PERFORMANCE ANALYSIS OF A RECURSIVE FRACTIONAL SUPER-EXPONENTIAL ALGORITHM

João Gomes *Victor Barroso*

Instituto Superior Técnico — Instituto de Sistemas e Robótica
Av. Rovisco Pais, 1096 Lisboa Codex, Portugal
{jpg,vab}@isr.ist.utl.pt

ABSTRACT

The super-exponential algorithm is a block-based technique for blind channel equalization and system identification. Due to its fast convergence rate, and no *a priori* parameterization other than the block length, it is a useful tool for linear equalization of moderately distortive channels. This paper presents a recursive implementation of the super-exponential algorithm for fractionally-sampled PAM signals. Although the resulting algorithm is still block-based, recursive propagation of several key variables allows the block length to be significantly reduced without compromising the algorithm's accuracy or speed, thereby enhancing its ability to track channel variations. The convergence rate is only mildly influenced by specific channel responses, and oversampling provides smaller output variance and almost perfect tolerance to sampling errors. Simulation results demonstrate the effectiveness of the proposed technique.

1. INTRODUCTION

Intersymbol interference (ISI) is a major source of distortion in high speed digital communications over frequency selective channels. Blind equalization is a processing technique whose aim is to approximately invert the channel using only its output and some prior knowledge of the source statistics or input alphabet structure.

Blind equalization algorithms that rely on (implicit) computation of higher-order statistics (HOS), such as CMA, have been commonly used due to their simplicity, despite a relatively slow convergence rate. Recently, several algorithms that are closely related to CMA have significantly improved the convergence rate with modest computational increase [4, 8]. The class of HOS algorithms proposed by Shalvi and Weinstein, that explicitly maximize cumulants at the equalizer output, have also found wide acceptance [6]. In addition to the recursive/sequential algorithms of [6], Shalvi and Weinstein also proposed a class of iterative block processing algorithms based on a similar cost function [7]. These *super-exponential* algorithms converge at a very fast rate; typically, approximate steady-state solutions are obtained in less than ten iterations for sufficiently large data blocks.

The main purpose of this paper is to present a recursive implementation of the super-exponential algorithm and to show that it has important advantages when compared with other popular blind equalizers. Strictly speaking, this is not a sequential algorithm since data blocks are still used, and the equalizer parameters are not updated once per symbol interval. However, each input sample vector is processed only once, and blocks of input or output samples need not be stored, hence reducing the memory requirements of the algorithm. A reduction in block size is possible since several algorithmic quantities are propagated from one block to the next, which helps to decrease the transient behavior.

The resulting recursive algorithm displays a fast convergence rate when compared with other alternatives such as CMA, and the selection of a step size is handled transparently. This kind of self-tuning ability is especially relevant for practical applications with widely varying channel responses. This recursive super-exponential algorithm uses the generalization for fractionally-spaced sampling described in [2], which allows the equalizer to share many of the desirable properties of decision-directed FSE's.

This work was partially funded by FEDER and PRAXIS XXI, under contract PRAXIS/2/2.1/TPAR/2042/95.

2. BACKGROUND

Vectors and matrices will be represented by lowercase boldface and uppercase boldface letters, respectively. The notations $(\cdot)^T$, $(\cdot)^*$ and $(\cdot)^H$ stand for transpose, complex conjugate and hermitian (conjugate transpose).

2.1. Data Model

A baseband representation of the received signal is considered, in which the transmitted symbols $a(k)$ belong to a possibly complex discrete alphabet. After the received signal has been demodulated, filtered, and fractionally sampled at L times the symbol rate, the following discrete single-input multiple-output (SIMO) model is applicable [2]

$$y^{(i)}(n) = \sum_{k=-\infty}^{+\infty} a(k) h_{n-k}^{(i)} + \eta^{(i)}(n)\,, \quad i = 0, \ldots, L-1\,. \tag{1}$$

In (1), $y^{(i)}(n)$ denotes the observed sequence in the i-th *channel*, and $h_n^{(i)}$ is the corresponding discrete impulse response. The additive noise sequence is denoted by $\eta^{(i)}(n)$.

