afcet

RAI / IPAR' 86

IASTED

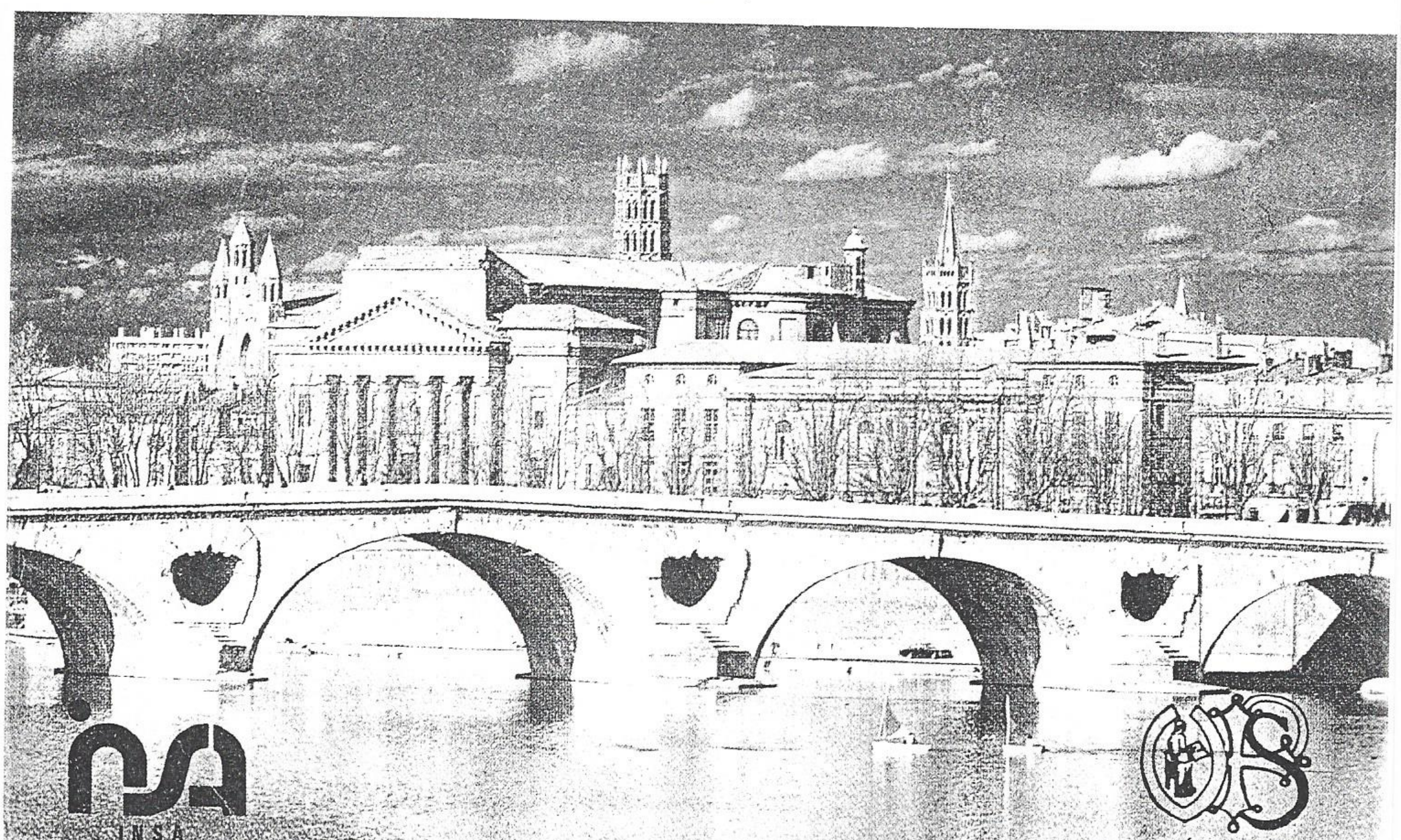
ROBOTIQUE ET INTELLIGENCE ARTIFICIELLE
ROBOTICS AND ARTIFICIAL INTELLIGENCE

IDENTIFICATION ET RECONNAISSANCE DES FORMES
IDENTIFICATION AND PATTERN RECOGNITION

TOME III

UNIVERSITÉ PAUL SABATIER

18-20 JUIN 1986
JUNE 18-20, 1986



INSA
avenue de Rangueil
31077 TOULOUSE FRANCE

UNIVERSITÉ PAUL SABATIER
118, route de Narbonne

Proceedings of the IASTED International Symposium and Exhibition : Identification and Pattern Recognition / Identification et Reconnaissance des Formes

International Programme Committee / Comité Scientifique :

J. AGUILAR	France
E. BACKER	Netherlands
C. BELLISSANT	France
P. BRATLEY	Canada
M. BRIOT	France
S. CASTAN	France
P. COIFFET	France
F. CONRAD	Denmark
F. CUOZZO	France
M. DE BRUCQ	France
D. DUBOIS	France
B. DUBUISSON	France
J.-O. EKLUNDH	Sweden
H. EMPTOZ	France
H. FARRENY	France
O. FAUGERAS	France
A. FAURE	France
L. GOLDFARB	Canada
T.-C. HSIA	U.S.A.
M. LABARRERE	France
Ph. DE LARMINAT	France
J.-C. LATOMBE	France
V. LATTUATI	France
J.-F. LE MAITRE	France
A. LHOTE	France
G. LIEGEOIS	France
P. LOPEZ (Chairman)	France
M. LOPEZ DE MANTARAS	Spain
M. LUQUE	Spain
R.-C. MICHELINI	Italie
C. MIRA	France
R. MOHR	France
M. NAGAO	Japan
M. NOUGARET	France
R.-P. PAUL	U.S.A.
H. PRADE	France
M. PRAJOUX	France
M. RENAUD	France
J.-C. SIMON	France
C. SOL	France
G. STAMON	France
S. TANIMOTO	U.S.A.
M. TRANNOY	France
P. VIDAL	France
M. MALVACHE	France

Steering Committee / Comité d'Organisation :

- P. LOPEZ, I.N.S.A.
- G. STAMON, A.F.C.E.T.
- S. CASTAN, U.P.S.
- G. FAYSSE, U.P.S.
- E. MALLET, I.N.S.A.

Publication Committee / Comité de Publication :

- P. LOPEZ,
- G. STAMON,
- S. CASTAN.

Adaptive Filtering Algorithms and Target Tracking

D.G. LAINIOTIS, V.C. MOUSSAS

School of Engineering
Department of Computer Engineering
University of Patras
26500 Patras, GREECE

ABSTRACT

Two forms of the Adaptive Lainiotis Filter (ALF) are analysed and compared to the commonly used Extended Kalman Filter (EKF), with respect to performance, computational and storage requirements and convergence. The above non-linear filters are applied to the target tracking and their specific performances are studied. Specifically, the filters are implemented for the manoeuvering target tracking problem, and their estimation errors and convergence, as well as the computational requirements (operation counts) and storage requirements (memory) for all three algorithms, are investigated and analysed, over a variety of tracking schemes and multisensor problems. Finally, useful conclusions are drawn on the relative merits of the above non-linear filtering algorithms through a comparative analysis and extensive Monte Carlo simulations.

KEYWORDS

Adaptive filtering, parameter identification, parallel processing, tracking, non-linear filtering.

I. INTRODUCTION

Estimation algorithms are widely applied in a variety of linear and non-linear problems. The design of these algorithms (estimators) necessitates the availability of a mathematical model which represents the underlying "physical" situation.

Unfortunately, in most physical situations, complete knowledge of the model is neither usually available, nor readily forthcoming. If, moreover, the estimator design is to be done in real time, it constitutes an adaptive estimation problem. The most basic adaptive estimation problem consists of linear models with parametric uncertainty, described by the following model equations:

$$\frac{dx(t)}{dt} = F(t,\theta)\cdot x(t) + u(t) \tag{1}$$

$$z(t) = H(t,\theta)\cdot x(t) + v(t) \tag{2}$$

where $x(t)$ and $z(t)$ are the n- and m-dimensional state and measurement processes, respectively, $\{u(t)\}$ and $\{v(t)\}$ are the input and measurement noise random processes which, conditioned on θ, are independent zero-mean white-Gaussian processes with covariances $Q(t,\theta)$ and $R(t)$, respectively. The initial state vector $x(t_o) = x_o$ is independent of $\{u(t)\}$ and $\{v(t)\}$, and has a θ-conditional Gaussian density $P(t_o/\theta)$ with mean $\hat{x}(t_o/t_o,\theta)$ and covariance $P(t_o,t_o/\theta)$.

