1) = g_k \cdot \text{cum}(a_t; p; a_t^*; q+1). \] 
\[ \text{(31)} \]

Substituting (31) into (28),
\[ \text{cum}(z_t; z_{t-1}; queue q+1) = \text{cum}(a_t; p; a_t^*; q+1) = \sum_k h_{t-n} g_k \]
\[ = \text{cum}(a_t; p; a_t^*; q+1) (H^* g)_n. \] 
\[ \text{(32)} \]

Using (27) and (32) in (23a), (33b), we obtain:
\[ c' = R^{-1} d. \] 
\[ \text{(33a)} \]
\[ c'' = \left(\frac{1}{\sqrt{c' + Rc'}}\right) c', \] 
\[ \text{where } R \text{ is the } L \times L \text{ matrix whose elements are} \]
\[ R_{nm} = \frac{\text{cum}(y_{t-m}; y_{t-n})}{\text{cum}(a_t; a_t^*)} = \frac{\text{cov}(y_{t-m}; y_{t-n})}{\text{var}(a_t)} \] 
\[ \text{and } d \text{ is the } L \times 1 \text{ vector whose elements are} \]
\[ d_n = \frac{\text{cum}(z_t; p; z_t^*; f; y_{t-n})}{\text{cum}(a_t; p; a_t^*; q+1)}. \] 
\[ \text{(34)} \]

In this setting, the algorithm is expressed in terms of the equalizer taps and some data cumulants. The term \( \text{cum}(a_t; p; a_t^*; q+1) \) appearing in the denominator of (35) is just a multiplying constant that can be incorporated into the normalization operation in (33b). Thus, to implement the algorithm we only need to know the variance (average power) of \( a_t \). The algorithm is universal in the sense that it does not impose any restrictions on the probability distribution of \( a_t \), provided that it is non-Gaussian satisfying the conditions in (17) and (18).

We note that (33a) is similar in structure to linear prediction. In both cases, we need to multiply by the inverse \( R^{-1} \) of the data covariance matrix. The difference is that in linear prediction \( d \) is a vector of second-order cumulants (correlations), while here it is a vector of higher order cumulants.

By P1,
\[ \text{cum}(z_t; p; z_t^*; q; y_{t-n}) = \text{cum}(z_t; \ldots; z_t^*; q; y_{t-n}) \]
\[ = \left( \sum_{l_1=L_1}^{L_2} c_{l_1} y_{t-l_1}; \ldots; \sum_{l_p=L_1}^{L_2} c_{l_p} y_{t-l_p}; \ldots; \sum_{k_1=L_1}^{L_2} c_{k_1} y_{t-k_1}; \ldots; \sum_{k_q=L_1}^{L_2} c_{k_q} y_{t-k_q}; y_{t-n} \right) \]
\[ = \sum_{l_1=L_1}^{L_2} \sum_{l_p=L_1}^{L_2} \sum_{k_1=L_1}^{L_2} \sum_{k_q=L_1}^{L_2} c_{l_1}^* c_{l_p}^* c_{k_1}^* c_{k_q}^* \text{cum}(y_{t-l_1}; \ldots; y_{t-l_p}; y_{t-k_1}; \ldots; y_{t-k_q}; y_{t-n}). \] 
\[ \text{(35)} \]

Substituting (36) into (35), we obtain an alternative implementation of the algorithm. Instead of computing joint cumulants of \( y_t \) at each iteration cycle, we may first compute the cumulants of \( y_t \) and then perform the iterations.

C. The Algorithm with Empirical Cumulants

In practice the exact cumulants are unknown, and therefore can only be approximated by their sample estimates using the available data. Therefore, instead of (33a), (33b), we use
\[ c' = R^{-1} d \] 
\[ \text{(37a)} \]
\[ c'' = \left(\frac{1}{\sqrt{c' + Rc'}}\right) c' \] 
\[ \text{(37b)} \]

where \( \hat{R} \) is the \( L \times L \) matrix whose elements are
\[ \hat{R}_{nm} = \frac{\text{cum}(y_{t-m}; y_{t-n})}{\text{cum}(a_t; a_t^*)} \] 
\[ \text{(38)} \]
and \( \hat{d} \) is the \( L \times 1 \) vector whose elements are
\[ \hat{d}_n = \frac{\text{cum}(z_t; p; z_t^*; f; y_{t-n})}{\text{cum}(a_t; p; a_t^*; q+1)}, \] 
\[ \text{(39)} \]

where \( \text{cum}(\cdot) \) denotes the estimate of \( \text{cum}(\cdot) \), obtained by approximating ensemble averages with empirical averages. For example, if \( a_t \) is zero-mean, then
\[ \text{cum}(y_{t-m}; y_{t-n}) = E(y_{t-m} y_{t-n}), \]
\[ \text{cum}(z_t; p; z_t^*; f; y_{t-n}) \big|_{p=q=1} = E(z_t^2 y_{t-n}) \]
\[ \text{cum}(z_t; p; z_t^*; f; y_{t-n}) \big|_{p=q=2} = E(z_t^2 z_t y_{t-n}) - 2E(z_t^2) E(z_t y_{t-n}) \]
\[ - E(z_t^2) E(z_t y_{t-n}), \]
in which case
\[ \text{cum}(y_{t-m}; y_{t-n}) = \frac{1}{N} \sum_{t=1}^{N} y_{t-m} y_{t-n}, \] 
\[ \text{(40)} \]
\[ \text{cum}(z_t; p; z_t^*; f; y_{t-n}) \big|_{p=q=1} = \frac{1}{N} \sum_{t=1}^{N} z_t^2 y_{t-n}^2, \] 
\[ \text{(41)} \]
where $N$ is the number of available data samples. By [9, theorem 2], these cumulant estimates are consistent with probability one (w.p.1) and in the mean-square (m.s.) sense provided that $E[|a_1|^{2(p+q+1)}] < \infty$, in which case the algorithm in (37a), (37b) approaches (33a), (33b) as $N \to \infty$. However, we note that there may be more efficient techniques for estimating cumulants from finite data records.

Equations (40)-(42) completely specify the algorithm for the case $p = 1, q = 1$, and for the case $p = 2, q = 1$, in terms of the observed $y_t$ and $z_t$ at the input and output of the equalizer. To assess the computational complexity of the algorithm we note that the calculation of $\hat{R}$ (to be computed only once) requires $NL$ operations, the generation of $z_1, \ldots, z_N$ requires $N$ operations, and the calculation of $d$ requires about $(L+p+q-1)N$ operations. The inversion of $\hat{R}$, the multiplication of $\hat{R}^{-1}$ by $d$, and the normalization in (37b) require $O(L^2)$ operations which is relatively negligible for $L \ll N$. As a baseline, we note that linear prediction methods, which can only be applied to the minimum phase case, requires $2NL$ operations ($NL$ operations for the calculation of the equalizer taps, and $NL$ operations for the generation of $z_t$). Therefore, the number of operations per iteration is essentially equivalent to that of linear prediction, and since the algorithm converges within a few iterations, then its overall computational complexity is not exceedingly larger than that of predictive deconvolution.

