Int. Journal of Math. Analysis, Vol. 1, 2007, no. 13, 635 - 659

Discrete Time Kalman and Lainiotis[1] Filters Comparison

N. Assimakis

Department of Informatics with Applications to Biomedicine
University of Central Greece, 35100 Lamia, Greece
Department of Electronics, Technological Educational Institute of Lamia,
35100 Lamia, Greece (assimakis@teilam.gr)

M. Adam

Department of Informatics with Applications to Biomedicine
University of Central Greece, 35100 Lamia, Greece
(maria@math.ntua.gr)

Abstract

A comparison study between the discrete time Kalman filter and the Lainiotis filter is presented. It is pointed out that the two filters calculate the same estimates, thus the filters are equivalent with respect to their behavior. The calculation burdens of the filtering algorithms are determined. A method is proposed to a-priori (before the filters' implementation) decide which filter is the faster one.

Mathematics Subject Classification: 93E11, 93C55, 68Q25, 68U20

Keywords: Kalman filter, Lainiotis filter, Discrete-time, Time-invariant, Time-varying, Analysis of algorithms, Simulation

1 Discrete time Kalman and Lainiotis filters

Estimation plays an important role in many fields of science: applications to aerospace industry, chemical process, communication systems design, control, civil engineering, filtering noise from 2−dimensional images, pollution prediction, power systems are mentioned in [1]. The estimation/filtering problem arises in linear estimation and is associated with time varying systems

[1]This paper is dedicated to the memory of Professor Demetrios Lainiotis.

described by the following state space equations:

$$x(k+1) = F(k+1,k)x(k) + w(k) \tag{1}$$
$$z(k+1) = H(k+1)x(k+1) + v(k+1) \tag{2}$$

where $x(k)$ is the $n-$dimensional state vector at time k, $z(k)$ is the $m-$dimensional measurement vector, $F(k+1,k)$ is the $n \times n$ system transition matrix, $H(k)$ is the $m \times n$ output matrix, $\{w(k)\}$ and $\{v(k)\}$ are Gaussian zero-mean white and uncorrelated random processes, $Q(k)$ and $R(k)$ are the plant and measurement noise covariance matrices, respectively.

The filtering/estimation problem is to produce an estimate at time L of the state vector using measurements till time L, i.e. the aim is to use the measurements set $\{z(1), \ldots, z(L)\}$ in order to calculate an estimate value $x(L/L)$ of the state vector $x(L)$. The covariance matrix $P(L/L)$ of the estimate value $x(L/L)$ is also calculated. The discrete time Kalman filter in [1], [2] and Lainiotis filter in [3] 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)

$$x(k+1/k) = F(k+1,k)x(k/k) \tag{3}$$
$$P(k+1/k) = F(k+1,k)P(k/k)F^T(k+1,k) + Q(k) \tag{4}$$
$$K(k+1) = P(k+1/k)H^T(k+1) \left[H(k+1)P(k+1/k)H^T(k+1) + R(k+1)\right]^{-1} \tag{5}$$
$$x(k+1/k+1) = x(k+1/k) + K(k+1)\left[z(k+1) - H(k+1)x(k+1/k)\right] \tag{6}$$
$$P(k+1/k+1) = P(k+1/k) - K(k+1)H(k+1)P(k+1/k) \tag{7}$$

Time Varying Lainiotis Filter (TVLF)

$$x_n(k+1/k+1) = K_n(k+1)z(k+1) \tag{8}$$
$$M_n(k+1) = K_m(k+1)z(k+1) \tag{9}$$
$$x(k+1/k+1) = x_n(k+1/k+1) + F_n(k+1,k)\left[I + P(k/k)O_n(k+1)\right]^{-1}\left[P(k/k)M_n(k+1) + x(k/k)\right] \tag{10}$$
$$P(k+1/k+1) = P_n(k+1,k) + F_n(k+1,k)\left[I + P(k/k)O_n(k+1)\right]^{-1} P(k/k)F_n^T(k+1,k) \tag{11}$$

where

$$A(k+1) = \left[H(k+1)Q(k)H^T(k+1) + R(k+1)\right]^{-1} \tag{12}$$

$$K_n(k+1) = Q(k)H^T(k+1)A(k+1) \tag{13}$$
$$K_m(k+1) = F^T(k+1,k)H^T(k+1)A(k+1) \tag{14}$$
$$P_n(k+1,k) = Q(k) - K_n(k+1)H(k+1)Q(k) \tag{15}$$
$$F_n(k+1,k) = F(k+1,k) - K_n(k+1)H(k+1)F(k+1,k) \tag{16}$$
$$O_n(k+1) = K_m(k+1)H(k+1)F(k+1,k) \tag{17}$$

Note that the existence of the inverse of the matrices in (5) and (12) is ensured assuming that every covariance matrix $R(k)$ is positive definite; this has the significance that no measurement is exact.

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)

$$x(k+1/k) = F\,x(k/k) \tag{18}$$
$$P(k+1/k) = F\,P(k/k)F^T + Q \tag{19}$$
$$K(k+1) = P(k+1/k)H^T\left[H\,P(k+1/k)H^T + R\right]^{-1} \tag{20}$$
$$x(k+1/k+1) = x(k+1/k) + K(k+1)\left[z(k+1) - Hx(k+1/k)\right] \tag{21}$$
$$P(k+1/k+1) = P(k+1/k) - K(k+1)H\,P(k+1/k) \tag{22}$$

Time Invariant Lainiotis Filter (TILF)

$$x_n(k+1/k+1) = K_n\,z(k+1) \tag{23}$$
$$M_n(k+1) = K_m\,z(k+1) \tag{24}$$
$$\begin{aligned} x(k+1/k+1) = {} & x_n(k+1/k+1) \\ & + F_n\left[I + P(k/k)O_n\right]^{-1}\left[P(k/k)M_n(k+1) + x(k/k)\right] \end{aligned} \tag{25}$$
$$P(k+1/k+1) = P_n(k+1,k) + F_n\left[I + P(k/k)O_n\right]^{-1}P(k/k)F_n^T \tag{26}$$

where the following quantities are calculated off-line:

$$A = \left[H\,Q\,H^T + R\right]^{-1} \tag{27}$$
$$K_n = Q\,H^T\,A \tag{28}$$
$$K_m = F^T H^T A \tag{29}$$
$$P_n = Q - K_n\,H\,Q \tag{30}$$
$$F_n = F - K_n H\,F \tag{31}$$
$$O_n = K_m H\,F \tag{32}$$

For time invariant systems, in [1] it is well known that if the signal process model is asymptotically stable, then there exist steady state values $\overline{P_p}$ and $\overline{P_e}$ of the prediction and estimation error covariances 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)

$$x(k+1/k+1) = A_{Kf}x(k/k) + B_{Kf}\,z(k+1) \tag{33}$$

where the matrices

$$A_{Kf} = \left[I - \overline{K}H\right]F \qquad \text{and} \qquad B_{Kf} = \overline{K} \tag{34}$$

are calculated off-line by first solving the corresponding discrete time Riccati equation as it is reported in [1]

$$\begin{aligned} P(k+1/k) = {} & F\,P(k/k-1)F^T + Q \\ & -FP(k/k-1)H^T\left[HP(k+1/k)H^T + R\right]^{-1}HP(k/k-1)F^T, \end{aligned} \tag{35}$$

and then calculating the steady state gain $\overline{K}$:

$$\overline{K} = \overline{P_p}H^T\left[H\overline{P_p}H^T + R\right]^{-1} \tag{36}$$

Steady State Lainiotis Filter (SSLF)

$$x(k+1/k+1) = A_{Lf}x(k/k) + B_{Lf}\,z(k+1) \tag{37}$$

where the matrices

$$A_{Lf} = F_n\left[I + \overline{P_e}O_n\right]^{-1} \qquad \text{and} \qquad B_{Lf} = K_n + F_n\left[I + \overline{P_e}O_n\right]^{-1}\overline{P_e}K_m \tag{38}$$

are calculated off-line by solving the corresponding discrete time Riccati equation, which is reported in [4]:

$$P(k+1/k+1) = P_n + F_n\left[I + P(k/k)O_n\right]^{-1}P(k/k)F_n^T \tag{39}$$

2 Kalman and Lainiotis filters performance

Using the Kalman and Lainiotis filters equations, it is pointed out in Appendix A that the Kalman and Lainiotis filters calculate theoretically the same estimates as well as the same estimation error covariances. This means that the two filters are theoretically equivalent with respect to their performance.

