Maximum A Posteriori Deconvolution of Sparse Spike Trains

Kjetil F. Kaaresen[1]

Department of Mathematics
University of Oslo
P.B. 1053 Blindern
N-0316 Oslo, Norway
E-mail: kjetilka@math.uio.no

February 27, 1996

This research report is organized as two separate papers. The first paper describes a new deconvolution algorithm for sparse spike trains. The second paper compares the new algorithm to a number of existing alternatives.

[1]Supported by the Research Council of Norway.

Deconvolution of Sparse Spike Trains by Iterated Window Maximization

Kjetil F. Kaaresen

Abstract— **A new algorithm for deconvolution of sparse spike trains is presented. To maximise a joint MAP criterion, an initial configuration is iteratively improved through a number of small changes. Computational savings are achieved by pre-computing and storing two correlation functions, and by employing a window strategy. The resulting formulas are simple, intuitive, and efficient. In addition, they allow much more complicated transitions than state-space solutions such as Kormylo and Mendel's Single-Most-Likely-Replacement algorithm. This makes it possible to reduce significantly the probability that the algorithm terminates in a local maximum.**

Application of the algorithm is illustrated on a synthetic data set.

Keywords— **deconvolution, sparse spike train, MAP estimation, iterated window maximization, parallel processing.**

I. Introduction

MANY natural phenomena can be approximated by the discrete-time convolutional model

$$z(n) = \sum_{k=-\infty}^{\infty} h(n-k)x(k) + e(n). \qquad (1)$$

For example, in echographic applications, h will be a transmitted wavelet, x is the reflectivity of the material, e is additive noise, and z is the observed reflection. The purpose of deconvolution is to estimate x, based on knowledge of z and h. In practical applications h will often be narrow-band. Such problems are ill-conditioned, and highly different x will be compatible with the same observation z. The consequence is that meaningful results can only be achieved by employing some a priori information about x.

The model considered here is based on the assumption that only a small part of the components of x are non-zero. Signals with this characteristic are commonly referred to as "sparse spike trains" or, under a special distributional assumption, as Bernoulli-Gaussian processes. Such models arise naturally within a number of fields: e.g. seismic exploration [1], ultrasonic non-destructive evaluation [2], communication theory [3], and speech processing [4].

Due to the sparse structure of x, classical linear methods such as Wiener filtering are not appropriate, and a large number of alternatives have been proposed. An incomplete list includes: One-at-a-time spike extraction techniques [3], [5], the Single-Most-Likely-Replacement (SMLR) algorithm [6], Viterbi algorithm detector [7], L_p deconvolution [2], [8], stochastic Bayesian methods [9], and Multipulse methods [10].

K. F. Kaaresen is with the Department of Mathematics, University of Oslo, P.B. 1053 Blindern, N-0316 Oslo, Norway. E-mail: kjetilka@math.uio.no.

In the present article a Bayesian viewpoint will be taken, and reconstruction will be based on a Maximum A Posteriori (MAP) estimator. The sparse spike train will be represented by two vectors, $\mathbf{a}$ giving the amplitudes of the spikes and $\mathbf{t}$ giving their (time) positions. The MAP estimator is the values of $\mathbf{a}$ and $\mathbf{t}$ that maximizes the *posterior* density $p(\mathbf{a}, \mathbf{t}|\mathbf{z})$. For given $\mathbf{t}$ the optimal amplitudes, $\hat{\mathbf{a}}$, is found by linear methods. But $p(\hat{\mathbf{a}}, \mathbf{t}|\mathbf{z})$ is non-linear and maximization with respect to $\mathbf{t}$ is carried out by iteration. Since $\hat{\mathbf{a}}$ depends on $\mathbf{t}$, each evaluation of $p(\hat{\mathbf{a}}, \mathbf{t}|\mathbf{z})$ requires the initialization and inversion of a new linear system. Fast initialization is achieved by exploiting a simple relationship between the linear system and two correlation functions. These are computed and stored prior to the iteration.

The iterative search starts by comparing a reference value of $\mathbf{t}$ to a number of neighbors. The transitions linking two neighbors will typically consist of changing one or a couple of components. As soon as a neighbor is found which increases $p(\hat{\mathbf{a}}, \mathbf{t}|\mathbf{z})$, it is adopted as the new reference value, and the search is repeated. The iteration stops when no improving neighbors can be found. Similar to existing MAP/Maximum Likelihood estimators [6], [7], [11], [12], the search may terminate with a sub-optimal value for $\mathbf{t}$.

Because only neighboring $\mathbf{t}$ values are compared, great computational savings are possible. The strategy is to recompute only the components of $\hat{\mathbf{a}}$ in a small window covering the area where the neighboring $\mathbf{t}$ values differ. Although the immediate result is only local optimality of $\hat{\mathbf{a}}$, increase of $p(\hat{\mathbf{a}}, \mathbf{t}|\mathbf{z})$ is still guaranteed for each accepted update. Furthermore, repeating the local optimization with changing window positions will continuously improve the global fit of $\hat{\mathbf{a}}$. As long as $\mathbf{t}$ changes the fit will remain an approximation, but when $\mathbf{t}$ reaches its final value, $\hat{\mathbf{a}}$ will converge quickly to its corresponding global optimum.

Combination of window maximization and initialization from correlation functions yields highly efficient evaluation of $p(\hat{\mathbf{a}}, \mathbf{t}|\mathbf{z})$. The major burden is inversion of the linear system, whose order will typically vary in the range 1-5.

The MAP criterion and some of the formulas employed here are similar to those used by Kwakernaak [3]. But Kwakernaak maximizes his criterion by a one-at-a-time spike extraction technique, which may easily estimate the spikes at wrong locations when the wavelets are overlapping [5]. This problem is not shared by the approach developed here. In addition, higher efficiency may be achieved since much smaller matrices need to be inverted.

The present iterative search has more in common with the approach used by Kormylo and Mendel in their pioneering work on the SMLR detector [6]. But a major difference is that the efficiency of the SMLR detector is based on

a restrictive definition of neighboring sequences. No such restrictions apply here, and a higher degree of optimality can be achieved. In addition, the present approach may in many cases lead to faster computation. In particular this will be true for wavelets that do not permit a low-order state-space representation.

The rest of the paper is organized as follows: In Section II the algorithm is derived in detail. Selection of parameters is discussed in Section III. In Section IV the performance of the algorithm is illustrated on a synthetic data set, and parallelization possibilities are pointed out in Section V.

In the following no assumptions on the wavelet such as symmetric, minimum phase or low-order ARMA will be needed, but it is necessary that it has a finite support. In most cases of practical interest this should be satisfied, either exactly or by appropriate truncation.

II. Derivation of the Algorithm

A. Convolutional Model

Assume that a data record of N samples is observed and let M denote the number of spikes. In terms of $\mathbf{a}$ and $\mathbf{t}$ the model (1) becomes

$$z(n) = \sum_{i=1}^{M} h(n - t_i)a_i + e(n), \quad n = 1, 2, \ldots, N. \tag{2}$$

Equation (2) can be rewritten in matrix as form as

$$\mathbf{z} = \mathbf{H}\mathbf{a} + \mathbf{e}, \tag{3}$$

with obvious interpretations of $\mathbf{z}$ and $\mathbf{e}$. Note that $\mathbf{H}$ depends on $\mathbf{t}$ and is given by $H_{ni} = h(n - t_i)$. Thus, each column of H contains a copy of the wavelet that are shifted to the corresponding spike position.

B. Distributional Assumptions and MAP Estimator

By Bayes formula the posterior density can be factored as

$$p(\mathbf{a}, \mathbf{t}|\mathbf{z}) \propto p(\mathbf{z}|\mathbf{a}, \mathbf{t})p(\mathbf{a}|\mathbf{t})p(\mathbf{t}). \tag{4}$$

The following distributional assumptions are introduced: The noise is zero mean Gaussian and white, independent of $\mathbf{a}$ and $\mathbf{t}$, and has variance σ_e^2. This implies a Gaussian likelihood, which except for constant factors, can be written as

$$p(\mathbf{z}|\mathbf{a}, \mathbf{t}) \propto \exp\left\{-(\mathbf{a}'\mathbf{H}'\mathbf{H}\mathbf{a} - 2\mathbf{z}'\mathbf{H}\mathbf{a})/(2\sigma_e^2)\right\}. \tag{5}$$

The prior distribution of $\mathbf{a}$ given $\mathbf{t}$ is also zero mean Gaussian and white. The variance is σ_a^2. Thus

$$p(\mathbf{a}|\mathbf{t}) = \left(2\pi\sigma_a^2\right)^{-M/2} \exp\left\{-\mathbf{a}'\mathbf{a}/(2\sigma_a^2)\right\}. \tag{6}$$

The prior density of $\mathbf{t}$ can be arbitrary as long as it can easily be evaluated at any given point. If $\mathbf{t}$ is a geometric process, then $\mathbf{x}$ has the commonly used Bernoulli-Gaussian distribution [6], [7], [9], [13]. This will be referred to as the Bernoulli case.

Combining (4), (5), and (6) and reorganizing gives the following expression for the log-posterior density:

$$\begin{aligned}\ln p(\mathbf{a}, \mathbf{t}|\mathbf{z}) = \\ &- \left[(\mathbf{a} - \mathbf{S}^{-1}\mathbf{v})'\mathbf{S}(\mathbf{a} - \mathbf{S}^{-1}\mathbf{v}) - \mathbf{v}'\mathbf{S}^{-1}\mathbf{v}\right]/(2\sigma_e^2) \\ &- \tfrac{M}{2}\ln\left(2\pi\sigma_a^2\right) + \ln p(\mathbf{t}) + \text{const.}\end{aligned} \tag{7}$$

The matrices

$$\mathbf{S} = \mathbf{H}'\mathbf{H} + \delta\mathbf{I} \quad \text{and} \quad \mathbf{v} = \mathbf{H}'\mathbf{z} \tag{8}$$

will be important in the following. The parameter δ is σ_e^2/σ_a^2 and can be thought of as an inverse signal-to-noise ratio. Since $\mathbf{S}$ is positive definite symmetric, it is clear that the maximizing value of $\mathbf{a}$ is

$$\hat{\mathbf{a}} = \mathbf{S}^{-1}\mathbf{v}. \tag{9}$$

The remaining problem is to maximize $\ln p(\hat{\mathbf{a}}, \mathbf{t}|\mathbf{z})$ with respect to $\mathbf{t}$. Defining $\theta_1 = 2\sigma_e^2$ and $\theta_2 = \sigma_e^2\ln(2\pi\sigma_a^2)$, it is easily seen that it is equivalent to maximize

$$l(\mathbf{t}) = \mathbf{v}'\hat{\mathbf{a}} + g(\mathbf{t}), \tag{10}$$

where

$$g(\mathbf{t}) = \theta_1 \ln p(\mathbf{t}) - \theta_2 M. \tag{11}$$

Unfortunately, there is no easy way to locate the maximum of $l(\mathbf{t})$, even for particularly simple $g(\mathbf{t})$. Except in pathological cases, such as when $g(\mathbf{t})$ is an increasing function of M, exact maximization seems to require something close to an exhaustive search. Evaluating the function for all 2^N possible values of the argument soon gets prohibitive as N grows. The solution is to limit the search by the iterative procedure described in the introduction. Computationally efficient formulas will be derived in the next two sub-sections.

An interpretation of the criterion to be maximized is possible. For given $\mathbf{t}$, (9) is the Bayes estimator of $\mathbf{a}$, and if $\delta = 0$ it is the least squares estimator from multivariate regression. Using some algebra on (10) shows that

$$l(\mathbf{t}) = \|\mathbf{z}\|^2 - \|\hat{\mathbf{e}}\|^2 - \delta\|\hat{\mathbf{a}}\|^2 + g(\mathbf{t}), \tag{12}$$

where $\hat{\mathbf{e}} = \mathbf{z} - \mathbf{H}\hat{\mathbf{a}}$. The first term of (12) is only a constant. The second term is the square-sum of residuals, and the third is the square-sum of estimated amplitudes weighted by an inverse signal-to-noise ratio.

With the Bernoulli assumption $g(\mathbf{t})$ also assumes a particularly simple form. In that case $p(\mathbf{t}) = \lambda^M(1 - \lambda)^{N-M}$, where λ is the probability of a spike at any given point. Inserting this in (11), ignoring a constant term, and defining $\theta = \theta_2 - \theta_1\ln[\lambda/(1 - \lambda)]$, yields $g(\mathbf{t}) = -\theta M$, which is simply a deduction proportional to the number of estimated spikes.