The above model is specified up to a set of unknown parameters, denoted by the vector θ. The parameter vector θ is considered a random variable with known a priori probability density function $P(\theta/t_o) \equiv P(\theta)$. The obvious approach to the solution of the above problem is to augment the state-vector with θ, namely $x_\alpha(t) \equiv [x^T(t) \vdots \theta^T]^T$. Thus the model defining equations become the non-linear equations:

$$\frac{dx_\alpha(t)}{dt} = \varphi_\alpha[x_\alpha(t),t] + g_\alpha[x_\alpha(t),t]\cdot u(t) \tag{3}$$

$$z(t) = h_\alpha[x_\alpha(t),t] + v(t) \tag{4}$$

where:

$$\varphi_\alpha[x_\alpha(t),t] \equiv [x^T(t)F^T(t,\theta) \vdots \emptyset]^T$$

$$g_\alpha[x_\alpha(t),t] \equiv [G^T(t,\theta) \vdots \emptyset]^T$$

and

$$h_\alpha[x_\alpha(t),t] \equiv H(t,\theta)\cdot x(t)$$

The desired state estimate is then given by $\hat{x}(t/t,t_\emptyset) = [I:\emptyset]\cdot\hat{x}_\alpha(t/t,t_\emptyset)$.

To obtain the state estimate we have to use a non-linear filter like Extended Kalman Filter [1], or utilize the partitioning approach, introduced by Lainiotis [2,3].

In this paper, two partitioned algorithms are presented and studied, the Adaptive Lainiotis Filter and the Linear Lainiotis Per-Step Partitioning Filtering Algorithm, in comparison with the commonly used Extended Kalman Filter. The two filters ALF and EKF are applied on the target tracking and manoeuver detection problem, where the time constraints of a real-time application are tight. The use of parallel processors to implement the partitioned algorithms has been considered in order to show the merits of the partitioning approach as the recently developed parallel machines make it possible to work faster for a parallel algorithm with high computational burden than for a less complicated sequential algorithm.

II. PARTITIONED ALGORITHMS

A) Adaptive Lainiotis Filter (ALF). The ALF is presented explicitly in [2,3]. This estimator is decomposable into two parts. One linear part consisting of linear filters matched to each admissible value of the unknown parameter θ, and a non-linear part, consisting of likelihood ratios, that incorporates the adaptive, learning or system identifying nature of the estimator. The equations of the filter follow:

$$\hat{x}(k/k) = \int \hat{x}(k/k,\theta)\cdot P(\theta/k)\,d\theta \tag{5}$$

$$p(\theta/k) = \frac{|P_z(k/k-1,\theta)|^{-1/2}\exp[-\frac{1}{2}\|\tilde{z}(k/k-1,\theta)\|^2_{P_z^{-1}(k/k-1,\theta)}]}{\int p(\theta/k-1)\,|P_z(k/k-1,\theta)|^{-1/2}\exp[-\frac{1}{2}\|\tilde{z}(k/k-1,\theta)\|^2_{P_z^{-1}(k/k-1,\theta)}]\,d\theta}\cdot p(\theta/k-1) \tag{6}$$

where $$\tilde{z}(k/k-1,\theta) \equiv z(k) - H(k,\theta)\cdot \hat{x}(k/k,\theta) \tag{6a}$$

and $$P_z(k/k-1,\theta) = H(k,\theta)P(k/k,\theta)H^T(k,\theta) + R(k) \tag{6b}$$

The parameter-conditional estimate $\hat{x}(k/k-1,\theta)$ and the corresponding error-covariance matrix $P(k/k-1,\theta)$ are given by the linear filter (e.g. Kalman) matched to the system with parameter value θ.
The linear filters that constitute the nonadaptive part of the ALF can be Kalman filters, as well as the Linear Lainiotis Per-Step Partitioning Filtering Algorithms depending on the application. The two algorithms, the ALF with Kalman filters (ALF(k)), and the ALF with the linear Lainiotis algorithms (ALF(L)), have the same performance but as we will see different computational requirements.

B) Lainiotis Per-Step Partitioning Filtering Algorithm (LPSPFA). The LPSPFA is a realization of the optimal MSE estimator different than the Kalman filter realization with theoretically the same performance as the Kalman filter. However, the actual performance need not be the same.
The Lainiotis filtering equations follow (time-invariant case):

$$\hat{x}(k+1/k+1) = \hat{x}_n(k+1/k+1) + \Phi_n P(k/k+1)\left[M_n(k+1) + P^{-1}(k,0)\hat{x}(k/k)\right] \tag{7}$$

$$P(k+1,0) = P_n + \Phi_n P(k/k+1)\Phi_n^T \tag{8}$$

$$P(k/k+1) = \left[P(k,0)O_n + I\right]^{-1}\cdot P(k,0) \tag{9}$$

and $$O_n = \Phi^T hAh^T\Phi \tag{10}$$

$$P_n = \left[I-K_nh^T\right]\gamma q\gamma^T \tag{11}$$

$$\Phi_n = \left[I-K_nh^T\right]\Phi \tag{12}$$

$$K_n = \gamma q\gamma^T hA \tag{13}$$

$$A = \left[h^T\gamma q\gamma^T h + R\right]^{-1} \tag{14}$$

$$K_m = \Phi^T hA \tag{15}$$

$$M_n(k+1) = K_m z(k+1) \tag{16}$$

$$\hat{x}_n(k+1/k+1) = K_n z(k+1) \tag{17}$$

where $P(k,\emptyset)$ corresponds to $P(k/k)$ of the Kalman filter.
As one can see, equations (10) through (15) can be calculated only once at the beginning, and so the computational burden of the filter per iteration is reduced. Of course, this also means a need of more memory for storing the calculated quantities.

In the following chapters we will make a comparative analysis of two forms of the ALF (one using Kalman filters - ALF(k), and the other using LPSPFA - ALF(L)) and the Extended Kalman Filter (EKF).

III. COMPUTATIONAL ANALYSIS

Apart from the performance of the algorithms it is of great interest to analyse their computational requirements. This includes both the storage requirements and the operations involved.
The parameters of the general analysis are the dimension n of the state vector, the dimension m of the measurement vector, the number of linear filters s in the adaptive filter, and the dimension p of the input vector $u(\cdot)$ if it is not equal to n. The results of this analysis can be found in general forms, in Table I for the storage requirements and in Table II for the operations involved. The analysis is based on the assumptions given in [4] and also used in [5].
As we can see in Figures 1 and 2, the use of Kalman filter or Lainiotis linear filter (LPSPFA) affects the computational characteristics of the Adaptive Lainiotis Filter. The choice of the linear filter depends on the specific application and its requirements.

Parallelism: If we consider the use of a parallel machine for the implementation of the ALF, the computational characteristics of the filter are changing. This filter is naturally structured for parallel processing, so even if the total computational burden is the same, the filter's time consumption is dependent only on the longest sequentially processed part of the filter. This can be as long as a Kalman filter. We have then new general forms for the case of parallel processing in Table III for the operations involved. From the part of storage requirements the Table I equations are still valid. The difference between using or not parallel processing for the ALF is concentrated on the time requirements of the filter and can be seen clearly in Figure 3.

The detailed analysis of the EKF and ALF algorithms is presented in Appendix A.