The use of empirical cumulants instead of the exact cumulants may affect the point of convergence of the algorithm. Asymptotic small error analysis is carried out in Appendix A. Assuming that the cumulant estimates are consistent, as in (40)-(42), the stationary point of the algorithm up to a constant delay and a constant phase shift is given approximately by

$$\hat{s} = \frac{1}{\sqrt{\delta^T H (H^+ H)^{-1} H^+ \delta}} H (H^+ H)^{-1} H^+ \delta, \quad \text{(43)}$$

where $\delta = \delta^{(0)}$ is the vector whose $n$th element is $\delta_n$, and where $\delta$ is the vector whose elements are

$$\delta_n = \begin{cases} \frac{1 + \text{cum}(a_1, a_1^*) - \text{cum}(a_1, a_1^*)}{2 \text{cum}(a_1, a_1^*)}, & n = 0, \\ \frac{\text{cum}(a_1, p, q; a_1^*, a_1^*) - \text{cum}(a_1, p, q; a_1^*, a_1^*)}{\text{cum}(a_1, p, q; a_1^*, a_1^*)}, & n \neq 0, \end{cases} \quad \text{(44)}$$

where $\text{cum}(a_1; a_1^*; q_1^*)$ is the empirical cumulant of the (unobserved) input process $a_1$. Taking the expectation of (43) and assuming that the cumulant estimates are unbiased, so that $E[\delta] = \delta$, we obtain

$$E[\hat{s}] \approx \frac{1}{\sqrt{\delta^T H (H^+ H)^{-1} H^+ \delta}} H (H^+ H)^{-1} H^+ \delta \triangleq \hat{s}, \quad \text{(45)}$$

This is the convergence point of the algorithm based on the exact cumulants (see (25)), indicating that the asymptotic bias is essentially due to the finite length of the equalizer.

Taking the covariance of (43)

$$\text{cov}(\hat{s}) = H (H^+ H)^{-1} H^+ \text{cov}(\delta), \quad \text{(46)}$$

where $\text{cov}(\hat{s})$ and $\text{cov}(\delta)$ stand for the covariances of $\hat{s}$ and $\delta$, respectively.

The expected ISI at the point of convergence can be closely approximated by

$$E\{\text{ISI}(\hat{s})\} \approx E\left\{ \frac{1}{|\delta_0|^2} \sum_{n \neq 0} |\delta_n|^2 \right\} 
\approx \frac{1}{|\delta_0|^2} \sum_{n \neq 0} |\delta_n|^2 + \frac{1}{|\delta_0|^2} \sum_{n \neq 0} \text{var}(\delta_n) 
= \text{ISI}(\hat{s}) + \frac{1}{|\delta_0|^2} [\text{tr}(\text{cov}(\hat{s})) - \text{cov}(\delta_0)], \quad \text{(47)}$$

where $\text{tr}(\cdot)$ stands for the trace of the bracketed matrix. The term $\text{ISI}(\hat{s})$, given by (26), is the residual ISI due to the finite length of the equalizer, where the second term represents the contribution due to the finite length of the data.

We must now substitute (46) into (47) and carry out the indicated operations. The calculations for the case $p = 1, q = 1$ (third-order cumulants) is carried out in Appendix B. For simplicity, we assume that $\alpha_t$ is a real valued, zero-mean random variable in which case the condition in (18) requires that $E[\alpha_t^2] \neq 0$. The result is

$$E\{\text{ISI}(\hat{s})\} = \text{ISI}(\hat{s}) + \frac{1}{N} \left[ (L-1) E[\alpha_t^2] E[\alpha_t^2] - E[\alpha_t^2] E[\alpha_t^2] \right] + \frac{1}{N} \left[ (L-1) \sum_{n \neq 0} E[\alpha_t^2] E[\alpha_t^2] \right], \quad \text{(48)}$$

where $1 = (\cdots 1, 1, 1, \ldots)^T$. We note that $\text{ISI}(\hat{s})$ monotonically decreases with $L$ while the second term in (48) increases with $L$, indicating a trade-off between bias and variance.

For the purpose of comparison, the asymptotic performance of predictive deconvolution based on the minimum phase assumption is given by (e.g., see [5], [14]):

$$E\{\text{ISI}(\hat{s})\} \approx \text{ISI}(\hat{s}) + \frac{L-1}{N}. \quad \text{(49)}$$

If $L$ is large compared to the effective length of the inverse $N^{-1}$ of $\mathcal{H}$ then $\text{ISI}(\hat{s}) \approx 0$, and if $L$ is very large compared with the effective length of $N^{-1}$ then $1^T H (H^+ H)^{-1} H^+ 1 - L \approx 0$, in which case the ratio of (48) to (49) (relative efficiency) is given by

$$\rho = \frac{E[\alpha_t^2] E[\alpha_t^2] - E[\alpha_t^2] E[\alpha_t^2]}{E[\alpha_t^2] E[\alpha_t^2]}. \quad \text{(50)}$$

For example, if $\alpha_t$ is a nonsymmetric zero-mean binary r.v. admitting the values $-1$ and $1$ with probabilities $\alpha/(1+\alpha)$ and $1/(1+\alpha)$, respectively, then a simple calculation yields $\rho = \alpha/(1-\alpha)^2$. For $\alpha < 0.38$ or $\alpha > 2.62$, $\rho < 1$, indicating that in this range the asymptotic performance of the proposed
algorithm is better than that of linear prediction based on the minimum phase assumption. The reason is that third-order cumulants carry useful information that can be used to improve performance. However, in the range \(0.38 < \alpha < 0.62\) the relative efficiency is greater than one, and as \(\alpha\) approaches 1, \(\rho\) approaches infinity. This is since in this limit \(\sigma_{c}\) becomes a binary symmetric random variable for which the third-order cumulant is identically zero and therefore, it does not carry any statistical information. In this case we must use the information contained in the higher order cumulants.

The calculation of (47) for the case \(p = 2\) and \(q = 1\) (fourth-order cumulants) is carried out in Appendix B. In this case, we let \(\sigma_{c}\) be either real or complex valued random variable. For simplicity we have assumed that \(E\{a_{1}\} = E\{a_{1}^{*}\}\) are zero. In the complex case, we have further assumed that \(E\{a_{3}^{2}\}, E\{a_{1}a_{3}a_{2}^{*}\} \) and \(E\{|a_{2}|^{4}\}\) are zero, in which case \(E\{z_{i}^{4}\} = 0\) so that the third term on the right-hand side of (42) need not be estimated. These assumptions are satisfied for most signal constellations used in data communication. By (18), \(\sigma_{c}\) is restricted to have a nonzero fourth-order cumulant:

\[
\text{cum}(a_{1}, 2; a_{1}^{*}, 2) = E\{|a_{1}|^{4}\} - \lambda E^{2}\{|a_{1}|^{2}\} \neq 0,
\]

where \(\lambda = 3\) in the real case and \(\lambda = 2\) in the complex case.

