Optimal Seismic Deconvolution: Distributed Algorithms

Konstantinos N. Plataniotis, *Member, IEEE*, Sokratis K. Katsikas, *Associate Member, IEEE*, Demetrios G. Lainiotis, and Anastasios N. Venetsanopoulos, *Fellow, IEEE*

***Abstract*—Deconvolution is one of the most important aspects of seismic signal processing. The objective of the deconvolution procedure is to remove the obscuring effect of the wavelet's replica making up the seismic trace and therefore obtain an estimate of the reflection coefficient sequence. This paper introduces a new deconvolution algorithm. Optimal distributed estimators and smoothers are utilized in the proposed solution. The new distributed methodology, perfectly suitable for a multisensor environment, such as the seismic signal processing, is compared to the centralized approach, with respect to computational complexity and architectural efficiency. It is shown that the distributed approach greatly outperforms the currently used centralized methodology offering flexibility in the design of the data fusion network.**

***Index Terms*— Data fusion, deconvolution, distributed filters, filtering, parallel algorithms, smoothing.**

I. Introduction

DECONVOLUTION is a fundamental aspect of seismic signal processing. The objective of the deconvolution procedure is to remove the obscuring effect of the wavelet's replica making up the seismic trace and consequently obtain an estimate of the reflection coefficient sequence. Thus, through the determination of the coefficient sequence, important conclusions concerning subsurface geometry and lithography can be derived.

The most commonly used model is the convolution summation introduced by Robinson [1]

$$z(k) = V_R(k) + n(k) = \sum_{j=1}^{k} \mu(j) V^{+}(k-j) + n(k) \qquad (1)$$

where $z(k)$ is the received seismic trace, $V_R(k)$ is the noise-free seismic trace resulting from a superposition of wavelet replica reflected from the interfaces of earth's subsurface layers, $n(k)$ is the measurement noise, $V^{+}(k-j)$ is a sequence associated with the basic seismic wavelet, and $\mu(j)$ is the earth's reflectivity function.

Many procedures exist for the solution of the seismic deconvolution problem. The state-space approach [2]–[10] alleviated disadvantages associated with previously popular methods, whose applicability depends heavily on underlying assumptions, which unfortunately are not usually valid [10]. Following the work in [2], [8], we assume that the starting point for the deconvolution process is a state-space equivalent description of (1) given by

$$x(k+1) = \Phi x(k) + \gamma \mu(k) \qquad (2)$$

$$z(k+1) = Hx(k+1) + v(k+1) \qquad (3)$$

where x and γ are $(n \times 1)$ vectors, Φ is the $(n \times n)$ state transition matrix, H is the $(m \times n)$ output matrix, $z(k)$ is the m-dimensional measurement vector $Hx(k) = V_R(k)$, and $\mu(k)$ and $v(k)$ are uncorrelated, zero-mean sequences. The plant and measurement noise covariances are q and R, respectively. The initial value $x(0)$ of the state vector $x(k)$ at time $t_0 = 0$ is modeled as Gaussian random variable with mean $\hat{x}(0 \mid 0)$ and variance $P(0 \mid 0)$, and it is assumed independent of both $\mu(k)$ and $v(k)$.

Usually, the measurement dimensionality is higher than the state dimensionality, since in case of $n > m$, the measurements provide only partial information about the system state. In geophysical applications, the number of sensors (geophones) used to capture the seismic reflection varies from a few hundred to several thousand. To reduce uncertainty and to obtain complete knowledge of the state of nature, the sensors are usually allocated in different local subsystems (geophone clusters). Notice that the measurement vector $z(k)$ is assumed to be of dimension $(m \times 1)$ in contrast to (1). If we wish to process all geophone measurements simultaneously, m represents the total number of geophones used to capture the seismic trace [10]. Since, however, geophones in practice are used in clusters, a distributed algorithm that allows local measurements to be processed near the sensing devices is preferable. Under such a distributed scenario, the observation vector $[z(k)]$ is partitioned into r subvectors, each of them corresponding to a local sensor system (geophone cluster), with local dimensionality m_i [16], [17]

$$z^{\tau}(k) = \left[z_1^{\tau}(k), z_2^{\tau}(k), \ldots, z_r^{\tau}(k)\right] \qquad (4)$$

$$H^{\tau}(k) = \left[H_1^{\tau}, H_2^{\tau}, \ldots, H_r^{\tau}\right] \qquad (5)$$

$$v^{\tau}(k) = \left[v_1^{\tau}(k), v_2^{\tau}(k), \ldots, v_r^{\tau}(k)\right]. \qquad (6)$$

Manuscript received February 7, 1996; revised December 9, 1996.

K. N. Plataniotis is with the Department of Mathematics, Physics, and Computer Science, Ryerson Polytechnic University, Toronto, Ont., Canada, M5B 2K3 (e-mail: kplatani@acs.ryerson.ca).

S. K. Katsikas is with the Department of Mathematics, University of the Aegean, Karlovassi, 83200 Greece (e-mail: ska@aegean.ariadne-t.gr).

D. G. Lainiotis is with Intelligent Systems Technology, Melbourne Beach, FL 32951 USA.

A. N. Venetsanopoulos is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ont., Canada M5S 3G4 (e-mail: anv@dsp.toronto.edu).

Publisher Item Identifier S 0196-2892(98)01139-5.

Consequently, the local measurement equation, based on the state vector is given by

$$z_i(k+1) = H_i x(k+1) + v_i(k+1) \tag{7}$$

where $z_i(k+1)$ is the m_i-dimensional measurement vector at the local processor system i, H_i is the $(m_i \times n)$ local output matrix, and $v_i(k+1)$ is the m_i local measurement noise vector. In this analysis, it is assumed that

$$m = \sum_{i=1}^{r} m_i. \tag{8}$$

The model described by (2) and (3) or by (2) and (7) is linear and time invariant. However, the authors would like to emphasize the point that the results obtained herein are immediately applicable to linear time varying, linear time invariant, and periodic models, as well as to nonlinear models.

For the state-space model described in (1)–(3), Katsikas has developed a *Minimum Variance Deconvolution* formula, which utilizes a centralized filtering/smoothing scheme to obtain optimal estimates of the reflection coefficient sequence [7]–[10]. Assuming that $\hat{x}(k \mid j)$ denote the minimum variance estimator of the state $x(k)$, given all available measurements, the following recursive deconvolution formula for a general time-varying linear state-space model holds [8]:

$$\begin{aligned}\hat{\mu}(k+1 \mid k+l) \\ &= q(k)\gamma^{\tau}(k)[\Phi(k+1,k)P(k+1 \mid k+1)\Phi^{\tau}(k+1,k) \\ &\qquad +\gamma(k)q(k)\gamma^{\tau}(k)]^{-1} \\ &\quad \times [\hat{x}(k+1 \mid k+l) - \Phi(k+1,k)\hat{x}(k \mid k)]\end{aligned} \tag{9}$$

where $P(k \mid k) = E[(x(k) - \hat{x}(k \mid k))(x(k) - \hat{x}(k \mid k))^{\tau}]$. The above formula has been used to obtain fixed point and fixed lag estimates of the sequence $\mu(k)$ by fixing k and l, respectively. It can also be used to obtain fixed interval estimates of $\mu(k)$ by letting $k+l = N$, where N is the last available data point. Katsikas used a centralized linear estimator, the (centralized) linear Lainiotis filter [7], [8] in conjunction with the (centralized) Lainiotis smoother [11], [12] to obtain the estimates in (9). Taking advantage of the particular structure of the centralized smoother, Katsikas implemented the filter and the smoother in parallel, combining their estimates at the end of the processing cycle through (9) to obtain the requested estimate $\hat{\mu}(k+1 \mid k+l)$.

Due to space limitation, we will not repeat here the approach discussed in [8]. Refer to [7] and [8] for the centralized approach to seismic deconvolution and to the tutorial paper [13] or the monograph [15] for a detailed presentation of the partitioning approach.

If the approach discussed in [8] has to be utilized to provide the requested $\hat{\mu}(k+1 \mid k+l)$ in a multisensor environment, such the one described in (4)–(8), the central processing facility has to handle all measurements and interim results and has to interpret the results. However, the implementation of such an approach is often problematic in practice. Indeed, the centralized algorithm of [8] imposes severe computational burden on the computing facility. Depending on the complexity of the model used and the number of sensors utilized to capture the seismic trace, the number of computations required is rather extensive. For the time-invariant, linear state-space model described in (2)–(3), the complexity is of order $O(\max(n^3, m^3))$ [7]. In addition, the large number of sensors (geophones) required in seismic deconvolution for oil exploration introduces considerable communication overhead, bandwidth waste, and time delays, since all measurements have to be transferred from the (possibly) remote-sensing locations to the central facility for processing. Centralized implementation means that the entire processing system depends on the single computing facility. Since the availability of the processor is critical to the operation, backup systems are often used in practice, resulting in increased cost and computing power waste. Finally, the centralized approach is inflexible to the expansion or to the modification of the existing sensor technology since utilization of additional sensors or reconfiguration of the existing geophones array, something common in practice, requires the modification of the processing algorithm.

To overcome these problems and to achieve robustness and redundancy, a new distributed approach is necessary. The main objective is the development of a data processing system, which provides redundancy and scalability, increases robustness, and achieves improved performance in terms of computational speed. A distributed minimum variance deconvolution algorithm that can achieve all of the above is presented in this paper. Section II introduces the new distributed formulation of the optimal minimum variance deconvolution algorithm. Motivation, design characteristics, and implementation issues are discussed in detail. In Section III, the computational complexity of the new algorithm is discussed. A comparative computational analysis, with respect to computer requirements, is made between the new distributed algorithm and the classical approach discussed in [7] and [8]. The practical usefulness of the distributed algorithm is investigated by applying the comparative analysis to specific examples of multisensor deconvolution. Finally, Section IV summarizes our conclusions.