This result is verified through the following simulation example.

Example 2.1 As in [5] let us try to estimate a scalar random constant c using the time invariant Kalman filter. Let us assume a scalar model ($n = 1$ and $m = 1$), where the state does not change from step to step ($F = 1$)

and the noisy measurement is taken of the state directly ($H = 1$). Assume that the process has positive covariance Q, and that the measurements of the constant are corrupted by a white measurement noise with positive covariance R. Let estimate the random constant $c = 0.75$ setting the following filter parameters: $Q = 10^{-6}$ (the process covariance is very small) and $R = 10^{-4}$ with the initial conditions $x(0/0) = 0$ and $P(0/0) = 1$. We simulated 100 distinct measurements that have error normally distributed around zero with standard division of $\sigma = 0.01$. In Figure 1, we plot the noisy measurements $z(k)$ and the filter estimate $x(k/k)$ for both Kalman and Lainiotis filters. In Figure 2, we plot the corresponding estimation error covariance $P(k/k)$. The two filters calculate the same estimates as well as the same estimation error covariances.

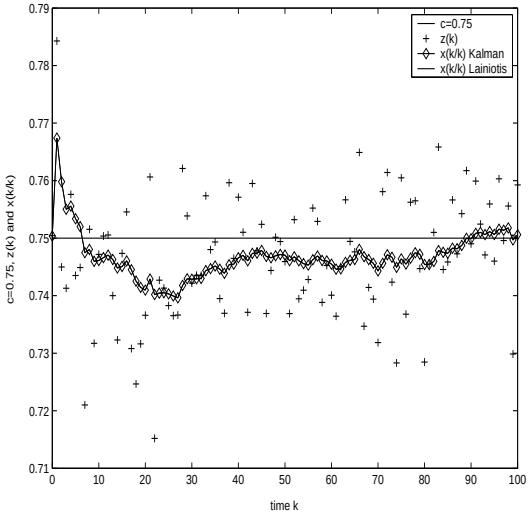

Figure 1: Kalman and Lainiotis filters calculate the same estimates.

3 Kalman and Lainiotis filters comparison study

Both Kalman and Lainiotis filters are recursive algorithms; thus, the total computational time required for the sequential implementation of each is :

$$t_{(alg)} = CB_{(alg)} s_{(alg)} t_{op} \tag{40}$$

where t_{op} is the time required to perform a scalar operation, $CB_{(alg)}$ is the per recursion calculation burden and $s_{(alg)}$ is the number of recursions that each algorithm executes.

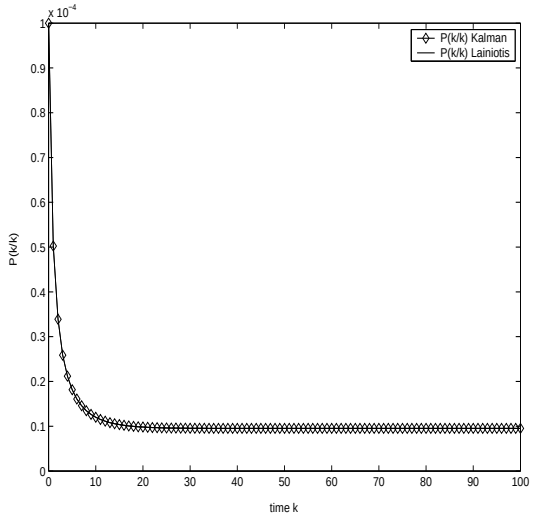

Figure 2: Kalman and Lainiotis filters calculate the same estimation error covariances.

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

Scalar operations are involved in matrix manipulation operations, which are needed for the implementation of the filtering algorithms. Table 1 summarizes the calculation burden of needed matrix operations. The details are given in Appendix B.

Table 1. Calculation burden of matrix operations

Matrix Operation	Calculation Burden
$A_{n\times m} + B_{n\times m} = C_{n\times m}$	nm
$A_{n\times n} + B_{n\times n} = S^{\dagger}_{n\times n}$	$0.5n^2 + 0.5n$
$I^{\ddagger}_{n\times n} + A_{n\times n} = B_{n\times n}$	n
$A_{n\times m} \cdot B_{m\times k} = C_{n\times k}$	$2nmk - nk$
$A_{n\times m} \cdot B_{m\times n} = S^{\dagger}_{n\times n}$	$n^2m + nm - 0.5n^2 - 0.5n$
$[A_{n\times n}]^{-1} = B_{n\times n}$	$\frac{1}{6}(16n^3 - 3n^2 - n)$

† S is a symmetric matrix ‡ I is the identity matrix

The per recursion calculation burdens of the filtering algorithms are determined in Appendix C and summarized in Table 2.

Table 2. Per recursion calculation burden of the filtering algorithms.

System	Filter	Algorithm	Calculation Burden
Time Varying	Kalman Filter	TVKF	$4n^3 + 3.5n^2 - 1.5n + 4n^2m + nm$ $+3nm^2 + (16m^3 - 3m^2 - m)/6$
Time Varying	Lainiotis Filter	TVLF	$8n^2m + 5nm^2 + 3nm$ $+(58n^3 + 6n^2 - 10n)/6$ $+(16m^3 - 3m^2 - m)/6$
Time Invariant	Kalman Filter	TIKF	$4n^3 + 3.5n^2 - 1.5n + 4n^2m + nm$ $+3nm^2 + (16m^3 - 3m^2 - m)/6$
Time Invariant	Lainiotis Filter	TILF	$4nm + (58n^3 + 9n^2 - 7n)/6$
Steady State	Kalman Filter	SSKF	$2n^2 + 2nm - n$
Steady State	Lainiotis Filter	SSLF	$2n^2 + 2nm - n$

4 Selection of the faster filter

In the following, a method is proposed to select the faster filter. Due to the fact that the algorithms' calculation burdens depend on the state vector dimension n and the measurement vector dimension m, the selection of the faster implementation depends on the relationship between n and m. From Table 2, the following results are concluded:

1. The calculation burden required for the time varying Kalman filter implementation is less than the calculation burden required for the time varying Lainiotis filter implementation. Thus, for time varying systems, the faster filter is the Kalman filter.

2. For time invariant systems, the selection of the faster filter depends on the relationship between n and m. In fact, it depends on the difference between the calculation burden required for the time invariant Kalman filter implementation and the calculation burden required for the time invariant Lainiotis filter implementation, which equals to:

$$\begin{aligned} q &= \mathrm{CB}_{TILF} - \mathrm{CB}_{TIKF} \\ &= \frac{34n^3 - 12n^2 + 2n}{6} - 4n^2m + 3nm - 3nm^2 - \frac{16m^3 - 3m^2 - m}{6} \end{aligned} \tag{41}$$

   Thus, for time invariant systems the faster filter is the Kalman filter when $q > 0$ or the Lainiotis filter when $q < 0$.

3. The calculation burden required for the steady state Kalman filter implementation is equal to the calculation burden required for the steady state Lainiotis filter implementation.

4. The faster filter decision can be decided a-priori, given the system and the model order (n and m). Defining the ratio $r = \frac{m}{n}$, from (41) we take the following trinomial with respect to n

$$\widetilde{q} = \frac{6q}{n} = (-16r^3 - 18r^2 - 24r + 34)n^2 + (3r^2 + 18r - 12)n + (r + 2)$$

from where we are able to compute the exact value of r, that defines the faster filter.

For time invariant systems, the areas where the time invariant Kalman or Lainiotis filter implementation is faster, for various values of the model order ($n = 1 \dots 100$ and $m = 1 \dots 100$) are shown in Figure 3. The faster filter's area with respect to the ratio $r = \frac{m}{n}$ is presented in Figure 4.

