

Kalman and Lainiotis Filters for Partially Time-Varying Models

NICHOLAS ASSIMAKIS

Department of Digital Industry Technologies,
National and Kapodistrian University of Athens,
34400 Psachna Evias,
GREECE

Abstract: - The Kalman and the Lainiotis filters are integral parts of estimation theory, arising in linear estimation. They are related to time-varying state-space models which describe the relation between the state consisting of n elements and the observation (measurement) consisting of m elements. The equivalence of the Kalman and the Lainiotis filters means that they compute iteratively the same estimation statistics, i.e. state estimation and estimation error covariance. It is known from previous works that the time-varying Kalman filter outperforms the time-varying Lainiotis filter, in the sense that it is always faster. In this paper, the partially time-varying state-space models are considered, where the transition matrix is time-varying while the observation (measurement) matrix and the noises covariances are time-invariant. In this case, the Lainiotis filter is likely to be faster than the Kalman filter; in fact, the dimensions of the model determine which filter is the fastest.

Key-Words: - Kalman filter, Lainiotis filter, State-Space Model, Time-Varying Model, Partially Time-Varying Model, Estimation.

Received: April 17, 2026. Revised: June 25, 2026. Accepted: July 27, 2026. Published: October 6, 2026.

1 Introduction

The Kalman filter (KF) is a very famous estimation and prediction algorithm [1] and the Lainiotis filter (LF) is a known estimation algorithm [2]. Filtering (estimating, predicting, smoothing) the state from noisy observations (measurements) is without doubt a very important process, as it has been successfully used in prediction and estimation in many real world problems and applications.

The Kalman filter (KF) is considered as the most important estimation algorithm in linear estimation theory because it provides an optimal method to derive the estimation of the state of a model (system) from the observations (measurements) containing noise. Under certain statistical conditions, the Kalman filter is the optimal estimator, meaning it minimizes the MSE (Mean Squared Error) of the estimated variables of the state. It is extremely useful in real-time applications; for example, the Kalman filter has been implemented for autonomous orbit determination [3], satellite orbit determination [4], GPS position estimation [5], power generation prediction [6], cases prediction of Covid-19 [7], structural parameter tracking [8], Kalman Filter-based Multi-Person Tracking [9] and many other applications [10].

The Lainiotis filter (LF) is a very important estimation algorithm in linear estimation theory because it provides a computationally efficient and often faster alternative to the Kalman filter for state estimation. It is based on the “partitioning” approach by partitioning the estimation problem into smaller and decoupled elementary solutions. It is a “prediction-free” estimation algorithm that computes the state estimation without needing the intermediate state prediction, as the Kalman filter does. It is useful in real-time applications; for example, the Lainiotis filter has been implemented for sonar and radar tracking [11], seismic signal processing [12], electric load estimation [13] and many other applications [14].

The KF and the LF appear in the theory of linear estimation and they are related to time-varying state-space models [15]:

$$x(k+1) = A(k+1) \cdot x(k) + n_x(k) \quad (1)$$

$$y(k) = B(k) \cdot x(k) + n_y(k) \quad (2)$$

This discrete-time model is a mathematical framework representing a dynamic system whose state changes at distinct instants of time (the time is not continuous). It models systems using linear difference equations to map state and observation

(measurement) relations. Instead of continuous time derivatives, the model uses future state updates based on the current state and the current observation (measurement). Continuous systems can be converted to discrete models via sampling. Such models are often used in digital control systems, in digital signal processing, in time series analysis.

In this time-varying model, the time is denoted by k (discrete non negative time). Also, the state is a vector with n elements and is denoted as $x(k)$ and the observation (measurement) is a vector with m elements and is denoted as $y(k)$. Furthermore, $A(k)$ denotes the square transition matrix and $B(k)$ denotes the non-square observation (measurement) matrix. Moreover, $n_x(k)$ is the state noise, having the same dimension as the state and $n_y(k)$ is the noise of the observation (measurement), having the same dimension as the observation. The noises of the state and the observation (measurement) are both normal (Gaussian) processes with mean equal to zero and covariances that are defined (known); in fact $C(k)$ is the covariance of the noise of the state and $D(k)$ is the covariance of the noise of the observation (measurement) and they are both symmetric. In addition, the initial state is a normal (Gaussian) random vector variable, for which the mean x_0 and the covariance S_0 are assumed to be known.

The KF is an iterative algorithm that computes iteratively a) the state estimation statistics, i.e. the $n \times 1$ estimation (mean) $x(k|k)$ with the related square $n \times n$ estimation error covariance $S(k|k)$ and b) the state prediction statistics, i.e. the $n \times 1$ prediction (mean) $x(k+1|k)$ with the related square $n \times n$ prediction error covariance $S(k+1|k)$. These computations involve the $n \times m$ Kalman filter gain $G(k)$. The LF is also an iterative algorithm that computes iteratively the state estimation statistics. Both filters derive the desired estimation using all the available observations (measurements) until the estimation time.

The model considered in (1)-(2) is a linear model since it contains linear functions and is the basis for linear estimation. In linear prediction, the KF is optimal in relation to the first and second order statistics of the estimation. That means that no other filter is better than Kalman filter; this is not true in non-linear estimation. It is also known [15] that the equivalence of the KF and the LF means that they compute the same estimation, given the same initial conditions. In fact, when the initial conditions for both filters are the same, the two filters compute the same statistics of the estimation. It is then clear that

both filters are optimal, concerning the linear estimation.