II. Distributed Seismic Deconvolution

We start the derivation of our distributed deconvolution algorithm by assuming a hierarchical data fusion network, in which a central processing facility is connected through communication links to a number of local nodes (see Fig. 1)

In such a processing architecture, each local sensor or group of sensors (geophone cluster) collects data and generates its own local estimates, which are then communicated to the central facility for further processing. The following points should be noted regarding this formulation.

1) Since the number of measurements far exceeds the dimensionality of the state used to describe the seismic signal, considerable savings in terms of communication bandwidth can be achieved. The local estimates can be transmitted in the central facility at a fraction of the time required to transfer the raw data. Given that the flow of information is only from the local subgroups to the central facility, the system is robust to communication bottlenecks.

2) Since part of the data processing takes place at the different local sensor locations, the computing power required in the central cite to complete the processing

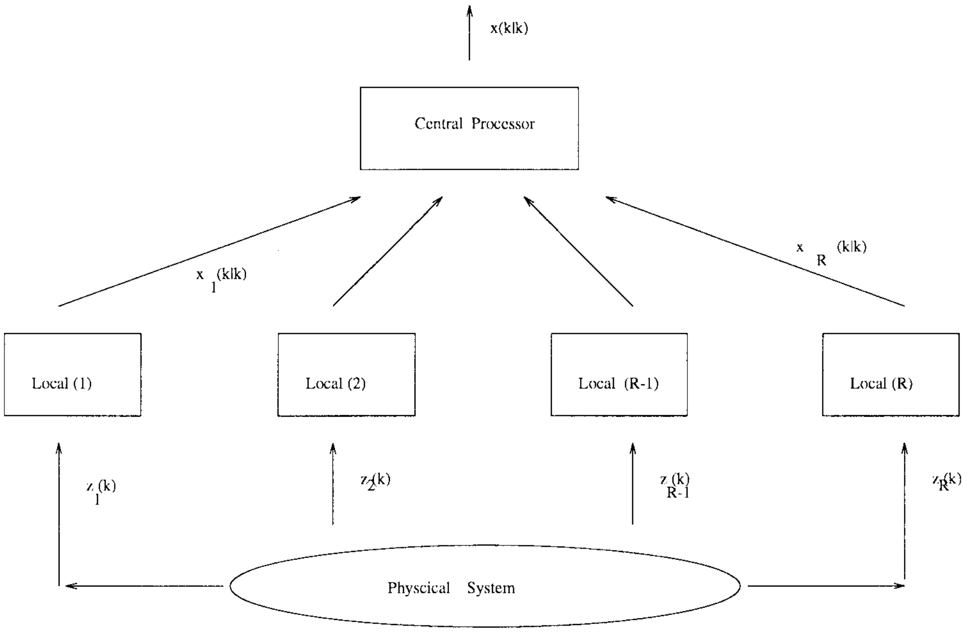

Fig. 1. The distributed approach to state estimation.

task is considerably smaller than that required by the centralized methodology. Thus, a less powerful processor can generate the same results, resulting in a more efficient utilization of computing power.

3) Sensor failures in the local subsystems, commonly encountered in practice, since geophone arrays are often deployed in remote locations or underwater, will not result in a complete failure of the whole processing system. Only a gradual deterioration of the performance should be expected. Replacement of the faulty component and integration of a new one can be done while the data processing system is on-line.

In the new formulation, the state-space equivalent model used is given by (1) and (7). The distributed form of the deconvolution algorithm combines a distributed filter introduced in [19] and a new distributed smoother to provide the optimal estimate of the sequence $\mu(k)$. The equations of the algorithm are summarized in the following theorem.

Theorem II.1) Distributed Deconvolution Algorithm: Consider the system (1), (7). For this linear, time-invariant, discrete-time system, the optimal deconvolution algorithm is given by

$$\begin{aligned}\hat{\mu}(k+1 \mid k+l) &= q(k)\gamma^{\tau}(k)[\Phi(k+1,k)P(k+1 \mid k+1)\Phi^{\tau}(k+1,k) \\ &\qquad +\gamma(k)q(k)\gamma^{\tau}(k)]^{-1} \\ &\quad \times [\hat{x}(k+1 \mid k+l) - \Phi\hat{x}(k \mid k)] \qquad (10)\end{aligned}$$

where $P(k \mid k) = E[(x(k) - \hat{x}(k \mid k))(x(k) - \hat{x}(k \mid k))^{\tau}]$. The equation delivers the fixed-point smoothed estimate of $\mu(k)$, utilizing distributed filtering/smoothing schemes.

A distributed linear filter [19] is used to provide the initial estimate required by the above deconvolution formula. The filter equations (for the general time varying, linear state space model) are the following.

- Distributed Lainiotis Filter (DLF); Central Processor Calculations

$$\begin{aligned}\hat{x}(k+1 \mid k+1) &= \hat{x}_n(k+1 \mid k+1) + \Phi_n(k+1,k) \\ &\quad \times P_r(k \mid k+1)\hat{x}_r(k \mid k+1) \qquad (11)\\ \hat{x}_r(k \mid k+1) &= [M_n(k+1) + P^{-1}(k \mid k)\hat{x}(k \mid k)] \qquad (12)\\ P(k+1 \mid k+1) &= P_n(k+1 \mid k+1) + \Phi_n(k+1,k) \\ &\quad \times P_r^{-1}(k \mid k+1)\Phi_n^{\tau}(k+1,k) \qquad (13)\\ P_r(k \mid k+1) &= [O_n(k+1) + P^{-1}(k \mid k)] \qquad (14)\\ P_n^{-1}(k+1 \mid k+1) &= Q^{-1}(k) + \sum_{i=1}^{r} B_i(k+1) \qquad (15)\\ F_n(k+1) &= Q^{-1}(k)\Phi(k+1,k) \qquad (16)\end{aligned}$$

where

$$\begin{aligned}Q(k) &= \gamma(k)q(k)\gamma^{\tau}(k) \\ \Phi_n(k+1,k) &= P_n(k+1 \mid k+1)F_n(k+1)\end{aligned} \qquad (17)$$

with initial condition $\Phi(0,0) = I_{n\times n}$

$$\begin{aligned}O_n(k+1) &= F_n(k+1)^{\tau} \\ &\quad \times [Q(k) - P_n(k+1 \mid k+1)] \\ &\quad \times F_n(k+1) \qquad (18)\\ M_n(k+1) &= F_n(k+1)^{\tau}\hat{x}_n(k+1 \mid k+1) \qquad (19)\\ \hat{x}_n(k+1 \mid k+1) &= P_n(k+1 \mid k+1) \\ &\quad \times \left(P_n^{-1}(k+1 \mid k)\hat{x}_n(k+1 \mid k)\right. \\ &\qquad \left. + \sum_{i=1}^{r} A_i(k+1)\right) \qquad (20)\end{aligned}$$

with

$$\hat{x}_n(k+1 \mid k) = \Phi(k+1,k)\hat{x}_n(k \mid k).$$

- DLF; Local Processor Calculations (for the ith local subsystem)

$$\begin{aligned} B_i(k+1) &= P_{ni}^{-1}(k+1 \mid k+1) - Q^{-1}(k) \\ &= H_i^\tau(k+1)R_i^{-1}(k+1)H_i(k+1) \qquad (21) \\ A_i(k+1) &= P_{ni}^{-1}(k+1 \mid k+1)\hat{x}_{ni}(k+1 \mid k+1) \\ &= H_i^\tau(k+1)R_i^{-1}(k+1)z_i(k+1). \qquad (22) \end{aligned}$$

The distributed smoother also operates in both the local subsystems and in the central processor. For a fixed initial smoothing point k, the equations of the distributed fixed point smoother (for a general time varying, linear system) are as follows.

- Distributed Lainiotis Smoother (DLS); Central Processor Calculations

$$\hat{x}(k \mid l,k) = P(k \mid l)[M_{sn}(l,k) + P^{-1}(k \mid k)\hat{x}(k \mid k)] \qquad (23)$$

$$\begin{aligned} P(k \mid l) &= [O_{sn}(l,k) + P^{-1}(k \mid k)]^{-1} \qquad (24) \\ C_{sn}(l,k) &= \left[P_{sn}^{-1}(l \mid l-1)\Phi(l,l-1)\Phi_{sn}(l,l-1)\right] \qquad (25) \\ O_{sn}(l,k) &= O_{sn}(l-1,k) + C_{sn}(l-1,k)^\tau \\ &\quad \times [P_{sn}(l \mid l-1) - P_{sn}(l \mid l)]C_{sn}(l-1,k) \qquad (26) \\ M_{sn}(l,k) &= M_{sn}(l-1,k) + C_{sn}(l-1,k)^\tau \\ &\quad \times [\hat{x}_{sn}(l \mid l) - \hat{x}_{sn}(l \mid l-1)] \qquad (27) \\ \Phi_{sn}(l,k) &= P_{sn}(l \mid l)C_{sn}(l-1,k) \qquad (28) \\ \hat{x}_{sn}(l \mid l) &= P_{sn}(l \mid l)\Bigg(P_{sn}^{-1}(l \mid l-1)\hat{x}_{sn}(l \mid l-1) \\ &\quad + \sum_{i=1}^{r} A_{sni}\Bigg) \qquad (29) \\ P_{sn}^{-1}(l \mid l) &= P_{sn}^{-1}(l \mid l-1) + \sum_{i=1}^{r} B_{sni} \qquad (30) \\ P_{sn}(l \mid l-1) &= \Phi(l,l-1)P_{sn}(l-1 \mid l-1)\Phi^\tau(l,l-1) \\ &\quad + \gamma(k)q(k)\gamma^\tau(k) \qquad (31) \\ \hat{x}_{sn}(l \mid l-1) &= \Phi(l,l-1)\hat{x}_{sn}(l-1 \mid l-1) \qquad (32) \end{aligned}$$

with $l = k+1, k+2, \ldots$ and initial condition $P(k \mid k)$, $\hat{x}(k \mid k)$.

