Viterbi Algorithm Detector for Bernoulli–Gaussian Processes

CHONG-YUNG CHI, MEMBER, IEEE, AND JERRY M. MENDEL, FELLOW, IEEE

Abstract—**This paper shows how to apply the Viterbi algorithm to detect randomly located impulses which have Gaussian distributed amplitudes. Our detector can deal with cases of severely overlapping wavelets. Experimental results and comparisons to Kormylo and Mendel's [12] single-most-likely-replacement detector are provided, using synthetic data.**

I. Introduction

THE Viterbi algorithm (VA) [1], [6], [7] has been applied to a wide range of problems in the communication's area. Recently, it was also applied to well-log deglitching and seismogram inversion [2], [14].

In this paper, we show how to apply the VA to the detection of a Bernoulli–Gaussian sequence, and we compare it, via some simulation studies, to Kormylo and Mendel's single-most-likely-replacement (SMLR) detector [12]. Both the VA and SMLR detectors are derived based on the same likelihood function and are suboptimal. The VA detector has a very nice parallel processing structure and its performance is comparable to the SMLR detector. The VA detector is noniterative and so its computational load is constant. On the other hand, the SMLR detector is iterative, so its computational load depends upon its initial conditions.

The total computation of the VA detector depends upon the dimension of the state innovation diagram. When parallel processing is available, the VA detector requires only about two Kalman filters. On the other hand, the SMLR detector requires $2I$ Kalman filters, where I is the total number of iterations for its convergence. In this case, the VA detector becomes much faster than the SMLR detector.

We assume that all the information needed to implement the VA detector is known *a priori*. What that information is will be made clear in a later section.

II. System and Statistical Models

As in Kormylo and Mendel [12], our starting point is the familiar discrete-time convolutional model (see Fig. 1)

$$z(k) = \sum_{j=1}^{k} V(k-j)\,\mu(j) + n(k). \tag{1}$$

Manuscript received August 31, 1983. This work was supported by the National Science Foundation under Grant ECS-8200882 and by the sponsors of the USC Geo-Signal Processing Program.

C.-Y. Chi is with the Jet Propulsion Laboratory, Pasadena, CA 91109.

J. M. Mendel is with Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA 90089.

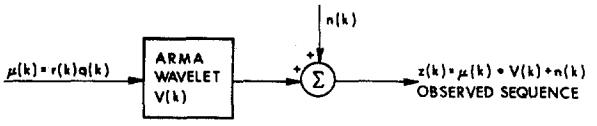

Fig. 1. System model.

In this model, $\mu(k)$ is the input impulse signal train which is assumed to be Bernoulli–Gaussian [3]. It can be expressed as the following product model:

$$\mu(k) = r(k)\,q(k) \tag{2}$$

in which $r(k)$ is zero mean, white, and Gaussian, with variance

$$E[r^2(k)] = C \tag{3}$$

and, $q(k)$ is a Bernoulli sequence, with

$$Pr[q(k)] = \begin{cases} \lambda: & q(k) = 1 \\ 1-\lambda: & q(k) = 0. \end{cases} \tag{4}$$

Noise $n(k)$ is zero mean, white, and Gaussian, with variance

$$E[n^2(k)] = R. \tag{5}$$

Signal $V(k)$ (which can be thought of as the impulse response of a linear time-invariant system) is assumed to have an autoregressive moving average (ARMA) structure, i.e.,

$$V(z) = \frac{\sum_{i=1}^{n} \beta_i z^{-i+1}}{1 - \sum_{i=1}^{n} \alpha_i z^{-i}}. \tag{6}$$

System (1)–(6) can also be realized by the state-variable model

$$\boldsymbol{x}(k) = \Phi \boldsymbol{x}(k-1) + \Upsilon \mu(k) \tag{7}$$

and

$$\boldsymbol{z}(k) = \boldsymbol{h}'\boldsymbol{x}(k) + n(k) \tag{8}$$

where Φ is an $n \times n$ matrix, Υ and $\boldsymbol{h}$ are $n \times 1$ vectors, and Φ, Υ, and $\boldsymbol{h}$ are functions of α_i and β_i. Of course, given a transfer function of a linear time-invariant system, there exist many $(\Phi, \Upsilon, \boldsymbol{h})$'s which generate the same output $z(k)$.

In Section III, we begin with the data $\{z(1), z(2), \cdots,$

$z(N)\}$ and, as based on the above system and statistical models, derive the VA for detecting $\{q(1), q(2), \cdots, q(N)\}$ by maximizing the likelihood function $S\{\boldsymbol{q}_N|z_N\}$

$$S\{\boldsymbol{q}_N|z_N\} = p(z_N|\boldsymbol{q}_N)\ \Pr(\boldsymbol{q}_N) \tag{9}$$

where z_N and $\boldsymbol{q}_N$ are the two column vectors of

$$z_k = \text{col}\ (z(k),\ z(k-1), \cdots, z(1)) \tag{10}$$

and

$$\boldsymbol{q}_k = \text{col}\ (q(k),\ q(k-1), \cdots, q(1)) \tag{11}$$

when $k = N$. Before we derive the VA, we briefly review the SMLR detector so that one can easily see the characteristics of both detectors.

The SMLR detector is an iterative search algorithm that compares the likelihood for a "reference" sequence $\boldsymbol{q}_r$ to the likelihoods of a limited number of different "test" sequences $\boldsymbol{q}_t$, in each iteration. The SMLR detector was derived by assuming that $\boldsymbol{q}_t$ differs from $\boldsymbol{q}_r$ at only one location, so that there are only N possible test sequences for a given reference sequence. The log-likelihood-ratio decision rule for choosing between $\boldsymbol{q}_r$ and $\boldsymbol{q}_t$ is given by

$$\ln \Lambda_{\text{tr}}(k) = \ln \frac{S\{\boldsymbol{q}_t|z_N\}}{S\{\boldsymbol{q}_r|z_N\}} = \ln \frac{p(z_N|\boldsymbol{q}_t)\ \Pr(\boldsymbol{q}_t)}{p(z_N|\boldsymbol{q}_r)\ \Pr(\boldsymbol{q}_r)} \underset{q_r}{\overset{q_t}{\gtrless}} 0 \tag{12}$$

where $\boldsymbol{q}_t$ is the sequence

$$q_t(i) = \begin{cases} q_r(i), & \text{for all } i \neq k \\ 1 - q_r(i), & i = k. \end{cases} \tag{13}$$

Let k' be associated with the maximum value of $\ln \Lambda_{\text{tr}}(k)$ $(k = 1, 2, \cdots, N)$. Then the SMLR test sequence is

$$q_t'(i) = \begin{cases} q_r(i), & \text{for all } i \neq k' \\ 1 - q_r(i), & i = k'. \end{cases} \tag{14}$$

It is also true that the log-likelihood function evaluated for $\boldsymbol{q}_t'$ is at least as large as its value evaluated for $\boldsymbol{q}_r$.

