

# DISTRIBUTED COMPUTING FILTERS : MULTISENSOR MARINE APPLICATIONS

D.G. Lainiotis, K. Plataniotis, and C. Charalampous

Florida Institute of Technology, Melbourne, FL and  
University of Patras, Patras, Greece.

**Abstract** - The distributed multisensor estimation problem is considered in this paper, and both the centralized and the decentralized approach are discussed. The Kalman filter based approaches are compared with the Lainiotis filter based approach, in terms of their computational efficiency and transmission requirements. The comparative evaluation shows that using the Lainiotis filter, both in its sequential and distributed form, orders of magnitude decrease in computational requirements and processing time can be achieved.

## 1. INTRODUCTION

The multisensor estimation problem is of considerable practical significance and applicability in such marine applications as marine geophysical data processing for oil exploration, underwater target tracking etc. In these applications a large number of different sensors is employed to obtain information, which is then used for estimation, detection or control purposes. The basic goal of such multisensor systems is to provide redundancy, increase robustness and to improve the system performance obtaining high performance speeds.

There are two major approaches to obtain the required estimates for such multiple-sensor systems :

*The centralized approach*, where all the sensor observations (system's data) are transmitted from the local sensor positions to a central location for processing, using an estimation algorithm, such as the well known and widely used Kalman filter [7] or Lainiotis linear filters [15,8,14]. This approach, developed in the classical theory, requires large communication bandwidth for sensor information transmission, time delays and high computational load at the central processor.

*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 global estimate. A lot of studies have recently been reported using this approach. In [1]-[5] the algorithms used to obtain the

optimal system estimates are based on the different Kalman filter (KF) formulations, namely the standard covariance-type KF, the information-type KF or the square-root KF. In [11] Lainiotis et al, have obtained the distributed formulation of the Lainiotis per-step partitioning filter.

A comparative evaluation of the Kalman filter-based approach and the Lainiotis filter-based approach, is presented in this paper. The comparison study is based on the computational requirements of the algorithms, both at the local level and at the central processor, and also on their communication cost. It is shown that the Lainiotis filter based approach alleviates the computational complexity of the Kalman filter based approach, offering orders of magnitude reduction in processing time and communication bandwidth.

The paper is organized as follows : In section 2 the mathematical description of the problem is presented and the fundamental assumptions for distributed estimation are given. In section 3, the centralized approach is considered and the comparison of the above filters is given. In section 4, the decentralized approach is considered and the computational and communication requirements of the filters are presented. Finally, section 6 summarizes the conclusions.

## 2. PROBLEM FORMULATION

The system's state dynamics are described by the following state transition equation :

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

where

$x(k)$  is the  $n$ -dimensional state vector, whose initial value  $x(0)$  is Gaussian distributed, with known mean  $x(0/0)$  and covariance  $P(0/0)$ ,

$\Phi(k+1,k)$  is the  $n \times n$  state transition matrix, and

$w(k)$  is the input noise, which is assumed Gaussian random process with zero mean  $E[w(k)]=0$ , and known covari-

ance  $E[w(k)w^T(j)] = Q(k)\delta_{kj}$ , independent of the initial state  $x(0)$ .

A network of sensors takes observations  $z(k)$  of the system's state according to the following dynamic measurement equation :

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

where

$z(k+1)$  is the  $m$ -dimensional global measurement vector,

$H(k+1)$  is the  $m \times n$  global observation matrix,

$v(k+1)$  is the measurement noise which is assumed Gaussian process with zero mean  $E[v(k)] = 0$ , and known covariance  $E[v(k)v^T(j)] = R(k)\delta_{kj}$ , independent of the initial state  $x(0)$  and the process noise  $w(k)$ .

Given the measurement record  $L_k = \{z(1), z(2), \dots, z(k)\}$  it is desired to obtain the optimal, in the minimum mean square error sense (mmse) state estimate  $x(k/k)$  of  $x(k)$ .

