Chandrasekhar-type Algorithms with Gain Elimination

NICHOLAS ASSIMAKIS[1], MARIA ADAM[2]
[1]Department of Digital Industry Technologies,
National and Kapodistrian University of Athens,
34400 Psachna Evias,
GREECE

[2]Department of Computer Science and Biomedical Informatics,
University of Thessaly,
2-4 Papasiopoulou Str., 35131, Lamia,
GREECE

Abstract: - Chandrasekhar-type algorithms are associated with the Riccati equation emanating from the Kalman filter in linear systems which describe the relationship between the n-dimensional state and the m-dimensional measurement. The traditional Chandrasekhar-type algorithms use the Kalman filter gain to compute the prediction error covariance. In this paper, two variations of Chandrasekhar-type algorithms eliminating the Kalman filter gain are proposed. The proposed Chandrasekhar-type algorithms with gain elimination may be faster than the traditional Chandrasekhar-type algorithms, depending on the model dimensions.

Key-Words: - Discrete time, Kalman filter, Discrete algebraic Riccati equation, algebraic Lyapunov equation, Chandrasekhar-type algorithms, Kalman filter gain, convergence theory.

Received: April 13, 2023. Revised: December 18, 2023. Accepted: December 27, 2023. Published: December 31, 2023.

1 Introduction

Consider discrete time, $k \geq 0$, invariant linear systems, which are traditionally formulated by the state space equations, [1]:

$$x(k+1) = Fx(k) + w(k) \tag{1}$$

$$z(k) = Hx(k) + v(k) \tag{2}$$

Here, $x(k)$ defines the state vector of dimension n with Gaussian noise $w(k)$ of zero mean and covariance Q and $z(k)$ defines the measurement vector of dimension m with Gaussian noise $v(k)$ of zero mean and covariance R. In addition, F is the transition matrix and H is the output matrix. All the model parameters F, H, Q, R are constant. The initial state $x(0)$ is Gaussian with mean x_0 and covariance P_0.

The discrete time Kalman filter, [1], [2], is the celebrated algorithm, which computes the state estimation $x(k/k)$ and the estimation error covariance $P(k/k)$ as well as the state prediction $x(k/k-1)$ and the prediction error covariance matrix $P(k/k-1)$. The prediction and estimation error covariances do not depend on the measurements; thus they can be computed off-line using the equations

$$O(k) = HP(k/k-1)H^T + R \tag{3}$$

$$K(k) = P(k/k-1)H^T O^{-1}(k) \tag{4}$$

$$P(k/k) = [I - K(k)H]P(k/k-1) \tag{5}$$

$$P(k+1/k) = Q + FP(k/k)F^T \tag{6}$$

with initial condition $P(0/-1) = P_0$.

Here, M^T denotes the transpose of M, I denotes the identity matrix, and $K(k)$ is the Kalman filter gain. Note that the existence of the inverse of $O(k)$ is ensured assuming that R is positive definite (this has the reasonable meaning that no measurement is accurate).

It is well known, [1], that $P(k/k-1)$ can be computed independently of the measurements, using the discrete algebraic Riccati equation:

$$\begin{aligned} P(k+1/k) = {} & Q + FP(k/k-1)F^T \\ & -FP(k/k-1)H^T \\ & [HP(k/k-1)H^T + R]^{-1}FP(k/k-1)F^T \end{aligned} \tag{7}$$

In the infinite measurement noise covariance case, where $R = \infty$, the discrete algebraic Riccati equation takes the form of the algebraic Lyapunov equation:

$$P(k+1/k) = Q + FP(k/k-1)F^T \quad (8)$$

In addition, if the model is asymptotically stable, then there is a unique steady state prediction error covariance P, which satisfies the discrete algebraic Riccati equation:

$$P = Q + FPF^T - FPH^T[HPH^T + R]^{-1}HPF^T \quad (9)$$

In the case $R = \infty$, if the model is asymptotically stable, then there is a unique steady state prediction error covariance matrix P, which satisfies the algebraic Lyapunov equation:

$$P = Q + FPF^T \quad (10)$$

Due to the importance of the Riccati equation, significant bibliography exists about iterative or algebraic solutions, [1], [3], [4], [5], [6], [7]. Chandrasekhar-type algorithms have been part of the folklore associated with the Riccati equation, [1]. Methods are described in [8], based on the solution of so-called Chandrasekhar-type equations than the classical Riccati-type equation. An important advantage of this method is the reduction in computational burden, when the state dimension is much greater than the measurement dimension, [1]. Chandrasekhar-type algorithms can be used to iteratively compute the prediction error covariance, [1], [8] or to compute the steady state solution of the Riccati equation. Chandrasekhar-type algorithms are applicable to Kalman filters, [9], [10] and to time varying as well as time invariant distributed systems, [11].

All Chandrasekhar-type algorithms use the Kalman filter gain. Iterative and algebraic algorithms for the computation of the steady state Kalman filter gain have been derived in [12]. The basic idea of this work is to eliminate the Kalman filter gain from the Chandrasekhar-type algorithms' equations, to reduce the computational effort, [10].