As pointed out by Kormylo and Mendel [12], the SMLR search algorithm, initiated by $\boldsymbol{q}_r = \hat{\boldsymbol{q}}^{(o)}$, computes N log-likelihood ratios corresponding to N different $\boldsymbol{q}_t$ sequences. The most likely $\boldsymbol{q}_t$ sequence is used as the reference sequence $\hat{\boldsymbol{q}}^{(1)}$ for the next iteration. If after i iterations we obtain a reference $\boldsymbol{q}_r = \hat{\boldsymbol{q}}^{(i)}$, which is more likely than any of the corresponding $\boldsymbol{q}_t$ sequences, then the search stops and $\hat{\boldsymbol{q}} = \hat{\boldsymbol{q}}^{(i)}$ is the final detected event sequence.

III. Derivation of the VA for Detecting $q(k)$

A. Likelihood Function

Our objective is to obtain optimal estimates $\hat{\boldsymbol{q}}_N$, such that $S\{\boldsymbol{q}_N|z_N\}$ is a maximum when $\boldsymbol{q}_N = \hat{\boldsymbol{q}}_N$. We now obtain expressions for $p(z_N|\boldsymbol{q}_N)$ and $\Pr(\boldsymbol{q}_N)$.

First, we observe that

$$p(z_N|\boldsymbol{q}_N) = p(z(N)|z_{N-1}, \boldsymbol{q}_N) \cdot p(z_{N-1}|\boldsymbol{q}_N). \tag{15}$$

From (1) and (2), we observe, also, that $\{z_{N-1}, \boldsymbol{q}_{N-1}\}$ is independent of $q(N)$; hence, (12) can be simplified to

$$p(z_N|\boldsymbol{q}_N) = p(z(N)|z_{N-1}, \boldsymbol{q}_N) \cdot p(z_{N-1}|\boldsymbol{q}_{N-1}). \tag{16}$$

Continuing the backward development of (13), we find that

$$\begin{aligned} p(z_N|\boldsymbol{q}_N) &= p(z(N)|z_{N-1}, \boldsymbol{q}_N) \\ &\quad \cdot p(z(N-1)|z_{N-2}, \boldsymbol{q}_{N-1}) \\ &\quad \cdots p(z(2)|z_1, \boldsymbol{q}_2) \cdot p(z(1)|\boldsymbol{q}_1) \\ &= \prod_{k=1}^{N} p(z(k)|z_{k-1}, \boldsymbol{q}_k) \end{aligned} \tag{17}$$

where z_0 indicates no observations.

Because $q(1), q(2), \cdots, q(N)$ are independent,

$$\Pr(\boldsymbol{q}_N) = \prod_{k=1}^{N} \Pr(q(k)). \tag{18}$$

Substituting (17) and (18) into (9), we obtain

$$S\{\boldsymbol{q}_N|z_N\} = \prod_{k=1}^{N} p(z(k)|z_{k-1}, \boldsymbol{q}_k)\ \Pr(q(k)). \tag{19}$$

B. Objective Function

Maximizing $S\{z_N|\boldsymbol{q}_N\}$ is equivalent to minimizing $-\ln S\{\boldsymbol{q}_N|z_N\}$, because $-\ln(\cdot)$ is a monotonically decreasing function. We, therefore, define the objective function to be minimized as

$$J(\boldsymbol{q}_N) = -\ln S\{\boldsymbol{q}_N|z_N\} = \sum_{k=1}^{N} d(k, \boldsymbol{q}_k) \tag{20}$$

where

$$d(k, \boldsymbol{q}_k) = -\ln p(z(k)|z_{k-1}, \boldsymbol{q}_k) - \ln \Pr(q(k)). \tag{21}$$

The Viterbi algorithm detector to be derived later requires that function $d(k, \boldsymbol{q}_k)$ be expressed as a function of a *finite*-state occurring at times k and $k-1$. Let

$$\boldsymbol{Q}(k) \triangleq \text{col}\ (q(k), q(k-1), \cdots, q(k-L+1)) \tag{22}$$

where L is an arbitrary positive integer and $q(i) = 0$ for all $i \leq 0$. Because $q(k)$ can take on only two values, unity or zero, $\boldsymbol{Q}(k)$ can have 2^L possible (vector) values. Let

$$\begin{aligned} T_L = \{s|s &= \text{col}\ (a_1, a_2, \cdots, a_L), \\ &a_i = 0 \text{ or } 1, \quad i = 1, 2, \cdots, L\}. \end{aligned} \tag{23}$$

This set includes all possible vectors assumed by $\boldsymbol{Q}(k)$. The total number of elements in T_L is $M = 2L$. Because T_L is a finite set, we, therefore, refer to $\boldsymbol{Q}(k)$ as a finite-state, and we number them $s_1, s_2, \cdots, s_M$. Next, we try to express the objective function J as a function of $\boldsymbol{Q}(1)$, $\boldsymbol{Q}(2), \cdots, \boldsymbol{Q}(N)$, instead of $\boldsymbol{q}_N$.

Let θk be the following column vector (i.e., *admissible state sequence*)

$$\boldsymbol{\theta}_k = \text{col}\ (\boldsymbol{Q}(k), \boldsymbol{Q}(k-1), \cdots, \boldsymbol{Q}(1)) \tag{24}$$

in which $\boldsymbol{Q}(j)$ satisfies the following constraint. When

$$\boldsymbol{Q}(k) = \text{col}\ (a_1, a_2, \cdots, a_L) \in T_L \tag{25}$$

and

$$\boldsymbol{Q}(k-1) = \text{col}\ (b_1, b_2, \cdots, b_L) \in T_L \tag{26}$$

then

$$a_i = b_{i-1} \qquad \forall\ 2 \le i \le L. \tag{27}$$

For any $\boldsymbol{q}_N$ sequence, there exists a unique admissible state sequence $\boldsymbol{\theta}_N$ formed from the elements of $\boldsymbol{q}_N$ via (22) and (24). Additionally, for any admissible state sequence $\boldsymbol{\theta}_N$, there exists a unique $\boldsymbol{q}_N$ sequence formed by

$$q(k) = \text{first element of } \boldsymbol{Q}(k). \tag{28}$$

For example, assume that $L = 3$, $N = 5$, and $\boldsymbol{q}_N = (1, 1, 0, 1, 1)'$. Then the corresponding admissible state sequence $\boldsymbol{\theta}_N$ is

$$\boldsymbol{\theta}_N = \text{col}(\underbrace{1\ 1\ 0}_{\boldsymbol{Q}'(5)}\ \underbrace{1\ 0\ 1}_{\boldsymbol{Q}'(4)}\ \underbrace{0\ 1\ 1}_{\boldsymbol{Q}'(3)}\ \underbrace{1\ 1\ 0}_{\boldsymbol{Q}'(2)}\ \underbrace{1\ 0\ 0)}_{\boldsymbol{Q}'(1)} \tag{29}$$

Similarly, when $\boldsymbol{\theta}_N$ is given by (29) then $\boldsymbol{q}_N = (1, 1, 0, 1, 1)'$. Hereafter, we assume that $\boldsymbol{\theta}_N$ denotes an admissible state sequence.