Under the decentralized/distributed scenario, the observation vector  $z(k)$  is partitioned into  $r$  subvectors, each of them corresponding at each local sensor system with dimension  $m_i$  [1].

$$z(k) = [ z_1^T \quad z_2^T \quad \dots \quad z_r^T ]$$

$$H(k) = [ H_1^T(k) \quad H_2^T(k) \quad \dots \quad H_r^T(k) ]^T$$

$$v(k) = [ v_1^T(k) \quad v_2^T(k) \quad \dots \quad v_r^T(k) ]$$

*Assumption I* : The noise partitions are assumed uncorrelated [1]. This assumption is valid in many cases, but in general we can put constraints on the choice of the local partitions. Recently, some schemes have been proposed, relaxing this assumption [4]

$$E[v(k)v^T(k)] = \begin{bmatrix} R_1(k) & & 0 \\ & R_2(k) & \\ 0 & & R_r(k) \end{bmatrix}$$

*Assumption II* : For simplicity in this paper we assume that the state transition description is the same at every subsystem and at the center location.

Combining the above equations, the local measurement equation based on the state vector is given by :

$$z_i(k+1) = H_i(k+1) x(k+1) + v_i(k+1) \quad (3)$$

where

$z_i(k+1)$  is the  $m_i$ -dimensional measurement vector at the local processor subsystem  $i$ ,

$H_i(k+1)$  is the  $m_i \times n$  local observation matrix,

$v_i(k+1)$  is the local measurement noise vector,

with the restriction that  $m = \sum_{i=1}^r m_i$

Given all the local measurement records  $L_i = \{z_i(1), z_i(2), \dots, z_i(k)\}$ ,  $i=1, \dots, r$  available at the associate local processors and the local estimates  $x_i(k/k)$ ,  $i=1, \dots, r$ , it is desired to obtain the optimal, in the minimum mean square error sense (mmse) global filter estimate  $x(k/k)$  of  $x(k)$ .

In many marine-related problems, the dimensionality of the state vector ( $x(k)$ ) is 9 (position, velocity, acceleration in x-y-z axes), while the total number of sensors can be from a few hundred to several thousands. Such a typical example is the data processing for oil exploration that often employs arrays with thousands of hydrophones in order to capture the seismic signals. This typical example will be used in the following for comparison purposes.

### 3. CENTRALIZED ESTIMATION

When centralized estimation is considered, all the sensor information is transmitted to the central location for processing. The state-space model is described by (1) and (2). The well known Kalman filter (KF), and a member of the Lainiotis filters family, namely the linear Lainiotis per-step partitioning filter (LLPSPF), can be used in order to obtain the optimal mmse estimate, based on the global measurement record.

Both these algorithms have the same performance for linear Gaussian models, since they are different realizations of the same optimal mmse estimator. However they differ at the computational requirements, especially when they are applied to multisensor problems.

It should be noted that the computational requirements of a given filter are a realistic measure of its practicality and usefulness, since these quantities determine the required computing power and processing time.

The general assumptions under which the comparative evaluation is done, are given in [9]-[10]. Moreover, only real time computation of the filter gains is considered because :

- the number of measurements may not be known in advance,
- the memory retrieval time may be too costly,
- the sampling rate may change on-line, and finally
- the model can be piecewise linear.

The general computational requirements, in terms of normalized operations, of the above algorithms are shown in table I,, while Fig. 1 shows the computational requirements of the filters when time invariant or periodic in time models are considered., keeping the state dimensionality fixed (n=9) and varying the number of sensors from 1-5000. For the typical marine application, where the state dimensionality is 9, and the number of sensors are 1000, the normalized operations of both the algorithms are shown in table II.

Table I

Centralized approach, Normalized operations [8,13]

KF	$7.5n^3 + 6.5n^2 + 7.5n^2 m + 7.5nm^2 + 13mn^2 + 6m^2 + 2.5m^3 + 32.5m - n - 6$
LSPSPF (preliminary operations)	$2.5m^3 + 10m^2 + 8.5nm^2 + 20n^2 m + 4.5mn^2 + 0.5n^2 + 32.5m - 0.5n - 6$
LLPSPF (per step calculations)	$12.5n^3 + 24n^2 + 10nm + 63.5n - 12$

---

Table II

Centralized approach, Normalized operations

	Time invariant models	Periodic models
KF	2,57E+009	2,57E+009
LLPSPF	1,01E+005	1,27E+005

---

It can be seen , that the LLPSPF requires remarkable less computations than the Kalman filter, especially when the number of sensors is increasing.

