

Journal of Computations & Modelling, vol.1, no.1, 2011, 115-130
ISSN: 1792-7625 (print), 1792-8850 (online)
International Scientific Press, 2011

Lainiotis filter implementation via Chandrasekhar type algorithm

Nicholas Assimakis¹ and Maria Adam²

Abstract

An implementation of the time invariant Lainiotis filter using a Chandrasekhar type algorithm is presented and compared to the classical one. The size of model determines which algorithm is faster; a method is proposed to a-priori decide, which implementation is faster. In the infinite measurement noise case, the proposed method is always faster than the classical one.

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

Keywords: Lainiotis filter, Chandrasekhar algorithm

1 Introduction

Estimation plays an important role in many fields of science. The estimation problem has been solved by means of a recursive algorithm based on Riccati

¹ Department of Electronics, Technological Educational Institute of Lamia, Lamia 35100, Greece, e-mail: assimakis@teilam.gr

² Department of Computer Science and Biomedical Informatics, University of Central Greece, Lamia 35100, Greece, e-mail: madam@ucg.gr

type difference equations. In the last decades, several authors have proposed faster algorithms to solve the estimation problems by substituting the Riccati equations by a set of Chandrasekhar type difference equations [1, 5, 7, 8, 9]. The discrete time Lainiotis filter [6] is a well known algorithm that solves the estimation/filtering problem. In this paper, we propose an implementation of the time invariant Lainiotis filter using Chandrasekhar type recursive algorithm to solve the estimation/filtering problem. It is established that the classical and the proposed implementations are equivalent with respect to their behavior. It is also developed a method to a-priori (before the Lainiotis filter's implementation) decide which implementation is faster. This 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.

The paper is organized as follows: In Section 2 the classical implementation of Lainiotis filter is presented. In Section 3 the Chandrasekhar type algorithm is presented and the proposed implementation of Lainiotis filter via Chandrasekhar type algorithm is introduced. In Section 4 the computational requirements both implementations of Lainiotis filter are established and comparisons are carried out. It is pointed out that the proposed implementation may be faster than the classical one. In addition, a rule is established in order to decide which implementation is faster.

2 Classical Implementation of Lainiotis filter

The estimation problem arises in linear estimation and is associated with time invariant systems described for $k \geq 0$ by the following state space equations:

$$\begin{aligned}x_{k+1} &= Fx_k + w_k \\z_k &= Hx_k + v_k\end{aligned}$$

where x_k is the $n \times 1$ state vector at time k , z_k is the $m \times 1$ measurement vector, F is the $n \times n$ system transition matrix, H is the $m \times n$ output matrix, $\{w_k\}$ and $\{v_k\}$ are independent Gaussian zero-mean white and uncorrelated random processes, respectively, Q is the $n \times n$ plant noise covariance matrix, R is the $m \times m$ measurement noise covariance matrix and x_0 is a Gaussian

random process with mean \bar{x}_0 and covariance P_0 . In the sequel, consider that Q, R are positive definite matrices and denote $Q, R > O$.

The filtering 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 $\{z_1, z_2, \dots, z_L\}$ in order to calculate an estimate value $x_{L|L}$ of the state vector x_L . The discrete time Lainiotis filter [6] is a well known algorithm that solves the filtering problem. The estimation $x_{k|k}$ and the corresponding estimation error covariance matrix $P_{k|k}$ at time k are computed by the following equations, consisting the Lainiotis filter,

$$P_{k+1|k+1} = \mathcal{P}_n + \mathcal{F}_n [I + P_{k|k} \mathcal{O}_n]^{-1} P_{k|k} \mathcal{F}_n^T \quad (1)$$

$$\begin{aligned} x_{k+1|k+1} = & \mathcal{F}_n [I + P_{k|k} \mathcal{O}_n]^{-1} x_{k|k} + \\ & + (\mathcal{K}_n + \mathcal{F}_n [I + P_{k|k} \mathcal{O}_n]^{-1} P_{k|k} \mathcal{K}_m) z_{k+1} \end{aligned} \quad (2)$$

for $k \geq 0$, with initial conditions $P_{0|0} = P_0$ and $x_{0|0} = \bar{x}_0$, where the following constant matrices are calculated off-line:

$$A = [HQH^T + R]^{-1} \quad (3)$$

$$\mathcal{K}_n = QH^T A \quad (4)$$

$$\mathcal{K}_m = F^T H^T A \quad (5)$$

$$\mathcal{P}_n = Q - \mathcal{K}_n H Q = Q - QH^T A H Q \quad (6)$$

$$\mathcal{F}_n = F - \mathcal{K}_n H F = F - QH^T A H F \quad (7)$$

$$\mathcal{O}_n = \mathcal{K}_m H F = F^T H^T A H F \quad (8)$$

with \mathcal{F}_n is a $n \times n$ matrix, while $\mathcal{K}_n, \mathcal{K}_m$ are $n \times m$ matrices. The $n \times n$ matrices \mathcal{P}_n and \mathcal{O}_n are symmetric. Also, the $m \times m$ matrix $HQH^T + R$ is nonsingular, since $R > O$, which means that no measurement is exact; this is reasonable in physical problems. Moreover, since $Q, R > O$, A is a well defined $m \times m$ symmetric and positive definite matrix as well as \mathcal{O}_n is positive definite. Furthermore, since $Q > O$ using the matrix inversion Lemma³ and substituting the matrix A by (3) in (6) we may rewrite \mathcal{P}_n as

$$\mathcal{P}_n = Q - QH^T [HQH^T + R]^{-1} H Q = [Q^{-1} + H^T R^{-1} H]^{-1} \quad (9)$$