We can therefore express (20) as

$$J(\boldsymbol{q}_N) = J(\boldsymbol{\theta}_N) = \sum_{i=1}^{N} d(k, \boldsymbol{\theta}_k) \tag{30}$$

where

$$d(k, \boldsymbol{\theta}_k) = d(k, \boldsymbol{q}_k). \tag{31}$$

Because $q(k)$ is a binary sequence, determining the globally optimal $\boldsymbol{q}_N$ requires 2^N evaluations of (30), where 2^N is an enormous number. It is infeasible, therefore, to find the globally optimal $\boldsymbol{q}_N$ (or $\boldsymbol{\theta}_N$) by this method. We are forced to find a suboptimal value for $\boldsymbol{q}_N$ (or $\boldsymbol{\theta}_N$), denoted $\hat{\boldsymbol{q}}_N$ (or $\hat{\boldsymbol{\theta}}_N$).

C. Viterbi Algorithm

Next, we propose a recursive suboptimal detector, a Viterbi Algorithm (VA) detector, whose performance is comparable to Kormylo and Mendel's SMLR detector. The SMLR detector is also a suboptimal, and is based on the same criterion. It is an iterative detector that is not self-starting. The VA detector is a noniterative detection algorithm that is self-starting.

the traditional VA is identical to finding the shortest route through a certain graph. To do this one constructs a state innovation diagram called a trellis. Fig. 2 depicts a state innovation diagram in which each node represents a distinct state at a given time, and each branch represents a transition to some new state at the next instant of time. In Fig. 2, $s_1, s_2, \cdots, s_M$ comprise the complete set of

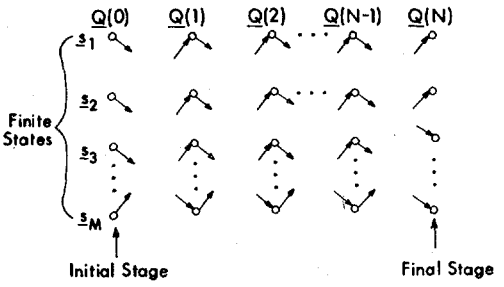

Fig. 2. State innovation diagram.

finite-states. Note that a finite-state is not necessarily a vector. Each node in Fig. 2 can have M incoming branches and outgoing branches except $\boldsymbol{Q}(0)$ and $\boldsymbol{Q}(N)$. In Fig. 2, we just show some of the branches and use two arrows at each node to indicate other incoming and outgoing branches which are not shown in the diagram. Any branch length $d_k(i, j)$ [see (35)] from s_j at time point $k-1$ to s_i at time point k must be assigned ahead of time. The VA finds the shortest path through the trellis, and the path length of the shortest path is the desired minimum value of the objective function.

Let

$$\boldsymbol{\theta}_k^i = \boldsymbol{\theta}_k|_{\boldsymbol{Q}(k)=s_i} = \begin{pmatrix} s_i \\ \boldsymbol{\theta}_{k-1} \end{pmatrix} \tag{32}$$

where $s_i \in T_L$. Note that $\boldsymbol{\theta}_k^i$ can be an arbitrary admissible state sequence $\boldsymbol{Q}(1), \boldsymbol{Q}(2), \cdots, \boldsymbol{Q}(k)$, with $\boldsymbol{Q}(k) = s_i$, and that $\boldsymbol{Q}(k) = s_i$ will put some restrictions on $\boldsymbol{Q}(k-1), \boldsymbol{Q}(k-2), \cdots, \boldsymbol{Q}(k-L+1)$ due to the constraints stated in (25) through (27).

From (30), (31), and (32) we see that for a fixed value of i,

$$J(\boldsymbol{\theta}_k^i) = J(\boldsymbol{\theta}_{k-1}) + d(k, \boldsymbol{\theta}_k^i) \tag{33}$$

and

$$\min_{\boldsymbol{\theta}_k^i} J(\boldsymbol{\theta}_k^i) = \min_{1 \le j \le M} \left\{ \min_{\boldsymbol{\theta}_{k-1}^j} [J(\boldsymbol{\theta}_{k-1}^j) + d_k(i, j)] \right\} \tag{34}$$

where

$$d_k(i, j) = d(k, \boldsymbol{\theta}_k^i)\Big|_{\boldsymbol{\theta}_k^i = \begin{pmatrix} s_i \\ \boldsymbol{\theta}_{k-1}^j \end{pmatrix}}. \tag{35}$$

$J(\boldsymbol{\theta}_k^i)$ can be thought of as the "distance" from $\boldsymbol{Q}(1)$ to $\boldsymbol{Q}(k) = s_i$, and can be expressed as the sum of the distances from $\boldsymbol{Q}(1)$ to $\boldsymbol{Q}(k-1)$, namely $J(\boldsymbol{\theta}_{k-1})$, and the distance from $\boldsymbol{Q}(k-1)$ to $\boldsymbol{Q}(k) = s_i$, namely $d(k, \boldsymbol{\theta}_k^i)$. The minimum value of $J(\boldsymbol{\theta}_k^i)$ can be obtained by finding the minimum distances from $\boldsymbol{Q}(1)$ to $\boldsymbol{Q}(k-1) = s_j$ and then from s_j to $\boldsymbol{Q}(k) = s_i$ (for all $1 \le j \le M$), and, then finding the minimum value among these M values. Fig. 3 depicts (33) and (34) by showing $J(\boldsymbol{\theta}_k^j)$ associated with $\boldsymbol{Q}(k-1) = s_j$ $(1 \le j \le M)$ and $d_k(i, j)$ (which is the metric from state s_j to s_i), from time point $k-1$ to k. The main recursive equation of the VA is derived from (34).

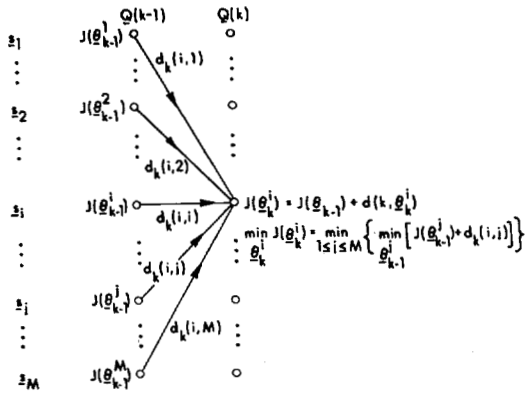

Fig. 3. Branches from the state $Q(k-1)$ to the state $Q(k)$.