IV. MODEL EQUATIONS

The target tracking problem is still challenging system engineers. The manoeuver detection is the most important part of the trajectory estimation problem. Up to now, two approaches to this problem are providing more accurate state estimates [6]. The "State Augmentation" and the "Multiple Model Estimator". The first method produces a non-linear model and uses an EKF to estimate the augmented state vector. The second method uses an ALF with two linear filters matched to a manoeuvering and to a non-manoeuvering dynamic model, respectively. This method has the advantages of the first method and of other approaches but at a cost of a much larger computational burden. If the processing is sequential, this computational complexity is a disadvantage for the ALF due to the time limits of a real-time application. But the ALF is suitable for parallel processing and if we use this merit, it works even faster than the EKF.
The model we will use is presented by Singer [7] and most of the researchers used it in a more or less simplified form.
The equations of the model are:

$$x(k+1) = \Phi(T,\alpha)\cdot x(k) + u(k) \tag{18}$$

$$z(k) = H\cdot x(k) + v(k) \tag{19}$$

where $H = [100]$, $v(k)$ and $u(k)$ have variances σ_R^2 and $Q(k)$,

$$\Phi(T,x) = \begin{bmatrix} 1 & T & \frac{1}{\alpha^2}[-1+\alpha T + e^{-\alpha T}] \\ 0 & 1 & \frac{1}{\alpha}[1 - e^{-\alpha T}] \\ 0 & 0 & e^{-\alpha T} \end{bmatrix}$$

and

$$Q(k) = 2\alpha\sigma_m^2 \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix}$$

The sampling rate T of the radar is small enough (e.g. $T = 0.1$), and the state vector is

$$x(k) = \begin{bmatrix} p \\ \dot{p} \\ \ddot{p} \end{bmatrix} \begin{matrix} \text{target position at time k} \\ \text{target speed at time k} \\ \text{target acceleration at time k} \end{matrix}$$

There are two parameters in the mode, α and σ_m. The parameter a is the reciprocal of the manoeuver time constant. For example, $\alpha \cong 1/60$ for a lazy term and $\cong 1$ for atmospheric turbulence. The parameter σ_m^2 is the variance of the target acceleration and represents the type of the target we are tracking.

To apply the ALF, the model is ready and we only have to choose the parameter values for each linear filter but, for the application of the EKF we have to augment the state vector with the parameters, in order to estimate their values. So the new state vector is $x_\alpha(k) = [p(k)\dot{p}(k)\ddot{p}(k)\alpha(k)\sigma_m(k)]^T$ and the model equations are:

$$x_\alpha(k+1) = \varphi_\alpha(T, x_\alpha(k)) + u_\alpha(k) \tag{20}$$

$$z(k) = h_\alpha \cdot x_\alpha(k) + v(k) \tag{21}$$

where $h_\alpha = [10000]$, the variance of u_α is

$$q(k) = \left[\begin{array}{c|cc} Q(k) & & 0 \\ \hline & q_\alpha & 0 \\ 0 & 0 & q_{\sigma_m} \end{array}\right]$$

and the transition matrix is

$$\varphi_a(T, x_\alpha(k)) = \begin{bmatrix} p(k) + \dot{p}(k)\cdot T + \ddot{p}(k)\,(-1 + a(k)\cdot T + e^{-a(k)\cdot T})/a^2(k) \\ \dot{p}(k) + \ddot{p}(k)\cdot(1 - e^{-a(k)\cdot T})/a(k) \\ \ddot{p}(k)\cdot e^{-a(k)\cdot T} \\ a(k) \\ \sigma_m^2(k) \end{bmatrix}$$

Now the dimensionality of the problem is increased, and also the computational burden.

V. SIMULATION RESULTS

An example is presented, illustrating how the two non-linear filters are estimating the target state ang identify the parameter a. For simplicity we present one cartesian coordinate, but the results are similar for the others also. For the scenario considered, the target moves at 900 ft/sec at a distance of 20,000 ft, and suddenly accelerates. The radar data rate is 10 samples per second and the sensor noise is $\sigma_R = 200$ ft. The average time constant of the manoeuver class used in the scenario is 10 seconds ($a = 0.1$). The ALF has three linear filters ($s = 3$) and the second filter matches the target characteristics. The EKF has a state dimensionality of 4 instead of 3, or 5 (depending on how many parameters we identify, one (α) or two (α,σ_m)). The normalized errors (%) of the two filters are presented in Figures 4 to 7 for the position, velocity, acceleration error and the identification of parameter a error. Also, the a posteriori probability density functions of the three linear filters of ALF and the acceleration of the model are shown in Figures 8 and 9. All the results are obtained after 100 Monte Carlo runs. In Table IV the Mean Square Error is presented for ALF and EKF for different stages of the example, to show the different behaviour of the two filters.

The computational requirements of the filters for this example are presented in Table V. In a more general case, when we use more sensors, the dimension of the measurement vector m is augmented. In Figures 10 to 12 the operational requirements of the filters ALF and EKF are presented for different values of m.

VI. CONCLUSIONS

In this paper the Adaptive Lainiotis Filter is studied theoretically and through computer simulations. Two forms of this algorithm, one using Kalman filters and a second with Lainiotis Per-Step Partitioned Filtering Algorithms are compared with the Extended Kalman Filter. As shown, the computational complexity of the ALF is greater, but in identification problems may work faster than the EKF.

The advantages of ALF in convergence and estimation errors are clearly presented via simulations. Considering also the parallel machines that are lately available, it makes clear that the ALF is not only more accurate but also more time-efficient than the non-linear EKF, especially in multisensor and parameter identification problems.

TABLE I

Filter	Operation
EKF	$0.5n^2 + 2.5n + 0.5m^2 + 1.5m + 0.5P^2 + 0.5P + 2\max(n^2, nm) + \max\left(\frac{n^2+n}{2}, \frac{m^2+m}{2}\right)$
ALF(k)	$0.5m^2 + 2.5m + n + s\cdot(2 + 0.5n^2 + 1.5n) + s\cdot(2n^2 + 2n + nm + 0.5m^2 + 1.5m) +$ $+ 0.5m^2 + 2nm + 0.5m + \max\left(\frac{m^2+m}{2}, n^2\right)$
ALF(L)	$0.5m^2 + 2.5m + n + s\cdot(2 + 0.5n^2 + 1.5n) + s\cdot(3.5n^2 + 2.5n + 3nm + 0.5m^2 + 1.5m) +$ $+ n^2 + n + 0.5m^2 + 1.5m$

TABLE II

EKF	MULTS	$1.5n^3 + 0.5n^2 + 0.5m^3 + 1.5m^2 - m + 1.5n^2m + 1.5nm^2 + 2nm + np^2 +$ $+ 0.5n^2p + 0.5np$
	ADDS	$1.5n^3 - n^2 - 0.5n + 0.5m^3 + 0.5m + 1.5n^2m + 1.5nm^2 + np^2 + 0.5n^2p$ $- 0.5np$
	DIVS,SR	$2m-1, m$
ALF(k)	MULTS	$s\cdot(0.5m^3 + 2.5m^2 + n + 3 + 1.5n^3 + 1.5n^2 + 0.5m^3 + 1.5m^2 - m +$ $+ 1.5n^2m + 1.5nm^2 + 3nm)$
	ADDS	$s\cdot(1.5m^3 + m^2 - 0.5m + n + 1.5n^3 + 0.5n^2 - n + 0.5m^3 - 0.5m +$ $+ 1.5n^2m + 1.5nm^2 + nm) - n-1$
	DIVS,SR	$s\cdot(4m+1)$, $s\cdot(2m+1)$
ALF(L)	MULTS	$s\cdot(0.5m^3 + 2.5m^2 + n + 3 + 2.5n^3 + 5.5n^2 - 2n + 2nm)$
	ADDS	$s\cdot(1.5m^3 + m^2 - 0.5m + n + 2.5n^3 + 2n^2 - 2.5n + 2nm) - n-1$
	DIVS,SR	$s\cdot(2m+4n)$, $s\cdot(2n+m+1)$

TABLE III

ALF(k)	max{A, Kalman}	
ALF(L)	max{A, LPSPFA}	

A	MULTS	$s(0.5m^3 + 2.5m^2 + n + 3)$
	ADDS	$s(1.5m^3 + m^2 - 0.5m + n) - n-1$
	DIVS,SR	$s(2m+2)$, $s(m+1)$
KALMAN	MULTS	$1.5n^3 + 1.5n^2 + 0.5m^3 + 1.5m^2 - m + 1.5n^2m + 1.5nm^2 + 3nm$
	ADDS	$1.5n^3 + 0.5n^2 - n + 0.5m^3 - 0.5m + 1.5n^2m + 1.5nm^2 + nm$
	DIVS,SR	$2m-1$, m
LPSPFA	MULTS	$2.5n^3 + 5.5n^2 - 2n + 2nm$
	ADDS	$2.5n^3 + 2n^2 - 2.5n + 2nm$
	DIV,SR	$4n-2$, $2n$

TABLE IV

K (Time) / M.S.E. (Position	1-60	61-120	121-180	181-240	σ_R
A.L.F.	115	11	124	42	200
E.K.F.	62	76	96	95	

TABLE V

	EKF		Sequential		Parallel proc.	
	NE = 4	NE = 5	ALF(k)	ALF(L)	ALF(k)	ALF(L)
Memory req. (words)	63	93	142	203	142	203
Operations req.	778	1356	1607	2627	422	762

REFERENCES

[1] B.D.O. ANDERSON and J.B. MOORE, "Optimal Filtering", Prentice Hall Inc., Englewood Cliffs N.J., 1979.