³Let A, C be nonsingular matrices, then holds:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

from which it is clear that the symmetric \mathcal{P}_n is a positive definite matrix. Equation (1) is the Riccati equation emanating from Lainiotis filter.

In the case of infinite measurement noise ($R \rightarrow \infty$), we have $A = O$, $\mathcal{K}_n = O$, $\mathcal{K}_m = O$, $\mathcal{P}_n = Q$, $\mathcal{F}_n = F$, $\mathcal{O}_n = O$ and the Lainiotis filter becomes:

$$P_{k+1|k+1} = \mathcal{P}_n + \mathcal{F}_n P_{k|k} \mathcal{F}_n^T = Q + F P_{k|k} F^T \quad (10)$$

$$x_{k+1|k+1} = \mathcal{F}_n x_{k|k} = F x_{k|k} \quad (11)$$

Equation (10) is the Lyapunov equation emanating from Lainiotis filter.

3 Implementation of Lainiotis filter via Chandrasekhar type algorithm

For time invariant systems, it is well known [2] that if the signal process model is asymptotically stable (i.e. all eigenvalues of F lie inside the unit circle), then there exists a steady state value P of the estimation error covariance matrix. The steady state solution P is calculated by recursively implementing the *Riccati equation* emanating from Lainiotis filter (1) for $k = 0, 1, \dots$, with initial condition $P_{0|0} = P_0$. The steady state or limiting solution of the Riccati equation is independent of the initial condition [2]. The discrete time Riccati equation emanating from the Lainiotis filter equations has attracted enormous attention. In view of the importance of the Riccati equation, there exists considerable literature on its recursive solutions [4, 7], concerning per step or doubling algorithms. The Chandrasekhar type algorithm has been used [7, 8] to solve the Riccati equation (1). The Chandrasekhar type algorithm consists of the recursion

$$P_{k+1|k+1} = P_{k|k} + Y_k S_k Y_k^T$$

using recursions for the suitable quantities Y_k and S_k . Hence, the algorithm is based on the idea of defining the difference equation

$$\delta P_k = P_{k+1|k+1} - P_{k|k}, \quad (12)$$

and its factorization

$$\delta P_k = Y_k S_k Y_k^T, \quad (13)$$

where Y_k is an $n \times r$ matrix and S_k is an $r \times r$ matrix, with

$$0 \leq r = \text{rank}(\delta P_0) \leq n.$$

For every $k = 0, 1, \dots$, denoting

$$O_k = P_{k|k} + \mathcal{O}_n^{-1} \quad (14)$$

we note that O_k is a $n \times n$ symmetric and positive definite matrix due to the presence of \mathcal{O}_n , (recalling that \mathcal{O}_n in (8) is a positive definite matrix and $P_{k|k}$ is a positive semidefinite as estimation error covariance matrix). Also, since O_k is a nonsingular matrix for every $k = 0, 1, \dots$, the equation (14) may be written:

$$P_{k|k} O_k^{-1} = I - \mathcal{O}_n^{-1} O_k^{-1} \quad (15)$$

Using the above notations and substituting the equations of the Lainiotis filter by a set of Chandrasekhar type difference equations, a recursive filtering algorithm is proposed, as established in the following theorem, which presents computational advantage compared to the classical filtering algorithm, (see 4 and 5 statements in the next Section 4).

Theorem 3.1. *Let the measurement noise R be a positive definite matrix, the plant noise Q be a positive definite matrix and $P_{k|k}$ is a nonsingular matrix, for every $k = 0, 1, 2, \dots$. The set of the following recursive equations compose the new algorithm for the solution of the discrete time Lainiotis filter,*

$$O_{k+1} = O_k + Y_k S_k Y_k^T \quad (16)$$

$$Y_{k+1} = \mathcal{F}_n \mathcal{O}_n^{-1} O_k^{-1} Y_k \quad (17)$$

$$S_{k+1} = S_k - S_k Y_k^T O_{k+1}^{-1} Y_k S_k \quad (18)$$

$$P_{k+1|k+1} = P_{k|k} + Y_k S_k Y_k^T \quad (19)$$

$$x_{k+1|k+1} = \mathcal{F}_n \mathcal{O}_n^{-1} O_k^{-1} x_{k|k} + (\mathcal{K}_n + \mathcal{F}_n \mathcal{O}_n^{-1} O_k^{-1} P_{k|k} \mathcal{K}_m) z_{k+1}, \quad (20)$$

