

## Speedup of Kalman and Lainiotis filters for partitionable models

Maria G. Koziri<sup>1</sup>, Thanasis Loukopoulos<sup>2\*</sup>, Maria Adam<sup>2</sup> and Nicholas Assimakis<sup>3</sup>

Department of Computer Science, University of Thessaly, Lamia, Greece<sup>1</sup>

Department of Computer Science and Biomedical Informatics, University of Thessaly, Lamia, Greece<sup>2</sup>

Department of Electronic Engineering, Technological Educational Institute of Central Greece, Lamia, Greece<sup>3</sup>

Received: 30-June-2016; Revised: 03-August-2016; Accepted: 05-August-2016  
©2016 ACCENTS

### Abstract

Both Kalman and Lainiotis filters arise in linear estimation and are associated with linear systems. In this paper, we investigate the partition of linear systems into lower dimensioned independent linear systems. We study the effects of partitioning on filter behaviour and computational burden. It is concluded that the partition does not affect the behaviour of Kalman and Lainiotis filters. Simulation results show that Kalman and Lainiotis filters compute the same outputs (estimation and estimation error covariance), both in the single and in the partitioned model. It is also concluded that the partition leads to significant speedup in both Kalman and Lainiotis filters. Specifically, it is shown that the partition into  $p$  independent models leads to super-linear speedup in the range  $(p, p^2)$  for both filters.

### Keywords

Linear systems, Kalman filter, Lainiotis filter, Partitioning.

## 1. Introduction

Estimation plays a very important role in many fields of science [1]. The estimation/filtering problem arises in linear estimation and is associated with linear systems, which can be time varying or time invariant. The discrete time Kalman filter [1-3] and the discrete time Lainiotis filter [3, 4] are the most popular algorithms that solve the filtering problem. Both filters have time varying as well as time invariant versions. In this paper, we investigate the partition of linear systems into lower dimensioned linear systems. We focus on the case where a linear system can be partitioned into lower dimensioned independent systems. Real time problems require accurate and fast computation in order to deal with more realistic models. The novelty of this paper is the derivation of accurate and fast filters when model partitioning is feasible, in which case it is shown that significant speedups can be achieved. The rest of the paper is organised as follows: Section 2 describes the system partition. Kalman and Lainiotis filters are briefly presented in Section 3. In Section 4 the effects of partitioning on filter behaviour are discussed, while Section 5 illustrates the achievable speedups. Finally, Section 6 concludes the paper.

## 2. System partition

The estimation/filtering problem arises in linear estimation and is associated with linear systems described by the following state space equations:

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

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

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,  $\{w(k)\}$  and  $\{v(k)\}$  are independent Gaussian zero-mean white and uncorrelated random processes,  $Q(k)$  and  $R(k)$  are the plant and measurement noise covariance matrices respectively,  $x(0)$  is a Gaussian random process with mean  $x_0$  and covariance  $P_0$ .

For multi-state and multi-sensor environment, both the state vector and the measurement vector can be partitioned into  $p$  parts of lower dimensions than the initial single vectors. Then partitionable models can be derived. The model partition is allowable when the derived models are independent. Then, the resulting  $p$  models have  $N$ -dimensional state vectors and  $M$ -dimensional measurement vectors, satisfying the relation:

\*Author for correspondence

$$p = \frac{n}{N} = \frac{m}{M} \quad (3)$$

Note that  $n$  and  $m$  are the partitionable model dimensions, while  $N$  and  $M$  are the partitioned model dimensions.

It is clear that when the models are independent, then the initial state vector can be recovered by merging the partitioned state vectors and the initial measurement vector can be recovered by merging the partitioned measurement vectors. Also the partitioned transition matrices are part of the initial transition matrix and the partitioned output matrices are part of the initial output matrix. This holds for the plant and measurement noise covariance matrices.

### 3. Kalman and Lainiotis filters

The filtering/estimation problem is to produce an estimate at the time  $k$  of the state vector using measurements prior to time  $k$ ; i.e. the aim is to use these measurements in order to calculate an estimate value  $x(k/k)$  of the state vector  $x(k)$  and the corresponding covariance matrix  $P(k/k)$ .