The Kalman filter gain elimination concept and the proposed variations of Chandrasekhar-type algorithms can find application in steady state Kalman filter design, where the Riccati equation solution is required. In addition, the proposed algorithms can be applied in control problems. The basic problems in control theory are (a) the controller design problem (control law design for the dynamical system) and (b) the state estimation problem (computation of the estimate of the states of the dynamical system). The Linear Quadratic Regulator (LQR) and the Kalman filter solve the associated problems, [13]. The proposed algorithms can be applied in the case of linear dynamical systems, to estimate the control effectiveness of the actuator on behalf of an actuator stuck fault incident occurring on airplanes, [14], to Kalman filter design that accounts for measurement differences, for the case of time-correlated measurement errors, [15], to Global Positioning System (GPS) and Inertial Navigation System (INS) integration during GPS outages using machine learning augmented with Kalman filter, [16].

The novelty of this work concerns: (a) the use of the Kalman filter gain elimination concept in the Riccati equation solution, (b) the derivation of Chandrasekhar-type algorithms with gain elimination, (c) the computation of the calculation burdens of the Chandrasekhar-type algorithms, (d) the determination of the faster Chandrasekhar-type algorithm via the system dimensions.

The paper is organized as follows: Section 2 summarizes the traditional Chandrasekhar-type algorithms. The Chandrasekhar-type algorithms based on Kalman filter gain elimination are derived in Section 3. In Section 4, the traditional and the proposed Chandrasekhar-type algorithms are compared concerning their calculation burdens. Section 5 summarizes the conclusions.

2 Traditional Chandrasekhar-type Algorithms

The basic idea in Chandrasekhar- type algorithms is to factorize the difference:

$$\delta P(k) = P(k+1/k) - P(k/k-1) \quad (11)$$

as

$$\delta P(k) = Y(k)S(k)Y^T(k) \quad (12)$$

where $S(k)$ is square symmetric matrix with dimension equal to:

$$r = \mathrm{rank}\big(\delta P(0)\big) \quad (13)$$

with $r \le n$.

There exist various equivalent Chandrasekhar-type equation sets. In this section, we deal with two Chandrasekhar-type algorithms, which are well described in [1]; we refer to these algorithms as Chandrasekhar-type algorithm – version 1 and Chandrasekhar-type algorithm – version 2.

Chandrasekhar-type algorithm – version 1

$$O(k+1) = O(k) + HY(k)S(k)Y^T(k)H^T$$
$$K(k+1) = [K(k)O(k) + Y(k)S(k)Y^T(k)H^T]O^{-1}(k+1)$$
$$Y(k+1) = F[I - K(k+1)H]Y(k)$$
$$S(k+1) = S(k) + S(k)Y^T(k)H^TO^{-1}(k)HY(k)S(k)$$
$$P(k+1/k) = P(k/k-1) + Y(k)S(k)Y^T(k)$$

for $k = 0,1,\ldots$, with initial conditions
$P(0/-1) = P_0$
$O(0) = HP(0/-1)H^T + R$
$K(0) = P(0/-1)H^TO^{-1}(0)$
$Y(0)$ and $S(0)$ are derived by factoring

$$\delta P(0) = Y(0)S(0)Y^T(0) = P(1/0) - P(0/-1) = Q + FP(0/-1)F^T - FK(0)O(0)K^T(0)F^T - P(0/-1)$$

The easiest initialization is proposed, $P(0/-1) = P_0 = 0$, as then the dimensions of $Y(0)$ and $S(0)$ can be helpfully low, [1]. In this case:
$P(0/-1) = P_0 = 0$
$O(0) = R$
$K(0) = 0$
$Y(0)$ and $S(0)$ are derived by factoring $\delta P(0) = Y(0)S(0)Y^T(0) = Q$

In particular, Chandrasekhar-type algorithms may be more attractive computationally for $P(0/-1) = P_0 = 0$, [1].

In the case where $P(0/-1) = P_0 = 0$ and Q has full rank, we get $Y(0) = I$ and $S(0) = Q$.

Chandrasekhar-type algorithm – version 2

$$O(k+1) = O(k) + HY(k)S(k)Y^T(k)H^T$$
$$Y(k+1) = F[I - K(k)H]Y(k)$$
$$S(k+1) = S(k) - S(k)Y^T(k)H^TO^{-1}(k+1)HY(k)S(k)$$
$$K(k+1) = K(k) + F^{-1}Y(k+1)S(k)Y^T(k)H^TO^{-1}(k+1)$$
$$P(k+1/k) = P(k/k-1) + Y(k)S(k)Y^T(k)$$

for $k = 0,1,\ldots$, with initial conditions
$P(0/-1) = P_0$
$O(0) = HP(0/-1)H^T + R$
$K(0) = P(0/-1)H^TO^{-1}(0)$
$Y(0)$ and $S(0)$ are derived by factoring

$$\delta P(0) = Y(0)S(0)Y^T(0) = P(1/0) - P(0/-1) = Q + FP(0/-1)F^T - FK(0)O(0)K^T(0)F^T - P(0/-1)$$

It is known that if $S(0) \geq 0$, then the version 1 is preferred, while if $S(0) \leq 0$, then the version 2 is preferred. In addition, if the initial condition $P(0/-1)$ is equal to the solution of the algebraic Lyapunov equation (10), then $S(0) \leq 0$, [1].