with initial conditions:

$$P_{0|0} = P_0$$

$$x_{0|0} = \bar{x}_0$$

$$O_0 = P_0 + \mathcal{O}_n^{-1} \quad (21)$$

$$Y_0 S_0 Y_0^T = \mathcal{P}_n + \mathcal{F}_n [I + P_0 \mathcal{O}_n]^{-1} P_0 \mathcal{F}_n^T - P_0 \quad (22)$$

where $\mathcal{F}_n, \mathcal{O}_n, \mathcal{K}_n, \mathcal{K}_m, \mathcal{P}_n$ are the matrices in (3)-(8).

Proof. Combining (14) and (12) we write

$$O_{k+1} = P_{k+1|k+1} + \mathcal{O}_n^{-1} = P_{k+1|k+1} + O_k - P_{k|k} = O_k + \delta P_k,$$

i.e., for every $k = 0, 1, 2, \dots$, holds

$$\delta P_k = O_{k+1} - O_k, \quad (23)$$

in which substituting δP_k by (13) the recursion equation in (16) is obvious. Moreover, using elementary algebraic operations and properties we may write

$$M + N = N(M^{-1} + N^{-1})M, \quad M^{-1} - N^{-1} = N^{-1}(N - M)M^{-1}, \quad (24)$$

when M, N are $n \times n$ nonsingular matrices, as well as

$$[I + P_{k|k}\mathcal{O}_n]^{-1}P_{k|k} = [P_{k|k}(P_{k|k}^{-1} + \mathcal{O}_n)]^{-1}P_{k|k} = [P_{k|k}^{-1} + \mathcal{O}_n]^{-1}, \quad (25)$$

due to the nonsingularity of $P_{k|k}$ for every $k = 0, 1, \dots$. Combining (12), (1), (25), the first equality in (24), (14) and (15), we derive:

$$\begin{aligned} \delta P_{k+1} &= P_{k+2|k+2} - P_{k+1|k+1} \\ &= \mathcal{F}_n \left([I + P_{k+1|k+1}\mathcal{O}_n]^{-1}P_{k+1|k+1} - [I + P_{k|k}\mathcal{O}_n]^{-1}P_{k|k} \right) \mathcal{F}_n^T \\ &= \mathcal{F}_n \left([P_{k+1|k+1}^{-1} + \mathcal{O}_n]^{-1} - [P_{k|k}^{-1} + \mathcal{O}_n]^{-1} \right) \mathcal{F}_n^T \\ &= \mathcal{F}_n \left(\left[\mathcal{O}_n(P_{k+1|k+1} + \mathcal{O}_n^{-1})P_{k+1|k+1}^{-1} \right]^{-1} - \left[\mathcal{O}_n(P_{k|k} + \mathcal{O}_n^{-1})P_{k|k}^{-1} \right]^{-1} \right) \mathcal{F}_n^T \\ &= \mathcal{F}_n \left(P_{k+1|k+1}[P_{k+1|k+1} + \mathcal{O}_n^{-1}]^{-1}\mathcal{O}_n^{-1} - P_{k|k}[P_{k|k} + \mathcal{O}_n^{-1}]^{-1}\mathcal{O}_n^{-1} \right) \mathcal{F}_n^T \\ &= \mathcal{F}_n \left(P_{k+1|k+1}[P_{k+1|k+1} + \mathcal{O}_n^{-1}]^{-1} - P_{k|k}[P_{k|k} + \mathcal{O}_n^{-1}]^{-1} \right) \mathcal{O}_n^{-1}\mathcal{F}_n^T \\ &= \mathcal{F}_n \left(P_{k+1|k+1}\mathcal{O}_{k+1}^{-1} - P_{k|k}\mathcal{O}_k^{-1} \right) \mathcal{O}_n^{-1}\mathcal{F}_n^T \\ &= \mathcal{F}_n\mathcal{O}_n^{-1} \left(\mathcal{O}_k^{-1} - \mathcal{O}_{k+1}^{-1} \right) \mathcal{O}_n^{-1}\mathcal{F}_n^T \end{aligned}$$