- DLS; Local Processor Calculations (for the ith local subsystem)

$$\begin{aligned} B_{sni} &= P_{sni}^{-1}(j+1 \mid j+1) - P_{sni}^{-1}(j+1 \mid j) \\ &= H_i^\tau(j+1)R_i^{-1}(j+1)H_i(j+1) \qquad (33) \\ A_{sni} &= P_{sni}^{-1}(j+1 \mid j+1)\hat{x}_{sni}(j+1 \mid j+1) \\ &\quad - P_{sni}^{-1}(j+1 \mid j)\hat{x}_{sni}(j+1 \mid j) \\ &= H_i^\tau(j+1)R_i^{-1}(j+1)z_i(j+1) \qquad (34) \end{aligned}$$

with $j = k+1, k+2, \ldots$ and arbitrary nominal initial conditions $\hat{x}_{sni}(k \mid k)$ and $P_{sni}(k \mid k)$, which can be selected independent from the actual filtered $\hat{x}(k \mid k)$ and $P(k \mid k)$. The proof is given in the Appendix.

The distributed filter and smoother presented above correspond to the general time-varying, linear-dynamic, discrete-time, state-space model. However, since the seismic wavelet is time invariant, the algorithm simplifies further by substituting the time-varying matrices $\Phi(j, j-1)$, $H(j)$, $R(j)$, $\gamma(k)$, and $q(k)$ with the time-invariant matrices Φ, H, R, γ, and q. In this way, the vast majority of the calculations required at the central processor and the different local processors, such as those of (13)–(17), (18), (21), (24)–(26), (28), (30)–(31), and (33), become nonrecursive, and therefore, it is sufficient to be performed only once.

To fully appreciate how interesting the above point is, it must be noticed that formulations based on the Kalman (or information Kalman) filter and the Meditch smoother, such as those presented in [6] and [9] are time varying, even for time-invariant wavelet models. However, the distributed algorithm presented here as well as its centralized counterpart of [7] and [8] have reduced computational requirements not only for time-invariant models, but for the following models as well.

1) Periodic models: In this time-varying case, the matrices $\Phi(j, j-1)$, $H(j)$, $R(j)$, $\gamma(k)$, and $q(k)$, although time-varying, are periodic in nature. Thus, calculations involving these quantities can be performed for a data subinterval equal to the model period and stored for subsequent use in the remaining part of the total data interval.
2) Slowly time-varying models: In this case, the total data interval can be partitioned into subintervals, of which the wavelet model is approximately time invariant, although different in each subinterval. Once again, calculations, such as those of (13)–(17), (18), (21), (24)–(26), (28), (30)–(31), and (33), need only be computed once for each subinterval and stored for use throughout the subinterval to which they pertain.

It must be noted that the central processor does not require the optimal local estimates at each local subsystems. If, however, the optimal local estimates are requested, they can be computed by each local processor, in parallel with the calculations in the central agent, without additional computational delays. One of the characteristics of the proposed solution is that the distributed algorithm uses the same equations for filtering and smoothing. More than that, since only nominal smoothing quantities are calculated at the local systems, the smoothing part can start with arbitrary initial conditions running in this way, independent of the filtering operation at the local level. The algorithm will compensate correcting, at the central processor, the suboptimal initial smoothing values at the end of the data interval. As for the optimal filtered estimates, if the optimal smoothed estimates are required at the local level, these can be obtained from each local processor independently, without delaying the calculations in the main processor. The nominal calculations required by the algorithm at the different local subsystems (geophone clusters) are only nonrecursive simple multiplications. These memoryless operations can be implemented by using a very simple processor or even an optical device. Thus, using the proposed distributed scheme, there is no need to install expensive computing facilities on remote-sensing locations or in hazardous environments. Due to the structure of the proposed algorithm, there is no two-way communication between the local processors and the central computing facility. Because bidirectional communication between slower local stations and the central agent is not desirable in a highly parallel processing environment, both the distributed estimator and the smoother have been designed to require the lowest possible interprocessor communication.

The distributed approach to seismic deconvolution presented in the above theorem utilizes distributed filters and distributed smoothers to obtain the quantities required in (10). There are two ways in which we can achieve parallelism by using the proposed deconvolution algorithm:

- parallelism in the system's level;
- parallelism in the algorithmic computations.

Parallelism in the system's level is based on the fact that any local agent (processor) can execute the calculations required by the central agent in parallel with the other local processors. In this way, in one step of the algorithm, *r-operations* are executed in parallel at the different local processors. Parallelism in the algorithmic computations is based on the fact that the filtering equations can be implemented independently of the smoother calculations. The only coupling between filtering and smoothing is present in (23) and (24), which are used only once at the end of the smoothing session. Using (23) and (24) with the structure of (10), we are able to implement the algorithm in a totally parallel/distributed fashion. It must be noted that the implementation of the filter in parallel with the smoother is also a characteristic of the centralized approach presented in [7] and [8]. However, the distributed formulation of the algorithm provides tremendous parallel processing capabilities. For each one of the r local systems (geophones clusters), the nominal quantities required from the filtering portion are calculated in parallel with the nominal quantities required from the smoothing portion at the same local system. At the same time, all of the nominal calculations can be executed in parallel at the r local processing systems. In addition, at the central processor, the algorithm allows the parallel calculation of the filtering and smoothing part for all the sampling periods under consideration. To the best of the authors' knowledge, the solution given in this work is the only algorithm capable of providing this form of parallel processing capability.

In this paper, only the distributed/hierarchical form of the algorithm is considered. The distributed scheme described in Fig. 1 requires the existence of a central processing facility (central node) and of several peripheral computing devices installed at the local sensor clusters. We adopted this data fusion network configuration since it is the most appropriate for the seismic deconvolution problem. In practice, when geophysical signal processing is requested, a central processing facility is always available and less powerful computing facilities are usually deployed in the field sites. In addition, the hierarchical distributed configuration is the most commonly used configuration for multisensor data fusion [16], [17], [20]. However, our distributed formulation is immediately applicable to any type of distributed or decentralized data fusion network. Specifically, in case no central agent is present, each local processor can be considered to be a central node. If the optimal overall estimate of the system state is required at a particular location, this node can act as central node and generate the overall estimate, considering all other nodes as local sensing groups, with exactly the same equations.

It must emphasized that the distributed deconvolution algorithm developed here is mathematically equivalent to the optimal solution derived in [7]. The distributed filter and smoother that are developed are different realizations of the optimal solutions and exact mathematical equivalents to the optimal centralized approaches. The distributed deconvolution algorithm introduced in this work has identical performance with the deconvolution algorithm studied in [7] and [8], since both algorithms involve the same system matrices. However, the solutions have different computational requirements, especially when they are applied to the multisensor seismic deconvolution problem [7], [17], [18].

III. Computational Requirements of the Solutions

Apart from the numerical behavior of any proposed algorithm, its computational complexity is a realistic measure of its practicality and usefulness, since it determines the required computing power and processing (execution) time. A general framework to evaluate the computational requirements of recursive algorithms is given in [7] and [18]. The framework of that analysis is used here to evaluate the computational requirements of the algorithms.

Because the algorithms provide the solution in a recursive manner, the algorithms' total execution time is equal to the product of their per step calculation multiplied by the number of recursions required to obtain a solution. Since in the distributed formulation the calculations at the local level are done in parallel, the overall computational requirements for a distributed algorithm, per recursion, are computed as the sum of the per recursion calculation requirements in the central processor and the calculations required at one local processor. Two more assumptions are introduced to have a meaningful comparison among the different algorithms. First, the overall computational requirements for a distributed algorithm, per recursion, are computed as the sum of the per recursion calculation requirements in one of the local processors plus the computational requirements in the central processor, since the calculations in all local processors are performed in parallel. Secondly, the fundamental operations involved in the algorithms are matrix and vector operations. A detailed analysis of the computations involved in such operations is provided in [8] and [19] (refer to them for more information on the subject). In this context, the total time required to complete an operation (or a sequence of operations) is proportional to the normalized total number of equivalent scalar operations defined as

$$\begin{aligned}\mathit{Time} = k \times (4 \times (\mathit{MULTS}) + (\mathit{ADDS}) + 6 \times (\mathit{DIVS})\\ + 25 \times (\mathit{SQRTS}))\end{aligned}$$

where *MULTS* is the number of scalar multiplications required, *ADDS* is the number of scalar additions required, *DIVS* is the number of scalar divisions required, and *SQRTS* is the number of the scalar square roots. The weights used in the above formula do not pertain to any particular machine. Rather, they can be considered mean values of those coefficients commonly encountered. All qualitative results presented in the sequence hold, even if the weighting coefficients in the above formula are different for a specific computing platform [8], [19].

In the seismic deconvolution problem, the algorithms (centralized and distributed) require a filtering operation on the data up to point k (for the evaluation of $\hat{x}(k \mid k)$) and then a smoothing operation on the whole data interval. Moreover, filtering and smoothing calculations can be executed in parallel, and a number of operations are executed only at the end of the period under consideration. Hence, our comparisons will be on a per interval basis and not per recursion to accommodate the smoothing process.

Next, we will compare the centralized and the distributed approach under two different parallelism scenarios. First, in a scenario called sequential/distributed, it is assumed that only one central processor and r local processors are available (where r is the number of local systems). Under this scenario, each processor calculates the filtering/smoothing operations sequentially in time. The only parallel processing operation is the execution of the distributed algorithm calculations in the different local processors in parallel. On the contrary, in a second scenario, parallel processing machines are assumed available. It is assumed that a central parallel processor and r local parallel machines are available. Under this scenario, the machines can implement the filter in parallel with the smoother. The availability of the parallel machines allows for the full utilization of the proposed distributed approach to the optimal seismic deconvolution.