We discuss two cases next. The first is the traditional VA case in which $d_k(i, j)$ depends only on $Q(k) = s_i$ and $Q(k-1) = s_j$. In this case, the VA is optimal, i.e., it can find min $\{J(Q_N)\}$. The second case is one in which $d_k(i, j)$ not only depends upon $Q(k) = s_i$ and $Q(k-1) = s_j$ but also upon $\theta_{k-2} = \text{col}\,(Q(k-2), Q(k-3), \cdots, Q(1))$. In this case the VA is suboptimal.

Returning to the first case, we easily see that

$$\min_{\theta^j_{k-1}} [J(\theta^j_{k-1}) + d_k(i, j)] = \min_{\theta^j_{k-1}} [J(\theta^j_{k-1})] + d_k(i, j) \tag{36}$$

because $d_k(i, j)$ does not depend on θ^j_{k-1} when j is fixed. In this case (34) can be simplified to the following equation:

$$\Gamma_{i,k} = \min_{1 \le j \le M} \Gamma_k(i, j) \tag{37}$$

where

$$\Gamma_k(i, j) = \Gamma_{j,k-1} + d_k(i, j) \tag{38}$$

and

$$\Gamma_{i,k} = \min_{\theta^i_k} \{J(\theta^i_k)\}. \tag{39}$$

which implies

$$\Gamma_{j,k-1} = \min_{\theta^j_{k-1}} \{J(\theta^j_{k-1})\}.$$

Fig. 4 depicts some branches and associated branch lengths from the state $Q(k-1)$ to the state $Q(k)$. It also includes $\Gamma_{i,k-1}$ and $\Gamma_{i,k}$ at each mode. Value $\Gamma_{i,k}$ is obtained by scanning the entire state space at time $k-1$. For example, $\Gamma_{1,k}$ is obtained by computing $\{\Gamma_{1,k-1} + d_k(1, 1)\}, \{\Gamma_{2,k-1} + d_k(1, 2)\}, \cdots, \{\Gamma_{M,k-1} + d_k(1, M)\}$ and choosing the smallest of these values.

Finally, by recursively using (37) until $k = N$, we have that

$$\min_{\theta_N} \{J(\theta_N)\} = \min_{1 \le i \le M} \{\min_{\theta^i_N} J(\theta^i_N)\} = \min_{1 \le i \le M} \Gamma_{i,N}. \tag{40}$$

Now we return to the second case which, as it turns out,

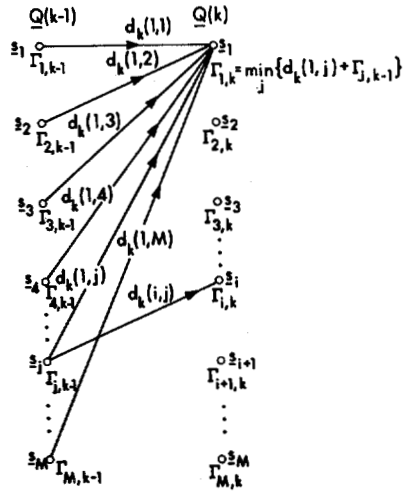

Fig. 4. Branches from the state $Q(k-1)$ to the state $Q(k)$.

is our situation. Recall that in this case $d_k(i, j)$ depends on $Q(1), Q(2), \cdots, Q(k-1) = s_j, Q(k) = s_i$. In other words, the branch length from $Q(k-1) = s_j$ to $Q(k) = s_i$ is different for different sequences $Q(1), Q(2), \cdots, Q(k-2)$. The beautiful recursive equations (37) and (38) are no longer true. Finding min $J(\theta_k)$ is independent of finding min $J(\theta^j_{k-1})$ because we have to compute $J(\theta^i_k)$ for *all* possible θ^i_k sequences even if min $J(\theta^j_{k-1})$ has been obtained. Next, we derive a suboptimal VA for this case.

Assume that we have obtained $\hat{\theta}^i_{k-1}$ and $\Gamma_{i,k-1}$ at time point $k-1$, for all $1 \le i \le M$, where $\hat{\theta}^i_{k-1}$ is an estimate of θ^i_{k-1} and

$$\Gamma_{i,k-1} = J(\hat{\theta}^i_{k-1}). \tag{41}$$

For simplicity, we use the same notation here, namely $\Gamma_{i,k-1}$, as in (39). Note, although that for different cases, this notation can have different meanings. For a known value of $\hat{\theta}^j_{k-1}$, $1 \le j \le M$, we determine $\hat{\theta}^i_k$, $1 \le i \le M$, such that $\Gamma_{i,k} = J(\hat{\theta}_k)$ is minimal just over $\hat{\theta}_{k-1}$, $1 \le j \le M$, instead of over all possible $Q_1, Q_2, \cdots, Q_k$ $(=s_i)$. In other words,

$$\Gamma_{i,k} = J(\hat{\theta}^i_k) = \min_{A_{i,k}} J(\theta_k) \tag{42}$$

where

$$A_{ik} = \{\theta^i_k | \theta_k = \text{col}\,(Q(k) = s_i, \hat{\theta}^j_{k-1}), \quad 1 \le j \le M\}. \tag{43}$$

Therefore, from (42) we have

$$\Gamma_{i,k} = \min_{A_{i,k}} J(\theta_k) = \min_{1 \le j \le M} \{\Gamma_{j,k-1} + d_k(i, j)\} = \min_{1 \le j \le M} \Gamma_k(i, j) \tag{44}$$

where $d_k(i, j)$ is now defined as

$$d_k(i, j) = d(k, \theta^i_k)|_{\theta^i_k = \text{col}(s_i, \hat{\theta}^j_{k-1})}. \tag{45}$$

Note that $d_k(i, j)$ depends only on $Q(k) = s_i$ and a particular state sequence $\hat{\theta}^j_{k-1}$ with $Q(k-1) = s_j$, which is

already obtained at the time point $k-1$. Therefore, $d_k(i, j)$ now depends only on $\boldsymbol{Q}(k) = s_i$ and $\boldsymbol{Q}(k-1) = s_j$. Observe that (44) is exactly the same as (37). Although the same recursive equation is obtained, $d_k(i, j)$ in this case cannot be computed ahead of time. It can be computed only when $\hat{\boldsymbol{\theta}}^j_{k-1}$ is obtained. The suboptimal estimate $\hat{\boldsymbol{\theta}}_N$, is associated with

$$J(\hat{\boldsymbol{\theta}}_N) = \min_{1 \le i \le M} \{\Gamma_{i,N}\}. \tag{46}$$

Next, we compute $d_k(i, j)$ for all $\le i, j \le M$ when $\hat{\theta}^j_{k-1}$ is available. We choose $d_k(i, j)$, as

$$d_k(i, j) = \begin{cases} d(k, \boldsymbol{\theta}_k = \text{col}\ (s_i, \hat{\theta}^j_{k-1})) \\ \quad \text{if} \quad \boldsymbol{Q}(k) = s_i \quad \text{and} \quad \boldsymbol{Q}(k-1) = s_j \\ \quad \text{satisfying (25)–(27)} \\ \infty \quad \text{otherwise.} \end{cases} \tag{47}$$