The result is

\[
E\{\text{ISI}(\hat{a})\} = |\text{ISI}(\hat{a})| + \frac{L - 1}{N} E\{|a_{1}|^{2}\} E\{|a_{1}|^{6}\} - E^{2}\{|a_{1}|^{4}\},
\]

(51)

We note that if \(E\{|a_{1}|^{2}\sigma_{c}\} \neq 0\), then (51) contains an additional term as indicated in Appendix B. Ignoring the contribution of the bias terms in (51) and (49), their relative efficiency is:

\[
\rho = \frac{E\{|a_{1}|^{2}\} E\{|a_{1}|^{6}\} - E^{2}\{|a_{1}|^{4}\}}{E\{|a_{1}|^{4}\} - \lambda E^{2}\{|a_{1}|^{2}\}}.
\]

(52)

For example, if \(\sigma_{c}\) is a uniformly distributed random variable, then \(\rho = 3/7\) indicating that fourth-order cumulants provide useful information that can be used to improve performance (reduce the variance). For V22 and V29 signal constellations that are commonly used in data communication (see Fig. 4) \(\rho\) is equal to 0.471 and 0.719, respectively, indicating that in both cases the proposed algorithm is superior to predictive deconvolution even for minimum phase systems.

Invoking Cauchy–Schwarz inequality, it can be shown that

\[
E\{|a_{1}|^{2}\} E\{|a_{1}|^{6}\} \geq E^{2}\{|a_{1}|^{4}\},
\]

(53)

where equality holds, if and only if the probability weight of \(\sigma_{c}\) is zero unless \(\sigma_{c} = 0\) or \(|a_{1}| = A\) for some positive real \(A\). Some of the most popular signals used in communications (see [24]) and in particular in digital communications, e.g., BPSK, V27, 4-QAM (see Fig. 4) satisfy this condition. For this class of signal constellations, \(\rho = 0\), implying that the statistical variance is zero even for finite data, in which case the expected ISI given by (51) is not zero only because of the bias effect due to the finite length of the equalizer.

\[
y_{t} = h_{t} \circ a_{t} + e_{t},
\]

To explain this striking result, suppose that \(|a_{t}| = A\) with probability one (w.p.1). Then, for \(n \neq 0\),

\[
v_{n} = \frac{\text{cum}(a_{1}, 2; a_{1}^{*}, 2; a_{n}^{*})}{\text{cum}(a_{1}, 2; a_{1}^{*}, 2)} - \frac{\text{cum}(a_{1}, a_{1}^{*})}{\text{cum}(a_{1}; a_{1}^{*})}
\]

\[
= \frac{1}{N} \sum_{t=1}^{N} |a_{1}|^{2} a_{1} a_{n}^{*} - \frac{1}{N} \sum_{t=1}^{N} |a_{1}|^{2} \frac{1}{N} \sum_{t=1}^{N} a_{t} a_{n}^{*}
\]

\[
= \frac{1}{A^{2}} \sum_{t=1}^{N} a_{t} a_{n}^{*} - \frac{(1 - \lambda) A^{4}}{A^{4}} \sum_{t=1}^{N} \frac{1}{N} \sum_{t=1}^{N} a_{t} a_{n}^{*}
\]

\[
= 0 \quad \text{w.p.1.}
\]

This rather interesting applicable result is confirmed with the observation by Godard [10] that blind deconvolution based on his criterion might be an easier task for signals having constant amplitude.

Comment: We may consider combining the algorithm based on third-order and fourth-order cumulants, and perhaps other algorithms in the class, in order to improve statistical stability. We may also consider incorporating linear prediction methods, particularly when the input distribution is close to Gaussian. By appropriately weighting and linearly combining the two methods, we obtain a method that is capable of identifying the phase ambiguities associated with predictive deconvolution, and whose expected ISI is lower than that of either one of the methods separately (and in particular lower than that of predictive deconvoation). Detailed considerations are beyond the scope of this paper.

D. Effect of Additive Noise

Consider the system illustrated in Fig. 5 in which

\[
y_{t} = h_{t} \circ a_{t} + e_{t},
\]
where $e_t$ represents the additive noise at the input to the equalizer. Then,

$$z_t = c_t \circ y_t = s_t \circ a_t + c_t \circ e_t.$$  

If $a_t$ and $e_t$ are statistically independent ($e_t$ need not be an i.i.d. process) then by P1 and P2,

$$\begin{align*}
\text{cum}(y_{t-n}; y_{t-m}^*) &= \text{cum}(h_{t-n} \circ a_{t-n}; h_{t-m}^* \circ a_{t-m}^*) \\
&+ \text{cum}(e_{t-n}; e_{t-m}^*),
\end{align*}$$

(54)

$$\begin{align*}
\text{cum}(z_t; p; z_t^*; q; y_{t-n}^*) &= \text{cum}\left((s_t \circ a_t) \circ p; (s_t \circ a_t)^* \circ q; h_{t-n}^* \circ a_{t-n}^*\right) \\
&+ \text{cum}\left((c_t \circ e_t) \circ p; (c_t \circ e_t)^* \circ q; e_{t-n}^*\right).
\end{align*}$$

(55)

The first terms on the right hand sides of (54) and (55) represent the cumulants in the noise-free case, where the second terms represent the contribution due to additive noise. If the noise cumulants are known a priori, or can be measured independently, their effect can be subtracted from (54) and (55), respectively. If $e_t$ is a Gaussian process then all its cumulants of order greater then 2 are zero, in which case it has no effect on the computation of (55).

In the actual implementation of the algorithm we use empirical cumulants and therefore the effect of additive noise on the computation of (55) is not zero even in the Gaussian case. This issue must be explored in depth.

Since the equalizer converges to the inverse $H^{-1}$ of $H$, it may emphasize certain components of the noise at its output, depending on the frequency response of the unknown system. Therefore, as a final step we suggest to perform one more iteration without subtracting the noise covariance from (54). As shown in [21], this corresponds to performing Wiener filtering, instead of inverse filtering, that minimizes the mean-square restoration error. If the input symbols are drawn from a discrete known source constellation then we can minimize the mean-square restoration error by switching to a decision directed mode [16], [18].

**E. Simulation Results**

To verify the results of the analysis, the algorithm in (37a), (37b) was examined for $p = 2, q = 1$. In this case the elements of $R$ and $d$ are calculated by substituting (40) and (42) into (38) and (39), respectively. In all the examples that we have considered $E(z_t^2) = 0$ whenever $a_t$ is complex valued in which case $E(z_t^2) = 0$ and the third term on the right-hand side of (42) is not estimated.