The term "normalized operations" is used in both the sequential and parallel versions of the algorithms. In the parallel implementation, the number of operations remains the same as in the sequential version of the algorithm, but the algorithm is executed faster because a portion of the required operations is implemented concurrently with some other portion. Many factors affect the parallel algorithms' performance. Such factors include the structure of the algorithm implemented on the parallel machine, the programming system, the actual platform, and the run time [21]. The parallel implementations need, in general, less operation time than the sequential ones. However, the memory requirements or the run time of the parallel algorithm can be more than that of the sequential one. The reason is that the run time includes operations, such as memory access delays, operating system functions, etc., which are performed internally in the parallel machine and have nothing to do with the actual computations [8]. This situation is of course outside the scope of the present paper, since it heavily depends on the specific configuration of the computing platform. It is assumed here that, in all parallel machines used, the operating system provides the same CPU time to all processes running in parallel [21].

In order to establish the algorithms requirements in a multiprocessor parallel machine, two additional assumptions are introduced. Namely, we assume that a) two processes running in parallel cannot access the same memory location at the same time and b) the number of available processors, per parallel machine, is large enough so that no process that can be activated is delayed due to the lack of processors. Thus, any process can be activated as soon as its inputs are available. In other words at least two processors, per parallel machine, are assumed available to implement the filtering part in parallel with the smoother.

TABLE I
NOTATION

CEDEP	Centralized Deconvolution solution, Parallel mode
DEDES	Distributed Deconvolution solution, Sequential mode
DEDEP	Distributed Deconvolution solution, Parallel mode
CLF	Centralized Lainiotis Filter
CLS	Centralized Lainiotis Smoother
DLF	Distributed Lainiotis Filter
DLS	Distributed Lainiotis Smoother

In the computational analysis presented here, only fixed point smoothers are considered (see Table I for the notation used in our analysis). In the case of the sequential/distributed scenario, the algorithms' operations are executed sequentially for each step in the interval. At the end, the computational requirements for the implementation of (10) and (23) and (24) must be added. When a parallel/distributed scenario is selected, the filter and smoother equations are implemented in parallel and (10) is used to combine the results. The total number of operations is therefore calculated according to the following formula:

$$\max[OT_f, (0,k); OT_s, (k,N)] + (N-k+1) \times OT_d$$

where OT_f is the operation time to execute the filtering part over the interval $(0,k)$ and OT_s is the operation time to execute (in a per step basis) the nominal smoothing calculations over the interval (k,N) and the calculations at (23) and (24). Finally, OT_d is the time needed for the implementation of the deconvolution formula of (10).

To determine the algorithms' complexity both in sequential and parallel mode, we have to first consider the computational requirements of the different components of the algorithm, namely, the filter, the associated smoother, and finally, the deconvolution formula. Furthermore, since the model for the optimal seismic deconvolution is assumed time invariant [8], [10], the computational requirements for the recursive implementation of the algorithms' can be divided into two parts. A preliminary part, which contains both the operations that are executed only once in the beginning of the process (or off-line) since their computation does not require the actual measurements or recursively evaluated quantities and those that are calculated at the end of the smoothing operation. The per step part includes the calculations that have to be executed at each step of the recursive algorithm.

As far as centralized implementations are concerned, several computational analyses of the filtering and the smoothing

TABLE II
CENTRALIZED APPROACH: COMPUTATIONAL REQUIREMENTS

Part	Normalized Operations
CLF (preliminary)	$22.5n^2m + 17.5m^2n + 5m^3 + 12m^2 + 65m + 5mn + n^2 + 5n - 12$
CLF (per step)	$12.5n^3 + 23.5n^2 + 64n + 10nm - 12$
CLS (per step)	$7.5n^3 + 7.5n^2 + 20mn^2 + 12.5m^2n + 18.5mn + 2.5m^3 + 6m^2 + 32.5m - 6$
CLS (final)	$5n^3 + 22.5n^2 + 64.5n - 12$

TABLE III
DISTRIBUTED APPROACHES
LOCAL PROCESSORS: COMPUTATIONAL REQUIREMENTS

Part	Normalized Operations
DLF (preliminary)	$2.5m_i^3 + 6m_i^2 + 2.5n^2m_i + 5nm_i^2 + 1.5nm_i + 32.5m_i - 0.5n^2 - 0.5n - 6$
DLF (per step)	$5nm_i - n$
DLS (preliminary)	$2.5m_i^3 + 6m_i^2 + 2.5n^2m_i + 5nm_i^2 + 1.5nm_i + 32.5m_i - 0.5n^2 - 0.5n - 6$
DLS (per step)	$5nm_i - n$

TABLE IV
DISTRIBUTED APPROACHES
CENTRAL PROCESSOR: COMPUTATIONAL REQUIREMENTS

Part	Normalized Operations
DLF (preliminary)	$20n^3 + 14.5n^2 + 65n - 12 + 0.5r(n^2 + n)$
DLF (per step)	$12.5n^3 + 27n^2 + 64.5n - 12 + rn$
DLS (preliminary)	$17.5n^3 + 42n^2 + 97.5n - 24 + 0.5r(n^2 + n)$
DLS (per step)	$15n^3 + 22.5n^2 + 2.5n$
DLS (final)	$10n^3 + 23.5n^2 + 29.5n - 6$

TABLE V
SEISMIC WAVELET ESTIMATION PROBLEM

method	normalized operations
CEDES	$7.5e + 08$
DEDES	$3.e + 03$

algorithms employed by the deconvolution formula of (9) have appeared in the literature, including [10]. The normalized operations needed for the implementation of the centralized deconvolution approach of [8] are summarized in Table II. Tables III and IV summarize the number of normalized operations needed for the implementation of the proposed distributed implementation.

To compare the algorithms' computational efficiency, the total normalized operations required by centralized and distributed algorithms were plotted against the estimation point as well as against the state and measurement dimensionality. In this analysis, the parameters are the number of channels/sensors m, the order of the basic seismic wavelet n, the running estimation point k, and the local systems (geophone clusters) r. The values used in these studies are $m = 50$ or $m = 500$ and $n = 4$ or $n = 9$, respectively. The total number of data points (time instants when measurements were taken) was assumed to be $N = 1000$. The number of local systems, when it is not a parameter, is assumed to be $r = 5$. Moreover, for the parallel version of the algorithm, it is assumed that at least two processors, per parallel machine, are available for the parallel implementation of the smoothing and filtering part.

It was mentioned above that the parallel version of the algorithm is always more efficient regarding the operations involved than the sequential version. On the other hand, the storage requirements are larger and of course the realization is more complex. Since today's technology has minimized memory cost and sizes, what actually interests us is the realization complexity. The question arises then, whether it is worth implementing the deconvolution algorithm in parallel, in view of its reduced operations, while at the same time the realization complexity increases. To answer this question, the *para-efficiency* ratios

$$O_{\text{paref}} = \frac{\text{Processing Time (parallel)}}{\text{Processing Time (sequential)}} \tag{35}$$

will be plotted for both the centralized and the distributed versions of the solution. The plots are made against k (the estimation point) for the fixed point smoother. The above mentioned ratio can serve as a measure of the parallelism efficiency. It is defined as the ratio of the execution time of the parallel implementation (proportional to the number of normalized operations) to the execution time of the sequential implementation. By its definition, the para-efficiency ratio is less than unity, and as it tends to zero parallelism efficiency improves. In case only two processors per parallel machine are assumed available, the lower limit of the ratio is set to 0.5 [10], [18].

Experimentation via simulation analysis of the system that represents the particular design problem can help us to assess further the effectiveness of the distributed realization of the deconvolution algorithm. Using the complexity determination techniques introduced in this paper, the designer will be able to determine the most suitable solution for the problem on hand. To illustrate the discussion, a typical seismic deconvolution scenario is utilized. In this typical seismic deconvolution

TABLE VI
STORAGE REQUIREMENTS

Centralized Deconvolution solution	$4n^2 + 8n + 7mn + 0.5m^2 + 2m + 0.5m^2$
Distributed Deconvolution solution	$0.5n(n+3) + 2r(3.5n^2 + 3.5n + 2nm_i + m_i)$

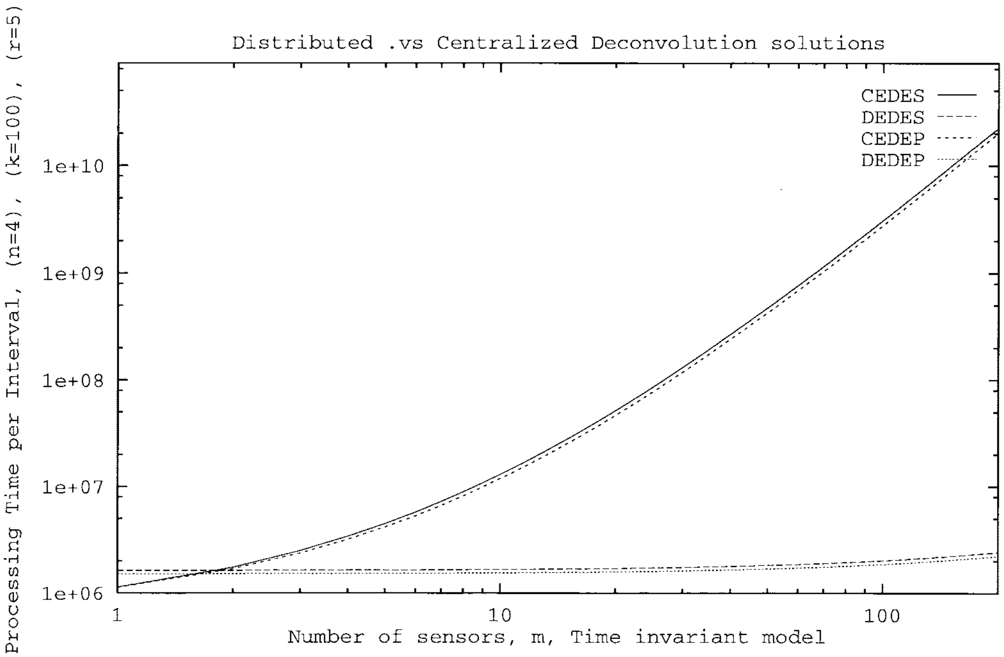

Fig. 2. Distributed versus centralized seismic deconvolution. Total operations $(n = 4)$.

example, a time-invariant wavelet is utilized to describe the signal received by the seismic sensors [7]. A convolution summation of order $n = 4$ describes the seismic wavelet. For this time-invariant seismic wavelet of order $n = 4$, assuming $m = 1000$ sensors divided in $r = 10$ local subsystems (geophones clusters), the normalized operations (computational requirements) for the deconvolution formula of (9) or (10), which are required in the source estimation process are summarized in Table V.