The discrete time Kalman Filter [1-3] and Lainiotis filter [3, 4] are the most well-known algorithms that solve the filtering problem. For time varying systems, the filtering algorithms of interest are summarized in the following:

#### Time varying Kalman filter (TVKF)

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

#### Time varying Lainiotis filter (TVLF)

$$\begin{aligned} x_n(k+1/k+1) &= K_n(k+1)z(k+1) \\ M_n(k+1) &= K_m(k+1)z(k+1) \\ x(k+1/k+1) &= x_n(k+1/k+1) + \\ &\quad + F_n(k+1,k)[I + P(k/k)O_n(k+1)]^{-1} \\ &\quad [P(k/k)M_n(k+1) + x(k/k)] \\ P(k+1/k+1) &= P_n(k+1,k) \\ &\quad + F_n(k+1,k)[I + P(k/k)O_n(k+1)]^{-1}P(k/k)F_n^T(k+1,k) \end{aligned}$$

where:

$$\begin{aligned} A(k+1) &= [H(k+1)Q(k)H^T(k+1) + R(k+1)]^{-1} \\ K_n(k+1) &= Q(k)H^T(k+1)A(k+1) \\ K_m(k+1) &= F^T(k+1,k)H^T(k+1)A(k+1) \\ P_n(k+1,k) &= Q(k) - K_n(k+1)H(k+1)Q(k) \\ F_n(k+1,k) &= F(k+1,k) - K_n(k+1)H(k+1)F(k+1,k) \\ O_n(k+1) &= K_m(k+1)H(k+1)F(k+1,k) \end{aligned}$$

For **time invariant systems** where the system transition matrix, the output matrix, the plant and measurement noise covariance matrices are constant, the resulting time invariant Kalman and Lainiotis Filters take the following form:

#### Time invariant Kalman filter (TIKF)

$$\begin{aligned} x(k+1/k) &= Fx(k/k) \\ P(k+1/k) &= Q + FP(k/k)F^T \\ K(k+1) &= P(k+1/k)H^T[HP(k+1/k)H^T + R]^{-1} \\ x(k+1/k+1) &= x(k+1/k) \\ &\quad + K(k+1)[z(k+1) - Hx(k+1/k)] \\ P(k+1/k+1) &= P(k+1/k) - K(k+1)HP(k+1/k) \end{aligned}$$

#### Time invariant Lainiotis filter (TILF)

$$\begin{aligned} x_n(k+1/k+1) &= K_n(k+1)z(k+1) \\ M_n(k+1) &= K_m z(k+1) \\ x(k+1/k+1) &= x_n(k+1/k+1) \\ &\quad + F_n[I + P(k/k)O_n]^{-1}[P(k/k)M_n(k+1) + x(k/k)] \\ P(k+1/k+1) &= P_n(k+1,k) \\ &\quad + F_n[I + P(k/k)O_n]^{-1}P(k/k)F_n^T \end{aligned}$$

where the following quantities are calculated off-line:

$$\begin{aligned} A &= [HQH^T + R]^{-1} \\ K_n &= QH^T A \\ K_m &= F^T H^T A \\ P_n &= Q - K_n H Q \\ F_n &= F - K_n H F \\ O_n &= K_m H F \end{aligned}$$

For time invariant systems, it is well known [1] that if the signal process model is asymptotically stable, i.e., all eigenvalues of the transition matrix  $F$  lie inside the Unit Circle, then there exist steady state values  $\bar{P}_p$  and  $\bar{P}_e$  of the prediction and estimation error variances matrices, respectively. In this case, the resulting discrete time steady state Kalman Filter and Lainiotis Filter take the following form:

#### Steady state Kalman filter (SSKF)

$$x(k+1/k+1) = A_{KF}x(k/k) + B_{KF}z(k+1)$$

where the matrices  $A_{KF} = [I - \bar{K}H]F$  and  $B_{KF} = \bar{K}$  are calculated off-line by first solving the corresponding discrete time Riccati equation:

$$P(k+1/k) = Q + FP(k/k-1)F^T - FP(k/k-1)H^T [HP(k+1/k)H^T + R]^{-1} HP(k/k-1)F^T$$

and then calculating the steady state gain:  $\bar{K} = \bar{P}_p H^T [H\bar{P}_p H^T + R]^{-1}$ .