In all experiments, the unknown system is an FIR filter of length 7 with the tap values (0.4, 1, -0.7, 0.6, 0.3, -0.4, 0.1), corresponding to an initial ISI of 1.27. We have used an equalizer of length $L = 16$ which was initialized to $c^{0}(0) = (0.001, 0.0000, 0.0000, 0.00000)^T$.

In the first set of experiments the input symbols were independent realizations from V29 source (see Fig. 4). The algorithm was tested in 50 Monte Carlo trials and the average ISI, denoted by $<\text{ISI}>$, is tabulated and plotted in Fig. 6. We have used a logarithmic (dB) scale because of the very fast convergence rate. When using data blocks of $N = 2400$ samples, the algorithm converged in all trials within 2-4 iterations to an average ISI which is very close to the asymptotic performance predicted by (51). For data blocks of $N = 1600$ samples, the algorithm converged within 2-5 iterations, but there was a deviation of approximately 25% (1 dB) from the asymptotic result, which may be because of second-order stochastic effects due to the finite length of the data.

In Figs. 7 and 8, the input symbols were independent realizations from a V27 source (see Fig. 4) and from a binary source, respectively. In both cases, $p = 0$, so that according to our analysis the algorithm is relatively insensitive to stochastic effects due to the finite length of the data, and the residual ISI at the point of convergence is essentially due to the finite length of the equalizer. Thus, in the case of binary source with $N = 800$ and $N = 400$, which are considered very short data blocks, the algorithm converged in all trials within 2-4 iterations to the predicted performance level. To push the algorithm to its limit, it was tested for data blocks of $N = 200$ points, in which case it still converged in 88% of the trials to a sufficiently small ISI level. These results suggest that the algorithm may be potentially useful for communication over fast fading channels.
additive white Gaussian noise with SNR = 20 dB and SNR = 10 dB (where SNR is the ratio of the average power of $a_t$ to the average power of $e_t$). In both cases the algorithm converged in all trials, and the effect of additive noise on the convergence rate was small.

In the last set of experiments, the input symbols where independent realizations from a uniformly distributed random variable. The outcome results are shown in Fig. 10. For data blocks of $N = 1000$ points the algorithm converged within 2–4 iterations very close to the expected ISI predicted by (51). For data blocks of $N = 1000$ points, the algorithm converged within 2–5 iterations, and the presence of additive white Gaussian noise with SNR = 20 dB did not have any significant effect.

**F. Recursive and Sequential Algorithms**

In a variety of applications, it is desirable to process the data recursively/sequentially. The advantage of a recursive/sequential algorithm over a batch algorithm is not necessarily in the final result, but in the computational efficiency, reduced storage requirements and in the fact that an outcome may be provided without having to wait for all the data to be processed. Moreover, if the underlying system (channel) exhibits time changes, processing all the available data jointly is not desirable, even if we can accommodate the computational and storage load of the batch algorithm, since different data segments correspond to different channel responses. In that case, we want to have an adaptive algorithm that is capable of tracking the varying characteristics of the channel.
Consider the algorithm in (37a), (37b) for the case $p = 1$, $q = 1$ (third-order cumulants). Following the development in Appendix C, we obtain the following recursion for computing $c_t$ at each iteration cycle:

$$c_t = c_{t-1} + \frac{\beta_t}{\gamma} Q_t y_t^T \left[ \left| z_t \right|^2 - \gamma y_t^T c_{t-1} \right]$$  

(56)

and

$$Q_t = \frac{1}{1 - \beta_t} \left[ Q_{t-1} - \frac{\beta_t Q_{t-1} y_t y_t^T Q_{t-1}}{1 - \beta_t + \beta_t y_t^T y_{t-1}} \right].$$  

(57)

where $c_t$ is the value of $c$ based on sample averages to time $t$,

$$y_t = (y_{t-L_1} y_{t-L_2} \cdots y_{t-L_2})^T,$$

$$y_t^T = (y_{t-L_1}^T y_{t-L_2}^T \cdots y_{t-L_2}^T)^T,$$

and

$$z_t = \sum_{t=L_1}^{L_2} c_t y_{t-1} = y_t^T c,$$

where $c$ are the values of the equalizer taps before entering the iteration, and

$$\gamma = \frac{\text{cum}(a_t; a_t^2; \gamma)}{\text{cum}(a_t; a_t^2)} = \frac{E\{a_t^4\}}{E\{|a_t|^2\}}.$$

Similarly, in case $p = 2$, $q = 1$ (fourth-order cumulants), we obtain

$$c_t = c_{t-1} + \frac{\beta_t}{\gamma} Q_t y_t^T \left[ \left| z_t \right|^2 - \gamma E\{|a_t|^2\} z_t - \delta y_t^T c_{t-1} \right],$$  

(58)

where

$$\delta = \frac{\text{cum}(a_t; a_t^2; 2)}{\text{cum}(a_t; a_t^2)} = \frac{E\{a_t^4\} - \lambda E\{|a_t|^2\}^2}{E\{|a_t|^2\}}.$$

If $\beta_t = 1/t$ and $Q_t$ is appropriately initialized, then the outcome $c_{N}$ at the end of the recursion coincides with $c^*$ obtained by calculating (37a) in one step using the sample averages in (40) and (41) in the case $p = 1$, $q = 1$, or the sample averages in (40) and (42) in the case $p = 2$, $q = 1$. If $\beta_t = \beta$ (a constant), it corresponds to exponential weighting that gives more weight to current data in the sample averaging operations. The normalization operation required by (37b) can be performed at the end of the recursion by normalizing $c_{N}$, or by scaling the output of the equalizer to have the same average power as $a_t$.

To convert these iterative-recursive algorithms in which each iteration is performed recursively into sequential algorithms, we suggest to replace the iteration index by the time index ($-\$) which is a common procedure in stochastic approximation, and use the most current state of the equalizer to generate $z_t$. Under these approximations (56) becomes

$$c_t = c_{t-1} + \frac{\beta_t}{\gamma} Q_t y_t^T z_t \left[ \left| z_t \right|^2 - \frac{E\{|a_t|^4\}}{E\{|a_t|^2\}} \right],$$  

(59)

and (58) becomes

$$c_t = c_{t-1} + \frac{\beta_t}{\delta} Q_t y_t^T z_t \left[ \left| z_t \right|^2 - \frac{E\{|a_t|^4\}}{E\{|a_t|^2\}} \right],$$  

(60)

where

$$z_t = y_t^T c_{t-1},$$  

(61)

where the additional normalization operation (scaling $z_t$ to have the same average power as $a_t$) may be required. Both (59) and (60) are fully sequential algorithms in which we perform one iteration per symbol. Using well-known results from stochastic approximation methods [6], [12] it can be shown that if

$$\lim_{t \to \infty} \beta_t = 0, \quad \sum_t \beta_t = \infty, \quad \sum_t \beta_t^2 < \infty,$$

e.g., $\beta_t = 1/t$, then under certain stationary-ergodic conditions these algorithms converge almost surely (a.s.) and in the mean-square (m.s.) to the desired solution, and the limit distribution at the point of convergence can also be evaluated. However, such an analysis is beyond the scope of this paper. If we choose $\beta_t = \beta$ (a constant), it corresponds to exponential weighting that reduces the effect of past data relative to new data, and we effectively obtain adaptive algorithms.