It is obvious that, in this case, the preferable realization of the deconvolution formula, from a computational point of view, is the distributed one that can achieve the same levels of performance, while being at the same time extremely cost attractive (five orders of less expensive magnitude).

Storage requirements per iteration had been used in [7] and [18] to determine the computational characteristics of the algorithm. The storage requirements are divided into two types. Namely, permanent storage, which refers to the amount of computer memory that must be permanently assigned to the algorithm and auxiliary storage, which corresponds to the extra computer memory that is used for temporary storage of auxiliary variables needed per iteration to 'run' the algorithm. This analysis involves several complexities, such as those associated with computer word len gth, software used, etc. A complete treatment of these questions is beyond the scope of this paper. In addition, recent advances in technology and the availability of cheap inexpensive memory devices allows users to cope with the amount of storage required in most demanding applications. However, for comparison purposes, Table VI summarizes storage requirements for the centralized and distributed version of the minimum variance deconvolution formula.

Significant findings and corresponding remarks are herein organized in a series of comments, which are supported by the appropriate figures.

1) In a multisensor scenario, such as the seismic deconvolution for oil exploration, the distributed algorithm is more efficient, from a computational point of view, than the centralized approach utilized in [7]–[10]. Depending on the order of the seismic wavelet and the total number of geophones used to capture the seismic trace, the difference in computing power required to deliver the same estimation performance can be up to four orders of magnitude. For fixed wavelet order and a given number of local processing cites, the efficiency of the distributed approach *vis a vis* the centralized one increases exponentially as the total number of geophones increases (Figs. 2 and 3).
2) For a given wavelet order and fixed number of measurement channels (geophones), the efficiency of the distributed approach increases as the number of local processors increases. Increasing the local processing cites by an order of magnitude results in orders of magnitude difference in the computing power required to implement the estimation procedure (Figs. 4 and 5).

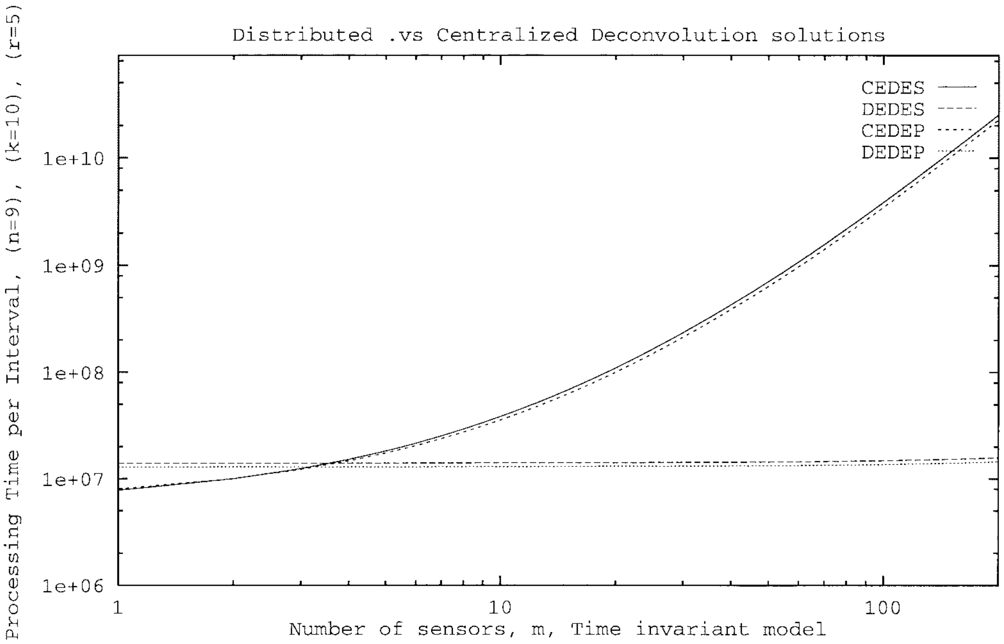

Fig. 3. Distributed versus centralized seismic deconvolution. Total operations $(n = 9)$.

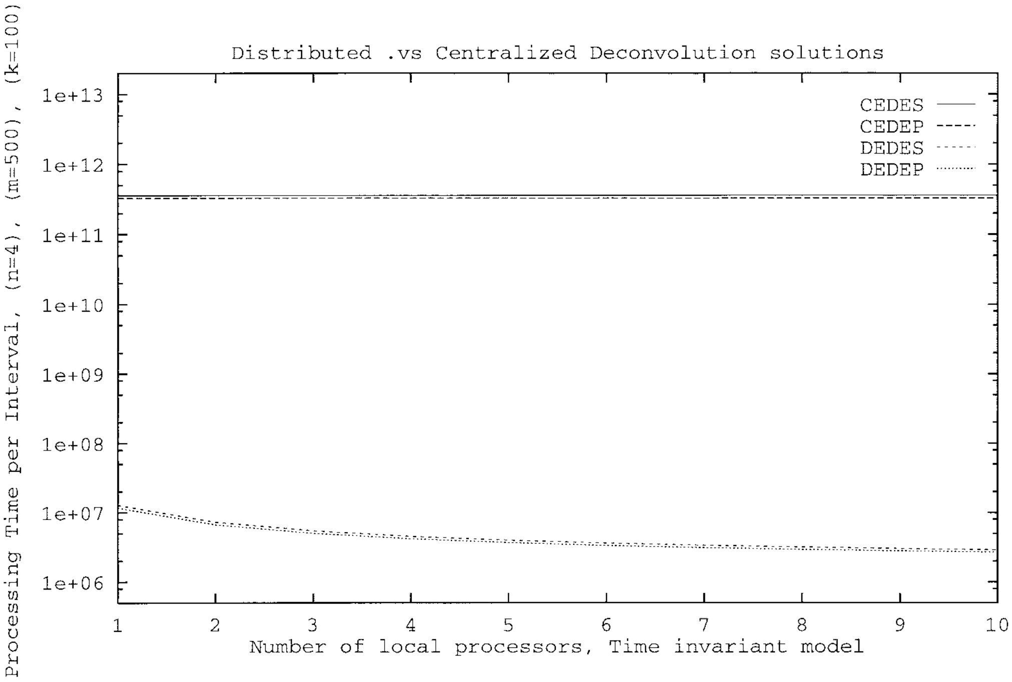

Fig. 4. Total operations versus the number of local processors $(n = 4,\ m = 500)$.

3) For given wavelet order and plotted against the initial smoothing point k, the distributed algorithm is more computationally attractive from its centralized counterpart. Depending on the number of sensors utilized, there is orders of magnitude difference in computing power between the distributed and centralized approach (Figs. 6 and 7).
4) In order to evaluate the parallelism efficiency of the algorithms for both the distributed and sequential formulation, we plot the para-efficiency ratio against the initial smoothing point k. From the results we conclude the following.

   a) The choice between parallel or sequential implementation depends on both the wavelet order (n) and the number of sensors (m).
   b) As expected, both the distributed and the centralized algorithm obtain their maximum parallelism efficiency when the initial smoothing point is placed around the middle of the observation record. The

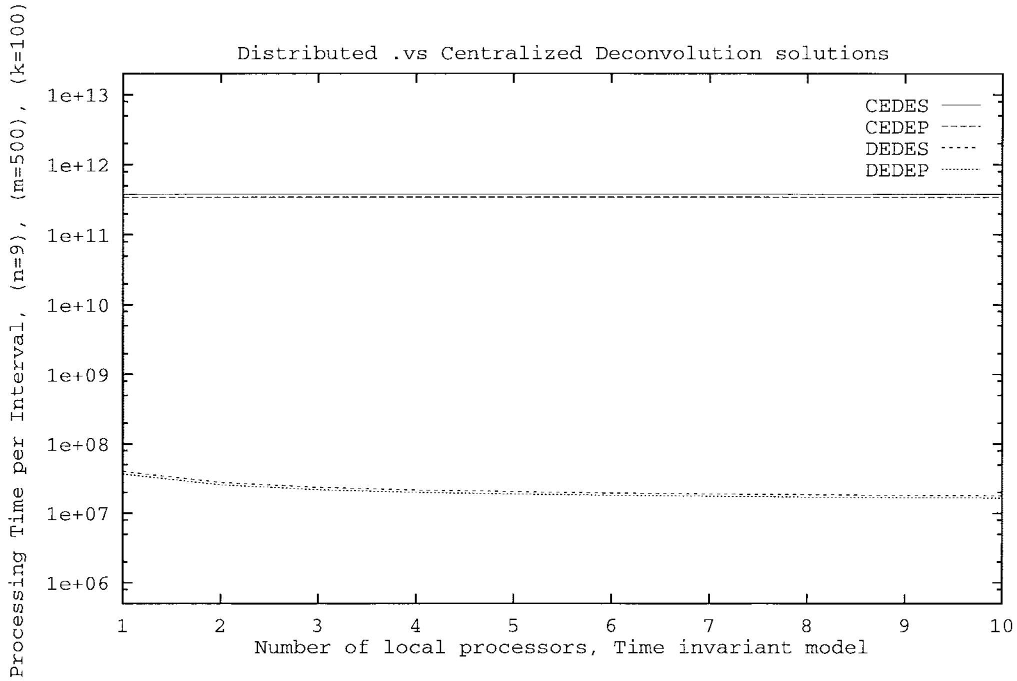

Fig. 5. Total operations versus the number of local processors $(n = 9,\ m = 500)$.

