Performance of the SMLR Deconvolution Algorithm
Chong Yung Chi

Abstract—In this correspondence, we present a performance analysis regarding false alarms, correct detections, and the resolution of the well-known suboptimal maximum-likelihood deconvolution (MLD) algorithm, called the single most likely replacement (SMLR) algorithm. We assume that the source wavelet and statistical parameters are given a priori. We analytically show that the performance improves as the signal-to-noise ratio (SNR) increases and as the mainlobe width of the normalized autocorrelation function of the source wavelet decreases. For the same performance, a higher SNR is required as the mainlobe width of the normalized autocorrelation function increases. We also show some simulation results which are consistent with the analytic results.

I. INTRODUCTION

Seismic deconvolution deals with the estimation of the reflectivity sequence $\mu(k)$ with noisy measurements $z(k)$, $k = 1, 2, \ldots, N$, based on the convolutional model

$$
    z(k) = \mu(k) * v(k) + n(k) = \sum_{i=0}^{k} v(i) \mu(k-i) + n(k)
$$

where $n(k)$ is the measurement noise and $v(k)$ is the source wavelet. In the past decade, Mendel [1], Kormylo [2], Kormylo and Mendel [3] proposed a Bernoulli–Gaussian (B-G) model for the reflectivity sequence $\mu(k)$ as follows:

$$
    \mu(k) = r(k) \cdot q(k)
$$

where $r(k)$ is a white Gaussian random process with zero mean and variance $\sigma_r^2$ and $q(k)$ is a Bernoulli process for which

$$
    P_r[q(k)] = \begin{cases} 
      \lambda, & q(k) = 1 \\
      1 - \lambda, & q(k) = 0.
    \end{cases}
$$

Kormylo and Mendel [3] developed a suboptimal maximum-likelihood deconvolution (MLD) algorithm, called the single most likely replacement (SMLR) algorithm, based on the B-G model and the assumption that $n(k)$ is white Gaussian with zero mean and variance $\sigma_n^2$. This algorithm performs well and is computationally efficient [4], [5]. Other algorithms for estimating the B-G signal $\mu(k)$ with $z(k)$, $k = 1, 2, \ldots, N$, can be found in [6]–[10].

Regarding the performance of the SMLR algorithm, we observed that for some wavelets there are fewer false alarms and more correct detections, while for some other wavelets there are many false alarms and missing detections even when the signal-to-noise ratio (SNR) is the same. These observations motivated a performance analysis.

In this correspondence, we assume that statistical parameters $\lambda$, $\sigma_r^2$, $\sigma_n^2$ and the source wavelet $v(k)$ are given a priori and present, in addition to SNR, which characteristics of $v(k)$ determine the performance of the SMLR algorithm. In Section II, we briefly review the background of the SMLR deconvolution algorithm for detection of $q(k)$ and estimation of $r(k)$. We then present the associated performance analysis for the SMLR algorithm in Section III. In Section IV, some simulation results are shown to support the proposed analysis. Finally, we draw some conclusions from this analysis.

II. BACKGROUND OF THE SMLR DECONVOLUTION ALGORITHM

The convolutional model (1) can be expressed in the following linear vector form:

$$
    z = VQr + n
$$

where $z = (z(1), z(2), \ldots, z(N))'$, $Q = \text{diag}(q(1), q(2), \ldots, q(N))$, $r = (r(1), r(2), \ldots, r(N))'$, $n = (n(1), n(2), \ldots, n(N))'$ and $V = (v_1, v_2, \ldots, v_N)$ in which

$$
    v_k = (0, 0, \ldots, v(0), \ldots, v(N-k)).
$$

The covariance matrix $\Omega$ of $z$ is given by

$$
    \Omega = E[zz'] = \sigma_r^2 VQV' + \sigma_n^2 I
$$

where $q = (q(1), q(2), \ldots, q(N))'$ and $I$ is an $N \times N$ identity matrix.

The SMLR algorithm is an iterative algorithm for detecting $q(k)$ based on the log-likelihood ratio

$$
    \ln \Lambda(k, q) = \ln S(q|z) / S(q|z)
$$

where $S(q|z) = p(z|q)$, $q$ is a reference sequence and $q_0$ is a test sequence which differs from $q_0$ only at a single time location $k$. The detection procedure [3] is summarized as follows:
a. Compute $\ln \Lambda(k, q_0)$ for $k = 1, 2, \ldots, N$.

b. Assume that $\ln \Lambda(k', q_0) = \max \{\ln \Lambda(k, q_0), 1 \leq k \leq N\}$.

