Digital Signal Processing 23 (2013) 442–452

Contents lists available at SciVerse ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Optimal decentralized Kalman filter and Lainiotis filter

Nicholas Assimakis [a], Maria Adam [b,*], Maria Koziri [c], Stamatis Voliotis [d], Konstantinos Asimakis [e]

[a] *Department of Electronics, Technological Educational Institute of Lamia, 3rd km Old National Road Lamia–Athens, Lamia, Greece*
[b] *Department of Computer Science and Biomedical Informatics, University of Central Greece, 2-4 Papasiopoulou str., PO 35100 Lamia, Greece*
[c] *Department of Computer and Communication Engineering, University of Thessaly, Greece*
[d] *Department of Electrical Engineering, Technological Educational Institute of Chalkis, Greece*
[e] *Department of Computer Engineering and Informatics, University of Patras, Greece*

ARTICLE INFO

Article history:
Available online 16 August 2012

Keywords:
Kalman filter
Lainiotis filter
Discrete time
Decentralized algorithms
Analysis of algorithms
Optimization

ABSTRACT

A method to implement the optimal decentralized Kalman filter and the optimal decentralized Lainiotis filter is proposed; the method is based on the a priori determination of the optimal distribution of measurements into parallel processors, minimizing the computation time. The resulting optimal Kalman filter and optimal Lainiotis filter require uniform distribution or near to uniform distribution of measurements into parallel processors. The optimal uniform distribution has the advantages of elimination of idle time for the local processors and of low hardware cost, but it is not always applicable. The optimal filters present high parallelism speedup; this is verified through simulation results and is very important due to the fact that, in most real-time applications, it is essential to obtain the estimate in the shortest possible time.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

Estimation plays an important role in many fields of science and engineering. The discrete time Kalman filter [1] and Lainiotis filter [2] are well-known algorithms that solve the estimation problem. Real time problems require fast and accurate computation of large amount of data in order to deal with larger and more realistic models. The advances in the technology of integrated multi-sensor network systems allow the use of distributed or decentralized signal processing algorithms. The literature is very rich of contributions addressing several aspects of distributed or decentralized estimation.

There are several works in literature that address several aspects of distributed estimation. These works can be classified works into classes based on the modeling adopted: static versus dynamic estimation, distributed versus hierarchical estimation, and all-to-all versus multi-hop communication networks. A distributed Kalman filter presented in [3,4], wherein a system with an m-dimensional measurement vector is first split into subsystems of measurement sub-vectors, then these subsystems are individually processed by micro Kalman filters in the nodes of the network. In this system, the sensors compute an average inverse covariance and average measurements using consensus filters. These averaged values are then used by each node to individually compute the estimated state of the system using the information form of the Kalman filter. The distributed Kalman filtering algorithms are based on an average-consensus computing first the mean of the sensor measurements, and then to update and predict the local estimates using the centralized Kalman optimal gains; the Kalman Consensus Filter or the Generalized Kalman Consensus Filter are distributed algorithms for fusing multiple measurements from different sensors as stated in [3,5–7].

Decentralized Kalman filtering and distributed Kalman filtering are two separate problems. The decentralized Kalman filter was introduced in [8]. There are several works in literature that propose decentralized versions of the Kalman filter [9–16]. Fusion center (FC) based WSNs can perform decentralized estimation [15,16]. Decentralized estimation of Gaussian random parameters is presented in [10] for stationary environments and decentralized estimation of random signals in arbitrary nonlinear and non-Gaussian setups is presented in [12]. A comparison between the distributed Kalman filter and the decentralized sensor fusion algorithms both with and without fusion centers is presented in [17–19].

There are two major approaches to obtain the required estimates for such multiple-sensor systems as stated in [20]: (i) The centralized approach, where all the sensor observations are transmitted from the local sensor to a central location for processing, using Kalman filter [1,21–23] or Lainiotis filter [20,21,24–27]; this approach requires high computational load at the central processor. (ii) The decentralized approach, where the data are obtained by the different sensor subsystems, processed locally and the results are transmitted to the central processor which combines the information from the local subsystems to produce the

[*] Corresponding author.
E-mail addresses: assimakis@teilam.gr (N. Assimakis), madam@ucg.gr (M. Adam), mkoziri@gmail.com (M. Koziri), svoliotis@teihal.gr (S. Voliotis), inshame@gmail.com (K. Asimakis).

1051-2004/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.dsp.2012.08.005

global estimate. This approach is used in [28–30], where the algorithms used to obtain the estimates are based on the different Kalman filter formulations (standard covariance-type Kalman filter, information-type Kalman filter, square root Kalman filter) and a formulation of the Lainiotis per-step partitioning filter was presented in [24].

In this paper we use the ideas in [21,28,31] in order to partition the measurement noise process into statistically uncorrelated parts. A typical multi-sensor environment consists of several sensors observing a dynamic system, where each sensor is attached to a local processor; in these decentralized structures some amount of processing is done at the local processors of the network and the results are then communicated to a central processor, also referred to as a fusion center. The hierarchical approach for signal processing is used in the case where the sensors are both collocated and dispersed [28]. The resulting decentralized algorithms are decomposed into two parts: the local level and the central level. The data of each local processor is communicated to the fusion center where the global estimate is computed. The local processors can operate concurrently, since there is no need for communication among local processors and no communication is needed from the central processor downwards in the hierarchy of the local processors.

In this paper we investigate the problem of decentralized computing is the derivation of the optimal solution to the decentralized problem: the optimal number of processors required for the parallel implementation of Kalman filter and Lainiotis filter, for which the maximum parallelism speedup is achieved. The optimal decentralized Kalman filter and Lainiotis filter have been discussed in [32] assuming uniform distribution of measurements into the parallel processors. Concerning the novelty of the paper, we mention that: (i) decentralized implementation of both Kalman and Lainiotis filters is derived, (ii) a method to implement the optimal decentralized Kalman and Lainiotis filters is proposed; the method is based on the a priori determination of the optimal distribution of measurements into parallel processors, minimizing the computation time; this means that we are able to design the optimal sensor network for a given problem, (iii) the optimal distribution is derived for three separate cases: time varying, time invariant and steady state; this is significant due to the fact that there is a difference between the time varying case and the time invariant and steady state cases concerning the optimal distribution determination, and (iv) it is pointed out that the optimal distribution of measurements into parallel processors is not always uniform, but it can be near to uniform; this leads to maximum parallelism speedup mainly in cases when the number of measurements has a small number of dividers, especially when it is a prime number. The contribution of the paper is that we propose a method for implementing the optimal decentralized Kalman filter and Lainiotis filter by a priori determination of the distribution of measurements into parallel processors.

The paper is organized as follows: In Section 2 the decentralized filtering problem is formulated. In Section 3 the centralized Kalman filter and Lainiotis filter are presented for time varying, time invariant and steady state systems. In Section 4 the decentralized Kalman filter and Lainiotis filter are derived for time varying, time invariant and steady state systems. In Section 5 the computational requirements of both centralized and decentralized filtering algorithms are calculated. In Section 6 the optimal decentralized Kalman filter and Lainiotis filter are defined. First, the optimal theoretical uniform distribution is determined, but it is not always applicable. In the sequel, the optimal uniform distribution is derived, but it is not always applicable as well. Then, the optimal general distribution is derived: it is a uniform distribution or a near to uniform distribution with high parallelism speedup. In Section 7 simulation results are presented. Finally, Section 8 summarizes the conclusions.

2. Problem formulation

The estimation/filtering problem is associated with the following state space equations:

$$x(k+1) = F(k+1,k)x(k) + w(k) \tag{1}$$

$$z(k) = H(k)x(k) + v(k) \tag{2}$$

where $x(k)$ is the n-dimensional state vector at time k, $z(k)$ is the m-dimensional measurement vector, $F(k+1,k)$ is the $n \times n$ system transition matrix, $H(k)$ is the $m \times n$ output matrix, $\{w(k)\}$ is the plant noise process and $\{v(k)\}$ is the measurement noise process. These processes are assumed to be Gaussian, zero-mean, white and uncorrelated random processes with $n \times n$ plant noise and $m \times m$ measurement noise covariance matrices $Q(k)$ and $R(k)$, respectively. The initial value $x(0)$ of the state vector $x(k)$ at time $k = 0$ is considered as Gaussian random variable with mean x_0 and covariance P_0. Also, $x(0)$, $\{w(k)\}$ and $\{v(k)\}$ are independent.