However, the two filters are competitive in terms of estimation computation time. Since both filters are iterative algorithms, the computational time of interest is the computational time of each iteration. This per-iteration computational time is due to the structure of the equations of the filters. In fact, the computational times of the filters depends on the model dimensions. This has been established in [15], where it has been stated that the time-varying KF is always faster than the time-varying LF. Moreover, in the case of time-invariant state-space models where all the parameters of the model are time-invariant, the time-invariant KF is likely to be faster than the time-invariant LF and this depends on the model dimensions n and m : in fact, the time-invariant KF is favored over the time-invariant LF when n is sufficiently greater than m , while the time-invariant LF is favored over the time-invariant KF when m is sufficiently greater than n . Finally, in the steady-state case the two steady-state filters are identical and therefore they have the same computational times.

In this paper, the partially time-varying models are considered, where the transition matrix is time-varying while the observation (measurement) matrix is time-invariant. Such models arise when it is desired to estimate all or some of the measured quantities; such a model is referred in [16]. The novelty of the paper consists in showing that the Lainiotis filter is likely to be faster than the Kalman filter, in the case where the noises covariances are also time-invariant. This is very important because it allows the production of the estimation in the shortest time, when the model is partially time-varying.

2 Time-Varying Kalman and Lainiotis Filtering Algorithms

In the general case of *time-varying models*, where all the model parameters are time-varying, the time-varying KF and LF algorithms are derived.

Time-Varying Kalman Filter (TVKF)
Initial prediction
Iterations: starting from time zero
Input: previous prediction observation (measurement)
Output: next estimation next prediction
On-line computations
Kalman filter gain
$G(k) = S(k k-1)B^T(k)$

$[B(k)S(k k-1)B^T(k) + D(k)]^{-1}$ estimation $x(k k) = [I - G(k)B(k)]x(k k-1) + G(k)y(k)$ $S(k k) = [I - G(k)B(k)]S(k k-1)$ prediction $x(k+1 k) = A(k)x(k k)$ $S(k+1 k) = C(k) + A(k+1)S(k k)A^T(k+1)$
--

Comments.

1. When no observation (measurement) is exact – which is a normal condition – $D(k)$ is positive definite ensuring the required matrix reversibility in equation of the Kalman filter gain.
2. At each iteration, implementing the time-varying Kalman filter requires inverting an $m \times m$ matrix.

Note that the time-varying KF computes successively (iteratively) the estimates starting from time zero using the available observations (measurements) and the available initial prediction. It is evident that the order of appearance of the time-varying KF equations can be changed using the initial estimation. This initialization is used in the time-varying LF equations.

Time-Varying Lainiotis Filter (TVLF)
Initial prediction Initialization $x(0 0) = [I - G(0)B(0)]x(0 -1) + G(0)y(0)$ $= [I - G(0)B(0)]x_0 + G(0)y(0)$ $S(0 0) = [I - G(0)B(0)]S(0 -1) = [I - G(0)B(0)]S_0$ where $G(0) = S(0 -1)B^T(0)[B(0)S(0 -1)B^T(0) + D(0)]^{-1}$ $= S_0B^T(0)[B(0)S_0B^T(0) + D(0)]^{-1}$
Iterations: starting from time zero Input: previous estimation observation (measurement) Output: next estimation On-line computations estimation $x(k+1 k+1) = G_n(k+1)y(k+1)$ $+ F_n(k+1)[I + S(k k)O_n(k+1)]^{-1}$ $\{x(k k) + S(k k)G_m(k+1)y(k+1)\}$ $S(k+1 k+1) = P_n(k+1)$ $+ F_n(k+1)[I + S(k k)O_n(k+1)]^{-1}S(k k)F_n^T(k+1)$
Lainiotis filter parameters On-line computations $E(k+1) = [B(k+1)C(k)B^T(k+1) + D(k+1)]^{-1}$ $G_n(k+1) = C(k)B^T(k+1)E(k+1)$ $G_m(k+1) = A^T(k+1)B^T(k+1)E(k+1)$ $P_n(k+1) = [I - G_n(k+1)B(k+1)]C(k)$ $F_n(k+1) = [I - G_n(k+1)B(k+1)]A(k+1)$ $O_n(k+1) = A^T(k+1)B^T(k+1)E(k+1)B(k+1)A(k+1)$

Comments.

1. The parameters of the Lainiotis filter are calculated on-line, at all iterations.
2. When no observation (measurement) is exact – which is a normal condition – $D(0)$ is positive

definite ensuring the required matrix reversibility in the initialization equations.

3. When no observation (measurement) is exact – which is a normal condition – $D(k)$ is positive definite ensuring the required matrix reversibility in the Lainiotis filter parameters equation, at all iterations.
4. The identity matrix I ensures the required matrix reversibility in the estimation equations, at all iterations.
5. At each iteration, implementing the time-varying Lainiotis filter requires inverting an $m \times m$ matrix as well as inverting an $n \times n$ matrix.

Note that the time-varying LF computes the estimates starting from time zero ($x(0|0)$ is computed in initialization phase) using the observations (measurements) and the available initial prediction.

Concerning the time-varying filters, is established has been [15] that the two filters compute the same statistics of the estimation, i.e. the state estimation (mean) and the estimation error covariance, given the same initial conditions. Moreover, the participation of the observations (measurements) in the estimation computation depends on the Kalman filter gain, which is the factor that is multiplied by the observation (measurement) in the estimation equation. In [15] it has been shown that this factor remains the same for both filters.

In the case of *partially time-varying models*, where $A(k)$ is time-varying and all other model parameters are time- invariant, the partially time-varying KF and LF algorithms are derived:

Partially Time-Varying Kalman Filter (PTVKF)
Initial prediction Iterations: starting from time zero Input: previous prediction observation (measurement) Output: next estimation next prediction On-line computations Kalman filter gain $G(k) = S(k k-1)B^T[BS(k k-1)B^T + D]^{-1}$ estimation $x(k k) = [I - G(k)B]x(k k-1) + G(k)y(k)$ $S(k k) = [I - G(k)B]S(k k-1)$ prediction $x(k+1 k) = A(k+1)x(k k)$ $S(k+1 k) = C + A(k+1)S(k k)A^T(k+1)$

Comments.

