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 maximize 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. Synthetic data examples are presented that support these claims.**

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].

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

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].

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 initializa-

tion 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 algorithm [6]. But a major difference is that the efficiency of the SMLR 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 will in many cases lead to faster computation.

The rest of the paper is organized as follows: In Section II the mathematical structure of the algorithm is derived in detail. Implementation issues are considered in Section III and selection of parameters in Section IV. In Section V some synthetic data examples are provided which demonstrate that the new algorithm can simultaneously improve both optimality and execution time relative to an efficient version of the SMLR algorithm. Finally, parallelization possibilities are pointed out in Section VI.

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:

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

The matrices

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

will be important in the following. The parameter γ 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 $\gamma = 0$ it is the least squares estimator from multivariate regression. Using some algebra on (10) shows that

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

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. In the Bernoulli case $g(\mathbf{t})$ also assumes a particularly simple form. Then, $p(\mathbf{t}) = \lambda^M(1-\lambda)^{N-M}$, 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 as described next.

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:

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

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

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

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 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:

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

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 $\mathbf{S}^w = (\mathbf{H}^w)'\mathbf{H}^w + \gamma\mathbf{I}$ 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:

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

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

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

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

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

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) and (23) 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

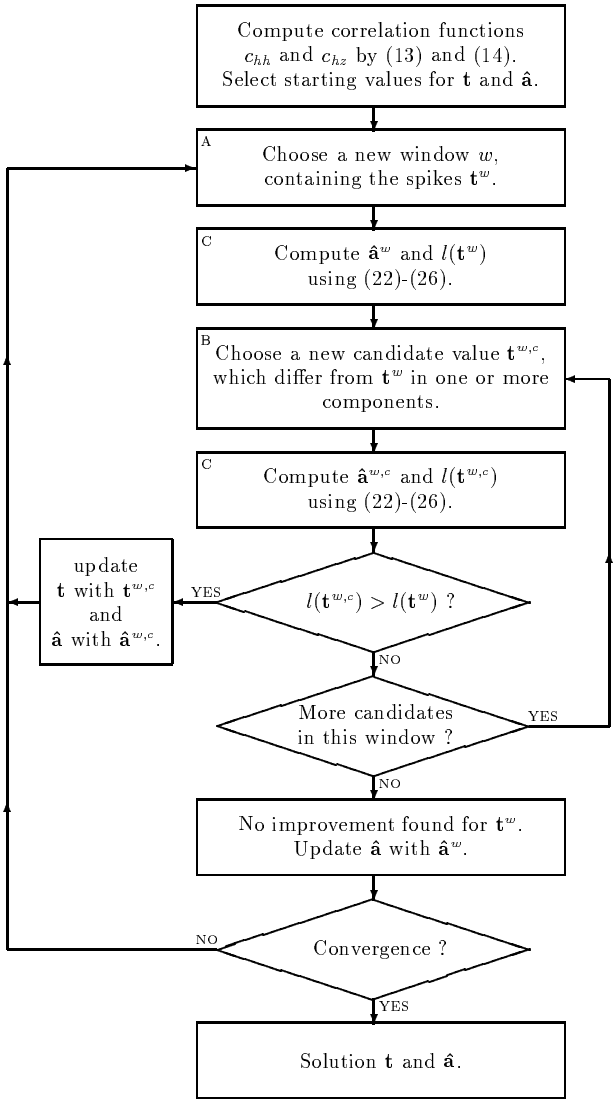

Fig. 1. The structure of the proposed algorithm. Boxes labeled A, B, and C are discussed in corresponding sub-sections to Section III.

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 should 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.

III. Implementation

Based on the formulas of the previous section, the algorithm can be implemented as depicted in Fig. 1. This basic structure leaves considerable flexibility, and some important issues will be discussed further here.

A. Window Selection

A reasonable choice of window positions is to cycle sequentially through all currently existing spikes, and for each spike try transitions with a window centered on that spike. It is proved in Appendix A that such systematic update guarantees convergence to amplitudes that are globally optimal given the final value of $\mathbf{t}$. The iterations should be continued until a complete scan fails to change either $\mathbf{t}$ or $\hat{\mathbf{a}}$ by more than an error threshold. (Note from Fig. 1 that an iteration may improve $\hat{\mathbf{a}}$ even though no improvement is found for $\mathbf{t}$.)