For a multi-sensor environment, the measurement vector is partitioned into ρ parts [21,28,31]:

$$z^T(k) = \begin{bmatrix} z_1^T(k) & z_2^T(k) & \dots & z_\rho^T(k) \end{bmatrix} \tag{3}$$

where $z_i(k)$ is the m_i-dimensional vector for every $i = 1, 2, \dots, \rho$ and it holds:

$$\sum_{i=1}^{\rho} m_i = m \tag{4}$$

Partitioning the measurement noise process into ρ statistically uncorrelated parts as

$$v^T(k) = \begin{bmatrix} v_1^T(k) & v_2^T(k) & \dots & v_\rho^T(k) \end{bmatrix} \tag{5}$$

the measurement noise covariance matrix is given by

$$E\left[v(k)v^T(k)\right] = \operatorname{diag}\left(R_1(k), R_2(k), \dots, R_\rho(k)\right) \tag{6}$$

where $R_i(k)$ is the $m_i \times m_i$ (measurement noise) covariance matrix and partitioning the output matrix as

$$H^T(k) = \begin{bmatrix} H_1^T(k) & H_2^T(k) & \dots & H_\rho^T(k) \end{bmatrix} \tag{7}$$

where $H_i(k)$ is the corresponding $m_i \times n$ submatrix of $H(k)$, the measurement equation takes the form:

$$z_i(k) = H_i(k)x(k) + v_i(k), \quad i = 1, 2, \dots, \rho \tag{8}$$

3. Centralized Kalman filter and Lainiotis filter

In this section we present the discrete time Kalman filter [1] and the discrete time Lainiotis filter [2] in a centralized form. This means that all the required computations are carried out in one central processor.

For time varying systems, the classical implementation of the Kalman filter is summarized in the following equations:

Kalman filter

$$x(k+1/k) = F(k+1,k)x(k/k) \tag{9}$$

$$P(k+1/k) = F(k+1,k)P(k/k)F^T(k+1,k) + Q(k) \tag{10}$$

$$K(k+1) = P(k+1/k)H^T(k+1)\big[H(k+1)P(k+1/k) \times H^T(k+1) + R(k+1)\big]^{-1} \tag{11}$$

$$x(k+1/k+1) = x(k+1/k) + K(k+1)\big[z(k+1) - H(k+1)x(k+1/k)\big] \tag{12}$$

$$P(k+1/k+1) = P(k+1/k) - K(k+1)H(k+1) \times P(k+1/k) \tag{13}$$

Lainiotis filter

$$A(k+1)=\left[H(k+1)Q(k)H^T(k+1)+R(k+1)\right]^{-1} \tag{14}$$

$$K_n(k+1)=Q(k)H^T(k+1)A(k+1) \tag{15}$$

$$K_m(k+1)=F^T(k+1,k)H^T(k+1)A(k+1) \tag{16}$$

$$P_n(k+1,k)=Q(k)-K_n(k+1)H(k+1)Q(k) \tag{17}$$

$$F_n(k+1,k)=F(k+1,k)-K_n(k+1)H(k+1)F(k+1,k) \tag{18}$$

$$O_n(k+1)=K_m(k+1)H(k+1)F(k+1,k) \tag{19}$$

$$x_n(k+1/k+1)=K_n(k+1)z(k+1) \tag{20}$$

$$M_n(k+1)=K_m(k+1)z(k+1) \tag{21}$$

$$\begin{aligned}P(k+1/k+1)={}&P_n(k+1,k)+F_n(k+1,k)\\&\times\left[I+P(k/k)O_n(k+1)\right]^{-1}\\&\times P(k/k)F_n^T(k+1,k)\end{aligned} \tag{22}$$

$$\begin{aligned}x(k+1/k+1)={}&x_n(k+1/k+1)+F_n(k+1,k)\\&\times\left[I+P(k/k)O_n(k+1)\right]^{-1}\\&\times\left[P(k/k)M_n(k+1)+x(k/k)\right]\end{aligned} \tag{23}$$

The equivalence of the two filters is confirmed in [33]. Hence, using only the equations of Kalman filter, (10)–(11) in (13) and (9)–(11) in (12), the following implementation of the centralized time varying Kalman filter/Lainiotis filter is derived, as shown in Appendix A.

Centralized time varying Kalman filter/Lainiotis filter

$$\begin{aligned}P(k+1/k+1)={}&\Big\{\left[F(k+1,k)P(k/k)F^T(k+1,k)+Q(k)\right]^{-1}\\&+H^T(k+1)R^{-1}(k+1)H(k+1)\Big\}^{-1}\end{aligned} \tag{24}$$

$$\begin{aligned}x(k+1/k+1)={}&P(k+1/k+1)\Big\{H^T(k+1)R^{-1}(k+1)z(k+1)\\&+\left[F(k+1,k)P(k/k)F^T(k+1,k)+Q(k)\right]^{-1}\\&\times F(k+1,k)x(k/k)\Big\}\end{aligned} \tag{25}$$

For time invariant systems, the system transition matrix, the output matrix, the plant and measurement noise covariance matrices are constant, following the similar way as (24)–(25) the centralized time invariant Kalman filter/Lainiotis filter is obtained:

Centralized time invariant Kalman filter/Lainiotis filter

$$P(k+1/k+1)=\left\{\left[FP(k/k)F^T+Q\right]^{-1}+H^TR^{-1}H\right\}^{-1} \tag{26}$$

$$\begin{aligned}x(k+1/k+1)={}&P(k+1/k+1)\Big\{H^TR^{-1}z(k+1)\\&+\left[FP(k/k)F^T+Q\right]^{-1}Fx(k/k)\Big\}\end{aligned} \tag{27}$$

where the constant matrices H^TR^{-1} and $H^TR^{-1}H$ are calculated off-line.

For time invariant systems, it is well known [34] that if the signal process model is asymptotically stable (i.e. all eigenvalues of F lie inside the unit circle), then there exists a steady state value $\overline{P}_e$ of the estimation error covariance matrix, which is calculated off-line by

$$\overline{P}_e=[I-\overline{K}H]\overline{P}_p \tag{28}$$

where the steady state gain $\overline{K}$ is calculated by

$$\overline{K}=\overline{P}_pH^T\left[H\overline{P}_pH^T+R\right]^{-1} \tag{29}$$

and the steady state prediction error covariance matrix $\overline{P}_p$ is calculated off-line by off-line solving of the corresponding discrete time Riccati equation [34,35]:

$$\overline{P}_p=Q+F\overline{P}_pF^T-F\overline{P}_pH^T\left[H\overline{P}_pH^T+R\right]^{-1}H\overline{P}_pF^T \tag{30}$$

Then, the centralized steady state Kalman filter/Lainiotis filter is obtained:

Centralized steady state Kalman filter/Lainiotis filter

$$x(k+1/k+1)=A_{ss}x(k/k)+B_{ss}z(k+1) \tag{31}$$

where the following constant matrices are calculated off-line:

$$A_{ss}=[I-\overline{K}H]F \tag{32}$$

$$B_{ss}=\overline{K} \tag{33}$$

4. Decentralized Kalman filter and Lainiotis filter

In this section we present the discrete time Kalman and Lainiotis filters in a decentralized form [28]. As was pointed out earlier, the main drawback of the above centralized approaches is that they require a large amount of computations to be carried out in the central processor, demanding therefore large computational power. Moreover in the case of very large m, there is a tremendous computational burden in the processor. In the following, the results of the previous section are extended and the corresponding decentralized algorithms are decomposed into two parts: the *local level* and the *central level*. At the local level each processor computes its local quantities using its own measurement. The data of each local processor is communicated to the fusion center where the global estimate is computed. The local processors can operate concurrently, since there is no need for communication among local processors and no communication is needed from the central processor downwards in the hierarchy of the local processors. The generation of the global estimate can be thought of as overhead, due to the fact that the central processor needs information from the local processors, but not vice versa.