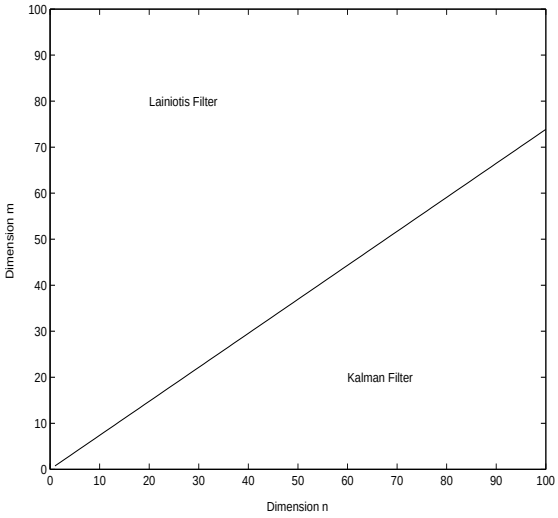

Figure 3: Time invariant systems: The faster filter depends on n and m.

From Figure 4, it is easy to conclude the following Rule of Thumb for time invariant systems. The time invariant Lainiotis filter is faster than the time invariant Kalman filter, when

$$\frac{m}{n} > 0.74 \tag{42}$$

This conclusion is verified through the following examples:

Example 4.1 An AR type model which is taken from [4] is considered in this example, where $n = 3$ and $m = 1$ (note that $\frac{m}{n} = 0.333 < 0.74$). The time invariant Kalman filter is 1.5 times faster than the time invariant Lainiotis

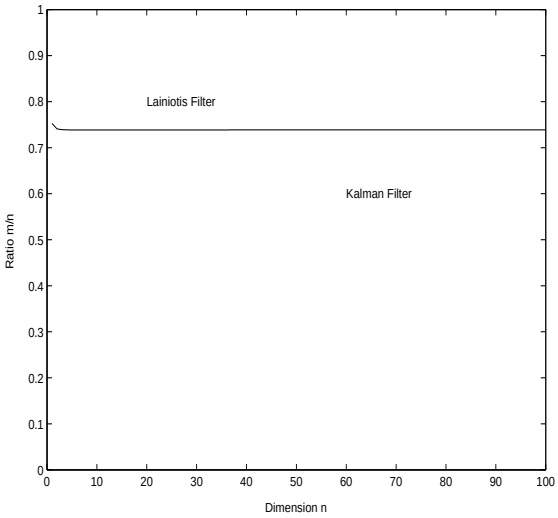

Figure 4: Time invariant systems: The faster filter with respect to the ratio r.

filter: the time improvement (ti) from Lainiotis filter to Kalman filter is equal to:

$$ti_{_{TILF/TIKF}} = \frac{t_{TILF}}{t_{TIKF}} = \frac{\mathrm{CB}_{TILF}}{\mathrm{CB}_{TIKF}} = 1.529$$

Example 4.2 A typical multisensor problem (seismic signal processing) is considered in this example, where $n = 4$ and $m = 1000$ (here $\frac{m}{n} = 250 > 0.74$). The time invariant Lainiotis filter is 160000 times faster than the time invariant Kalman filter: the time improvement from Kalman filter to Lainiotis filter is equal to:

$$ti_{_{TIKF/TILF}} = \frac{t_{TIKF}}{t_{TILF}} = \frac{\mathrm{CB}_{TIKF}}{\mathrm{CB}_{TILF}} = 160971$$

Summarizing the above comparison results, a method is proposed to decide a-priori which filter is faster, given the system (time varying, time invariant, steady state) and model order (n, m):

case time varying system implement Kalman filter

time invariant system

if $\frac{m}{n} > 0.74$ then implement Lainiotis filter

else implement Kalman filter

endif

steady state system implement Kalman filter

endcase

5 Conclusions

In this paper, a comparison study between the Kalman filter and the Lainiotis filter was presented. It was pointed out that the two filters are equivalent with respect to their behavior: they calculate the same estimates. The calculation burdens of the filtering algorithms were determined. A method was developed to a-priori decide which filter is the faster one: for time varying systems, the Kalman filter is faster than the Lainiotis filter; for time invariant systems, the Lainiotis filter is faster than the Kalman filter, when the state vector dimension is less than twice the measurement vector dimension, i.e. when $m > 0.74n$, for the steady state case, both the Kalman filter and the Lainiotis filter have equal computational times. This result is very important due to the fact that, in most real-time applications, it is essential to obtain the estimate in the shortest possible time.

References

[1] B.D.O. Anderson and J.B. Moore, *Optimal Filtering,* Prentice Hall inc., 1979.

[2] R.E. Kalman, A new approach to linear filtering and prediction problems, *J. Bas. Eng., Trans. ASME, ser. D,* **82**, no. 1 (1960), 34-45.

[3] D.G. Lainiotis, Partitioned linear estimation algorithms: Discrete case, *IEEE Trans. on AC,* **AC-20** (1975), 255-257.

[4] D.G. Lainiotis, N.D. Assimakis D. and S. K. Katsikas, Fast and numerically robust recursive algorithms for solving the discrete time Riccati equation: The case of nonsingular plant noise covariance matrix, *Neural Parallel and Scientific Computations,* **3**, no. 4 (1995), 565-583.

[5] G. Welch and G. Bishop, *An Introduction to the Kalman filter,* UNC-Chapel Hill, TR 95-041 2001, In Computer Graphics, Annual Conference on Computer Graphics Interactive Techniques. ACM Press, Addison-Wesley, Los Angeles, CA, USA, SIGGRAPH 2001, course pack edition, http://www.cs.unc.edu/∼welch, http://www.cs.unc.edu/∼gb

Appendices

A Equivalence of Kalman and Lainiotis filters

It will be proved that the Kalman and Lainiotis filters calculate theoretically the same estimates as well as the same estimation error covariances for time varying, time invariant and steady state systems.

The following proposition has to be proved: If the estimate and the estimation error covariances at time k calculated by implementing Kalman filter equal to those calculated by implementing Lainiotis filter, then the estimate and the estimation error covariances at time $k+1$ calculated by implementing Kalman filter equal to those calculated by implementing Lainiotis filter:

$$x_{Kf}(k/k) = x_{Lf}(k/k) \implies x_{Kf}(k+1/k+1) = x_{Lf}(k+1/k+1) \tag{A.1}$$
$$P_{Kf}(k/k) = P_{Lf}(k/k) \implies P_{Kf}(k+1/k+1) = P_{Lf}(k+1/k+1) \tag{A.2}$$

beginning with the same initial conditions:

$$x_{Kf}(0/0) = x_{Lf}(0/0) = x_0 \tag{A.3}$$
$$P_{Kf}(0/0) = P_{Lf}(0/0) = P_0 \tag{A.4}$$

A.1 Time varying filters

The proof that follows assumes, without loss of generality, that the covariance matrices $Q(k)$, $R(k+1)$ and $P(0/0)$ are positive definite; then the existence of all inverse matrices can be verified. We use the Kalman filter, the Lainiotis filter equations, and the Matrix Inversion Lemma (MIL):

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1} \tag{A.5}$$

Substituting (13) in (15), we have

$$\begin{aligned} P_n(k+1,k) &= Q(k) - K_n(k+1)H(k+1)Q(k) \\ &= Q(k) - \left[Q(k)H^T(k+1)\right] A(k+1) \left[H(k+1)Q(k)\right] \end{aligned}$$

and applying the Matrix Inversion Lemma in the last relationship we take

$$\begin{aligned} &P_n^{-1}(k+1,k) \\ &= Q^{-1}(k) + H^T(k+1) \\ &\qquad \left[A^{-1}(k+1) + H(k+1)Q(k)Q^{-1}(k)(-Q(k)H^T(k+1))\right]^{-1} H(k+1) \\ &= Q^{-1}(k) + H^T(k+1)\left[A^{-1}(k+1) - H(k+1)Q(k)H^T(k+1)\right]^{-1} H(k+1), \end{aligned}$$

and due to (12) $A^{-1}(k+1) - H(k+1)Q(k)H^T(k+1) = R(k+1)$, thus, we conclude that:

$$P_n^{-1}(k+1,k) = Q^{-1}(k) + H^T(k+1)R^{-1}(k+1)H(k+1) \tag{A.6}$$

Obviously, $Q(k), R(k+1)$ are symmetric matrices and due to (A.6) we conclude that the matrix $P_n^{-1}(k+1,k)$ is symmetric, since

$$\begin{aligned}&\left[P_n^{-1}(k+1,k)\right]^T \\ &\quad = Q^{-T}(k) + H^T(k+1)R^{-T}(k+1)H(k+1) = P_n^{-1}(k+1,k).\end{aligned} \tag{A.7}$$