The output of a fractionally-spaced equalizer is obtained by filtering each channel with an FIR filter $c_n^{(i)}$ and summing the L contributions. In a finite equalizer, the coefficient vector for channel i is defined as $\mathbf{c}^{(i)} = [c_{N_1^{(i)}}^{(i)} \cdots c_{N_2^{(i)}}^{(i)}]^T$, $N_1^{(i)} \leq N_2^{(i)}$, with an associated data vector

$$\mathbf{y}^{(i)}(n) = [y^{(i)}(n - N_1^{(i)}) \ldots y^{(i)}(n - N_2^{(i)})]^T\,. \tag{2}$$

The equalizer output is given by the inner product $z(n) = \mathbf{c}^T \mathbf{y}(n)$, where $\mathbf{c}$ and $\mathbf{y}(n)$ are obtained by stacking $\mathbf{c}^{(0)}, \ldots, \mathbf{c}^{(L-1)}$ and $\mathbf{y}^{(0)}(n), \ldots, \mathbf{y}^{(L-1)}(n)$. The equalizer order will be denoted by $N = \sum_i N_2^{(i)} - N_1^{(i)} + 1$.

2.2. Fractional Super-Exponential Algorithm

The super-exponential algorithm iteratively processes a block of N_t symbols using a two-step procedure. The first step computes an unnormalized coefficient vector that projects an ideal nonlinear memoryless transformation of the unknown total impulse response coefficients (from source to equalizer output) onto the reachable space of the equalizer [7]. This operation typically brings the total response very close to an impulse in a few iterations. The second step is just a normalization that ensures constant power at the equalizer output. In terms of measurable cumulants, the iteration is expressed as [2]:

$$\mathbf{c}'(k) = \mathbf{R}^{-1}\mathbf{d}(k)\ , \quad \mathbf{c}(k) = \frac{\mathbf{c}'(k)}{\|\mathbf{R}^{\frac{1}{2}}\mathbf{c}'(k)\|}\ , \tag{3}$$

where $\mathbf{R} = \sigma_a^{-2}E\{\mathbf{y}(n)\mathbf{y}^H(n)\}$ is the normalized covariance matrix and $\mathbf{d}(k)$ is obtained by concatenating the nonlinear transformation vectors $\mathbf{d}^{(0)}(k), \ldots, \mathbf{d}^{(L-1)}(k)$, whose l-th element is

$$\mathbf{d}_l^{(i)}(k) = \frac{\mathrm{cum}(z(n),\, z(n),\, z^*(n),\, y^{(i)\,*}(n-l))}{C_{2,2}^a}\ . \tag{4}$$

The denominator of (4) is the fourth-order cumulant $C_{2,2}^a = \mathrm{cum}(a(n), a(n),\, a^*(n), a^*(n))$, which is negative in the signal constellations of interest.

3. RECURSIVE ALGORITHM

Using the cumulant to moments formula [5], a generic fourth-order cumulant of zero-mean random variables is written in terms of joint expectations as

$$\begin{aligned} &\mathrm{cum}(x_1, x_2, x_3, x_4) = \\ &\qquad E\{x_1x_2x_3x_4\} - E\{x_1x_2\}\,E\{x_3x_4\} \\ &\qquad - E\{x_1x_3\}\,E\{x_2x_4\} - E\{x_1x_4\}\,E\{x_2x_3\}\ . \end{aligned} \tag{5}$$

Applying (5) to each element of $\mathbf{d}$, this vector may be expressed as a crosscorrelation

$$\begin{aligned} C_{2,2}^a\mathbf{d} &= E\left\{|z(n)|^2 z(n)\mathbf{y}^*(n)\right\} \\ &\quad - \lambda E\left\{|z(n)|^2\right\} E\left\{z(n)\mathbf{y}^*(n)\right\} \\ &= E\left\{\left(|z(n)|^2 - \lambda E\left\{|z(n)|^2\right\}\right) z(n)\mathbf{y}^*(n)\right\} \\ &\triangleq E\left\{\psi_z(n)\mathbf{y}^*(n)\right\}\ , \end{aligned} \tag{6}$$

where $\lambda = 3$ for real signals and $\lambda = 2$ in the complex case. The latter value of λ assumes that $E\{z^2(n)\} = 0$ for an arbitrary impulse response, which means that the input constellation must satisfy $E\{a^2(n)\} = 0$. The discrete sequence $\psi_z(n) = \left(|z(n)|^2 - \lambda E\left\{|z(n)|^2\right\}\right) z(n)$ is obtained by applying a nonlinearity to the equalizer output. Note that the exact normalization of $\mathbf{c}$ in the batch algorithm of section 2. guarantees that $E\{|z(n)|^2\} = E\{|a(n)|^2\}$; then ψ_z is essentially the same fixed memoryless function used in Godard's algorithm $\psi(z(n)) = (|z(n)|^2 - K)z(n)$, possibly differing only in the constant K that causes a scaling effect in the output constellation [1]. It should also be remarked that a function ψ_z may be defined for complex signals even if the assumption $E\{a^2(n)\} = 0$ is not valid, such as in binary PSK. However, the resulting algorithm is much more sensitive to phase jitter and carrier frequency offset.