Using the formulas of the matrices in (6), (7) and (3), we have

$$\begin{aligned}&H^T(k+1)R^{-1}(k+1)H(k+1)\\&\quad=\sum_{i=1}^{\rho}H_i^T(k+1)R_i^{-1}(k+1)H_i(k+1)\end{aligned} \tag{34}$$

$$\begin{aligned}&H^T(k+1)R^{-1}(k+1)z(k+1)\\&\quad=\sum_{i=1}^{\rho}H_i^T(k+1)R_i^{-1}(k+1)z_i(k+1)\end{aligned} \tag{35}$$

which allow us through Eqs. (24)–(25) to derive the following implementation of the decentralized time varying Kalman filter/Lainiotis filter:

Decentralized time varying Kalman filter/Lainiotis filter

Local level

$$B_i(k+1)=H_i^T(k+1)R_i^{-1}(k+1)H_i(k+1),\quad i=1,\dots,\rho \tag{36}$$

$$b_i(k+1)=H_i^T(k+1)R_i^{-1}(k+1)z_i(k+1),\quad i=1,\dots,\rho \tag{37}$$

Central level

$$\begin{aligned}P(k+1/k+1)={}&\Bigg[\left[F(k+1,k)P(k/k)F^T(k+1,k)+Q(k)\right]^{-1}\\&+\sum_{i=1}^{\rho}B_i(k+1)\Bigg]^{-1}\end{aligned} \tag{38}$$

$$x(k+1/k+1) = P(k+1/k+1)\left\{\sum_{i=1}^{\rho} b_i(k+1) + \left[F(k+1,k)P(k/k)F^T(k+1,k) + Q(k)\right]^{-1} \times F(k+1,k)x(k/k)\right\} \tag{39}$$

For time invariant systems, the following implementation of the decentralized time invariant Kalman filter/Lainiotis filter is derived:

Decentralized time invariant Kalman filter/Lainiotis filter

Local level

$$b_i(k+1) = H_i^T R_i^{-1} z_i(k+1), \quad i = 1, \ldots, \rho \tag{40}$$

where $H_i^T R_i^{-1}$, $i = 1, \ldots, \rho$, are constant matrices calculated off-line.

Central level

$$P(k+1/k+1) = \left[\left[FP(k/k)F^T + Q\right]^{-1} + \sum_{i=1}^{\rho} B_i\right]^{-1} \tag{41}$$

$$x(k+1/k+1) = P(k+1/k+1)\left\{\sum_{i=1}^{\rho} b_i(k+1) + \left[FP(k/k)F^T + Q\right]^{-1} Fx(k/k)\right\} \tag{42}$$

where B_i, $i = 1, \ldots, \rho$, and $\sum_{i=1}^{\rho} B_i$, are constant matrices calculated off-line by

$$B_i = H_i^T R_i^{-1} H_i, \quad i = 1, \ldots, \rho \tag{43}$$

In the steady state case the following implementation of the decentralized steady state Kalman filter/Lainiotis filter is derived:

Decentralized steady state Kalman filter/Lainiotis filter

Local level

$$d_i(k+1) = \overline{P}_e H_i^T R_i^{-1} z_i(k+1), \quad i = 1, \ldots, \rho \tag{44}$$

where the constant matrices $\overline{P}_e H_i^T R_i^{-1}$, $i = 1, \ldots, \rho$, are calculated off-line by the steady state value $\overline{P}_e$ of the estimation error covariance matrix using (28), that is calculated by the steady state value $\overline{P}_p$ from the solving of Eq. (30) and the steady state gain $\overline{K}$ from (29).

Central level

$$x(k+1/k+1) = A_{ss}x(k/k) + \sum_{i=1}^{\rho} d_i(k+1) \tag{45}$$

where the constant matrix A_{ss} is calculated off-line using (32).

5. Computational requirements

The centralized and the decentralized Kalman filters and Lainiotis filters are recursive algorithms. The computational times required for the implementation of the centralized algorithms are:

$$T_{TV}^C = B_{TV}^C \cdot s \cdot t_{op}, \qquad T_{TI}^C = B_{TI}^C \cdot s \cdot t_{op}, \qquad T_{SS}^C = B_{SS}^C \cdot s \cdot t_{op} \tag{46}$$

and the computational times required for the implementation of the decentralized algorithms are:

$$T_{TV}^D = B_{TV}^D \cdot s \cdot t_{op}, \qquad T_{TI}^D = B_{TI}^D \cdot s \cdot t_{op}$$
$$T_{SS}^D = B_{SS}^D \cdot s \cdot t_{op} \tag{47}$$

where

- B_{TV}^C, B_{TI}^C, B_{SS}^C are the per recursion calculation burden required for the on-line calculations of the centralized time varying, time invariant, steady state algorithms, respectively,
- B_{TV}^D, B_{TI}^D, B_{SS}^D are the per recursion calculation burden required for the on-line calculations of the decentralized time varying, time invariant, steady state algorithms, respectively,
- s is the number of recursions (steps) that each algorithm executes,
- t_{op} is the time required to perform a scalar operation.

Table 1
Calculation burden of matrix operations.

Matrix operation	Calculation burden
$A(n \times m) + B(n \times m) = C(n \times m)$	nm
$A(n \times n) + B(n \times n) = S(n \times n)$ [S: symmetric]	$0.5n^2 + 0.5n$
$I(n \times n) + A(n \times n) = B(n \times n)$ [I: identity]	n
$A(n \times m) \cdot B(m \times k) = C(n \times k)$	$2nmk - nk$
$A(n \times m) \cdot B(m \times n) = S(n \times n)$ [S: symmetric]	$n^2m + nm - 0.5n^2 - 0.5n$
$[A(n \times n)]^{-1} = B(n \times n)$	$(16n^3 - 3n^2 - n)/6$

The centralized and the decentralized algorithms presented above are equivalent with respect to their behavior: they calculate theoretically the same estimates. Then, it is reasonable to assume that both implementations compute the estimate value $x(L/L)$ of the state vector $x(L)$, executing the same number of recursions. Thus, in order to compare the algorithms with respect to their computational time, we have to compare their per recursion calculation burden required for the on-line calculations; the calculation burden of the off-line calculations (initialization process for time invariant and steady state filters) is not taken into account.

The computational analysis is based on the analysis in [33]: scalar operations are involved in matrix manipulation operations, which are needed for the implementation of the filtering algorithms. Table 1 summarizes the calculation burden of needed matrix operations.

In the centralized algorithms case, all required computations for the calculation of the global estimate are carried out in one central processor. The computational requirements of all centralized algorithms depend on the state vector dimension n and the measurement vector dimension m.

In the decentralized algorithms case, each local processor computes its local data using its own measurement (the local processors operate in parallel, since there is no need for communication among local processors). The data of each local processor is communicated to the central processor (there is no two way communication between the local and the central processor), where the global estimate is computed. The computational requirements of all decentralized algorithms depend on the state vector dimension n and the maximum local measurement vector dimension M:

$$M = \max\{m_i\}, \quad i = 1, \ldots, \rho \tag{48}$$

The per recursion calculation burdens of all the Kalman/Lainiotis filtering algorithms are determined in Appendix B and summarized in Table 2.

It is obvious that when $\rho = 1$ and $M = m$, then the decentralized algorithms become the same as the centralized algorithms with equivalent calculation burdens.

Table 2
Computational requirements. Per recursion calculation burden of the Kalman/Lainiotis filtering algorithms.

Implementation	System	Calculation burden
Centralized	Time varying	$B^C_{TV} = (31n^3 + 6n^2 - 7n)/3 + (16m^3 - 3m^2 - m)/6 + n^2m + 2nm + 2nm^2$
Centralized	Time invariant	$B^C_{TI} = 2nm + (62n^3 + 15n^2 - 11n)/6$
Centralized	Steady state	$B^C_{SS} = 2n^2 + 2nm - n$
Decentralized	Time varying	$B^D_{TV} = (62n^3 + 9n^2 - 23n)/6 + (16M^3 - 3M^2 - M)/6 + n^2M + 2nM + 2nM^2 + (n^2 + 3n)\rho/2$
Decentralized	Time invariant	$B^D_{TI} = 2nM + n\rho + (62n^3 + 15n^2 - 17n)/6$
Decentralized	Steady state	$B^D_{SS} = 2n^2 - 2n + 2nM + n\rho$

6. Optimal decentralized Kalman filter and Lainiotis filter definition

In this section, a method to define the optimal decentralized Kalman and Lainiotis filters is proposed. Our aim is to minimize the computation time of the decentralized filter. From the previous section, it becomes obvious that in order to minimize the computation time of the decentralized algorithm, we have to minimize the corresponding per recursion calculation burden. From Table 2 we can easily see that the calculation burden of the decentralized Kalman filter/Lainiotis filter depends on the distribution of m measurements into ρ parallel local processors. Each local processor can transfer data to the central processor after its local data is computed in the local processor; this communication process can be performed while the other local processors compute their local estimates. The global estimate is computed in the central processor after all needed information is transferred from the local processors to the central processor.

(i) Theoretical uniform distribution. We examine the uniform distribution of measurement into parallel processors. We make the following assumption: The measurement vector is partitioned into ρ equal parts. Then, it is clear that the following relation holds

$$m_i = M, \quad i = 1, \dots, \rho \tag{49}$$

and then by (4) we have:

$$\rho \cdot M = m \tag{50}$$

This uniform distribution of measurements into local processors has the following advantages:

1. All local processors perform the same calculations concerning quantities of the same type and dimensionality. Thus, all local processors have the same structure and therefore, low hardware cost is required for the implementation of the decentralized algorithms.
2. There is no idle time for the local processors.