C. Initialization of Matrix Elements from Correlation Functions

The criterion (10) depends on the two matrices $\mathbf{S}$ and $\mathbf{v}$. Note that their dimensions, $M \times M$ and $M \times 1$, depend on the number of spikes, which will normally be much

smaller than the number of data points. Note further that the matrices depend on $\mathbf{t}$ (through $\mathbf{H}$) and must therefore be reinitialized for each new candidate value of $\mathbf{t}$. Direct computation from the defining equations (8) would be burdensome, but can fortunately be avoided:

Assume that the wavelet has finite support, and let D be such that $h(d) = 0$ for $|d| > D$. (No loss of generality follows by assuming that the support of h is placed roughly symmetrically around zero.) Consider two copies of the wavelet that are separated by a distance d, and introduce their correlation:

$$\begin{aligned} c_{hh}(d) &= c_{hh}(-d) = \sum_{k=-\infty}^{\infty} h(k-d)h(k) \\ &= \begin{cases} \sum_{k=d-D}^{D} h(k-d)h(k), & d = 0,1,\ldots,2D \\ 0, & d = 2D+1,\ldots \end{cases} \end{aligned} \tag{13}$$

Consider also a copy of the wavelet placed at position n, and introduce its correlation with the observed data:

$$\begin{aligned} c_{hz}(n) &= \sum_{k=1}^{N} h(k-n)z(k) \\ &= \sum_{k=\max(1,n-D)}^{\min(N,n+D)} h(k-n)z(k), \quad n = 1,2,\ldots,N. \end{aligned} \tag{14}$$

The elements of $\mathbf{v}$ can now be given as:

$$\begin{aligned} v_i &= \sum_{k=1}^{N} H_{ki}z_k = \sum_{k=1}^{N} h(k-t_i)z(k) \\ &= c_{hz}(t_i). \end{aligned}$$

To link $\mathbf{S}$ and c_{hh}, assume for simplicity that all t_i are chosen such that the columns of $\mathbf{H}$ contain non-truncated versions of h. This is satisfied if $D < t_i \leq N - D$. Consider the non-diagonal elements of $\mathbf{S}$ first:

$$\begin{aligned} S_{ij} &= \sum_{k=1}^{N} H_{ki}H_{kj} = \sum_{k=1}^{N} h(k-t_i)h(k-t_j) \\ &= \sum_{k=-\infty}^{\infty} h(k-t_i)h(k-t_j) \\ &= \sum_{k=-\infty}^{\infty} h\big(k-(t_i-t_j)\big)h(k) \\ &= c_{hh}(|t_i-t_j|). \end{aligned} \tag{15}$$

The diagonal elements of $\mathbf{S}$ are constant, and given by

$$S_{ii} = c_{hh}(0) + \delta.$$

In sum, it has been shown that if the two correlation functions (13) and (14) are computed and stored prior to the iterative phase of the algorithm, the necessary matrices can be initialized directly in each iteration, without any computation. The implication is that the work of computing $l(\mathbf{t})$ will be independent of the wavelet length and affected only by the number of spikes. Note finally that the correlation functions may be reformulated as a convolution and computed by the fast Fourier transform. This may be more efficient than direct computation for long wavelets.

D. Local Maximization

If the number of spikes is large, computing $\hat{\mathbf{a}} = \mathbf{S}^{-1}\mathbf{v}$ may still require a substantial effort. To reduce the dimension of the problem, only a subset of the components will be recomputed at each iteration. As a motivation, suppose the algorithm has determined a good fit of $\mathbf{a}$ for a given $\mathbf{t}$-value. The next step is to change a few components of $\mathbf{t}$ and perform a new fitting of $\mathbf{a}$. The new optimal fit may possibly be different from the old one in all components, but large changes are likely to take place only close to the changed components of $\mathbf{t}$. The large number of remaining amplitudes will probably be quite optimal already. In particular, when the algorithm is far from convergence, it seems inefficient to spend a lot of time on doing minor adjustments to these.

Maximization will thus be constrained to a small window, w. The window will be chosen to cover at least the area where the competing $\mathbf{t}$ values differ. The components of $\mathbf{a}$ and $\mathbf{t}$ that are inside the window will be denoted by $\mathbf{a}^w$ and $\mathbf{t}^w$. For each candidate value of $\mathbf{t}^w$ the posterior density will be maximized with respect to $\mathbf{a}^w$, giving a maximizing value denoted by $\hat{\mathbf{a}}^w$. Finally, the current configuration will be updated with the $\mathbf{t}^w$ and $\hat{\mathbf{a}}^w$ pair corresponding to the largest value of the posterior density.

It is proved in Appendix B that this window maximization strategy still guarantees convergence to globally optimal amplitudes. The condition is that the window positions must be systematically changed to contain all spikes, and that local maximization must be repeated until neither $\mathbf{t}$ nor $\hat{\mathbf{a}}$ changes any more. Similar to the non-windowed case, the optimality is relative to the final value of $\mathbf{t}$, which may still be sub-optimal. It is possible that window maximization may lead the algorithm to converge to a different (and possibly less optimal) local maximum for $\mathbf{t}$. But simulation tests indicates that this is not a severe problem, provided that the window size is not chosen very small relative to the length of the wavelet.

It will now be shown that the formulas for windowed maximization can be obtained from previous results. In addition to the notation introduced above, let $\mathbf{a}^{\overline{w}}$ and $\mathbf{t}^{\overline{w}}$ be the components of $\mathbf{a}$ and $\mathbf{t}$ that are outside the window. Start by restating the matrix model (3) in a partitioned form:

$$\mathbf{z} = \begin{pmatrix} \mathbf{H}^w, & \mathbf{H}^{\overline{w}} \end{pmatrix} \begin{pmatrix} \mathbf{a}^w \\ \mathbf{a}^{\overline{w}} \end{pmatrix} + \mathbf{e}. \tag{16}$$

The blocking of $\mathbf{H}$ corresponds to that of $\mathbf{a}$ and $\mathbf{t}$. The components of $\mathbf{a}$ that are inside the window may without loss of generality be positioned first in the vector. (If this

is not the case, it can be arranged by performing the same permutation to the components of $\mathbf{a}$ and the columns of $\mathbf{H}$. Such an operation will not change the equation.) Further, rewrite (16) as

$$\mathbf{z}^w = \mathbf{H}^w\mathbf{a}^w + \mathbf{e}, \tag{17}$$

where

$$\mathbf{z}^w = \mathbf{z} - \mathbf{H}^{\overline{w}}\mathbf{a}^{\overline{w}}. \tag{18}$$

Consider now the posterior density (4), which can be factored as

$$p(\mathbf{a},\mathbf{t}|\mathbf{z}) = p(\mathbf{a}^w,\mathbf{t}^w|\mathbf{z},\mathbf{a}^{\overline{w}},\mathbf{t}^{\overline{w}})p(\mathbf{a}^{\overline{w}},\mathbf{t}^{\overline{w}}|\mathbf{z}). \tag{19}$$

The second factor is constant in the present setting and can be ignored. The first factor can be rewritten as follows:

$$\begin{aligned} & p(\mathbf{a}^w,\mathbf{t}^w|\mathbf{z},\mathbf{a}^{\overline{w}},\mathbf{t}^{\overline{w}}) \\ &= p(\mathbf{a}^w,\mathbf{t}^w|\mathbf{z}^w,\mathbf{a}^{\overline{w}},\mathbf{t}^{\overline{w}}) \\ &\propto p(\mathbf{z}^w|\mathbf{a}^w,\mathbf{t}^w,\mathbf{a}^{\overline{w}},\mathbf{t}^{\overline{w}})\cdot \\ &\quad p(\mathbf{a}^w|\mathbf{t}^w,\mathbf{a}^{\overline{w}},\mathbf{t}^{\overline{w}})p(\mathbf{t}^w|\mathbf{a}^{\overline{w}},\mathbf{t}^{\overline{w}}) \\ &= p(\mathbf{z}^w|\mathbf{a}^w,\mathbf{t}^w)p(\mathbf{a}^w|\mathbf{t}^w)p(\mathbf{t}^w|\mathbf{t}^{\overline{w}}). \end{aligned} \tag{20}$$

Here, the first equality is due to the fact that the conditioning variables in the second density are a one-to-one transformation of those in the first. The proportionality is Bayes formula, and the last equality follows from (17) and the distributional assumptions of Section II-B. Comparing (17) with (3) and (20) with (4) makes it clear that the situation is completely analogous to the non-windowed case. After substituting $\mathbf{a}$ with $\mathbf{a}^w$, $\mathbf{t}$ with $\mathbf{t}^w$, $\mathbf{z}$ with $\mathbf{z}^w$ and $p(\mathbf{t})$ with $p(\mathbf{t}^w|\mathbf{t}^{\overline{w}})$, all the formulas derived in Section II-B can be used.

It is possible to interpret this result. Note from (18) that $\mathbf{z}^w$ is obtained from $\mathbf{z}$ by removing the effect of the spikes outside the window. It is intuitively reasonable that the amplitudes within the window should be fitted to the "unexplained" part of $\mathbf{z}$.

To make the window maximization effective, it is important that the matrices $\mathbf{S}^w = (\mathbf{H}^w)'\mathbf{H}^w + \delta\mathbf{I}$ and $\mathbf{v}^w = (\mathbf{H}^w)'\mathbf{z}^w$ can be initialized efficiently. Consider first $\mathbf{v}^w$, which by (18) can be written as

$$\mathbf{v}^w = (\mathbf{H}^w)'\mathbf{z} - (\mathbf{H}^w)'\mathbf{H}^{\overline{w}}\mathbf{a}^{\overline{w}}.$$

The components of the second term can be found from the correlation functions (13) and (14) as follows:

$$\begin{aligned} \left((\mathbf{H}^w)'\mathbf{H}^{\overline{w}}\mathbf{a}^{\overline{w}}\right)_i &= \sum_{kl} H^w_{ki}H^{\overline{w}}_{kl}a^{\overline{w}}_l \\ &= \sum_l a^{\overline{w}}_l \sum_k h(k-t^w_i)h(k-t^{\overline{w}}_l) \\ &= \sum_{\{l:|t^w_i - t^{\overline{w}}_l|\le 2D\}} a^{\overline{w}}_l c_{hh}(|t^w_i - t^{\overline{w}}_l|). \end{aligned} \tag{21}$$

The derivation for the first term of $\mathbf{v}^w$ and for $\mathbf{S}^w$ is completely analogous to the non-windowed case, and is omitted. For easy reference all formulas needed by the iterative part of the algorithm are now stated:

$$S^w_{ij} = \begin{cases} c_{hh}(|t^w_i - t^w_j|), & i \neq j, \\ c_{hh}(0) + \delta, & i = j, \end{cases} \tag{22}$$

$$v_i = c_{hz}(t^w_i) - \sum_{\{l:|t^w_i - t^{\overline{w}}_l|\le 2D\}} a^{\overline{w}}_l c_{hh}(|t^w_i - t^{\overline{w}}_l|), \tag{23}$$

$$\hat{\mathbf{a}}^w = (\mathbf{S}^w)^{-1}\mathbf{v}^w, \tag{24}$$

$$l^w(\mathbf{t}^w) = (\mathbf{v}^w)'\hat{\mathbf{a}}^w + g^w(\mathbf{t}^w), \tag{25}$$

$$g^w(\mathbf{t}^w) = \begin{cases} \theta_1 \ln p(\mathbf{t}^w|\mathbf{t}^{\overline{w}}) - \theta_2 M^w, & \text{In general,} \\ -\theta M^w, & \text{Bernoulli.} \end{cases} \tag{26}$$

In (26) M^w denotes the number of spikes within the window. Equations (25) and (26) can be seen to be valid also in the case of a window containing no spikes, provided that $(\mathbf{v}^w)'\hat{\mathbf{a}}^w$ is interpreted as zero.

Finally note that (22) is derived under the assumption that no spikes are closer to the border than D (the half-length of the wavelet). If this assumption is removed, it can be seen that such spikes will normally be estimated somewhat too small and possibly at wrong locations. Ignoring this should usually be of minor importance, since typically $D << N$. Alternatively, it can be seen that the border effects can be removed at the expense of computing and storing an additional $2D^2 + D$ correlation elements. Each element would correspond to a possible combination of truncated versions of h. This would also allow (highly uncertain) estimation of spikes slightly outside the observed data record.

E. Algorithm

Based on the formulas of the previous section, the complete algorithm is specified:

1. Compute the correlation functions c_{hh} and c_{hz} by (13) and (14). Initialize $\mathbf{t}$ and $\hat{\mathbf{a}}$ with suitable starting values.
2. Choose a window w, containing the spikes $\mathbf{t}^w$.
3. Compute $\hat{\mathbf{a}}^w$ and $l(\mathbf{t}^w)$ using (22)-(26).
4. Choose a new candidate value $\mathbf{t}^{w,c}$, which differ from $\mathbf{t}^w$ in one or more components.
5. Compute $\hat{\mathbf{a}}^{w,c}$ and $l(\mathbf{t}^{w,c})$ using (22)-(26).
6. if $l(\mathbf{t}^{w,c}) > l(\mathbf{t}^w)$
   update $\mathbf{t}$ with $\mathbf{t}^{w,c}$ and $\hat{\mathbf{a}}$ with $\hat{\mathbf{a}}^{w,c}$. Continue from 9.
7. As long as more candidates remain to be tested in this window, continue from 4.
8. No improvement found for $\mathbf{t}^w$. Update $\hat{\mathbf{a}}$ with $\hat{\mathbf{a}}^w$.
9. As long as a suitable criterion of convergence is not satisfied, continue from 2.

Next, the various points of the algorithm will be discussed in more detail. A large number of alternatives exist, and in any particular application other choices may be better than the suggestions below.

For the initial values of **a** and **t** (step 1 above), the simplest choice is to start with an empty spike list. Another possibility is some matched filter estimate. This would be computationally convenient because the essential information for the matched filter is c_{hz}.