If $\ln \Lambda(k, q_0) > 0$, update $q_0(k')$ by $1 - q_0(k')$ and go to (a).

When $\ln \Lambda(k, q_0) \leq 0$ for all $1 \leq k \leq N$, the detection procedure is finished. The log-likelihood ratio $\ln \Lambda(k, q_0)$ was shown [3] to be

$$
    \ln \Lambda(k, q_0) = \frac{1}{2} \sum_{k=1}^{N} \left[ \frac{1}{1 + \sigma_n^2(1 - 2q(k))a_k} - \frac{1}{2} \ln \left( \frac{1}{1 + \sigma_n^2(1 - 2q(k))a_k} \right) \right]
$$

where

$$
    a_k = v_k^T \Omega_{-1} z
$$

and $\Omega = \Omega(q = q_0)$ (see (6)).

After detection of $q$, the maximum-likelihood estimate, $r_{ML}$, which is also equal to the minimum-variance estimate $r_{MV}$ because $r$ and $z$ are jointly Gaussian when $q(k)$ is known, is given by [1].

Manuscript received July 28, 1989; revised August 11, 1990. This work was supported by the National Science Council under Grant NSC 78-0404-E-002-27.

The author is with the Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, Republic of China.

IEEE Log Number 9101331.

1053-587X/91/0900-2082$1.00 © 1991 IEEE
analyze what determines the performance of the SMLR algorithm for (I-A) and (II-A). The other cases can be similarly performed for completeness and thus are omitted.

Case I: \( q_t(k) = \delta(k - k_t) \):

\( I-A: \ q_t(k) = 0, \) no spikes in \( q_t \).

The measurement vector \( z \) for this case can be easily seen from (4) to be

\[ z = r(k_t) \varepsilon_{k_t} + n. \]

One can also see, from (6), (10), and (9), that, \( \Omega_q = \sigma_w^2 I, \ a_q = \varphi(0)/\sigma_w^2 \) and \( f_t = v_t^\top \Omega_q^{-1} z = \frac{1}{\sigma_a^2} \varphi(k_t - k_t) r(k_t) + \frac{1}{\sigma_v^2} v_t^\top n \)

from which and (8) one can show that

\[
E[\ln \Lambda(k, q_t)] = \frac{F}{2(1 + F)} \left[ 1 + F \gamma^2(k - k_t) \right]
\]

\[
- \frac{1}{2} \ln (1 + F) + \ln \frac{\lambda}{1 - \lambda}
\]

\[
\approx 1 + \frac{1}{2} F \gamma^2(k - k_t) - \frac{1}{2} \ln F + \ln \frac{\lambda}{1 - \lambda}
\]

\[
= A_t(k, k_t), \quad (\text{since } F \gg 1)
\]

where we have used (17) to define \( A_t(k, k_t) \). Next, we discuss the effects of \( \gamma(k) \) and \( F \) on the performance based on (17).

Let us consider all possible cases about \( k_t \) as follows:

(I-A-1) \( k_t = k_1 \), a correct detection occurs;

(I-A-2) \( k_t \neq k_1 \), a false alarm occurs.

From (17), we see that max \{ \( E[\ln \Lambda(k, q_t)] \) \} = \( E[\ln \Lambda(k_t, q_t)] \).

Let

\[ \Delta(k) = E[\ln \Lambda(k, q_t)] - E[\ln \Lambda(k, q)] \]

\[ = (1 - \gamma^2(k - k_t))F/2. \]

It is easy to see that \( 0 \leq \Delta(k) \leq F/2 \) and that \( \Delta(k) \) increases as \( F \) increases and \( \gamma^2(k) \) decreases but has nothing to do with the length of \( \gamma(k) \) which is about twice the length of \( \gamma(k) \). When the mainlobe of \( \gamma(k) \) is narrow (i.e., \( \gamma^2(k) \ll 1 \) for \( k \neq 0 \)), \( \Delta(k) = (F/2) \) for \( k \neq k_t \). However, when the mainlobe of \( \gamma(k) \) is broad (i.e., \( \gamma^2(k) = 1, \ |k| \leq W \) for some \( W \)) it is then possible that \( \Delta(k) \) for \( |k - k_t| \leq W \) is very small when \( F \) is not large enough. In other words, a false alarm associated with the true spike located at \( k = k_1 \) could occur near \( k = k_1 \) when the mainlobe of \( \gamma(k) \) is broad and \( F \) is not large. Finally, \( \Delta(k) \) for \( k \neq k_t \) can be made arbitrarily large by increasing \( F \) or \( \gamma(k) \).