1. When no observation (measurement) is exact – which is a normal condition – D is positive definite

ensuring the required matrix reversibility in equation of the Kalman filter gain.

2. At each iteration, implementing the partially time-varying Kalman filter requires inverting an $m \times m$ matrix.

Note that the partially time-varying KF computes successively (iteratively) the estimates starting from time zero using the available observations (measurements) and the available initial prediction. It is evident that the order of appearance of the partially time-varying KF equations can be changed using the initial estimation. This initialization is used in the partially time-varying LF equations.

Partially Time-Varying Lainiotis Filter (PTVLF)
Initial prediction Initialization $x(0 0) = [I - G(0)B]x((0 -1)) + G(0)y(0)$ $= [I - G(0)B]x_0 + G(0)y(0)$ $S(0 0) = [I - G(0)B]S(0 -1) = [I - G(0)B]S_0$ where $G(0) = S(0 -1)B^T[BS(0 -1)B^T + D]^{-1}$ $= S_0B^T[BS_0B^T + D]^{-1}$
Iterations: starting from time zero Input: previous estimation observation (measurement) Output: next estimation On-line computations estimation $x(k+1 k+1) = G_n y(k+1)$ $+ F_n(k+1)[I + S(k k)O_n(k+1)]^{-1}$ $\{x(k k) + S(k k)G_m(k+1)y(k+1)\}$ $S(k+1 k+1) = P_n$ $+ F_n(k+1)[I + S(k k)O_n(k+1)]^{-1}S(k k)F_n^T(k+1)$
Lainiotis filter parameters Off-line computations $E = [BCB^T + D]^{-1}$ $G_n = CH^TE$ $P_n = [I - G_nB]C$ On-line computations $G_m(k+1) = A^T(k+1)B^TE$ $F_n(k+1) = [I - G_nB]A(k+1)$ $O_n(k+1) = A^T(k+1)B^TEBA(k+1)$

Comments.

1. It is clear that the Lainiotis filter parameters E, G_n, P_n are calculated off-line, since they are time-invariant.
2. The other parameters of the Lainiotis filter are calculated on-line, at all iterations.
3. When no observation (measurement) is exact – which is a normal condition – D is positive definite ensuring the required matrix reversibility in the initialization equations.
4. When no observation (measurement) is exact – which is a normal condition – D is positive definite

ensuring the required matrix reversibility in the Lainiotis filter parameters equation.

5. The identity matrix I ensures the required matrix reversibility in the estimation equations, at all iterations.

6. At each iteration, implementing the partially time-varying Lainiotis filter requires inverting an $n \times n$ matrix.

Note that the partially time-varying LF computes the estimates starting from time zero ($x(0|0)$ is computed in initialization phase) using the observations (measurements) and the available initial prediction.

Concerning the partially time-varying filters, following the ideas in [15] it can be stated that the two filters compute the same statistics of the estimation, i.e. the state estimation (mean) and the estimation error covariance, given the same initial conditions. Moreover, the participation of the observations (measurements) in the estimation computation depends on the Kalman filter gain, which is the factor that is multiplied by the observation (measurement) in the estimation equation. Following the ideas in [15] it can be shown that this factor remains the same for both filters.

Recapitulating, the conclusion is that a) the Kalman filter computes successively the estimates $x(0|0), x(1|1), x(2|2), \dots$ using the observations (measurements) $y(0), y(1), y(2), \dots$ and b) the Lainiotis filter computes the estimate $x(0|0)$ using the observation (measurement) $y(0)$ (initialization) and computes successively the estimates $x(1|1), x(2|2), \dots$ using the observations (measurements) $y(1), y(2), \dots$; this means that both filters compute the same estimates given the same initial conditions.

It is also worth noting that a) in the general time-varying model case, the Kalman filter algorithm inverses a $m \times m$ matrix at each iteration, while the Lainiotis filter algorithm inverses an $m \times m$ matrix plus an $n \times n$ matrix and b) in the special partially time-varying model case, the Kalman filter algorithm inverses a $m \times m$ matrix at each iteration, while the Lainiotis filter algorithm inverses only an $n \times n$ matrix.

3 Computational Requirements of the Algorithms

Both KF and LF are clearly iterative algorithms, so they compute the estimation at a time instant, performing equal numbers of iterations. Therefore, the comparison of the filters will be made comparing their computational time, i.e. the required calculation burden needed for the on-line calculations in each iteration. Thus, the calculations that are repeated iteratively are taken into account, while the calculations that are done only once (initialization) are not taken into account. All matrix manipulation operations required to implement the filters, reduce scalar operations [15].

The calculation burdens of Kalman and Lainiotis filtering algorithms are determined analytically in Appendix in Table 4, Table 5, Table 6 and Table 7.

For *time-varying models*, the calculation burden CB_{TVKF} of the time-varying KF and the calculation burden CB_{TLKF} of the time-varying LF are presented in Table 1.