An alternative approach for generating sequential algorithms is based on Robbins–Monro first-order stochastic approximation method [19], [12] in which expectations are replaced by current realizations, and the iteration index is substituted by the time index. The algorithm development is given in Appendix D. In the case $p = 1$, $q = 1$, we obtain

$$c_t = c_{t-1} + \frac{\beta_t}{\gamma} Q_t y_t^T z_t \left[ \left| z_t \right|^2 - \frac{E\{|a_t|^4\}}{E\{|a_t|^2\}} \right],$$  

(62)
and in the case \( p = 2, q = 1 \), we obtain
\[
c_t = c_{t-1} + \beta_t \frac{1}{\text{cum}(a_t:2; a_t^*:2)} y_t^* s_t \left( |c_t|^2 - \frac{E\{a_t^*|t\}^2}{E\{|a_t|^2\}} \right).
\]
(63)

This algorithm is recognized as Godard’s algorithm proposed in [10] and [24]. The return to Godard’s algorithm should not be surprising since the super-exponential methods can be viewed as iterative algorithms for optimizing a family of deconvolution criteria (see [21]), a special case of which is Godard’s criterion. We can now go back and analyze the convergence behavior of Godard’s algorithm based on the observation that it is a first-order stochastic approximation of the super-exponential method whose net action in the \( s \)-domain is well understood. In the case where \( \text{cum}(a_t:2; a_t^*:2) \leq 0 \), the algorithm in (63) converges to the desired solution. However, in the general case, the algorithms in (63), (62) should be modified to maintain the normalization operation, as explained in [21].

The difference between (59) and (62), and between (60) and (63), appears only in the multiplying matrix \( Q_t \). Since the algorithms in (59) and (60) attempt to perform cumulative averaging, they are expected to be statistically more stable than the algorithms in (62) and (63). Since \( Q_t \) is proportional to the inverse of the estimated covariance of \( y_t \), then the multiplication of \( Q_t \) corresponds to a spectral whitening operation and, as pointed out in [2], [21], this may significantly improve convergence rate. To illustrate that, we have considered the previous scenario in which the unknown system is an FIR filter of length 7 whose tap values are \( 0.4, 1, -0.7, 0.6, 0.3, -0.4, 0.1 \), corresponding to an initial IS of 1.27. The input symbols were independent random variables uniformly distributed in \( (-0.5, 0.5) \). We have used an equalizer of length \( L = 16 \), initialized to \( e^{(0)} = 0.00010 \cdots 0 \). The matrix \( Q_t \) needed in (60) was initialized by calculating the covariance matrix \( R \) of \( y_t \) using the first 100 data samples, and taking its inverse. The algorithms were tested in 50 Monte Carlo trials, and the average ISI is plotted in Fig. 11 as a function of the number \( t \) of data samples. In applying (63), that is Godard’s algorithm, we have used \( \beta_t = 1.67 \times 10^{-3} \) and in applying (60) we have used \( \beta_t = 5 \times 10^{-3} \). These are the largest step sizes for which the algorithms converged in all trials. We see that the algorithm in (60) converged almost twice as fast as Godard’s algorithm (to reach an ISI level of 0.05 it required 1140 symbols compared with 2270 required by Godard’s algorithm). The normalization operation was not necessary in implementing the sequential algorithms. As a baseline, we have also plotted the expected performance of the iterative batch algorithm as predicted by (51).

In terms of computational complexity, the algorithms in (62) and (63) requires \( 2L \) operations per data sample, while the algorithms in (59) and (60) require additional \( O(L^2) \) operations. As a compromise, we need not update \( Q_t \) every data sample, or we may compute the data covariance based on a sufficiently large data segment only once, and then proceed with the algorithm.

V. THE ALGORITHM IN THE FREQUENCY DOMAIN

In this section, we relax the assumption that \( a_t \) is an i.i.d. process. More generally, let \( a_t \) be a discrete stationary process, and define its \((p,q)\)-order spectrum by
\[
S_{a:p,q}(\omega; \omega_{t+q}) = \sum_{l_1} \cdots \sum_{l_p} \text{cum}(a_{l_1}; a_{t-l_1}; \cdots; a_{t-l_{p-1}}; a_{t-l_{p}}; a_{t-l_{p+q}}; e^{-j\sum_{i=1}^{p+q} l_i}).
\]
(64)

We note that the \((1,0)\)-order spectrum is the power spectrum of the process, the \((1,1)\)-order spectrum is the bispectrum, and the \((2,1)\)-order spectrum is the trispectrum.

Suppose that
\[
S_{a:p,q}(\omega) = \omega, \quad -\pi \leq \omega < \pi,
\]
and
\[
S_{a:p,q+1}(\omega_1; \cdots; \omega_{p+q}) \neq 0, -\pi \leq \omega_1; \cdots; \omega_{p+q} < \pi,
\]
for some combination of \( p \) and \( q \) such that \( p + q \geq 2 \). We note that if \( a_t \) is an i.i.d. process then (64) and (65) reduce to the conditions in (17) and (18), respectively, since in this case, for all \( \omega_1; \cdots; \omega_{p+q} \),
\[
S_{a:p,q+1}(\omega_1; \cdots; \omega_{p+q}) = \text{cum}(a_{\omega}; a_{\omega}^*; \omega+1).
\]

The relations between the \((p,q)\)-order spectra of the input \( a_t \) and the output \( y_t \) of the unknown system is given by [3]
\[
S_{a:p,q+1}(\omega_1; \cdots; \omega_{p+q}) = H^\dagger \left( \sum_{l=1}^{p+q} \omega_l \right) H_{-1} \left( \prod_{m=1}^{p+q} H_m(\omega_l) \right).
\]
(66)

where
\[
H(\omega) = \sum_{l} h_l e^{-j\omega l}\text{is the frequency response of the unknown system. In the special case where } (p,q) = (1,0), \text{ the relation between the power spectra of } a_t \text{ and } y_t \text{ is given by}
\]
\[
S_{a:p}(\omega) = |H(\omega)|^2 S_{a:0}(\omega).
\]
(67)