The window size is important for efficiency. Generally, the computational burden is an increasing function of this variable, since the cost of computing (24) grows rapidly when the number of spikes in the window increases. Very small windows may, however, also slow the algorithm down. In this case the necessary number of iterations will typically increase considerably because the strong interactions between close spikes are not taken into account at each update.

The window size may also affect the optimality of the algorithm. Even though the conditional optimality of the final amplitudes is independent of the window size, the final value of $\mathbf{t}$ is not. In general, a smaller window size can be expected to yield reduced optimality. The window size should therefore be selected to balance the conflicting interests of optimality and computational efficiency. The simulations in Section V (and numerous others performed by the author), indicate that a window size approximately equal to the length of the wavelet will usually lead to negligible loss of optimality, while still retaining efficient execution.

B. Choice of Transitions

The number of new candidate values for $\mathbf{t}$ considered in each window 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.
2. In addition to the transitions above, try also a number of transitions involving two spikes, e.g. move both to the left, 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. Such a procedure may in many cases present a computationally feasible solution that are practically equivalent to the (unattainable) global brute force method, which consist of testing all 2^N possibilities.

It is not necessary to try the most complicated transitions when the algorithm is far from convergence. For increased efficiency it is 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. This approach was used in the implementations tested in Section V.

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.

C. Solution of the Linear Systems

The major computational burden of the algorithm is to compute the optimal amplitudes by (24). Since $\mathbf{S}^w$ is positive definite symmetric, a natural choice is to compute the Cholesky decomposition of $\mathbf{S}^w$ and "back substitute" with $\mathbf{v}^w$. 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 is more efficient to handle each of the low order cases separately, using closed form solutions of the linear system which directly give the components of $\hat{\mathbf{a}}^w$ in terms of the components of $\mathbf{S}^w$ and $\mathbf{v}^w$. This approach was used in the implementations tested in Section V. A further simplification is obtained with appropriate normalization of the wavelet, which makes the diagonal elements of $\mathbf{S}^w$ equal to one. (Together with the symmetry property this makes the number of variable elements in $\mathbf{S}^w$ equal to only 0,1 and 3 for respectively 1,2, and 3 spikes.)

IV. Selecting Parameter Values

The inverse signal-to-noise ratio γ 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 γ

If estimates or a priori knowledge about the noise and amplitude variance are available, γ can be determined directly from its definition, $\gamma = \sigma_e^2/\sigma_a^2$. Otherwise, an alternative is to use $\gamma = 0$. This corresponds to a non-informative prior [16] for the amplitudes. Simulation tests 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. Using a small value of γ removes the problem.

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, cf. (26). 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 θ. It is, however, argued in Appendix B that the value thus implied by the MAP criterion is in general not suitable. Some other suggestions for selection will be given here instead.

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 with identical statistical properties to the real data could be generated. Then, θ could be selected to optimize average performance on this data set relative to a realistic loss function. An example is found in [17]. 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 [17]). A simple guideline for choosing a sensible value will now be given. The argument is based on requiring a certain probability of false detection to equal a (small) constant. This leads to a value of θ proportional to σ_e^2. To define the error probability, suppose the algorithm is trying to decide whether to insert a spike at a given position in an otherwise empty window. Assume 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, suppose for simplicity that $\gamma = 0$ is used. Define

P^{false} as the probability that the algorithm accepts the spike, given that the window contains no real spikes. In Appendix C it is shown that

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

where F is the cumulative distribution function of a chi-square variable with one degree of freedom. From (27) it is clear that choosing $\theta = \chi\sigma_e^2$, where χ is a quantile high up in the tail of the chi-square distribution with one degree of freedom, yields a small error probability, independent of the noise variance. For the simulation examples in the next section, $\chi = 10$ was used, corresponding to $P^{\text{false}} \approx 0.0016$. Numerous other simulations performed by the author show that choosing χ in the range 10-20 works well 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 larger χ.

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)-(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 parameter values implied by the MAP criterion can not be expected to give optimal results. 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 training set. A sensible starting point is to choose both parameters on the form of a constant multiplied by σ_e^2.

Concrete specification of alternatives to the Bernoulli/geometric form of $p(\mathbf{t})$ is application dependent, and will not be considered here.