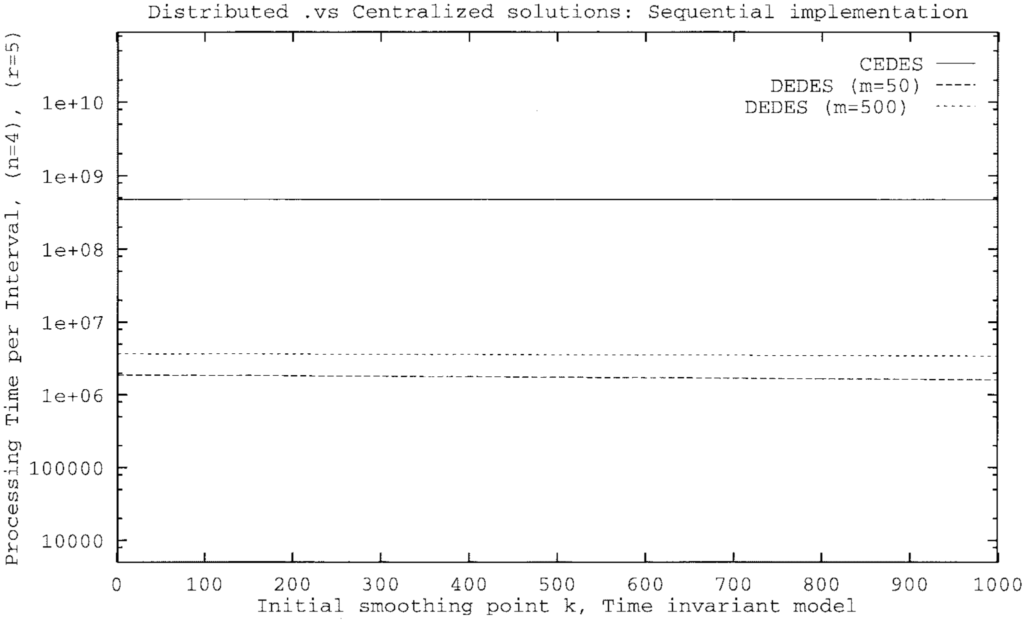

Fig. 6. Sequential implementation: distributed versus centralized solution $(n = 4)$.

para-efficiency ratio for different initial selections can be seen in Figs. 8 and 9.

In summary, results indicate that the distributed/parallel formulation of the proposed algorithmic solution is most attractive from a computational standpoint. The use of a distributed approach in the multisensor/multichannel deconvolution problem is always preferable.

IV. Conclusion

In this paper, a state-space deconvolution formula suitable for use in a multisensor/multichannel environment was developed and investigated. A new efficient and computationally attractive algorithm was obtained by employing new distributed filters and smoothers. The new algorithm implements in a distributed, parallel fashion the optimal minimum variance deconvolution formula. We have studied in detail the computational requirements of various options of the new algorithm, with respect to operation counts and parallelism efficiency.

Both the centralized and the new distributed algorithm are different realizations of the optimal minimum variance algorithm. However, their performance in practice is determined from their computational complexity. The centralized approach is computationally demanding, especially in a highly interactive environment. For such an algorithm to be practical, a

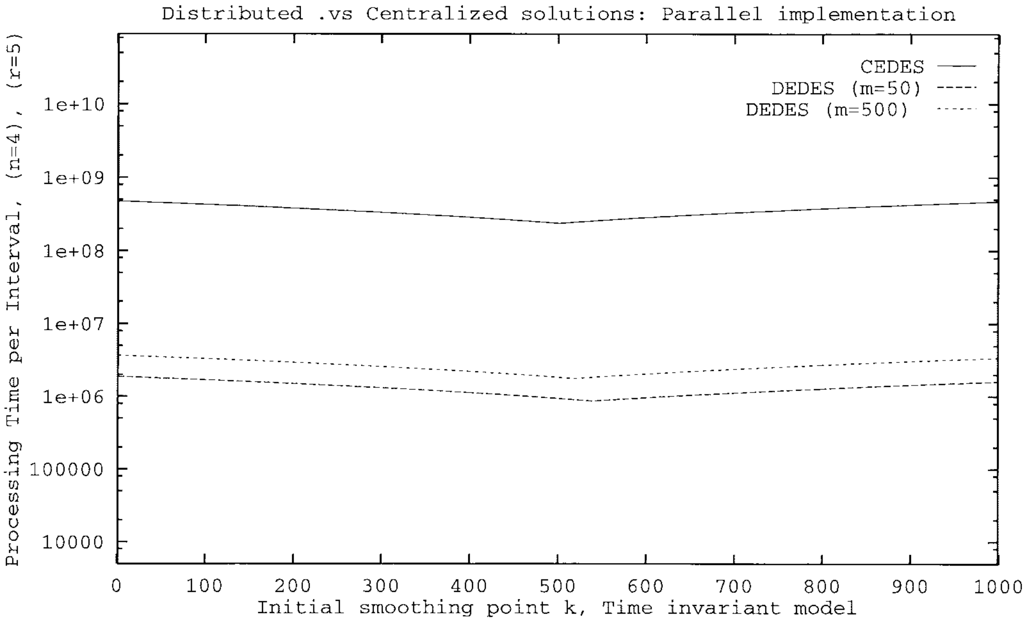

Fig. 7. Parallel implementation: distributed versus centralized solution $(n = 4)$.

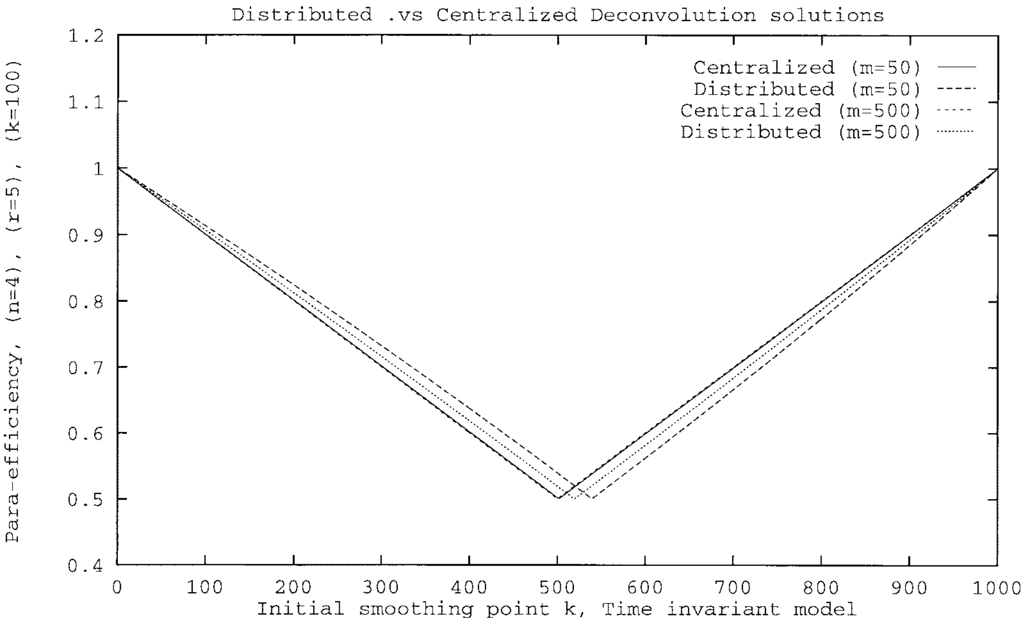

Fig. 8. Parallelism efficiency $(n = 4)$.

reduced number of data points or a smaller array of geophones should be used at the expense of resolution and accuracy. On the other hand, the distributed algorithm, with its reduced complexity, allows for the utilization of additional sensors and/or extra processing through larger processing windows or higher sampling rate, and thus, it can generate better deconvolution results.

APPENDIX

In this appendix, we outline the solution of the distributed approach to seismic deconvolution. The deconvolution formula is executed only on the central processor and does not require direct use of the measurements or any measurement-related matrix. Thus, the key part of the solution is the derivation of the distributed estimator and distributed smoother utilized by the deconvolution formula of (10).

First, we devote our attention in the development of the distributed filter. Our objective is to eliminate the measurement equation matrices from the filter's equations. Thus, we modify the equations of the centralized Lainiotis filter (CLF) used in [7]. The most important step is to obtain an expression for the nominal observability matrix $O_n(k+1)$ without utilizing any measurement related matrix.