By (15) and (16) we take :

$$\begin{aligned}&P_n^{-1}(k+1,k)F_n(k+1,k) \\ &= [Q(k) - K_n(k+1)H(k+1)Q(k)]^{-1} \\ &\qquad\qquad [F(k+1,k) - K_n(k+1)H(k+1)F(k+1,k)] \\ &= \left[(I - K_n(k+1)H(k+1))\,Q(k)\right]^{-1}\left[(I - K_n(k+1)H(k+1))\,F(k+1,k)\right] \\ &= Q^{-1}(k)\left[I - K_n(k+1)H(k+1)\right]^{-1}\left[I - K_n(k+1)H(k+1)\right]F(k+1,k) \\ &= Q^{-1}(k)F(k+1,k)\end{aligned} \tag{A.8}$$

Combining the relationships (A.7) and (A.8) arises :

$$\begin{aligned}F_n^T(k+1,k)P_n^{-1}(k+1,k) &= F_n^T(k+1,k)\left[P_n^{-1}(k+1,k)\right]^T \\ &= \left[P_n^{-1}(k+1,k)F(k+1,k)\right]^T \\ &= \left[Q^{-1}(k)F(k+1,k)\right]^T \\ &= F^T(k+1,k)Q^{-1}(k)\end{aligned} \tag{A.9}$$

From (A.9), (A.8) and the equations (17), (14),(15) and (13), we conclude

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

$$
\begin{aligned}
&+ Q^{-1}(k)\left[Q(k) - Q(k)H^T(k+1)A(k+1)H(k+1)Q(k)\right]Q^{-1}(k)]F(k+1,k)\\
&= F^T(k+1,k)[H^T(k+1)A(k+1)H(k+1)\\
&\quad + Q^{-1}(k)Q(k)\left[I - H^T(k+1)A(k+1)H(k+1)Q(k)\right]Q^{-1}(k)]F(k+1,k)\\
&= F^T(k+1,k)[H^T(k+1)A(k+1)H(k+1)\\
&\quad + Q^{-1}(k) - H^T(k+1)A(k+1)H(k+1)]F(k+1,k)\\
&= F^T(k+1,k)Q^{-1}(k)F(k+1,k).
\end{aligned}
$$

Thus, it is proved the relationship

$$
\begin{aligned}
&O_n(k+1) + F_n^T(k+1,k)P_n^{-1}(k+1,k)F_n(k+1,k)\\
&\quad = F^T(k+1,k)Q^{-1}(k)F(k+1,k).
\end{aligned}
\tag{A.10}
$$

It is obvious that

$$
\begin{aligned}
&[I + P(k/k)O_n(k+1)]^{-1}P(k/k)\\
&\quad = \left[P(k/k)P^{-1}(k/k) + P(k/k)O_n(k+1)\right]^{-1}P(k/k)\\
&\quad = \left[P(k/k)\left(P^{-1}(k/k) + O_n(k+1)\right)\right]^{-1}P(k/k)\\
&\quad = \left[P^{-1}(k/k) + O_n(k+1)\right]^{-1}.
\end{aligned}
\tag{A.11}
$$

By (11) and (A.11) we have

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

whereby, using the assumption that $Q(k), P_n(k+1,k)$ are nonsingular matrices and (A.5), we compute the inverse matrix $P^{-1}(k+1/k+1)$:

$$
\begin{aligned}
&P^{-1}(k+1/k+1)\\
&= \{P_n(k+1,k) + F_n(k+1,k)\left[P^{-1}(k/k) + O_n(k+1)\right]^{-1}F_n^T(k+1,k)\}^{-1}\\
&= P_n^{-1}(k+1,k) - P_n^{-1}(k+1,k)F_n(k+1,k)\\
&\qquad \left[P^{-1}(k/k) + O_n(k+1) + F_n^T(k+1,k)P_n^{-1}(k+1,k)F_n(k+1,k)\right]^{-1}\\
&\qquad\qquad F_n^T(k+1,k)P_n^{-1}(k+1,k).
\end{aligned}
\tag{A.12}
$$

By (A.12) we take $P(k+1/k+1),$ in which we substitute (A.6), (A.8), (A.10), (A.9):

$$
\begin{aligned}
&P(k+1/k+1)\\
&= \{P_n^{-1}(k+1,k) - P_n^{-1}(k+1,k)F_n(k+1,k)\\
&\qquad \left[P^{-1}(k/k) + O_n(k+1) + F_n^T(k+1,k)P_n^{-1}(k+1,k)F_n(k+1,k)\right]^{-1}\\
&\qquad\qquad F_n^T(k+1,k)P_n^{-1}(k+1,k)\}^{-1}\\
&= \{Q^{-1}(k) + H^T(k+1)R^{-1}(k+1)H(k+1)\\
&\qquad - Q^{-1}(k)F(k+1,k)\left[\,P^{-1}(k/k) + F^T(k+1,k)Q^{-1}(k)F(k+1,k)\,\right]^{-1}\\
&\qquad\qquad F^T(k+1,k)Q^{-1}(k)\}^{-1}
\end{aligned}
\tag{A.13}
$$

Moreover, the nonsingularity of matrices $Q(k)$, $P(k+1/k)$, $P(k/k)$, using (A.5) in (4), they allow us write :

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

Finally for the Kalman filter, after the substitution of the equation (5) in (7), the use of the Matrix Inversion Lemma and the substitution of the matrix $P^{-1}(k+1/k)$ from the last equality, we conclude:

$$\begin{aligned}&P(k+1/k+1) \qquad\qquad (A.14)\\ &= P(k+1/k)\\ &\quad - P(k+1/k)H^T(k+1)\left[H(k+1)P(k+1/k)H^T(k+1)+R(k+1)\right]^{-1}\\ &\qquad\qquad H(k+1)P(k+1/k)\\ &= \left[P^{-1}(k+1/k)\right]^{-1}-\left[P^{-1}(k+1/k)\right]^{-1}H^T(k+1)\\ &\qquad \left[(R^{-1}(k+1))^{-1}+H(k+1)(P^{-1}(k+1/k))^{-1}H^T(k+1)\right]^{-1}\\ &\qquad\qquad H(k+1)\left[P^{-1}(k+1/k)\right]^{-1}\\ &= \{P^{-1}(k+1/k)+H^T(k+1)R^{-1}(k+1)H(k+1)\}^{-1}\\ &= \{H^T(k+1)R^{-1}(k+1)H(k+1)+Q^{-1}(k)\\ &\quad - Q^{-1}(k)F(k+1,k)\left[\,P^{-1}(k/k)+F^T(k+1,k)Q^{-1}(k)F(k+1,k)\,\right]^{-1}\\ &\qquad\qquad F^T(k+1,k)Q^{-1}(k)\}^{-1}\end{aligned}$$

Comparing the equalities (A.13) and (A.14), it is obvious that (A.2) holds, i.e., Kalman and Lainiotis filters calculate theoretically the same values for $P(k+1/k+1)$.
The equation (6) can be written

$$x(k+1/k+1)=\left[I-K(k+1)H(k+1)\right]x(k+1/k)+K(k+1)z(k+1)$$

and by (3) the equivalent equality is written

$$\begin{aligned}&x(k+1/k+1) \qquad\qquad (A.15)\\ &\quad =\left[I-K(k+1)H(k+1)\right]F(k+1,k)x(k/k)+K(k+1)z(k+1).\end{aligned}$$