It is obvious that when $\rho = 1$ and $M = m$, then the decentralized algorithms become the same as the centralized algorithms (all measurements into one processor).

It is also obvious that when $\rho = m$ and $M = 1$, then the decentralized algorithms use measurements fully decentralized (one measurement per processor).

In order to determine the optimal theoretical uniform decentralized Kalman filter/Lainiotis filter, we observe that ρ and M must be positive integers satisfying (50). From Table 2 we conclude that we are able to determine the optimal uniform distribution of m measurements into ρ^{th}_{opt} processors, each one of which deals with M^{th}_{opt} measurements, using the optimality criterion of minimizing the computation time of the decentralized Kalman filter/Lainiotis filter.

For time varying systems, the calculation burden of the decentralized Kalman filter/Lainiotis filter B^D_{TV} can be written as a function of ρ, M, denoted $f_{TVF}(\rho)$, $f_{TVF}(M)$, and given by:

$$\begin{aligned} f_{TVF}(\rho) &= \frac{1}{6}\left(\frac{16m^3}{\rho^3} - \frac{3m^2}{\rho^2} - \frac{m}{\rho} + 62n^3 + 9n^2 - 23n\right) \\ &\quad + \left(n^2 + 2n\right)\frac{m}{\rho} + 2n\frac{m^2}{\rho^2} + \frac{n^2 + 3n}{2}\rho \\ f_{TVF}(M) &= \frac{1}{6}\left(16M^3 - 3M^2 - M + 62n^3 + 9n^2 - 23n\right) \\ &\quad + \left(n^2 + 2n\right)M + 2nM^2 + \frac{m(n^2 + 3n)}{2M} \end{aligned} \tag{51}$$

The first and second derivatives of $f_{TVF}(\rho)$, $f_{TVF}(M)$ with respect to ρ, M are:

$$\begin{aligned} &f'_{TVF}(\rho) \\ &\quad = -\frac{8m^3}{\rho^4} - \frac{m^2(4n-1)}{\rho^3} - \frac{m(n^2 + 2n - \frac{1}{6})}{\rho^2} + \frac{n^2 + 3n}{2} \\ &f'_{TVF}(M) \\ &\quad = \left(8M^2 + (4n-1)M - \frac{1}{6}\right) + \left(n^2 + 2n\right) - \frac{m(n^2 + 3n)}{2M^2} \end{aligned} \tag{52}$$

$$\begin{aligned} f''_{TVF}(\rho) &= \frac{32m^3}{\rho^5} + \frac{3m^2(4n-1)}{\rho^4} + \frac{2m(n^2 + 2n - \frac{1}{6})}{\rho^3} \\ f''_{TVF}(M) &= 16M + 4n - 1 + \frac{m(n^2 + 3n)}{M^3} \end{aligned} \tag{53}$$

Then, it becomes clear (from the first and the second derivatives) that the function $f_{TVF}(\rho)$ for $\rho > 0$ has a unique minimum value at ρ^{th}_{opt} and the function $f_{TVF}(M)$ for $M > 0$ has a unique minimum value at M^{th}_{opt} determined by solving:

$$\begin{aligned} &3\left(n^2 + 3n\right)\rho^4 - \left(6n^2 + 12n - 1\right)m\rho^2 - (24n - 6)m^2\rho \\ &\quad - 48m^3 = 0 \\ &48M^4 + (24n - 6)M^3 + \left(6n^2 + 12n - 1\right)M^2 \\ &\quad - 3m\left(n^2 + 3n\right) = 0 \end{aligned} \tag{54}$$

For time invariant and for steady state systems, the calculation burden of the decentralized Kalman filter/Lainiotis filter B^D_{TI} and B^D_{SS} can be written as a function of ρ, M, denoted $f_{TIF}(\rho)$, $f_{TIF}(M)$ and $f_{SSF}(\rho)$, $f_{SSF}(M)$, and given by:

$$\begin{aligned} f_{TIF}(\rho) &= \frac{2nm}{\rho} + n\rho + \frac{62n^3 + 15n^2 - 17n}{6} \\ f_{TIF}(M) &= 2nM + \frac{nm}{M} + \frac{62n^3 + 15n^2 - 17n}{6} \end{aligned} \tag{55}$$

and

$$\begin{aligned} f_{SSF}(\rho) &= \frac{2nm}{\rho} + n\rho + 2n^2 - 2n \\ f_{SSF}(M) &= 2nM + \frac{nm}{M} + 2n^2 - 2n \end{aligned} \tag{56}$$

The first and second derivatives of the above functions with respect to ρ, M are:

$$f'_{TIF}(\rho)=f'_{SSF}(\rho)=-\frac{2nm}{\rho^2}+n$$
$$f'_{TIF}(M)=f'_{SSF}(M)=-\frac{nm}{M^2}+2n \tag{57}$$

$$f''_{TIF}(\rho)=f''_{SSF}(\rho)=\frac{4nm}{\rho^3}$$
$$f''_{TIF}(M)=f''_{SSF}(M)=\frac{2nm}{M^3} \tag{58}$$

Then, it becomes clear (from the first and the second derivatives) that the functions $f_{TIF}(\rho)$ and $f_{SSF}(\rho)$ for $\rho>0$ have a unique minimum value at ρ^{th}_{opt} and the functions $f_{TIF}(M)$ and $f_{SSF}(M)$ for $M>0$ have a unique minimum value at M^{th}_{opt} determined by solving:

$$\rho^2=2m$$
$$M^2=\frac{m}{2} \tag{59}$$

Thus, the optimum theoretical uniform distribution is determined theoretically by solving (54) for time varying systems or (59) for time invariant or steady state systems; the optimum distribution consists of ρ^{th}_{opt} processors with M^{th}_{opt} measurements in each of them. It is obvious that these theoretical solutions hold for $m>1$; in fact, when $m=1$, then $\rho^{th}_{opt}=1$ and $M^{th}_{opt}=1$.

(ii) Uniform distribution. The above theoretical solution has the disadvantage that it is not always applicable. In fact, it is obvious that ρ^{th}_{opt} and M^{th}_{opt} have to be positive integers; but the theoretical solutions by (54) or (59) are real numbers in general. If these theoretical values are integers, then they define the unique optimal uniform distribution: ρ^{u}_{opt} processors with M^{u}_{opt} measurements in each of them. If they are not integers, then we have to seek all the uniform distributions minimizing the computation time using $f_{TVF}(\rho)$ or $f_{TVF}(M)$ for time varying systems, $f_{TIF}(\rho)$ or $f_{TIF}(M)$ for time invariant systems and $f_{SSF}(\rho)$ or $f_{SSF}(M)$ for steady state systems in order to a priori define the optimal uniform decentralized Kalman filter/Lainiotis filter. Then, there exists either one optimal uniform distribution minimizing the computational cost consisting of ρ^{u}_{opt} processors with M^{u}_{opt} measurements in each of them, or two optimal uniform distributions, because it is possible that there exist two integer values $M1^{u}_{opt}<M^{th}_{opt}<M2^{u}_{opt}$ and two integer values $\rho1^{u}_{opt}<\rho^{th}_{opt}<\rho2^{u}_{opt}$ minimizing the computational cost. In the case where there exist two optimal uniform distributions, then we propose to define the optimum distribution with criterion the minimum number of processors in order to minimize the hardware cost.

(iii) General distribution (uniform or non-uniform). The uniform distribution has the disadvantage that it is not always applicable; for example when the number of measurements m has a small number of dividers, especially when it is a prime number. In the following we examine the case of general (uniform or non-uniform) distributions of measurements into processors.

So we have to seek all the distributions (uniform or non-uniform) of measurements into processors minimizing the computation time using $f_{TVF}(\rho)$ or $f_{TVF}(M)$ for time varying systems, $f_{TIF}(\rho)$ or $f_{TIF}(M)$ for time invariant systems and $f_{SSF}(\rho)$ or $f_{SSF}(M)$ for steady state systems in order to a priori define the optimal uniform decentralized Kalman filter/Lainiotis filter. From the previous analysis, it becomes obvious that there exist either one or two optimal uniform distributions minimizing the computational cost. Furthermore, it is possible that there exist non-uniform distributions (in fact near to uniform distributions) consisting of ρ^{g}_{opt} processors with M^{g}_{opt} measurements in each of them, which also minimize the computational cost as well.

Table 3
Percent speedup gain from optimal uniform distribution to optimal general distribution.