Window selection (step 3 above) consists of two parts; position and size. A reasonable choice of positions is to cycle sequentially through all currently existing spikes, and for each spike try transitions with a window centered on that spike. The window size is important for the efficiency of the algorithm. Large windows will make each computation of l^w costly. Too small ones will dramatically increase the number of iterations before convergence. In either case the algorithm is slowed down. An additional point is that very small windows may reduce the ability of the algorithm to escape from local maxima. Simulation tests indicate that a window size approximately equal to the length of the wavelet is a good choice. This is based on transitions involving only one spike. When several spikes are involved, the window size should be increased correspondingly.

The number of new candidates considered in each window (step 4 above) can dramatically change the behavior of the algorithm. One possibility is to determine the candidates from a limited number of possible transitions. This will give a fast, but quite sub-optimal algorithm. The other extreme is a very rich transition set, which will make the algorithm slow, but nearly optimal. The possible choices include the following variants:

1. Consider only transitions involving one spike at a time. For example: delete the spike, insert a spike, move the spike one sample to the left or one sample to the right. If the wavelet has a well defined wavelength, the list may be completed with a long left and right move, using a distance of approximately half of this wavelength.

2. In addition to the transitions above, try also a number of transitions involving two spikes, e.g. move both one sample to the right, move the two spikes together and so on.

3. The brute force method: Try all possible combinations of spikes in the window, maybe excluding combinations with more spikes than a predetermined maximum. For example, if the window size is 20 and the maximum number of spikes in the window is 4, the number of possible combinations are $\binom{20}{4} = 4845$. Due to the low cost of computing l^w, repeated testing of this number of possibilities would still be manageable. This is in contrast to the global-brute-force method which consists of testing all 2^N possibilities. Based on simulation tests, it is conjectured that in many cases there exists a computationally feasible version of the windowed-brute-force method which is highly unlikely to be trapped in a local maximum. Such a procedure would thus be practically equivalent to the (unattainable) global-brute-force method.

It may be noted that even transition set 1 above is more general than the one used in the SMLR algorithm [6]. For example the left move, will require two iterations in the SMLR, one deletion and one insertion. Even though the left move increases the posterior probability, it may very well be that neither the insertion nor deletion do, and the SMLR will be stuck in a local maximum. The effects of this are pointed out by Chi and Mendel [14]. They also proposed a modification of the SMLR that can perform the left and right move (at the expense of some additional complexity), but it is not obvious how to generalize to even more complicated transitions.

In the remainder of this section a couple of implementation suggestions will be given. These are not critical, but may improve the speed of the algorithm further.

1. It is not necessary to try the most complicated transitions when the algorithm is far from convergence. It may be better to start with the simplest transitions involving one spike, and to try the more advanced transitions only when the simple ones fail to produce any more changes.

2. It may be a good idea to keep track of convergence in the different areas of a long data record. If there only exist a few problem spots where the algorithm still has not converged, it is a waste of time to try repeatedly the same transitions in areas where the configuration is unchanged.

3. The major computational burden of the algorithm is to compute $\hat{\mathbf{a}} = \mathbf{S}^{-1}\mathbf{v}$. Since $\mathbf{S}$ is positive definite symmetric, a natural choice is to compute the Cholesky decomposition of $\mathbf{S}$ and "back substitute" with $\mathbf{v}$. This requires a computational effort of approximately $(M^w)^3/6 + (M^w)^2$ multiplications and additions and M^w square roots [15]. However, with a sensible choice of window size, one may find that in a majority of cases the dimensions of the matrices are 3 or less. For such small matrices the overhead in the general Cholesky decomposition routine is considerable. It may be found far more efficient to handle each of the low order cases separately, using analytical expressions which directly give the components of $\hat{\mathbf{a}}$ in terms of the components of $\mathbf{S}$ and $\mathbf{v}$. A further simplification is obtained with appropriate normalization of the wavelet, which makes the diagonal elements of $\mathbf{S}$ equal to one. (Together with the symmetry property this makes the number of variable elements in $\mathbf{S}$ equal to only 0,1 and 3 for respectively 1,2, and 3 spikes.)

III. Selecting Parameter Values

The inverse signal-to-noise ratio δ will be treated first. Then the discussion will be specialized to the Bernoulli case, and selection of the spike-penalty parameter θ will be considered in detail. Finally some comments on the general case will be given.

A. The Inverse Signal-to-Noise Ratio δ

The inverse signal-to-noise ratio δ can determined from estimates or a priori knowledge about σ_e^2 and σ_a^2. If such information is not available, an alternative is the (non-informative) choice $\delta = 0$. Simulation tests (including the example in the next section) indicate that little is lost by this simplification. An exception is when the wavelet is distinctly non-spiky. In that case the algorithm may come up with two or more closely spaced spikes with unreasonably large amplitudes. Such spikes will invariably have opposite signs, and their effects will almost completely cancel. A

small value of δ removes the problem. A too large value will give the deconvolution output the characteristic of a simple matched filter: Due to the failure of taking the interaction between the estimated spikes properly into account, each real spike will be estimated by many closely spaced spikes with equal signs.

B. The Spike-Penalty Factor θ

In the Bernoulli case the only parameter left to determine is θ. This parameter clearly determines the number of spikes produced by the algorithm. Small values increase the risk of false detections, and large values increase the risk of missing true spikes. The optimal value will thus depend on the relative importance given to each of these two sources of error. Since θ is defined as a function of σ_e^2, σ_a^2, and λ, estimates for these quantities could be used to determine a value for θ. As argued in Appendix A, this is usually not a good idea.

A practical approach is to regard θ as a filter-tuning parameter which is adjusted to obtain the best visual deconvolution result. An important advantage is that detailed knowledge of the statistical properties of the data (parameter values, fit to model, etc.) is not necessary. If such knowledge really is available, a "training" approach may be a better alternative. A large synthetic data set similar to the real data, could be generated. Then, θ could be selected to optimize average performance relative to a realistic loss function. An example is found in [16]. The advantage of this approach is that the influence of factors such as estimation of the wavelet can easily be incorporated.

Within a reasonable range the performance of the algorithm is not critically dependent on the optimal choice of θ (see [16]). To open for more specific guidelines, two error probabilities will be defined. For this purpose, suppose the algorithm is trying to decide whether to insert a spike at a given position in an otherwise empty window. Suppose further that possible spikes outside the window which are "overlapping" with the spike under consideration (distance $2D$ or less), have already been correctly estimated. Finally, assume for simplicity that δ has been chosen equal to zero. Define the probability of false detection, P^{false}, as the probability that the algorithm accepts the spike, given that the window contains no real spikes. Conversely, define the probability of missed detection, P^{miss}, as the probability that the algorithm rejects the spike, given that the window contains exactly one spike at the given location. The defined probabilities are given by

$$P^{\text{false}} = 1 - F_{\chi_1^2}\left(\frac{\theta}{\sigma_e^2}\right), \tag{27}$$

and

$$P^{\text{miss}} = F_{\chi_1^2}\left(\frac{\theta}{\sigma_e^2}\frac{1}{1+d^2}\right), \tag{28}$$

where $F_{\chi_1^2}$ denotes the cumulative distribution function of a chi-square variable with one degree of freedom. The quantity d is defined by

$$d^2 = \frac{\sigma_a^2}{\sigma_e^2}\sum_k h^2(k) \tag{29}$$

and corresponds to the detection index from communication theory [3]. Derivations are given in Appendix C.

Now, note that θ must be chosen such as to make P^{false} small, otherwise the estimate will be filled with false detections. This suggests a value for θ on the form $c\sigma_e^2$, where c is a fractile high up in the tail of the chi-square distribution with one degree of freedom. For the simulation example in the next section, $c = 15$ was used, corresponding to $P^{\text{false}} \approx 0.0001$. For the particular noise level used P^{miss} was 0.12. Note that θ proportional to σ_e^2 was also proposed by the MAP criterion, but with a different constant of proportionality.

The simple analysis above is most relevant in the case of "non-overlapping" spikes. Interference between the spikes will usually make the error rates worse than indicated. For example, the algorithm might typically substitute three real spikes with two estimated ones, none of which match the exact position of any of the real spikes. But this does not imply that the proposed form of θ is inappropriate. Indeed, simulation tests shows that $\theta = 15\sigma_e^2$ will at least be a good starting point for a wide range of wavelet shapes, spike densities, and noise levels. The conclusion is, however, dependent on the correctness of the assumed model. In practice, slight misspecification of the wavelet and other modeling errors may be taken into account by choosing a higher value for c. The value of σ_e^2 could be estimated from the data. (For example by assuming a reasonable smooth wavelet and considering the spectral energy of the data at high frequencies, where the spectrum of the wavelet is negligible.)

A different approach to determination of θ can be based on the following fact, which may easily be established: If all spikes have been considered for deletion after the last major change performed by the algorithm, the final amplitudes will all have greater absolute value $[\theta/\sum_k h^2(k)]^{1/2}$. Thus, θ implicitly determines the smallest spikes which may be detected. This is similar to the findings of other authors [3], [9].

C. The General Case

In the non-Bernoulli case, there are three quantities left to determine: The prior density $p(\mathbf{t})$ and the parameters θ_1 and θ_2. Note from (25) and (26) that θ_1 and θ_2 weights the relative influence of three factors: The fit to the data, the prior knowledge about $\mathbf{t}$, and the number of produced spikes. As in the Bernoulli case, the "true" parameter values can not be expected to give optimal results (Appendix A). A reasonable choice is to model $p(\mathbf{t})$ as realistic as possible, but to select θ_1 and θ_2 without regard to their definition. Selection may be based on trial and error, or the training approach. In the latter case the some gradient search could be used. Due to the rapid execution of the algorithm this should be feasible even for a quite large

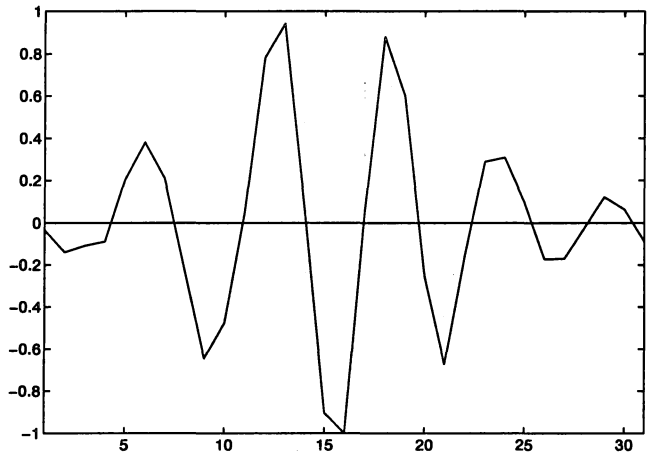

Fig. 1. Wavelet.

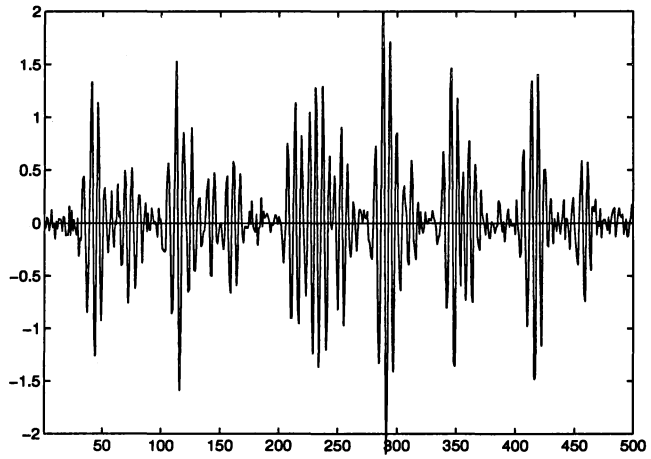

Fig. 2. Noisy data.

training set. A sensible starting point is to choose both parameters on the form of a constant multiplied by σ_e^2.

Another solution is to model $g(t)$ directly, without regard to its definition in terms of $p(t)$. The viewpoint should be to construct a function which "penalizes" unwanted configurations.

A concrete specification of alternatives to the Bernoulli/geometric form of $p(t)$ will be application dependent, and only one simple modification shall be mentioned here. A characteristic of the geometric distribution is that it gives relatively high probability to very closely spaced spikes. Estimates containing such configurations may be considered either unrealistic or unwanted in many situations. This is easily remedied by adding a term to $g(t)$ which penalizes all spike pairs with short inter-distance. A particularly simple solution is to set $p(t)$ equal to zero if any inter-spike distance is smaller than a given threshold. The corresponding algorithm is realized by using the Bernoulli case formulas, but simply not considering transitions that would move any spike too close to another one. Prohibiting very closely spaced spikes will also alleviate the problem of unreasonably large amplitude estimates, thus making it even more attractive to use $\delta = 0$.

IV. Computer Simulation

A record of 500 samples was generated from the Bernoulli-Gaussian distribution with spike density $\lambda = 0.05$ and amplitude standard deviation $\sigma_a = 1$. The Bernoulli-Gaussian spike train was convolved with the wavelet shown in Fig. 1, and Gaussian white noise with standard deviation $\sigma_e = 0.1$ was added. (The wavelet was extracted from an actual ultrasound image and scaled to have its maximum amplitude equal to 1.) The generated data and the result of deconvolution are given in Figs. 2 and 3.

All large spikes were located correctly in this example while some small ones were missed or located incorrectly. The tested version of the algorithm was based on transition set 1 in Section II-E. The window size was 21 samples. Parameter values were $\delta = 0$ and $\theta = 15\sigma_e^2$. Implementation suggestions 2 and 3 at the end of section II-E were followed.