By (8), (9), (13), (14) and (A.11) the equation (10) is formed

$$\begin{aligned}&x(k+1/k+1) \qquad\qquad (A.16)\\ &= \; x_n(k+1/k+1)+F_n(k+1,k)\left[I+P(k/k)O_n(k+1)\right]^{-1}\\ &\qquad P(k/k)M_n(k+1)+F_n(k+1,k)\left[I+P(k/k)O_n(k+1)\right]^{-1}x(k/k)\end{aligned}$$

$$
\begin{aligned}
&= K_n(k+1)z(k+1) \\
&\quad + F_n(k+1,k)\left[I + P(k/k)O_n(k+1)\right]^{-1} P(k/k)K_m(k+1)z(k+1) \\
&\quad + F_n(k+1,k)\left[I + P(k/k)O_n(k+1)\right]^{-1} x(k/k) \\
&= \left[K_n(k+1) + F_n(k+1,k)\left[I + P(k/k)O_n(k+1)\right]^{-1} P(k/k)K_m(k+1)\right] \\
&\qquad z(k+1) + F_n(k+1,k)\left[I + P(k/k)O_n(k+1)\right]^{-1} x(k/k) \\
&= \left[Q(k) + F_n(k+1,k)\left[I + P(k/k) + O_n(k+1)\right]^{-1} F^T(k+1,k)\right] \\
&\qquad\qquad H^T(k+1)A(k+1)z(k+1) \\
&\quad + F_n(k+1,k)\left[I + P(k/k)O_n(k+1)\right]^{-1} x(k/k) \\
&= F_n(k+1,k)\left[I + P(k/k)O_n(k+1)\right]^{-1} x(k/k) \\
&\quad + \left[Q(k) + F_n(k+1,k)\left[P^{-1}(k/k) + O_n(k+1)\right]^{-1} F^T(k+1,k)\right] \\
&\qquad\qquad H^T(k+1)A(k+1)z(k+1)
\end{aligned}
$$

Comparing the two filters due to (A.15) and (A.16), it is clear that (A.1) is equivalent with the force of the following equations :

$$
\begin{aligned}
&\left[I - K(k+1)H(k+1)\right]F(k+1,k) \\
&\qquad = F_n(k+1,k)\left[I + P(k/k)O_n(k+1)\right]^{-1}
\end{aligned}
\tag{A.17}
$$

and

$$
\begin{aligned}
&K(k+1) \\
&\quad = \Big(Q(k) + F_n(k+1,k)\left[P^{-1}(k/k) + O_n(k+1)\right]^{-1} F^T(k+1,k)\Big) \\
&\qquad \left(H^T(k+1)A(k+1)\right)
\end{aligned}
\tag{A.18}
$$

From (5) arises

$$
\begin{aligned}
&P(k+1/k)H^T(k+1) \\
&\quad = K(k+1)\left[H(k+1)P(k+1/k)H^T(k+1) + R(k+1)\right],
\end{aligned}
\tag{A.19}
$$

and by (7) we have

$$
\begin{aligned}
P(k/k) &= P(k/k-1) - K(k)H(k)P(k/k-1) \\
&= \left[I - K(k)H(k)\right]P(k/k-1).
\end{aligned}
\tag{A.20}
$$

Using (A.2) and substituting (A.20) in (4) we take

$$
\begin{aligned}
&P(k+1/k) - Q(k) = F(k+1,k)P(k/k)F^T(k+1,k) \\
&\quad = F(k+1,k)\left[I - K(k)H(k)\right]P(k/k-1)F^T(k+1,k).
\end{aligned}
\tag{A.21}
$$

By (17) and (14) yields

$$
\begin{aligned}
&O_n(k+1) = K_m(k+1)H(k+1)F(k+1,k) \\
&\qquad = F^T(k+1,k)H^T(k+1)A(k+1)H(k+1)F(k+1,k).
\end{aligned}
\tag{A.22}
$$

Using (A.21), (A.22) and (A.20) we are able to write

$$\begin{aligned}
&\left[I + [P(k+1/k) - Q(k)]\, H^T(k+1)A(k+1)H(k+1)\right] F(k+1,k) \\
&\quad = F(k+1,k) + [P(k+1/k) - Q(k)]\, H^T(k+1) \\
&\qquad\qquad A(k+1)H(k+1)F(k+1,k) \\
&\quad = F(k+1,k) + F(k+1,k)\,[I - K(k)H(k)]\, P(k/k-1)F^T(k+1,k) \\
&\qquad\qquad H^T(k+1)A(k+1)H(k+1)F(k+1,k) \\
&\quad = F(k+1,k) + F(k+1,k)\,[I - K(k)H(k)]\, P(k/k-1)O_n(k+1) \\
&\quad = F(k+1,k) + F(k+1,k)P(k/k)O_n(k+1) \\
&\quad = F(k+1,k)\,[\,I + P(k/k)O_n(k+1)\,]\,.
\end{aligned}$$

Furthermore, the matrices $I+[P(k+1/k) - Q(k)]\, H^T(k+1)A(k+1)H(k+1)$, $I + P(k/k)O_n(k+1)$ are positive definite thus, the above equality is implied :

$$\begin{aligned}
&F(k+1,k)\,[\,I + P(k/k)O_n(k+1)\,]^{-1} \qquad (A.23)\\
&\quad = \left[I + [P(k+1/k) - Q(k)]\, H^T(k+1)A(k+1)H(k+1)\right]^{-1} F(k+1,k)
\end{aligned}$$

By (16), (13) and (A.23), we have

$$\begin{aligned}
&F_n(k+1,k)\,[I + P(k/k)O_n(k+1)]^{-1} \\
&\quad = [F(k+1,k) - K_n(k+1)H(k+1)F(k+1,k)]\,[I + P(k/k)O_n(k+1)]^{-1} \\
&\quad = [I - K_n(k+1)H(k+1)]\,F(k+1,k)\,[I + P(k/k)O_n(k+1)]^{-1} \\
&\quad = \left[I - Q(k)H^T(k+1)A(k+1)H(k+1)\right] \\
&\qquad F(k+1,k)\,[I + P(k/k)O_n(k+1)]^{-1} \\
&\quad = \left[I - Q(k)H^T(k+1)A(k+1)H(k+1)\right] \\
&\qquad \left[I + [P(k+1/k) - Q(k)]\, H^T(k+1)A(k+1)H(k+1)\right]^{-1} F(k+1,k) \\
&\quad = \left[I - Q(k)H^T(k+1)A(k+1)H(k+1)\right] \\
&\qquad \left[I + [-Q(k) + P(k+1/k)]\, H^T(k+1)A(k+1)H(k+1)\right]^{-1} F(k+1,k) \\
&\quad = B\,[B+V]^{-1}\,F(k+1,k), \qquad (A.24)
\end{aligned}$$

where $B \equiv I - Q(k)H^T(k+1)A(k+1)H(k+1)$, and $V \equiv P(k+1/k)H^T(k+1)A(k+1)H(k+1)$.
Note that, B and $I - K(k+1)H(k+1)$ are positive definite matrices, consequently, using (A.19) after manipulations we can take:

$$\begin{aligned}
&B\,[B+V]^{-1} \qquad (A.25)\\
&\quad = \left(B^{-1}\right)^{-1}[B+V]^{-1} = \left[(B+V)\,B^{-1}\right]^{-1} = \left[I + VB^{-1}\right]^{-1} \\
&\quad = \left[I + VB^{-1}\right]^{-1}[I - K(k+1)H(k+1)]^{-1}\,[I - K(k+1)H(k+1)] \\
&\quad = \left[[I - K(k+1)H(k+1)]\left(I + VB^{-1}\right)\right]^{-1}[I - K(k+1)H(k+1)]
\end{aligned}$$

$$
\begin{aligned}
&= \{I - K(k+1)H(k+1) + P(k+1/k)H^T(k+1)A(k+1)H(k+1)B^{-1} \\
&\quad - K(k+1)H(k+1)P(k+1/k)H^T(k+1)A(k+1)H(k+1)B^{-1}\}^{-1} \\
&\qquad\qquad [I - K(k+1)H(k+1)] \\
&= \{I - K(k+1)H(k+1) \\
&\quad + \left[P(k+1/k)H^T(k+1) - K(k+1)H(k+1)P(k+1/k)H^T(k+1)\right] \\
&\qquad A(k+1)H(k+1)B^{-1}\}^{-1} [I - K(k+1)H(k+1)] \\
&= \{I - K(k+1)H(k+1) + K(k+1)R(k+1)A(k+1)H(k+1)B^{-1}\}^{-1} \\
&\qquad\qquad [I - K(k+1)H(k+1)]
\end{aligned}
$$