Substituting \( S(\omega) = H(\omega) C(\omega) \), \( S_t(\omega) = H(\omega) C_t(\omega) \) and \( S''(\omega) = H(\omega) C''(\omega) \) into (5a) (5b) and using the relations
in (66) and (67), we obtain
\[
C'(\omega) = \frac{S_{x,y}^n(\omega)}{S_{y,y}(\omega)} \int_{-\pi}^{\pi} \cdots \int_{-\pi}^{\pi} C' \left( \omega - \sum_{l=1}^{p+q-1} \omega_l \right) \\
\prod_{l=1}^{p-1} C(\omega_l) \prod_{j=1}^{p+q-1} C^*(-\omega_j) \\
\frac{S_{y,y}^p y_{p+q+1}(-\omega_1, \ldots, -\omega_{p+q-1}, \omega)}{S_{y,y}^p y_{p+q+1}(-\omega_1, \ldots, -\omega_{p+q-1}, \omega)} \\
d\omega_1 \cdots d\omega_{p+q-1},
\]
(68)
\[
C''(\omega) = \frac{1}{\int_{-\pi}^{\pi} \frac{S_{x,x}(\omega)}{S_{y,y}^n(\omega)} |C'(\omega)|^2 d\omega} C'(\omega),
\]
(69)
where \( C(\omega) = \sum_l c_l e^{-j\omega l} \) is the frequency response of the equalizer. This algorithm does not require that \( a_l \) be an i.i.d. process. In turn, it requires prior knowledge of its power spectrum and some other nonzero higher order spectrum.

In the actual implementation of the algorithm, the power spectrum and the \((p, q)\)-order spectrum of \( y_t \) are replaced by their sample estimates. Since the equalizer being used is of finite-length \( L \), its frequency response can be adjusted only at \( L \) independent frequencies, say the discrete Fourier transform (DFT) frequencies \( \omega_k = \frac{2\pi k}{L}, k = 0, 1, \ldots, (L-1), \) and therefore, it has a finite resolution. The resulting formula is given by
\[
C'(\omega_k) = \frac{S_{x,x}(\omega_k)}{S_{y,y}(\omega_k)} \sum_{k_0=0}^{L-1} \sum_{k_{p+q-1}=0}^{L-1} C' \left( \omega_k - \sum_{l=1}^{p+q-1} \omega_{k_l} \right) \\
\prod_{l=1}^{p-1} C(\omega_{k_l}) \prod_{j=p}^{p+q-1} C^*(-\omega_{k_j}) \\
\frac{S_{y,y}^p y_{p+q+1}(-\omega_1, \ldots, -\omega_{k_{p+q-1}}, \omega_k)}{S_{y,y}^p y_{p+q+1}(-\omega_1, \ldots, -\omega_{k_{p+q-1}}, \omega_k)} \\
C''(\omega_k) = \frac{1}{\left( \int_{-\pi}^{\pi} \frac{S_{x,x}(\omega)}{S_{y,y}^n(\omega)} |C'(\omega)|^2 d\omega \right)^{1/2}} C'(\omega_k),
\]
(70)
where \( S_{x,x}(\omega) \) and \( S_{y,y}^p y_{p+q+1}(\omega_1, \ldots, \omega_{p+q}) \) denote the estimates of the power spectrum and the \((p, q)\)-order spectrum of \( y_t \) at the indicated set of frequencies. Statistical analysis of the algorithm, similar to that performed in the time domain, can also be carried out here.

VI. THE SYSTEM IDENTIFICATION ASPECT

The approach presented in the paper can also be used for the purpose of system identification. Since the equalizer \( C \) is directed at finding the inverse of the unknown system \( \mathcal{H} \), it can therefore be viewed as modeling \( \mathcal{H}^{-1} \). Since \( C \) is an all-zero FIR filter, it corresponds to modeling the unknown system \( \mathcal{H} \) by an all-pole (AR) filter, where the taps \( c_0 \) of the equalizer can be regarded as the direct estimates of the AR parameters of that model. Close form expressions for the resulting bias and error variance can easily be extracted from (45) and (46), where we note that the bias term reflects the possible mismatch between the actual \( \mathcal{H} \) and its assumed AR model. The main feature of this system identification approach is that it uses a non-Gaussian input distribution and therefore it is capable of identifying nonminimum phase systems, unlike the classical approaches for system identification that are based on the Gaussian assumption and therefore are incapable of identifying nonminimum phase characteristics. The method is universal in the sense that it does not impose any restrictions on the probability distribution of the input process (driving noise) and it only requires prior knowledge of its average power. Since the proposed method uses information contained in the high-order statistics of the observed data, it often exhibits better performance compared with second-order methods even for low orders of the observed data. Finally we note that the proposed technique for reducing the effect of additive noise by cumulant subtraction can also be applied here.

VII. CONCLUSION

We have presented a class of computationally efficient iterative algorithms for blind deconvolution that converge monotonically, regardless of initialization (no spurious local stationary points), at a very fast nearly super-exponential rate to the desired solution in which the inverse of the unknown system is identified, and its input is recovered up to a constant delay and possibly a constant phase shift. We assume that the input process is a sequence of i.i.d. real/complex random variables with non-Gaussian but otherwise completely arbitrary continuous/discrete probability distribution. When implementing the algorithms in the frequency domain, the i.i.d. assumption can be relaxed and instead we require prior knowledge of the power-spectrum and some higher order spectrum of the input process. The effects of the finite length of the equalizer and finite length of the data are analyzed and the results are confirmed via Monte-Carlo simulations. It is shown that in many cases of interest the performance (intersymbol-interference) of the proposed methods is far superior to linear prediction methods even for minimum phase systems. Recursive and sequential forms of the algorithms are also presented.

APPENDIX A

DERIVATION OF (43)

Under the consistency requirement, the cumulant estimates asymptotically preserve the linear property of cumulants. Thus, by (27)-(29),
\[
\hat{c}(y_{t-m}; y_{t-n}) \approx \sum_{k_1}^{k_2} h_{k_1-m} h_{k_2-n} \hat{C}(a_{t-k_1}; a_{t-k_2}), 
\]
(A.1)
\[
\hat{c}(z_{t-p}; z_{t-q}) \approx \sum_{k} h_{k-n} \hat{c}(z_{t-p}; z_{t-q}; a_{t-k}), 
\]
(A.2)
\[
\hat{c}(z_{t-p}; z_{t-q} a_{t-n}) \approx \sum_{l_1}^{l_2} \cdots \sum_{k_1}^{k_2} \cdots \sum_{s_1}^{s_2} \cdots \sum_{s_{k_1}}^{s_{k_2}} \cdots \cdot s_{k_2} \hat{c}(a_{t-l_1}; a_{t-l_2}; a_{t-k_1}; \cdots; a_{t-k_2}; a_{t-n}), 
\]
(A.3)
Note that the cumulant estimates in e.g., (40)–(42) satisfies (A.1)–(A.3) for any $N$ without any approximation.