#### Steady state Lainiotis filter (SSLF)

$$x(k+1/k+1) = A_{LF}x(k/k) + B_{LF}z(k+1)$$

where the matrices  $A_{LF} = F_n[I + \bar{P}_e O_n]^{-1} \bar{P}_e$  and  $B_{LF} = K_n + F_n[I + \bar{P}_e O_n]^{-1} \bar{P}_e K_m$  are calculated off-line by solving the corresponding discrete time Riccati equation

$$P(k+1/k+1) = P_n + F_n[I + P(k/k)O_n]^{-1} P(k/k)F_n^T$$

#### 4.Partitioned models and filter behaviour

It is obvious that both Kalman and Lainiotis filter can be implemented to the basic model (1)-(2) as well as to the partitioned models satisfying (3). The implementations concern time varying, time invariant or steady state filters. In all cases, the estimations  $x(k/k)$  of the state vector  $x(k)$  and the corresponding covariance matrix  $P(k/k)$  derived using the initial single system, are related to the estimations and the corresponding covariance matrices derived using the partitioned systems. In fact, we are able to reconstruct the initial single state vector estimate by merging the partitioned state vectors estimates. Also we are able to reconstruct the initial single covariance matrix putting the partitioned covariance matrices in a block-diagonal matrix. It is clear that that the partition does not affect the behaviour of any filter.

In order to confirm this conclusion we consider the model taken from [5] to track head movements in x-axis and y-axis simultaneously (single model): The state vector is of dimension  $n=2$  and contains the position in x-axis and y-axis:  $x(k) = [s_x(k) \quad s_y(k)]^T$

The measurement vector is of dimension  $m=2$  and contains the measured position x-axis and y-axis:

$$z(k) = [z_x(k) \quad z_y(k)]^T.$$

Then we take:

$$F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The plant noise  $w(k) = [w_x(k) \quad w_y(k)]^T$  is Gaussian zero-mean with covariance matrix  $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ .

The measurement noise  $v(k) = [v_x(k) \quad v_y(k)]^T$  is Gaussian zero-mean with covariance matrix  $R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$

This model allows us to partition the system into  $p=2$  independent systems, which describe the movement in x-axis and y-axis separately (partitioned models). In each axis, the state vector is of dimension  $N=2$  and contains the position:  $x(k) = s(k)$ . The measurement vector is of dimension  $M=1$  and contains the measured position  $z(k)$ .

Then we take:

$$F = 1, H = 1, Q = 1, R = 0.1.$$

The plant noise  $w(k)$  is Gaussian zero-mean with covariance matrix  $Q$  and the measurement noise  $v(k)$  is Gaussian zero-mean with covariance matrix  $R$ .

It is obvious that we are able to describe the movement in both axis using two separate states:  $x_x(k) = s_x(k)$  for the x-axis and  $x_y(k) = s_y(k)$  for the y-axis. If we merge these two states, we take the state vector  $x(k) = [s_x(k) \quad s_y(k)]^T$  of the single model. By merging the estimation  $x_x(k/k) = s_x(k/k)$  for the movement in x-axis and the estimation  $x_y(k/k) = s_y(k/k)$  for the movement in y-axis, we take the estimation of state vector of the single model:

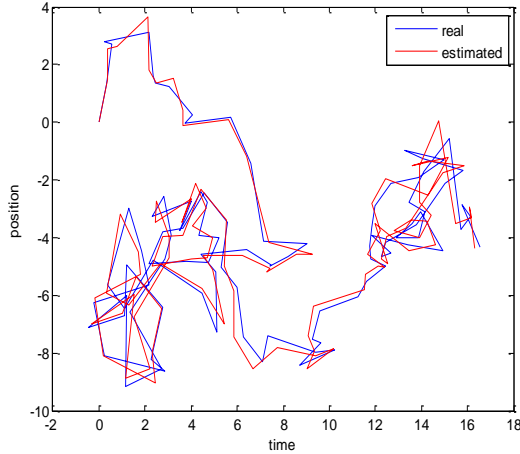
$$x(k/k) = [s_x(k/k) \quad s_y(k/k)]^T = [x_x(k/k) \quad x_y(k/k)]^T$$

Also, the estimation error covariance matrices  $P_x(k/k)$  and  $P_y(k/k)$  for each movement can be merged to the estimation error covariance of the single model:

$$P(k/k) = \begin{bmatrix} P_x(k/k) & 0 \\ 0 & P_y(k/k) \end{bmatrix}$$

We implemented the time invariant as well as the steady state Kalman and Lainiotis filters with initial conditions  $x_0 = 0$  and  $P_0 = 1$  for both axis movements.