Table 1. Calculation burdens of Kalman and Lainiotis filters for time-varying models

Filter	Model Dimensions	Calculation Burden
TVKF	$n \geq 2$ $m \geq 2$	$CB_{TVKF} = \frac{1}{2}(8n^3 + 7n^2 - 3n) + 4n^2m + nm + 3nm^2 + \frac{1}{6}(16m^3 - 3m^2 - m)$
TVKF	$n \geq 2$ $m = 1$	$CB_{TVKF} = \frac{1}{2}(8n^3 + 15n^2 + 5n + 2)$
TVKF	$n = 1$ $m \geq 2$	$CB_{TVKF} = \frac{1}{6}(16m^3 + 15m^2 + 29m + 36)$
TVKF	$n = 1$ $m = 1$	$CB_{TVKF} = 15$
TVLF	$n \geq 2$ $m \geq 2$	$CB_{TVLF} = \frac{1}{6}(58n^3 + 6n^2 - 10n) + 8n^2m + 3nm + 5nm^2 + \frac{1}{6}(16m^3 - 3m^2 - m)$
TVLF	$n \geq 2$ $m = 1$	$CB_{TVLF} = \frac{1}{6}(58n^3 + 54n^2 + 38n + 6)$
TVLF	$n = 1$ $m \geq 2$	$CB_{TVLF} = \frac{1}{6}(16m^3 + 27m^2 + 65m + 48)$
TVLF	$n = 1$ $m = 1$	$CB_{TVLF} = 25$

Source: created by the author

From Table 1 it is confirmed that Kalman filter is superior to Lainiotis filter since $CB_{TVKF} - CB_{TVLF} < 0$.

For *partially time-varying models*, the calculation burden CB_{TVKF} of the time-varying KF and the calculation burden CB_{TLKF} of the time-varying LF are presented in Table 2.

Table 2. Calculation burdens of Kalman and Lainiotis filters for partially time-varying models

Filter	Model Dimensions	Calculation Burden
TVKF	$n \geq 2$ $m \geq 2$	$CB_{PTVKF} = \frac{1}{2}(8n^3 + 7n^2 - 3n) + 4n^2m + nm + 3nm^2 + \frac{1}{6}(16m^3 - 3m^2 - m)$
TVKF	$n \geq 2$ $m = 1$	$CB_{PTVKF} = \frac{1}{2}(8n^3 + 15n^2 + 5n + 2)$
TVKF	$n = 1$ $m \geq 2$	$CB_{PTVKF} = \frac{1}{6}(16m^3 + 15m^2 + 29m + 36)$
TVKF	$n = 1$ $m = 1$	$CB_{PTVKF} = 15$
TVLF	$n \geq 2$ $m \geq 2$	$CB_{PTVLF} = \frac{1}{6}(58n^3 + 6n^2 - 10n) + 5n^2m + 3nm + 2nm^2$
TVLF	$n \geq 2$ $m = 1$	$CB_{PTVLF} = \frac{1}{6}(58n^3 + 36n^2 + 20n)$
TVLF	$n = 1$ $m \geq 2$	$CB_{PTVLF} = \frac{1}{6}(12m^2 + 48m + 48)$
TVLF	$n = 1$ $m = 1$	$CB_{PTVLF} = 18$

Source: created by the author

It is worth noting that the partially time-varying models a) do not affect the calculation burden of the KF, since $CB_{TVKF} = CB_{PTVKF}$ and b) do affect the calculation burden of LF, since $CB_{TVKF} = CB_{PTVKF}$; in fact, the Lainiotis filter parameters E, G_n, P_n are calculated off-line, since they are time-invariant, while the remaining Lainiotis filter parameters are calculated on-line.

From Table 2 it is confirmed that, for partially time-varying models, the Lainiotis filter is likely to be faster than the Kalman filter; which filter is faster depends on the dimensions of the model:

i) $n \geq 2$ and $m \geq 2$

In this the general multi-dimensional case, it holds

$$CB_{PTVLF} - CB_{PTVKF} = \frac{1}{6}(34n^3 - 15n^2 - n) + n^2m + 2nm - nm^2 - \frac{1}{6}(16m^3 - 3m^2 - m)$$

Figure 1 depicts the faster partially time-varying filter with respect to the model dimensions n and m .

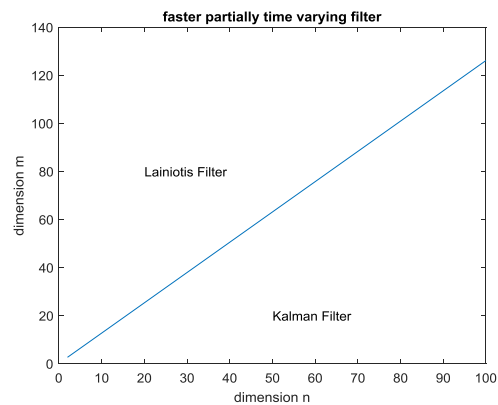


Fig. 1: The faster partially time-varying filter

Source: created by the author

Figure 1 was created following the ideas in [15]. From Figure 1 it can be decided which filter is faster given the model dimensions: the Kalman filter is faster for dimensions that occur below the straight line, while the Lainiotis filter is faster for dimensions that occur above the straight line. From Figure 1 results that the LF is likely to be faster than the KF when the following inequality (rule of thumb) holds:

$$m \geq 1.25 \cdot n \quad (3)$$

The speedup from the one filter to the other can be computed using Table 2. For example, if $n = 20$ and $m = 10$ Kalman filter outperforms Lainiotis filter with a speedup from LF to KF of 1.758, while if $n = 10$ and $m = 20$ Lainiotis filter outperforms Kalman filter with a speedup from KF to LF of 1.611.

Table 3 presents the percentages of the cases of which filter is the fastest as well as the corresponding maximum and average speedup.