Remark 1.
The Lyapunov equation is a special case of the Riccati equation in the infinite measurement noise covariance case, where $R = \infty$. Then, form both the above two versions of Chandrasekhar-type algorithms we get the Chandrasekhar-type algorithm for the Lyapunov equation:

Chandrasekhar-type algorithm – Lyapunov equation

$$\Psi(k+1) = F\Psi(k)$$
$$P(k+1/k) = P(k/k-1) + \Psi(k)\Psi^T(k)$$

for $k = 0,1,\ldots$, with initial conditions
$P(0/-1) = P_0$
$Y(0)$ is derived by factoring

$$\delta P(0) = \Psi(0)\Psi^T(0) = P(1/0) - P(0/-1) = Q + FP(0/-1)F^T - P(0/-1)$$

Remark 2.
Chandrasekhar-type algorithms can be applied to compute the steady state limiting solution of the Riccati equation. In this case, Chandrasekhar-type algorithms are implemented for $k = 0,1,\ldots$, until $\|P(k+1/k) - P(k/k-1)\| < \varepsilon$, where ε is the convergence criterion and $\|M\|$ denotes the norm of the matrix M.

3 Chandrasekhar-type Algorithms with Gain Elimination

The basic idea is to eliminate the Kalman filter gain from the equations of Chandrasekhar-type algorithms, working as in [17].

This can be achieved by defining the ratio $\Lambda(k)$ (the term Ratio corresponds to the Greek term 'Λόγος'):

$$\Lambda(k) = [I - K(k)H)]^{-1}K(k) \qquad (14)$$

In this section we are going to develop two Chandrasekhar-type algorithms with gain elimination that correspond to the two versions of the traditional Chandrasekhar-type algorithms of the previous section; we refer to these algorithms as Chandrasekhar-type algorithm with gain elimination – version 1 and Chandrasekhar-type algorithm with gain elimination – version 2.

Chandrasekhar-type algorithm with gain elimination – version 1

$$
\begin{aligned}
&O(k+1) = O(k) + HY(k)S(k)Y^T(k)H^T \\
&\Lambda(k+1) = \Lambda(k) + Y(k)S(k)Y^T(k)H^T R^{-1} \\
&Y(k+1) = F[I + \Lambda(k+1)H]^{-1}Y(k) \\
&S(k+1) = S(k) + S(k)Y^T(k)H^T O^{-1}(k)HY(k)S(k) \\
&P(k+1/k) = P(k/k-1) + Y(k)S(k)Y^T(k)
\end{aligned}
$$

for $k = 0,1, \dots$, with initial conditions
$P(0/-1) = P_0$
$O(0) = HP(0/-1)H^T + R$
$\Lambda(0) = P(0/-1)H^T R^{-1}$
$Y(0)$ and $S(0)$ are derived by factoring

$$
\begin{aligned}
\delta P(0) = Y(0)S(0)Y^T(0) &= P(1/0) - P(0/-1) \\
&= Q + FP(0/-1)F^T \\
&\quad - FK(0)O(0)K^T(0)F^T - P(0/-1)
\end{aligned}
$$

Proof.
For the Kalman filter gain, from (3) and (4) we get:

$$
\begin{aligned}
&K(k) = P(k/k-1)H^T[HP(k/k-1)H^T + R]^{-1} \\
&\Rightarrow K(k)HP(k/k-1)H^T + K(k)R(k) \\
&\qquad = P(k/k-1)H^T \\
&\Rightarrow K(k)HP(k/k-1)H^T + K(k) \\
&\qquad = P(k/k-1)H^T R^{-1} \\
&\Rightarrow K(k) = [I - K(k)H]P(k/k-1)H^T R^{-1}
\end{aligned}
$$

Then using (14) we derive:

$$\Lambda(k) = P(k/k-1)H^T R^{-1} \tag{15}$$

Then, using (15) and (4) we derive:

$$K(k)O(k) = \Lambda(k)R \tag{16}$$

Then we are able to eliminate the Kalman filter gain from the Chandrasekhar-type algorithm – version 1:

$$
\begin{aligned}
&K(k+1) = \\
&[K(k)O(k) + Y(k)S(k)Y^T(k)H^T]O^{-1}(k+1) \\
&\Rightarrow K(k+1)O(k+1) \\
&\qquad = K(k)O(k) + Y(k)S(k)Y^T(k)H^T \\
&\Rightarrow \Lambda(k+1)R = \Lambda(k)R + Y(k)S(k)Y^T(k)H^T
\end{aligned}
$$

Hence

$$\Lambda(k+1) = \Lambda(k) + Y(k)S(k)Y^T(k)H^T R^{-1} \tag{17}$$

In addition, using (3), (4) and the Matrix Inversion Lemma[1] we get:

$$
\begin{aligned}
&K(k) = P(k/k-1)H^T[HP(k/k-1)H^T + R]^{-1} \\
&\Rightarrow P(k/k-1) - K(k)HP(k/k-1)H^T \\
&= P(k/k-11) - P(k/k-1)H^T \\
&[HP(k/k-1)H^T + R]^{-1}HP(k/k-1) \\
&= [P^{-1}(k/k-1) + H^T R^{-1} H]^{-1} \\
&\Rightarrow [I - K(k)H]P(k/k-1) \\
&\qquad = [P^{-1}(k/k-1) + H^T R^{-1} H]^{-1} \\
&\Rightarrow [I - K(k)H] = [P^{-1}(k/k-1) \\
&\qquad + H^T R^{-1} H]^{-1} P^{-1}(k/k-1) \\
&\Rightarrow [I - K(k)H] = [I + P(k/k-1)H^T R^{-1} H]^{-1}
\end{aligned}
$$

Then using (14) we derive:

$$[I - K(k)H] = [I + \Lambda(k)H]^{-1} \tag{18}$$

Hence, we are able to eliminate the Kalman filter gain from the $Y(k+1)$ equation of the Chandrasekhar-type algorithm – version 1:

$$Y(k+1) = F[I + \Lambda(k+1)H]^{-1}Y(k) \tag{19}$$

It is obvious that equations (17) and (19) substitute equations for $K(k+1)$ and $Y(k+1)$ of the Chandrasekhar-type algorithm – version 1, eliminating the use of Kalman filter gain.