When other than fourth-order statistics are used in the super-exponential algorithm, the joint cumulants that comprise $\mathbf{d}$ are still linear in $y^{(i)}(n)$. Then, this vector may still be written as a crosscorrelation between $\mathbf{y}(n)$ and an order-dependent nonlinear function as in (6). Hence, the basic approach used in the recursive implementation can be easily extended to non-CMA equalization.

3.1. RLS Estimation of the Coefficient Vector

Taking into account the expression for $\mathbf{d}$ developed in (6), $\mathbf{c}'$ in (3) is seen to be the least-squares solution to the problem

$$\mathbf{c}'(k) = \arg\min_{\mathbf{c}} E\left\{\left|\frac{C_{2,2}^a}{C_{1,1}^a}\mathbf{c}^T\mathbf{y}(n) - \psi_z(n)\right|^2\right\}\ . \tag{7}$$

The scale factor $C_{2,2}^a/C_{1,1}^a$ is largely irrelevant because $\mathbf{c}'$ will be subsequently normalized, and its sign is the only higher-order *a priori* statistical knowledge about the source that needs to be considered. Therefore, $\psi_z(n)$ is redefined as

$$\psi_z(n) = \mathrm{sgn}(C_{2,2}^a)\left(|z(n)|^2 - \lambda E\left\{|z(n)|^2\right\}\right) z(n)\ , \tag{8}$$

and a scaled $\mathbf{c}'$ is found as the solution of

$$\mathbf{c}'(k) = \arg\min_{\mathbf{c}} E\left\{\left|\mathbf{c}^T\mathbf{y}(n) - \psi_z(n)\right|^2\right\}\ . \tag{9}$$

Even if $\mathrm{sgn}(C_{2,2}^a)$ is not included in (8), it may only cause a sign reversal in $\mathbf{c}'$, and therefore in the equalizer output. Since the super-exponential algorithm does not guarantee an absolute phase reference, any further ± 1 ambiguity is immaterial and the channel will still be correctly equalized. However, sign reversals are very important in the recursive algorithm due to the nature of the initialization procedure described below.

In a practical implementation, cumulants are estimated using sample averages [5]. Then $\mathbf{c}'(k)$ is given by the solution to the (weighted) time average over the N_t input vectors and output symbols in the k-th block[1]:

$$\mathbf{c}'(k) = \arg\min_{\mathbf{c}} \sum_{n=1}^{N_t} w^{N_t-n}\left|\mathbf{c}^T\mathbf{y}(n) - \hat{\psi}_z(n)\right|^2\ , \tag{10}$$

where $0 < w \leq 1$ is a forgetting factor that allows more effective tracking of time variations in the impulse response, and $\hat{\psi}_z$ is an estimate of ψ_z. This expression may be recognized as a form of CMA with exponential window RLS-type updating within a data block.

The normalization step of this super-exponential iteration is not performed explicitly, that is, the coefficients $\mathbf{c}(k)$ need not be computed. Instead, the unnormalized equalizer output $z'(n) = \mathbf{c}'^T\mathbf{y}(n)$ is divided by an estimate of its power to obtain the desired sequence $z(n)$. Then, the recursive super-exponential algorithm essentially consists on repeated evaluation of (10) in consecutive data blocks.

The RLS algorithm is used to recursively find the optimal coefficients in (10) when the block length ranges from 1 to N_t symbols and, in this sense, provides much more information than what is actually needed. However, the availability of intermediate coefficient vectors and output errors within a data block may be helpful in establishing whether the

[1] At the end of a data block it is assumed that the time index n in (10) is reset to 1.

RLS solution has reached steady-state. When coupled with a cumulant accuracy assessment, this information can be used as part of an adaptive block length selection scheme that is conceptually similar to dynamic step adaptation in standard steepest descent algorithms.