Table 3. Statistics for the faster partially time-varying filter

Fastest Filter	Statistics	n,m 2-4	n,m 2-10	n,m 2-20	n,m 2-100
KF	%	68.75	64	62	60.68
KF	average speedup	1.3411	1.4368	1.4949	1.5569
KF	max speedup	1.7779	2.0857	2.2321	2.3758
LF	%	31.25	36	38	39.32
LF	average speedup	1.3843	1.9662	2.6173	4.7601
LF	max speedup	1.9578	5.2524	11.5322	64.4805

Source: created by the author

From Table 3 it can be seen that a) KF overcomes to LF in the 60% of the cases with a maximum speedup of 2.5 with tendency to increase as the model dimensions increase and b) LF overcomes to KF in the 40% of the cases with an maximum speedup of 10 with very strong tendency to increase as the model dimensions increase.

ii) $n \geq 2$ and $m = 1$

In this case, the Kalman filter is the fastest filter, since:

$$CB_{PTVLF} - CB_{PTVKF} = \frac{1}{6}(34n^3 - 9n^2 + 5n - 6) > 0$$

The speedup from LF to KF is depicted in Figure 2. KF overcomes to LF with a speedup that increases slowly as the model dimension n increases.

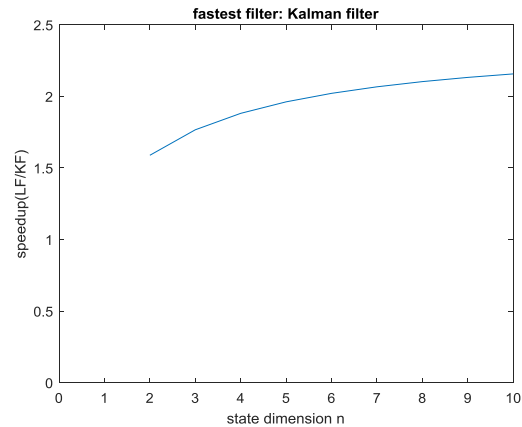


Fig. 2: Speedup from LF to KF when $n \geq 2$ and $m = 1$

Source: created by the author

iii) $n = 1$ and $m \geq 1$

In this case, the Lainiotis filter is the fastest filter, since:

$$CB_{PTVLF} - CB_{PTVKF} = \frac{1}{6}(-16m^3 - 3m^2 + 19m + 12) < 0$$

The speedup from KF to LF is depicted in Figure 3. LF overcomes to KF with a speedup that increases rapidly as the model dimension m increases.

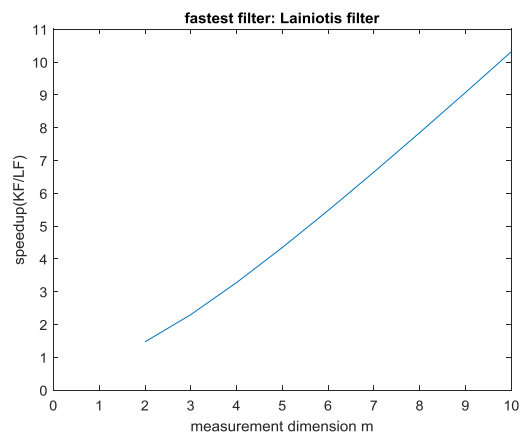


Fig. 3: Speedup from LF to KF when $n = 1$ and $m \geq 2$

Source: created by the author

iv) $n = 1$ and $m = 1$

In this case, the Kalman filter is the fastest filter, since:

$$CB_{PTVLF} - CB_{PTVKF} = 3 > 0$$

The speedup from LF to KF is equal to 1.2.

In the following, two simulations examples are presented.

Simulation example 1.

Consider tracking the weather temperature (scalar state, $n=1$) using a time-varying factor $A(k)$ and one sensor (measurement, $m=1$).

The model parameters are:

$$A(k) = 1 + 0.1 \sin(k), B = 1, C = 0.2, D = 0.1.$$

The initial conditions are: $x_0 = 20, S_0 = 1$.

Simulation is done for one day (the measurements are taken every 5 minutes). The Kalman and the Lainiotis filters produce identical estimations. The real state and the estimations computed by KF and LF are depicted in Figure 4.

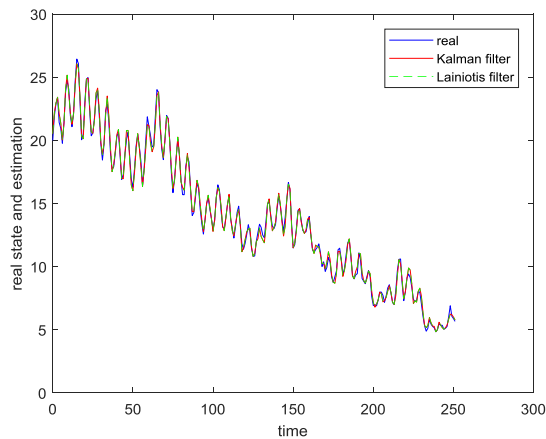


Fig. 4: Simulation example 1: estimation using the Kalman and the Lainiotis filters

Source: created by the author

From Figure 4 it becomes clear that the estimations follow with confidence the real state. Both filters compute identical estimations. Both filters compute identical estimation error variances with a mean value of 0.0733. In this example, the KF is 1.2 times faster than the LF, concluding that KF outperforms LF.

Simulation example 2.

Consider tracking the weather temperature (scalar state, $n=1$) using a time-varying factor $A(k)$ and two sensors (measurement vector, $m=2$): one slow but accurate and one fast but noisy.

The model parameters are:

$$A(k) = 1 + 0.1 \sin(k)$$

$$B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$C = 0.2$$

$$D = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.5 \end{bmatrix}.$$

The initial conditions are: $x_0 = 20, S_0 = 1$.

Simulation is done for one day (the measurements are taken every 5 minutes). The Kalman and the Lainiotis filters produce identical estimations. The real state and the estimations computed by KF and LF are depicted in Figure 5.