Using the second equality of (24) and (23) the last equation may be written as:

$$\begin{aligned} \delta P_{k+1} &= \mathcal{F}_n\mathcal{O}_n^{-1}\mathcal{O}_{k+1}^{-1} \left(\mathcal{O}_{k+1} - \mathcal{O}_k \right) \mathcal{O}_k^{-1}\mathcal{O}_n^{-1}\mathcal{F}_n^T \\ &= \mathcal{F}_n\mathcal{O}_n^{-1} \left(\mathcal{O}_k^{-1}\mathcal{O}_k \right) \mathcal{O}_{k+1}^{-1} \left(\delta P_k \right) \mathcal{O}_k^{-1}\mathcal{O}_n^{-1}\mathcal{F}_n^T \\ &= \mathcal{F}_n\mathcal{O}_n^{-1}\mathcal{O}_k^{-1} \left(\mathcal{O}_{k+1} - \delta P_k \right) \mathcal{O}_{k+1}^{-1} \left(\delta P_k \right) \mathcal{O}_k^{-1}\mathcal{O}_n^{-1}\mathcal{F}_n^T \\ &= \mathcal{F}_n\mathcal{O}_n^{-1}\mathcal{O}_k^{-1} \left(\delta P_k \right) \mathcal{O}_k^{-1}\mathcal{O}_n^{-1}\mathcal{F}_n^T - \mathcal{F}_n\mathcal{O}_n^{-1}\mathcal{O}_k^{-1} \left(\delta P_k \right) \mathcal{O}_{k+1}^{-1} \left(\delta P_k \right) \mathcal{O}_k^{-1}\mathcal{O}_n^{-1}\mathcal{F}_n^T \end{aligned}$$

From (13) the last equation yields

$$\delta P_{k+1} = \mathcal{F}_n \mathcal{O}_n^{-1} \mathcal{O}_k^{-1} Y_k S_k Y_k^T \mathcal{O}_k^{-1} \mathcal{O}_n^{-1} \mathcal{F}_n^T - \mathcal{F}_n \mathcal{O}_n^{-1} \mathcal{O}_k^{-1} Y_k S_k Y_k^T \mathcal{O}_{k+1}^{-1} Y_k S_k Y_k^T \mathcal{O}_k^{-1} \mathcal{O}_n^{-1} \mathcal{F}_n^T$$

in which setting the matrix $Y_{k+1} = \mathcal{F}_n \mathcal{O}_n^{-1} \mathcal{O}_k^{-1} Y_k$ by (17) immediately arises

$$\delta P_{k+1} = Y_{k+1} S_k Y_{k+1}^T - Y_{k+1} S_k Y_k^T \mathcal{O}_{k+1}^{-1} Y_k S_k Y_{k+1}^T. \quad (26)$$

By (13) we have $\delta P_{k+1} = Y_{k+1} S_{k+1} Y_{k+1}^T$, thus the equation in (26) may be formulated as :

$$Y_{k+1} S_{k+1} Y_{k+1}^T = Y_{k+1} (S_k - S_k Y_k^T \mathcal{O}_{k+1}^{-1} Y_k S_k) Y_{k+1}^T$$

Multiplying with Y_{k+1}^T on the left and Y_{k+1} on the right the last equality the recursion equation in (18) is derived.

Furthermore, rewriting $x_{k+1|k+1}$ in (2) with different way and due to (14) we conclude

$$\begin{aligned} x_{k+1|k+1} &= \mathcal{F}_n \mathcal{O}_n^{-1} \mathcal{O}_n [I + P_{k|k} \mathcal{O}_n]^{-1} x_{k|k} + \\ &\quad + \left(\mathcal{K}_n + \mathcal{F}_n \mathcal{O}_n^{-1} \mathcal{O}_n [I + P_{k|k} \mathcal{O}_n]^{-1} P_{k|k} \mathcal{K}_m \right) z_{k+1} \\ &= \mathcal{F}_n \mathcal{O}_n^{-1} [(I + P_{k|k} \mathcal{O}_n) \mathcal{O}_n^{-1}]^{-1} x_{k|k} + \\ &\quad + \left(\mathcal{K}_n + \mathcal{F}_n \mathcal{O}_n^{-1} [(I + P_{k|k} \mathcal{O}_n) \mathcal{O}_n^{-1}]^{-1} P_{k|k} \mathcal{K}_m \right) z_{k+1} \\ &= \mathcal{F}_n \mathcal{O}_n^{-1} [P_{k|k} + \mathcal{O}_n^{-1}]^{-1} x_{k|k} + \left(\mathcal{K}_n + \mathcal{F}_n \mathcal{O}_n^{-1} [P_{k|k} + \mathcal{O}_n^{-1}]^{-1} P_{k|k} \mathcal{K}_m \right) z_{k+1} \\ &= \mathcal{F}_n \mathcal{O}_n^{-1} \mathcal{O}_k^{-1} x_{k|k} + \left(\mathcal{K}_n + \mathcal{F}_n \mathcal{O}_n^{-1} \mathcal{O}_k^{-1} P_{k|k} \mathcal{K}_m \right) z_{k+1} \end{aligned}$$

showing thus the equation (20).