Figure 1 depicts the real and estimated position using invariant Kalman and Lainiotis filters, where position corresponds to head movements in x-axis and y-axis simultaneously.



**Figure 1** Position estimation using Kalman and Lainiotis filters

Concerning the behaviour of the filters, we found that the time invariant Kalman and Lainiotis filters are equivalent, since they compute the same outputs (estimation and estimation error covariance), in both the single and the partitioned model. These results are depicted in Figure 1, where the estimated position concerns both Kalman and Lainiotis filters.

### 5.Speedup due to partition

In this section we investigate the partition effects on the computational burdens of the filters. Kalman and Lainiotis filters are iterative algorithms. Thus, their computational burdens depend on their per-iteration calculation burdens. For multi-state and multi-sensor environment, where  $n \geq 2$  and  $m \geq 2$ , the per-iteration calculation burdens, computed using the ideas in [4], are shown in Table 1.

When implementing the filters to  $p$  partitioned models with  $N \geq 1$  and  $M \geq 1$ , where  $p$  an integer is  $p \geq 2$ , the per-iteration calculation burdens, are shown in Table 2. It is worth remarking that, due to the fact that the partitioned models have the same structure, both Kalman and Lainiotis filters have also the same structure and can be implemented in parallel, resulting to  $p$  times faster filters.

For both Kalman and Lainiotis filters, and for all their versions (time varying, time invariant, steady state), we define the speedup from the single model

to the  $p$  partitioned models as the per iteration calculation burden of filters for a single model divided by the per-iteration calculation burden of filters for the  $p$  partitioned model. Then, we are able to determine lower and upper bounds for the speedup, as follows.

**Table 1** Per-iteration calculation burden of filters for the single model

Filter	Calculation burden
TVKF	$4n^3 + 3.5n^2 - 1.5n + 4n^2m + nm + 3nm^2 + \frac{1}{6}(16m^3 - 3m^2 - m)$
TVLF	$8n^2m + 3nm + 5nm^2 + \frac{1}{6}(58n^3 + 6n^2 - 10n) + \frac{1}{6}(16m^3 - 3m^2 - m)$
TIKF	$4n^3 + 3.5n^2 - 1.5n + 4n^2m + nm + 3nm^2 + \frac{1}{6}(16m^3 - 3m^2 - m)$
TILF	$4nm + \frac{1}{6}(58n^3 + 9n^2 - 7n)$
SSKF	$2n^2 + 2nm - n$
SSLF	$2n^2 + 2nm - n$

**Table 2** Per-iteration calculation burden of filters for partitioned models

Filter	Calculation burden
TVKF	$p[4N^3 + 3.5N^2 - 1.5N + 4N^2M + NM + 3NM^2 + \frac{1}{6}(16M^3 - 3M^2 - M)]$
TVLF	$p[8N^2M + 3NM + 5NM^2 + \frac{1}{6}(58N^3 + 6N^2 - 10N) + \frac{1}{6}(16M^3 - 3M^2 - M)]$
TIKF	$p[4N^3 + 3.5N^2 - 1.5N + 4N^2M + NM + 3NM^2 + \frac{1}{6}(16M^3 - 3M^2 - M)]$
TILF	$p[4NM + \frac{1}{6}(58N^3 + 9N^2 - 7N)]$
SSKF	$p[2N^2 + 2NM - N]$
SSLF	$p[2N^2 + 2NM - N]$

### Time Varying Filters

For the time varying Kalman filter, we have:

$$\begin{aligned}
 \text{speedup} &= \frac{4n^3 + 3.5n^2 - 1.5n + 4n^2m + nm + 3nm^2 + \frac{1}{6}(16m^3 - 3m^2 - m)}{p[4N^3 + 3.5N^2 - 1.5N + 4N^2M + NM + 3NM^2 + \frac{1}{6}(16M^3 - 3M^2 - M)]} \\
 &= \frac{4p^2N^3 + 3.5pN^2 - 1.5N + 4p^2N^2M + pNM + 3p^2NM^2 + \frac{1}{6}(16p^2M^3 - 3pM^2 - M)}{4N^3 + 3.5N^2 - 1.5N + 4N^2M + NM + 3NM^2 + \frac{1}{6}(16M^3 - 3M^2 - M)}
 \end{aligned}$$