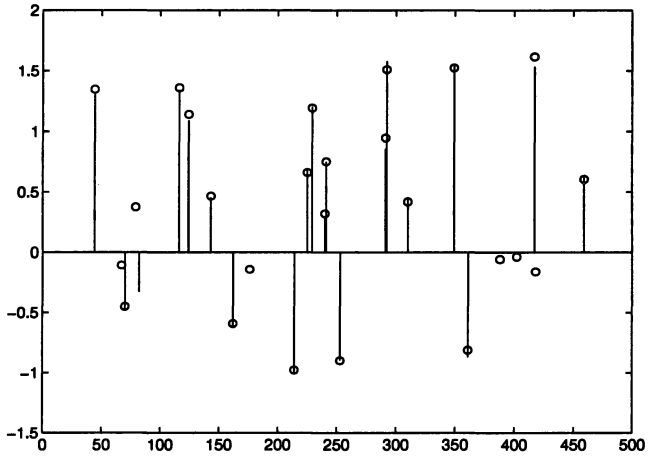

Fig. 3. Deconvolution output. Bars denote estimates and circles denote true values.

Convergence required 9 passes through the entire data record. The execution time was 35 msec. on a 60 MHz Pentium PC. The part of the time spent on calculating correlation functions (13) and (14) was 13 msec. (Direct calculation was used. Incidentally this required almost exactly the same time as frequency domain calculation.) The work of computing c_{hh} is negligible, and computing c_{hz} corresponds to a simple matched filter. Thus, the execution time of the algorithm can be given relative to a matched filter. For the given example the ratio was 2.7. This compares favorably with other algorithms proposed for deconvolution of sparse spike trains, see [16].

Due to the iterative nature of the algorithm, the execution time will depend on the particular realization of the data record. In general the execution time will be an increasing function of the spike density, since more spikes will require more iterations and the average number of spikes in a fixed size window will increase.

V. Parallelization and Recursive Processing

Two forms of parallelization are possible. First, note that it is valid to perform updating in several windows simultaneously. The only condition is that the windows must be

separated by a distance corresponding to the length of the wavelet. This resembles the parallelization property often noted for Monte Carlo methods such as Gibbs sampling (see [17]).

The second possibility has no obvious analogue in the Monte Carlo methods. Within each window, any number of new configurations can be examined simultaneously, before the best is finally chosen. If the number of configurations to be tested within each window is large, this could mean a vast improvement in speed.

In some applications it is desirable to process the data on-line before the entire data record is collected [18]. A modification to allow such *recursive* processing is: Start with a block of data, $n = 1, 2, \ldots, B$. Run the algorithm on the data contained in the block to obtain spike estimates. When the next sample arrives the algorithm is rerun on the data $n = 1, 2, \ldots, B, B+1$. But this time all windows are constrained to the interval $n = 2, \ldots, B+1$. Thus, an eventual spike estimate in position 1 is fixed and will not be changed any more. In general, when the k'th sample arrives, a new estimate for the block $k-B+1, \ldots, k$ is computed. The spikes in the interval $1, \ldots, k-B$ are fixed, but observe that those in the interval $k-B-2D+1, \ldots, k-B$ will still influence the new estimate.

When rerunning the algorithm on a new block of data, simplifications occur. The only necessary recomputation of correlation functions is one additional element of c_{hz}, cf. (14). Furthermore, since a good initial estimate exists from the previous restoration, convergence is likely to be very fast.

For a small block size B, a particularly simple strategy is to use only one window containing the entire block. Combined with parallel examination of the possible transitions, very fast execution would result.

VI. Conclusion and Summary

A promising new algorithm for deconvolution of sparse spike trains has been presented. A sub-optimal iterative search is used to maximize a joint MAP criterion. Overfitting problems [3], [6] are avoided by appropriate choice of parameter values. The level of sub-optimality is determined by the number of transitions considered at each iteration. Since the derived formulas allow arbitrarily complicated transitions, any tradeoff between execution speed and quality of reconstruction can be made. The given simulation example shows that good reconstructions can be obtained while still retaining fast execution. A comprehensive comparison with a number of existing algorithms will be reported in [16]. Since the algorithm allows extensive parallelization and recursive processing, it may also be interesting for real-time applications.

The window strategy is central to the efficiency of the algorithm. It is based on the fact that the dependencies in the posterior distribution tends to die out over large spatial distances. Since local maximization is iterated, it does not in a certain sense introduce additional sub-optimality.

Implementation of the basic version of the algorithm is simple. In particular, evaluation of all necessary formulas is straightforward. However, considerable flexibility exists in how the iterative search is performed. To optimize execution speed for a desired level of optimality, a rather sophisticated implementation may be necessary.

In the present paper the wavelet has been assumed known, but the fast execution of the algorithm suggests its use also for "blind deconvolution". Combined deconvolution and estimation of the wavelet could be performed by a "block component method" similar to those used by Mendel [1]. Due to the simple structure of the derived formulas, a number of other generalizations is also possible. Research in this area is in progress. Results will be reported in later publications.

Acknowledgments

This work was supported by grants from the Research Council of Norway. I am grateful to Prof. Erik Bølviken, University of Oslo and Prof. Torfinn Taxt, University of Bergen. Their comments have greatly improved the presentation of this material.

Appendices

A. The MAP Estimator; Deficiencies and Remedies

It is known that uncritical maximization of joint MAP criteria will not give satisfactory estimates for sparse spike trains. The problem has been compared to model-order selection [6], and overfitting has been reported [3]. It has also been found that Akaike's criterion for model selection does not present a satisfactory solution [3].

In the present context, the problems are most easily demonstrated in the Bernoulli case. For example, if $\lambda^2/(1-\lambda)^2 > 2\pi\sigma_a^2$, then θ will be negative. And with a negative θ, it can be seen that the estimate will contain spikes at all possible positions. As another example, consider increasing σ_a by a given factor, and decreasing h by the same factor. In this case the deconvolution problem is not really altered, see [19]. As expected, $\mathbf{v}'\hat{\mathbf{a}}$ is unchanged, cf. (10). But since θ is a function of σ_a^2, its value will indeed change. Actually, it can be seen that maximization of the unmodified MAP-criterion involves comparison of quantities with different dimension. As a result, the corresponding estimate will not be scale invariant.

In a decision theoretic framework the problems of the MAP estimator may be seen as a consequence of its implicit loss function. It is known that the MAP estimator corresponds to a loss function which assigns loss 0 to a completely correct configuration, and loss 1 to all other configurations [17]. Assigning the same loss to an estimate that has only missed a single small spike as to an estimate which is nowhere near the true solution, is of course highly unrealistic in most practical situations. The most satisfying solution would be to redo the whole calculation with a realistic loss function, but this would hardly give simple computational formulas.

The approach adopted in this article is to use the functional form suggested by the MAP estimator, but to allow

other parameter values than those which have generated the data. From (26) it is clear that overfitting can be avoided by appropriate choice of θ in the Bernoulli case or θ_1 and θ_2 in the general case. It can also be seen that selecting these parameters proportional to σ_e^2, as suggested in section III, will make the estimator scale invariant.

Another common solution [1], [6], [12], [13] is to do the deconvolution by a two stage procedure. First, detection of the spike-positions through maximizing of their *marginal* posterior distribution. Then estimation of the amplitudes conditional to the detected spike-positions.

Since the number of spikes can be controlled in the joint MAP approach by appropriate choice of parameter values, it is interesting to examine which remaining differences exist. Within the framework of this paper a marginal MAP estimator for $\mathbf{t}$ is easily derived. Except for constants and terms which can be absorbed in θ (assuming the Bernoulli case), the corresponding criterion differs from the joint criterion (10) only by a term of the form $-\sigma_e^2 \ln|\mathbf{S}|$. As long as δ is not chosen very large, $\mathbf{S}$ is approximately equal to $\mathbf{H}'\mathbf{H}$. Thus, relative to the joint criterion, the marginal criterion favors configurations where the columns of $\mathbf{H}$ are close to linearly dependent, since this makes the determinant small. (One such typical situation is when two or more spikes are very close together.) The implication is that the configurations favored by the marginal criterion are exactly those which make the amplitude estimation difficult. This is also intuitively reasonable; the marginal criterion favors the values of $\mathbf{t}$ which correspond to a wide range of probable values for $\mathbf{a}$. If, in a given practical problem, valuable information is also contained in the amplitudes, the desirability of this property seems doubtful.

B. Proof of a Convergence Property

Theorem:

1. The algorithm is convergent.
2. Let $(\hat{\mathbf{a}}, \hat{\mathbf{t}})$ be an estimate produced by the algorithm. Suppose that after reaching the final estimate, each of the estimated spikes has been included in at least one window for which the algorithm could find no improvements. Then $\hat{\mathbf{a}}$ globally maximizes $p(\mathbf{a}, \hat{\mathbf{t}}|\mathbf{z})$ with respect to $\mathbf{a}$.

Proof:

1. Since each accepted update in the algorithm increases $p(\mathbf{a}^w, \mathbf{t}^w|\mathbf{z}, \mathbf{a}^{\overline{w}}, \mathbf{t}^{\overline{w}})$, the factorization (19) makes it clear that the algorithm will increase the posterior density at each step. Since this density is a bounded function, the convergence follows from the bounded monotone sequence property of calculus.
2. From (7) it is clear that the log-posterior density is a concave function of $\mathbf{a}$ which has only a global maximum. Suppose that the global maximum for $\mathbf{a}$ has not been reached. Since the function to be maximized is differentiable, there exist an i and a small quantity d such that when d is added to $\hat{a}_i$ the function value is increased. This can be written as

$$p(\hat{\mathbf{a}} + \mathbf{d}, \hat{\mathbf{t}}|\mathbf{z}) > p(\hat{\mathbf{a}}, \hat{\mathbf{t}}|\mathbf{z}), \tag{30}$$

where $\mathbf{d}$ is a vector which has its i'th component equal to d, and zeros elsewhere. Now use the assumption of the theorem and denote by w a window, containing the i'th spike, for which the algorithm could not improve the estimate for $\mathbf{a}$. Using the factorization (19) on both sides of (30) and canceling the two equal factors gives

$$p(\hat{\mathbf{a}}^w + \mathbf{d}^w, \hat{\mathbf{t}}^w|\mathbf{z}, \hat{\mathbf{a}}^{\overline{w}}, \hat{\mathbf{t}}^{\overline{w}}) > p(\hat{\mathbf{a}}^w, \hat{\mathbf{t}}^w|\mathbf{z}, \hat{\mathbf{a}}^{\overline{w}}, \hat{\mathbf{t}}^{\overline{w}}),$$

where $\mathbf{d}^w$ is the part of $\mathbf{d}$ corresponding to the spikes inside the window. But this is a contradiction since the algorithm has already determined $\hat{\mathbf{a}}^w$ to maximize this conditional density with respect to $\mathbf{a}^w$.

C. Derivation of Error Probabilities

The distributional results reached here will be somewhat more general than necessary to deduce the two error probabilities defined in section III-B. All probabilistic statements are conditional on $\mathbf{t}$ and $\mathbf{a}^{\overline{w}}$.

Note from (24), (25) and the Bernoulli case of (26) that the maximization criterion may be written as

$$l^w = ||(\mathbf{S}^w)^{-1/2}\mathbf{v}^w||^2 - \theta M^w. \tag{31}$$

Consider the probability of false detection first. Assume that the spikes outside the window have been correctly estimated and that there are no true spikes inside the window. From (17) it follows that $\mathbf{z}^w = \mathbf{e}$, which implies that $\mathbf{v}^w = (\mathbf{H}^w)'\mathbf{z}^w$ has covariance matrix $\sigma_e^2\mathbf{S}^w$. (Recall that it was assumed that δ had been chosen equal to zero, giving $\mathbf{S}^w = (\mathbf{H}^w)'\mathbf{H}^w$.) The variable $(\mathbf{S}^w)^{-1/2}\mathbf{v}^w$ will thus have a covariance matrix of $\sigma_e^2\mathbf{I}_{M^w}$. In addition, it is easily seen to be Gaussian with zero mean. Combining this with (31) shows that l^w has the distribution of $\sigma_e^2\chi^2_{M^w} - \theta M^w$, where χ^2_m denotes a chi-squared variable with m degrees of freedom. The error probability (27) follows by setting $M^w = 1$ and noting that the spike will be inserted if $l^w > 0$. (An empty window yields $l^w = 0$, cf. (25), (26), and the following comment.)

Consider now the probability of missed detection. Assumptions are as above, except that there is one or more true spikes inside the window, and that $\mathbf{t}^w$ contains their correct locations. In this case $\mathbf{z}^w = \mathbf{H}^w\mathbf{a}^w + \mathbf{e}$, and the covariance matrix of $\mathbf{v}^w$ is $\sigma_a^2(\mathbf{S}^w)^2 + \sigma_e^2\mathbf{S}^w$. The resulting covariance matrix of $(\mathbf{S}^w)^{-1/2}\mathbf{v}^w$ is $\sigma_a^2\mathbf{S}^w + \sigma_e^2\mathbf{I}_{M^w}$. In particular for $M^w = 1$, the distribution of l^w is given by $\left(\sigma_a^2\sum_k h^2(k) + \sigma_e^2\right)\chi_1^2 - \theta$, which gives the error probability (28).