Chandrasekhar-type algorithm with gain elimination – version 2

$$
\begin{aligned}
&O(k+1) = O(k) + HY(k)S(k)Y^T(k)H^T \\
&Y(k+1) = F[I + \Lambda(k)H]^{-1}Y(k) \\
&S(k+1) = S(k) - S(k)Y^T(k)H^T O^{-1}(k+1)HY(k)S(k) \\
&\Lambda(k+1) = \Lambda(k) + Y(k)S(k)Y^T(k)H^T R^{-1} \\
&P(k+1/k) = P(k/k-1) + Y(k)S(k)Y^T(k)
\end{aligned}
$$

for $k = 0,1, \dots$, with initial conditions
$P(0/-1) = P_0$
$O(0) = HP(0/-1)H^T + R$
$\Lambda(0) = P(0/-1)H^T R^{-1}$
$Y(0)$ and $S(0)$ are derived by factoring

$$
\begin{aligned}
\delta P(0) = Y(0)S(0)Y^T(0) &= P(1/0) - P(0/-1) \\
&= Q + FP(0/-1)F^T \\
&\quad - FK(0)O(0)K^T(0)F^T - P(0/-1)
\end{aligned}
$$

Proof.
We are able to eliminate the Kalman filter gain from the $Y(k+1)$ equation of the Chandrasekhar-type algorithm – version 2, by using (18):

$$Y(k+1) = F[I + \Lambda(k)H]^{-1}Y(k) \tag{20}$$

In addition, are able to eliminate the Kalman filter gain from the Chandrasekhar-type algorithm – version 2:

$$
\begin{aligned}
&K(k+1) = K(k) \\
&+F^{-1}Y(k+1)S(k)Y^T(k)H^T O^{-1}(k+1)
\end{aligned}
$$

1

Let A, C be nonsingular matrices. Then,
$[A + BCD]^{-1} = A^{-1} - A^{-1}B[C^{-1} + DA^{-1}B]^{-1}DA^{-1}$

$$K(k+1)O(k+1) = K(k)O(k+1)$$
$$+F^{-1}Y(k+1)S(k)Y^T(k)H^T$$
$$K(k+1)O(k+1)$$
$$= K(k)[O(k) + HY(k)S(k)Y^T(k)H^T]$$
$$+F^{-1}Y(k+1)S(k)Y^T(k)H^T$$
$$K(k+1)O(k+1) = K(k)O(k) + K(k)HY(k)S(k)Y^T(k)H^T + F^{-1}Y(k+1)S(k)Y^T(k)H^T$$
$$\Lambda(k+1)R = \Lambda(k)R + K(k)HY(k)S(k)Y^T(k)H^T + F^{-1}Y(k+1)S(k)Y^T(k)H^T$$
$$\Lambda(k+1)R = \Lambda(k)R + K(k)HY(k)S(k)Y^T(k)H^T + F^{-1}F[I+\Lambda(k)H]^{-1}Y(k)S(k)Y^T(k)H^T$$
$$\Lambda(k+1)R = \Lambda(k)R + K(k)HY(k)S(k)Y^T(k)H^T + [I+\Lambda(k)H]^{-1}Y(k)S(k)Y^T(k)H^T$$
$$\Lambda(k+1)R = \Lambda(k)R + \{K(k)H + [I+\Lambda(k)H]^{-1}\}Y(k)S(k)Y^T(k)H^T$$
$$\Lambda(k+1)R = \Lambda(k)R + \{K(k)H + I - K(k)H\}Y(k)S(k)Y^T(k)H^T$$
$$\Lambda(k+1)R = \Lambda(k)R + Y(k)S(k)Y^T(k)H^T$$

Hence
$$\Lambda(k+1) = \Lambda(k) + Y(k)S(k)Y^T(k)H^TR^{-1} \quad (21)$$

It is obvious that equations (20) and (21) substitute equations for $K(k+1)$ and $Y(k+1)$ of the Chandrasekhar-type algorithm – version 2, eliminating the use of Kalman filter gain.

4 Comparison of the Algorithms

It is established that the Chandrasekhar-type algorithms with gain elimination have been derived from the traditional Chandrasekhar-type algorithms. Thus the traditional as well as the proposed Chandrasekhar-type algorithms are equivalent algorithms concerning their behavior, since they compute theoretically the same prediction error covariances. Since all the algorithms are iterative, it is reasonable to compare the algorithms concerning their per iteration calculation burdens.

Scalar operations are involved in matrix manipulation operations, which are needed for the implementation of the Chandrasekhar-type algorithms. Table 1 summarizes the calculation burden of needed matrix operations. Note that S denotes a symmetric matrix. The details for the multi-dimensional model $n \geq 2, m \geq 2$ are given in [17].

Table 1. Calculation Burden of Matrix Operations

Matrix Operation	Matrix Dimensions	Calculation Burden
$C = A + B$	$(n \times m) + (n \times m)$	nm
$S = A + B$	$(n \times n) + (n \times n)$	$\frac{1}{2}n^2 + \frac{1}{2}n$
$B = I + A$	$(n \times n) + (n \times n)$	n
$C = A \cdot B$	$(n \times m) \cdot (m \times \ell)$	$2nm\ell - n\ell$
$S = A \cdot B$	$(n \times m) \cdot (m \times n)$	$n^2m + nm - \frac{1}{2}n^2 - \frac{1}{2}n$
$B = A^{-1}$	$n \times n$	$\frac{1}{6}(16n^3 - 3n^2 - n)$

The per iteration calculation burdens of the Chandrasekhar-type algorithms for the general multidimensional case, where $n \geq 2, m \geq 2$, are analytically calculated in the Appendix and summarized in Table 2.