Moreover, $P_{0|0} = P_0$ and $x_{0|0} = \bar{x}_0$ are given as the initial conditions of the problem; by (14) \mathcal{O}_0 is computed for $k = 0$ and the matrices Y_0, S_0 are computed by the factorization of the matrix $\mathcal{P}_n + \mathcal{F}_n [I + P_0 \mathcal{O}_n]^{-1} P_0 \mathcal{F}_n^T - P_0$ in (22) in order to used as initial conditions. \square

Remark 3.1. 1. For the boundary values of $r = \text{rank}(\delta P_0)$ we note that:

- If $r = 0$, then, from (12) arises that the estimation covariance matrix remains constant, i.e. $P_{k|k} = P_0$, and equation (23) yields $\mathcal{O}_k = \mathcal{O}_0$, for every $k = 0, 1, 2, \dots$. Thus the algorithm of Theorem 3.1 computes iteratively only the estimation $x_{k+1|k+1}$ taking the form:

$$x_{k+1|k+1} = \mathcal{F}_n \mathcal{O}_n^{-1} \mathcal{O}_0^{-1} x_{k|k} + \left(\mathcal{K}_n + \mathcal{F}_n \mathcal{O}_n^{-1} \mathcal{O}_0^{-1} P_0 \mathcal{K}_m \right) z_{k+1}$$

- If $r = n$ and $P_{0|0} = P_0 = O$, then we are able to use the initial conditions $Y_0 = I$ and $S_0 = \mathcal{P}_n$.
2. For the zero initial condition $P_{0|0} = P_0 = O$, by (1) we derive $P_{1|1} = \mathcal{P}_n$; recalling that by (9) holds $\mathcal{P}_n > O$, it is evident that for every $k = 1, 2, \dots$ arises $P_{k|k} > O$, that guarantees $P_{k|k}$ be a nonsingular matrix. Hence Theorem 3.1 is applicable for initial condition $P_{0|0} = P_0 = O$; in this case by (21)-(22) we are able to use the following initial conditions: $O_0 = \mathcal{O}_n^{-1}$ and $Y_0 S_0 Y_0^T = \mathcal{P}_n$.

3.1 Infinite measurement noise $R \rightarrow \infty$

In the following, the special case of infinite measurement noise is presented. In this case $\mathcal{P}_n = Q$, $\mathcal{F}_n = F$ and $\mathcal{O}_n = O$, then the Riccati equation (1) becomes the Lyapunov equation (10). Using (12) and combining (10) with (13) we have

$$\begin{aligned} \delta P_{k+1} &= P_{k+2|k+2} - P_{k+1|k+1} = F (P_{k+1|k+1} - P_{k|k}) F^T = F (\delta P_k) F^T \\ &= F Y_k S_k Y_k^T F^T, \end{aligned}$$

where setting $Y_{k+1} = F Y_k$ the above equality is formulated $Y_{k+1} S_{k+1} Y_{k+1}^T = \delta P_{k+1} = Y_{k+1} S_k Y_{k+1}^T$ and after some algebra arises :

$$S_{k+1} = S_k$$

Since the last equality of the matrices holds for every $k = 1, 2, \dots$, without loss of generality, we consider an arbitrary $r \times r$ symmetric matrix

$$S_k = S, \tag{27}$$

with $\text{rank}(S) = r$ and $0 < r \leq n$. Thus, using (19), (11) and (27) the following filtering algorithm, which is based on the Chandrasekhar type algorithm, is established.

$$\begin{aligned} Y_{k+1} &= F Y_k \\ P_{k+1|k+1} &= P_{k|k} + Y_k S Y_k^T \\ x_{k+1|k+1} &= F x_{k|k}, \end{aligned}$$

and with initial conditions:

$$\begin{aligned} P_{0|0} &= P_0, \quad x_{0|0} = \bar{x}_0, \\ Y_0 S Y_0^T &= Q + F P_0 F^T - P_0 \end{aligned}$$

Since in (27) the matrix S can be arbitrarily chosen, we propose as S the $r \times r$ identity matrix; thus we are able to establish the proposed algorithm, which is formulated in the next theorem.

Theorem 3.2. *Let R be the infinite measurement noise ($R \rightarrow \infty$), the plant noise Q be a positive definite matrix and F be a transition matrix. The set of the following recursive equations compose the algorithm for the solution of the discrete time Lainiotis filter, for $k = 1, 2, \dots$,*

$$Y_{k+1} = FY_k \quad (28)$$

$$P_{k+1|k+1} = P_{k|k} + Y_k Y_k^T \quad (29)$$

$$x_{k+1|k+1} = Fx_{k|k}, \quad (30)$$

with initial conditions:

$$\begin{aligned} P_{0|0} &= P_0, \quad x_{0|0} = \bar{x}_0, \\ Y_0 Y_0^T &= Q + FP_0 F^T - P_0 \end{aligned} \quad (31)$$

Remark 3.2. In the special case $P_{0|0} = P_0 = O$, then equation (31) becomes $Y_0 Y_0^T = Q$.

4 Computational comparison of algorithms

The two implementations of the Lainiotis filter presented above are equivalent with respect to their behavior: they calculate theoretically the same estimates, due to the fact that equations (1)-(2) are equivalent to equations in Theorem 3.1 (i.e. (16)-(20)) and equations (10)-(11) are equivalent to equations (28)-(30) for the case of infinite measurement noise. Then, it is reasonable to assume that both implementations of 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, 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) is not taken into account.