System	Maximum gain (%)	Average gain (%)
Steady state	253.571	9.614
Time invariant	186.842	0.649
Time varying	58.199	1.589

In the case where there exist many optimal uniform or non-uniform distributions, which minimize the computational cost, then we propose to define the optimum distribution with criterion the minimum number of processors in order to minimize the hardware cost; of course, the physical topology of sensors (measurements) may change this criterion. Thus, the resulting optimal distribution is a uniform distribution or a near to uniform distribution minimizing the computation time.

In the sequel, we compare the decentralized implementation of Kalman/Lainiotis filters to the centralized implementation. Using (46) and (47) the theoretical parallelism speedup of each decentralized Kalman/Lainiotis algorithm is defined as the ratio of the computational time required for its centralized implementation by the computational time required for its decentralized implementation:

$$speedup^{th}_{TV}=\frac{T^C_{TV}}{T^D_{TV}}=\frac{B^C_{TV}}{B^D_{TV}},\qquad speedup^{th}_{TI}=\frac{T^C_{TI}}{T^D_{TI}}=\frac{B^C_{TI}}{B^D_{TI}}$$
$$speedup^{th}_{SS}=\frac{T^C_{SS}}{T^D_{SS}}=\frac{B^C_{SS}}{B^D_{SS}} \tag{60}$$

Thus, the optimal decentralized algorithms present the maximum parallelism speedup, which is achieved minimizing the computational time required for the algorithm's decentralized implementation. The maximum parallelism speedup increases as the measurement vector dimension increases (and the state vector dimension remains constant), for time varying, time invariant and steady state systems. This is very important for multi-sensor problems. The optimal distribution leads to the maximum speedup, which is greater than or equal to the maximum speedup of the optimal uniform distribution; it is much greater when the number of measurements m has a small number of dividers, especially when it is a prime number (see in Section 7, Example 4).

Furthermore, in order to compare the optimal general (uniform or non-uniform) distribution to the (real) optimal uniform distribution we use the percent speedup gain:

$$\%gain=100\cdot\frac{speedup^g-speedup^u}{speedup^u} \tag{61}$$

where $speedup^g$ is the speedup achieved for the optimal general (uniform or non-uniform) distribution and $speedup^u$ is the speedup achieved for the optimal uniform distribution. The percent speedup gain from optimal uniform distribution to optimal distribution for $n=1,\dots,100$ and $m=1,\dots,100$ is presented in Table 3. Of course, there is no speedup gain when the optimal distribution is a uniform one. If this is not the case, then the maximum speedup gain is very high.

Finally, in order to compare the optimal general (uniform or non-uniform) distribution to the theoretical optimal uniform distribution we use the percent speedup efficiency:

$$\%efficiency=100\cdot\frac{speedup^g}{speedup^{th}} \tag{62}$$

Table 4
Percent speedup efficiency from optimal theoretical uniform to optimal general distribution.

System	Average efficiency (%)	Minimum efficiency (%)
Steady state	99.483	90.351
Time invariant	99.965	96.028
Time varying	99.855	87.841

Table 5
Optimal distributions (processors × measurements).

Distribution type	Optimal distribution	ρ_{opt}	M_{opt}	Max speedup
Theoretical uniform	258.757×3.865	258.757	3.865	$3.743297 \cdot 10^6$
Uniform	250×4	250	4	$3.736932 \cdot 10^6$
General	250×4	250	4	$3.736932 \cdot 10^6$

Table 6
Optimal distributions (processors × measurements).

Distribution type	Number of distributions	ρ_{opt}	M_{opt}
Uniform	1	40	25
Near to uniform	22	42	24
Near to uniform	77	44	23
Near to uniform	77	46	22
Near to uniform	22	48	21
Uniform	1	50	20

where $speedup^g$ is the speedup achieved for the optimal general (uniform or non-uniform) distribution and $speedup^{th}$ is the speedup achieved for the theoretical optimal uniform distribution. The percent speedup efficiency from optimal theoretical uniform distribution to optimal distribution for $n = 1, \ldots, 100$ and $m = 1, \ldots, 100$ is presented in Table 4. Of course, the speedup efficiency is 100% when the optimal distribution is a uniform one. If this is not the case, then the average speedup efficiency is very important.

7. Simulation results

In this section it is pointed out through simulation results that the proposed optimal decentralized Kalman filter/Lainiotis filter presents high parallelism speedup.

Example 1 *(Time varying Kalman filter/Lainiotis filter).* A typical multi-sensor example taken from [21] is presented. A time varying system with $n = 1$ and $m = 1000$ is considered. The optimal theoretical uniform distribution (processors × measurements) and the real optimal distribution are shown in Table 5.

The optimal decentralized Kalman filter/Lainiotis filter is achieved for the optimum uniform distribution: $M_{opt} = 4$ measurements in each of $\rho_{opt} = 250$ local processors. The optimal decentralized Kalman filter/Lainiotis filter presents a very high maximum parallelism speedup: the optimal decentralized Kalman filter/Lainiotis filter can be implemented $3.736932 \cdot 10^6$ times faster than the centralized Kalman filter/Lainiotis filter. The speedup efficiency is 99.830%.

Example 2 *(Time invariant Kalman filter/Lainiotis filter).* A typical seismic deconvolution example taken from [36] is presented. The time invariant wavelet used to describe the signal received by the seismic sensors is assumed of order $n = 4$. A number of $m = 1000$ sensors divided into local geophone clusters are utilized in order to capture the seismic trace. The optimal decentralized Kalman filter/Lainiotis filter arises for 200 different distributions, as shown in Table 6.

The optimal distributions are 2 uniform distributions: $M_{opt} = 25$ measurements in each of $P_{opt} = 40$ local processors and $M_{opt} = 20$ measurements in each of $P_{opt} = 50$ local processors and 198 near to uniform distributions. In this case, we propose to determine the optimum distribution using the criterion of minimizing the number of processors (except of the criterion of minimizing the computation time). This is reasonable since the hardware cost is minimized.

Table 7
Optimal distributions (processors × measurements).

Distribution type	Optimal distribution	ρ_{opt}	M_{opt}	Max speedup
Theoretical uniform	44.721×22.360	44.721	22.360	8.297616
Uniform	40×25	40	25	8.280000
General	40×25	40	25	8.280000

The optimal theoretical uniform distribution (processors × measurements) and the real optimal distribution are shown in Table 7.

The optimal decentralized Kalman filter/Lainiotis filter can be implemented 8.280000 times faster than the centralized Kalman filter/Lainiotis filter. This is achieved for all the above distributions. The speedup efficiency is 99.788%.

Table 8
Optimal distributions (processors × measurements).

Distribution type	Optimal distribution	ρ_{opt}	M_{opt}	Max speedup
Theoretical uniform	258.181×3.862	258.181	3.862	$3.717816 \cdot 10^6$
Uniform	997×1	997	1	$1.316193 \cdot 10^6$
General	$247 \times 4 + 3 \times 3$	250	4	$3.703407 \cdot 10^6$
General	$248 \times 4 + 1 \times 3 + 1 \times 2$	250	4	$3.703407 \cdot 10^6$
General	$249 \times 4 + 1 \times 1$	250	4	$3.703407 \cdot 10^6$

Example 3 *(Time varying Kalman filter/Lainiotis filter with prime number of measurements).* In this example a time varying system is assumed with scalar state and prime number of measurements: $n = 1$ and $m = 997$ (prime number). The optimal theoretical uniform distribution (processors × measurements) and the real optimal distributions are shown in Table 8.

The optimal decentralized Kalman filter/Lainiotis filter can be implemented $3.703407 \cdot 10^6$ times faster than the centralized Kalman filter/Lainiotis filter. This is achieved for 3 different near to uniform distributions. The speedup gain is 181.373%. The speedup efficiency is 99.612%.

8. Conclusions

Centralized and decentralized algorithms for the solution of the discrete time estimation problem for multi-sensor environment were presented. The discrete time centralized and decentralized Kalman filters/Lainiotis filters were analyzed for time varying, time invariant and steady state systems and their computational requirements were discussed. A method was developed to define the optimal decentralized Kalman and Lainiotis filters a priori (before the implementation of the filter). The method is based on the determination of the optimum distribution of measurements into parallel processors using the criterion of minimizing the computation time. The paper proposes the optimal distribution of measurements into parallel processors. Thus we are able to design the optimal sensor network choosing among equivalent (optimal) distributions. So we can design such an optimal network for a given problem. On the other hand, if the sensor network is given, we have to use the measurements distribution to this given sensor network (which may be not the optimal one). In both cases, the computation time depends on the maximum local measurement vector dimension. Of course the decision of designing the optimal

Table 9
Centralized time varying Kalman filter/Lainiotis filter.

