

# Multisensor Estimation: New Distributed Algorithms

K. N. PLATANIOTIS<sup>a,\*</sup> and D.G. LAINIOTIS<sup>b</sup>

<sup>a</sup>*Postdoctoral Fellow, Digital Signal/Image Processing Laboratory, Department of Electrical and Computer Engineering., University of Toronto, 10 King's College-Road, Toronto, Ontario, Canada, M5S 1A4;*  
<sup>b</sup>*Intelligent Systems Technology, 3207 S. Hwy. A1A, Melbourne Beach, FL-32951, USA*

*(Received 15 January 1996; In final form 18 March 1996)*

The multisensor estimation problem is considered in this paper. New distributed algorithms, which are able to locally process the information and which deliver identical results to those generated by their centralized counterparts are presented. The algorithms can be used to provide robust and computationally efficient solutions to the multisensor estimation problem. The proposed distributed algorithms are theoretically interesting and computationally attractive.

*Keywords:* estimation; data fusion; distributed algorithms; Kalman filter; Linear Lainiotis Filter

## I. INTRODUCTION

The multisensor estimation problem is of considerable practical significance in applications such as, geophysical data processing for oil exploration, process monitoring and surveillance, command, control and communication systems, underwater target tracking and air traffic control systems. In a distributed multisensor environment a large number of dif-

---

\*Corresponding author: Tel.: (416) 978-6845. Fax: (416) 978-4425. E-mail: kostas@dsp.toronto.edu.

ferent sensors, each equipped with its own processing facility, is utilized to collect data. Then the overall information is used to estimate, detect or control the state of a dynamic system.

Consider a dynamic system described by the following linear, discrete time, state transition equation:

$$x(k + 1) = \Phi(k + 1, k)x(k) + w(k) \quad (1)$$

where  $x(k)$  is the  $n$ -dimensional state of the system at time  $k$ ,  $\Phi(k + 1, k)$  is the  $(n \times n)$ -state transition matrix from time  $k$  to  $k + 1$  and  $w(k)$  is the associated process noise, modeled as an uncorrelated white sequence with covariance  $Q(k)$ . It is assumed that the initial state vector value  $x(0)$  is Gaussian with mean value  $\hat{x}(0|0)$  and variance  $P(0|0)$ . The initial value is considered uncorrelated with the plant noise  $w(k)$ .

In classical estimation theory, a network of sensors take measurements  $z(k)$  of the system's state according to the following linear equation:

$$z(k + 1) = H(k + 1)x(k + 1) + v(k + 1) \quad (2)$$

where  $H(k)$  is the  $(m \times n)$ -observation matrix and  $v(k)$  is the associated observation noise modeled as an uncorrelated white Gaussian sequence with covariance  $R(k)$ . It is assumed that,  $E[v(j)w(k)] = 0$  and  $E[x(0)v(j)] = 0$ .

In multisensor estimation problems an observation model other than the one used in (2) is required to describe the measurement assimilation process. Under the distributed/decentralized estimation scenario introduced in [5] a system comprising  $r$  sensors with the composite observation model of (2) is considered. The observation vector  $z(k)$  is partitioned into  $r$  sub-vectors of dimension  $m_i$  corresponding to the observations made by each local sensor system.

$$z^\tau(k) = [z_1^\tau(k), z_2^\tau(k), \dots, z_r^\tau(k)], \quad (3)$$

The observation matrix is also partitioned into sub-matrices corresponding to these observations as follows:

$$H^\tau(k) = [H_1^\tau(k), H_2^\tau(k), \dots, H_r^\tau(k)], \quad (4)$$

and thus, the observation noise vector is partitioned accordingly

$$v^\top(k) = [v_1^\top(k), v_2^\top(k), \dots, v_r^\top(k)], \quad (5)$$

with the assumption that these partitions are uncorrelated

$$R(k) = E[v(k)v(k)^\top] = \text{blockdiag}(R_1^\top(k), \dots, R_r^\top(k)). \quad (6)$$

The uncorrelatedness of the measurement noise partitions is justified when the output partitioning represents different sensors [14]. This is a valid assumption in a large number of applications, such as seismic deconvolution for oil exploration, where the sensors (geophones) used to capture the seismic reflection are usually placed in clusters which are mutually independent. In any case, the designer can always find a partition of the measurement array which satisfies (6). Thus, despite possible constraints on the number of local systems ( $r$  partitions) the authors would like to emphasize that the results obtained herein are immediately applicable to any sensor partitioning scenario, since the distributed estimators discussed here allow for arbitrary partitioning of the observation array.

Under these assumptions the sensor model now consists of  $r$  equations in the form

$$z_i(k+1) = H_i(k+1)x(k+1) + v_i(k+1), \quad (7)$$

where,  $z_i(k+1)$  is the  $m_i$ -dimensional measurement vector at the local processor system  $i$ ,  $H_i(k+1)$  is the  $m_i \times n$  local observation matrix, and  $v_i(k+1)$  is the  $m_i$  local measurement noise vector. In this analysis, without loss of generality, it is assumed that:

$$m = \sum_{i=1}^r m_i, \quad (8)$$

Given all local measurement records  $L_i = (z_i(1), z_i(2), \dots, z_i(k+1))$ ,  $i = 1, 2, \dots, r$  available at the associated local sensor groups and the corresponding local estimates  $\hat{x}_i(k+1|k+1)$  the optimal estimate  $\hat{x}(k+1|k+1)$  is required (Fig. 1).

Two different approaches can be utilized in order to obtain the required estimates in such a multisensor environment.

- First, the so called centralized approach where all measurements from the different sensor devices are transmitted from their local sensing positions to a central processor. An algorithm similar to those devised for single sensor systems can be used to process and to interpret the measurements. The well known and widely used *Kalman Filter* (KF) and the *Per Step Linear Lainiotis Filter* (PSLIF) have been used extensively for this task [13]. However, such an approach introduces severe computational load, and considerable communication overhead and delays since all measurements have to be transferred to a central facility for processing.
- Secondly, a distributed/decentralized approach where the data obtained by the different local sensor subsystems are processed locally. The resulting local estimates are then transmitted to a central facility, which combines the information received (data fusion) to generate the overall estimate. Such an architecture relieves the computational burden on the central processor, increases robustness and survivability and delivers enhanced processing speed.