The computational analysis is based on the analysis in [3]: 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.

Table 1. Calculation burden of matrix operations

Matrix Operation	Calculation Burden
$A(n \times m) + B(n \times m) = C(n \times m)$	nm
$A(n \times n) + B(n \times n) = S(n \times n)$ S : symmetric	$\frac{1}{2}(n^2 + n)$
$I(n \times n) + A(n \times n) = B(n \times n)$ I : identity	n
$A(n \times m) \cdot B(m \times k) = C(n \times k)$	$2nmk - nk$
$A(n \times m) \cdot B(m \times n) = S(n \times n)$ S : symmetric	$n^2m + nm - \frac{1}{2}(n^2 + n)$
$[A(n \times n)]^{-1} = B(n \times n)$	$\frac{1}{6}(16n^3 - 3n^2 - n)$

The per recursion calculation burden of the Lainiotis filter implementations are summarized in Table 2. The details are given in the Appendix.

Table 2. Per recursion calculation burden of algorithms

Implementation	Noise	Per recursion calculation burden
Classical	$R > O$	$CB_{c,1} = \frac{1}{6}(64n^3 - 4n) + 2n^2m + 2nm$
Classical	$R \rightarrow \infty$	$CB_{c,2} = 3n^3 + 2n^2 - n$
Proposed	$R > O$	$CB_{p,1} = \frac{1}{6}(56n^3 - 3n^2 - 5n) + 3nr^2$ $- 2nr + 7n^2r + 2n^2m + 2nm$
Proposed	$R \rightarrow \infty$	$CB_{p,2} = 3n^2r + 2n^2 - n$

From Table 2, we derive the following conclusions:

1. The per recursion calculation burden of the classical implementation depends on the state vector dimension n .
2. The per recursion calculation burden of the proposed implementation depends on the state vector dimension n and on $r = \text{rank}(\delta P_0)$.
3. Concerning the non-infinite measurement noise case ($R > O$) and defining $q(n, r) = CB_{c,1} - CB_{p,1}$, from Table 2 the respective calculation burdens yield the relation:

$$q(n, r) = \frac{1}{6}(8n^3 + 3n^2 + n) - 3nr^2 + 2nr - 7n^2r \quad (32)$$

From Remark 3.1 the case $r = 0$ gives degenerated algorithm; thus consider $r \geq 1$ we investigate two cases : (a) $r = n$, and (b) $r < n$.

- (a) $1 \leq r = n$. In this case, it is obvious that $q(n, n) = \frac{1}{6}(-52n^3 + 15n^2 + n)$ and since $q(n, n)$ is a decreasing function, we compute $q(n, n) \leq q(1, 1) = -6 < 0$. Hence, if $r = n$, then the classical implementation is faster than the proposed one.

(b) $1 \leq r < n$. In this case, we rewrite the equality in (32) as

$$q(n, r) = \frac{n}{6}(-18r^2 + (-42n + 12)r + 8n^2 + 3n + 1) = \frac{n}{6}f(r, n), \quad (33)$$

with $f(r, n) = -18r^2 + (-42n + 12)r + 8n^2 + 3n + 1$. The discriminant of $f(r, n)$ is

$$\Delta(n) = 2340n^2 - 792n + 216 > 0,$$

and its zeros are :

$$r_1(n) = \frac{-42n + 12 - \sqrt{\Delta(n)}}{36}, \quad r_2(n) = \frac{-42n + 12 + \sqrt{\Delta(n)}}{36} \quad (34)$$

Hence, the factorization of $f(r, n)$ is $f(r, n) = -18(r - r_1(n))(r - r_2(n))$, thus, the equality of $q(n, r)$ in (33) can be written as

$$q(n, r) = -3n(r - r_1(n))(r - r_2(n)). \quad (35)$$

Also, it is easily proved that for $n = 1, 2, \dots$ holds $\sqrt{\Delta(n)} > 42n - 12$, from which immediately arises $r_1(n) < 0$ and $r_2(n) > 0$; thus, due to the fact $r \geq 1$, it is obvious

$$r - r_1(n) > 0.$$

Consequently, in (35) the sign of $q(n, r)$ depends on the sign of $r - r_2(n)$, with $r_2(n)$ in (34), i.e., the choice of implementation of the suitable algorithm is related to the comparison of quantities $r, r_2(n)$;

- if $r > r_2(n) \Rightarrow q(n, r) < 0$, thus the classical implementation is faster than the proposed one.
- if $r < r_2(n) \Rightarrow q(n, r) > 0$, thus the proposed implementation is faster than the classical one.

4. Figure 1 depicts the relation between n and r that may hold in order to decide, which implementation is faster. In fact r is plotted as function of n using $r_2(n)$ in (34). Then, we are able to establish the following *Rule of Thumb*: the proposed Lainiotis filter implementation via Chandrasekhar type algorithm is faster than the classical implementation if the following relation holds:

$$r < 0.18n \quad (36)$$

Figure 1: Proposed algorithm may be faster than the classical one.