[2] D.G. LAINIOTIS, "Optimal Adaptive Estimation: Structure and Parameter Adaptation", IEEE Trans. on A.C., Vol. AC-16, April 1971, pp.160-169.

[3] D.G. LAINIOTIS, "Partitioning: A Unifying Framework for Adaptive Systems, I: Estimation", IEEE Proceedings, Vol. 64, August 1976, pp.1126-1143.

[4] K.E. ANAGNOSTOY and D.G. LAINIOTIS, "Comparative Computational Analysis of a New Per-Sample Partitioning Linear Filter and of Kalman Filter", accepted for publication, in Int. J. of Syst. and Sciences, 1985.

[5] D.G. LAINIOTIS, S.D. LIKOTHANASIS and S.K. KATSIKAS, "Adaptive Control Algorithms: A Comparative Computational Analysis-Parallelism", 5th Int. Symp. on Modelling Identification and Control (MIC '86), Innsbruck, Austria, Feb. 1986.

[6] CHAW-BING CHANG and J.A. TABACZYNSKI, "Application of State Estimation to Target Tracking", IEEE Trans. on A.C., Vol. 29, February 1984, pp.98-109.

[7] R.A. SINGER, "Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets", IEEE Trans. on A.E.S., Vol. AES-6, July 1970, pp.473-483.

APPENDIX A

MEMORY AND OPERATIONS REQUIRED BY THE FILTERS

A) LINEAR FILTERS

The linear filter that we will use has been studied in detail in [4]. The gener formulas present the requirements of the filters for each step (iteration). Sma modifications have to be made, in order to make them suitable for the ALF.

Kalman filter: There is no change in the equations of this filter, but we need to store the quantity $P_z(k/k-1)$ for later use by the ALF. This augments the memory required by: $(m^2+m)/2$ memory words.

LPSPF algorithm: The two equations (6a) and (6b) are added to the filter equations in order to calculate the quantities $P_z(k/k-1)$ and $\tilde{z}(k/k-1)$. The memory required augments by: $n^2 + nm + m^2 + 2m$ memory words, and the operations by:

$$n^2 + nm + n^2m + nm^2/2 + nm/2 \quad \text{MULTS}$$

$$n^2 - n + n^2m + nm^2/2 + nm/2 \quad \text{ADDS}$$

B) NON-LINEAR FILTERS

1. EKF: The equations of Extended Kalman Filter [1] follow:

$$P(k/k-1) = F\cdot P(k-1/k-1)\cdot F^T + G(k-1)\cdot Q\cdot G^T(k-1)$$

$$K(k) = P(k-1)\cdot H^T\cdot (H\cdot P(k/k-1)\cdot H^T+R)^{-1}$$

$$P(k/k) = P(k/k-1) - K(k)\cdot H\cdot P(k/k-1)$$

$$\tilde{z}(k/k-1) = z(k) - h(k,f(k,\hat{x}(k-1/k-1)))$$

$$\hat{x}(k/k) = f(k,\hat{x}(k-1/k-1)) + K(k)\cdot\tilde{z}(k/k-1)$$

The parameters involved are: n the $\hat{x}$ dimension, m the z dimension and P the Q dimension. The calculation of the operations of the filter can be seen at Table A1. The memory required to store the matrices and the results is:

$$0.5n^2 + 2.5n + 0.5m^2 + 1.5m + 0.5(P^2+P) + 2\max(n^z,nm) + \max\left(\frac{n^2+n}{2},\frac{m^2+m}{2}\right) \text{ memory words.}$$

2. ALF: The equations of Adaptive Lainiotis Filter are presented in Chapter II (5) and (6). In the ALF the parameter s represents the number of the linear filters that we have in the bank of filters. To calculate the requirements of ALF, one has to calculate the requirements for the equations (5) and (6), and then add the requirements of the s linear filters.
The operations required for (5) and (6) are presented in detail in Table A2. The memory required to store the matrices and the results is:

$$0.5m^2 + 2.5m + n + s\left(2 + n + \frac{n^2+n}{2}\right)$$

The memory added by the s linear filters is:

s·(constant matrices storage) + temporary result storage

The results are in Table I.

The operations of the ALF are:

MULTS	$s(0.5m^3 + 2.5m^2 + n + 3)$ + s· (filter's MULTS)
ADDS	$s(1.5m^3 + m^2 - 0.5m + n) - n - 1 + s$ (Filter's ADDS)
DIVS	$s(2m + 2)$ + s (Filter's DIVS)
S.R.	$s(m+1)$ + s (Filter's square roots)
EXP	s

For the sequential case the results are in Table II. The calculation of the total burden has been done under the assumption [4] that:

1 multiplication = 4 additions
1 division = 6 additions
1 square root = 25 additions

Parallel implementation: When we use parallel processing to implement the ALF we need s+1 processors. One for the equations (5) and (6) and one for each of the s linear filters.

The filter's time consumption depends on the computational burden of one of the s+1 processes and, of course, of the longest one.
To find the computational burden that affects the characteristics of the ALF we have to calculate the maximum between the linear filter burden and the ALF equations (5) and (6) burden. This is clearly presented in Table III.
The memory consumption does not change with the parallel processing.

TABLE A1 - EKF OPERATIONS

Variable	Operation	MULTS	ADDS	DIVS	SI
	$G(k-1)\cdot Q$	$n\cdot p^2$	np^2-np		
	$G(k-1)\cdot Q\cdot G^T(k-1)$	$(n^2p+np)/2$	$(n^2p+np-n^2-n)/2$		
	$\Phi\cdot P(k-1/k-1)$	n^3	n^3-n^2		
	$\Phi\cdot P(k-1/k-1)\cdot\Phi^T$	$(n^3+n^2)/2$	$(n^3-n)/2$		
$P(k/k-1)$	$\Phi P\Phi^T + GQG^T$		$(n^2+n)/2$		
$\tilde{z}(k/k-1)$	$z-h(k,\varphi(k,\hat{x}(k-1/k-1)))$		m		
$P_1(k)$	$H\cdot P(k/k-1)$	n^2m	n^2m-nm		
	$H\cdot P(k/k-1)H^T$	$(nm^2+nm)/2$	$(nm^2+nm-m^2-m)/2$		
	$H\cdot P(k/k-1)\cdot H^T + R$		$(m^2+m)/2$		
	$(HPH^T+R)^{-1}$	$(m^3+3m^2-2m)/2$	$(m^3-m)/2$	$2m-1$	
$K(k)$	$P_1(k)\cdot(HPH^T+R)^{-1}$	nm^2	nm^2-nm		
	$K(k)\cdot\tilde{z}(k/k-1)$	nm	$nm-n$		
$\hat{x}(k/k)$	$\varphi(k\hat{x}(k-1/k-1)+K(k)\cdot\tilde{z}(k/k-1)$		n		
	$K(k)\cdot P_1(k)$	$(n^2m+nm)/2$	$(n^2m+nm-n^2-n)/2$		
$P(k/k)$	$P(k/k-1)-K(k)\cdot P_1(k)$		$(n^2+n)/2$		

TABLE A2 - ALF OPERATIONS

Variable	Operation	MULTS	ADDS	DIVS	S
	$P_{\tilde{z}}(k/k-1,\theta)^{-1}$	$s\cdot(m^3+3m^2-2m)/2$	$s\cdot(m^3-m)/2$	$s\cdot(2m-1)$	s
	$\lVert P_{\tilde{z}}(k/k-1,\theta)\rVert^{-1/2}$			s	s
	$\tilde{z}(k/k-1,\theta)^T\cdot P_z(k/k-1,\theta)^{-1}$	$s\cdot m^2$	$s(m^2-m)$		
	$\tilde{z}(k/k-1,\theta)^T\cdot P_z^{-1}\cdot\tilde{z}^T(k/k-1,\theta)$	$s\cdot m$	$s(m-1)$		
$\ell(k,\theta)$	$\lvert P_z\rvert^{-1/2}\cdot\exp(-\tilde{z}^2\cdot P_z^{-1}/2)$ *)	s		s	
	$\sum_{i=1}^{s} P(\theta_i/k-1)\cdot\ell(k,\theta_i)$	s	$s-1$		
$P(\theta_i/k)$	$\ell(k,\theta_i)\cdot P(\theta_i/k-1)/\sum_{i=1}^{s} p\cdot\ell$	s		s	
$\hat{x}(k/k)$	$\sum_{i=1}^{s}\hat{x}(k/k,\theta_i)\cdot P(\theta_i/k)$	$s\cdot n$	$n(s-1)$		