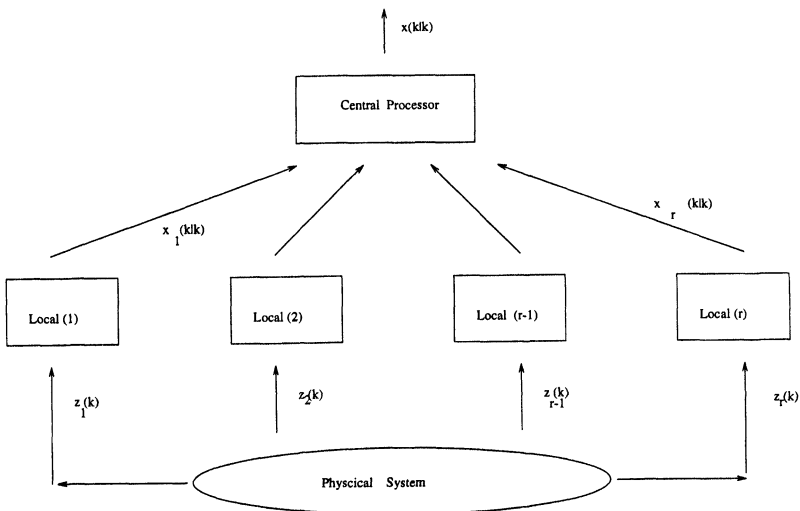


FIGURE 1 Distributed State Estimation.

Thus, a number of distributed estimation schemes have been developed and implemented recently. In [5] Hashemipour, Roy and Laub derived a distributed Kalman estimator. The behavior of their distributed estimator when reduced state space models are used to model the local sensor subgroups has been examined in [14]. A similar decentralized scheme based on the square root of the Kalman filter was introduced and studied by Carson in [4]. Finally, an decentralized estimator similar to the one introduced in [5] was discussed in [3]. A different approach was followed in [8], [13] where new distributed estimators and detectors based on the *Per Step Linear Lainiotis Filter* (PSLLF) [9], [7] have been developed and analyzed.

In this paper, we propose new and improved distributed algorithms for the multisensor estimation problem. The main objective is to develop estimation schemes, which provide redundancy and scalability, increase robustness and achieve improved performance in terms of computational speed.

The proposed algorithms are more computationally efficient than those already in use and can deliver excellent results in a fraction of the processing time required by other distributed schemes. In addition, the new algorithms are able to locally process information and can provide optimal results even if suboptimal initial values and process noise intensity values are used in the different local subsystems.

The paper is organized as follows: In section II, a new distributed estimator, named Distributed Lainiotis Filter (DLF) is introduced and analyzed. Its properties and its advantages over the commonly used Distributed Kalman Filter (DKF) are also discussed in section II. In section III, a two-stage distributed state estimator, named Two Stage Distributed Lainiotis Filter (TSDLF) is analyzed. Motivation, design characteristics and comparisons with other estimators are discussed in detail. Computational and implementation issues are the subject of section IV. An important multisensor problem, namely, seismic deconvolution for oil exploration is also discussed in section IV. Finally, section V summarizes our conclusions.

## II. THE DISTRIBUTED LAINIOTIS FILTER

The first distributed state estimator introduced is the Distributed Lainiotis Filter (DLF). The new algorithm can be considered as the distributed

equivalent of the Per Step Linear Lainiotis Filter (PSLLF). The new estimator can be viewed as a two-filter algorithm. Namely, it utilizes a nominal Distributed Kalman Filter with zero initial conditions which operates on the different local subsystems and a correction mechanism at the central processor. For the multisensor problem discussed in section I, the equations of the DLF are as follows:

**THEOREM II.1** *Distributed Lainiotis Filter (DLF)*

- Filter: central processor calculations:

$$\hat{x}(k+1|k+1) = \hat{x}_n(k+1|k+1) + \Phi_n(k+1, k)P_r(k|k+1)\hat{x}_r(k|k+1) \quad (9)$$

$$\hat{x}_r(k|k+1) = [M_n(k+1) + P^{-1}(k|k)\hat{x}(k|k)] \quad (10)$$

$$P(k+1|k+1) = P_n(k+1|k+1) + \Phi_n(k+1, k)P_r^{-1}(k|k+1)\Phi_n^T(k+1, k) \quad (11)$$

$$P_r(k|k+1) = [O_n(k+1) + P^{-1}(k|k)] \quad (12)$$

$$P_n^{-1}(k+1|k+1) = Q^{-1}(k) + B_i(k+1) \quad (13)$$

$$B_i(k+1) = \sum_{i=1}^r (P_{ni}^{-1}(k+1|k+1) - Q^{-1}(k)) \quad (14)$$

$$F_n(k+1) = Q^{-1}(k)\Phi(k+1, k) \quad (15)$$

$$\Phi_n(k+1, k) = P_n(k+1|k+1)F_n(k+1) \quad (16)$$

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

$$O_n(k+1) = F_n(k+1)^T[Q(k) - P_n(k+1|k+1)]F_n(k+1) \quad (17)$$

$$M_n(k+1) = F_n(k+1)\hat{x}_n(k+1|k+1) \quad (18)$$

- Filter local processor calculations (for the  $i^{th}$  local subsystem):

$$P_{ni}^{-1}(k+1|k+1) - Q^{-1}(k) = H_i^c(k+1)R_i^{-1}(k+1)H_i(k+1) \quad (19)$$

$$\begin{aligned} P_{ni}^{-1}(k+1|k+1)\hat{x}_{ni}(k+1|k+1) - P_{ni}^{-1}(k+1|k)\hat{x}_{ni}(k+1|k) \\ = H_i^c(k+1)R_i^{-1}(k+1)z_i(k+1) \end{aligned} \quad (20)$$

The proof is given in Appendix A.