We start with an interim result, the expression for the filter transition matrix $\Phi_n(k+1,k)$. The filter transition matrix in the centralized linear Lainiotis filter, defined as $\Phi_n(k+1,k) =$

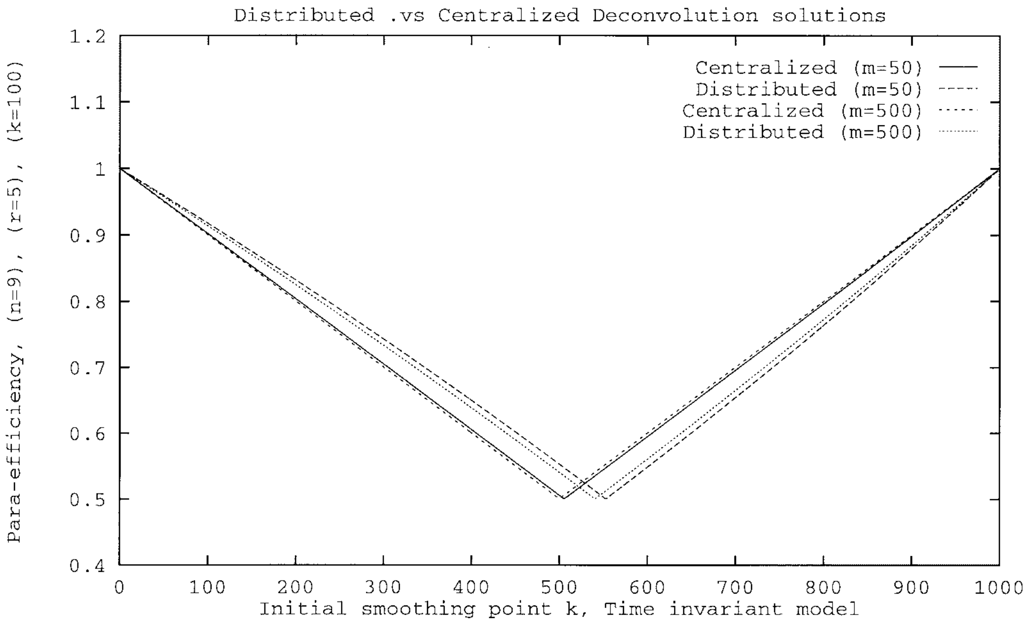

Fig. 9. Parallelism efficiency ($n = 9$).

$[I-K_n(k+1)H(k+1)]\Phi(k+1,k)$ requires for its realization the overall output matrix $H(k+1)$. The equations of the standard Kalman filter can be used to eliminate this quantity from the expression for the filter transition matrix [22]. It is well known that the Kalman filter covariance $P(k \mid k) = [I-K(k)H(k)]P(k \mid k-1)$ is a square matrix for which the inverse $P^{-1}(k \mid k-1)$ exists. Thus, we can claim that

$$P(k \mid k)P^{-1}(k \mid k-1) = [I-K(k)H(k)]. \tag{36}$$

Furthermore, in the nominal Kalman filter used in the derivation of the CLF, it was assumed that $P(k \mid k-1) = Q(k-1)$, since the nominal filter is initialized at each step with zero nominal initial conditions [13]. Thus, for such a filter, (36) can be written as $P_n(k \mid k)Q^{-1}(k-1) = [I-K_n(k)H(k)]$. Utilizing this result, the transition matrix for the distributed filter can be defined accordingly as

$$\Phi_n(k+1,k) = P_n(k \mid k)Q^{-1}(k-1)\Phi(k+1,k). \tag{37}$$

We apply the same methodology to all quantities involved in the derivation of the CLF. We can see from [8] that both observability matrix $O_n(k+1) = \Phi^\tau(k+1,k)H^\tau(k+1)P_{zn}^{-1}(k+1 \mid k)H(k+1)\Phi(k+1,k)$ and the information quantity $M_n(k+1) = \Phi^\tau(k+1,k)H^\tau(k+1)P_{zn}^{-1}(k+1 \mid k)z(k+1)$ contain measurement-related quantities. As before, our objective is to rewrite the above equations without explicit reference to any measurement-related matrix. It is not hard to see that in the standard Kalman filter the expressions $H^\tau(k)P_z^{-1}(k \mid k-1) = P^{-1}(k \mid k-1)P(k \mid k)H^\tau(k)R^{-1}(k)$, $P^{-1}(k \mid k) - P^{-1}(k \mid k-1) = H^\tau(k)R^{-1}(k)H(k)$ and $P^{-1}(k \mid k)\hat{x}(k \mid k) - P^{-1}(k \mid k-1)\hat{x}(k \mid k-1) = H^\tau(k)R^{-1}(k)z(k)$ hold true [16].

Since we initialize our filter at each time instant with zero nominal conditions, we can claim that the equivalent formulas are $P_n^{-1}(k \mid k) - Q^{-1}(k-1) = H^\tau(k)R^{-1}(k)H(k)$ and $P_n^{-1}(k \mid k)\hat{x}(k \mid k) = H^\tau(k)R^{-1}(k)z(k)$, respectively.

Utilizing these results, the observability matrix for the distributed filter can be written after some simple manipulations as follows:

$$F_n(k+1) = Q^{-1}(k)\Phi(k+1,k) \tag{38}$$

$$\begin{aligned} O_n(k+1) &= F_n(k+1)^\tau \\ &\quad \times [Q(k) - P_n(k+1 \mid k+1)]F_n(k+1). \end{aligned} \tag{39}$$

Moreover, it can be seen that

$$M_n(k+1) = F_n(k+1)^\tau \hat{x}_n(k+1 \mid k+1). \tag{40}$$

Having completed the derivation of the filter equations at the central processor, we focus our attention to the different local systems. The distributed estimator requires only calculations on the nominal part of the local filter. Following the CLF, the nominal filter at each local processor is initialized using zero initial conditions. Thus, $\hat{x}_n(k \mid k) = 0$ and $P_n(k \mid k) = 0$ for the cycle starting at the time instant $(k+1)$.

From the distributed form of the Kalman filter [16], we know that $P_i^{-1}(k+1 \mid k+1) - P_i^{-1}(k+1 \mid k) = H_i^\tau(k+1)R_i^{-1}(k+1)H_i(k+1)$ and that $P_i^{-1}(k+1 \mid k+1)\hat{x}_i(k+1 \mid k+1)$ $-P_i^{-1}(k+1 \mid k)\hat{x}_i(k+1 \mid k)$ $= H_i^\tau(k+1)R_i^{-1}(k+1)z_i(k+1)$. Taking into consideration the initial conditions for the nominal filter at the ith local cluster the above equations can be rewritten as $P_{ni}^{-1}(k+1 \mid k+1) - Q^{-1}(k) = H_i^\tau(k+1)R_i^{-1}(k+1)H_i(k+1)$ with $P_{ni}^{-1}(k+1 \mid k+1)\hat{x}_{ni}(k+1 \mid k+1) = H_i^\tau(k+1)R_i^{-1}(k+1)z_i(k+1)$.

Similarly, the distributed smoother is derived through algebraic operations from the optimal centralized smoother discussed in [7]. Once again, the measurement-related matrices have to be eliminated from the recursive solution. In the centralized smoother, the nominal transition matrix Φ_n is calculated as [12]

$$\Phi_n(l,k) = [I - K_n(l)H(l)]\Phi(l,l-1)\Phi_n(l-1,k). \quad (41)$$

Using the equation of the covariance update for the standard Kalman filter, the nominal filter transition matrix can now be written as

$$\Phi_n(l,k) = P_n(l \mid l)P_n^{-1}(l \mid l-1)\Phi(l,l-1)\Phi_n(l-1,k). \quad (42)$$

The nominal partitioning quantity $O_n(l,k)$ in the centralized smoother is given as

$$\begin{aligned} O_n(l,k) = O_n(l-1,k) + \Phi_n^\tau(l-1,k)\Phi^\tau(l,l-1)H^\tau(l) \\ \times P_{zn}^{-1}(l \mid l-1)H(l)\Phi(l,l-1)\Phi_n(l-1,k). \quad (43) \end{aligned}$$

Using the equations for the nominal Kalman filter in the distributed smoother, the following relations hold:

$$H^\tau(k)P_{zn}^{-1}(l \mid l-1) = P_n^{-1}(l \mid l-1)P_n(l \mid l)H^\tau(l)R^{-1}(l) \quad (44)$$

$$P_n^{-1}(l \mid l) - P_n^{-1}(l \mid l-1) = H^\tau(l)R^{-1}(l)H(l). \quad (45)$$

Combining the last three equations, the quantity $O_n(l,k)$ can be written as follows:

$$\begin{aligned} O_n(l,k) &= O_n(l-1,k) + \Phi_n^\tau(l-1,k)\Phi^\tau(l,l-1) \\ &\quad \times P_n^{-1}(l \mid l-1)P_n(l \mid l)H^\tau(l)R^{-1}(l)H(l) \\ &\quad \times \Phi(l,l-1)\Phi_n(l-1,k) \quad (46) \\ O_n(l,k) &= O_n(l-1,k) + \Phi_n^\tau(l-1,k)\Phi^\tau(l,l-1) \\ &\quad \times P_n^{-1}(l \mid l-1)P_n(l \mid l)\big[P_n^{-1}(l \mid l) \\ &\quad - P_n^{-1}(l \mid l-1)\big]\Phi(l,l-1)\Phi_n(l-1,k) \quad (47) \\ O_n(l,k) &= O_n(l-1,k) + \big[P_n^{-1}(l \mid l-1)\Phi(l,l-1) \\ &\quad \times \Phi_n(l-1,k)\big]^\tau\big[P_n(l \mid l-1) - P_n(l \mid l)\big] \\ &\quad \times \big[P_n^{-1}(l \mid l-1)\Phi(l,l-1)\Phi_n(l-1,k)\big]. \quad (48) \end{aligned}$$

Since from the standard form of the Kalman filter, we have $H^\tau(l)P_z^{-1}(l \mid l-1)\tilde{z}(l \mid l-1) = P^{-1}(l \mid l-1)$ $[\hat{x}(l \mid l) - \hat{x}(l \mid l-1)]$ [22], and from the centralized smoother $M_n(l,k) = M_n(l-1,k) + \Phi_n^\tau(l-1,k)\Phi^\tau(l,l-1)H^\tau(l)P_{zn}^{-1}(l \mid l-1)\tilde{z}_n(l \mid l-1)$ [7], we can claim that, by combining the two of them, the following relation holds true:

$$\begin{aligned} M_n(l,k) &= M_n(l-1,k) + \Phi_n^\tau(l-1,k)\Phi^\tau(l,l-1) \\ &\quad \times P^{-1}(l \mid l-1)[\hat{x}(l \mid l) - \hat{x}(l \mid l-1)] \quad (49) \\ M_n(l,k) &= M_n(l-1,k) + \big[P_n^{-1}(l \mid l-1)\Phi(l,l-1) \\ &\quad \times \Phi_n(l-1,k)\big]^\tau \\ &\quad \times [\hat{x}(l \mid l) - \hat{x}(l \mid l-1)]. \quad (50) \end{aligned}$$

REFERENCES

[1] E. A. Robinson, “Predictive decomposition of seismic traces,” *Geophysics*, vol. 22, no. 4, pp. 767–778, 1957.