For example, when the state vector dimensionality is nine (9) (target tracking) and the number of sensors (m) is 1000, the Lainiotis filter has four (4) orders of magnitude less computational requirements than the Kalman filter.

The main drawbacks of the centralized approach is that re-

quires large communication bandwidth for data transmission, time delays and high computational load at the central processor. In order to overcome these deficiencies, and also to achieve robustness and redundancy, the decentralized approach can be used.

#### 4. DECENTRALIZED ESTIMATION

In this approach, the data obtained from the different sensor subsystems are processed locally and the results are transmitted to the central processor (data fusion center). The state-space model is described by (1) and (3). The decentralized Kalman filter in its covariance and information-type formulation given in [1]-[6] as well as the decentralized Lainiotis filter in its covariance and information-type formulation given in [11], can be used in order to construct the global mmse estimate using the local ones.

*Remark 1 :* The Lainiotis decentralized linear per-step algorithm does not compute the optimal estimate at the local level but only the nominal part of it. This is the only quantity required by the central processor in order to compute the overall optimal estimate. If the local optimal estimate of the state is required this can be done by the local processor while the central processor computes the global estimate in parallel without affecting the total throughput time.

*Remark 2 :* The local processor equations of the distributed Lainiotis filter, are non-recursive simple multiplications, memoryless, and can be implemented using a very simple processor or even an optical device.

*Remark 3 :* Both Kalman and Lainiotis decentralized filters are again different realizations of the same optimal mse estimator, thus their performance for linear Gaussian models is the same.

The general comparative evaluation of the above filters, with respect to their computational requirements (in terms of normalized operations) and their communication cost (in terms of memory words) are shown in tables III-IV. Considering a specific configuration (Number of local subsystems r=10), figs. 2-5 show a plot of the normalized operations as a function of the number of sensors at each subsystem, both for time invariant and periodic models. Finally, the results for the same typical marine problem are presented in tables V-VI.

The notation used throughout the tables and the graphs is :

- DKF : Decentralized Kalman filter (covariance type)
- DIKF : Decentralized Kalman filter (information type)
- DLF : Decentralized Lainiotis filter (covariance type)
- DILF : Decentralized Lainiotis filter (information type).

From these graphs and tables the following important conclusions can be made :

- The decentralized formulations of the Lainiotis filters require considerably less computations than the decentralized Kalman filter formulations. This results in significant processing time savings, both in the local level and in the total throughput. For example, the target tracking case ( $n=9$ ,  $m=1000$ ), the decentralized versions of the Lainiotis filter are two (2) orders of magnitude faster than the decentralized Kalman filter and incredible five (5) orders of magnitude faster than the centralized Kalman filter.

- The communication requirements of the decentralized Lainiotis filters are less than that of the decentralized Kalman filters, when time invariant models are considered. Specifically, in both time invariant and periodic cases the communication cost of the decentralized Lainiotis filter is one order of magnitude less than that of the decentralized Kalman filter.

## 5. CONCLUSIONS

The centralized and decentralized approaches to the multisensor estimation problem were discussed in this paper. The Kalman filter based approach was compared to Lainiotis filter based approach, in terms of their computational requirements and communication costs. The comparative evaluation of these approaches have been made using a typical scenario for marine multisensor problem. It has been shown that the Lainiotis filter based approach achieves orders of magnitude decrease both in processing time and communication requirements, without affecting the overall performance.