Matrix operation	Matrix dimensions	Calculation burden
$F(k+1,k)P(k/k)$	$(n\times n)\cdot(n\times n)$	$2n^3-n^2$
$F(k+1,k)P(k/k)F^T(k+1,k)$	$(n\times n)\cdot(n\times n)$ symmetric	$n^3+0.5n^2-0.5n$
$W(k)\equiv F(k+1,k)P(k/k)F^T(k+1,k)+Q(k)$	$(n\times n)+(n\times n)$ symmetric	$0.5n^2+0.5n$
$W^{-1}(k)$	$(n\times n)$	$(16n^3-3n^2-n)/6$
$R^{-1}(k+1)$	$(m\times m)$	$(16m^3-3m^2-m)/6$
$H^T(k+1)R^{-1}(k+1)$	$(n\times m)\cdot(m\times m)$	$2nm^2-nm$
$H^T(k+1)R^{-1}(k+1)H(k+1)$	$(n\times m)\cdot(m\times n)$ symmetric	$n^2m+nm-0.5n^2-0.5n$
$W^{-1}(k)+H^T(k+1)R^{-1}(k+1)H(k+1)$	$(n\times n)+(n\times n)$ symmetric	$0.5n^2+0.5n$
$P(k+1/k+1)=[W^{-1}(k)+H^T(k+1)R^{-1}(k+1)H(k+1)]^{-1}$	$(n\times n)$	$(16n^3-3n^2-n)/6$
$H^T(k+1)R^{-1}(k+1)z(k+1)$	$(n\times m)\cdot(m\times 1)$	$2nm-n$
$W^{-1}(k)F(k+1,k)$	$(n\times n)\cdot(n\times n)$	$2n^3-n^2$
$W^{-1}(k)F(k+1,k)x(k/k)$	$(n\times n)\cdot(n\times 1)$	$2n^2-n$
$H^T(k+1)R^{-1}(k+1)z(k+1)+W^{-1}(k)F(k+1,k)x(k/k)$	$(n\times 1)+(n\times 1)$	n
$x(k+1/k+1)=P(k+1/k+1)\{H^T(k+1)R^{-1}(k+1)z(k+1)+W^{-1}(k)F(k+1,k)x(k/k)\}$	$(n\times n)\cdot(n\times 1)$	$2n^2-n$
Total		$B_{TV}^C=(31n^3+6n^2-7n)/3$ $+(16m^3-3m^2-m)/6$ $+n^2m+2nm+2nm^2$

Table 10
Centralized time invariant Kalman filter/Lainiotis filter.

Matrix operation	Matrix dimensions	Calculation burden
$FP(k/k)$	$(n\times n)\cdot(n\times n)$	$2n^3-n^2$
$FP(k/k)F^T$	$(n\times n)\cdot(n\times n)$ symmetric	$n^3+0.5n^2-0.5n$
$W(k)\equiv FP(k/k)F^T+Q$	$(n\times n)+(n\times n)$ symmetric	$0.5n^2+0.5n$
$W^{-1}(k)$	$(n\times n)$	$(16n^3-3n^2-n)/6$
$W^{-1}(k)+H^TR^{-1}H$	$(n\times n)+(n\times n)$ symmetric	$0.5n^2+0.5n$
$P(k+1/k+1)=[W^{-1}(k)+H^TR^{-1}H]^{-1}$	$(n\times n)$	$(16n^3-3n^2-n)/6$
$H^TR^{-1}z(k+1)$	$(n\times m)\cdot(m\times 1)$	$2nm-n$
$W^{-1}(k)F$	$(n\times n)\cdot(n\times n)$	$2n^3-n^2$
$W^{-1}(k)Fx(k/k)$	$(n\times n)\cdot(n\times 1)$	$2n^2-n$
$H^TR^{-1}z(k+1)+W^{-1}(k)Fx(k/k)$	$(n\times 1)+(n\times 1)$	n
$x(k+1/k+1)=P(k+1/k+1)\{H^TR^{-1}z(k+1)+W^{-1}(k)Fx(k/k)\}$	$(n\times n)\cdot(n\times 1)$	$2n^2-n$
Total		$B_{TI}^C=2nm+(62n^3+15n^2-11n)/6$

sensor network or reclaiming a given (not surely optimal) sensor network depends on the corresponding costs. The resulting optimal Kalman/Lainiotis filters require uniform distribution or near to uniform distribution of measurements into parallel processors. The optimal uniform distribution has the advantages of elimination of idle time for the local processors and of low hardware cost, but it is not always applicable. The optimal filters present high parallelism speedup; this is verified through simulation results and is very important due to the fact that, in most real-time applications, it is essential to obtain the estimate in the shortest possible time.

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful feedback which improved exposition of this paper.

Appendix A. Centralized time varying Kalman filter/Lainiotis filter

In (13) substituting the matrices $P(k+1/k)$, $K(k+1)$ by (10), (11), respectively, and using the matrix inversion lemma[1] (24) is derived:

Table 11
Centralized steady state Kalman filter/Lainiotis filter.

Matrix operation	Matrix dimensions	Calculation burden
$A_{ss}x(k/k)$	$(n\times n)\cdot(n\times 1)$	$2n^2-n$
$B_{ss}z(k+1)$	$(n\times m)\cdot(m\times 1)$	$2nm-n$
$x(k+1/k+1)$ $=A_{ss}x(k/k)+B_{ss}z(k+1)$	$(n\times 1)+(n\times 1)$	n
Total		$B_{SS}^C=2n^2+2nm-n$

$$P(k+1/k+1)=\Big\{\big[F(k+1,k)P(k/k)F^T(k+1,k)+Q(k)\big]^{-1}+H^T(k+1)R^{-1}(k+1)H(k+1)\Big\}^{-1}$$

which can be formulated as

$$P(k+1/k+1)=\big[P^{-1}(k+1/k)+H^T(k+1)R^{-1}(k+1)H(k+1)\big]^{-1} \qquad \text{(A.1)}$$

due to (10).

The vector $x(k+1/k+1)$ in (12) can be written as

$$x(k+1/k+1)=x(k+1/k)-K(k+1)H(k+1)x(k+1/k)+K(k+1)z(k+1)$$

and substituting in the last equation the quantities $x(k+1/k)$, $K(k+1)$ by (9) and (11) we derive:

[1] $(A+BCD)^{-1}=A^{-1}-A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1}$.

Table 12
Decentralized time varying Kalman filter/Lainiotis filter.

Matrix operation	Matrix dimensions	Calculation burden
Local level		
$R_i^{-1}(k+1)$	$(M \times M)$	$(16M^3 - 3M^2 - M)/6$
$H_i^T(k+1)R_i^{-1}(k+1)$	$(n \times M) \cdot (M \times M)$	$2nM^2 - nM$
$B_i(k+1) = H_i^T(k+1)R_i^{-1}(k+1)H_i(k+1)$	$(n \times M) \cdot (M \times n)$ symmetric	$n^2M + nM - 0.5n^2 - 0.5n$
$b_i(k+1) = H_i^T(k+1)R_i^{-1}(k+1)z_i(k+1)$	$(n \times M) \cdot (M \times 1)$	$2nM - n$
Central level		
$F(k+1,k)P(k/k)$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$F(k+1,k)P(k/k)F^T(k+1,k)$	$(n \times n) \cdot (n \times n)$ symmetric	$n^3 + 0.5n^2 - 0.5n$
$W(k) \equiv F(k+1,k)P(k/k)F^T(k+1,k) + Q(k)$	$(n \times n) + (n \times n)$ symmetric	$0.5n^2 + 0.5n$
$W^{-1}(k)$	$(n \times n)$	$(16n^3 - 3n^2 - n)/6$
$\sum_{i=1}^{\rho} B_i(k+1)$	$(n \times n) + \cdots + (n \times n)$ symmetric	$(0.5n^2 + 0.5n)(\rho - 1)$
$W^{-1}(k) + \sum_{i=1}^{\rho} B_i(k+1)$	$(n \times n) + (n \times n)$ symmetric	$0.5n^2 + 0.5n$
$P(k+1/k+1) = [W^{-1}(k) + \sum_{i=1}^{\rho} B_i(k+1)]^{-1}$	$(n \times n)$	$(16n^3 - 3n^2 - n)/6$
$\sum_{i=1}^{\rho} b_i(k+1)$	$(n \times 1) + \cdots + (n \times 1)$	$n(\rho - 1)$
$W^{-1}(k)F(k+1,k)$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$W^{-1}(k)F(k+1,k)x(k/k)$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
$\sum_{i=1}^{\rho} b_i(k+1) + W^{-1}(k)F(k+1,k)x(k/k)$	$(n \times 1) + (n \times 1)$	n
$x(k+1/k+1)$ $= P(k+1/k+1)\{\sum_{i=1}^{\rho} b_i(k+1) + W^{-1}(k)F(k+1,k)x(k/k)\}$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
Total		$B_{TV}^D = (62n^3 + 9n^2 - 23n)/6$ $+ (16M^3 - 3M^2 - M)/6 + n^2M$ $+ 2nM + 2nM^2 + (n^2 + 3n)\rho/2$

Table 13
Decentralized time invariant Kalman filter/Lainiotis filter.