[2] J. M. Mendel, “Single channel white-noise estimators for deconvolution,” *Geophysics*, vol. 43, pp. 102–124, 1978.

[3] J. M. Mendel, J. Kormylo, F. Aminzadeh, J. S. Lee, and F. Habibi-Ashrafi, “A novel approach to seismic signal processing and modeling,” *Geophysics*, vol. 46, no. 10, pp. 1398–1414, 1981.

[4] A. K. Mahalanabis, S. Prasad, and K. P. Mohandas, “A fast optimal deconvolution algorithm for real seismic data using Kalman predictor model,” *IEEE Trans. Geosci. Remote Sensing*, vol. GE-19, pp. 216–221, Mar. 1981.

[5] ——, “On the application of fast Kalman algorithm to adaptive deconvolution of seismic data,” *IEEE Trans. Geosci. Remote Sensing*, vol. GE-21, pp. 426–433, May 1983.

[6] J. Kormylo and J. M. Mendel, “Maximum-likelihood seismic deconvolution,” *IEEE Trans. Geosci. Remote Sensing*, vol. GE-21, pp. 72–83, Jan. 1983.

[7] S. K. Katsikas, “Optimal algorithms for seismic signal processing in oil exploration,” Ph.D dissertation, Univ. Patras, Patras, Greece, 1987, in Greek.

[8] D. G. Lainiotis, S. K. Katsikas, and S. D. Likothanassis, “Optimal seismic deconvolution,” *Signal Process.*, vol. 15, no. 4, pp. 375–404, 1988.

[9] J. M. Mendel, “Comments on ‘Optimal seismic deconvolution’,” *Signal Process.*, vol. 18, no. 4, pp. 447–448, 1989.

[10] D. G. Lainiotis, S. K. Katsikas, and S. D. Likothanassis, “Adaptive deconvolution of seismic signals: Performance, computational analysis, parallelism,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1715–1734, Nov. 1988.

[11] D. G. Lainiotis, “Optimal nonlinear estimation,” *Int. J. Contr.*, vol. 14, pp. 1137–1148, 1971.

[12] ——, “Optimal linear smoothing: Continuous data case,” *Int. J. Contr.*, vol. 17, no. 5, pp. 921–930, 1973.

[13] ——, “Partitioning: A unifying framework for adaptive systems I: Estimation,” *Proc. IEEE*, vol. 64, pp. 1126–1143, Aug. 1976.

[14] D. Andrisani II and C. F. Gau, “Multistage linear estimation using partitioning,” *IEEE Trans. Automat. Contr.*, vol. AC-30, pp. 182–185, Feb. 1985.

[15] K. Watanabe, *Adaptive Estimation and Control: Partitioning Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[16] H. R. Hashemipour, S. Roy, and A. J. Laub, “Decentralized structures for parallel Kalman filtering,” *IEEE Trans. Automat. Contr.*, vol. 33, pp. 88–94, Jan. 1988.

[17] D. G. Lainiotis, K. N. Plataniotis, and C. J. Charalampous, “Distributed computing filters: Multisensor marine applications,” in *IEEE OCEANS’92 Proc.*, pp. 265–270.

[18] S. K. Katsikas, S. D. Likothanassis, and D. G. Lainiotis, “On the parallel implementation of linear Kalman and Lainiotis filters and their efficiency,” *Signal Process.*, vol. 25, pp. 289–306, 1991.

[19] K. N. Plataniotis, “Distributed parallel processing state estimation algorithms,” Ph.D. dissertation, Florida Inst. Technol., Melbourne, 1994.

[20] J. Willner, C. B. Chang, and K. P. Dunn, “Kalman filter algorithms for a multi-sensor system,” in *Proc. 15th Conf. Decision Contr.*, 1976, pp. 570–574.

[21] L. A. Crowl, “How to measure, present and compare parallel performance,” *IEEE Trans. Parallel Distrib. Technol.: Syst. Applicat.*, vol. 2, pp. 9–26, Jan. 1994.

[22] J. V. Candy, *Signal Processing: The Model-Based Approach*. New York: McGraw Hill, 1986.

Konstantinos N. Plataniotis (S’90–M’92–S’93–M’95) received the B.Eng. degree in computer engineering and information science from the University of Patras, Patras, Greece, in 1988 and the M.S. and Ph.D. degrees in electrical engineering from the Florida Institute of Technology (Florida Tech), Melbourne, in 1992 and 1994, respectively.

He was with the Computer Technology Institute (C.T.I.), Patras, Greece, from 1989 to 1991 and with the Digital Signal & Image Processing Laboratory, Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ont., Canada, from 1995 to 1997. Since August 1997, he has been with the Department of Mathematics, Physics, and Computer Science, Ryerson Polytechnic University, Toronto. His research interests include multimedia signal processing, computer graphics, color image processing, video coding, fuzzy logic, and stochastic estimation.

Dr. Plataniotis is a member of the Technical Chamber of Greece.

Sokratis K. Katsikas (S'82–M'84–A'87) received the electrical engineering degree from the University of Patras, Patras, Greece, the M.Sc. degree in electrical and computer engineering from the University of Massachusetts at Amherst, and a the Ph.D. degree in computer engineering from the University of Patras.

He has held teaching and research positions with the University of Massachusetts at Amherst, University of Patras, Computer Technology Institute, Patras, Office of Naval Research of the Hellenic Navy, and Technological Institute of Athens, Athens, Greece. Since 1990, he has been an Associate Professor of informatics with the Department of Mathematics of the University of Aegean, Karlovassi, Greece. He has been involved with several funded R&D projects in the areas of artificial intelligence, computer security, and robotics. He has authored more than 80 papers and conference presentations in areas that include geophysical signal processing, estimation theory, adaptive control, artificial intelligence, and computer security.

Dr. Katsikas is a member of the Technical Chamber of Greece.

Demetrios G. Lainiotis received the Ph.D. degree from Purdue University, West Lafayette, IN, in 1964.

He has served as Professor of Electrical Engineering and Associate Director of the Electronics Research Center, the University of Texas at Austin, and as Professor and Chairman of the Electrical Engineering Department, the State University of New York at Buffalo. He was also the Chair of Pattern Recognition, the University of Patras, Patras, Greece, from 1980 to 1989. He received the Harris Corporation endowed Professorship at Florida Institute of Technology, Melbourne, during 1990–1994. Since 1994, he has been retired.

Dr. Lainiotis has served as Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, *International Journal of Information Sciences*, *Journal of Pattern Recognition*, *Journal of Neural*, *Parallel and Scientific Computations*, and *Journal of Mathematical Problems in Engineering*. He has also served as Guest Editor of the PROCEEDINGS OF THE IEEE: Special issue on Adaptive Systems, August 1976, *Journal of Information Sciences*, Fall 1974, and IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING: Special issue on Geophysical Signal Processing, January 1997.

Anastasios N. Venetsanopoulos (S'66–M'69–SM'79–F'88) received the B.Eng. degree from the Technical University of Athens (NTU), Athens, Greece, in 1965 and the M.S., M.Phil., and Ph.D. degrees in electrical engineering from Yale University, New Haven, CT, in 1966, 1968, and 1969, respectively.

He joined the University of Toronto, Toronto, Ont., Canada, in September 1968, where he has been a Professor in the Department of Electrical and Computer Engineering since 1981. He has served as Chairman of the Communications Group and Associate Chairman of the Department of Electrical Engineering. He was on research leave at the Federal University of Rio de Janeiro, Brazil, the Imperial College of Science and Technology, London, U.K., the National Technical University of Athens, Swiss Federal Institute of Technology, Lausanne, Switzerland, and the University of Florence, Florence, Italy, and was Adjunct Professor at Concordia University, Montreal, P.Q., Canada. He has served as Lecturer in 130 short courses to industry and continuing education programs and as Consultant to several organizations; he is a contributor to 24 books, co-author of *Nonlinear Filters in Image Processing: Principles and Applications* (Norwell, MA: Kluwer, 1990) and of *Artificial Neural Networks: Learning Algorithms, Performance Evaluation and Applications* (Norwell, MA: Kluwer, 1993) and has published over 500 papers on digital signal and image processing and digital communications. Present research interests include multimedia signal processing, linear M-D and nonlinear filters, processing of multispectral (color) images and image sequences, telecommunications, and image compression, in particular, the development of efficient techniques for multispectral image transmission, restoration, filtering, and analysis.

Dr. Venetsanopoulos has served as Chairman on numerous boards, councils, and technical conference committees. He has been a Guest Editor or Associate Editor for several journals and the Editor of the *Canadian Electrical Engineering Journal* from 1981 to 1983. He is a Fellow of the EIC and was awarded an Honorary Doctorate from the National Technical University of Athens for his "contribution to engineering" in October 1994.