V. Computer Simulation

To evaluate the new algorithm, two implementations were tested. The first, denoted by IWM 1 (iterated window maximization 1), corresponded to transition set 1 from Section III-B where only single insertions, deletions, and motion of one spike with one sample were considered. The second, IWM 2, was based on transition set 2, where the following two spike transitions were considered: Splitting one spike into two, joining two spikes into one, and moving two spikes simultaneously. In each case all possible replacements were tried, subject to the condition that each new spike should be positioned no more than 5 samples from one of the replaced ones. Only spike pairs with inter-distance of no more than 10 samples were considered for join or simultaneous move. In both versions border effects were treated as indicated at the end of Section II-D. The two IWM versions were compared to an efficient implementation of the SMLR algorithm [6] proposed by Goussard et. al. [12]. (This implementation was chosen for comparison because it shares the advantage of the IWM that no state-space representation of the wavelet is needed. Though state-space based alternatives [6], [11], [13] are highly efficient for synthetic low-order models, their efficiency rapidly declines for higher-order models that are typically needed to fit real wavelets satisfactory.) For all algorithms, the iterations were started from an initial estimate without any spikes.

TABLE I
CPU execution times used to computed the reconstructions in Figs. 2-4. Measured on a 60 MHz Pentium PC.

Wavelet	SNR	Execution times (s)		
		IWM 1	IWM 2	SMLR
broad-band	10 dB	0.044	0.071	0.720
narrow-band	10 dB	0.028	0.049	0.500
narrow-band	20 dB	0.066	0.110	1.100

For easy comparison with previously published results, a 300 samples long sparse spike train explicitly defined in [18] was used for the simulations. The spike train was convolved with a broad-band Kramer wavelet (Fig. 2a), defined e.g. in [19] p. 85. White Gaussian noise was added to make the SNR (defined as the ratio of mean power of noise free signal to noise variance) equal to 10 dB. For these data, the IWM 1 detected all major spikes correctly, whereas several small ones that were "buried in noise" went undetected (Fig 2). This compares well with other results in the literature [6], [7], [11], [20]. The IWM 2 and the SMLR gave identical reconstructions to the IWM 1 for this example. However, the IWM 1 and IWM 2 were faster than the SMLR by a factor 16 and 10 respectively (Table I).

The Kramer wavelet was considered mainly because of its widespread use in the literature. The broad-band nature of this wavelet makes the deconvolution problem unrealistically easy compared to many wavelets commonly encountered in seismic and ultrasonic applications. Therefore a narrow-band wavelet (Fig. 3a), explicitly defined in [19] p. 85, was selected for the next simulations. Using the same spike train as before and keeping the SNR at 10 dB, showed that all algorithms