The Distributed Lainiotis Filter (DLF) introduced here is superior to the commonly used Distributed Kalman Filter (DKF) [5], [13] from both a theoretical and computational point of view. Namely:

1. As can be seen from (9)–(11), the new DLF constitutes a two-filter formula which is given in terms of: (i) a forward nominal Distributed Kalman Filter with zero initial conditions and (ii) a second filter which operates only on the central processor and is used for the calculations in (10), (12), (16)–(18). Thus, DKF is part of the new filter although in a special form due to its re-initialization at every step with zero initial conditions.
2. It must be noted that theorem 1 corresponds to the general time varying, linear dynamic, discrete time, state space model. However, in important application problems, such as the problem of seismic deconvolution for oil exploration the process model is time invariant. For such a case, DLF simplifies further by substituting the time varying matrices  $\Phi(k, k-1)$ ,  $H_i(k)$ ,  $R_i(k)$ ,  $Q(k)$  with the time invariant matrices  $\Phi$ ,  $H_i$ ,  $R_i$  and  $Q$ . Thus, due to the nature of the new algorithm equations (12)–(16), (17) and (19) become time invariant and require only one calculation at the beginning of the corresponding process. After that, the results can be stored in the memory and can be used in the recursive calculations of the algorithm immediately, without any further processing. In this way, the only calculations required at the local level are those of (20). On the contrary, the DKF for the time invariant model is identical to the that required for a time-varying system due to the fact that the DKF equations have exactly the same form for time varying, time invariant or periodic state space models [13].
3. The Distributed Kalman Filter (DKF) although is theoretically the optimal (in the mean square error sense) estimator is known to be numerically unreliable due to the matrix inverse operations required

for its implementation [5]. To overcome this limitation complex and computationally intensive solutions, such as U-D factorization or square root formulations have been proposed [4], [14]. On the contrary, DLF is numerically stable retaining at the same time the simplicity of a classical Kalman-like recursive filter. Indeed, as it can be seen from (11) and (17), the overall covariance update formula has a natural quadratic form which prevents the propagation of numerical errors encountered during the inversion process, and guarantees the positive nature of the error covariance matrix. Thus, due to its quadratic form and without the introduction of any numerical transformation our distributed algorithm can guarantee numerical stability during the implementation process.

4. In our filter, the optimal local estimate at each of the different local sub-systems is not required by the central processor. If however, this estimate is requested it can be computed by each local processor in parallel with the calculations in the central agent with out additional computational delays. These memoryless operations can be implemented using a very simple processor or even an optical device. Thus, using DLF there is no need to install expensive computing facilities on local sensing locations. On the contrary, the DKF requires the calculations of the optimal local estimates in a sequential form. At this point it must be emphasized the theoretical as well as practical importance of this formulation. Conventional distributed estimators, such as the Distributed Kalman Filter (DKF) has to rely on reduced-order suboptimal local models in order to improve the computational efficiency of the overall algorithm. However, the new DLF it can use the actual local state-space model and still be computationally efficient. Its unique two-filter structure, with the correction mechanism of (10), (12) and (16)–(18), allows the utilization of a simplified nominal Distributed Kalman Filter at the different local systems. These nominal filters since they are re-initialized at every step with zero nominal initial conditions have minimal computational complexity and require simple devices, such as optical arrays for their implementations. Finally, due to the structure of the proposed algorithm there is no two-way communication between the local processors and the central computing facility. Because bi-directional communication between slower local stations and the central agent is not desirable in a highly parallel processing environment, DLF has been designed with the lowest possible communication requirements.



### III. A DISTRIBUTED TWO-STAGE ESTIMATOR

Both the DKF and the DLF discussed above are based on the assumptions discussed in section I. However, in a actual distributed/decentralized environment it is not realistic to expect that each local submodule will have complete and accurate knowledge of the actual state space model.

Decentralized and distributed algorithms available today [4], [14] allows the introduction of reduced-order suboptimal local models in the system. Their suboptimal models are usually scaled-down versions of the equation of the system, primarily because the existing distributed algorithms can not cope with the computational complexity of the model. However, in the state estimation problem is fairly easy for each local system to obtain a copy of the exact state transition equation, which is determined by the laws that govern the physical phenomenon. On the other hand, it is unrealistic to assume that the initial values and the noise process intensity values will be perfectly known. It is well known that in the estimation problem the level of process noise and the initial conditions must frequently be determined on a trial and error basis. Their knowledge is essential for the development of the recursive procedure that estimates the states of the system. Moreover, initial conditions are crucial to the behavior of a dynamic system during its transient response and of course determine the convergence of the estimator [9].

Thus, flexibility on the selection of these parameters in the local level and robustness of the estimator in erroneous value selections is required. Our main objective is to develop a flexible distributed algorithm, which can handle local information efficiently when the exact initial conditions and process noise statistics are not available. A two stage distributed estimator, is introduced here to study the effects of the initial settings and the noise variances in a decentralized data fusion network. The cascade structure of the proposed distributed estimator provides the natural framework for studying the effect of initial covariance and process noise intensities not only in linear estimation but in the dual optimal control problem as well. It must be emphasized at this point that the proposed estimator discussed in this paper is the only algorithm which can be handle this kind of uncertainty. To the best of the authors' knowledge there is no other distributed or decentralized estimator available in the literature, which is capable of that.

To formalize the uncertainty on the initial conditions and the noise pro-

cess, it is assumed here that the initial state condition and the process noise vectors are decomposed into two statistically independent parts. Namely,

$$x(0) = x_n + x_r$$

$$w(k) = w_n(k) + w_r(k)$$

where  $x_n$  and  $x_r$  are Gaussian with mean values  $\hat{x}_n$  and  $\hat{x}_r$  respectively. Furthermore,  $x_n$  and  $x_r$  have variances selected so that:

$$P(0|0) = P_n + P_r$$

The process noise parts  $w_n(k)$ ,  $w_r(j)$  are considered zero mean Gaussian and statistically independent for all  $j$  and  $k$ . It also assumed that  $w_n(k)$  and  $w_r(j)$  have variances  $Q_n(k)$  and  $Q_r(j)$  which are selected so that:

$$Q(k) = Q_n(k) + Q_r(k)$$

Using this setting, the initial state-vector and the process noise covariance are decomposed into two parts, with the remainder quantities  $x_r$ ,  $P_r$  and  $w_r$  considered as parameters to which we adapt. The equations of the (TSDLF) which estimates, under these assumptions, the state of the system in (1) using the network of distributed sensors introduced in (7) are summarized in the next theorem.