Therefore, we conclude that the performance is better for larger \( F \) and \( \gamma(k) \) with a narrower mainlobe.

Case II: \( q_t(k) = \delta(k - k_t) + \delta(k - k_t - 1), \ k_t \neq k_1; \)

\( II-A: \ q_t(k) = 0, \) no spikes in \( q_t \):

The mean of log-likelihood ratio \( \Lambda(k, q_t) \) can be shown to be

\[
E[\ln \Lambda(k, q_t)] = A_t(k, k_t) + \frac{1}{2} F \gamma^2(k - k_t).
\]

Let us consider the following situations for \( k_t \):

(II-A-1) \( k_t = k_1 \), a correction detection occurs;

(II-A-2) \( k_t = k_1 \), a correction detection occurs;

(II-A-3) \( k_t \neq k_1, k_t \neq k_2 \), a false alarm occurs.

From (19) and (17), we see that max \{ \( E[\ln \Lambda(k, q_t)] \) \} = \( E[\ln \Lambda(k_t, q)] \) when the mainlobe of \( \gamma(k) \) is narrow. We can infer that when the mainlobe of \( \gamma(k) \) is narrow, \( E[\ln \Lambda(k_t, q_t)] \) is much smaller than the mean of log-likelihood ratio \( \Lambda(k, q_t) \) for \( k_t \neq k_1 \) and \( k_t \neq k_2 \). How-
ever, max \{E[\ln \Lambda(k, q)]\} could happen at some \(k\) where \(k_1 < k < k_2\), when \(k_1\) is close to \(k_2\) and the mainlobe of \(\gamma(k)\) is broad. In other words, two close spikes could lead to a false alarm located between \(k = k_1\) and \(k = k_2\) when the mainlobe of \(\gamma(k)\) is broad. This also implies that the resolution is better for \(\gamma(k)\) with a narrow
mainlobe than for \(\gamma(k)\) with a broad mainlobe.

From the performed analyses we obtain the following conclusions:

(R1) The performance is better for a larger SNR and \(\gamma(k)\) with
a narrower mainlobe;

(R2) the resolution is better for \(\gamma(k)\) with a narrower mainlobe;

(R3) the performance is not dependent on the wavelet length;

(R4) the performance can be infinitely improved by increasing
SNR no matter when the mainlobe of \(\gamma(k)\) is broad or nar-
row;

(R5) when the mainlobe of \(\gamma(k)\) is broad, false alarms cannot
be removed by increasing SNR, but their amplitudes tend
to be smaller for a larger SNR by Lemma 1;

(R6) for the same performance, a higher SNR is required for
\(\gamma(k)\) with a broad mainlobe than for \(\gamma(k)\) with a narrow
mainlobe.

IV. SIMULATION EXAMPLES

In order to illustrate the analytic results presented in Section III,
we selected two different wavelets \(v_1(k)\) (solid line) and \(v_2(k)\)
(dashed line) shown in Fig. 1(a). The associated normalized auto-
correlation functions \(\gamma_1(k)\) (solid line) and \(\gamma_2(k)\) (dashed line)
are shown in Fig. 1(b) which apparently indicates the different
mainlobe widths of \(\gamma_1(k)\) and \(\gamma_2(k)\). The synthetic data was
generated with parameters \(\lambda = 0.07\) and \(\sigma^2 = 0.0225\).
\(q(k) = 0\) for all \(k\) was used to initialize the SMLR algorithm.
The deconvolved results associated with \(v_1(k)\) and \(v_2(k)\) are shown in Figs. 2 and 3, respectively, where *’s denote true spikes and bars denote estimated ones.

From Fig. 2, where SNR = 10 dB, we see that the deconvolved
results are very good in spite of two false alarms and five missing
spikes whose amplitudes are too small to be detected since SNR is
not high enough. Note that the two close spikes located at \(k = 55\)
and 57 were correctly detected, and the two close spikes located at
\(k = 262\) and 263 were also correctly detected. These results are
consistent with the predicted results (R1) and (R2) since the main-
lobe of \(\gamma_1(k)\) is narrow.