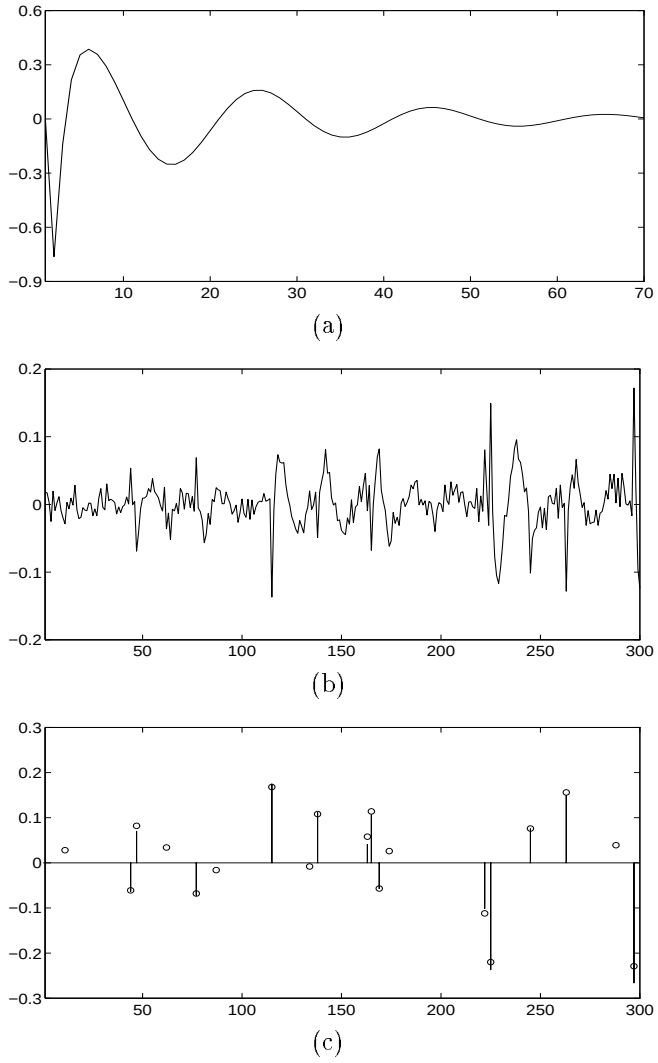

Fig. 2. Deconvolution of broad-band data: (a) Fourth order Kramer source wavelet; (b) Synthetic data, SNR = 10 dB; (c) The identical results obtained by two versions of the proposed algorithm, IWM 1 and IWM 2, and by the SMLR algorithm. Bars depict estimates and circles depict true values.

performed markedly worse (Fig. 3) than in the broad-band situation . The estimate from the IWM 1 was still identical to the SMLR, whereas the IWM 2 managed to resolve some closely spaced spikes where both the others failed (cf. the estimate around sample 170 and 220).

Increasing the SNR to 20 dB lead to improved detection for all algorithms (Fig. 4), but this time also the IWM 1 performed better than the SMLR, which appeared to have locked into sub-optimal configurations at several places (cf. e.g. the estimate around sample 50 and 160). The IWM 2 again improved further on the IWM 1, thus demonstrating the advantage of including two-spike moves.

For both narrow-band examples the IWM versions improved the execution times of the SMLR by approximately the same factors as in the broad-band example (Table I). The speed advantage of the IWM will, however, increase with the record length due to a non-linearity of the SMLR version used here [12], [17]. Table I also shows some variation in execution time between the different data sets. This is mainly due to differences in the number of estimated spikes, with more spikes giving slower execution. If very dense estimates are sought, the IWM may become inefficient, and other methods should be considered.

TABLE II
Execution time as a function of window size when processing the data in Fig 4 with the IWM 1.

Window size	Execution time (s)
3	0.038
11	0.033
21	0.060
31	0.066
41	0.099
61	0.154
∞	1.582

For all reconstructions above, a window size of 31 samples was used. Execution times for some other window sizes are given in Table II. For the given example, the execution time increased with the window size, except for the two smallest sizes (3 and 11) where increasing the window size actually decreased the execution time slightly. The corresponding reconstructions (not shown) were virtually identical for window sizes from 21 to ∞, but slightly inferior for window sizes 3 and 11. These results quantify the discussion in Section III-A, and confirms the suggestion that a window size approximately equal to the length of the wavelet (disregarding small samples at the end) combines efficient execution with no significant loss of optimality. Note in particular from Table II that the chosen window size of 31 samples reduced the execution time with a factor 24 compared to the non-windowed strategy (window size ∞). This illustrates the computational savings resulting from the window maximization. This advantage will of course increase rapidly with the record length.