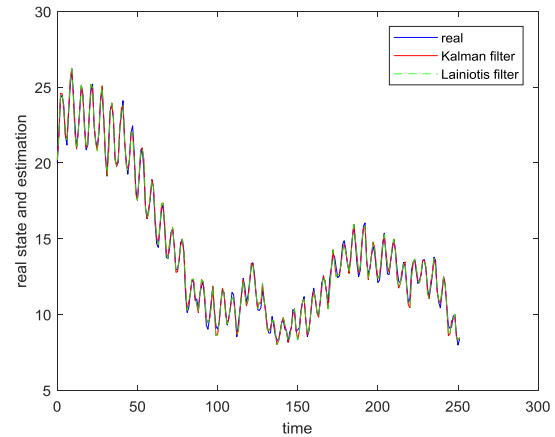


Fig. 5: Simulation example 2: estimation using the Kalman and the Lainiotis filters

Source: created by the author

From Figure 5 it becomes clear that the estimations follow with confidence the real state. Both filters compute identical estimations. Both filters compute identical estimation error variances with a mean value of 0.0634. In this example, the LF is 1.46875 times faster than the KF, concluding that LF outperforms KF.

4 Conclusion

The Kalman filter (KF) is considered as the most important filtering algorithm in linear estimation due to the fact that it is optimal. The Lainiotis filter (LF) is a very important filtering algorithm due to the fact that it provides a computationally efficient alternative to the Kalman filter for state estimation. It is known that the mathematic equivalence of the KF and the LF means that they compute the same estimation, given the same initial conditions. Then it is evident that both filters are optimal.

However, the two filters are competitive in terms of computation time. It is known that in case where the model is time-varying, Kalman filter is superior to Lainiotis filter regarding the calculation speed. In this paper the partially time-varying model has been considered, where the transition matrix is time-varying while the observation (measurement) matrix is time-invariant and the noises covariances are also time-invariant. The effect of the partially time-varying models on the computational times of the KF and the LF has been investigated.

It has been observed that the partially time-varying Kalman filter does not take advantage of the partially time-varying model, while the partially time-varying Lainiotis filter takes advantage of the partially time-varying model. It has been found that in this case, the Lainiotis filter can be faster than the Kalman filter; in fact, which filter is faster depends on dimensions of the model. It has been observed that in the case of partially time-varying models, the implementation of the Kalman filter requires the calculation of the inverse of an $m \times m$ dimensional matrix (recall that m is the observation (measurement) dimension) in each iteration, while the implementation of the Lainiotis filter requires the calculation of the inverse of an $n \times n$ dimensional matrix (recall that n is the state dimension) in each iteration. In fact, when $n \geq 2$ and $m \geq 2$ (multi-dimensional case), the Lainiotis filter is faster than the Kalman filter when $m \geq 1.25n$ (when m is sufficiently greater than n). This result is very important because it allows the production of the estimate in the shortest possible time, when the model is partially time-varying.

Future research may investigate the effect of partially time-varying models on the computational requirements of linear filters (information filters, complex filters) and non-linear filters (Extended Kalman filter, Adaptive Lainiotis filter).

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

The authors wrote, reviewed and edited the content as needed and they have not utilised artificial intelligence (AI) tools. The authors take full responsibility for the content of the publication.

References:

- [1] R. E. Kalman, A new approach to linear filtering and prediction problems, *J. Bas. Eng., Trans. ASME, Ser. D*, vol. 82, pp. 34-45, 1960.
- [2] D. G. Lainiotis, Partitioned linear estimation algorithms: Discrete case, *IEEE Transactions on AC*, vol. 20, pp. 255-257 (1975).
- [3] Y. Li, Q. Gui, S. Han, Y. Gu, Tikhonov Regularized Kalman Filter and its Applications in Autonomous Orbit Determination of BDS, *WSEAS Transactions on Mathematics*, vol. 16, pp. 187-196, 2017.
- [4] X. Ren, Y. Yang, J. Zhu, T., Comparing satellite orbit determination by batch processing and extended Kalman filtering using inter-satellite link measurements of the next-generation beidou satellites, *Gps Solutions*, vol. 23, 2019.
- [5] R. Verma, L. Shrinivasan and K. Shreedarshan, GPS/INS integration during GPS outages using machine learning augmented with Kalman filter, *WSEAS Transactions on Systems and Control*, vol.16, pp. 294-301, 2021.
- [6] Y. Yang, T. Yu, W. Zhao, X. Zhu, Kalman Filter Photovoltaic Power Prediction Model Based on Forecasting Experience, *Front. Energy Res., Sec. Smart Grids*, vol. 9, 2021.
- [7] V. C. S. Rao, B. G. Devi, S. Pratapagiri, C. Srinivas, S. Venkatramulu, D. Raghavakumari, Prediction of Covid-19 using Kalman filter algorithm, *AIP Conference Proceedings*, vol. 2418, issue 1, 2022.
- [8] M. Diaz, P.-É. Charbonnel, L. Chamoin, A new Kalman filter approach for structural parameter tracking: Application to the monitoring of damaging structures tested on shaking-tables, *Mechanical Systems and Signal Processing*, vol. 182, 2023.
- [9] H. Jung, S. Kang, T. Kim, H. Kim, ConfTrack: Kalman Filter-based Multi-Person Tracking by Utilizing Confidence Score of Detection Box, *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6583-6592, 2024.
- [10] B. Ristic, S. Arulampalam, N. Gordon, *Beyond The Kalman Filter*, Boston London, Artech House, 2004.
- [11] S. K. Katsikas, D. G. Lainiotis, V. C. Moussas, Lainiotis Filters Application to Sonar and Radar Tracking: A Survey, *IFAC Workshop on Expert Systems and Signal Processing in Marine Automation*, pp. 303-310, 1989.
- [12] S. K. Katsikas, D. G. Lainiotis, Lainiotis filters applications in seismic signal processing, *Nonlinear Analysis: Theory, Methods & Applications*, vol. 30, issue 4, pp. 2385-2395, 1997.
- [13] N. Assimakis, C. Manasis, A. Ktena, Electric load estimation using Kalman and Lainiotis filters, *The 8th Mediterranean Conference on Embedded Computing - MECO'2019*, 2019.
- [14] V. Moussas, The Multi-Model Partitioning Theory: Current Trends and Selected Applications, *3rd TTSAAMS International Conference*, N. Peramos, Greece, 2015.
- [15] N. Assimakis and M. Adam, Discrete time Kalman and Lainiotis filters comparison, *International Journal of Mathematical Analysis (IJMA)*, vol. 1, no 13, pp. 635-659, 2007.
- [16] R. Zanetti, C. D'Souza, Recursive Implementations of the Schmidt-Kalman 'Consider' Filter, *J. of Astronaut Sci.*, vol. 60, pp. 672-685, 2013.