Then

$$p < \text{speedup}$$

since:

$$\begin{aligned}
 & (4N^3 + 4N^2M + 3NM^2 + \frac{16}{6}M^3)p^2 \\
 & - (4N^3 + 4N^2M + 3NM^2 + \frac{16}{6}M^3 - 1.5N - \frac{1}{6}M)p \\
 & - (1.5N + \frac{1}{6}M) \\
 & = (4N^3 + 4N^2M + 3NM^2 + \frac{16}{6}M^3) \\
 & (p-1)(p + \frac{1.5N + \frac{1}{6}M}{4N^3 + 4N^2M + 3NM^2 + \frac{16}{6}M^3}) > 0
 \end{aligned}$$

and

$$speedup < p^2$$

since:

$$\begin{aligned}
 & (3.5N^2 - 1.5N + NM - \frac{1}{6}3M^2 - \frac{1}{6}M)p^2 \\
 & - (3.5N^2 + NM - \frac{1}{6}3M^2)p + (1.5N + \frac{1}{6}M) \\
 & = (3.5N^2 - 1.5N + NM - \frac{1}{6}3M^2 - \frac{1}{6}M) \\
 & (p-1)(p + \frac{1.5N + \frac{1}{6}M}{3.5N^2 - 1.5N + NM - \frac{1}{6}3M^2 - \frac{1}{6}M}) > 0
 \end{aligned}$$

For the time varying Lainiotis filter, we have:

$$\begin{aligned}
 & \frac{8n^2m + 3nm + 5nm^2}{p} \\
 & + \frac{1}{6}(58n^3 + 6n^2 - 10n) \\
 speedup = & \frac{1}{p} \frac{\frac{1}{6}(16m^3 - 3m^2 - m)}{8N^2M + 3NM + 5NM^2} \\
 & + \frac{1}{6}(58N^3 + 6N^2 - 10N) \\
 & + \frac{1}{6}(16M^3 - 3M^2 - M)
 \end{aligned}$$

$$\begin{aligned}
 & \frac{8p^2N^2M + 3pNM + 5p^2NM^2}{8N^2M + 3NM + 5NM^2} \\
 & + \frac{1}{6}(58p^2N^3 + 6pN^2 - 10N) \\
 & + \frac{1}{6}(16p^2M^3 - 3pM^2 - M) \\
 = & \frac{8p^2N^2M + 3pNM + 5p^2NM^2}{8N^2M + 3NM + 5NM^2} \\
 & + \frac{1}{6}(58N^3 + 6N^2 - 10N) \\
 & + \frac{1}{6}(16M^3 - 3M^2 - M)
 \end{aligned}$$

Then

$$p < speedup$$

since:

$$\begin{aligned}
 & (8N^2M + 5NM^2 + \frac{1}{6}58N^3 + \frac{1}{6}16M^3)p^2 \\
 & - (8N^2M + 5NM^2 + \frac{1}{6}58N^3 + \frac{1}{6}16M^3 - \frac{1}{6}10N - \frac{1}{6}M)p \\
 & - (\frac{1}{6}10N + \frac{1}{6}M) \\
 & = (8N^2M + 5NM^2 + \frac{1}{6}58N^3 + \frac{1}{6}16M^3) \\
 & (p-1)(p + \frac{\frac{1}{6}10N + \frac{1}{6}M}{8N^2M + 5NM^2 + \frac{1}{6}58N^3 + \frac{1}{6}16M^3}) > 0
 \end{aligned}$$

and

$$speedup < p^2$$

since:

$$\begin{aligned}
 & (3NM + N^2 - \frac{1}{6}10N - \frac{1}{6}M)p^2 \\
 & - (3NM + N^2) - (\frac{1}{6}10N + \frac{1}{6}M) \\
 & = (3NM + N^2 - \frac{1}{6}10N - \frac{1}{6}M) \\
 & (p-1)(p + \frac{\frac{1}{6}10N + \frac{1}{6}M}{3NM + N^2 - \frac{1}{6}10N - \frac{1}{6}M}) > 0
 \end{aligned}$$

### Time invariant filters

The time invariant Kalman filter case is similar to the time varying Kalman filter case.