## REFERENCES

- [1] H.R Hashemipour, S. Roy, A.J. Laub, "Decentralized structures for parallel Kalman filtering", IEEE Trans. on Automatic Control, AC-33, n, 1, 88-94, 1988.
- [2] S. Roy, R.H. Hashemi, A.J. Laub, "Square root parallel Kalman filtering using reduced-order local filters", IEEE trans. on Aerospace and Electronic Systems, vol. 27, n. 2, 276-288, 1991.
- [3] N.A. Carlson, "Federated square root filter for decentralized parallel processes", IEEE trans. on Aerospace and Electronic Systems, vol. 26, n. 3, 517-525, 1990.
- [4] S. Roy, R.A. Iltis, "Decentralized linear estimation in correlated measurement noise", IEEE trans. on Aerospace and Electronic Systems, vol. 27, n. 6, 939-941, 1991.
- [5] T.H. Kerr, Comments on "Federated square root filter for decentralized parallel processes", IEEE trans. on Aerospace and Electronic Systems, vol. 27, n. 6, 946-949, 1991.
- [6] T.M. Berg, H.F. Durrant-Whyte, "Model distribution in decentralized multi-sensor data fusion", Proc. of the American Control Conference, 2292-2293, 1991.
- [7] R.E. Kalman, "A new approach to linear filtering and prediction problems", Trans ASME J Bas Eng. Ser D, 35-45, 1960.
- [8] D.G. Lainiotis, "Partitioning : A unifying framework for adaptive systems I : Estimation", IEEE Proc., v. 64, 1126-1143, 1976.
- [9] S.K. Katsikas, S.D. Likothanasis, D.G. Lainiotis, "On the parallel implementations of linear Kalman and Lainiotis filters and their efficiency", Signal Processing, vol. 25, n. 3, 289-306, 1991.
- [10] D.G. Lainiotis, S.K. Katsikas, S.D. Likothanasis, "Adaptive deconvolution of seismic signals : Performance, computational analysis, parallelism", IEEE Trans. Acoustic, Speech and Signal Processing, ASSP-36, 1715-1734, 1988.
- [11] D.G. Lainiotis, K. Plataniotis, C. Charalampous, "Distributed Lainiotis filters", submitted in Signal Processing, 1992.
- [12] D.G. Lainiotis, P.K. Giannakopoulos, S.K. Katsikas, "Optimal state estimation for uncertain, time varying systems with non Gaussian initial state, ICASP, pp. 1877-1880, 1991.
- [13] D.G. Lainiotis et al. "Real time ship motion estimation using Lainiotis filters", IFAC Workshop on Expert systems and Signal Processing in Marine Automation, Denmark, 1989.
- [14] D.G. Lainiotis, S.K. Katsikas, "Linear and nonlinear Lainiotis filters : A survey and comparative evaluation", IFAC Workshop on Expert systems and Signal Processing in Marine Automation, Denmark, 1989.
- [15] D.G. Lainiotis, "Optimal adaptive estimation : Structure and parameter adaptation", IEEE Trans. on AC, v. 16, 160-170, 1971.

Table III

Decentralized approach, Normalized Operations

DKF(Local level)	$12.5n^3 + 29n^2 + 64.5n + 2.5m_i^3 + 6m_i^2 + 32.5m_i + 7.5nm_i + 7.5m_i^2 + 13nm_i - 12$
DKF(Central level)	$12.5n^3 + 29n^2 + 58.5n - 12 + r(0.5n + 1.5n)$
DKF(Total)	$25n^3 + (58 + 0.5r)n^2 + (123 + 1.5r)n + 2.5m_i^3 + 6m_i^2 + 32.5m_i + 7.5m_i^2 + 7.5m_i n + 13m_i n - 24$
DIKF(Local level)	$20n^3 + 16n^2 + 36n + 5nm_i^2 + 2.5m_i^3 + 6.5nm_i - 12$
DKF(Central level)	$22.5n^3 + (27 + 0.5r)n^2 + (63.5 + 1.5r)n - 12$
DIKF(Total)	$42.5n^3 + (43 + 0.5r)n^2 + (99.5 + 1.5r)n + 5nm_i^2 + 2.5m_i^3 + 6.5nm_i - 12$
DLF(Local level)	$5n^2 - 2n + 5nm_i$
DLF(Central level)	$12.5n^3 + (34 + 0.5r)n^2 + (63 + r)n - 12$
DLF(Total)	$12.5n^3 + (39 + 0.5r)n^2 + (61 + r)n + 5nm_i - 12$
DILF(Local level)	$-n + 5nm_i$
DLF(Central level)	$12.5n^3 + (34 + 0.5r)n^2 + (63 + r)n - 12$
DILF(Total)	$12.5n^3 + (34 + 0.5r)n^2 + (62 + r)n + 5nm_i - 12$