Note that $d_k(i, j) = \infty$ occurs when s_i and s_j do not satisfy (25)–(27), i.e., when it is impossible to have a transition from $\boldsymbol{Q}(k-1) = s_j$ to $\boldsymbol{Q}(k) = s_i$, because $\hat{\boldsymbol{\theta}}_k = \text{col}\ (s_i, \hat{\boldsymbol{\theta}}^j_{k-1})$ is not an admissible state sequence. We assume that

$$s_i = \text{col}\ (a_1, b_1, b_2, \cdots, b_{L-1}) \tag{48}$$

and

$$s_j = \text{col}\ (b_1, b_2, \cdots, b_L) \tag{49}$$

where $s_i, s_j \in T_L$. From (21), (31), and (47) we see that

$$d_k(i, j) = -\ln p(z(k)|z_{k-1}, \boldsymbol{Q}(k) = s_i, \boldsymbol{\theta}_{k-1} = \hat{\boldsymbol{\theta}}^j_{k-1}) - \ln \Pr(q(k) = a_1). \tag{50}$$

Because

$$\Pr\ (q(k) = a_1) = \begin{cases} \lambda & \text{if} \quad a_1 = 1 \\ 1 - \lambda & \text{if} \quad a_1 = 0 \end{cases} \tag{51}$$

then

$$\Pr\ (q(k) = a_1) = \lambda^{a_1}(1-\lambda)^{1-a_1}, \tag{52}$$

thus, $d_k(i, j)$ can be written as

$$d_k(i, j) = -\ln p(z(k)|z_{k-1}, \boldsymbol{Q}(k) = s_i, \boldsymbol{\theta}_{k-1} = \hat{\boldsymbol{\theta}}^j_{k-1}) - \ln \lambda^{a_1}(1-\lambda)^{a_1}. \tag{53}$$

Observing (53), we see that we need only compute $p(z(k)|z_{k-1}, \boldsymbol{Q}(k) = s_i, \boldsymbol{\theta}_{k-1} = \hat{\boldsymbol{\theta}}^j_{k-1})$ in order to compute $d_k(i, j)$.

From (1) and (2), we know that $p(z(k)|z_{k-1}, \boldsymbol{Q}(k) = s_i, \boldsymbol{\theta}_{k-1} = \hat{\boldsymbol{\theta}}^j_{k-1})$ is a Gaussian density function because, when $\boldsymbol{Q}_k$ (or $\boldsymbol{q}_k$) is given, the $z(k)$ is a linear combination of Gaussian random variables $r(k), r(k-1), \cdots, r(1), n(k), n(k-1), \cdots, n(1)$. Let

$$\hat{z}_{ij}(k|k-1) = E[z(k)|z_{k-1}, \boldsymbol{Q}(k) = s_i, \boldsymbol{\theta}_{k-1} = \hat{\boldsymbol{\theta}}^j_{k-1}] \tag{54}$$

$$\tilde{z}_{ij}(k|k-1) = z(k) - \hat{z}_{ij}(k|k-1) \tag{55}$$

and

$$\eta_{ij}(k|k-1) = E[\tilde{z}^2_{ij}(k|k-1)|\boldsymbol{Q}(k) = s_i, \boldsymbol{\theta}_{k-1} = \hat{\boldsymbol{\theta}}^j_{k-1}] \tag{56}$$

for all s_i and $s_j \in T_L$ which satisfy (48) and (49). Then [11]

$$p(z(k)|z_{k-1}, \boldsymbol{Q}(k) = s_i, \boldsymbol{\theta}_{k-1} = \hat{\boldsymbol{\theta}}^j_{k-1}) = \frac{1}{\sqrt{2\pi\eta_{ij}(k|k-1)}} \exp\left\{-\frac{\tilde{z}^2_{ij}(k|k-1)}{2\eta_{ij}(k|k-1)}\right\} \tag{57}$$

so that

$$-\ln p(z(k)|z_{k-1}, \boldsymbol{Q}(k) = s_i, \boldsymbol{\theta}_{k-1} = \hat{\boldsymbol{\theta}}^j_{k-1}) = \frac{1}{2}\left\{\ln 2\pi + \ln \eta_{ij}(k|k-1) + \frac{\tilde{z}^2_{ij}(k|k-1)}{\eta_{ij}(k|k-1)}\right\}. \tag{58}$$

Because $\ln 2\pi$ is a constant, it can be neglected from the objective function. Additionally, in order to save some computations we multiply the objective function by 2 because doing this does not affect its optimality.

Signals $\tilde{z}_{ij}(k|k-1)$, $\eta_{ij}(k|k-1)$, and consequently $d_k(i, j)$ can be obtained from the following Kalman filter equations (e.g., [10] and [11])

$$\hat{\boldsymbol{x}}_{ij}(k|k-1) = \Phi\hat{\boldsymbol{x}}_i(k-1)|k-1), \tag{59}$$

$$P_{ij}(k|k-1) = \Phi P_j(k-1)|k-1)\Phi + \boldsymbol{\gamma} Ca_1 \boldsymbol{\gamma}', \tag{60}$$

$$\hat{}_{ij}(k|k) = \hat{\boldsymbol{x}}_{ij}(k|k-1) + K_{ij}(k)\tilde{z}_{ij}(k|k-1), \tag{61}$$

$$\tilde{z}_{ij}(k|k-1) = z(k) - \boldsymbol{h}'\hat{\boldsymbol{x}}_{ij}(k|k-1), \tag{62}$$

$$K_{ij}(k) = P_{ij}(k|k-1)\boldsymbol{h}'\eta^{-1}_{ij}(k|k-1), \tag{63}$$

$$P_{ij}(k|k) = [I - K_{ij}(k)\boldsymbol{h}']P_{ij}(k|k-1) \tag{64}$$

and

$$\eta_{ij}(k|k-1) = \boldsymbol{h}'P_{ij}(k|k-1)\boldsymbol{h} + R \tag{65}$$

where

$$\hat{\boldsymbol{x}}_j(k|k) = E[\boldsymbol{x}(k)|z_k, \boldsymbol{\theta}_k = \hat{\boldsymbol{\theta}}^j_k] \tag{66}$$

is a filtered estimate of $\boldsymbol{x}(k)$, and

$$P_j(k|k) = E\{[\boldsymbol{x}(k) - \hat{\boldsymbol{x}}_j(k|k)]\ [\boldsymbol{x}(k) - \hat{\boldsymbol{x}}_j(k|k)]'|\ \boldsymbol{\theta}_k = \hat{\boldsymbol{\theta}}^j_k] \tag{67}$$

is its associated error-covariance matrix, for the state-space model

$$\boldsymbol{x}(k) = \Phi\boldsymbol{x}(k-1) + \boldsymbol{\gamma}q(k)\ r(k) \tag{68}$$

$$z(k) = \boldsymbol{h}'\boldsymbol{x}(k) + n(k). \tag{69}$$

Observe that $\hat{\boldsymbol{x}}_j(k|k)$ and $P_j(k|k)$ are conditioned not only upon all measurements z_k, but also upon $\hat{\boldsymbol{\theta}}^j_k$.