Thus, we are able to choose in advance the implementation of the faster algorithm comparing *only* the quantities r and n by (36).

5. Concerning the infinite measurement noise case ($R \rightarrow \infty$), the calculation burden of the classical implementation is greater than or equal to the calculation burden of the proposed implementation; the equality holds for $r = n$. Thus, the proposed implementation is faster than the classical one.

ACKNOWLEDGEMENTS. The authors are deeply grateful to referees for suggestions that have considerably improved the quality of the paper.

References

- [1] Abdelhakim Aknouche and Fayçal Hamdi, Calculating the autocovariances and the likelihood for periodic V ARMA models, *Journal of Statistical Computation and Simulation*, **79**(3), (2009), 227-239.
- [2] B.D.O. Anderson, J.B. Moore, *Optimal Filtering*, Prentice Hall inc., 1979.
- [3] N. Assimakis, M. Adam, Discrete time Kalman and Lainiotis filters comparison, *Int. Journal of Mathematical Analysis (IJMA)*, **1**(13-16), (2007), 635-659.
- [4] N.D. Assimakis, D.G. Lainiotis, S.K. Katsikas, F.L. Sanida, A survey of recursive algorithms for the solution of the discrete time Riccati equation, *Nonlinear Analysis, Theory, Methods and Applications*, **30**, (1997), 2409-2420.
- [5] J.S. Baras and D.G. Lainiotis, Chandrasekhar algorithms for linear time varying distributed systems, *Information Sciences*, **17**(2), (1979), 153-167.
- [6] D.G. Lainiotis, Discrete Riccati Equation Solutions: Partitioned Algorithms, *IEEE Transactions on AC*, **AC-20**, (1975), 555-556.

- [7] D.G. Lainiotis, N.D. Assimakis, S.K. Katsikas, A new computationally effective algorithm for solving the discrete Riccati equation, *Journal of Mathematical Analysis and Applications*, **186**(3), (1994), 868-895.
- [8] S. Nakamori, A. Hermoso-Carazo, J. Jiménez-López and J. Linares-Pérez, Chandrasekhar-type filter for a wide-sense stationary signal from uncertain observations using covariance information, *Applied Mathematics and Computation*, **151**, (2004), 315-325.
- [9] S. Nakamori, Chandrasekhar-type recursive Wiener estimation technique in linear discrete-time stochastic systems, *Applied Mathematics and Computation*, **188**(2), (2007), 1656-1665.

Appendix

Calculation burdens of algorithms

A Measurement noise is a positive definite matrix ($R > O$)

A.1 Classical implementation of Lainiotis filter

Matrix Operation	Matrix Dimensions	Calculation Burden
$P_{k k}\mathcal{O}_n$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$I + P_{k k}\mathcal{O}_n$	$(n \times n) + (n \times n)^\ddagger$	n
$[I + P_{k k}\mathcal{O}_n]^{-1}$	$n \times n$	$\frac{1}{6}(16n^3 - 3n^2 - n)$
$[I + P_{k k}\mathcal{O}_n]^{-1}P_{k k}$	$(n \times n) \cdot (n \times n)^*$	$n^3 + \frac{1}{2}(n^2 - n)$
$\mathcal{F}_n[I + P_{k k}\mathcal{O}_n]^{-1}P_{k k}$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$\mathcal{F}_n[I + P_{k k}\mathcal{O}_n]^{-1}P_{k k}\mathcal{F}_n^T$	$(n \times n) \cdot (n \times n)^*$	$n^3 + \frac{1}{2}(n^2 - n)$
$P_{k+1 k+1} = \mathcal{P}_n + \mathcal{F}_n[I + P_{k k}\mathcal{O}_n]^{-1}P_{k k}\mathcal{F}_n^T$	$(n \times n) + (n \times n)^*$	$\frac{1}{2}(n^2 + n)$
$\mathcal{F}_n[I + P_{k k}\mathcal{O}_n]^{-1}P_{k k}\mathcal{K}_m$	$(n \times n) \cdot (n \times m)$	$2n^2m - nm$
$\mathcal{K}_n + \mathcal{F}_n[I + P_{k k}\mathcal{O}_n]^{-1}P_{k k}\mathcal{K}_m$	$(n \times m) + (n \times m)$	nm
$(\mathcal{K}_n + \mathcal{F}_n[I + P_{k k}\mathcal{O}_n]^{-1}P_{k k}\mathcal{K}_m) z_{k+1}$	$(n \times m) \cdot (m \times 1)$	$2nm - n$
$\mathcal{F}_n[I + P_{k k}\mathcal{O}_n]^{-1}$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$\mathcal{F}_n[I + P_{k k}\mathcal{O}_n]^{-1}x_{k k}$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
$x_{k+1 k+1} = \mathcal{F}_n[I + P_{k k}\mathcal{O}_n]^{-1}x_{k k} + (\mathcal{K}_n + \mathcal{F}_n[I + P_{k k}\mathcal{O}_n]^{-1}P_{k k}\mathcal{K}_m) z_{k+1}$	$(n \times 1) + (n \times 1)$	n
Total	$CB_{c,1} = \frac{1}{6}(64n^3 - 4n) + 2n^2m + 2nm$	