Matrix operation	Matrix dimensions	Calculation burden
Local level		
$b_i(k+1) = H_i^T R_i^{-1} z_i(k+1)$	$(n \times M) \cdot (M \times 1)$	$2nM - n$
Central level		
$FP(k/k)$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$FP(k/k)F^T$	$(n \times n) \cdot (n \times n)$ symmetric	$n^3 + 0.5n^2 - 0.5n$
$W(k) \equiv FP(k/k)F^T + Q$	$(n \times n) + (n \times n)$ symmetric	$0.5n^2 + 0.5n$
$W^{-1}(k)$	$(n \times n)$	$(16n^3 - 3n^2 - n)/6$
$W^{-1}(k) + \sum_{i=1}^{\rho} B_i$	$(n \times n) + (n \times n)$ symmetric	$0.5n^2 + 0.5n$
$P(k+1/k+1) = [W^{-1}(k) + \sum_{i=1}^{\rho} B_i]^{-1}$	$(n \times n)$	$(16n^3 - 3n^2 - n)/6$
$\sum_{i=1}^{\rho} b_i(k+1)$	$(n \times 1) + \cdots + (n \times 1)$	$n(\rho - 1)$
$W^{-1}(k)F$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$W^{-1}(k)Fx(k/k)$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
$\sum_{i=1}^{\rho} b_i(k+1) + W^{-1}(k)Fx(k/k)$	$(n \times 1) + (n \times 1)$	n
$x(k+1/k+1) = P(k+1/k+1)\{\sum_{i=1}^{\rho} b_i(k+1) + W^{-1}(k)Fx(k/k)\}$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
Total		$B_{TI}^D = 2nM + n\rho + (62n^3 + 15n^2 - 17n)/6$

Table 14
Decentralized steady state Kalman filter/Lainiotis filter.

Matrix operation	Matrix dimensions	Calculation burden
Local level		
$d_i(k+1) = \overline{P}_e H_i^T R_i^{-1} z_i(k+1)$	$(n \times M) \cdot (M \times 1)$	$2nM - n$
Central level		
$\sum_{i=1}^{\rho} d_i(k+1)$	$(n \times 1) + \cdots + (n \times 1)$	$n(\rho - 1)$
$A_{ss}x(k/k)$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
$x(k+1/k+1) = A_{ss}x(k/k) + \sum_{i=1}^{\rho} d_i(k+1)$	$(n \times 1) + (n \times 1)$	n
Total		$B_{SS}^D = 2n^2 - 2n + 2nM + n\rho$

$$
\begin{aligned}
&x(k+1/k+1)\\
&\quad = F(k+1,k)x(k/k) - P(k+1/k)H^T(k+1)\\
&\qquad \times \big[H(k+1)P(k+1/k)H^T(k+1)+R(k+1)\big]^{-1}\\
&\qquad \times H(k+1)F(k+1,k)x(k/k) + P(k+1/k)H^T(k+1)\\
&\qquad \times \big[H(k+1)P(k+1/k)H^T(k+1)+R(k+1)\big]^{-1}z(k+1)\\
&\quad = P(k+1/k)P^{-1}(k+1/k)F(k+1,k)x(k/k)\\
&\qquad - P(k+1/k)H^T(k+1)\big[H(k+1)P(k+1/k)H^T(k+1)\\
&\qquad + R(k+1)\big]^{-1}H(k+1)P(k+1/k)P^{-1}(k+1/k)\\
&\qquad \times F(k+1,k)x(k/k) + P(k+1/k)H^T(k+1)\\
&\qquad \times \big[H(k+1)P(k+1/k)H^T(k+1)+R(k+1)\big]^{-1}z(k+1)\\
&\quad = \big\{P(k+1/k) - P(k+1/k)H^T(k+1)\big[H(k+1)P(k+1/k)\\
&\qquad \times H^T(k+1)+R(k+1)\big]^{-1}H(k+1)P(k+1/k)\big\}\\
&\qquad \times P^{-1}(k+1/k)F(k+1,k)x(k/k) + P(k+1/k)H^T(k+1)\\
&\qquad \times \big[H(k+1)P(k+1/k)H^T(k+1)+R(k+1)\big]^{-1}z(k+1)
\end{aligned}
\tag{A.2}
$$

Using the matrix inversion lemma and (A.1) the last equation gives:

$$
\begin{aligned}
&x(k+1/k+1)\\
&\quad = \big[P^{-1}(k+1/k) + H^T(k+1)R^{-1}(k+1)H(k+1)\big]^{-1}\\
&\qquad \times P^{-1}(k+1/k)F(k+1,k)x(k/k) + P(k+1/k)H^T(k+1)\\
&\qquad \times \big[H(k+1)P(k+1/k)H^T(k+1)+R(k+1)\big]^{-1}z(k+1)\\
&\quad = P(k+1/k+1)P^{-1}(k+1/k)F(k+1,k)x(k/k)\\
&\qquad + P(k+1/k)H^T(k+1)\big[H(k+1)P(k+1/k)H^T(k+1)\\
&\qquad + R(k+1)\big]^{-1}z(k+1)
\end{aligned}
\tag{A.3}
$$

Moreover, the non-singularity of the matrices $R(k+1)$, $P(k+1/k)$ and (A.1) allow us to write:

$$
\begin{aligned}
&P(k+1/k)H^T(k+1)\big[H(k+1)P(k+1/k)H^T(k+1)\\
&\qquad + R(k+1)\big]^{-1}\\
&\quad = \big[H^T(k+1)R^{-1}(k+1)H(k+1) + P^{-1}(k+1/k)\big]^{-1}\\
&\qquad \times H^T(k+1)R^{-1}(k+1)\\
&\quad = P(k+1/k+1)H^T(k+1)R^{-1}(k+1)
\end{aligned}
\tag{A.4}
$$

By (A.4) Eq. (A.3) is written

$$
\begin{aligned}
x(k+1/k+1) &= P(k+1/k+1)P^{-1}(k+1/k)F(k+1,k)x(k/k)\\
&\quad + P(k+1/k+1)H^T(k+1)R^{-1}(k+1)z(k+1)\\
&= P(k+1/k+1)\big\{P^{-1}(k+1/k)F(k+1,k)x(k/k)\\
&\quad + H^T(k+1)R^{-1}(k+1)z(k+1)\big\}
\end{aligned}
$$

and by (10) the proof of (25) is complete.

Appendix B. Per recursion calculation burdens of Kalman/Lainiotis filters

The calculation burden B_{TV}^C of centralized time varying Kalman filter/Lainiotis filter is computed by (24)–(25) and shown in Table 9.

The calculation burden B_{TI}^C of centralized time invariant Kalman filter/Lainiotis filter is computed by (26)–(27) and shown in Table 10.

The calculation burden B_{SS}^C of centralized steady state Kalman filter/Lainiotis filter is computed by (31) and shown in Table 11.

The calculation burden B_{TV}^D of decentralized time varying Kalman filter/Lainiotis filter includes the burden of local level computed by (36)–(37) and the burden of central level computed by (38)–(39) and is shown in Table 12.

The calculation burden B_{TI}^D of decentralized time invariant Kalman filter/Lainiotis filter includes the burden of local level computed by (40) and the burden of central level computed by (41)–(42) and is shown in Table 13.

The calculation burden B_{SS}^D of decentralized steady state Kalman filter/Lainiotis filter includes the burden of local level computed by (44) and the burden of central level computed by (45) and is shown in Table 14.

References

[1] R.E. Kalman, A new approach to linear filtering and prediction problems, Trans. ASME: J. Basic Eng. 82 (1960) 34–45.

[2] D.G. Lainiotis, Partitioned linear estimation algorithms: Discrete case, IEEE Trans. Automat. Control AC-20 (1975) 255–257.

[3] R. Olfati-Saber, Distributed Kalman filters with embedded consensus filters, in: 44th IEEE Conference on Decision and Control, Seville, Spain, December 2005, pp. 8179–8184.

[4] H. Medeiros, J. Park, A.C. Kak, Distributed object tracking using a cluster-based Kalman filter in wireless camera networks, IEEE Trans. Spec. Topics Signal Process. 2 (4) (August 2008) 448–463.