Near the point of convergence, $s_n \approx 0$ for $n \neq 0$ in which case (A.3) is closely approximated by

$$
cum(a_n; a_{n-1}) \approx s_n^0(s_n^0)^n \cum(a_n; a_{n-1}).
$$

(A.4)

Hence, recall (38) and (39),

$$
\hat{R} \approx H^* \hat{A} H.
$$

(A.5)

$$
d \approx H^* \hat{d}.
$$

(A.6)

where $\hat{A}$ is the matrix whose elements are

$$
\hat{A}_{nm} = \frac{\cum(a_{n-m}; a_{n-1})}{\cum(a_n; a_{n-1})}
$$

(A.7)

and $\hat{d}$ is the vector whose elements are

$$
\hat{d}_n = s_n^0(s_n^0)^n \frac{\cum(a_n; a_{n-1})}{\cum(a_n; a_{n-1})}.
$$

(A.8)

Neglecting small-order terms,

$$
\hat{R}^{-1} \approx R^{-1} - R^{-1}(\hat{R} - R)R^{-1} = (H^+H)^{-1} - (H^+H)^{-1}(\hat{A} - I)H(H^+H)^{-1}.
$$

(A.9)

Thus,

$$
c' = \hat{R}^{-1}d \approx (H^+H)^{-1}H^+u.
$$

(A.10)

where

$$
u = \hat{g} - (\hat{A} - I)H(H^+H)^{-1}H^+\hat{g}.
$$

(A.11)

Since $\hat{A} \approx I$ and $\hat{g} \approx s_n^0(s_n^0)^n \delta$, and since for large $L$ the product $H(H^+H)^{-1}H^+$ approaches the identity matrix, then the components of $\mathbf{u}$ are given to a very good approximation by

$$
u_n \approx \begin{cases} 
\frac{1}{2} \frac{\cum(a_n; a_{n-1})}{\cum(a_n; a_{n-1})}, & n = 0, \\
\frac{1}{2} \frac{\cum(a_{n+1}; a_{n+1})}{\cum(a_n; a_{n-1})}, & n \neq 0, 
\end{cases}
$$

(A.12)

Since $\mathbf{u} \approx s_n^0(s_n^0)^n \delta$ then

$$
c'^* \hat{R}c' \approx \mathbf{u}^*H(H^+H)^{-1}H^+\hat{A}H(H^+H)^{-1}H^+\mathbf{u}
$$

$$
\approx |u_n|^2 \frac{\cum(a_n; a_{n-1})}{\cum(a_n; a_{n-1})} \delta^2 H(H^+H)^{-1}H^+\delta.
$$

(A.13)

Hence, up to a constant phase,

$$
c' = \frac{1}{\sqrt{c'^* \hat{R}c'}} \frac{1}{\sqrt{\delta^2 H(H^+H)^{-1}H^+\delta}}
$$

where $\mathbf{v}$ is a vector whose elements are

$$
v_n \approx \begin{cases} 
\frac{\cum(a_n; a_{n-1})}{\cum(a_n; a_{n-1})}, & n = 0, \\
\frac{\cum(a_{n+1}; a_{n+1})}{\cum(a_n; a_{n-1})}, & n \neq 0, 
\end{cases}
$$

(A.14)

Thus,

$$
s'' \approx Hs'' \approx \mathbf{s}.
$$

(A.16)

where $\mathbf{s}$ is given by (43). Since $\mathbf{s}$ is independent of the iteration index, then it is approximately the stationary point of the algorithm.

**Appendix B**

**Derivation of (48) and (51)**

In analogy with (40)–(42),

$$
cum(a_n; a_{n-1}) = \frac{1}{N} \sum_{t=1}^{N} a_t a_{t-1}^*.
$$

(B.1)

$$
cum(a_n; a_{n-1}; a_{n-2}) = \frac{1}{N} \sum_{t=1}^{N} |a_t|^2 a_{t-1}^*,
$$

(B.2)

$$
cum(a_n; 2; a_{n-1}; a_{n-2}) = \frac{1}{N} \sum_{t=1}^{N} |a_t|^2 a_{t-1}^* - \lambda \frac{1}{N} \sum_{t=1}^{N} |a_t|^2 \cdot \frac{1}{N} \sum_{t=1}^{N} a_t a_{t-1}^* - \frac{1}{N} \sum_{t=1}^{N} a_t a_{t-1}^*.
$$

(B.3)

where $\lambda = 3$ in the real case and $\lambda = 2$ in the complex case, and where in the transition to the second line of (B.3) we have neglected zero mean terms whose variance is of the order of $O(1/N^2)$.

Substituting (B.1)–(B.3) into (44),

$$
v_n = \begin{cases} 
\frac{1}{N} - \frac{1}{2} \frac{\cum(a_n; a_{n-1})}{\cum(a_n; a_{n-1})}, & n = 0, \\
\frac{1}{N} \sum_{t=1}^{N} f_{p,q}(a_t) a_{t-1}^*, & n \neq 0. 
\end{cases}
$$

(B.4)

where

$$
f_{p,q}(a_t) = \begin{cases} 
\frac{|a_t|^2}{\cum(a_n; a_{n-1})} - \frac{a_t^* a_{t-1}^*}{\cum(a_n; a_{n-1})}, & p = q = 1, \\
\frac{|a_t|^2}{\cum(a_n; a_{n-1})} - \frac{a_t^* a_{t-1}^*}{\cum(a_n; a_{n-1})}, & p = 2, q = 1. 
\end{cases}
$$

(B.5)

Invoking the assumption that $a_t \sim 1, 2, \ldots$ is a sequence of zero mean i.i.d. r.v.'s and noting that $E\{a_t f_{p,q}(a_t)\} = 0$, $cov(v_m, v_m) = 0$,

$$
E\{f_{1,1}(a_t)\} = \frac{E\{|a_t|^2\}}{E\{|a_t|^2 a_{t-1}^*\}},
$$

(B.7)

$$
E\{|f_{1,1}(a_t)|^2\} = \frac{E\{|a_t|^2\} E\{|a_t|^2 a_{t-1}^*\}^2}{E\{|a_t|^2\} E\{|a_t|^2 a_{t-1}^*\}}.
$$

(B.8)
\[ E(f_{21}(a_i)) = \frac{E|a_i|^2a_i}{\text{cum}(a_i; 2, a_i^*; 2)}, \quad \text{(B.9)} \]
\[ E(f_{21}(a_i)^2) = \frac{E|a_i|^2 E|a_i|^4 - E^2|a_i|^4}{E|a_i|^2 \text{cum}(a_i; 2, a_i^*; 2)^2}. \quad \text{(B.10)} \]

where in (B.10) we have assumed that if \( a_i \) is complex valued
then \( E(a_i^2) = E|a_i|^2 a_i^2 = 0 \).