Furthermore, $I - K(k+1)H(k+1) + K(k+1)R(k+1)A(k+1)H(k+1)B^{-1}$ is a matrix equivalent with the identity matrix, as it is proved in the following

$$
\begin{aligned}
&I - K(k+1)H(k+1)BB^{-1} + K(k+1)R(k+1)A(k+1)H(k+1)B^{-1} \\
&\quad = I + K(k+1)\{-H(k+1)B + R(k+1)A(k+1)H(k+1)\}B^{-1} \\
&\quad = I + K(k+1)\{-H(k+1)\left[I - Q(k)H^T(k+1)A(k+1)H(k+1)\right] \\
&\qquad\qquad + R(k+1)A(k+1)H(k+1)\}B^{-1} \\
&\quad = I + K(k+1)\ \{-H(k+1) + H(k+1)Q(k)H^T(k+1) \\
&\qquad A(k+1)H(k+1) + R(k+1)A(k+1)H(k+1)\}B^{-1} \\
&\quad = I + K(k+1)\{-H(k+1) + \left[A^{-1}(k+1) - R(k+1)\right] \\
&\qquad A(k+1)H(k+1) + R(k+1)A(k+1)H(k+1)\}B^{-1} \\
&\quad = I + K(k+1)\{-H(k+1) + A^{-1}(k+1)A(k+1)H(k+1) \\
&\qquad - R(k+1)A(k+1)H(k+1) + R(k+1)A(k+1)H(k+1)\}B^{-1} \\
&\quad = I,
\end{aligned}
$$

note that from (12), $A(k+1)$ is a nonsingular matrix and

$$A^{-1}(k+1) - R(k+1) = H(k+1)Q(k)H^T(k+1).$$

Consequently, (A.25) can be written

$$B\left[B + V\right]^{-1} = I - K(k+1)H(k+1),$$

and substituting the above equality in (A.24), the proof of (A.17) is complete. By (4) the following equality occurs :

$$
\begin{aligned}
&H(k+1)Q(k)H^T(k+1) \qquad\qquad (A.26)\\
&\quad = H(k+1)P(k+1/k)H^T(k+1) \\
&\qquad - H(k+1)F(k+1,k)P(k/k)F^T(k+1,k)H^T(k+1)
\end{aligned}
$$

We obtain (A.18), if we use the equations (A.11), (A.17), (4), (A.19), (A.26), (12) sequentially, as it is described in the following:

$$
\begin{aligned}
&\{Q(k)+F_n(k+1,k)\left[P^{-1}(k/k)+O_n(k+1)\right]^{-1}F^T(k+1,k)\}\\
&\qquad\qquad H^T(k+1)A(k+1)\\
&=\{Q(k)+F_n(k+1,k)\left[I+P(k/k)O_n(k+1)\right]^{-1}P(k/k)F^T(k+1,k)\}\\
&\qquad\qquad H^T(k+1)A(k+1)\\
&=\{Q(k)+\left[I-K(k+1)H(k+1)\right]F(k+1,k)P(k/k)F^T(k+1,k)\}\\
&\qquad\qquad H^T(k+1)A(k+1)\\
&=\{[Q(k)+F(k+1,k)P(k/k)F^T(k+1,k)]\\
&-K(k+1)H(k+1)F(k+1,k)P(k/k)F^T(k+1,k)\}H^T(k+1)A(k+1)\\
&=\{P(k+1/k)H^T(k+1)\\
&-K(k+1)H(k+1)F(k+1,k)P(k/k)F^T(k+1,k)H^T(k+1)\}A(k+1)\\
&=\{K(k+1)H(k+1)P(k+1/k)H^T(k+1)+K(k+1)R(k+1)\\
&-K(k+1)H(k+1)F(k+1,k)P(k/k)F^T(k+1,k)H^T(k+1)\}A(k+1)\\
&=K(k+1)\{H(k+1)P(k+1/k)H^T(k+1)+R(k+1)\\
&\qquad-H(k+1)F(k+1,k)P(k/k)F^T(k+1,k)H^T(k+1)\}A(k+1)\\
&=K(k+1)\{H(k+1)Q(k)H^T(k+1)+R(k+1)\}A(k+1)\\
&=K(k+1)A^{-1}(k+1)A(k+1)\\
&=K(k+1)
\end{aligned}
$$

A.2 Time invariant filters

The proof is analogous to the proof concerning the time varying filters.

A.3 Steady state filters

The Kalman and Lainiotis filters calculate the same estimates as well as the same estimation error covariances for time invariant systems. Thus, they calculate the same steady state values $\overline{P_e}$ of the estimation error covariance matrix. Analogously as in proof of the time varying filters, we can prove the corresponding equality to (A.17) for $\overline{P_e}$ and $\overline{K}$ as in (36), which due to (34) and (38) leads to

$$\left[I-\overline{K}H\right]F=F_n\left[I+\overline{P_e}O_n\right]^{-1}\Rightarrow A_{Kf}=A_{Lf}.$$

Furthermore, if we follow the same steps of the proof for the equality (A.18), we conclude

$$\overline{K}=QH^TA+F_n\left[I+\overline{P_e}O_n\right]^{-1}\overline{P_e}F^TH^TA=K_n+F_n\left[I+\overline{P_e}O_n\right]^{-1}\overline{P_e}K_m,$$

and due to (34) and (38), it is clear that $B_{Kf}=B_{Lf}$.

B Calculation burden of matrix operations

Basic assumptions The calculation burden of matrix operations depends on the matrices' dimensions and involves scalar operations (additions, multiplications and divisions) the calculation burdens of which are assumed to be equal.

Matrix Addition

Matrix Operation	Scalar Additions
$(n \times m) + (n \times m)$	nm
$(n \times n) + (n \times n) = \text{symmetric}$	$\frac{n^2 - n}{2} + n = \frac{n^2 + n}{2}$
$I_{n \times n} + (n \times n) \quad I:$ Identity	n

Matrix Multiplication

Matrix Operation	Scalar Additions	Scalar Multiplications	Total
$(n \times m) \cdot (m \times k)$	$n(m-1)k$	nmk	$2nmk - nk$
$(n \times m) \cdot (m \times n)^*$	$\frac{n^2 + n}{2} m$	$\frac{n^2 + n}{2} (m-1)$	$n^2 m + nm - \frac{n^2 + n}{2}$

* The matrix is symmetric.

Matrix Inversion using LU Decomposition

Let $A = LU$ be an LU Decomposition (LUD) of an $n \times n$ nonsingular matrix A, which is described by the following algorithm :

LU Decomposition (LUD)

for $k := 1$ to $n - 1$

for $j := k + 1$ to n

$A(k, j) = A(k, j)/A(k, k)$

for $i := k + 1$ to n

$A(i, j) = A(i, j) - A(i, k) * A(k, j)$

LUD	Scalar Additions	Scalar Multiplications	Scalar Divisions
$k=1$ $j=2,\ldots,n$ $i=2,\ldots,n$	$(n-1)^2$	$(n-1)^2$	$n-1$
$k=2$ $j=3,\ldots,n$ $i=3,\ldots,n$	$(n-2)^2$	$(n-2)^2$	$n-2$
$\ldots$	$\ldots$	$\ldots$	$\ldots$
$k=n-1$ $j=n$ $i=n$	1	1	1
total	$\sum_{i=1}^{n-1} i^2$	$\sum_{i=1}^{n-1} i^2$	$\sum_{i=1}^{n-1} i$

$$2\sum_{i=1}^{n-1} i^2 + \sum_{i=1}^{n-1} i = 2\frac{(n-1)n(2n-1)}{6} + \frac{n(n-1)}{2} = \frac{n(n-1)(4n+1)}{6}$$

Having found an LU Decomposition for A we are able to solve equations of the form $Ax = b$ by solving the triangular linear systems $Ly = b$ and $Ux = y$:

$Ly = b$ (forward substitution)

$$y_1 = \frac{b_1}{\ell_{11}}, \quad y_i = \frac{1}{\ell_{ii}}\left(b_i - \sum_{j=1}^{i-1} \ell_{ij}\, y_j\right), \quad i = 2,3,\ldots,n$$