Table IV

Decentralized Approach, Communication requirements

DKF	$0.5n^2 + 1.5n$
DIKF	$0.5n^2 + 1.5n$
DLF	$n$
DILF	$n$

Table V

Decentralized Processing  
(Normalized OP's at a local processor)

$n=9, m=1000, r=10, m_i=100$

	Time invariant	Periodic
Kalman filter	3,32E+006	3,32E+006
Information KF	4,92E+005	4,92E+005
Lainiotis filter	4,88E+003	4,92E+003
Information LF	4,49E+003	4,52E+003

Table VI

Decentralized Processing  
(Total throughput)

$n=9, m=1000, r=10, m_i=100$

	Time invariant	Periodic
Kalman filter	3,33E+006	3,33E+006
Information KF	5,11E+005	5,12E+005
Lainiotis filter	1,78E+004	1,78E+004
Information LF	1,74E+004	1,74E+004

Table VII

Decentralized Processing  
(Communication Requirements)

$n=9, m=1000, r=10, m_i=100$

	Time invariant	Periodic
Kalman filter	0,54E+002	0,54E+002
Information KF	0,54E+002	0,54E+002
Lainiotis filter	0,90E+001	0,90E+001
Information LF	0,90E+001	0,90E+001

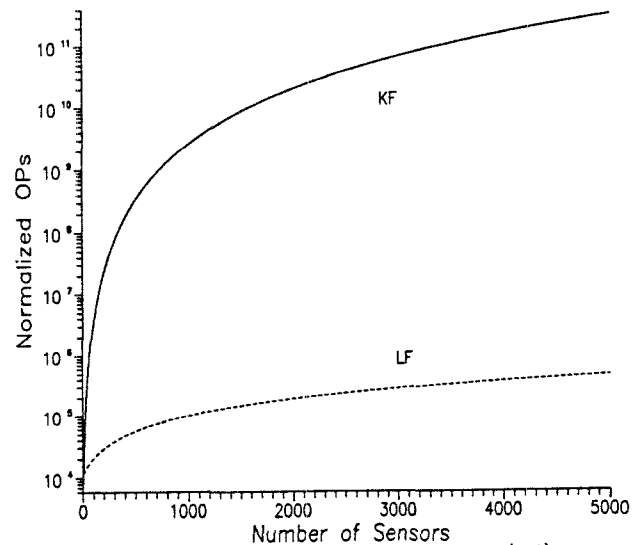


Fig.1 Centralized Processing, Time Invariant models (n=9)

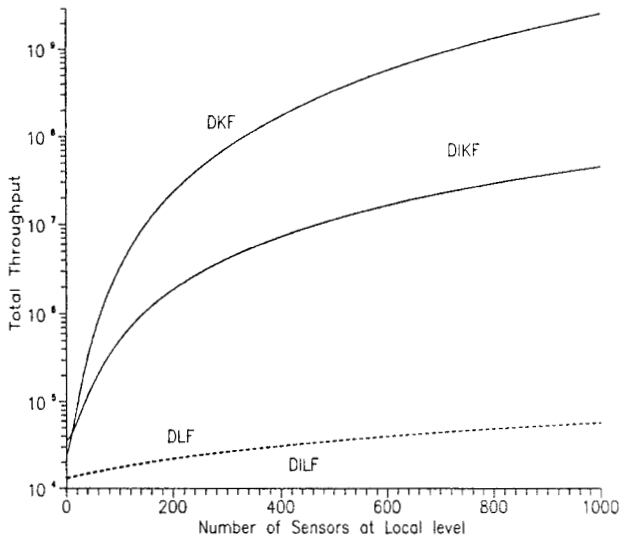


Fig. 2 Time Invariant Models ( $n=9, r=10$ )