Arranging the components of (B.6) in a matrix, the covariance of the vector \( \mathbf{v} \) of \( v_i \) can be written in the form
\[ \text{cov}(\mathbf{v}) = \gamma_1 \mathbf{1}^T + \gamma_2 \mathbf{I} + \gamma_3 \delta \delta^T + \gamma_4 \delta \delta^T, \quad \text{(B.11)} \]

where
\[ \gamma_1 = \frac{1}{N} E|a|^2 E(f_{p,q}(a_i))^2, \quad \text{(B.12)} \]
\[ \gamma_2 = \frac{1}{N} E|a|^2 \text{var}(f_{p,q}(a_i)), \quad \text{(B.13)} \]
\[ \gamma_3 = -\frac{1}{2N} \frac{E|a|^2a_i^*}{E|a|^2} E(f_{p,q}(a_i)) - \gamma_1, \quad \text{(B.14)} \]
\[ \gamma_4 = \frac{1}{4N} \frac{E|a|^2}{E|a|^2 - 1} - \gamma_1 - \gamma_2 - \gamma_3 \gamma_3. \quad \text{(B.15)} \]

Substituting (B.11) into (47) and carrying out the indicated matrix manipulations
\[ E\{ISI(\mathbf{v})\} = ISI(\mathbf{v}) + \gamma_1 \mathbf{1}^T W \mathbf{1} - |\mathbf{1}^T W \mathbf{1}|^2 + \gamma_2 (L - \delta^T W \delta) + 2 \text{Real}[\gamma_3 \delta^T W \mathbf{1} (1 - \delta^T W \delta)] + \gamma_4 \delta^T W \mathbf{1} (1 - \delta^T W \delta), \quad \text{(B.16)} \]

where
\[ W \overset{\Delta}{=} H (H^+ H)^{-1} H^+. \]

For sufficiently large \( L \) the first column of \( W \) approaches the vector \( \delta \) in which case \( \delta^T W \delta \approx 1, \mathbf{1}^T W \mathbf{1} \approx 1 \) and (B.16) is closely approximated by
\[ E\{ISI(\mathbf{v})\} = ISI(\mathbf{v}) + \gamma_1 \mathbf{1}^T W \mathbf{1} - 1 + \gamma_2 (L - 1). \quad \text{(B.17)} \]

Substituting (B.12) and (B.13) into (B.17) and using (B.7)-(B.10), we obtain (48) and (51).

\[ \text{APPENDIX C} \]

\[ \text{DERIVATION OF THE ITERATIVE-RECURSIVE ALGORITHMS} \]

By (37a), the value of \( c^* \) based on sample averages to time \( t \) is given by
\[ c^*_t = \frac{1}{\alpha_{p,q}} Q_t q_t, \quad \text{(C.1)} \]

where
\[ Q_t^{-1} = (1 - \beta_t) Q_{t-1}^{-1} + \beta_t \mathbf{y}_t^* \mathbf{y}_t^T, \quad \text{(C.2)} \]

where in the case \( p = 1, q = 1 \)
\[ q_t = (1 - \beta_t) q_{t-1} + \beta_t \mathbf{y}_t^* |z_t|^2, \quad \text{(C.3)} \]

and in the case \( p = 2, q = 1 \)
\[ q_t^2 = (1 - \beta_t) q_{t-1} + \beta_t \mathbf{y}_t^* |z_t|^2 - \lambda E(|a|^2) z_t, \quad \text{(C.4)} \]

where \( \beta_t = 1/t \) in case of the sample averages as in (40)-(42), where \( \alpha_{1,1} = \gamma \) and \( \alpha_{2,1} = \delta \), where \( y_1, \mathbf{y}_t, z_t, \) and \( \gamma \) are defined after (56)-(57) and where \( \delta \) is defined after (58).

Inverting (C.2), we obtain (57). Substituting (C.2) and (C.3) into (C.1),
\[ c_t^* = \frac{1}{\gamma} Q_t (1 - \beta_t) q_{t-1} + \beta_t \mathbf{y}_t^* |z_t|^2 \]
\[ = \frac{1}{\gamma} Q_t (1 - \beta_t) \gamma Q_t^{-1} c_{t-1} + \beta_t \mathbf{y}_t^* |z_t|^2 \]
\[ = \frac{1}{\gamma} Q_t (\gamma Q_t^{-1} - \beta_t \mathbf{y}_t^* \mathbf{y}_t - \beta_t \mathbf{y}_t^* |z_t|^2 \]
\[ = c_{t-1} + \frac{\beta_t}{\gamma} Q_t \mathbf{y}_t^* |z_t|^2 - \gamma \mathbf{y}_t^* c_{t-1}, \quad \text{(C.5)} \]

which coincides with (56).

Similarly, substituting (C.2) and (C.4) into (C.1) we obtain (58).

\[ \text{APPENDIX D} \]

\[ \text{DEVELOPMENT OF FIRST-ORDER STOCHASTIC APPROXIMATION ALGORITHMS} \]

Substituting \( c^* = c \) into (33a) and ignoring the normalization operation in (33b), the set of stationary points of the algorithm must satisfy the equation:
\[ e = R^{-1} d \iff d - Re = 0, \quad \text{(D.1)} \]

In the case \( p = 1, q = 1 \), (D.1) becomes
\[ \frac{1}{\text{cum}(a; \alpha_i^*; 2)} E \left\{ \mathbf{y}_t^* z_t (z_t^2 - E|a|^2) \right\} = 0, \quad \text{(D.2)} \]

and in the case \( p = 2, q = 1 \), (D.1) becomes
\[ \frac{1}{\text{cum}(a; \alpha_i^*; 2)} E \left\{ \mathbf{y}_t^* z_t (|z_t|^2 - E|a|^2) \right\} = 0, \quad \text{(D.3)} \]

where we used this at the stationary points \( E|z_t|^2 = E|a|^2 \).

Gradient-search algorithms for solving (D.2) and (D.3) are given, respectively, by
\[ e^{(l)} = e^{(l-1)} + \frac{\beta_t}{\text{cum}(a; \alpha_i^*; 2)} \]
\[ \cdot E \left\{ \mathbf{y}_t^* z_t (z_t^2 - E|a|^2) \right\} | e = e^{(l-1)}, \quad \text{(D.4)} \]
\[ e^{(l)} = d^{(l-1)} + \frac{\beta_t}{\text{cum}(a; 2; \alpha_i^*; 2)} \]
\[ \cdot E \left\{ \mathbf{y}_t^* z_t (|z_t|^2 - E|a|^2) \right\} | e = e^{(l-1)}, \quad \text{(D.5)} \]

In the case \( p = 2, q = 1 \),
\[ d = E|\mathbf{y}_t^* |z_t|^2 - \lambda E|z_t|^2 z_t, \]
\[ = E|\mathbf{y}_t^* |z_t|^2 - \lambda E|a|^2 z_t \]

where in the transition to the second line we have used that \( E|z_t|^2 = E|a|^2 \) because of the normalization operation. In this setting, \( d \) is obtained by recursively performing (C.4) with \( \delta_t = 1/t \).
Replacing the iteration index \( l \) by the time index \( t \) and substituting expectations with current realizations, we obtain (62) and (63), respectively.

XII. ACKNOWLEDGMENT

We would like to thank the reviewers for their useful and insightful comments.

REFERENCES