**THEOREM III.1** *Two Stages Distributed Lainiotis Filter (TSDLF)*

## **Central Processor**

### ***Optimal estimator***

$$\hat{x}(k+1|k+1) = \hat{x}_n(k+1|k+1) + U_n(k+1)\alpha_r(k+1|k+1) \quad (21)$$

$$P(k+1|k+1) = P_n(k+1|k+1) + U_n(k+1)P_r(K+1|k+1)U_n^T(k+1) \quad (22)$$

**Nominal estimator**

$$\hat{x}_n(k+1|k+1) = P_n(k+1|k+1)(P_n^{-1}(k+1|k)\hat{x}_n(k+1|k) + A_i(k+1)) \quad (23)$$

$$A_i(k) = \sum_{i=1}^r (P_{ni}^{-1}(k+1|k+1)\hat{x}_{ni}(k+1|k+1) - P_{ni}^{-1}(k+1|k)\hat{x}_{ni}(k+1|k)) \quad (24)$$

$$P_n^{-1}(k+1|k+1) = P_n^{-1}(k+1|k) + B_i(k+1) \quad (25)$$

$$B_i(k+1) = \sum_{i=1}^r (P_{ni}^{-1}(k+1|k+1) - P_{ni}^{-1}(k+1|k)) \quad (26)$$

$$\hat{x}_n(k+1|k) = \Phi(k+1, k)\hat{x}_n(k|k) \quad (27)$$

$$P_n(k+1|k) = \Phi(k+1, k)P_n(k|k)\Phi^T(k|k) + Q_n(k) \quad (28)$$

$$U_n(k+1) = P_n(k+1|k+1)P_n^{-1}(k+1|k) \quad (29)$$

**Remainder Estimator**

$$\hat{x}_r(k+1|k+1) = P_r(k+1|k+1)(P_r^{-1}(k+1|k)\hat{x}_r(k+1|k) + P_n^{-1}(k+1|k)((\hat{x}_n(k+1|k+1) - \hat{x}_n(k+1|k))) \quad (30)$$

$$\hat{x}_r(k+1|k) = \Phi(k+1, k)U_n(k)\hat{x}_r(k|k) \quad (31)$$

$$P_r^{-1}(k+1|k+1) = P_r^{-1}(k+1|k) + P_n^{-1}(k+1|k)P_n(k|k)(P_n^{-1}(k+1|k+1) - P_n^{-1}(k+1|k)) \quad (32)$$

$$P_r(k+1|k) = (\Phi(k+1, k)U_n(k))P_r(k|k)(\Phi(k+1, k)U_n(k))^T + Q_r(k) \quad (33)$$

## Local Processor

For each one of the  $i^{\text{th}}$  local subsystems

$$P_{ni}^{-1}(k+1|k+1) - P_{ni}^{-1}(k+1|k) = H_i^T(k+1)R_i^{-1}(k+1)H_i(k+1) \quad (34)$$

$$P_{ni}^{-1}(k+1|k+1)_{ni}(k+1|k+1) - P_{ni}^{-1}(k+1|k)_{ni}(k+1|k) = H_i^T(k+1)R_i^{-1}(k+1)z_i(k+1) \quad (35)$$

The proof is given in Appendix B.

1. The two-stage distributed estimator introduced above constitute a family of realizations of the optimal distributed estimator, one for each set of nominal conditions  $x_n$ ,  $P_n$  and  $Q_n$ , or equivalently one for each initial state vector and process noise partitioning. It must be emphasized at this point, that the Decentralized Kalman Filter (DKF) [5], [13] is a member of this family for nominal initial conditions equal to actual initial conditions and nominal noise process covariance equal to the actual one. As such, all the decentralized filters discussed previously in the literature can be viewed as special cases of the TSDLF. In a similar manner, (30) and (32) constitute a class of two-filter smoothing formulas, one for each possible partitioning of the state vector  $x(0)$  and the process noise covariance  $Q$ . In particular, the Mayne-Fraser smoothing formula [12] is a member of this class corresponding to the choice of zero nominal initial conditions [9], [6].
2. TSDLF's capability to incorporate arbitrary initial conditions in both the local models as well as in the central processor is of paramount importance in the distributed estimation of multisensor systems. The estimator allows simplified local models based on suboptimal initial conditions and process noise to be integrated in a distributed processing system. At the central processor the estimator compensates for the suboptimal local models used. When a time invariant state space system is used to describe the physical process the steady-state covariance of the local subsystem can be used as the initial covariance of the local estimator. Thus, each one of the local nominal Distributed Kalman filters used in (34)–(35) reduces to a time invariant Wiener filter with considerable computational savings [2].

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 this is the most commonly used configuration in practice. However, our distributed formulation is immediately applicable to any type of distributed or decentralized data fusion network. Specifically, in case that 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 then this node can act as central node and generate the overall estimate, considering all the other nodes as local sensing groups, with exactly the same equations.

## **IV. DISTRIBUTED ALGORITHMS: MULTISENSOR APPLICATIONS**

### **A. Computational Requirements of the Distributed Algorithms**

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], [11]. In this work, the rules defined there are used in the comparative evaluation of the Distributed Kalman Filter (DKF) and the Distributed Lainiotis Filter (DLF).

Due to the fact that the algorithms provide the solution in a recursive manner, the algorithms' total execution time is equal to the product of its per recursion computational complexity times the number of recursions required to obtain a solution. The comparisons introduced in this paper are on a per recursion basis. Two more assumptions are introduced in order to have a meaningful comparison among the algorithms.

1. 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 models are performed in parallel.

2. Secondly, the fundamental operations involved in the recursive solution are matrix and vector operations. A detailed analysis of the computational involved in the evaluation in such operations is provided in [7], [13]. The interested reader can 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:

$$Time = kX(4X(MULTS) + (ADDS) + 6X(DIVS) + 25X(SQRTS))$$

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 refer to any particular machine. Rather than that, they can be considered mean values of those coefficients commonly encountered. All the qualitative results presented in the sequence hold even if the weighting coefficients in the above formula are different for a specific computing platform [7], [13].

## B. Application to Seismic Deconvolution

Two distributed estimators, namely DKF and DLF, as well as their centralized counterparts, the Centralized Kalman Filter (CKF) and the Centralized Lainiotis Filter (CLF) [13] are compared using the framework compared above. The test problem selected, is that of seismic deconvolution for oil exploration.

In such a problem the signal received by a seismic sensor is described by the following convolution summation [10]:

$$z(k) = V(k)_R + n(k) = \sum_{j=1}^k V^+(k-j) + n(k) \quad (36)$$

where  $V(k)_R$  is the noise free seismic trace,  $h(k)$  is the measurement noise,  $V^+(k-j)$  is a sequence associated with the basic seismic wavelet and  $\mu(j)$  is the reflection coefficient sequence, which is modeled as white random noise.