Table 2. Calculation Burden of Chandrasekhar-type algorithms

Chandrasekhar-type algorithms	Calculation Burden
traditional (gain use) version1	$CB_{CTA1} = 2n^3 - n^2 + n$ $+\frac{2}{6}(16m^3 - 3m^2 - m)$ $+4n^2m - 2nm + 5nm^2$ $+3n^2r - nr + 2nr^2$ $+2m^2r - mr + r^2m + nmr$
traditional (gain use) version 2	$CB_{CTA2} = 4n^3 - 2n^2 + n$ $+\frac{1}{6}(16m^3 - 3m^2 - m)$ $+6n^2m - 3nm + 5nm^2$ $+3n^2r - nr + 2nr^2$ $+2m^2r - mr + r^2m + nmr$
proposed (gain elimination) version1	$CB_{CTAGE1} = 2n^3 - n^2 + n$ $+\frac{1}{6}(16n^3 - 3n^2 - n)$ $+\frac{1}{6}(16m^3 - 3m^2 - m)$ $+6n^2m - nm + nm^2$ $+3n^2r - nr + 2nr^2$ $+2m^2r - mr + r^2m + nmr$
proposed (gain elimination) version 2	$CB_{CTAGE2} = 2n^3 - n^2 + n$ $+\frac{1}{6}(16n^3 - 3n^2 - n)$ $+\frac{1}{6}(16m^3 - 3m^2 - m)$ $+6n^2m - nm + nm^2$ $+3n^2r - nr + 2nr^2$ $+2m^2r - mr + r^2m + nmr$

From Table 2 we realize that we are able to determine, which Chandrasekhar-type algorithm is faster:

1. Chandrasekhar-type algorithms – version 1

$$CB_{CTA1} - CB_{CTAGE1} = \frac{1}{6}\{16m^3 + (24n-3)m^2 - (12n^2 + 6n + 1)m - (16n^3 - 3n^2 - n)\}$$

The areas (wrt the model dimensions), where the proposed gain elimination algorithm or the traditional algorithm is faster, are shown in Figure 1.

The following Rule of Thumb is derived: The proposed Chandrasekhar-type algorithm with gain elimination is faster than the traditional Chandrasekhar-type algorithm, when $m/n > 0.835$

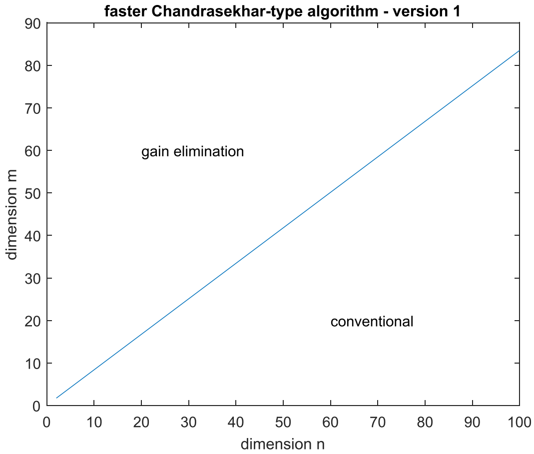

Fig. 1: The faster Chandrasekhar-type algorithm – version 1

2. Chandrasekhar-type algorithms – version 2

$$CB_{CTA2} - CB_{CTAGE2} = \frac{1}{6}n\{24m^2 - 12m - (4n^2 + 3n - 1)\}$$

The areas (wrt the model dimensions) where the proposed gain elimination algorithm or the traditional algorithm is faster are shown in Figure 2.

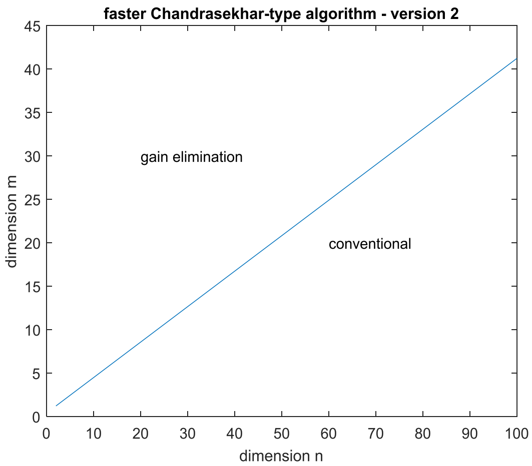

Fig. 2: The faster Chandrasekhar-type algorithm – version 2

The following Rule of Thumb is derived: The proposed Chandrasekhar-type algorithm with gain elimination is faster than the traditional Chandrasekhar-type algorithm, when $m/n > 0.412$.

3. Traditional Chandrasekhar-type algorithms

$$CB_{CTA1} - CB_{CTA2} = \frac{1}{6}\{16m^3 - 3m^2 - (12n^2 - 6n + 1)m - 6(2n^3 - n^2)\}$$

The areas (wrt the model dimensions) where version 1 or version 2 is faster, are shown in Figure 3.
The following Rule of Thumb is derived: version 1 is faster than version 2, when $m/n > 1.1738$.