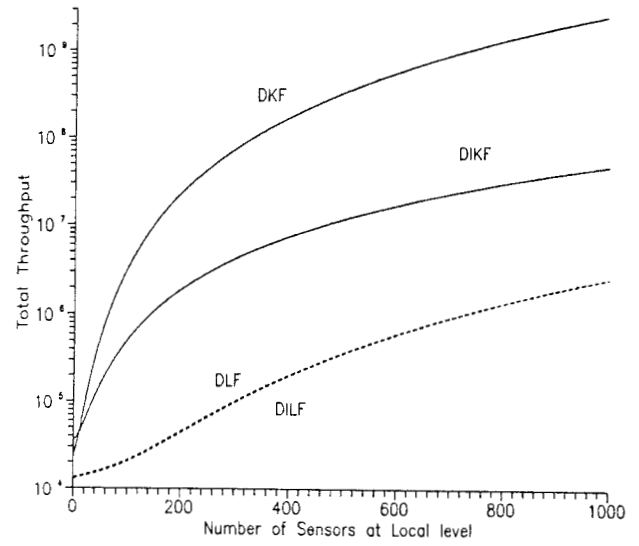


Fig. 4 Periodic Models ( $n=9, r=10$ )

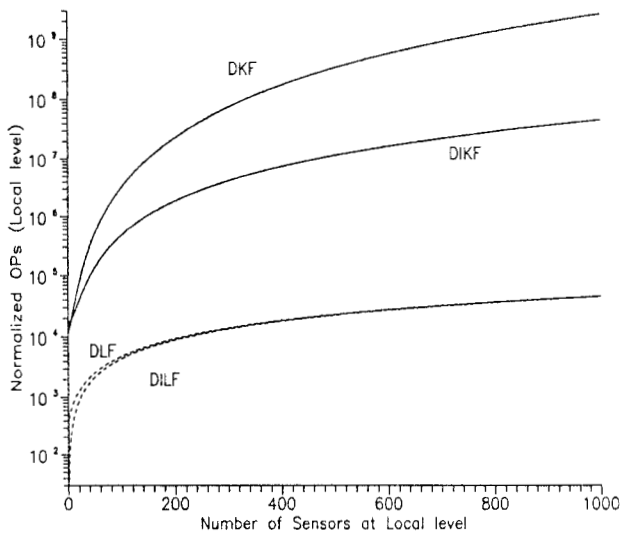


Fig. 3 Time Invariant Models ( $n=9, r=10$ )

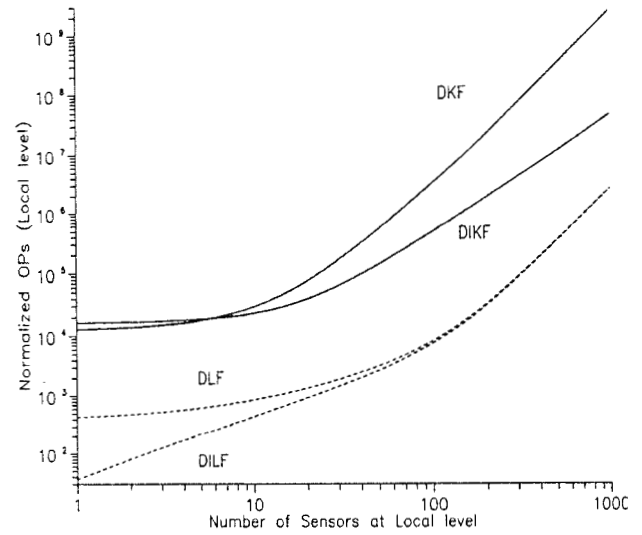


Fig. 5 Periodic Models ( $n=9, r=10$ )