The VA for finding $\min_i \{\Gamma_{i,N}\}$ is summarized next in a three-step procedure.

Step 1—Initialization:

$$\hat{\boldsymbol{\theta}}_0^i = \begin{cases} s_r & \text{for } i = r \\ \text{arbitrary state} & i \neq r \end{cases}. \tag{70}$$

$$\Gamma_{i,0} = \begin{cases} 0 & \text{for } i = r \\ \infty & \text{for } i \neq r. \end{cases} \tag{71}$$

Set $k = 1$ and store given values of $\hat{x}_r(0|0)$ and $P_r(0|0)$. Note that s_r can be thought of as the initial state of a forward dynamic programming procedure. In seismic applications we usually choose $s_r = \text{col } (0, 0, \cdots, 0)$.

Step 2—Recursion: Compute $d_k(i, j)$ and

$$\Gamma_k(i, j) = \Gamma_{j,k-1} + d_k(i, j) \quad \text{for all } i, j = 1, 2, \cdots, M. \tag{72}$$

Then determine $\Gamma_{i,k}$, $\hat{x}_i(k|k)$ and $P_i(k|k)$ for all $1 \leq i \leq M$ as follows:

$$\Gamma_{i,k} = \min_j \{\Gamma_k(i, j)\} \triangleq \Gamma_k(i, l) \tag{73}$$

$$\hat{x}_i(k|k) = \hat{x}_{i,l}(k|k) \tag{74}$$

$$P_i(k|k) = P_{i,l}(k|k). \tag{75}$$

Store $\hat{x}_i(k|k)$, $P_i(k|k)$, $\Gamma_{i,k}$ and the associated $\hat{\boldsymbol{\theta}}_k^i$ for all $\leq i \leq M$. Set k equal to $k + 1$ and repeat the above procedure until $k = N$.

Step 3: Choose the minimum value of $\Gamma_{i,N}$ for $1 \leq i \leq M$ as the desired objective function, which we designate J^*. Assume that

$$J^* = \min_i \{\Gamma_{i,N}\} = \Gamma_{m,N} \tag{76}$$

then, the desired suboptimum $\hat{\boldsymbol{\theta}}_N$ is $\hat{\boldsymbol{\theta}}_N^m$, which is associated with $\Gamma_{m,N}$, i.e.,

$$\Gamma_{m,N} = J(\hat{\boldsymbol{\theta}}_N^m) = J^* \tag{77}$$

so that

$$\hat{\boldsymbol{\theta}}_N = \text{col } (\hat{\boldsymbol{Q}}(N), \hat{\boldsymbol{Q}}(N - 1), \cdots, \hat{\boldsymbol{Q}}(1)) = \hat{\boldsymbol{\theta}}_N^m. \tag{78}$$

The suboptimal estimate of $q(k)$, $\hat{q}(k)$, is just the first element of $\hat{\boldsymbol{Q}}(k)$ [see (28)].

The VA, as given by our three-step procedure, simultaneously sets up the state innovation diagram and performs forward dynamic programming. Our application is not one in which it is possible to set up a complete state innovation diagram ahead of time because the branch length computations of $d_k(i, j)$ require knowledge of $\hat{x}_j(k - 1|k - 1)$ and $P_j(k - 1|k - 1)$ $\forall$ $1 \leq j \leq M$, and, these quantities only become available at stage $k - 1$.

Total computation for our three-step procedure is approximately equivalent to $2M$ Kalman filters which operate on N observations, plus $4NM$ multiplications for computing multiplications in (58), $6NM$ additions for computing the additions in (53), (58), and (72), and $2NM$ ln function evaluations in (58). If the VA is implemented using parallel processing, then the total computations will be reduced by a factor of M. In this case, we only require two Kalman filters which operate on N observations, plus $4N$ multiplications, $6N$ additions, and $2N$ ln function evaluations; but, we need M processors, where $M = 2^L$.

As parameter L, which determines the dimension of the trellis and the total computations, is chosen larger, then $\Gamma_{m,N}$ becomes smaller, i.e., the suboptimum $\hat{\theta}_N$ becomes better. We have found that when L is chosen equal to n (the order of the ARMA wavelet), the detected $\hat{\boldsymbol{q}}_N$ is comparable to that obtained from the SMLR detector (Kormylo and Mendel [12]). At present, we are limited to relatively small values of L; but in the future, it should be possible to perform many of the calculations in parallel, thereby opening the door to larger values of L.

D. Amplitude Estimation

Once $q(k)$ has been detected, we must estimate the amplitudes of the detected spikes. Let

$$\boldsymbol{r} = \text{col } [r(1), r(2), \cdots, r(N)]. \tag{79}$$

When $\boldsymbol{q}_N$ is known, z_N and $\boldsymbol{r}$ will be jointly Gaussian; hence, the maximum-likelihood and minimum-variance estimates of $\boldsymbol{r}$ are the same, namely

$$\hat{\boldsymbol{r}} = E[\boldsymbol{r}|z_N, \boldsymbol{q}_N]. \tag{80}$$

By combining the estimate of $q(k)$ into the state variable model (68) and (69), i.e.,

$$\boldsymbol{x}(k) = \Phi \boldsymbol{x}(k - 1) + \Upsilon \hat{q}(k)\, r(k) \tag{81}$$

and

$$z(k) = \boldsymbol{h}'\boldsymbol{x}(k) + n(k) \tag{82}$$

we can obtain $\hat{\boldsymbol{r}}$ using minimum-variance deconvolution formulas [4], [13].

IV. Computer Simulations

In the simulations described below we generated a Bernoulli–Gaussian sequence $\mu(k)$, convolved it with a known wavelet $V(k)$, added white noise $n(k)$ to the result to obtain noisy measurements $z(k)$, determined $q(k)$ using the VA and SMLR detectors, and, finally, used minimum-variance deconvolution to estimate spike amplitudes.

In our first example we used the fourth-order ARMA wavelet depicted in Fig. 5(a). Noisy data (signal-to-noise ratio equal to 10) are depicted in Fig. 5(b). Fig. 5(c) depicts results obtained from the VA detector for $L = 1$ or 2 (the same results were obtained for both values of L). Fig. 5(d) depicts similar results for $L = 3$, whereas Fig. 5(e) depicts them for $L = 4$ and 5. Fig. 5(f) depicts the results obtained from the SMLR detector (starting with $\boldsymbol{q}_N = \boldsymbol{O}$). Comparing Fig. 5(c)–(e), we see that performance of the VA detector improves as L increases. Additionally, comparing Fig. 5(e) and (f), we see that the performances of the VA SMLR detectors are almost the same, when parameter L is set equal to the wavelet order.