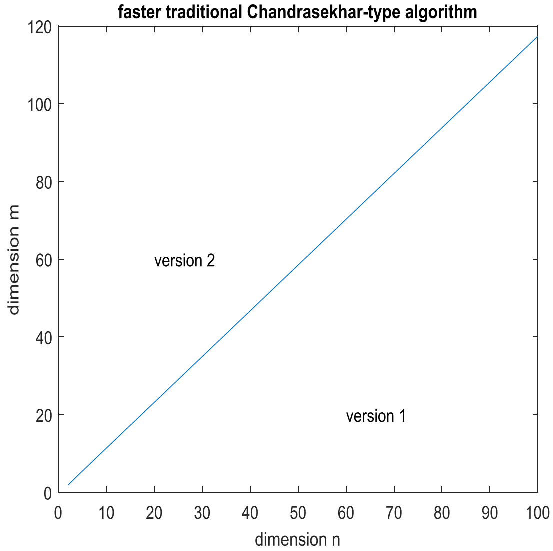

Fig. 3: The faster traditional Chandrasekhar-type algorithm

4. Proposed Chandrasekhar-type algorithms

$CB_{CTAGE1} = CB_{CTAGE2}$

The proposed version 1 is as fast as version 2.

Thus we conclude that which algorithm is faster depends on the state dimension n and the measurement dimension m and not on the dimension r defined in (13). Hence, the knowledge of the system dimensions n and m can determine, which Chandrasekhar-type algorithm is faster.

Finally, the per iteration calculation burdens of the traditional Lyapunov equation and the Chandrasekhar-type algorithm for the Lyapunov equation, are analytically calculated in the Appendix and summarized in Table 3.

Table 3. Calculation Burden of Algorithms for the Lyapunov equation solution

Algorithms	Calculation Burden
traditional	$CB_{LE} = 3n^3$
Chandrasekhar-type	$CB_{CTALE} = 3n^2r$

As in Table 3 appears the Chandrasekhar-type algorithm for the Lyapunov equation is faster than the traditional Lyapunov equation, when $r < n$.

Example. Consider the system dimensions $n = 6, m = 3$ for estimation of three-dimensional radar tracking [18]. Then $CB_{CTA2} - CB_{CTAGE2} > 0$ and hence the proposed Chandrasekhar-type algorithm – version 2 is faster than the traditional one.

5 Conclusions

In this paper, new variations of Chandrasekhar-type algorithms eliminating the Kalman filter gain are proposed. The calculation burdens of the Chandrasekhar-type algorithms are derived. The proposed Chandrasekhar-type algorithms may be faster than the traditional ones, depending on the model dimensions. It has been shown that the determination of the faster Chandrasekhar-type algorithm can be achieved via the system dimensions.

A subject of future research is to investigate the application of corresponding Chandrasekhar-type algorithms to dynamical continuous-time systems, [19], [20], [21], and to discrete-time anti-linear systems, [22]. Another area of future research may be the use of the derived Chandrasekhar-type algorithms with gain elimination in the derivation of time varying, time invariant, and steady state Kalman filters.

References:

[1] B. D .O Anderson, J. B. Moore, *Optimal Filtering*, Dover Publications, New York, 2005.

[2] R. E. Kalman, A new approach to linear filtering and prediction problems, *J. Bas. Eng., Trans. ASME*, Ser. D, vol. 8(1), 1960, pp. 34-45.

[3] N. Komaroff, Iterative matrix bounds and computational solutions to the discrete algebraic Riccati equation, *IEEE Trans. Autom. Control*, vol. 39, 1994, pp. 1676–1678.

[4] L. Wang, An improved iterative method for solving the discrete algebraic Riccati equation, *Mathematical Problems in Engineering*, vol. 2020, Article ID 3283157, 6 pages, https://doi.org/10.1155/2020/3283157.

[5] J. Zhang and J. Liu, New upper and lower bounds, the iteration algorithm for the solution of the discrete algebraic Riccati equation, Advances in Difference Equations, vol. 313, 2015, pp. 1-17.

[6] B. Zhou, On Linear Quadratic Optimal Control of Discrete-Time Complex-Valued Linear Systems, *Optimal Control Applications and Methods*, DOI: 10.1002/oca.2554, 2017.

[7] J. Liu, Z. Wang, and Z. Xie, Iterative algorithms for reducing inversion of discrete algebraic Riccati matrix equation, *IMA Journal of Mathematical Control and Information*, vol. 39, 2022, pp. 985–1007.

[8] M. Morf, G. S. Sidhu, T. Kailath, Some New Algorithms for Recursive Estimation in Constant, Linear, Discrete-time Systems, *IEEE Trans. Automatic Control*, vol. AC-19, no. 4, 1974, pp. 315–323.

[9] N. Assimakis, A. Kechriniotis, S. Voliotis, F. Tassis, M. Kousteri, Analysis of the time invariant Kalman filter implementation via general Chandrasekhar algorithm, *International Journal of Signal and Imaging Systems Engineering*, vol. 1(1), 2008, pp. 51-57.

[10] 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*, vol. 151(2), 2004, pp. 315-325, https://doi.org/10.1016/S0096-3003(03)00343-6.

[11] J.S. Baras and D.G. Lainiotis, Chandrasekhar algorithms for linear time varying distributed systems, *Information Sciences*, vol. 17(2), 1979, pp. 153-167, https://doi.org/10.1016/0020-0255(79)90037-9.