Forward substitution	Scalar Additions	Scalar Multiplications	Scalar Divisions
y_1	0	0	1
y_2 $i=2$ $j=1$	1	1	1
y_3 $i=3$ $j=1,2$	2	2	1
$\ldots$	$\ldots$	$\ldots$	$\ldots$
y_n $i=n$ $j=1,\cdots,n-1$	$n-1$	$n-1$	1
total	$\sum_{i=1}^{n-1} i$	$\sum_{i=1}^{n-1} i$	$\sum_{i=1}^{n} 1$

$$2\sum_{i=1}^{n-1} i + \sum_{i=1}^{n} 1 = 2\,\frac{n(n-1)}{2} + n = n^2$$

$Ux = y$ (backward substitution)

$$x_n = \frac{y_n}{u_{nn}}, \quad x_i = \frac{1}{u_{ii}} \left(y_i - \sum_{j=i+1}^{n} u_{ij}\, x_j \right), \quad i = n-1, \ldots, 2, 1$$

Backward substitution	Scalar Additions	Scalar Multiplications	Scalar Divisions
x_n	0	0	1
x_{n-1} $i = n-1$ $j = n$	1	1	1
x_{n-2} $i = n-2$ $j = n-1, n$	2	2	1
$\ldots$	$\ldots$	$\ldots$	$\ldots$
x_2 $i = 2$ $j = 2, \cdots, n$	$n-1$	$n-1$	1
total	$\sum_{i=1}^{n-1} i$	$\sum_{i=1}^{n-1} i$	$\sum_{i=1}^{n} 1$

$$2\sum_{i=1}^{n-1} i + \sum_{i=1}^{n} 1 = 2\,\frac{n(n-1)}{2} + n = n^2$$

Thus, the vector x is the solution of the equation $Ax = b$ because $Ax = LUx = Ly = b$. We are now able to compute the inversion of matrix A using the LU Decomposition of matrix A. The equation $AX = I_n$ can be viewed as a set of n distinct equations of the form $Ax = b$. These equations define the matrix X as the inverse of A, which denotes $X = A^{-1}$. Let X_i denote the $i-$th column of X and e_i the $i-$th column of I_n. Then, the equation $AX = I_n$ can be solved for X by using the LU Decomposition for A to solve each from the n equations $AX_i = e_i$ separately for X_i. The calculation burden of the matrix inversion using LU Decomposition has as follows:

Matrix Inversion	Scalar Additions	Scalar Multiplications	Scalar Divisions
LUD	$\frac{(n-1)n(2n-1)}{6}$	$\frac{(n-1)n(2n-1)}{6}$	$\frac{n(n-1)}{2}$
forward substitution n times	$\frac{n(n-1)}{2}n$	$\frac{n(n-1)}{2}n$	nn
backward substitution n times	$\frac{n(n-1)}{2}n$	$\frac{n(n-1)}{2}n$	nn
total	$\frac{n(n-1)(8n-1)}{6}$	$\frac{n(n-1)(8n-1)}{6}$	$\frac{n(5n-1)}{2}$

$$2\frac{n(n-1)(8n-1)}{6} + \frac{n(5n-1)}{2} = \frac{16n^3 - 3n^2 - n}{6}$$

C Calculation burden of the filters

Time varying filters

The Time Varying Kalman Filter (TVKF) consists of the implementation of equations (3)-(7):

Time Varying Kalman Filter (TVKF)		
Matrix Operation	Matrix Dimensions	Calculation Burden
$x(k+1/k) = F(k+1,k)x(k/k)$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
$Y_1 \equiv F(k+1,k)P(k/k)$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$Y_2 \equiv Y_1 F^T(k+1,k)$	$(n \times n) \cdot (n \times n)$ [1†]	$n^3 + 0.5n^2 - 0.5n$
$P(k+1/k) = Y_2 + Q(k)$	$(n \times n) + (n \times n)$ [2†]	$0.5n^2 + 0.5n$
$Y_3 \equiv H(k+1)P(k+1/k)$	$(m \times n) \cdot (n \times n)$	$2n^2m - nm$
$Y_4 \equiv Y_3 H^T(k+1)$	$(m \times n) \cdot (n \times m)$ [3†]	$nm^2 + nm - 0.5m^2 - 0.5m$
$Y_5 \equiv Y_4 + R(k+1)$	$(m \times m) + (m \times m)$ [4†]	$0.5m^2 + 0.5m$
$[Y_5]^{-1}$	$(m \times m)^{-1}$	$(16m^3 - 3m^2 - m)/6$
$K(k+1) = Y_3^T [Y_5]^{-1}$	$(n \times m) \cdot (m \times m)$	$2nm^2 - nm$
$Y_6 \equiv K(k+1)H(k+1)$	$(n \times m) \cdot (m \times n)$	$2n^2m - n^2$
$Y_7 \equiv I - Y_6$	$I_{n\times n} + (n \times n)$	n
$Y_8 \equiv Y_7\, x(k+1/k)$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
$Y_9 \equiv K(k+1)z(k+1)$	$(n \times m) \cdot (m \times 1)$	$2nm - n$
$x(k+1/k+1) = Y_8 + Y_9$	$(n \times 1) + (n \times 1)$	n
$P(k+1/k+1) = Y_7 P(k+1/k)$	$(n \times n) \cdot (n \times n)$ [5†]	$n^3 + 0.5n^2 - 0.5n$
$\mathrm{CB}_{TVKF} = 4n^3 + 3.5n^2 - 1.5n + 4n^2m + nm + 3nm^2 + (16\mathrm{m}^3 - 3m^2 - m)/6$		

† symmetric matrix

[1†] $Y_2^T = \left[Y_1 F^T(k+1,k)\right]^T = F(k+1,k)Y_1^T = F(k+1,k)P^T(k/k)F^T(k+1,k)$
$= Y_1 F^T(k+1,k) = Y_2$, since $P^T(k/k) = P(k/k)$.

[2†] Since $Q^T(k) = Q(k)$, and $Y_2^T = Y_2$.

[3†] $Y_4^T = \left[Y_3 H^T(k+1)\right]^T = H(k+1)Y_3^T = H(k+1)P^T(k+1/k)H^T(k+1)$
$= Y_3 H^T(k+1) = Y_4$, due to (4), $P(k+1/k)$ is a covariance matrix.

[4†] Since $R^T(k+1) = R(k+1)$, and $Y_4^T = Y_4$.

[5†] $P^T(k+1/k+1) = [Y_7 P(k+1/k)]^T = P^T(k+1/k)Y_7^T = P(k+1/k)\,[I - Y_6]^T$
$= P(k+1/k)\left[I - Y_6^T\right] = P(k+1/k)\left[I - H^T(k+1)K^T(k+1)\right]$
$= P(k+1/k) - P(k+1/k)H^T(k+1)Y_5^{-T}(k+1)Y_3$
$= P(k+1/k) - P(k+1/k)H^T(k+1)Y_5^{-1}(k+1)H(k+1)P(k+1/k)$
$= P(k+1/k) - K(k+1)H(k+1)P(k+1/k) = [I - K(k+1)H(k+1)]\,P(k+1/k)$
$= P(k+1/k+1)$.