The signal  $V(k)_R$  is a superposition of wavelet replica reflected from the interfaces of earth's sub-surface layers, while  $\mu(j)$  is related to interface reflection and transmission coefficients. Following the work in [10] we assume a state-space equivalent description of (36):

$$x(k + 1) = \Phi x(k) + \gamma \mu \quad (37)$$

$$z(k + 1) = Hx(k + 1) + v \quad (38)$$

where  $x(k)$  is the  $n$ -dimensional state vector,  $\Phi$  is the time invariant,  $n \times n$  state transition matrix,  $H$  is the time invariant,  $m \times n$  output matrix,  $z(k)$  is the  $m$ -dimensional measurement vector and  $\mu$ ,  $v(k)$  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  $(0|0)$  and variance  $P(0|0)$  and it is assumed independent of both  $\mu$  and  $v$ .

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 practice, 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 not placed in the same location but usually are allocated in different local subsystems (geophone clusters).

For such a time invariant state space model, the computational requirements for the implementation of the CLF algorithm are divided into two parts. The first part, called preliminary, summarizes the computational requirements for the operations that are calculated only once (or during the system's period) and the second part, named per step operations, summarizes the requirements for the per step calculations [13]. Similarly, the computational requirements of the DLF approach are also divided into two parts. As before, the first part, called preliminary, summarizes the computational requirements for the operations that are calculated only once (or during the system's period) and the second part, named per step operations, summarizes the requirements for the per step calculations. The computational requirements of the different algorithms (in terms of the normalized operations) when they applied to the problem of seismic deconvolution are summarized in table I.

TABLE I Seismic Deconvolution: Computational Requirements

<i>method</i>	<i>normalized operations</i>
CKF	$7.5n^3 + 6.5n^2 + 7.5n^2m + 7.5nm^2 + 13mn + 6m^2 + 2.5m^3 + 32.5m - n - 6$
CLF	$12.5n^3 + 23.5n^2 + 64n + 10nm - 12$
DKF	$25n^3 + 58n^2 + 123n + r(0.5n^2 + 1.5n) + 2.5m_i^3 + 7.5n^2m_i + 7.5nm_i^2 + 13nm_i$ $+ 6m_i^2 + 32.5m_i - 24$
DLF	$12.5n^3 + 34n^2 + 62n - 24 + (0.5n^2 + n)r + 5nm_i$

Up until now we have discussed the computational requirements of the different approaches to the problem when implemented in a centralized or distributed fashion. An answer however, to a fundamental question often posed by filter designers and practitioners was not given. Is the difficulty of implementation inherent in a distributed realization compensated by the gain in computing speed?

To answer this question and to compare the algorithms' computational efficiency, the total normalized operations required by centralized and distributed algorithms were plotted against measurement dimensionality (number of geophones) as well as against the number of local systems (geophone clusters). Total operations are computed using the framework explained in the previous paragraph. In this analysis the parameters are the number of channels/sensors  $m$ , and the local systems (geophone clusters)  $r$ . The values used for the state dimensionality ( $n$ ) (the order of the seismic wavelet) are  $n = 1$ ,  $n = 4$  and  $n = 9$  respectively. The number of geophone clusters, when is not a parameter, it is assumed to be  $r = 5$ .

Finally, a typical seismic deconvolution example is introduced. The time invariant wavelet used to describe the signal received by the seismic sensors is assumed of order  $n = 4$  [10]. To capture the seismic trace  $m = 1000$  sensors divided in  $r = 10$  local geophone clusters, are utilized. The computational requirements for the different estimators are summarized in table II.

From the analysis and the simulation studies the following conclusions can be drawn regarding our distributed state estimators:

1. Both the DLF and the TSDLF introduced in this work provide the optimal (in the minimum mean square error sense), estimate of the system state in (1). Thus, the algorithms are globally optimal and there is no loss of performance.



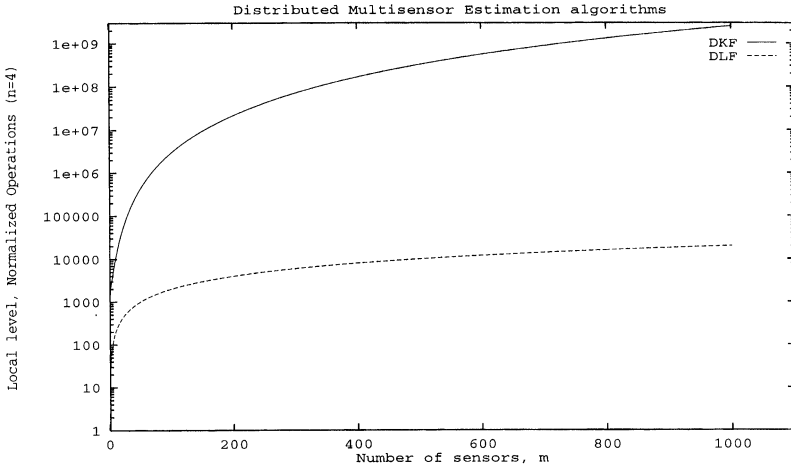


FIGURE 2 DKF vs. DLF: Seismic Deconvolution (Local Level Processing).

- Both the DLF and TSDLF require only simple, memoryless, non-recursive calculations for their implementation at the local level. As a result are computationally efficient, communicate less than the DKF with the central processor and can yield supreme input data rates and excellent computational data speeds as it can be seen from the simulation studies reported here (see tables I, II).

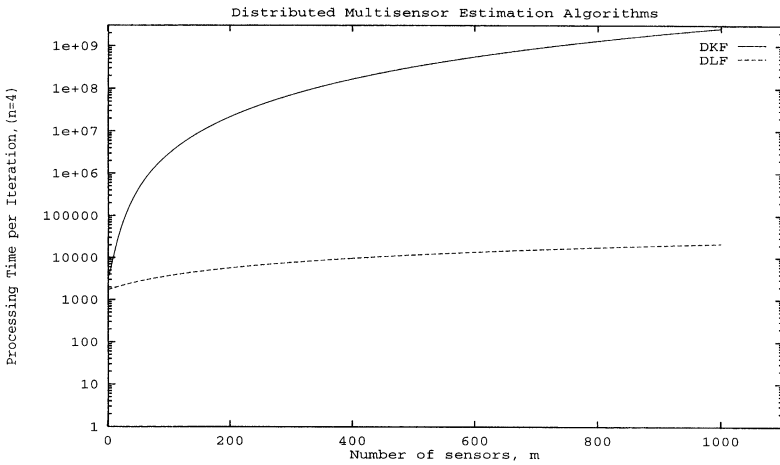


FIGURE 3 DKF vs. DLF: Seismic Deconvolution (Overall Processing Requirements).