[12] J.U. Sevinov, S.O. Zaripova, A stable iterative algorithm for estimating the elements of the matrix gain of a Kalman filter, *Electrical and Computer Engineering, Technical science and innovation*, no 3, 2023, pp. 99-103, https://scienceweb.uz/publication/15804.

[13] M.T. Augustine, A note on linear quadratic regulator and Kalman filter, 2023, DOI: 10.48550/arXiv.2308.15798.

[14] A. Guven and C. Hajiyev, Two-Stage Kalman Filter Based Estimation of Boeing 747 Actuator/Control Surface Stuck Faults, *WSEAS Transactions on Signal Processing*, vol. 19, 2023, pp. 32-40, https://doi.org/10.37394/232014.2023.19.4.

[15] C. Hajiyev and U. Hacizade, A Covariance Matching-Based Adaptive Measurement Differencing Kalman Filter for INS's Error

Compensation, *WSEAS Transactions on Systems and Control*, vol. 18, 2023, pp. 478-486, https://doi.org/10.37394/23203.2023.18.51.

[16] 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, 2021, pp. 294-301, doi: 10.37394/23203.2021.16.25

[17] N. Assimakis, Kalman Filter Gain Elimination in Linear Estimation, *International Journal of Computer and Information Engineering*, vol. 14(7), 2020, pp. 236-241.

[18] P. Aditya, E. Apriliani, D. K. Arif and K. Baihaqi, Estimation of three-dimensional radar tracking using modified extended Kalman filter, *Journal of Physics: Conf. Series* 974, 2018, doi :10.1088/1742-6596/974/1/012071

[19] Z.-P. Jiang, T. Bian and W. Gao, Learning-Based Control: A Tutorial and Some Recent Results, *Foundations and Trends in Systems and Control*, vol. 8(3), 2022, pp. 985–1007, (176-284).

[20] J. Liu, Li Wang and Y. Bai, New estimates of upper bounds for the solutions of the continuous algebraic Riccati equation and the redundant control inputs problems, *Automatica*, vol. 116, 2020, 108936, https://doi.org/10.1016/j.automatica.2020.108936.

[21] T. Simos, V. Katsikis, S. Mourtas and P. Stanimirović, Unique non-negative definite solution of the time-varying algebraic Riccati equations with applications to stabilization of LTV systems, *Mathematics and Computers in Simulation*, vol. 202, 2022, pp. 164-180, https://doi.org/10.1016/j.matcom.2022.05.033

[22] C.-Y. Chiang and H.-Y. Fan, Inheritance properties of the conjugate discrete-time algebraic Riccati equation, *Linear Algebra and its Applications*, vol. 683, 2024, pp. 71-97, https://doi.org/10.1016/j.laa.2023.11.011.

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

The authors equally 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 authors have no conflicts of interest to declare.

DOI: 10.37394/23203.2023.18.65

APPENDIX

The per iteration calculation burdens of the Chandrasekhar-type algorithms for the general multidimensional case, where $n \geq 2, m \geq 2$, are analytically calculated in Table 4, Table 5, Table 6 and Table 7. The per iteration calculation burdens of the traditional Lyapunov equation and the Chandrasekhar-type algorithm for the Lyapunov equation, are analytically calculated in Table 8 and Table 9.

Table 4. Chandrasekhar-type algorithm – version 1

Matrix Operation	Calculation Burden
$W_1(k) = Y(k)S(k)$	$2nr^2 - nr$
$W_2(k) = W_1(k)Y^T(k)$	$n^2r + nr - \frac{1}{2}n^2 - \frac{1}{2}n$
$W_3(k) = W_2(k)H^T$	$2n^2m - nm$
$W_4(k) = HW_3(k)$	$nm^2 + nm - \frac{1}{2}m^2 - \frac{1}{2}m$
$O(k+1) = O(k) + W_4(k)$	$\frac{1}{2}m^2 + \frac{1}{2}m$
$K(k)O(k)$	$2nm^2 - nm$
$W_5(k) = K(k)O(k) + W_3(k)$	nm
$O^{-1}(k+1)$	$\frac{1}{6}(16m^3 - 3m^2 - m)$
$K(k+1) = W_5(k)O^{-1}(k+1)$	$2nm^2 - nm$
$K(k+1)H$	$2n^2m - nm$
$I - K(k+1)H$	n
$F[I - K(k+1)H]$	$2n^3 - n^2$
$Y(k+1) = F[I - K(k+1)H]Y(k)$	$2n^2r - nr$
$O^{-1}(k)$	$\frac{1}{6}(16m^3 - 3m^2 - m)$
$W_6(k) = HW_1(k)$	$nmr - mr$
$W_7(k) = O^{-1}(k)W_6(k)$	$2m^2r - mr$
$W_8(k) = W_6^T(k)W_7(k)$	$r^2m + rm - \frac{1}{2}r^2 - \frac{1}{2}r$
$S(k+1) = S(k) + W_8(k)$	$\frac{1}{2}r^2 + \frac{1}{2}r$
$P(k+1/k) = P(k/k-1) + W_2(k)$	$\frac{1}{2}n^2 + \frac{1}{2}n$
$CB_{CTA1} = 2n^3 - n^2 + n + \frac{2}{6}(16m^3 - 3m^2 - m)$ $+4n^2m - 2nm + 5nm^2$ $+3n^2r - nr + 2nr^2 + 2m^2r - mr + r^2m + nmr$	