*) s exp functions

FIGURE 1
MEMORY REQUIRED by A.L.F.
Using KALMAN & LPSPFA Linear Filters

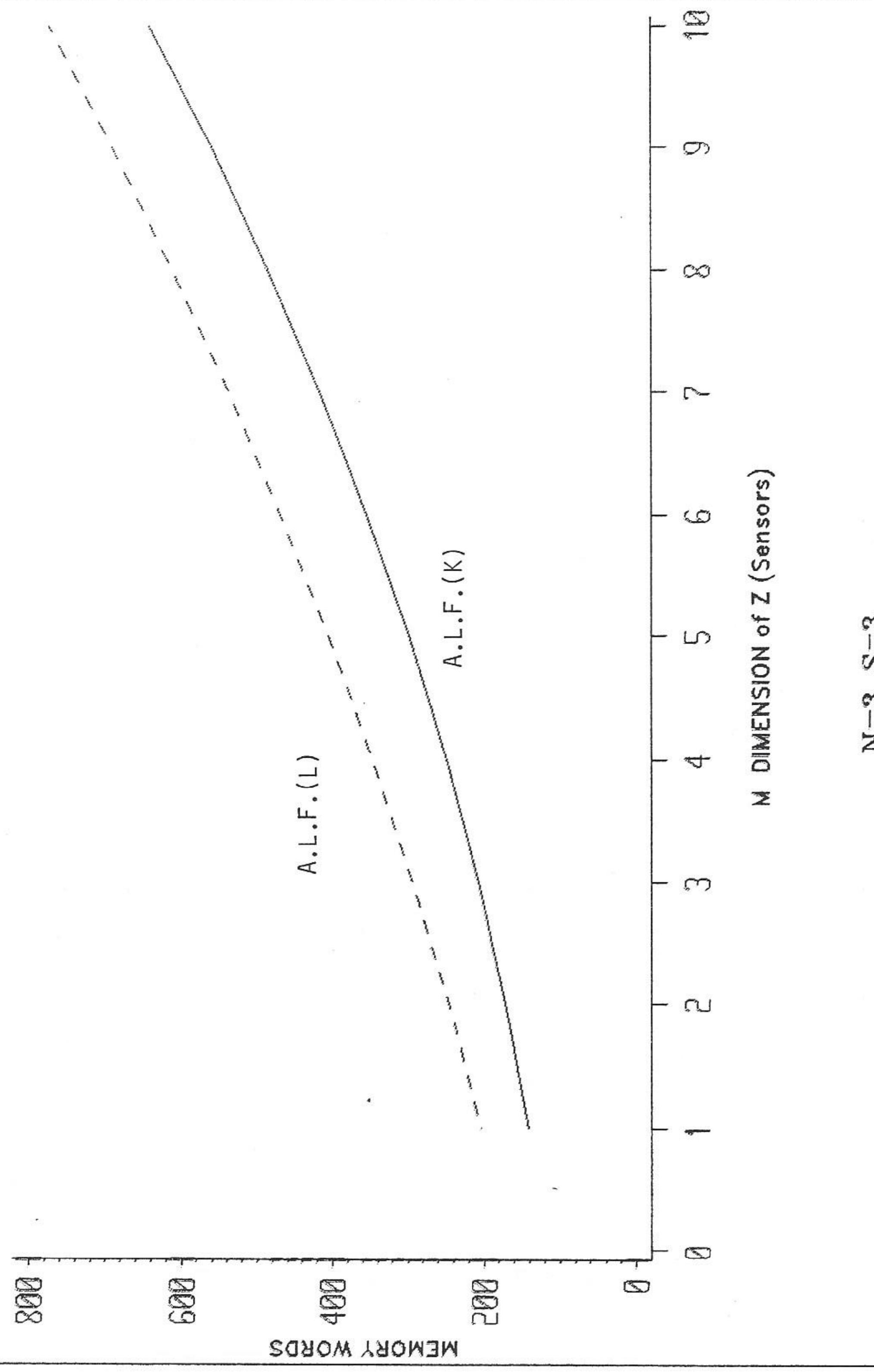
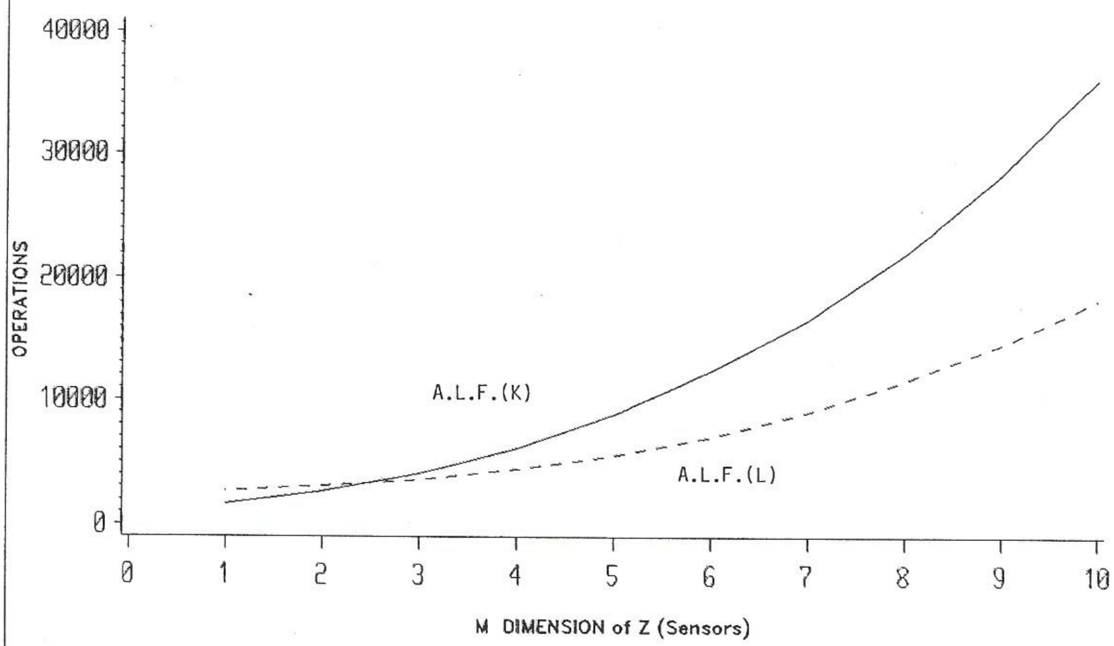
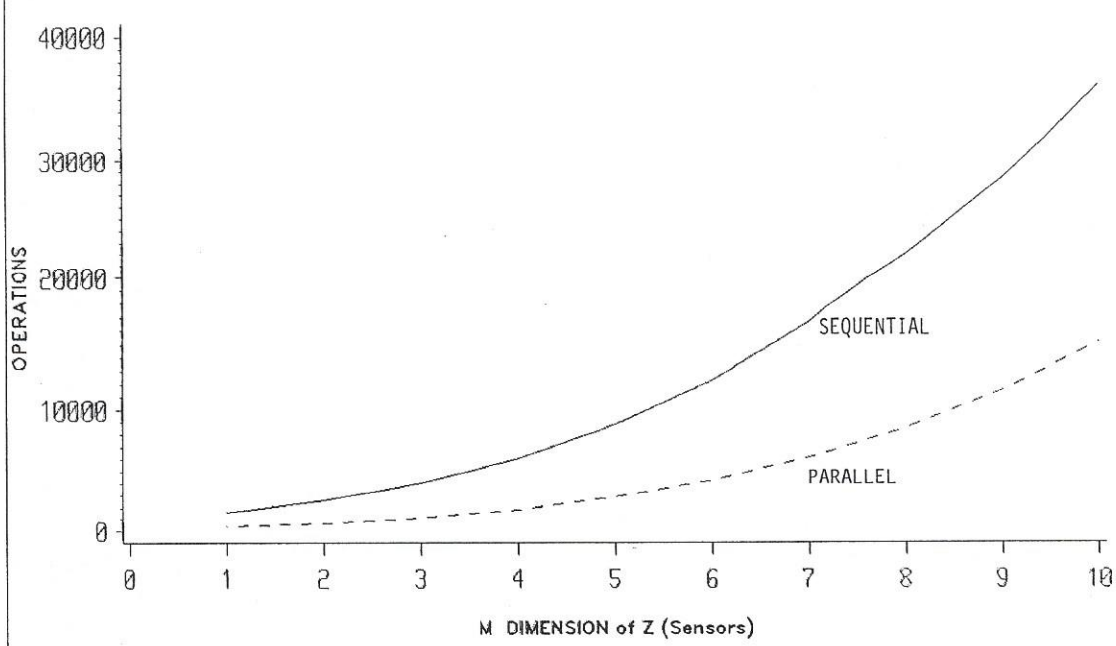