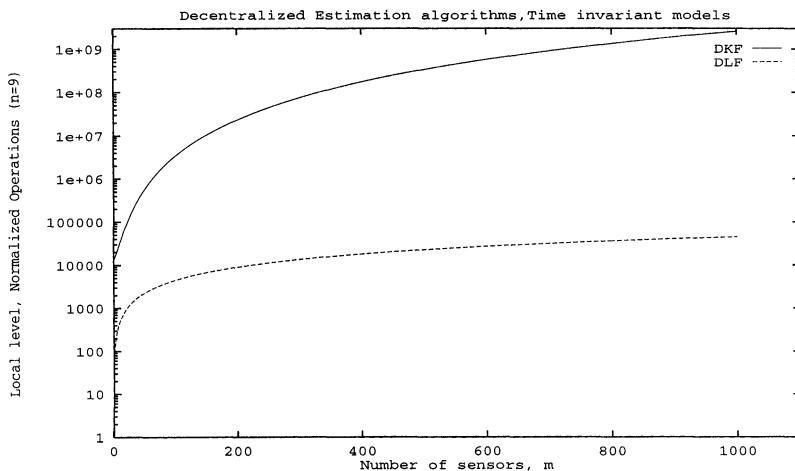


FIGURE 4 DKF vs. DLF: Seismic Deconvolution (Local Level Processing).

3. Due to their natural quadratic nature, both our filters are numerically stable and reduce the effect of ill conditioning without additional numerical transformations, such as U-D factorization.
4. The TSDLF introduced in this paper constitutes a family of distributed estimators and smoothers. Well known and extensively used distributed

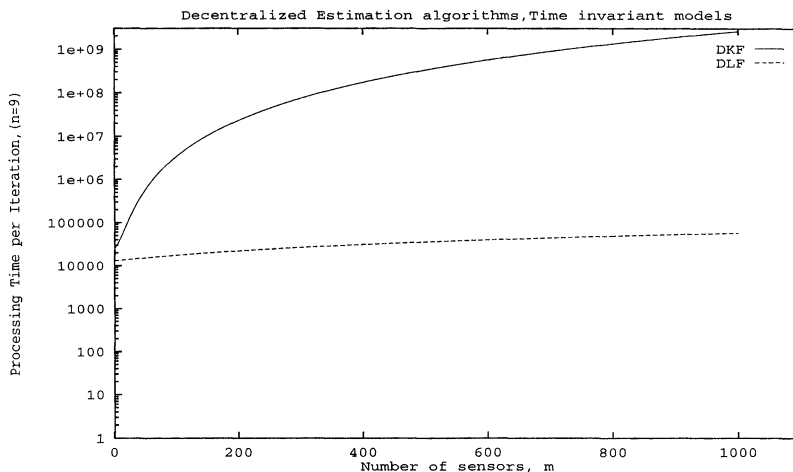


FIGURE 5 DKF vs. DLF: Seismic Deconvolution (Overall Processing Requirements).

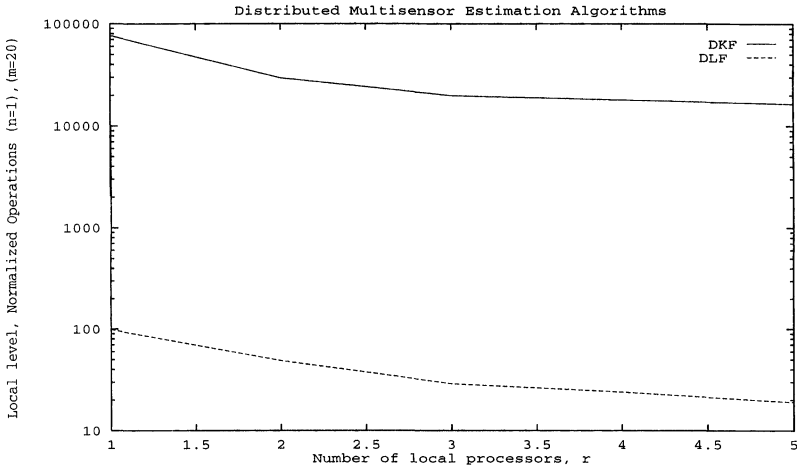


FIGURE 6 DKF vs. DLF: Seismic Deconvolution (Local Level Processing).

estimators, such as the Distributed Kalman Filter (DKF) [5] can be viewed as special case of TSDLF. More robust and computationally attractive filters can be derived from our framework using different partitions of the initial conditions and the process noise.

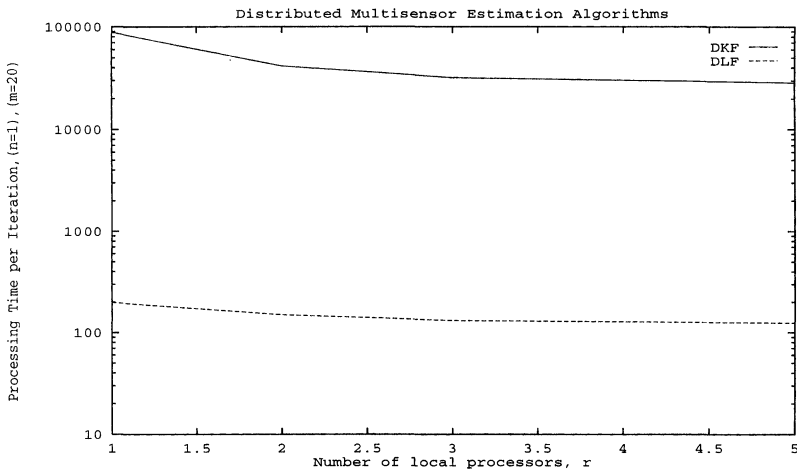


FIGURE 7 DKF vs. DLF: Seismic Deconvolution (Overall Processing Requirements).