APPENDIX

Table 4. Calculation burden of TVKF

TVKF	
Matrix Operation	Calculation Burden
$B(k)S(k k-1)$	$2n^2m - nm$
$B(k)S(k k-1)B^T(k)$	$nm^2 + nm - \frac{1}{2}m^2 - \frac{1}{2}m$
$B(k)S(k k-1)B^T(k) + D(k)$	$\frac{1}{2}m^2 + \frac{1}{2}m$
$[B(k)S(k k-1)B^T(k) + D(k)]^{-1}$	$\begin{cases} \frac{1}{6}(16m^3 - 3m^2 - m) & m \geq 2 \\ 1 & m = 1 \end{cases}$
$G(k) = (B(k)S(k k-1))^T$ $[B(k)S(k k-1)B^T(k) + D(k)]^{-1}$ $= S(k k-1)B^T(k)$ $[B(k)S(k k-1)B^T(k) + D(k)]^{-1}$	$2nm^2 - nm$
$G(k)B(k)$	$2n^2m - n^2$
$I - G(k)B(k)$	n
$[I - G(k)B(k)]x(k k-1)$	$2n^2 - n$
$G(k)y(k)$	$2nm - n$
$x(k k) = [I - G(k)B(k)]x(k k-1)$ $+ G(k)y(k)$	n
$S(k k) = [I - G(k)B(k)]S(k k-1)$	$n^3 + \frac{1}{2}n^2 - \frac{1}{2}n$
$x(k+1 k) = A(k)x(k k)$	$2n^2 - n$
$A(k+1)S(k k)$	$2n^3 - n^2$
$A(k+1)S(k k)A^T(k+1)$	$n^3 + \frac{1}{2}n^2 - \frac{1}{2}n$
$S(k+1 k) = C(k)$ $+ A(k+1)S(k k)A^T(k+1)$	$\frac{1}{2}n^2 + \frac{1}{2}n$

Source: created by the author

Table 5. Calculation burden of PTVKF

PTVKF	
Matrix Operation	Calculation Burden
$BS(k k-1)$	$2n^2m - nm$
$BS(k k-1)B^T$	$nm^2 + nm - \frac{1}{2}m^2 - \frac{1}{2}m$
$BS(k k-1)B^T + D$	$\frac{1}{2}m^2 + \frac{1}{2}m$
$[BS(k k-1)B^T + D]^{-1}$	$\begin{cases} \frac{1}{6}(16m^3 - 3m^2 - m) & m \geq 2 \\ 1 & m = 1 \end{cases}$
$G(k) = (BS(k k-1))^T$ $[BS(k k-1)B^T + D]^{-1}$ $= S(k k-1)B^T$ $[BS(k k-1)B^T + D]^{-1}$	$2nm^2 - nm$
$G(k)B$	$2n^2m - n^2$
$I - G(k)B$	n
$[I - G(k)B]x(k k-1)$	$2n^2 - n$
$G(k)y(k)$	$2nm - n$
$x(k k) = [I - G(k)B]x(k k-1)$ $+ G(k)y(k)$	n
$S(k k) = [I - G(k)B]S(k k-1)$	$n^3 + \frac{1}{2}n^2 - \frac{1}{2}n$
$x(k+1 k) = A(k)x(k k)$	$2n^2 - n$
$A(k+1)S(k k)$	$2n^3 - n^2$
$A(k+1)S(k k)A^T(k+1)$	$n^3 + \frac{1}{2}n^2 - \frac{1}{2}n$
$S(k+1 k) = C$ $+ A(k+1)S(k k)A^T(k+1)$	$\frac{1}{2}n^2 + \frac{1}{2}n$