Note finally that the assumption of correct estimation outside the window can be relaxed. Since $\mathbf{v}^w$ only depends on spikes with a distance less than $2D+1$ from the spike under consideration, only such spikes need to be correctly estimated.

References

[1] J. M. Mendel, *Optimal Seismic Deconvolution: An Estimation-Based Approach.* New York: Academic, 1983.

[2] M. S. O'Brien, A. N. Sinclair, and S. M. Kramer, "Recovery of a sparse spike time series by L_1 norm deconvolution" *IEEE Trans. Signal Processing*, vol. 42, no. 12, pp. 3353-3365, dec. 1994.

[3] H. Kwakernaak, "Estimation of pulse heights and arrival times," *Automatica*, vol. 16, pp. 367-377, 1980.

[4] B. S. Atal and J. R. Remde, "A new model of LPC excitation for producing natural sounding speech at low bit rates," *Proc. IEEE, Int. Conf. Acoust., Speech, Signal Processing*, vol. 3-5, pp. 614-617, 1982.

[5] I. Barrodale, C. A. Zala, and N. R. Chapman, "Comparison of the l_1 and l_2 norms applied to one-at-a-time spike extraction from seismic traces," *Geophysics*, vol. 49, no. 11, pp. 2048-2052, 1982.

[6] J. J. Kormylo and J. M. Mendel, "Maximum likelihood detection and estimation of Bernoulli-Gaussian processes," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 3, pp. 482-488, May 1982.

[7] C.-Y. Chi and J. M. Mendel, "Viterbi algorithm detector for Bernoulli-Gaussian process," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 3, pp. 511-519, June 1985.

[8] R. Yarlagadda, J. B. Bednar, and T. L. Watt, "Fast algorithms for l_p deconvolution" *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 1, pp. 174-182, Feb. 1985.

[9] M. Lavielle, "Bayesian deconvolution of Bernoulli-Gaussian processes," *Signal Processing*, Elsevier, vol. 33, no. 1, pp. 67-79, July 1993.

[10] M. Cookey, H. J. Trussell, and I. J. Won, "Seismic deconvolution by multipulse methods" *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-38, no. 1, pp. 156-160, Jan. 1990.

[11] G. B. Giannakis, J. M. Mendel, and X. Zhao, "A fast prediction-error detector for estimating sparse-spike sequences" *IEEE Trans. Geosci. Remote Sensing*, vol. GE-27, no. 3, pp. 344-351, May 1989.

[12] Y. Goussard, G. Demoment, and J. Idier, "A new algorithm for iterative deconvolution of sparse spike trains," *Proc. IEEE, Int. Conf. Acoust., Speech, Signal Processing*, pp. 1547-1550, 1990.

[13] C.-Y. Chi, "A fast maximum likelihood estimation and detection algorithm for Bernoulli-Gaussian processes" *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, no. 11, pp. 1636-1639, Nov. 1987.

[14] C.-Y. Chi, J. M. Mendel, "Improved maximum-likelihood detection and estimation of Bernoulli-Gaussian processes," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 429-435, March 1984.

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C, 2nd ed.* New York: Cambridge University Press, 1992.

[16] K. F. Kaaresen, "Comparison of deconvolution algorithms for sparse spike trains," to be published.

[17] J. Besag, "Towards Bayesian image analysis," *Journ. Applied Statistics*, vol. 16, no. 3, pp. 395-407, 1989.

[18] Y. Goussard and G. Demoment, "Recursive deconvolution of Bernoulli-Gaussian processes using a MA representation" *IEEE Trans. Geosci. Remote Sensing*, vol. GE-27, no. 4, pp. 384-394, July 1989.

[19] J. Goutsias and J. M. Mendel, "Maximum likelihood deconvolution: An optimization theory perspective," *Geophysics*, vol. 51, no 6, pp. 1206-1220, June 1986.

Comparison of Deconvolution Algorithms for Sparse Spike Trains

Kjetil F. Kaaresen

Abstract— **In seismic and ultrasonic applications, it is frequently desirable to restore a sparse reflectivity sequence which has been distorted by a time-invariant linear system and contaminated by additive noise. A new algorithm for this deconvolution problem, iterated window maximization (IWM), was recently proposed by Kaaresen. The present paper evaluates IWM against a number of well-established alternatives. Restoration quality is quantified by some loss functions, and average performance is studied through simulation. For all cases examined, IWM gave better average restoration and significantly faster execution than the established techniques. Some conclusions on the relative performance of the other algorithms are also given.**

Keywords— **Deconvolution, sparse spike train, comparative study.**

I. Introduction

DECONVOLUTION of sparse spike trains has been subject to extensive research. Important applications are found in the areas of reflection seismology [1], [2] and ultrasonic imaging [3], [4], [5], but the problem is also of interest elsewhere [6], [7]. A common starting point is the one-dimensional convolutional model

$$\mathbf{z} = \mathbf{h} * \mathbf{x} + \mathbf{e}. \tag{1}$$

Here $\mathbf{z}$ is an observed data vector, $\mathbf{h}$ is a sampled wavelet (impulse response), $\mathbf{x}$ is an unknown sparse spike train, and $\mathbf{e}$ is additive noise. (In the echographic applications, $\mathbf{x}$ will typically represent the reflectivity of a layered medium, and can to a good approximation be assumed to vanish everywhere except at layer boundaries.)

The objective of deconvolution is to estimate $\mathbf{x}$ based on knowledge of $\mathbf{z}$ and $\mathbf{h}$. In practice, the problem is often made difficult by a narrow-band $\mathbf{h}$. This can partially be compensated for by using the a priori information that $\mathbf{x}$ is sparse. The problem is then reduced to the detection of a limited number of non-zero components (spikes) and estimation of their values (amplitudes). Since standard linear deconvolution techniques can not take advantage of such a priori knowledge, a number of alternatives have been proposed.

The purpose of this paper is to compare a recent contribution, iterated window maximization (IWM) [8], to several well-established alternatives. The methods selected for comparison are: Single Most Likely Replacement (SMLR) [9], Iterated Conditional Modes (ICM) [10], Simulated Annealing (SA) [11], [12], and L_1 deconvolution with the Simplex algorithm [13]. This list is far from exhaustive, but represents a number of quite different solutions that have all been shown to perform well elsewhere [3], [9], [14], [15]. A short introduction to each algorithm is given in Section II.

Comparisons found in the literature [9], [14], [16], [17], [18], [19] have mainly been confined to visual inspection of single examples. Due to the non-linear nature of the tested algorithms, their successful resolution of severely overlapping wavelets will vary considerably with the relative positions of the different spikes. A small data set may easily fail to give a representative picture. The present approach is to define some measures of good recovery (loss functions), and to compare average performance on large simulated data sets. It may be argued that the choice of loss functions is somewhat arbitrary, but it is found that several different loss functions yield similar conclusions.

All test data are based on the conventional *Bernoulli-Gaussian* [1], [9] distribution for $\mathbf{x}$. The noise, $\mathbf{e}$, is taken as Gaussian and white. The algorithms are compared in low and high noise conditions. In addition, the unrealistic assumption of a perfectly known wavelet is removed by adding white noise to the wavelet prior to deconvolution. A detailed description of the test procedure is given in Section III

The results obtained are presented in Section IV and discussed in Section V. Section V also contains some results of a more informal character concerning the effect of the wavelet shape and of different sources of wavelet degradation.

In practical applications the wavelet (and other statistical parameters) may also be unknown. Such a *blind deconvolution* problem is often solved by a block component method [1]. In this case the algorithms tested here can be viewed as one part of such a larger system.

II. The Algorithms

The aim of this section is to pinpoint differences and similarities between the tested algorithms, and to state some implementation choices. Derivations and complete descriptions are found in the cited references.

The first 4 algorithms below are maximum a posteriori (MAP) methods. In these, the sparsity assumption is quantified by a statistical model. Then, the sparse spike train is estimated by the realization that has the highest probability (density) given the observed data. Unfortunately, this *posterior* probability is typically a complicated function with numerous local maxima. Global maximization is a difficult task, and each algorithm relies on a different iterative search technique. All are sub-optimal to some extent and may succeed only in locating a local maximum of the posterior.

K. F. Kaaresen is with the Department of Mathematics, University of Oslo, P.B. 1053 Blindern, N-0316 Oslo, Norway. E-mail: kjetilka@math.uio.no.

A. Iterated Window Maximization (IWM)

The IWM algorithm was recently proposed by Kaaresen [8]. From expressions given in [8] it can be seen that the MAP maximization performed is equivalent to minimization of

$$||\hat{\mathbf{e}}||^2 + \delta||\hat{\mathbf{x}}||^2 + \theta M, \tag{2}$$

where $\hat{\mathbf{x}}$ is the estimate of $\mathbf{x}$, $\hat{\mathbf{e}} = \mathbf{z} - h * \hat{\mathbf{x}}$, $||\cdot||$ denotes the usual vector norm, and M is the number of non-zero elements in $\hat{\mathbf{x}}$. The two "tuning" parameters θ and δ governs sparsity and size characteristics of the produced solution. Increasing θ increases the "penalty" given for each extra spike and will thus tend to give more sparse solutions. The importance of δ is generally less. For distinctly "spiky" wavelets it may simply be set equal to zero. In other cases it must be given a small value to avoid unreasonable large spike estimates, cf. [8].

The iterative search procedure used in the IWM is briefly explained as follows: In each iteration all updates are confined to a small window. First, all the amplitudes within the window are updated to the mode of their conditional probability density given the current spike positions, the amplitudes outside the window, and the data. (Computing this conditional mode amounts to a linear least squares fit with dimension equal to the number of spikes in the window.) Then, a number of new configurations are examined. Each is obtained by changing one or a few spike positions and refitting all the amplitudes within the window in the same manner as above. If a configuration is found which increases the posterior probability, it replaces the original one. When this happens, or when all changes in a predefined set are tried unsuccessfully, a new window is chosen and the procedure is repeated. The iteration stops when neither spike positions nor amplitudes change any more.

The window positions are scanned systematically through the entire data record. It is shown in [8] that this guarantees that the final amplitudes are globally optimal given the final spike positions, but the final spike positions may still be sub-optimal. The level of sub-optimality is determined by the number of new configurations examined within each window.

Two version were tested: IWM 1 corresponds to transition set 1 proposed in [8] where only one spike location is changed in each iteration. The exact transition set used was: delete a spike, insert a spike, move a spike one sample to the left or right, move a spike 3 samples to the left or right. IWM 2 corresponds to transition set 2 where also a number of transitions affecting two spike locations simultaneously were examined. Only spike pairs with interdistance of no more than 10 samples were considered for simultaneous update. For these pairs, replacement with only one spike was tried in all positions that had a distance of 5 samples or less from any one of the two original spikes. Furthermore, all possible replacements with two spikes were considered, the only condition being that the distance from the original spikes to their respective replacement should 5 samples or less.

For both algorithms the window size was chosen to include all spikes with a distance of 10 samples or less from the ones whose positions were changed. Three implementation proposals given in [8] were incorporated in both versions.

B. Single Most Likely Replacement (SMLR)

The SMLR algorithm, introduced by Kormylo and Mendel [9], maximizes a slightly different MAP criterion than the IWM.[1] The strategy is to maximize the *marginal* distribution for the spike positions first, and then estimate the amplitudes conditional to the detected spike positions. This is not equivalent to maximization of the *joint* distribution of positions and amplitudes such as the IWM does. (The SMLR technique can also be used to maximize the joint distribution, but the marginal approach seems more common and has been chosen here, see [2], [8], [9], [20] for discussion.)

The iterative search of the SMLR differs from the IWM in that only changes corresponding to insertion or deletion of a single spike are considered. This transition set is more restrictive than that of both IWM versions, and will reduce the ability of the SMLR to escape from local maxima [8], [16]. A further difference is that the SMLR is less "greedy" than the IWM. In each iteration it considers all possible insertions and deletions before choosing the one giving the largest increase in the posterior probability. (The clue of the SMLR algorithm is to be able to do all these comparisons very efficiently.) Finally, the SMLR does not operate with a globally sub-optimal configuration for the amplitudes at intermediate stages such as the IWM does.

Since the paper by Kormylo and Mendel, a number of variations of this algorithm have been proposed. (See [2], [20] for surveys.) The variant implemented here is due to Goussard et. al. [21].

C. Iterated Conditional Modes (ICM)

Lavielle [14] has considered a number of well-known Bayesian algorithms in the context of deconvolution of sparse spike trains. The simplest is the ICM. This algorithm aims at maximizing the same posterior distribution as the IWM. To do so, it repeatedly maximizes the conditional distribution of each component of $\mathbf{x}$, given all other components until convergence is achieved. The ICM is in fact a special case of the IWM: Limiting the IWM to only insertions and deletions and setting the window size to one sample gives the ICM. Realizing that the ICM corresponds to such a restricted special case may serve as an explanation of why it gets so easily stuck in a local maximum.

Lavielle parameterizes the ICM by a parameter $x_{\min}$ which determines the smallest (non-null) spike which may be produced. The following relationship [8] exists between $x_{\min}$ and the spike penalty parameter, θ, of the IWM:

$$x_{\min}^2 = \theta \Big/ \sum_n h_n^2. \tag{3}$$

[1] The SMLR is commonly referred to as a maximum likelihood method, but the term MAP is preferred here since the estimated quantities are considered as random.