3.1.1. RLS Initialization

In each iteration, the super-exponential algorithm takes an infinite step along the corrected gradient direction of a cost function defined in total impulse response space [7]. It is observed that an infinite step may only be used in practice when reliable cumulant estimates are available, which requires time averaging in data blocks that span close to a thousand symbol intervals. If smaller block lengths are desired, i.e., if the equalizer parameters are to be adjusted more often, then some kind of memory must be introduced among blocks, so that the effective block length used for cumulant estimation is large enough. This is accomplished by a suitable initialization strategy for the sample data covariance matrix Φ, the unnormalized coefficient vector $\mathbf{c}'$, and the estimated unnormalized power. In block k, these variables are initialized with the corresponding values at the end of block $k-1$, thus reducing the transient response in both the RLS input sequence and the weight adaptation process itself. As a result, only a relatively small number of points will be required before the RLS algorithm can improve the solution obtained in the previous block, given the new data.

3.2. Estimation of the Reference Sequence $\hat{\psi}_z$

Similarly to [6], the power estimate of the unnormalized output $z'(n)$ is based on an exponentially-weighted time average with forgetting factor $0 < \alpha \leq 1$,

$$\langle |z'|^2 \rangle_n = \alpha \langle |z'|^2 \rangle_{n-1} + (1-\alpha)|z'(n)|^2 \,. \tag{11}$$

Defining the power ratio $\gamma'(n) = \langle |z'|^2 \rangle_n / C^a_{1,1}$, a normalized output can be simply obtained as $z(n) = z'(n)\gamma'^{-\frac{1}{2}}(n)$. The nonlinear function $\hat{\psi}_z$, based on time averaging rather than ensemble averaging, acts on this sequence to produce

$$\hat{\psi}_z(n) = \mathrm{sgn}(C^a_{2,2}) \left(|z(n)|^2 - \lambda \langle |z|^2 \rangle_n \right) z(n) \,. \tag{12}$$

Due to normalization and estimation errors, the time average $\langle |z|^2 \rangle_n$ evaluated as in (11) will not equal $C^a_{1,1}$, although this identity will tend to be more closely approximated as additional samples in a data block are received. Then, either $\langle |z|^2 \rangle_n$ in (12) should be *defined* as $C^a_{1,1}$, or a second power tracking recursion should be used. The former approach is preferred, leading to the following nonlinearity which condenses the normalization and computation of $\hat{\psi}_z$:

$$\begin{aligned} \hat{\psi}_{z'}(n) &\triangleq \mathrm{sgn}(C^a_{2,2}) \left(|z'(n)|^2 - \lambda \langle |z'|^2 \rangle_n \right) z'(n) \gamma'^{-\frac{3}{2}}(n) \\ &= \hat{\psi}_z(n) \Big|_{z = z'\gamma'^{-\frac{1}{2}}} \,. \end{aligned} \tag{13}$$

Unlike the covariance matrix, initializing $\langle |z'|^2 \rangle$ with the final value from the previous block is just an approximation because the statistics of $z'(n)$ change when $\mathbf{c}'$ is updated. However, since the average power is similar even in the first few blocks, when $\mathbf{c}'$ is far from its steady-state value, this technique is useful in reducing the transient duration and amplitude. Naturally, transients in $\langle |z'|^2 \rangle_n$ will have direct consequences in the convergence speed of the RLS algorithm through the reference signal $\hat{\psi}_{z'}(n)$.

Table 1. Recursive Super-Exponential Algorithm

$\mathbf{c}'(0) = \delta_0; \quad \Phi(0) = \epsilon \mathbf{I}_N; \quad \gamma(0) = |C^a_{2,2}/C^a_{1,1}|^2$

```
for k = 1:N_blocks
    w(0) = c'(k-1);   P(0) = Φ^{-1}(k-1)
    ⟨|z'|^2⟩_0 = C^a_{1,1} γ(k-1)
    for n = 1:N_t
        y(n) = GetSamples()
        z'(n) = c'^T(k-1) y(n)
        ⟨|z'|^2⟩_n = α⟨|z'|^2⟩_{n-1} + (1-α)|z'(n)|^2
        γ'(n) = ⟨|z'|^2⟩_n / C^a_{1,1}
        ψ̂_{z'}(n) = sgn(C^a_{2,2}) (|z'(n)|^2 - λ⟨|z'|^2⟩_n) z'(n) γ'^{-3/2}(n)
        [w(n),P(n)] = RLS(ψ̂_{z'}(n),y(n),w(n-1),P(n-1))
        z(n) = z'(n) γ^{-1/2}(k-1)
    end
    c'(k) = w(N_t);   Φ(k) = P^{-1}(N_t);   γ(k) = γ'(N_t)
end
```