VI. 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

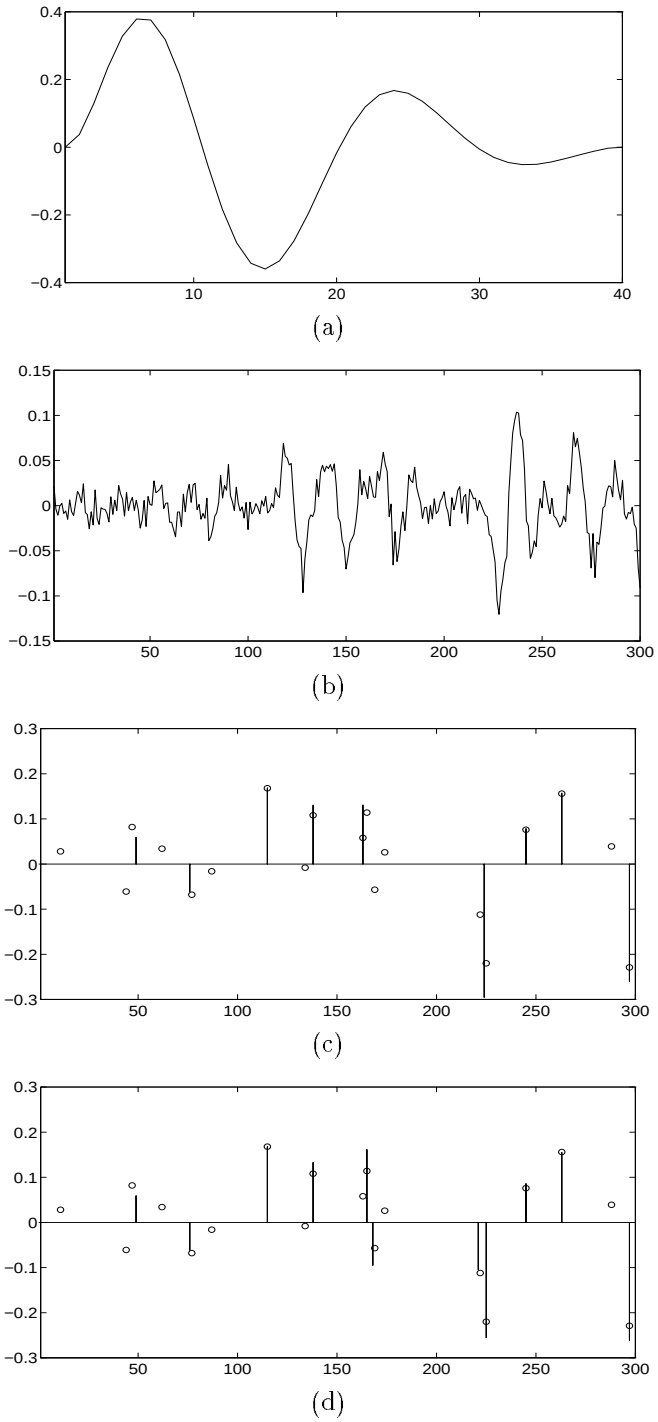

Fig. 3. Deconvolution of low SNR narrow-band data: (a) Fourth order narrow-band source wavelet; (b) Synthetic data, SNR = 10 dB; (c) Results obtained by both the IWM 1 and the SMLR; (d) Results obtained by the IWM 2. Bars depict estimates and circles depict true values.