Lavielle also proposed a modification of the ICM algorithm which has not been considered here.

D. Simulated Annealing (SA)

A stochastic generalization of the ICM is Gibbs sampling with SA. The difference is that instead of choosing each component of $\mathbf{x}$ as the maximizer of the conditional distribution, it is randomly sampled. The sampling distribution is proportional to the conditional distribution used by the ICM raised to the power $1/T$. The "temperature", T, is a control parameter that is decreased during the iterations to make the conditional densities more spiky around their modes. The reason for introducing a stochastic element is to make it less likely for the algorithm to get stuck in a local maximum. Geman and Geman [12] proved that if the temperature is decreased sufficiently slowly, the algorithm will certainly locate the global maximum. However, their definition of "sufficiently slowly" is too slow to be practically applicable. One thus have to use a faster annealing schedule and hope that the probability of the algorithm ending up in a local maximum is small, or that if a local maximum is found, it will not be much inferior to the global one. This aim is shared by the IWM. The difference may be illustrated by a hill-climbing analogy: To reach the global maximum, the stochastic algorithm accepts to go downhill part of the time, whereas the deterministic algorithm tries to take sufficiently long strides to walk directly from top to top until the global maximum is reached. In both algorithms computational efficiency can easily be traded for increased optimality: In the stochastic algorithm by using a slower annealing schedule and in the deterministic algorithm by increasing the number of transitions considered. In contrast, even though one is willing to spend more computational energy, it is not obvious how to use this to reduce the sub-optimality of the SMLR. The present implementation of Gibbs sampling with SA is based on the ideas given by Lavielle [14].

A difficult task with SA is to determine a good annealing schedule for a given computational constraint. Some tests indicated that for the present problem it was important to use a lot of time at high temperatures to avoid the algorithm being trapped in a local maximum at an early stage. The problem also seemed more severe when the noise level increased. After a good deal of experimenting it was decided to lower the temperature linearly 10 to 0 in the low-noise case considered (Section III) and from 30 to 0 in the high-noise case. A fast and a slow annealing schedule were tested, SA 1 which completed the annealing-schedule in 1,000 passes through the entire $\mathbf{x}$ vector, and SA 2 which used 100,000 passes.

E. L_1 Norm Deconvolution (Simplex)

The objective of L_1 norm deconvolution is minimizing of

$$||\hat{\mathbf{e}}||_1 + \gamma||\hat{\mathbf{x}}||_1, \tag{4}$$

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{e}}$ are defined as for the IWM and $||\cdot||_1$ denotes the L_1 norm (sum of absolute values). The parameter γ influences both the number and size of the produced spikes. Generally, a larger value will give a sparser solution.

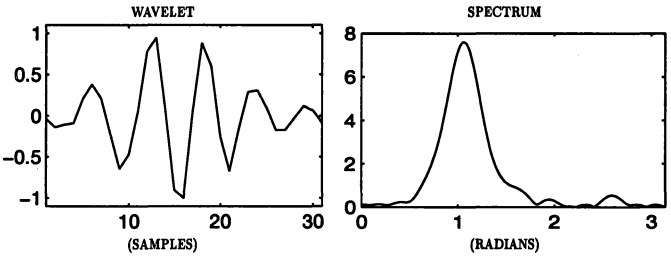

Fig. 1. The ultrasonic wavelet used for data generation and its amplitude spectrum.

The form of (4) is chosen such that the Simplex algorithm [22] can be used for minimization. The Simplex is only of the tested algorithms that is guaranteed to be optimal for its criterion. However, the L_1 criterion is quite different from the criteria employed in the other algorithms. In the other algorithms it is mainly the number of non-zero components of $\hat{\mathbf{x}}$, not its total size, that counts. Indeed, it will be seen that the L_1 criterion is quite far from optimal for a Bernoulli-Gaussian process, tending to produce far too many small spikes.

The present implementation of L_1 deconvolution is based on the exposition in [3] combined with the Simplex routine in [23].

III. Experimental Procedure

Sparse spike trains were generated from the Bernoulli-Gaussian distribution with spike density $\lambda = 0.05$ and amplitude standard deviation $\sigma_a = 1$. (The Bernoulli-Gaussian process can be defined as the product of a white Bernoulli process with success probability λ and a zero mean and white Gaussian process with standard deviation σ_a^2.) To avoid border effects, no spikes were generated closer to each end of the record than 15 samples. The Bernoulli-Gaussian spike trains were convolved with the wavelet shown in Fig. 1. This wavelet was extracted from an actual ultrasound image and standardized to have its maximum amplitude equal to 1. (The ultrasound image was produced by a Vingmed CFM sector Scanner 750.) Note the rather low frequency content which makes the deconvolution problem difficult. Finally, white zero-mean Gaussian noise was added. Two noise cases were considered: standard-deviation $\sigma_e = 0.05$ and $\sigma_e = 0.2$. These two cases will be referred to as RMS 5% and RMS 20% respectively.[2] The effect of slight misspeci-

[2] The signal-to-noise ratio, taken as the ratio of average signal power to average noise power, is a commonly used measure of noise level. It is felt that this measure is unfortunate in the present context since the value will depend on the spike density λ: A high value of λ will increase the signal-to-noise ratio whereas the deconvolution problem actually becomes more difficult. The measure reported here will simply be the noise standard deviation divided by the Root Mean Square of the largest signal value produced by an isolated spike, i.e. $\sigma_e/(\sigma_a \max_n h_n)$. Since both σ_a and $\max_n h_n$ is equal to 1 this reduces to σ_e. Another reasonable noise measure is the detection index [6] given by $||\mathbf{h}||\sigma_a/\sigma_e$. Its value is 49.9 and 12.5 for the two noise cases considered here.

fication of the wavelet was also investigated. For this purpose data records with RMS 5% were used, but in addition white noise with standard deviation 0.1 was added to the wavelet prior to deconvolution. This case will be referred to as RMS 5% + 10%.

The performance of the algorithms was quantified by four loss functions. The first three were based on an L_1 norm between the deconvolution output, $\hat{\mathbf{x}}$, and the true value, $\mathbf{x}$. In addition, an extra penalty was given for each missed detection ($\hat{x}_n = 0$, $x_n \neq 0$) and for each false detection ($\hat{x}_n \neq 0$, $x_n = 0$). The general form of these loss functions was

$$L = \sum_n |\hat{x}_n - x_n| + w^{\text{miss}} N^{\text{miss}} + w^{\text{false}} N^{\text{false}},$$

with N^{miss} counting the number of missed detections, N^{false} counting the number of false detections and the two weights w^{miss} and w^{false} determining the importance given to each of these two sources of detection error. (An even more realistic loss function might give partial credit for spikes that were estimated close to their true positions. However, due to the relatively low sampling rate of the used wavelet, the simpler form above was deemed sufficient.) The forth loss function is the common sum-of-squares loss, which was included to demonstrate that the choice of the L_1 norm is not critical for the results. The precise form of all employed loss functions was:

$$\begin{aligned} L^{\text{miss+false}} &= \textstyle\sum_n |\hat{x}_n - x_n| + N^{\text{miss}} + N^{\text{false}}, \\ L^{\text{miss}} &= \textstyle\sum_n |\hat{x}_n - x_n| + N^{\text{miss}}, \\ L^{\text{false}} &= \textstyle\sum_n |\hat{x}_n - x_n| + N^{\text{false}}, \\ L^{\text{SSQ}} &= \textstyle\sum_n |\hat{x}_n - x_n|^2. \end{aligned} \tag{5}$$

Note that that the number of missed or false detections can easily be obtained as differences of the given loss functions.

Before the algorithms could be compared, their "tuning parameters" had to be determined. It was decided to find the optimal values based on a set of training data. For each of the three noise cases above a training set was generated. The training sets consisted of 100 independent data records with 500 samples each. Each algorithm was allowed one scalar tuning parameter, which was determined to minimize the average value of $L^{\text{miss+false}}$ on the training data. (As a consequence $L^{\text{miss+false}}$ should be considered as the main yardstick of the following test.)

The parameters which were adjusted and the values determined are given in Table I. Some comments are in order: For the IWM, δ was simply set equal to zero, and only θ was optimized. For easy comparison to the other algorithms (3) was used to transform θ to a value of $x_{\min}$. The SMLR does not have a typical tuning parameter. Given the value of λ, σ_a and σ_e, the algorithm is completely determined. However, it is not obvious that the best results will be obtained with the "true" parameter values, and it was decided to use the training to determine an optimal input value of σ_e. (For RMS 5% the optimal value was found to be 20% larger than the "true" value, whereas for RMS 20% the two were approximately equal.) The SA 2 and the Simplex were too slow to allow full training. For SA 2 the $x_{\min}$ determined for IWM 2 was used. (A slight tuning difference still existed because of the simplification $\delta = 0$ used for the IWM. To achieve exact equal tuning, δ should have been chosen equal to σ_e^2/σ_a^2. However, experiments with the IWM showed minimal difference between the two alternatives.) The Simplex was simply tuned to give the best possible visual results.

TABLE I

OPTIMAL PARAMETER VALUES DETERMINED BY TRAINING THE ALGORITHMS ON A LARGE DATA SET.

	RMS=5%	RMS=20%	RMS=5%+10%
IWM 1 ($x_{\min}$)	0.06	0.28	0.27
IWM 2 ($x_{\min}$)	0.07	0.31	0.29
SMLR (σ_e)	0.06	0.20	0.25
ICM ($x_{\min}$)	0.13	0.30	0.30
SA 1 ($x_{\min}$)	0.10	0.60	0.40
SA 2 ($x_{\min}$)	0.07	0.31	0.29
SIMPLEX (γ)	2	3	3

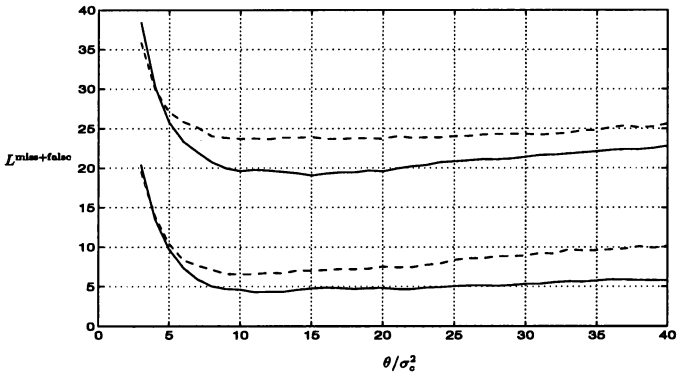

Fig. 2. Training results for the Iterated Window Maximization algorithm. Average loss as a function of tuning parameter. Dashed lines: IWM 1. Solid lines: IWM 2. The two lower lines represent low noise (RMS 5%), and the two upper lines represent high noise (RMS 20%). The parameter values corresponding to the minimum point of each graph were used in subsequent experiments.

Note that even though several of the algorithms maximizes the same criterion, the optimal parameter values are different. Some of the differences are quite large and can not be explained by random effects. (Rerunning the training phase showed minimal variability.) It is thus clear that the maximization method and the level of sub-optimality also have significant influence on the optimal tuning. The tuning results for the IWM are illustrated in Fig. 2. It can be seen that the optimal choice of θ is not very critical, and that any value between $10\sigma_e^2$ and $20\sigma_e^2$ would work well for both versions and for both low and high noise.

After completed training a new data set was generated for each noise case. These also consisted of 100 records with 500 samples each. Using the optimal parameter values, the algorithms were then run on all records and average losses

TABLE II

COMPARISON OF RESTORATION QUALITY BY AVERAGE LOSSES. LOW NOISE (RMS=5%).

	$L^{\text{miss+false}}$	L^{false}	L^{miss}	L^{SSQ}
IWM 1	7.01	3.86	4.69	0.43
IWM 2	4.72	2.21	3.49	0.17
SMLR	9.86	5.88	6.39	1.13
ICM	23.72	16.17	13.78	2.76
SA 1	8.09	4.02	6.04	0.68

TABLE III

AVERAGE LOSSES. HIGH NOISE (RMS=20%).

	$L^{\text{miss+false}}$	L^{false}	L^{miss}	L^{SSQ}
IWM 1	20.80	10.96	17.28	5.27
IWM 2	18.62	8.90	16.16	3.99
SMLR	22.59	12.55	18.08	5.36
ICM	39.76	26.33	27.88	9.56
SA 1	28.31	13.59	25.73	9.60

calculated. The exception was the two slowest algorithms, SA 2 and Simplex, which were run on only one record. For each noise case this particular record was selected as follows: The IWM 2, SMLR, and SA 1 were considered as the most interesting algorithms. The sum of $L^{\text{miss+false}}$ for these three algorithms were calculated for each record. Then, the record where this sum was closest to the corresponding average was selected. This record is also the one displayed in following figures.

For all algorithms the iterations were started from an initial configuration containing no spikes. The algorithms were implemented in C and all execution times were measured on the same 60 MHz Pentium PC.

IV. RESULTS

Average losses for all three noise cases are given in Tables II, III, and IV. As expected, all algorithms perform much better for RMS 5% than for RMS 20%. It can further be seen that the performance for RMS 5%+10% is quite similar to that for RMS 20%. Comparison of the different algorithms, shows that the IWM 2 performed better than all the others as measured by all loss functions and in all noise cases. Similarly, the IWM 1 performed second best, closely followed by the SMLR and SA 1. The SMLR performed worse than the SA 1 for RMS 5%, but better for RMS 20% and RMS 5%+10%. The ICM performed considerably worse than the other algorithms. (The SA 2 and Simplex are not included in this comparison since they were executed on only one record.)