For the time invariant Lainiotis filter, we have:

$$\begin{aligned}
 speedup = & \frac{1}{p} \frac{4nm + \frac{1}{6}(58n^3 + 9n^2 - 7n)}{4NM + \frac{1}{6}(58N^3 + 9N^2 - 7N)} \\
 = & \frac{1}{p} \frac{4m + \frac{1}{6}(58n^2 + 9n - 7)}{4M + \frac{1}{6}(58N^2 + 9N - 7)} \frac{n}{N} \\
 = & \frac{4pM + \frac{1}{6}(58p^2N^2 + 9pN - 7)}{4M + \frac{1}{6}(58N^2 + 9N - 7)}
 \end{aligned}$$

Then

$$p < speedup$$

since:

$$\begin{aligned}
 & (58N^2)p^2 - (58N^2 - 7)p - 7 \\
 = & 58N^2(p-1)(p + \frac{7}{58N^2}) > 0
 \end{aligned}$$

and

$$speedup < p^2$$

since:

$$\begin{aligned}
 & (24M + 9N - 7)p^2 - (24M + 9N)p + 7 \\
 = & (24M + 9N - 7)(p-1)(p - \frac{7}{24M + 9N - 7}) > 0
 \end{aligned}$$

### Steady state filters

For both the steady state Kalman and Lainiotis filters, we have:

$$\begin{aligned}
 speedup = & \frac{1}{p} \frac{2n^2 + 2nm - n}{2N^2 + 2NM - N} \\
 = & \frac{1}{p} \frac{2n + 2m - 1}{2N + 2M - 1} \frac{n}{N} \\
 = & \frac{2pN + 2pM - 1}{2N + 2M - 1}
 \end{aligned}$$

Then

$$p < speedup = \frac{2pN + 2pM - 1}{2N + 2M - 1}$$

since:

$$2pN + 2pM - 1 > p(2N + 2M - 1) = 2pN + 2pM - p$$

and

$$speedup = \frac{2pN + 2pM - 1}{2N + 2M - 1} < p^2$$

since:

$$\begin{aligned}
 & p^2(2N + 2M - 1) - p(2N + 2M) + 1 \\
 = & (2N + 2M - 1)(p-1)(p - \frac{1}{2N + 2M - 1}) > 0
 \end{aligned}$$

Thus, we proved that:

$$p < speedup < p^2 \quad (4)$$

This result holds for both Kalman and Lainiotis filters and for all the possible implementations (time varying/time invariant/steady state). So, the model partition leads to a significant speedup. This speedup increases as the number  $p$  of the partitioned models increases. It is worth to note that in the case of parallel implementation of the partitioned models, the speedup becomes  $p$  times greater. The important result of (4) is further illustrated through the following three examples.

### Example 1

Consider the 2D random walk system described in [5] for head tracking in two dimensions simultaneously (single model) with  $n=2$  and  $m=2$ . The 2D random walk system parameters are:

$$F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

We are able to partition the system into  $p=2$  1D random walk systems with  $N=1$  and  $M=1$ . The 1D random walk system parameters are:  $F=1, H=1, Q=1, R=0.1$ . The calculation burdens of time invariant as well as the steady state Kalman and Lainiotis filters and the corresponding speedups are shown in *Table 3*.

**Table 3** Example 1 speedup

System	Filter	Single model $n=2$ $m=2$	Partitioned models $N=1$ $M=1$ $p=2$	Speedup
Time	KF	122	32	3.813
Invariant	LF	97	28	3.464
Steady	KF	14	6	2.333
State	LF	14	6	2.333

**Table 4** Example 2 speedup

System	Filter	Single model $n=4$ $m=2$	Partitioned models $N=2$ $M=1$ $p=2$	Speedup
Time	KF	509	138	3.688
Invariant	LF	670	178	3.764
Steady	KF	44	20	2.200
State	LF	44	20	2.200

### Example 2

Consider the systems described in [6] for mobile position tracking in two dimensions simultaneously (single model) or separately (partitioned models). In

this example the single model with  $n=4$  and  $m=2$  is partitioned into  $p=2$  models with  $N=2$  and  $M=1$ . The calculation burdens of time invariant as well as the steady state Kalman and Lainiotis filters and the corresponding speedups are shown in *Table 4*.