The Time Varying Lainiotis Filter (TVLF) consists of the implementation of equations (8)-(17); equations (12)-(17) can be seen as an initialization process required for the implementation of equations (8)-(11):

Time Varying Lainiotis Filter (TVLF)		
equations (12) - (17)		
Matrix Operation	Matrix Dimensions	Calculation Burden
$W_1 \equiv H(k+1)Q(k)$	$(m \times n) \cdot (n \times n)$	$2n^2m - nm$
$W_2 \equiv W_1 H^T(k+1)$	$(m \times n) \cdot (n \times m)$ [1‡]	$nm^2 + nm$ $-0.5m^2 - 0.5m$
$W_2 + R(k+1)$	$(m \times m) + (m \times m)$ [2‡]	$0.5m^2 + 0.5m$
$A(k+1) = [W_2 + R(k+1)]^{-1}$	$(m \times m)^{-1}$	$\frac{16m^3 - 3m^2 - m}{6}$
$K_n(k+1)$ $= [H(k+1)Q(k)]^T A(k+1)$ $= W_1^T A(k+1)$	$(n \times m) \cdot (m \times m)$	$2nm^2 - nm$
$W_3 \equiv H(k+1)F(k+1,k)$	$(m \times n) \cdot (n \times n)$	$2n^2m - nm$
$K_m(k+1)$ $=$[H(k+1)F(k+1,k)]$^T A(k+1)$ $=$W$_3^T A(k+1)$	$(n \times m) \cdot (m \times m)$	$2nm^2 - nm$
$W_4 \equiv K_n(k+1)H(k+1)Q(k)$ $= K_n(k+1)W_1$	$(n \times m) \cdot (m \times n)$ [3‡]	$n^2m + nm$ $-0.5n^2 - 0.5n$
$P_n(k+1,k) = Q(k) - W_4$	$(n \times n) + (n \times n)$ [4‡]	$0.5n^2 + 0.5n$
W$_5 \equiv$ $K_n(k+1)H(k+1)F(k+1,k)$ $= K_n(k+1)W_3$	$(n \times m) \cdot (m \times n)$	$2n^2m - n^2$
$F_n(k+1,k) = F(k+1,k) - W_5$	$(n \times n) + (n \times n)$	n^2
$O_n(k+1) = K_m(k+1)W_3$	$(n \times m) \cdot (m \times n)$ [5‡]	$n^2m + nm$ $-0.5n^2 - 0.5n$
CB$_1 = 8n^2m + 5nm^2 - nm - 0.5n^2 - 0.5n + (16m^3 - 3m^2 - m)/6$		

‡ symmetric matrices

[1‡] $W_2^T = \left[W_1 H^T(k+1)\right]^T = H(k+1)W_1^T = H(k+1)\left[H(k+1)Q(k)\right]^T$
$= H(k+1)Q(k)H^T(k+1) = W_1 H^T(k+1) = W_2$, due to $Q^T(k) = Q(k)$.

[2‡] Since $R^T(k+1) = R(k+1)$, and $W_2^T = W_2$.

[3‡] $W_4^T = \left[K_n(k+1)W_1\right]^T = W_1^T K_n^T(k+1)$
$= \left[H(k+1)Q(k)\right]^T \left[Q(k)H^T(k+1)A(k+1)\right]^T$
$= Q(k)H^T(k+1)A(k+1)H(k+1)Q(k) = K_n(k+1)W_1 = W_4$, due to $A^T(k+1) = A(k+1)$.

[4‡] Since $Q^T(k) = Q(k)$, and $W_4^T = W_4$.

[5‡] $O_n^T(k+1) = \left[K_m(k+1)W_3\right]^T = W_3^T K_m^T(k+1)$
$= \left[H(k+1)F(k+1,k)\right]^T \left[W_3^T A(k+1)\right]^T = F^T(k+1,k)H^T(k+1)A(k+1)W_3$
$= K_m(k+1)W_3 = O_n(k+1)$.

equations (8)-(11)		
Matrix Operation	Matrix Dimensions	Calculation Burden
$x_n(k+1/k+1) = K_n(k+1)z(k+1)$	$(n \times m) \cdot (m \times 1)$	$2nm - n$
$M_n(k+1) = K_m(k+1)z(k+1)$	$(n \times m) \cdot (m \times 1)$	$2nm - n$
$P(k/k)O_n(k+1)$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$I + P(k/k)O_n(k+1)$	$I_{n\times n} + (n \times n)$	n
$\Omega_1 \equiv [I + P(k/k)O_n(k+1)]^{-1}$	$(n \times n)^{-1}$	$\frac{16n^3 - 3n^2 - n}{6}$
$\Omega_2 \equiv F_n(k+1,k)\Omega_1$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$P(k/k)M_n(k+1)$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
$\Omega_3 \equiv P(k/k)M_n(k+1) + x(k/k)$	$(n \times 1) + (n \times 1)$	n
$\Omega_2\Omega_3$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
$x(k+1/k+1) = x_n(k+1/k+1) + \Omega_2 \cdot \Omega_3$	$(n \times 1) + (n \times 1)$	n
$\Omega_4 \equiv \Omega_2 P(k/k)$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$\Omega_5 \equiv \Omega_4 F_n^T(k+1,k)$	$(n \times n) \cdot (n \times n)^{1\star}$	$n^3 + 0.5n^2 - 0.5n$
$P(k+1/k+1) = P_n(k+1/k) + \Omega_5$	$(n \times n) + (n \times n)^{2\star}$	$0.5n^2 + 0.5n$
$\mathrm{CB}_2 = 7n^3 + 2n^2 + 4nm - n + (16n^3 - 3n^2 - n)/6$ $= 4nm + (58n^3 + 9n^2 - 7n)/6$		
CB_{TVLF}=CB_1+CB_2 $= 8n^2m + 5nm^2 + 3nm + (58n^3 + 6n^2 - 10n + 16m^3 - 3m^2 - m)/6$		

* symmetric matrices

$^{1\star}\Omega_5^T = F_n(k+1,k)\Omega_4^T = F_n(k+1,k)P^T(k/k)\Omega_2^T = F_n(k+1,k)P^T(k/k)\Omega_1^T F_n^T(k+1,k)$
$= F_n(k+1,k)P^T(k/k)\left[I + P(k/k)O_n(k+1)\right]^{-T} F_n^T(k+1,k)$
$= F_n(k+1,k)\left[\left[I + P(k/k)O_n(k+1)\right]^{-1} P(k/k)\right]^T F_n^T(k+1,k)$
$= F_n(k+1,k)\left[P^{-1}(k/k) + O_n(k+1)\right]^{-T} F_n^T(k+1,k)$
$= F_n(k+1,k)\left[P^{-1}(k/k) + O_n(k+1)\right]^{-1} F_n^T(k+1,k),$
$= F_n(k+1,k)\left[\left[I + P(k/k)O_n(k+1)\right]^{-1} P(k/k)\right] F_n^T(k+1,k)$
$= F_n(k+1,k)\Omega_1 P(k/k)F_n^T(k+1,k)$
$= \Omega_2 P(k/k)F_n^T(k+1,k) = \Omega_4 F_n^T(k+1,k) = \Omega_5,$
since (A.11) holds and due to $5\ddagger$, $O_n^T(k+1) = O_n(k+1)$.

^{2*}By $4\ddagger$, and $\Omega_5^T = \Omega_5$.

Time invariant filters

The Time Invariant Kalman Filter (TIKF) consists of the implementation of equations (18)-(22). It is obvious, that the calculation burden required for the implementation of equations (18)-(22) is equal to the calculation burden required for the implementation of equations (3)-(7). Thus, the calculation burden of the Time Invariant Kalman Filter (TIKF) is equal to the calculation burden of the Time Varying Kalman Filter (TVKF).

The Time Invariant Lainiotis Filter (TILF) consists of the implementation of equations (23)-(26); the quantities in (27)-(32) are calculated off-line. It is obvious, that the calculation burden required for the implementation of equations (23)-(26) is equal to the calculation burden required for the implementation of equations (8)-(11).

Steady state filters

The Steady State Kalman Filter (SSKF) consists of the implementation of equation (33) and the Steady State Lainiotis Filter (SSLF) consists of the implementation of equation (37). It is obvious, that both the steady state algorithms possess the same calculation burden required for their implementation.

Steady State Kalman Filter (SSKF)		
Matrix Operation	Matrix Dimensions	Calculation Burden
$A_{Kf}x(k/k)$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
$B_{Kf}z(k+1)$	$(n \times m) \cdot (m \times 1)$	$2nm - n$
$x(k+1/k+1)$ $= A_{Kf}x(k/k) + B_{Kf}z(k+1)$	$(n \times 1) + (n \times 1)$	n
		$2n^2 + 2nm - n$

Steady State Lainiotis Filter (SSLF)		
Matrix Operation	Matrix Dimensions	Calculation Burden
$A_{Lf}x(k/k)$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
$B_{Lf}z(k+1)$	$(n \times m) \cdot (m \times 1)$	$2nm - n$
$x(k+1/k+1)$ $= A_{Lf}x(k/k) + B_{Lf}z(k+1)$	$(n \times 1) + (n \times 1)$	n
		$2n^2 + 2nm - n$

Received: February 23, 2007