The deconvolution output for one particular RMS 5% and RMS 20% record are displayed in Figs. 3 and 4 for visual inspection. The output for RMS 10%+5% were quite similar in appearance to that for RMS 20% and are not shown. The effect of the noise level is clearly seen: For RMS 20% the algorithms do not produce small spike estimates any longer (except the Simplex) and a number of small spikes have been missed. Slightly larger spikes have often been located in wrong locations. It can also be seen that some of the algorithms have inserted more small spikes than others. This reflects the different parameter values resulting from the training phase.

TABLE IV

AVERAGE LOSSES. LOW NOISE + MISSPECIFIED WAVELET (RMS=5%+10%).

	$L^{\text{miss+false}}$	L^{false}	L^{miss}	L^{SSQ}
IWM 1	20.23	11.27	16.44	4.80
IWM 2	17.45	8.83	14.87	3.81
SMLR	21.58	11.59	18.05	5.26
ICM	44.37	31.00	30.19	11.20
SA 1	24.56	12.58	21.19	6.70

For both noise levels the IWM 2 has successfully detected some spike combinations where one or more of IWM 1, SMLR, SA 1 and in particular ICM seems to have converged to sub-optimal configurations. The slow annealing SA 2 gave identical results to IWM 2 for RMS 5%. But for RMS 20% the two disagree in several areas. In some areas the IWM 2 seems to have performed better while the SA 2 seems to have performed better in others. Calculation of (2) showed that the IWM 2 actually had located the highest maximum of the posterior. This was not only true for the record as a whole, but also when calculation was performed for various sub-records. Even in the areas where the SA 2 apparently has performed better, the IWM 2 actually had located a higher maximum. (These results were true for both $\delta = 0$ and $\delta = \sigma_e^2/\sigma_a^2$, cf. the remark on tuning difference between IWM 2 and SA 2 in Section III.) Repeated execution of the SA 2 on the same record also showed that considerable variance remained in the estimates. This was not due to small adjustments in the final stages of the annealing schedule, but rather that some stochasticity existed in which local maximum was selected at an earlier stage.

The output from the Simplex was distinctly different from the other algorithms. Due to its different criterion, a large number of very small spikes was produced. If some post-processing had been used to remove the smallest spikes, the performance for RMS 5% would not be much inferior to the best of the other algorithms. But for RMS 20% the performance of the Simplex seems to have deteriorated more than that of the others.

Inspection of other records than the ones reproduced here showed considerable variability. Some records contained only "easy combinations" where all the best algorithms had performed almost identically. In other records differences larger than those displayed here were present. Although there were records where some of the other algorithms were closer to the true solution than the IWM 2, the overall impression was that this algorithm very often had managed to restore closely spaced spike combinations where one or more of IWM 1, SMLR, or SA 1 had failed. (The ICM consistently performed worse than the others.)

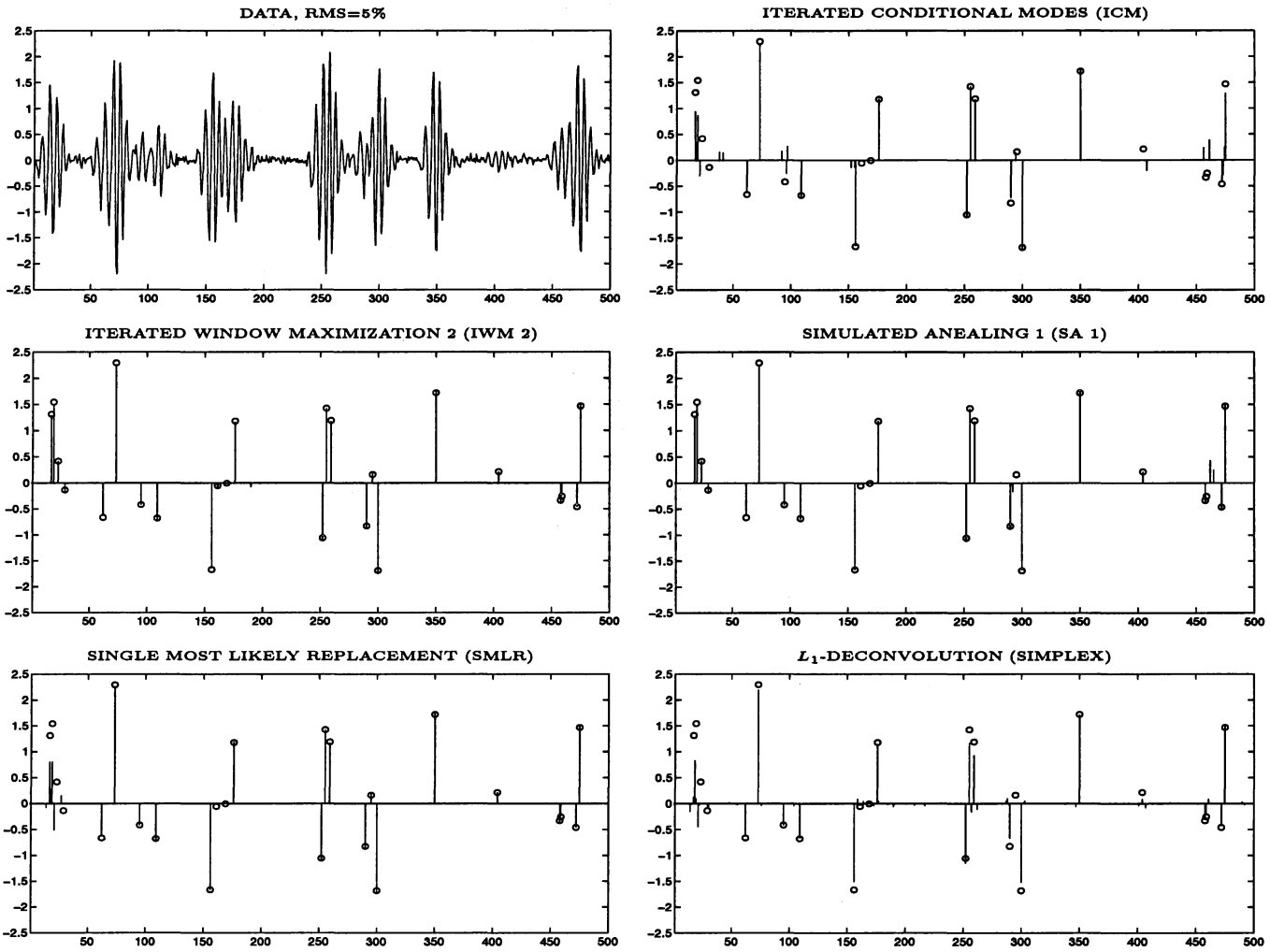

Fig. 3. Example of data and deconvolution results. Low noise (RMS 5%). Bars denote estimates and circles denote true values. The results for IWM 1 and SA 2 are not shown. The results for the IWM 1 were similar to those for the SMLR and the results for SA 2 were identical to those for the IWM 2.

Although large variations existed from record to record, rerunning the algorithms on a new data set showed that the variability of the average measures was negligible.

Average CPU execution times used by the different algorithms for RMS 5% and RMS 20% are displayed in Tables V and VI. Execution times for RMS 5%+10% were quite similar to those for RMS 20% and are not shown. As can be seen the IWM 1 executed fastest for both noise levels while the IWM 2 followed as good second, approximately a factor 2 slower. With exception of the ICM the fastest of the other algorithms was the SMLR which was approximately 50-80 times slower than the IWM 1. The SA 2 and Simplex were several orders of magnitude slower than all the others.

Some variations in execution time between the two noise levels can also be seen. Both IWM versions executed faster for RMS 20% than for RMS 5%. The reason was that fever spike estimates were produced in the high-noise case. On average, the number of spikes in each window was smaller and updating faster. The SMLR and ICM also executed faster for RMS 20%. For these algorithms the explanation was mainly that fewer iterations were needed for convergence. The SA versions, however, executed slower in the high-noise case. This was probably due to the different annealing schedules used. In the high-noise case more time was spent at high temperatures where many spikes were generated. Since the sparsity of the spike train was taken to advantage when computing the sampling distributions, many spikes slowed the algorithm down.

V. Discussion

A. IWM vs. SMLR

When a reasonable compromise between execution speed and quality of reconstruction is sought, the IWM and SMLR are probably the two most interesting alternatives examined. The IWM has been shown to perform considerably better in both respects, but the tested SMLR version has an advantage of a very simple implementation. In particular, there is no need for implementation choices such as window size and transition set, which are important for the performance of the IWM. At the expense of a more complicated implementation some improvements to the SMLR

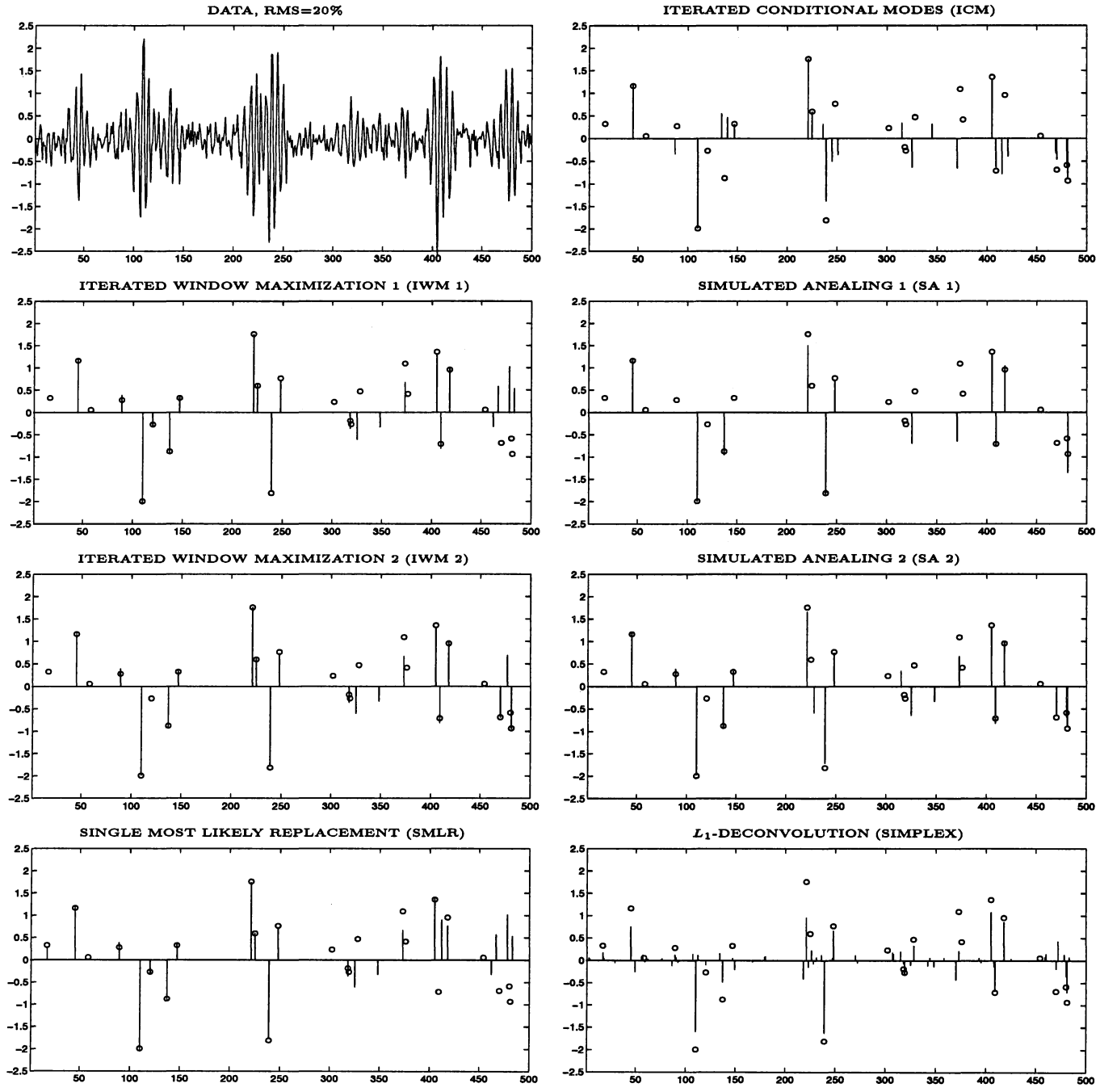

Fig. 4. Example of data and deconvolution results. High noise (RMS 20%). Bars denote estimates and circles denote true values.

are possible:

First, the SMLR and Simplex are the only of the tested algorithms whose execution time is not linear in the record length. Faster execution could be achieved by some sort of "block mode" restoration. For example, if the data is blocked in 9 blocks of 100 samples, each overlapping with 50%, the total execution time for the SMLR was determined to average close to 0.45s for the RMS 5% records. This would be approximately a factor 6 improvement, but the SMLR would still be considerably slower than both IWM versions. Even shorter blocks could also be considered, but the gain in execution time would have to be balanced against the additional sub-optimality introduced.