FIGURE 2
OPERATIONS REQUIRED by A.L.F.
Using KALMAN & LPSPFA Linear Filters



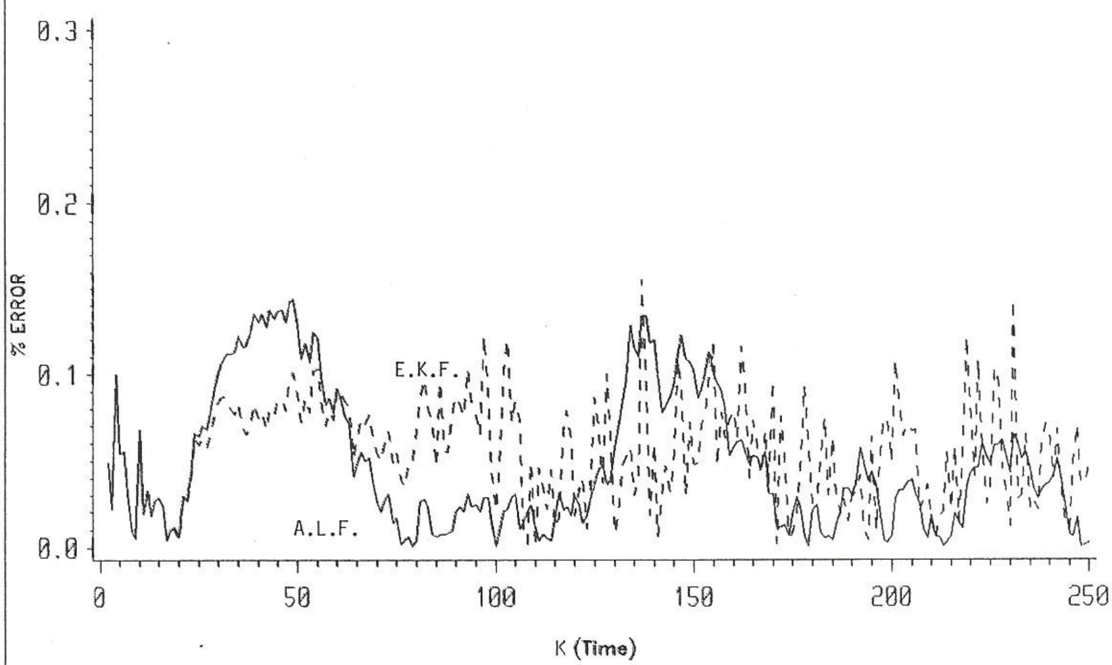
N=3 S=3

FIGURE 3
OPERATIONS REQUIRED by A.L.F.(K)
Using SEQUENTIAL & PARALLEL Processing



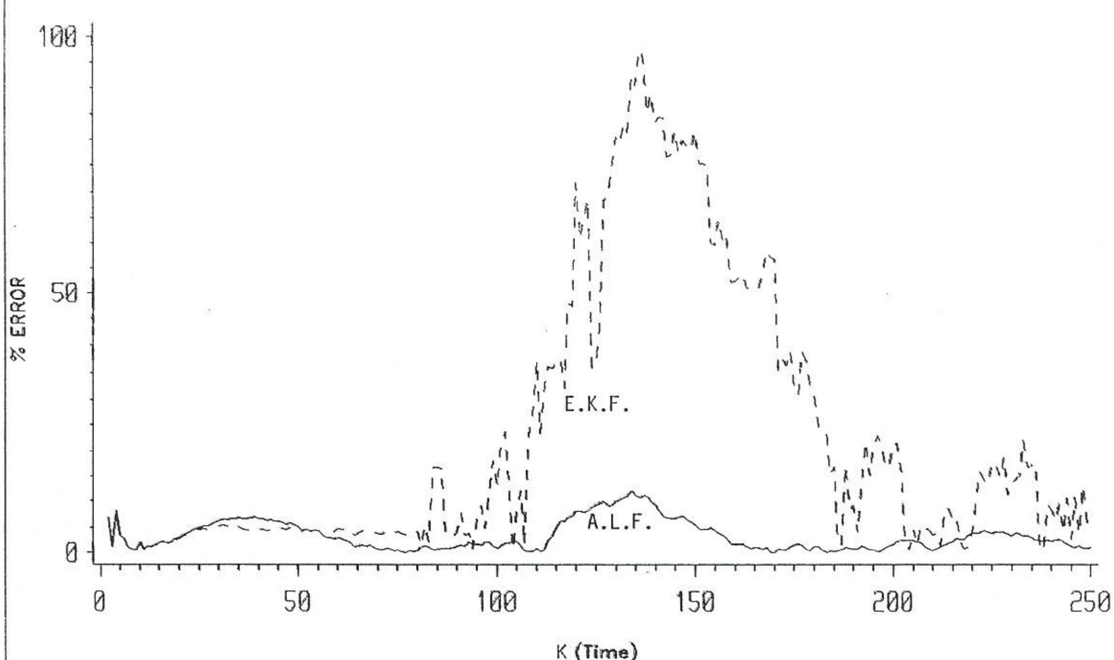
N=3 S=3

FIGURE 4
NORMALISED POSITION ERRORS of A.L.F & E.K.F. vs TIME



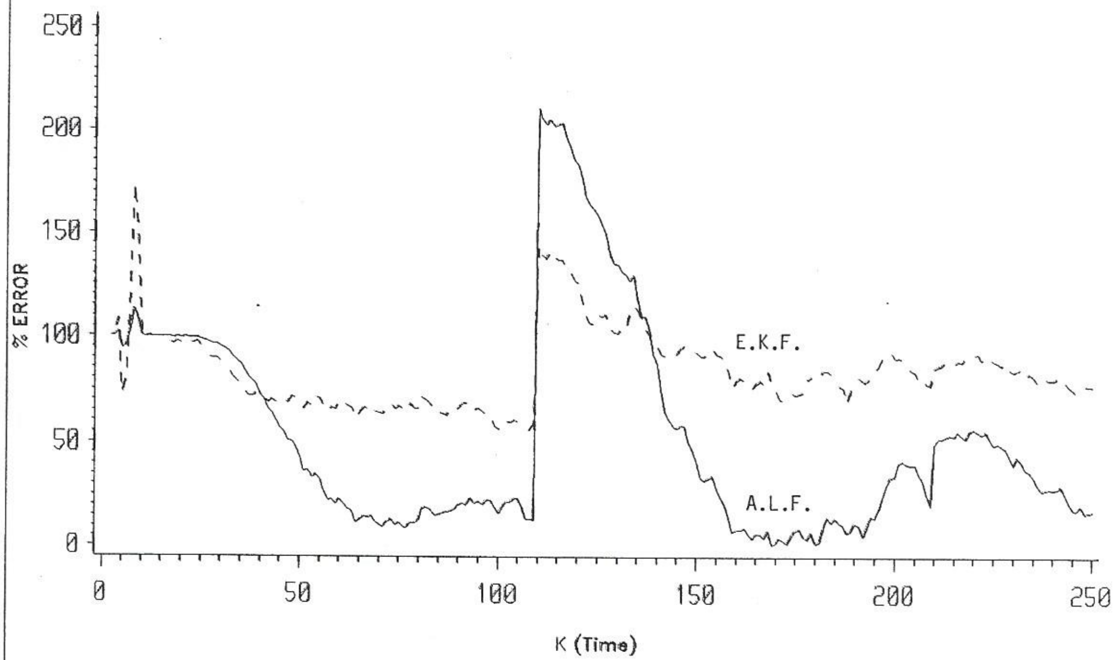
Monte–Carlo runs=100 , R=40000

FIGURE 5