From Fig. 3(a) where SNR = 10 dB, we see that only one spike
at \(k = 288\) was correctly detected. The other bars in this figure are
all false alarms. As analyzed in Case I-A, each false alarm is
associated with an isolated spike in its vicinity. Note that the two
spikes at \(k = 55\) and 57 were converted into a false alarm at \(k = 56\).
and the two spikes at \(k = 57\) and \(k = 66\) led to a false alarm at
\(k = 61\). These observations are consistent with (R2).

Next, let us compare Fig. 2 with Fig. 3(a) where both SNR’s are
equal to 10 dB. Although the wavelet lengths of \(v_1(k)\) and \(v_2(k)\)
are about the same, the mainlobe widths of \(\gamma_1(k)\) and \(\gamma_2(k)\) are very
different. Obviously, the results shown in Fig. 2 are much better
than those shown in Fig. 3(a). Again, this is consistent with (R1)
and (R3).

Figs. 3(b) and (c) show the deconvolved results for SNR equal
to 30 and 40 dB, respectively. From Fig. 3(b), one can see that
there are more correct detections but there are still many false
alarms. These results are consistent with (R5). From Fig. 3(c),
again, one can see that there are still many false alarms in spite of
increase of SNR. Nevertheless, false alarm amplitudes and ampli-
tude estimation errors of detected spikes decrease as SNR in-
creases. Again, these results are consistent with the previous con-
cclusions (R4) and (R5).

Finally, comparing Fig. 2 where SNR = 10 dB with Fig. 3(c)
where SNR = 40 dB, we see that their performances are compar-
able but SNR’s are very different. This is also consistent with our
conclusion (R6).

V. CONCLUSIONS

In this correspondence, we have presented an analysis based on
a heuristic approach for the performance of a well-known subop-
timal iterative MLD algorithm, the SMLR algorithm, for B-G pro-
cesses assuming that \(\lambda, \sigma^2,\) and \(\sigma^2\) and \(v(k)\) were given a priori.
From this analysis, we obtained six main conclusions (R1) through
(R6) summarized at the end of Section III with regard to the per-
where \( u = (r(k_1), r(k_2), \ldots, r(k_L)) \) and \( W = (\psi_1, \psi_2, \ldots, \psi_L) \). It is well known that the maximum-likelihood estimate \( \hat{u} \) is equal to the minimum-variance estimate since \( u \) and \( \varphi \) are jointly Gaussian as follows:

\[
\hat{u}_{ML} = \hat{u}_{MV} = E[u | z] = \sigma_u^2 W' (\sigma_u^2 W W' + \sigma_z^2 I)^{-1} z
\]

which is surely consistent with (11). It is also well known that (12)

\[
E[\varphi \varphi'] = \sigma_z^2 I - \sigma_z^2 W W' (\sigma_z^2 W W' + \sigma_z^2 I)^{-1} W
\]

\[
= \sigma^2_z I - \sigma_z^2 W W' \left( \frac{1}{\sigma_z^2} I - \frac{1}{\sigma_z^2} W W' \right)(W W')^{-1} W
\]

\[
= \sigma^2_z I - \sigma_z^2 W W' \left( \frac{1}{\sigma_z^2} I - \frac{1}{\sigma_z^2} W W' \right)(W W')^{-1} W
\]

\[
= \sigma^2_z I - \sigma_z^2 I = 0.
\]

(A4)

From (A4), we have that

\[
\lim_{\sigma^2_z \to 0} E[\varphi \varphi'] = \sigma_z^2 I - \left( \frac{1}{\sigma_z^2} I - \frac{1}{\sigma_z^2} W W' \right)(W W')^{-1} W
\]

\[
= \sigma_z^2 I - \sigma_z^2 I = 0.
\]

(A5)

In deriving (A5), we have used the first-order Taylor series approximation. From (12), we see that SNR is inversely proportional to \( \sigma_z^2 \). Therefore, (A5) leads to Lemma 1.

**ACKNOWLEDGMENT**

The research described in this paper was performed at the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan.