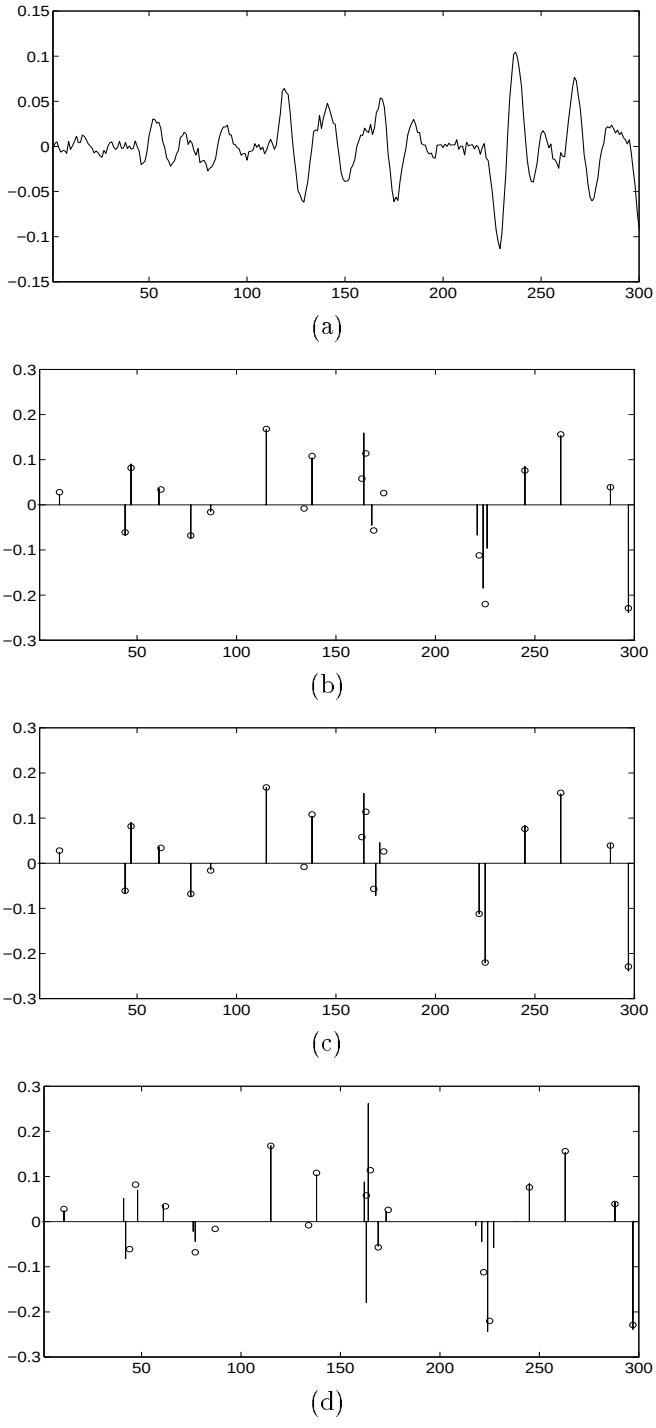

Fig. 4. Deconvolution of high SNR narrow-band data: (a) Synthetic data, SNR = 20 dB; (b) Results obtained by the IWM 1; (c) Results obtained by the IWM 2; (d) Results obtained by the SMLR. Bars depict estimates and circles depict true values. The wavelet used in this example is depicted in Fig. 3 (a).

property often noted for Monte Carlo methods such as Gibbs sampling (see [21]).

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 [20]. 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.

VII. 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 simulations indicate that simultaneous advantages in speed and optimality can be obtained relative to existing alternatives. Further evidence is presented in [17]. 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 in a certain sense not 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 are also possible. Some of these are discussed in [22].

Acknowledgments

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. This work was supported by grants from the Research Council of Norway.

Appendices

A. Proof of a Convergence Property

Theorem:

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

Proof:

Note first that since each accepted update in the algorithm increases $p(\mathbf{a}^w, \mathbf{t}^w|\mathbf{z}, \mathbf{a}^{\overline{w}}, \mathbf{t}^{\overline{w}})$, factorization (19) shows that the algorithm increases the posterior density at each step, and therefore must be convergent.

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 d such that when d is added to $\hat{a}_i$ the function value is increased. This can be written as

$$p(\hat{\mathbf{a}} + \mathbf{d}, \mathbf{t}|\mathbf{z}) > p(\hat{\mathbf{a}}, \mathbf{t}|\mathbf{z}), \tag{28}$$

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 (28) and canceling the two equal factors gives

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

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 $\mathbf{a}^w$.

B. 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 [23]. 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 [21]. 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 IV, will make the estimator scale invariant.

The approach taken here differs from the commonly advocated solution [1], [6], which 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. (Such a marginal MAP estimator is also used in the SMLR version tested here.) 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 γ 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.

C. Derivation of an Error Probability

The distributional result reached here will be somewhat more general than necessary to deduce the probability of false detection defined in section IV-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

$$l(\mathbf{t}^w) = ||(\mathbf{S}^w)^{-1/2}\mathbf{v}^w||^2 - \theta M^w. \tag{29}$$

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 γ 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 (29) shows that $l(\mathbf{t}^w)$ has the distribution of $\sigma_e^2 x - \theta M^w$, where x is a chi-squared variable with M^w degrees of freedom. The error probability (27) follows by setting $M^w = 1$ and noting that the spike will be inserted if $l(\mathbf{t}^w) > 0$. (An empty window yields $l(\mathbf{t}^w) = 0$, cf. (25), (26), and the following comment.)

The assumption of correct estimation outside the window can be relaxed. Since $\mathbf{v}^w$ only depends on spikes with 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] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis, 2nd ed.* New York: Springer-Verlag, 1988.

[17] K. F. Kaaresen, "Maximum a posteriori deconvolution of sparse spike trains," Statistical Research Report, Institute of Mathematics, University of Oslo, March 1996.

[18] J. M. Mendel and J. Kormylo, "Single-channel white-noise estimators for deconvolution," *Geophysics,* vol. 43, no. 1, pp. 102-124, Feb. 1978.

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

[20] 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.

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

[22] K. F. Kaaresen, "Efficient maximum a posteriori deconvolution of sparse structures," Statistical Research Report, Institute of Mathematics, University of Oslo, Aug. 1996.

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