NORMALISED VELOCITY ERRORS of A.L.F & E.K.F. vs TIME



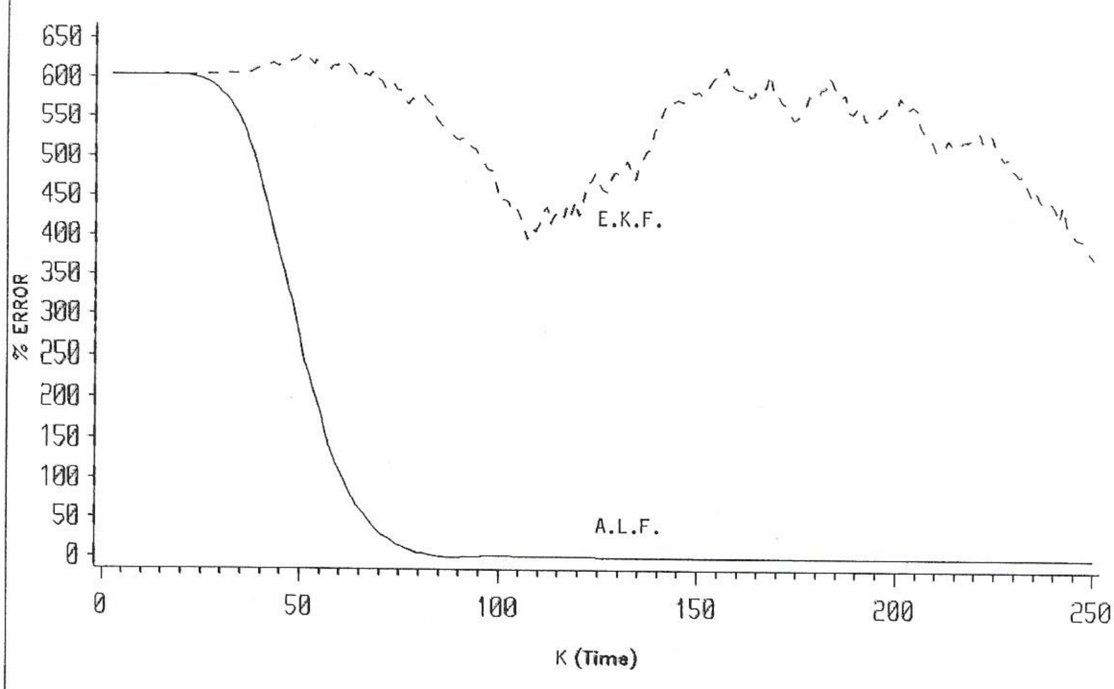
Monte–Carlo runs=100 , R=40000

FIGURE 6
NORMALISED ACCELERATION ERRORS of A.L.F & E.K.F. vs TIME



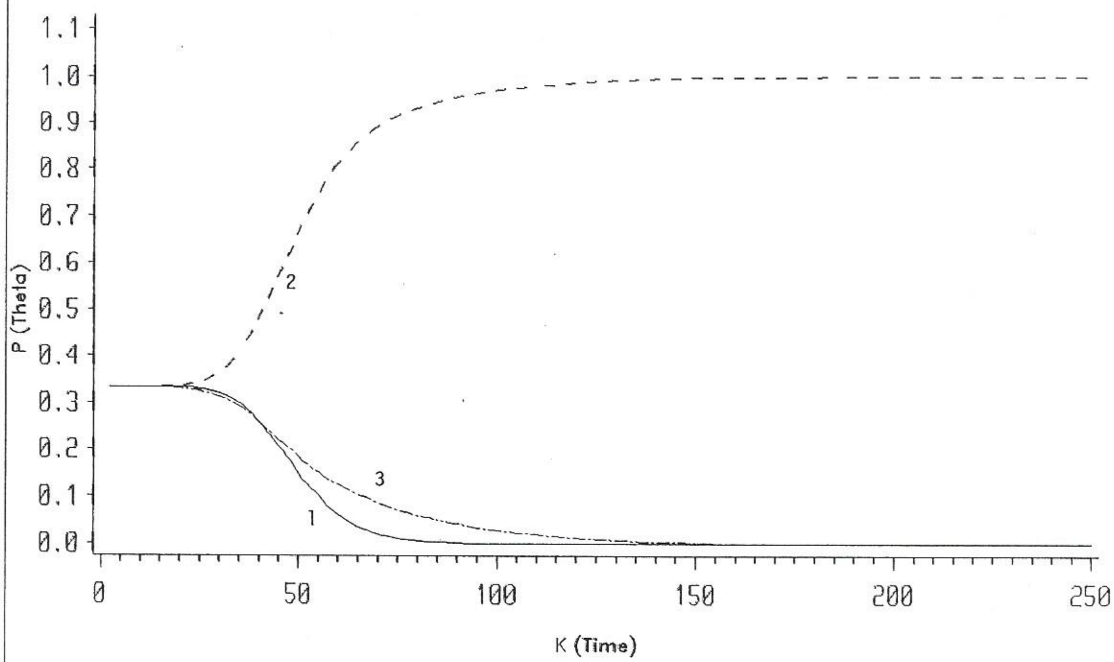
Monte-Carlo runs=100 , R=40000

FIGURE 7
PARAMETER A IDENTIFICATION ERROR vs TIME



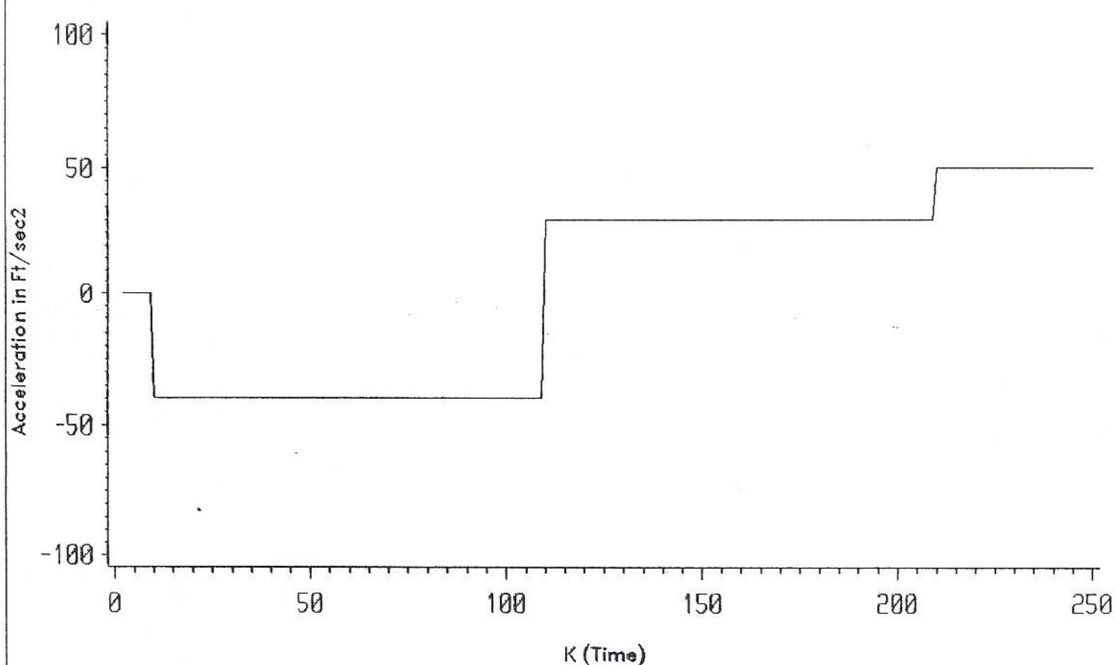
Monte-Carlo runs=100 , R=40000

FIGURE 8
A POSTERIORI PROBABILITY DENSITY FUNCTIONS of A.L.F.