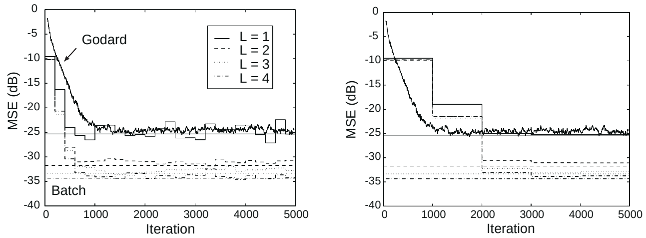
3.3. Computational Complexity

The complete recursive super-exponential algorithm is summarized in table 1. In addition to the estimated power ratio $\gamma'(n)$, a second variable $\gamma(k)$, which equals $\gamma'(N_t)$ at the end of block k, is defined. It is used to generate a normalized equalizer output $z'(n)\gamma^{-\frac{1}{2}}(k-1)$ which does not suffer from transient behavior because it only depends on parameters $\mathbf{c}'(k-1)$ and $\gamma(k-1)$ that are kept fixed in the current data block.

As pointed out in [7], if the block length is considerably larger than the equalizer order, $N_t \gg N$, then the main computational burden of the *nonrecursive* algorithm (3) is due to repeated evaluation of $z(n)$ and the cumulant vector $\mathbf{d}$. At each symbol interval, approximately $2N+2$ add/multiply operations are required, which is roughly the complexity of LMS. Regarding the recursive algorithm of table 1, generation of $z'(n)$ at each instant n requires N operations. Updating the power estimate, normalizing the equalizer output and computing the nonlinear function requires only 2 additions, 1 square-root and about 10 products. Updating the RLS variables is clearly the most time-consuming task, requiring $O(N^2)$ operations for a basic RLS algorithm. Using fast RLS techniques will lower the requirements to $O(NL^2)$ or $O(NL)$, depending on specific algorithms [3], but even for $L=1$ the complexity will always far outweight that of the remaining steps in the super-exponential loop. It can therefore be concluded that the proposed algorithm is essentially equivalent to multichannel RLS in terms of computational cost.

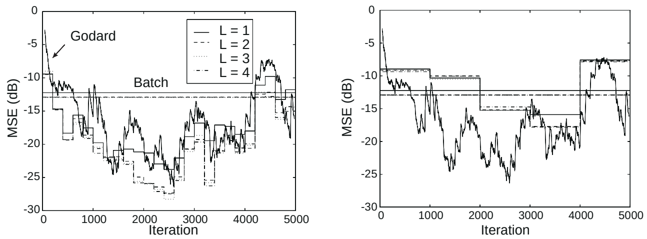
4. SIMULATION RESULTS

Multipath channel: In the first simulation, a binary source transmits i.i.d. symbols from the discrete alphabet $\{-1, +1\}$. The data modulate a sequence of raised cosine pulses, $p(t)$, with 11% rolloff and truncated by a rectangular window outside a six-symbol interval $[-3T_b, 3T_b]$. The transmitted signal propagates through a three-ray multipath channel, generating a continuous received pulse shape $h_c(t) = p(t) + 0.8p(t-0.25T_b) - 0.4p(t-2T_b)$. The input is sampled at time $t = nT_b/L$. White noise is added to the dis-



(a) (b)

Figure 1. MSE evolution with variable block length (a) $N_t = 200$ **(b)** $N_t = 1000$



(a) (b)

Figure 2. MSE evolution with time-varying channel (a) $N_t = 200$ **(b)** $N_t = 1000$

crete PAM sequence with an SNR of 30 dB, and equalizers with $N = 50$ coefficients are used. All subchannel coefficient vectors are initially null, except for $\mathbf{c}^{(L/2)}$ which has a single unitary central tap. The forgetting factors used in the RLS algorithm and in the unnormalized power estimate are set to 0.99.