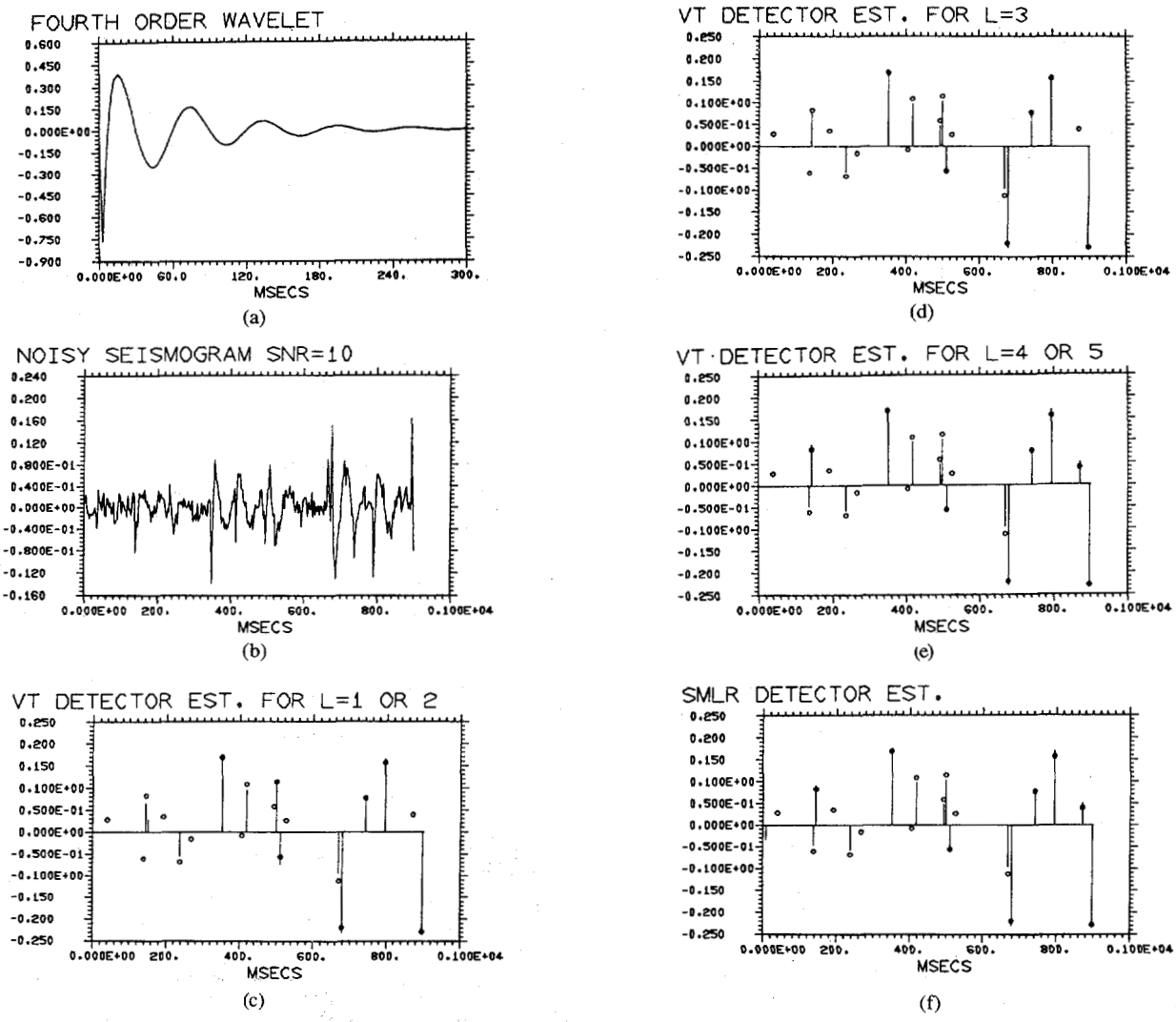

Fig. 5. (a) Fourth-order ARMA wavelets (sampling time $T = 3$ ms). (b) Synthetic noisy data; VA detector estimates for: (c) $L = 1$, or 2, (d) $L = 3$, and (e) $L = 4$ or 5. (f) SMLR detector estimates with starting $\boldsymbol{q}_N = \boldsymbol{O}$. Circles depict true spikes and bars depict estimates.

In our second example we used the fourth-order ARMA wavelet depicted in Fig. 6(a). Noisy data (signal-to-noise ratio equaled 10) are depicted in Fig. 6(b). Fig. 6(c) depicts the results obtained from the VA detector for $L = 4$, and Fig. 6(d) depicts similar results obtained from the SMLR detector (starting with $\boldsymbol{q}_N = \boldsymbol{O}$). Once again, we see that the performances of the two detectors are almost the same.

In both examples, as well as others performed by the authors, all missing spikes (i.e., missed detections) have very small amplitudes. Both the VA and SMLR detectors detect spikes of significant amplitude.

V. Conclusions

A VA detector has been presented as a method for locating randomly spaced spikes that have Gaussian amplitudes. In order to apply this detector, one must know the source wavelet and noise statistics, or, at least have estimates of them. Although the VA detector is suboptimum, simulation results show that it works well. Its performance is proportional to the parameter L, which determines the size of the state innovation diagram. The VA detector will become more feasible for larger value of L when, in the not-so-distant future, parallel processing becomes more feasible. In other words, the computational requirements for larger values of L can be handled when parallel processing becomes available.

If the source wavelet and/or noise parameters are unknown, then these quantities must be estimated. The combined estimation of these quantities and detection of $q(k)$ can be performed using a block component method, analogous to the one described in Kormylo [3], Mendel [5], and Chi *et al.* [15], in which the VA detector is used instead of other detectors, such as the SMLR detector.

Acknowledgment

The work reported on in this paper was performed at the University of Southern California, Los Angeles, CA.