\ddagger I identity matrix * symmetric matrix

A.2 Proposed implementation via Chandrasekhar type algorithm

Matrix Operation	Matrix Dimensions	Calculation Burden
$Y_k S_k$	$(n \times r) \cdot (r \times r)$	$2nr^2 - nr$
$Y_k S_k Y_k^T$	$(n \times r) \cdot (r \times n)^*$	$n^2 r + nr - \frac{1}{2}(n^2 + n)$
$O_{k+1} = O_k + Y_k S_k Y_k^T$	$(n \times n) + (n \times n)^*$	$\frac{1}{2}(n^2 + n)$
O_k^{-1}	$n \times n$	$\frac{1}{6}(16n^3 - 3n^2 - n)$
$O_k^{-1} Y_k$	$(n \times n) \cdot (n \times r)$	$2n^2 r - nr$
$Y_{k+1} = \mathcal{F}_n O_n^{-1} O_k^{-1} Y_k$	$(n \times n) \cdot (n \times r)$	$2n^2 r - nr$
O_{k+1}^{-1}	$n \times n$	$\frac{1}{6}(16n^3 - 3n^2 - n)$
$O_{k+1}^{-1} Y_k S_k$	$(n \times n) \cdot (n \times r)$	$2n^2 r - nr$
$S_k Y_k^T O_{k+1}^{-1} Y_k S_k$	$(r \times n) \cdot (n \times r)^*$	$r^2 n + rn - \frac{1}{2}(r^2 + r)$
$S_{k+1} = S_k - S_k Y_k^T O_{k+1}^{-1} Y_k S_k$	$(r \times r) + (r \times r)^*$	$\frac{1}{2}(r^2 + r)$
$P_{k+1 k+1} = P_{k k} + Y_k S_k Y_k^T$	$(n \times n) + (n \times n)^*$	$\frac{1}{2}(n^2 + n)$
$\mathcal{F}_n O_n^{-1} O_k^{-1}$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$\mathcal{F}_n O_n^{-1} O_k^{-1} x_{k k}$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
$\mathcal{F}_n O_n^{-1} O_k^{-1} P_{k k}$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$\mathcal{F}_n O_n^{-1} O_k^{-1} P_{k k} \mathcal{K}_m$	$(n \times n) \cdot (n \times m)$	$2n^2 m - nm$
$\mathcal{K}_n + \mathcal{F}_n O_n^{-1} O_k^{-1} P_{k k} \mathcal{K}_m$	$(n \times m) + (n \times m)$	nm
$(\mathcal{K}_n + \mathcal{F}_n O_n^{-1} O_k^{-1} P_{k k} \mathcal{K}_m) z_{k+1}$	$(n \times m) \cdot (m \times 1)$	$2nm - n$
$x_{k+1 k+1} = \mathcal{F}_n O_n^{-1} O_k^{-1} x_{k k} +$ $+ (\mathcal{K}_n + \mathcal{F}_n O_n^{-1} O_k^{-1} P_{k k} \mathcal{K}_m) z_{k+1}$	$(n \times 1) + (n \times 1)$	n
Total	$CB_{p,1} = \frac{1}{6}(56n^3 - 3n^2 - 5n) + 3nr^2 - 2nr + 7n^2 r + 2n^2 m + 2nm$	

* symmetric matrix

B Infinite measurement noise $R \rightarrow \infty$

B.1 Classical implementation of Lainiotis filter

Matrix Operation	Matrix Dimensions	Calculation Burden
$FP_{k k}$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$FP_{k k}F^T$	$(n \times n) \cdot (n \times n)^*$	$n^3 + \frac{1}{2}(n^2 - n)$
$P_{k+1 k+1} = Q + FP_{k k}F^T$	$(n \times n) + (n \times n)^*$	$\frac{1}{2}(n^2 + n)$
$x_{k+1 k+1} = Fx_{k k}$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
Total		$CB_{c,2} = 3n^3 + 2n^2 - n$

* symmetric matrix

B.2 Proposed implementation via Chandrasekhar type algorithm

Matrix Operation	Matrix Dimensions	Calculation Burden
$Y_{k+1} = FY_k$	$(n \times n) \cdot (n \times r)$	$2n^2r - nr$
$Y_k Y_k^T$	$(n \times r) \cdot (r \times n)^*$	$n^2r + nr - \frac{1}{2}(n^2 + n)$
$P_{k+1 k+1} = P_{k k} + Y_k Y_k^T$	$(n \times n) + (n \times n)^*$	$\frac{1}{2}(n^2 + n)$
$x_{k+1 k+1} = Fx_{k k}$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
Total		$CB_{p,2} = 3n^2r + 2n^2 - n$

* symmetric matrix