**REFERENCES**


On the Efficiency of Parallel Pipelined Architectures
Luciano da Fontoura Costa and Jan Frans Willem Slaets

Abstract—This correspondence describes an approach to help the design of efficient dedicated parallel pipelined architectures. Based on a previous publication where conditions for determining the most efficient mapping of digital signal processing algorithms are proposed, we develop a new approach that eliminates the restrictions and deficiencies in that paper. As an example of the presented approach, we design an efficient parallel pipelined architecture for the “butterfly” of a fast Fourier transform algorithm using operators with different execution rates.

I. INTRODUCTION

The use of dedicated computer architectures is growing rapidly in digital signal processing (DSP) and image processing. Considerable efforts have been made to enhance the performance of these systems not only by optimizing the algorithms, such as trying to minimize arithmetic operations (e.g., [2]), but also through the design of special hardware such as dedicated parallel pipelines architectures. Several DSP (e.g., FFT, filtering, and convolution) and image processing algorithms (e.g., border detection, filtering, correlation, Hough transform), usually with regular and deterministic processing sequences, are suitable for implementation in such architectures [1] and [3].

This correspondence presents a methodology to help the synthesis of dedicated parallel pipelined architectures in order to optimize the utilization of the hardware resources. This subject has already been worked by Siomulas and Bowen [1]; however, that paper has been prejudiced by the following deficiencies:

1) The definition of \( t_i \) (1) in Section II-C is not appropriate:

\[
t_i = \left[ \frac{N_i}{r} \right] = \text{an integer}. \tag{1}
\]

For example, for a given \( i \), if \( N_i = 2 \) and \( r = 10^6 \) ops (operations per second), we have that \( t_i = \left[ \frac{2 \times 10^6}{r} \right] = 1 \) for \( p_i \geq 2 \times 10^6 \) (please refer to [1] for the meaning of \( t_i, N_i, r \), and \( p_i \)).

A more appropriate definition is

\[
t_i = \left[ \frac{N_i}{p_i/r} \right] = \text{a real number}. \tag{2}
\]

2) The inadequate definition of \( t_i \) invalidates Theorems 1 and 2.

3) The proof of Theorems 1 and 2 considers only situations where all the operators have the same operation rate \( r \).

II. CONSIDERATIONS AND DEFINITIONS

A parallel pipelined architecture intended to execute a specific algorithm is here understood as a pipelined architecture with parallel operation within each of its stages, each stage corresponding to a level of the algorithm. The following definitions and assumptions are used henceforth:

1) An algorithm is defined as a finite number of operations partitioned into \( N \) levels.

2) A level in the algorithm is defined as the set of operations which can be started simultaneously. It is easy to verify that the levels, as defined, are determined from the independence of intermediate results in the algorithm.

3) An algorithm with the above characteristics can be implemented in a dedicated parallel pipelined architecture composed of \( N \) stages, each stage corresponding to one of the \( N \) levels of the algorithm. The operations in each stage are then performed by suitable operators.

4) The algorithm is continuously executed by the dedicated parallel pipelined architecture.

5) There are \( k_i \) types of operators in each level \( i = 1, 2, \ldots, N \). These operators can be simple operators such as adders and multipliers or complex operators such as the “butterfly” of an FFT.

6) The efficiency is defined in a similar way to [1] in order to express the utilization of the hardware resources (operators).

III. CONDITIONS FOR MAXIMUM EFFICIENCY

The efficiency of a parallel pipelined architecture is maximum (tends to 1) if and only if for each type \( j \) of operator \( (j \in \{1, 2, \ldots, k_j\}) \) within each stage \( i \) of the pipeline \((i \in \{1, 2, \ldots, N\})\) conditions (3) and (4) are both met:

\[
T_{i,j} = \frac{F_{i,j}}{P_{i,j}} t_{i,j} - T \tag{3}
\]

\[
F_{i,j} = \text{an integer multiple of } P_{i,j} \tag{4}
\]

where

\( F_{i,j} \) number of operations of type \( j \) in level \( i \) of the algorithm,

\( T_{i,j} \) the time interval for execution of all operations of type \( j \) in stage \( i \) of the pipeline,

\( P_{i,j} \) number of operators of type \( j \) in stage \( i \) of the processor,

\( t_{i,j} \) execution time for one execution of operator of type \( j \) in stage \( i \),

\( T \) a real number representing the basic cycle time of the pipeline.