Source: created by the author

Table 6. Calculation burden of TVLF

TVLF	
Matrix Operation	Calculation Burden
Lainiotis filter parameters	
$B(k+1)C(k)$	$2n^2m - nm$
$B(k+1)C(k)B^T(k+1)$	$nm^2 + nm - \frac{1}{2}m^2 - \frac{1}{2}m$
$B(k+1)C(k)B^T(k+1) + D(k+1)$	$\frac{1}{2}m^2 + \frac{1}{2}m$
$E(k+1)$ $= [B(k+1)C(k)B^T(k+1) + D(k+1)]^{-1}$	$\begin{cases} \frac{1}{6}(16m^3 - 3m^2 - m) & m \geq 2 \\ 1 & m = 1 \end{cases}$
$G_n(k+1) = (B(k+1)C(k))^T E(k+1)$ $= C(k)B^T(k+1)E(k+1)$	$2nm^2 - nm$
$B(k+1)A(k+1)$	$2n^2m - nm$
$G_m(k+1) = (B(k+1)A(k+1))^T E(k+1)$ $= A^T(k+1)B^T(k+1)E(k+1)$	$2nm^2 - nm$
$G_n(k+1)B(k+1)C(k)$	$n^2m + nm - \frac{1}{2}n^2 - \frac{1}{2}n$
$P_n(k+1) = C(k) - G_n(k+1)B(k+1)C(k)$ $= [I - G_n(k+1)B(k+1)]C(k)$	$\frac{1}{2}n^2 + \frac{1}{2}n$
$G_n(k+1)B(k+1)A(k+1)$	$2n^2m - n^2$
$F_n(k+1) = A(k+1)$ $- G_n(k+1)B(k+1)A(k+1)$ $= [I - G_n(k+1)B(k+1)]A(k+1)$	n^2
$O_n(k+1) = G_m(k+1)B(k+1)A(k+1)$ $= A^T(k+1)B^T(k+1)$ $E(k+1)B(k+1)A(k+1)$	$n^2m + nm - \frac{1}{2}n^2 - \frac{1}{2}n$
Iterations	
$G_n(k+1)y(k+1)$	$2nm - n$
$G_m(k+1)y(k+1)$	$2nm - n$
$S(k k)O_n(k+1)$	$2n^3 - n^2$
$I + S(k k)O_n(k+1)$	n
$[I + S(k k)O_n(k+1)]^{-1}$	$\begin{cases} (16n^3 - 3n^2 - n)/6 & n \geq 2 \\ 1 & n = 1 \end{cases}$
$F_n(k+1)[I + S(k k)O_n(k+1)]^{-1}$	$2n^3 - n^2$
$S(k k)G_m(k+1)y(k+1)$	$2n^2 - n$
$x(k k) + S(k k)G_m(k+1)y(k+1)$	n
$F_n(k+1)[I + S(k k)O_n(k+1)]^{-1}$ $\{x(k k) + S(k k)G_m(k+1)y(k+1)\}$	$2n^2 - n$
$x(k+1 k+1) = G_n(k+1)y(k+1)$ $+ F_n(k+1)[I + S(k k)O_n(k+1)]^{-1}$ $\{x(k k) + S(k k)G_m(k+1)y(k+1)\}$	n
$F_n(k+1)[I + S(k k)O_n(k+1)]^{-1}S(k k)$	$2n^3 - n^2$
$F_n(k+1)[I + S(k k)O_n(k+1)]^{-1}F_n^T(k+1)$	$n^3 + \frac{1}{2}n^2 - \frac{1}{2}n$
$S(k+1 k+1) = P_n(k+1)$ $+ F_n(k+1)$ $[I + S(k k)O_n(k+1)]^{-1}S(k k)F_n^T(k+1)$	$\frac{1}{2}n^2 + \frac{1}{2}n$

Source: created by the author

Table 7. Calculation burden of PTVLF

PTVLF	
Matrix Operation	Calculation Burden
Lainiotis filter parameters	
Off-line computations	
BC	$2n^2m - nm$
BCB^T	$nm^2 + nm - \frac{1}{2}m^2 - \frac{1}{2}m$
$BCB^T + D$	$\frac{1}{2}m^2 + \frac{1}{2}m$
$E = [BCB^T + D]^{-1}$	$\begin{cases} \frac{1}{6}(16m^3 - 3m^2 - m) & m \geq 2 \\ 1 & m = 1 \end{cases}$
$G_n = (BC)^TE = CB^TE$	$2nm^2 - nm$
$G_n BC$	$n^2m + nm - \frac{1}{2}n^2 - \frac{1}{2}n$
$P_n = C - G_n BC = [I - G_n B]C$	$\frac{1}{2}n^2 + \frac{1}{2}n$
On-line computations	
$BA(k+1)$	$2n^2m - nm$
$G_m(k+1) = (BA(k+1))^TE$ $= A^T(k+1)B^TE$	$2nm^2 - nm$
$G_n BA(k+1)$	$2n^2m - n^2$
$F_n(k+1) = A(k+1) - G_n BA(k+1)$ $= [I - G_n B]A(k+1)$	n^2
$O_n(k+1) = G_m(k+1)BA(k+1)$ $= A^T(k+1)B^TEBA(k+1)$	$n^2m + nm - \frac{1}{2}n^2 - \frac{1}{2}n$
Iterations	
$G_n y(k+1)$	$2nm - n$
$G_m(k+1)y(k+1)$	$2nm - n$
$S(k k)O_n(k+1)$	$2n^3 - n^2$
$I + S(k k)O_n(k+1)$	n
$[I + S(k k)O_n(k+1)]^{-1}$	$\begin{cases} (16n^3 - 3n^2 - n)/6 & n \geq 2 \\ 1 & n = 1 \end{cases}$
$F_n(k+1)[I + S(k k)O_n(k+1)]^{-1}$	$2n^3 - n^2$
$S(k k)G_m(k+1)y(k+1)$	$2n^2 - n$
$x(k k) + S(k k)G_m(k+1)y(k+1)$	n
$F_n(k+1)[I + S(k k)O_n(k+1)]^{-1}$ $\{x(k k) + S(k k)G_m(k+1)y(k+1)\}$	$2n^2 - n$
$x(k+1 k+1) = G_n(k+1)y(k+1)$ $+ F_n(k+1)[I + S(k k)O_n(k+1)]^{-1}$ $\{x(k k) + S(k k)G_m(k+1)y(k+1)\}$	n
$F_n(k+1)[I + S(k k)O_n(k+1)]^{-1}S(k k)$	$2n^3 - n^2$
$F_n(k+1)$ $[I + S(k k)O_n(k+1)]^{-1}F_n^T(k+1)$	$n^3 + \frac{1}{2}n^2 - \frac{1}{2}n$
$S(k+1 k+1) = P_n$ $+ F_n(k+1)$ $[I + S(k k)O_n(k+1)]^{-1}$ $S(k k)F_n^T(k+1)$	$\frac{1}{2}n^2 + \frac{1}{2}n$

Note that the off-line calculation is highlighted in gray.

Source: created by the author

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The author contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The author has no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US