Figure 1 depicts the evolution of MSE (Mean-Square Error) for oversampling factors $L = 1, 2, 3, 4$ and block lengths $N_t = 200, 1000$, as well as the MSE of a symbol-spaced Godard equalizer, averaged over 100 Monte Carlo trials. To establish the steady-state accuracy of the super-exponential curves, the MSE values obtained with the nonrecursive (batch) algorithm after 10 iterations using the full data record of 5000 symbols are also shown. The required Godard step size $\mu = 10^{-3}$ was empirically selected to maximize the convergence speed while avoiding instability.

The Godard algorithm converges in about 1000 iterations, whereas the convergence speed of the recursive super-exponential algorithm depends on block size. Although convergence is slower when larger blocks are used (measured in symbol intervals), fewer iterations are required before steady-state is reached since cumulants are more accurately estimated.

Time-varying channel: According to a Ricean fading model, independent, zero-mean Gaussian disturbances were added to each path gain of the previously used multipath channel. To simulate (relatively) slow channel variations, these disturbances were kept fixed in blocks of 100 symbol intervals, and updated with narrowband-filtered white noise. The power of each disturbance was 10 dB lower than the squared magnitude of the corresponding nominal gain, and the path delays were kept constant. The MSE evolution of the recursive and nonrecursive super-exponential algorithms, as well as that of the Godard equalizer, are shown in figure 2. The previous Godard adaptation step $\mu = 10^{-3}$ provided good tracking performance in the whole data record. The batch algorithm performs modestly, since a single coefficient vector is used to equalize all the symbols.

Regarding the recursive algorithm, its tracking performance is superior to that of the Godard equalizer when blocks of $N_t = 200$ symbols are used. In figure 2b the input statistics vary significantly within a data block, causing an increase in output MSE because the coefficient vector remains far from steady-state for each of the individual channel responses.

5. CONCLUSION

A recursive implementation of the super-exponential algorithm was proposed. Like the original nonrecursive algorithm, it is based on cumulant estimates using blocks of input and output samples. However, recursive propagation of several variables between consecutive blocks allows fewer samples to be used in each block. The RLS algorithm is used in computing part of the super-exponential iteration, and it provides a numerically robust tool that is important when fractionally-spaced sampling leads to numerically ill-conditioned input covariance matrices. The computational requirements of the recursive algorithm are essentially those of multichannel RLS.

The proposed approach replaces the difficult and channel-dependent selection of an adaptation step with the less critical choice of a block length. Since few blocks are required before the recursive algorithm reaches equilibrium, its convergence speed is greater than that of the fastest Godard equalizer for moderately large data blocks. Moreover, the transient behavior is robust in the sense that the number of required iterations does not strongly depend on channel characteristics due to the data whitening operation that is an integral part of the super-exponential algorithm. Adaptive step-size selection has also been addressed in recent normalized CMA algorithms, but the convergence speed still shows significant dependence on the channel characteristics.

By changing the order of the cumulants in the super-exponential cost function, non-CMA algorithms are obtained. These may prove more adequate with certain kinds of constellations or other deconvolution problems.

REFERENCES

[1] D. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Trans. on Comm.*, vol. COM-28, pp. 1867–1875, Nov. 1980.

[2] J. Gomes and V. Barroso, "A super-exponential algorithm for blind fractionally-spaced equalization," *IEEE Sig. Proc. Letters*, vol. 3, pp. 283–285, Oct. 1996.

[3] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, third ed., 1996.

[4] C. Papadias and D. Slock, "Normalized sliding window constant modulus and decision-directed algorithms: A link between blind equalization and classical adaptive filtering," *IEEE Trans. on Sig. Proc.*, vol. 45, pp. 231–235, Jan. 1997.

[5] B. Porat, *Digital Processing of Random Signals: Theory and Methods*, Prentice-Hall, 1994.

[6] O. Shalvi and E. Weinstein, "New criteria for blind deconvolution of nonminimum phase systems (channels)," *IEEE Trans. on Inf. Theory*, vol. 36, pp. 312–321, Mar. 1990.

[7] O. Shalvi and E. Weinstein, "Super-Exponential methods for blind deconvolution," *IEEE Trans. on Inf. Theory*, vol. 39, pp. 504–519, Mar. 1993.

[8] O. Tanrikulu, A. Constantinides and J. Chambers, "New normalized constant modulus algorithms with relaxation," *IEEE Sig. Proc. Letters*, vol. 4, pp. 256–258, Sept. 1997.