### Example 3

Consider the systems described in [7] for mobile position tracking in three dimensions simultaneously (single model) or separately (partitioned models). In this example the single model with  $n=6$  and  $m=3$  is partitioned into  $p=3$  models with  $N=2$  and  $M=1$ . The calculation burdens of time invariant as well as the steady state Kalman and Lainiotis filters and the corresponding speedups are shown in *Table 5*.

**Table 5** Example 3 speedup

System	Filter	Single model $n=6$ $m=3$	Partitioned models $N=2$ $M=1$ $p=3$	Speedup
Time	KF	1660	207	8.019
Invariant	LF	2207	267	8.266
Steady	KF	102	30	3.400
state	LF	102	30	3.400

The simulation results confirm that the model partition into  $p$  independent models leads to a significant speedup in the range  $(p, p^2)$  for both Kalman and Lainiotis filters. In fact, this speedup increases as the number  $p$  of the partitioned models increases.

## 6. Conclusion

We investigated the partition of linear systems into lower dimensioned independent linear systems when implementing Kalman filter and Lainiotis filters in linear estimation. We are able to separate a single model into  $p \geq 2$  partitioned models, if the partitioned models are independent of each other. Concerning the behaviour of the filters, it was shown that the implementation of the time varying/time invariant/steady state Kalman/Lainiotis filter for the single model is equivalent to the implementation for the partitioned models, since all filters compute the same estimates and estimation error covariance. Thus, the partition does not affect the behaviour of any filter; the filters remain optimal with the results of the same accuracy. Finally, it was shown that



model partition leads to a significant speedup. In fact the partition into  $p \geq 2$  independent models leads to a speedup, such that  $p < \text{speedup} < p^2$ , for both Kalman and Lainiotis filters. This speedup increases as the number  $p$  of the partitioned models increases.

### Acknowledgment

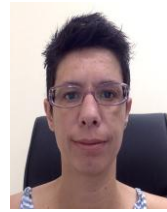
None.

### Conflicts of interest

The authors have no conflicts of interest to declare.

### References

- [1] Anderson BDO, Moore JB. Optimal filtering. Dover Publications, New York, USA; 2005.
- [2] Kalman RE. A new approach to linear filtering and prediction problems. Journal of Basic Engineering. 1960; 82(1):35-45.
- [3] Assimakis N, Adam M. Discrete time Kalman and Lainiotis filters comparison. International Journal of Mathematical Analysis. 2007; 1(13):635-59.
- [4] Lainiotis D. Partitioned linear estimation algorithms: Discrete case. IEEE Transactions on Automatic Control. 1975; 20(2):255-7.
- [5] Kim GM, Lin DC, Kim JH, Yoon SH. Robust head tracking based on hybrid color histogram and random walk Kalman filter. North Carolina Agricultural and Technical State University Greensboro department of Electrical Engineering; 2002.
- [6] Assimakis N, Adam M. Global systems for mobile position tracking using Kalman and Lainiotis filters. The Scientific World Journal. 2014:1-10.
- [7] Assimakis A, Adam M. Mobile position tracking in three dimensions using Kalman and Lainiotis filters. The Open Mathematics Journal. 2015; 8(1):1-6.



**Maria G. 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. She is currently a Visiting Lecturer in the Computer Science Department, University of Thessaly, Greece. Her research interests include video compression, scalable video coding, rate-distortion optimization, as well as computer architecture. Email: mkoziri@uth.gr



**Thanasis Loukopoulos** received his Ph.D. degree in Computer Science from the Hong Kong University of Science and Technology. He is currently Lecturer at the Department of Computer Science and Biomedical Informatics of the University of Thessaly, Greece. His research interests are energy saving, resource management and optimization problems in distributed systems with a focus on cloud systems and sensor networks.



**Maria 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 Thessaly, where she teaches linear algebra, calculus and applied mathematics. Her research interests include linear algebra, spectral analysis, matrix theory and its applications in dynamical systems and control theory.



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