Secondly, other versions of the SMLR algorithm may be more efficient. In contrast to the one tested here, most SMLR and "SMLR like" algorithms are based on a state-space approach, e.g. [9], [18], [19], [24]. These are very efficient when the wavelet admits a low-order state-space representation (e.g. 4). Unfortunately, their computational load will usually be cubic in the order of the state space model. For many wavelets encountered in practice a rather high order (e.g. 10) will be necessary [25]. In such cases the state-space approaches could easily be less effective than the version tested here. In addition, when the wavelet

TABLE V
AVERAGE EXECUTION TIMES (S). LOW NOISE (RMS=5%).

	mean	st. dev.	relative
IWM 1	0.055	0.025	1.0
IWM 2	0.100	0.052	1.8
SMLR	2.928	0.878	53.2
ICM	0.466	0.237	8.5
SA 1	11.694	0.451	212.6
SA 2	29 min	0.000	31115.3
SIMPLEX	35 min	0.000	38483.3

TABLE VI
AVERAGE EXECUTION TIMES (S). HIGH NOISE (RMS=20%).

	mean	st. dev.	relative
IWM 1	0.029	0.006	1.0
IWM 2	0.048	0.019	1.7
SMLR	2.194	0.529	76.9
ICM	0.197	0.102	6.9
SA 1	19.305	0.488	676.6
SA 2	32 min	0.000	67641.4
SIMPLEX	34 min	0.000	70671.1

originally exists as a sampled impulse response, their implementation is complicated by the necessity to obtain a suitable state-space realisation [25].

The degree of optimality achieved by the SMLR version tested here can also be improved. One possibility is the state-space solution proposed in [16]. In addition to insertion and deletion, this modified version also considers shifting spikes to the left and right. Such a transition set is comparable to that of the IWM 1, but still much more restricted than the one used by IWM 2. One could also envisage incorporation of other transitions into the SMLR, but such extensions have not been devised. In contrast, the general framework of the IWM can easily handle arbitrarily complicated transitions [8].

B. IWM vs. ICM

Since the ICM is a restricted special case of the IWM, it is not surprising that the IWM gives much better deconvolution results. It is, however, more surprising that the IWM also executes significantly faster. After all, each update in the ICM is much simpler. The explanation is that the ICM needed many more iterations to converge. For example, for RMS 5% the average number of total passes through the data record before convergence was 8 for the IWM 1 and 187 for the ICM. The poor performance of the ICM may be seen as a direct consequence of the update-one-amplitude-at-a-time strategy. For example, motion of an isolated spike to a neighboring position can only take place through a large number of small reductions and increases. In fact, in this respect, the IWM can be seen as a compromise between the ICM and the SMLR, using window sizes containing more than a single spike, but less than the entire data record.

C. IWM vs. SA

The simulated annealing approach is sometimes referred to as globally optimal. The examples studied here illustrate that this is only for the idealized situation with an unrealizable slow annealing schedule. The fact that the IWM 2 located a higher maximum than the SA 2 for the RMS 20% record of Fig. 4, indicates that an even slower annealing schedule would be necessary to match the IWM 2. Considering that the SA 2 is approximately a factor 70,000 slower, the difference in efficiency between the two strategies is clearly demonstrated. As with the ICM, the low efficiency can be ascribed to the update-one-spike-at-a-time strategy. A better solution might be a stochastic generalization of the IWM, where several spikes could be updated simultaneously. But if the sole purpose is to maximize the posterior distribution, the advantage seems doubtful. Since any degree of optimality can be achieved by the deterministic algorithm, it seems likely that an increased computational effort is better spent on increasing the transition set than on introducing a stochastic element. However, if other properties of the posterior distribution such as expectations or independent samples are sought, a stochastic alternative may be necessary.

An advantage of the simple Gibbs sampling technique studied here is its generality. It can quite easily be adapted to a number of other situations. However, a considerable generalization potential also exists for the IWM [26].

D. IWM vs. Simplex

A major drawback of the Simplex is its slow execution. But, similar to the SMLR, faster execution is possible with some sort of block mode restoration. For the blocking example stated for the SMLR, the total execution time was found to be approximately 17s. This is a dramatic improvement, but the Simplex would still be orders of magnitude slower than the IWM versions.

Figs. 3 and 4 indicate that the global optimality of the Simplex is outweighed by its less suitable criterion. It should, however, be kept in mind that the comparison has been based on exactly the distributional assumptions embodied in the MAP methods. In practice, those assumptions will hardly be fulfilled exactly. The data might for example be corrupted by isolated extreme data points (e.g. resulting from equipment malfunction). Such a deviation would probably be more detrimental to the Gaussian based inference of the MAP methods than to the L_1 criterion of the Simplex. On the other hand, this problem could to a large extent be removed by some preprocessing. If, in addition, the spike train really is sparse, the MAP methods seems to incorporate the more realistic criterion. Since the MAP criterion can also be maximized much faster and with nearly global optimally (e.g. by the IWM 2) the Simplex seems less suited to the present problem.

E. The Effect of the Wavelet

The presented results are based on a single rather narrow-banded wavelet. To investigate the effect of the wavelet shape on the deconvolution results, some further experiments were performed. Two other wavelets were considered. The first was a fourth-order wide-band wavelet used by many researchers for simulation examples. The other was a distinctly non-spiky and narrow-band fourth-order wavelet. Explicit definitions of both are found in [2] on p. 85 and p. 88. For the wide-band wavelet all algorithms, even including the ICM, performed almost similarly and very well. For the narrow-band wavelet the ranking of the algorithms was the same as in the presented test results, but the differences were even larger. The choice of algorithm thus seems most important when the wavelet is relatively narrow-band.

In practical situations, the probably most important source of modeling error concerns specification of the wavelet. The true wavelet may be poorly known, it may not be time-invariant, or both. It has already been demonstrated that the tested algorithms are quite robust with respect to a wavelet degraded by independent noise. In fact, the effect is very similar to degradation of the data with a similar amount of noise. Such degradation may be realistic when the wavelet has been obtained by direct measurement, but frequently more systematic errors will also be present. For example, both ultrasound and seismic wavelets are known to undergo a high frequency attenuation as they propagate through the medium [2], [27]. Another source of error, inherent in some wavelet estimation procedures [27], [28], is to assume a symmetric or minimum phase shape. Some experiments were performed to assess the importance of such sources of error. Only the IWM 2 and SMLR were tested, but it is reasonable to assume that the observed performance degradation is representative also for the other algorithms.

The results were generally discouraging. Ignoring only a moderate degree of high frequency attenuation degraded deconvolution performance considerably. (This situation was simulated by using a low-pass filtered version of the wavelet in Fig. 1 for data generation and then deconvolving the data with the unprocessed original.) Performing the deconvolution with either the minimum phase or symmetric version of the wavelet was clearly unsatisfactory. Not only did the algorithms have problems with detecting overlapping wavelets correctly, but problems were also encountered with detection of isolated wavelets. Either the algorithms had to be tuned such that small spikes were missed, or otherwise large spikes would be split. Some experiments were also performed with state-space approximations of the wavelet. For the wavelet in Fig. 1 an order of approximately 10 was necessary to obtain satisfactory deconvolution results.

In general, when large "systematic" errors are present in the wavelet estimate, none of the algorithms examined here seem appropriate. In such cases it may be better to model the uncertainty of the wavelet explicitly. An alternative is the modification of the SMLR algorithm proposed in [29] or a similar modification of the IWM [26]. A particular conclusion concerns the sampling frequency. Since real spikes will of course not be located on integer multiples of the sampling interval, a considerably higher sampling frequency than applied to the wavelet in Fig. 1 would be necessary to ensure approximate wavelet invariance.

VI. Conclusion

The IWM algorithm performed better than the other algorithms under consideration in all cases examined. Both quality of restoration and execution times were improved. The advantage of the IWM seems most important for difficult deconvolution problems when the wavelet is relatively narrow-band. It is believed that this new technique represents an interesting alternative to established procedures in many applications.

All examined algorithms proved reasonably robust when either data or wavelet were contaminated by independent noise, but were highly sensitive to more "systematic" distortions of the wavelet shape.

Acknowledgments

This work was supported by grants from the Research Council of Norway. I am grateful to Prof. Erik Bølviken, University of Oslo and Prof. Torfinn Taxt, University of Bergen for valuable comments.

References

[1] J. M. Mendel, *Optimal Seismic Deconvolution: An Estimation-Based Approach.* New York: Academic, 1983.

[2] J. M. Mendel, *Maximum-Likelihood Deconvolution: A Journey into Model-Based Signal Processing.* New York: Springer-Verlag, 1990.

[3] M. S. O'Brien, A. N. Sinclair, and S. M. Kramer, "Recovery of a sparse spike time series by L_1 norm deconvolution," *IEEE Trans. Signal Processing,* vol. 42, no. 12, pp. 3353-3365, dec. 1994.

[4] C. H. Chen and S. K. Sin, "On effective spectrum-based ultrasonic deconvolution techniques for hidden flaw characterization," *J. Acoust. Soc. Am.,* vol. 87, no. 3, pp. 976-987, March 1990.

[5] G. Demoment, R. Reynaud, and A. Herment, "Range resolution improvement by a fast deconvolution method," *Ultrasonic Imaging,* vol. 6, pp. 435-451, 1984.

[6] H. Kwakernaak, "Estimation of pulse heights and arrival times," *Automatica,* vol. 16, pp. 367-377, 1980.

[7] B. S. Atal and J. R. Remde, "A new model of LPC excitation for producing natural sounding speech at low bit rates," *Proc. IEEE, Int. Conf. Acoust., Speech, Signal Processing,* vol. 3-5, pp. 614-617, 1982.

[8] K. F. Kaaresen, "Deconvolution of sparse spike trains by iterated window maximization," submitted for publication, *IEEE Trans. Signal Processing.*

[9] J. J. Kormylo and J. M. Mendel, "Maximum likelihood detection and estimation of Bernoulli-Gaussian processes," *IEEE Trans. Inform. Theory,* vol. IT-28, no. 3, pp. 482-488, May 1982.

[10] J. Besag, "On the statistical analysis of dirty pictures," *Journ. Royal Statist. Soc. Ser. B,* vol. 48, no. 3, pp. 259-302, 1986.

[11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science,* vol. 220, pp. 671-680, 1983.

[12] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-6, pp. 721-741, Nov. 1984.

[13] H. Taylor, S. Banks, and F. McCoy, "Deconvolution with the L_1 norm," *Geophys.,* vol. 44, no. 1, pp. 39-52, dec. 1979.

[14] M. Lavielle, "Bayesian deconvolution of Bernoulli-Gaussian processes," *Signal Processing,* Elsevier, vol. 33, no. 1, pp. 67-79, July 1993.

[15] G. Hayward and J. E. Lewis, "Comparison of some non-adaptive deconvolution techniques for resolution enhancement of ultrasonic data," *Ultrasonics*, vol. 27, pp. 155-164, May 1989.

[16] C.-Y. Chi, J. M. Mendel, "Improved maximum-likelihood detection and estimation of Bernoulli-Gaussian processes," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 429-435, March 1984.

[17] C.-Y. Chi and J. M. Mendel, "Viterbi algorithm detector for Bernoulli-Gaussian process," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 3, pp. 511-519, June 1985.

[18] G. B. Giannakis, J. M. Mendel, and X. Zhao, "A fast prediction-error detector for estimating sparse-spike sequences" *IEEE Trans. Geosci. Remote Sensing*, vol. GE-27, no. 3, pp. 344-351, May 1989.

[19] A.-C. Hsueh and J. M. Mendel, "Minimum-variance and maximum-likelihood deconvolution for noncausal channel models" *IEEE Trans. Geosci. Remote Sensing*, vol. GE-23, no. 6, pp. 797-808, November 1985.

[20] J. Goutsias and J. M. Mendel, "Maximum likelihood deconvolution: An optimization theory perspective," *Geophys.*, vol. 51, no 6, pp. 1206-1220, June 1986.

[21] Y. Goussard, G. Demoment, and J. Idier, "A new algorithm for iterative deconvolution of sparse spike trains," *Proc. IEEE, Int. Conf. Acoust., Speech, Signal Processing*, pp. 1547-1550, 1990.

[22] F. A. Ficken, *The simplex method of linear programming.* New York: Holt, Reinhart, and Winston, 1961.

[23] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C, 2nd ed.* New York: Cambridge University Press, 1992.

[24] C.-Y. Chi, "A fast maximum likelihood estimation and detection algorithm for Bernoulli-Gaussian processes" *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, no. 11, pp. 1636-1639, Nov. 1987.

[25] J. M. Mendel, "Minimum-variance deconvolution" *IEEE Trans. Geosci. Remote Sensing*, vol. GE-19, no. 3, pp. 161-171, July 1981.

[26] K. F. Kaaresen, "Maximum a posteriori deconvolution of sparse structures by iterated window maximization," under preparation.

[27] J.A. Jensen and S. Leeman, "Nonparametric estimation of ultrasound pulses," *IEEE Trans. Biomed. Eng.* vol. BME-41, no. 10, pp. 929-936, Oct. 1994.

[28] E. A. Robinson and S. Treitel, *Geophysical Signal Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1980.

[29] F. Champagnat, J. Idier and G. Demoment, "Deconvolution of sparse spike trains accounting for wavelet phase shifts and colored noise," *Proc. IEEE, Int. Conf. Acoust., Speech, Signal Processing*, vol. 3, pp. 452-455, 1993.