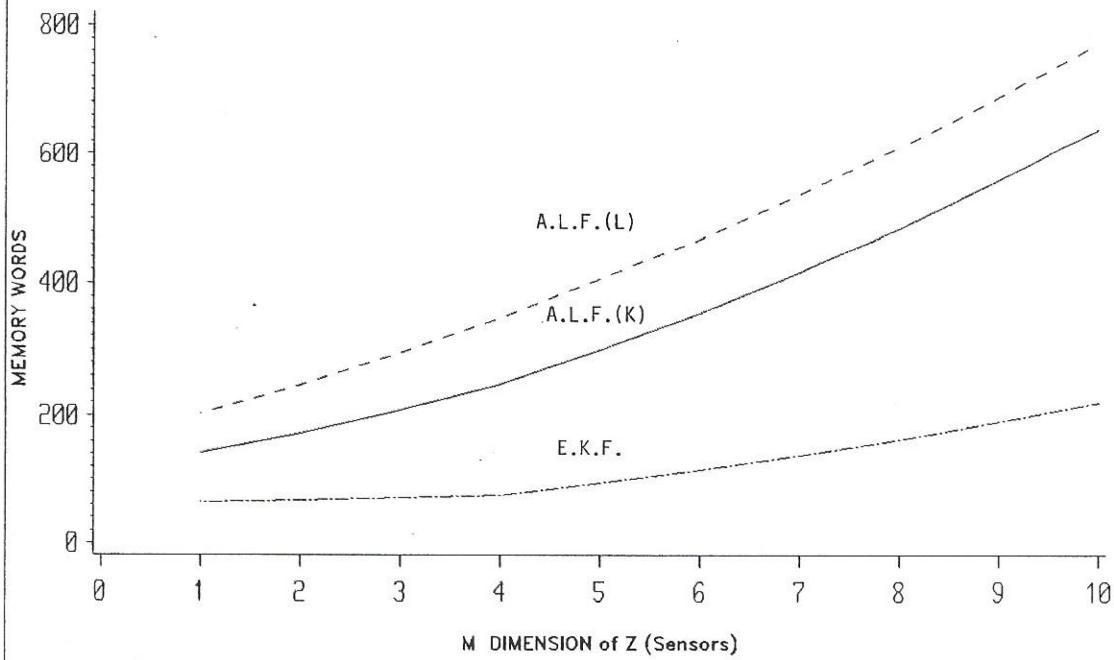
Monte-Carlo runs=100 , R=40000

FIGURE 9
ACCELERATION of the MODEL vs TIME



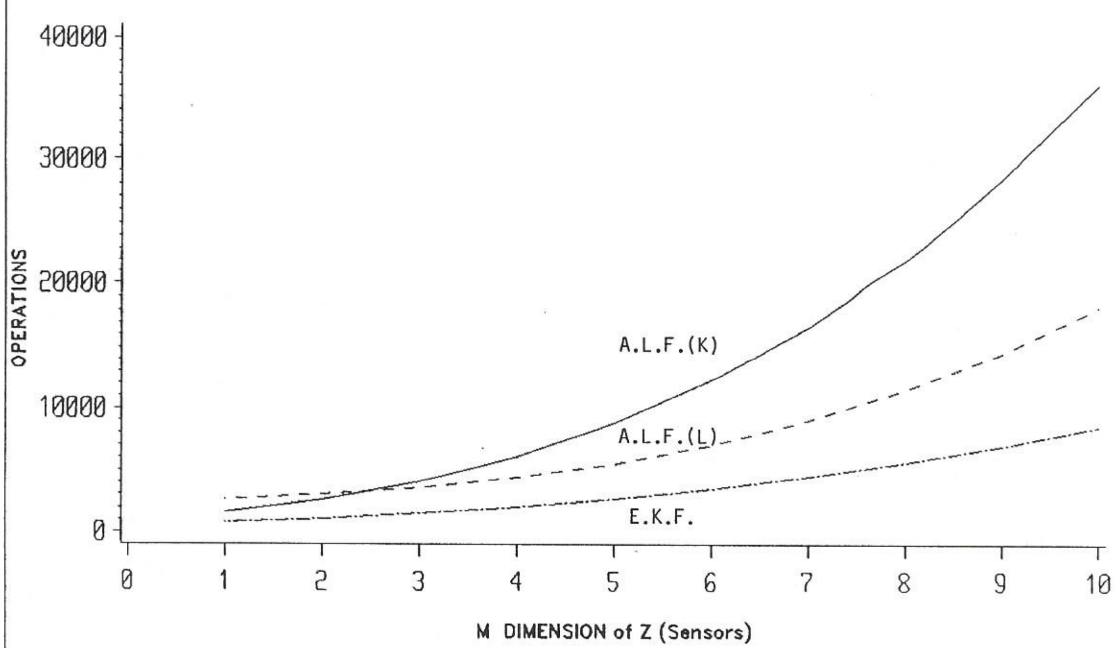
Monte-Carlo runs=100 , R=40000

FIGURE 10
MEMORY REQUIRED by A.L.F. & E.K.F.



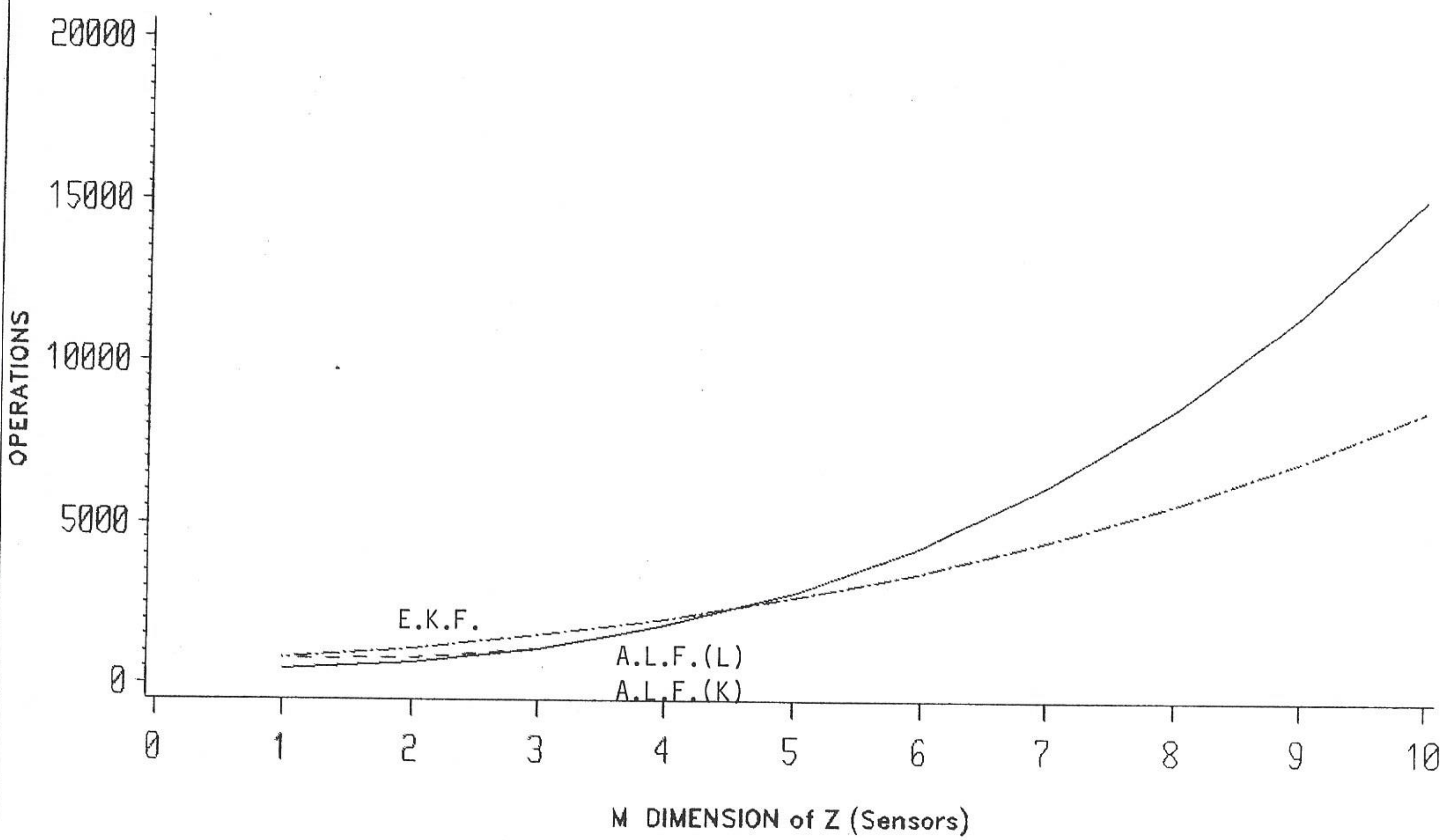
N=3 S=3 NE=4

FIGURE 11
OPERATIONS REQUIRED by A.L.F. & E.K.F.
Using SEQUENTIAL Processing



N=3 S=3 NE=4

FIGURE 12
OPERATIONS REQUIRED by A.L.F. & E.K.F.
Using PARALLEL Processing



N=3 S=3 NE=4