[5] R. Olfati-Saber, R.M. Murray, Consensus problems in networks of agents with switching topology and time-delays, IEEE Trans. Automat. Control 49 (9) (September 2004) 1520–1533.

[6] Ruggero Carli, Alessandro Chiuso, Luca Schenato, Sandro Zampieri, Distributed Kalman filtering based on consensus strategies, IEEE J. Sel. Areas Commun. 26 (4) (May 2008) 622–632.

[7] A.T. Kamal, C. Ding, B. Song, J.A. Farrell, A.K. Roy-Chowdhury, A generalized Kalman consensus filter for wide-area video networks, in: 50th IEEE Conference on Decision and Control and European Control Conference (CDC–ECC), Orlando, FL, USA, December 2011, pp. 7863–7869.

[8] J.L. Speyer, Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem, IEEE Trans. Automat. Control 24 (2) (February 1979) 266–269.

[9] B.S.Y. Rao, H.F. Durrant-Whyte, J.A. Sheen, A fully decentralized multi-sensor system for tracking and surveillance, Int. J. Robot. Res. 12 (1) (February 1993) 20–44.

[10] V. Delouille, R. Neelamani, R. Baraniuk, Robust distributed estimation in sensor networks using the embedded polygons algorithm, in: Proceedings of 3rd Int. Symp. Info. Processing Sensor Networks, Berkeley, CA, April 2004, pp. 405–413.

[11] S. Kirti, A. Scaglione, Scalable distributed Kalman filtering through consensus, in: IEEE International Conference on Acoustics, Speech and Signal Processing, April 2008, pp. 2725–2728.

[12] I.D. Schizas, A. Ribeiro, G.B. Giannakis, Consensus in ad hoc WSNs with noisy links—Part I: Distributed estimation of deterministic signals, IEEE Trans. Signal Process. 56 (1) (January 2008) 350–364.

[13] I.D. Schizas, G.B. Giannakis, S.I. Roumeliotis, A. Ribeiro, Consensus in ad hoc WSNs with noisy links—Part II: Distributed estimation and smoothing of random signals, IEEE Trans. Signal Process. 56 (4) (April 2008) 1650–1666.

[14] I.D. Schizas, G. Mateos, G.B. Giannakis, Distributed LMS for consensus-based in-network adaptive processing, IEEE Trans. Signal Process. 57 (6) (June 2009) 2365–2382.

[15] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, G.B. Giannakis, Distributed compression-estimation using wireless sensor networks, IEEE Signal Process. Mag. 23 (4) (July 2006) 27–41.

[16] G. Mateos, I.D. Schizas, G.B. Giannakis, Distributed recursive least-squares for consensus-based in-network adaptive estimation, IEEE Trans. Signal Process. 57 (11) (November 2009) 4583–4588.

[17] Y.M. Zhu, Z.S. You, J. Zhou, K.-S. Zhang, X.R. Li, The optimality for the distributed Kalman filtering fusion with feedback, Automatica 37 (2001) 1489–1493.

[18] Y. Sheng, X. Hu, P. Ramanathan, Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor networks, in: 4th International Symposium on Information Processing in Sensor Networks, April 2005.

[19] H. Durrant-Whyte, Data fusion in sensor networks, in: 4th International Symposium on Information Processing in Sensor Networks, April 2005.

[20] D.G. Lainiotis, K.N. Plataniotis, C.J. Charalampous, Distributed computing filters: Multisensor marine applications, in: Proceedings of IEEE OCEANS 92, vol. I, 1992, pp. 265–270.
[21] D.G. Lainiotis, K.N. Plataniotis, M. Papanikolaou, P. Papaparaskeva, Discrete Riccati equation solutions: Distributed algorithms, Math. Probl. Eng. 2 (1996) 319–332.
[22] D.G. Lainiotis, P. Papaparaskeva, Efficient algorithms of clustering adaptive nonlinear filters, IEEE Trans. Automat. Control 44 (7) (July 1999) 1454–1459.
[23] K.N. Plataniotis, S.K. Katsikas, D.G. Lainiotis, A.N. Venetsanopoulos, Optimal seismic deconvolution: Distributed algorithms, IEEE Trans. Geosci. Remote Sens. 36 (3) (May 1998) 779–792.
[24] K.N. Plataniotis, D.G. Lainiotis, Multi-sensor estimation: New distributed algorithms, Math. Probl. Eng.: Theory Methods Appl. 3 (1996) 27–52.
[25] D.G. Lainiotis, Optimal adaptive estimation: Structure and parameter adaptation, IEEE Trans. Automat. Control 16 (1971) 160–170.
[26] D.G. Lainiotis, Partitioning: A unifying framework for adaptive systems I: Estimation, IEEE Proc. 64 (1976) 1126–1143.
[27] D.G. Lainiotis, S.K. Katsikas, Linear and nonlinear Lainiotis filters: A survey and comparative evaluation, in: IFAC Workshop on Expert Systems and Signal Processing in Marine Automation, Denmark, 1989.
[28] H.R. Hashemipour, S. Roy, A.J. Laub, Decentralized structures for parallel Kalman filtering, IEEE Trans. Automat. Control 33 (1988) 88–94.
[29] N.A. Carlson, Federated square root filter for decentralized parallel processes, IEEE Trans. Aerosp. Electron. Syst. 26 (3) (1990) 517–525.
[30] S. Roy, R.A. Iltis, Decentralized linear estimation in correlated measurement noise, IEEE Trans. Aerosp. Electron. Syst. 27 (6) (1991) 939–941.
[31] U.A. Khan, J.M.F. Moura, Distributing the Kalman filter for large-scale systems, IEEE Trans. Signal Process. 56 (10) (October 2008) 4919–4935.
[32] N. Assimakis, M. Adam, M. Koziri, S. Voliotis, Optimal distributed Kalman and Lainiotis filters: Optimal uniform distribution of measurements into local processors, in: Proceedings of 16th International Workshop on Systems, Signals and Image Processing (IWSSIP), 2009, pp. 19–24.
[33] N. Assimakis, M. Adam, Discrete time Kalman and Lainiotis filters comparison, Int. J. Math. Anal. 1 (2007) 635–659.
[34] N. Assimakis, D. Lainiotis, S. Katsikas, F. Sanida, A survey of recursive algorithms for the solution of the discrete time Riccati equation, Nonlinear Anal. 30 (1997) 2409–2420.
[35] B.D.O. Anderson, J.B. Moore, Optimal Filtering, Dover Publications, New York, 2005.
[36] D.G. Lainiotis, S.K. Katsikas, S.D. Likothanassis, Optimal seismic deconvolution, Signal Process. 15 (1988) 375–404.

Nicholas Assimakis was born in 1962. He received his Ph.D. degree in Signal Processing from the Department of Computer Engineering and Informatics of the Polytechnic School of the University of Patras, Greece in 1993. He is currently Associate Professor at the Department of Electronics of the Technological Educational Institute of Lamia. His research interests are in digital signal processing, estimation theory, filtering, algorithm design and development.

M. Adam received her Ph.D. degree in Mathematics from National Technical University of Athens in 2001. Her current position is Assistant Professor at the Department of Computer Science and Biomedical Informatics of the University of Central Greece, where she teaches linear algebra, advanced calculus and applied mathematics. Her research interests include linear algebra, numerical linear algebra, spectral analysis, matrix theory and applications in dynamical systems, control theory, algorithms and statistics.

Maria Koziri received the Dip.Eng. degree in computer engineering from the Technical University of Crete, Greece, in 2003 and the Ph.D. degree from the University of Thessaly, Greece, in 2007. Since 2007 she has been in the Department of Computer and Communication Engineering at University of Thessaly, Greece. Her research interests include video compression, scalable video coding, rate-distortion optimization, as well as computer architecture. Results of her research work are published in some of the most well known international conferences in multimedia and signal processing areas, such as IEEE's ICME and ICIP.

Stamatis Voliotis was born in Chalkida, Greece, in 1963. He honored the degree of Computer Engineering and Informatics from University of Patras, Greece in 1985, the M.Sc. in Electrical Computer Engineering from Syracuse University, USA in 1989, and the Ph.D. from University of Patras, Greece in 1990. Currently he is a Professor at Technological Educational Institute of Chalkida. His research interests are in the area of computer automation architecture and network processing, mobile/wireless communications and robotics navigation and control.

Konstantinos Asimakis, born in 1988, is an undergraduate student at Computer Engineering and Informatics Department of University of Patras. He is currently a researcher at RACTI "Diophantus". His research interests include LTE mobile networks, genetic algorithms, neural networks, regression algorithms and natural language processing.