TABLE II Seismic Deconvolution: Example

<i>method</i>	<i>normalized operations</i>
CKF	$2.4e+09$
CLF	$4.24e+04$
DKF	$2.61e+06$
DLF	$3.37e+03$

5. Both the DLF and the TSDLF are well suited for fault detection and isolation, since the local filters operate in parallel without interaction. In fact the new filters are more suitable than the DKF for this task since they use simpler local processor and utilize an additional correction mechanism at the central processing unit.
6. Regarding the computational requirements, the following conclusions can be drawn:
  - The computational requirements for estimating the state of a multisensor linear system, using a distributed approach are less than the computational requirements for solving the same problem using a centralized filter. The distributed algorithm delivers the same performance as its centralized counterpart at a fraction of the centralized implementation computational requirements. For the problem of seismic deconvolution discussed above the CLF provides better results than DKF. However, our DLF is the most computationally efficient method as we can see from the illustrative example in table II.
  - Using DLF only a few per step calculations are required at the local level. Thus, DLF has significant less computational requirements compared to the DKF or to centralized approaches where actual measurements are communicated from the dispersed sensor locations to the computing facility at each time step.

## V. CONCLUSION

In conclusion, two new distributed state estimators were introduced in this paper. The two new estimators provide modularity, flexibility and increased robustness. The TSDLF introduced here can be viewed as a generalizations of many well known distributed filters. Its capability to allow for suboptimal local estimators enhances the flexibility and robustness of

the estimation procedure. Although only hierarchical, distributed algorithms were discussed here, the new algorithms can be extended to arbitrary network topology and is not limited to the hierarchical network presented here. All the above make our distributed state estimators ideal for real-time multisensor applications, such as seismic deconvolution for oil exploration.

## VI. APPENDIX A

*Proof of Theorem II.1* DLF utilizes a Distributed Kalman Filter with zero initial conditions in order to generate the nominal results required by (11) and (14). Since the derivation of the DKF is trivial, we concentrate our attention in the derivation of the correction filter employed by the DLF.

The most important step is to obtain an expression for the nominal observability matrix  $O_n(k + 1)$  which does not use any matrix associated with the measurement equation. We start with an interim result, the expression for the filter transition matrix  $\Phi_n(k + 1, k)$ . By definition [9], [13]

$$\Phi_n(k + 1, k) = [I - K_n(k + 1)H(k + 1)]\Phi(k + 1, k) \quad (39)$$

Moreover,

$$P_n(k|k)Q^{-1}(k - 1) = [I - K_n(k)H(k)] \quad (40)$$

since the nominal Distributed Kalman Filter (DKF) utilized by our filter at each one of the  $r$  local systems is reinitialized at every step with zero initial conditions. Thus,  $P_{ni}(k|k - 1) = Q(k - 1)$  and  $\hat{x}_{ni}(k|k - 1) = 0$  with  $i = 1, 2, \dots, r$ . Therefore, the filter transition matrix for the distributed filter can be written as:

$$\Phi_n(k + 1, k) = P_n(k|k)Q^{-1}(k - 1)\Phi(k + 1, k) \quad (41)$$

After that, the nominal observability matrix [9], [13] is calculated as:

$$O_n(k + 1) = K_n(k + 1)H(k + 1)\Phi(k + 1, k) \quad (42)$$

$$K_m(k+1) = \Phi^\tau(k+1, k)H^\tau(k+1)P_{zn}^{-1}(k+1|k) \quad (43)$$

Simple algebraic calculations of the standard Kalman filter gain equation lead us to the following solution:

$$K_m(k+1) = \Phi^\tau(k+1, k)Q^{-1}(k)P_n(k+1|k+1)H^\tau(k+1)R^{-1}(k+1)H(k+1)\Phi(k+1, k) \quad (44)$$

From the nominal Distributed Kalman Filter, utilized in the DLF the expression for the estimation covariance is as follows:

$$P_n^{-1}(k|k) - Q^{-1}(k-1) = H^\tau(k)R^{-1}(k)H(k) \quad (45)$$

Utilizing this result the nominal observability matrix for the distributed solution can be written after some simple manipulations as follows:

$$F_n(k+1) = Q^{-1}(k)\Phi(k+1, k) \quad (46)$$

$$O_n(k+1) = F_n(k+1)^\tau[Q(k) - P_n(k+1|k+1)]F_n(k+1) \quad (47)$$

## VII. APPENDIX B

*Proof of Theorem III.1* As in the development of the Distributed Lainiotis Filter (DFL) presented in Appendix A, our main objective in the derivation of the Two Stage Distributed Lainiotis Filter is to re-write the equations of the optimal estimator without any measurement-related matrix. Thus, we modify the equations of the two-stage centralized filter introduced in [2]. A centralized Kalman filter is used in the first part of the estimator in [2] to provide nominal estimates. Therefore, the distributed form of the Kalman filter (DKF) can be used as nominal estimator in the new distributed two stage estimator. Since the DKF is fairly known and its equation can be found in [5], [13] we discuss here only the derivation of the distributed remainder estimator.

In the centralized algorithm of [2] the remainder state estimator has the following form:

$$\begin{aligned}\hat{x}_r(k|k) &= \\ (I - K_r(k)H(k))\Phi(k, k-1)U_n(k-1)\hat{x}_r(k-1|k-1) + K_r(k)\hat{z}_n(k|k-1) \\ &= U_r(k)\Phi(k, k-1)U_n(k-1)\hat{x}_r(k-1|k-1) + K_r(k)\hat{z}_n(k|k-1)\end{aligned}\quad (48)$$

Utilizing the formula reported in [2] for the gain of the remainder estimator the ‘correction term’ of the remainder filter is defined as:

$$\begin{aligned}K_r(k)\hat{z}_n(k|k-1) &= \\ P_r(k|k-1)H^T(k)P_{zn}^{-1}(k|k-1)\hat{z}_n(k|k-1)\end{aligned}$$

The nominal state estimate  $\hat{x}_n(k|k)$  is calculated by the Kalman filter as follows:

$$\hat{x}_n(k|k) = \hat{x}_n(k|k-1) + K_n(k)\hat{z}_n(k|k-1)\quad (49)$$

with the gain of the nominal Kalman filter defined as:

$$K_n(k) = P_n(k|k-1)H^T(k)P_{zn}^{-1}(k|k-1)\quad (50)$$

Combining the above two equations we can see that:

$$P_n^{-1}(k|k-1)(\hat{x}_n(k|k) - \hat{x}_n(k|k-1)) = H^T(k)P_{zn}^{-1}(k|k-1)\hat{z}_n(k|k-1)\quad (51)$$

Thus, the ‘correction term’ in the remainder estimator can be re-written as:

$$K_r(k)\hat{z}_n(k|k-1) = P_r(k|k-1)P_n^{-1}(k|k-1)(\hat{x}_n(k|k) - \hat{x}_n(k|k-1))\quad (52)$$