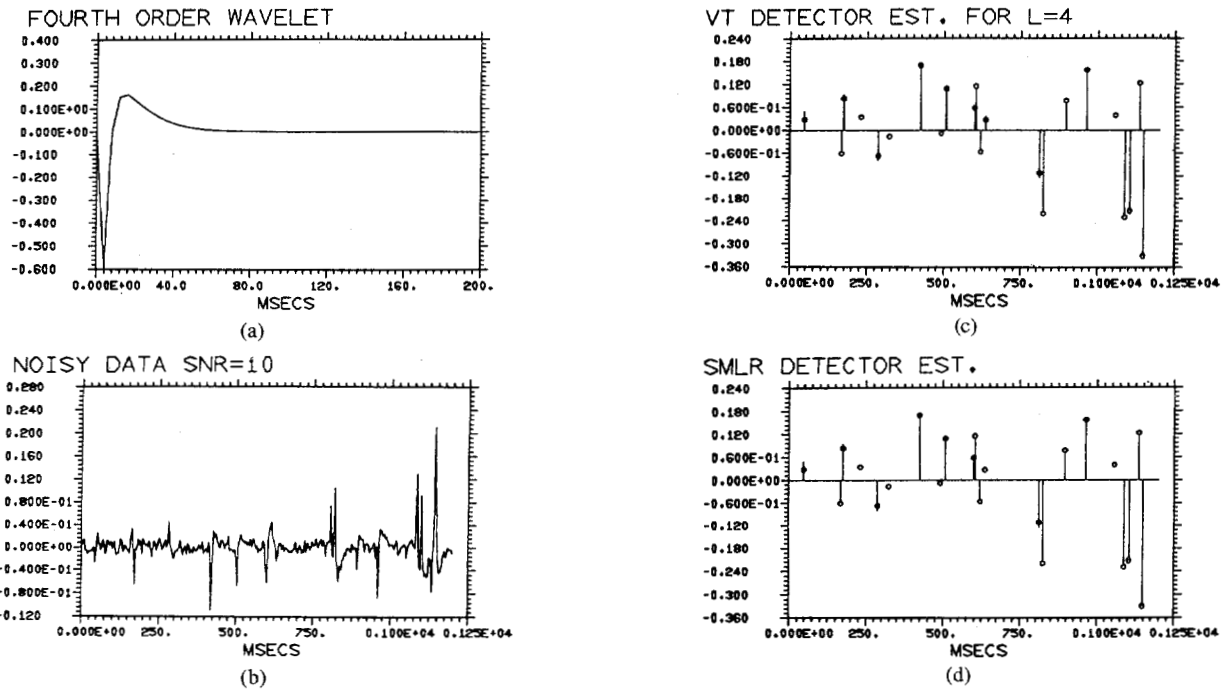

Fig. 6. (a) Fourth-order wavelet (sampling time $T = 4$ ms). (b) Synthetic noisy data. (c) VA detector estimates for $L = 4$. (d) SMLR detector estimates with starting $q_N = O$. Circles depict true spikes and bars depict estimates.

References

[1] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[2] R. J. Godfrey, "A stochastic model for seismogram analysis," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1979.

[3] J. J. Kormylo, "Maximum-likelihood seismic deconvolution," Ph.D. dissertation, Univ. Southern California, Los Angeles, 1979.

[4] J. M. Mendel, "White-noise estimators for seismic data processing in oil exploration," *IEEE Trans. Automat. Contr.*, vol. AC-22, no. 5, pp. 694–676, 1977.

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

[6] J. K. Omura, "On the Virterbi decoding altorithm," *IEEE Trans. Inform Theory*, vol. IT-15, pp. 177–179, 1969.

[7] A. J. Virterbi and J. P. Odenwalter, "Further results on optimal decoding of convolutional codes," *IEEE Trans. Inform. Theory,* vol. IT-15, pp. 732–734, 1969.

[8] E. A. Robinson, "Predictive decomposition of seismic traces," *Geophys.*, vol. 22, pp. 767–788, 1957.

[9] E. A. Robinson, "Predictive decomposition of time series with application to seismic exploration," *Geophys.*, vol. 32, pp. 418–484, 1967.

[10] J. S. Meditch, *Stochastic Optimal Linear Estimation and Control*. New York: McGraw-Hill, 1969.

[11] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[12] J. J. Kormylo and J. M. Mendel, "Maximum-likelihood detection and estimation of Bernoulli–Gaussian processes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 482–488, 1982.

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

[14] R. Godfrey, F. Muir, and F. Rocca, "Modeling seismic impedance with Markov chains," *Geophys.*, vol. 45, no. 9, pp. 1351–1372, 1980.

[15] C. Y. Chi, J. M. Mendel, and D. Hampson, "A computationally-fast approach to maximum-likelihood deconvolution," *Geophys.*, May 1984.

Chong-Yung Chi (S'83–M'83) was born in Taiwan, Republic of China, on August 7, 1952. He received the B.S. degree from the Tatung Institute of Technology, Taipei, Taiwan, in 1975, the M.S. degree from the National Taiwan University, Taipei, in 1977, and the Ph.D. degree from the University of Southern California, Los Angeles, in 1983, all in electrical engineering.

In September 1979 he was appointed a Teaching/Research Assistant, and later became a Research Assistant in the Department of Electrical Engineering-Systems, University of Southern California. Since July 1983 he has been a member of the Technical Staff in the Radar Science and Engineering Section at the Jet Propulsion Laboratory, Pasadena, CA. Currently, he is engaged in the studies of spaceborne radar scatterometer systems and radar image processing. His research interests include digital signal processing, system identification, and estimation theory.

Jerry M. Mendel (S'59–M'61–SM'72–F'78) received the B.S. degree in mechanical engineering and the M.S. and Ph.D. degrees in electrical engineering from the Polytechnic Institute of Brooklyn, Brooklyn, NY, in 1959, 1960, and 1963, respectively.

His experience has included teaching courses in the electrical engineering at the Polytechynic Institute of Brooklyn, from 1960 to 1963, and has also included various consulting positions. From July 1963 to January 1974 he was with McDonnell Douglas Astronautics Company on a full-time basis. Currently he is Processor and Chairman of Electrical Engineering-Systems at the University of Southern California, Los Angeles. He was Associate Chairman from 1982 to 1984. He teaches courses in estimation theory and seismic data processing for oil exploration, and was Director of the U.S.C. Geo-Signal Processing Program (1980–1983). He has published over 150 technical papers and is author of the monograph *Optimal Seismic Deconvolution: An Estimation-Based Approach* (New York: Academic, 1983), the text *Discrete Techniques of Parameter Estimation: The Equation Error Formulation* (New York: Marcel Dekker, 1973), and coeditor (with K. S. Fu of Purdue University) of *Adaptive, Learning and Pattern Recognition Systems* (New York: Academic, 1970).

Dr. Mendel served as Editor of the IEEE Control Systems Society's IEEE

TRANSACTIONS ON AUTOMATIC CONTROL, is Consulting Editor of the Control and Systems Theory Series for Marcel Dekker, Inc., and is Associate Editor of *Automatica* and the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He received the SEG 1976 Outstanding Presentation Award for a paper on the application of Kalman filtering to deconvolution. He is a Distinguished Member of the IEEE Control Systems Society, an elected member of the Administrative Committee and Vice-President for Technical Activities of the IEEE Control Systems Society, a member of the IEEE Geoscience and Remote Sensing Society, the Society of Exploration Geophysics, the European Association for Exploration Geophysicists, Tau Beta Pi, and Pi Tau Sigma, and a Registered Professional Control Systems Engineer in California.