The filter transition matrix in [2] is defined as:

$$U_r(k) = (I - K_r(k)H(k))\quad (53)$$

and the covariance associated with the remainder estimate of the state as:

$$P_r(k|k) = U_r(k)P_r(k|k-1) \quad (54)$$

Combining the above two equations the following relation holds:

$$U_r(k) = P_r(k|k)P_r^{-1}(k|k-1) \quad (55)$$

Thus, we can obtain an expression for the remainder estimate  $\hat{x}_r(k|k)$  which does not require the explicit knowledge of any measurement equation.

$$\begin{aligned} \hat{x}_r(k|k) &= P_r(k|k) \\ &(P_r^{-1}(k|k-1)\hat{x}_r(k|k-1) + P_n^{-1}(k|k-1)(\hat{x}_n(k|k) - \hat{x}_n(k|k-1))) \end{aligned} \quad (56)$$

In a similar manner we will derive a recursive formula for covariance update which will not rely in any system measurement equation. The covariance update equation in the centralized algorithm derived in [2] is given as:

$$\begin{aligned} P_r(k|k) &= U_r(k)P_r(k|k-1) \\ P_r(k|k) &= (I - K_r H(k))P_r(k|k-1) \\ &= (I - P_r(k|k-1)H^T(k)(H(k)P_r(k|k-1)H^T(k) \\ &\quad + P_{zn}(k|k-1))^{-1}H(k))P_r(k|k-1) \\ &= P_r(k|k-1) - P_r(k|k-1)H^T(k)(H(k)P_r(k|k-1)H^T(k) \\ &\quad + P_{zn}(k|k-1))^{-1}H(k)P_r(k|k-1) \end{aligned} \quad (57)$$

Using the matrix inversion lemma in [1] (pp. 138–140) and the symmetry of the covariance matrix the following relation holds:

$$P_r^{-1}(k|k) = P_r^{-1}(k|k-1) + H^T P_{zn}^{-1}(k|k-1)H(k) \quad (58)$$



It is not hard to see that the above equation is a generalization of the covariance updating formula encountered in the information version of the Kalman filter. However, there is an important difference. The innovation covariance matrix of the nominal distributed filter is used instead of the actual noise covariance. Thus, the above is in fact not a simple equation but a family of covariance update equations each one of them matched to a specific partition of the initial conditions and the process noise. After, simple algebraic calculations

$$H^T(k)P_n^{-1}(k|k-1) = P_n^{-1}(k|k-1)P_n(k|k)H^T(k)R^{-1}(k) \quad (59)$$

and thus,

$$P_r^{-1}(k+1|k+1) = P_r^{-1}(k+1|k) + P_n^{-1}(k+1|k)P_n(k|k)(P_n^{-1}(k+1|k+1) - P_n^{-1}(k+1|k)) \quad (60)$$

It must be emphasized in this point that (56) and (60) although part of the TSDLF estimator, constitute a family of distributed smoothers. In fact, for the trivial case of the zero initial conditions as  $x_n = 0$ ,  $P_n = 0$ , and  $Q_n = 0$ , respectively, the distributed form of the Mayne-Fraser smoother can be obtained from (56) and (60). However, this smoother is just only one of the possible smoothers which can be derived using different initial quantities in (56) and (60). More important smoothers from a theoretical as well as from an application point of view can be derived utilizing our framework [6].

### **Acknowledgment**

The authors would like to thank the referees for their constructive and valuable comments.

### **References**

- [1] Anderson B.D.O., G.B. Moore, *Optimal Filtering*, Prentice Hall, 1979.
- [2] Andrisani II D., Ching-Fu Gau, 'Multistage linear estimation using partitioning,' IEEE Trans. on Automatic Control, Vol. AC-30, No. 2, 1985, pp. 182-185.
- [3] Berg T.M., H.F. Durrant-Whyte, 'Model distribution in decentralized multisensor data fusion,' Proceedings of the American Control Conference, 1991, pp. 946-949.

- [4] Carlson N.A., 'Federated Square Root Filter for Decentralized Parallel Processes,' IEEE Trans. on Aerospace and Electronic Systems, vol. AES-26, 1990, pp. 517–525.
- [5] Hashemipour H.R., S. Roy, A.J. Laub, 'Decentralized Structures for Parallel Kalman Filtering,' IEEE Trans. on Automatic Control, Vol. AC-33, 1988, pp. 88–94.
- [6] Govindaraj K.S., D.G. Lainiotis, 'A unifying framework for discrete linear estimation: Generalized partitioned algorithms', International Journal of Control, vol. IJC-28, no. 4, 1978, pp. 571–588.
- [7] Katsikas S.K., S.D. Likothanassis, D.G. Lainiotis, 'On the parallel implementation of linear Kalman and Lainiotis filters and their efficiency,' Signal Processing, vol. 25, 1991, pp. 289–306.
- [8] Lainiotis D.G., K.N. Plataniotis, C.J. Charalampous, 'Distributed Computing Filters: Multisensor Marine Applications,' IEEE OCEANS'92 Proceedings, 1992, pp. 265–270.
- [9] Lainiotis D.G., 'Partitioning: A unifying framework for adaptive systems I: Estimation,' Proceedings of IEEE, vol. 64, pp. 1126–1143, 1976.
- [10] Lainiotis D.G., S.K. Katsikas, S.D. Likothanassis, 'Optimal seismic deconvolution,' Signal Processing, vol. 15, 1988, pp. 375–404.
- [11] Lainiotis D.G., N.D. Asimakis, S.K. Katsikas, 'New Doubling Algorithms for the Discrete Periodic Riccati Equation,' Applied Mathematics and Computation, vol. 1, 1993, pp. 467–476.
- [12] Mayne, D.Q. 'A solution of the smoothing problem for linear dynamic systems', Automatica, vol. 4, pp. 73–92, 1966.
- [13] Plataniotis K.N., *Distributed Parallel Processing State Estimation Algorithms*, Ph.D Dissertation, Florida Institute of Technology, Melbourne, Florida, 1994.
- [14] Roy S., H.R. Hashemipour, A.J. Laub, 'Square root parallel Kalman filtering using reduced order local filters,' IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-27, No. 2, 1991, pp. 276–288.
- [15] Willsky A.S., M.G. Bello, D.A. Castanon, 'Combining and updating of local estimates and regional maps along sets of one dimensional tracks,' IEEE Trans. on Automatic Control, vol. AC-27, 1982, pp. 799–812.