Table 5. Chandrasekhar-type algorithm with gain elimination – version 1

Matrix Operation	Calculation Burden
$W_1(k) = Y(k)S(k)$	$2nr^2 - nr$
$W_2(k) = W_1(k)Y^T(k)$	$n^2r + nr - \frac{1}{2}n^2 - \frac{1}{2}n$
$W_3(k) = W_2(k)H^T$	$2n^2m - nm$
$W_4(k) = HW_3(k)$	$nm^2 + nm - \frac{1}{2}m^2 - \frac{1}{2}m$
$O(k+1) = O(k) + W_4(k)$	$\frac{1}{2}m^2 + \frac{1}{2}m$
$W_5(k) = W_2(k)[H^TR^{-1}]$	$2n^2m - nm$
$\Lambda(k+1) = \Lambda(k) + W_5(k)$	nm
$\Lambda(k+1)H$	$2n^2m - nm$
$I + \Lambda(k+1)H$	n
$[I + \Lambda(k+1)H]^{-1}$	$\frac{1}{6}(16n^3 - 3n^2 - n)$
$F[I + \Lambda(k+1)H]^{-1}$	$2n^3 - n^2$
$Y(k+1) = F[I + \Lambda(k+1)H]^{-1}Y(k)$	$2n^2r - nr$
$O^{-1}(k)$	$\frac{1}{6}(16m^3 - 3m^2 - m)$
$W_6(k) = HW_1(k)$	$nmr - mr$
$W_7(k) = O^{-1}(k)W_6(k)$	$2m^2r - mr$
$W_8(k) = W_6^T(k)W_7(k)$	$r^2m + rm - \frac{1}{2}r^2 - \frac{1}{2}r$
$S(k+1) = S(k) + W_8(k)$	$\frac{1}{2}r^2 + \frac{1}{2}r$
$P(k+1/k) = P(k/k-1) + W_2(k)$	$\frac{1}{2}n^2 + \frac{1}{2}n$
$CB_{CTAGE1} = 2n^3 - n^2 + n + \frac{1}{6}(16n^3 - 3n^2 - n)$ $+\frac{1}{6}(16m^3 - 3m^2 - m) + 6n^2m - nm + nm^2$ $+3n^2r - nr + 2nr^2 + 2m^2r - mr + r^2m + nmr$	

Table 6. Chandrasekhar-type algorithm – version 2

Matrix Operation	Calculation Burden
$W_1(k) = Y(k)S(k)$	$2nr^2 - nr$
$W_2(k) = W_1(k)Y^T(k)$	$n^2r + nr - \frac{1}{2}n^2 - \frac{1}{2}n$
$W_3(k) = W_2(k)H^T$	$2n^2m - nm$
$W_4(k) = HW_3(k)$	$nm^2 + nm - \frac{1}{2}m^2 - \frac{1}{2}m$
$O(k+1) = O(k) + W_4(k)$	$\frac{1}{2}m^2 + \frac{1}{2}m$
$O^{-1}(k+1)$	$\frac{1}{6}(16m^3 - 3m^2 - m)$
$K(k)H$	$2n^2m - nm$
$I - K(k)H$	n
$F[I - K(k)H]$	$2n^3 - n^2$
$Y(k+1) = F[I - K(k)H]Y(k)$	$2n^2r - nr$
$W_5(k) = HW_1(k)$	$nmr - mr$
$W_6(k) = O^{-1}(k+1)W_5(k)$	$2m^2r - mr$
$W_7(k) = W_5^T(k)W_6(k)$	$r^2m + rm - \frac{1}{2}r^2 - \frac{1}{2}r$
$S(k+1) = S(k) - W_7(k)$	$\frac{1}{2}r^2 + \frac{1}{2}r$
$K(k)O(k)$	$2nm^2 - nm$
$W_8(k) = W_5^T(k)O^{-1}(k+1)$	$2nm^2 - nm$
$W_9(k) = F^{-1}Y(k+1)$	$2n^3 - n^2$
$W_{10}(k) = W_9(k)W_8(k)$	$2n^2m - nm$
$K(k+1) = K(k) + W_{10}(k)$	nm
$P(k+1/k) = P(k/k-1) + W_2(k)$	$\frac{1}{2}n^2 + \frac{1}{2}n$
$CB_{CTA2} = 4n^3 - 2n^2 + n + \frac{1}{6}(16m^3 - 3m^2 - m)$ $+6n^2m - 3nm + 5nm^2$ $+3n^2r - nr + 2nr^2 + 2m^2r - mr + r^2m + nmr$	

Table 7. Chandrasekhar-type algorithm with gain elimination – version 2

Matrix Operation	Calculation Burden
$W_1(k) = Y(k)S(k)$	$2nr^2 - nr$
$W_2(k) = W_1(k)Y^T(k)$	$n^2r + nr - \frac{1}{2}n^2 - \frac{1}{2}n$
$W_3(k) = W_2(k)H^T$	$2n^2m - nm$
$W_4(k) = HW_3(k)$	$nm^2 + nm - \frac{1}{2}m^2 - \frac{1}{2}m$
$O(k+1) = O(k) + W_4(k)$	$\frac{1}{2}m^2 + \frac{1}{2}m$
$W_5(k) = W_2(k)[H^T R^{-1}]$	$2n^2m - nm$
$\Lambda(k+1) = \Lambda(k) + W_5(k)$	nm
$\Lambda(k)H$	$2n^2m - nm$
$I + \Lambda(k)H$	n
$[I + \Lambda(k)H]^{-1}$	$\frac{1}{6}(16n^3 - 3n^2 - n)$
$F[I + \Lambda(k)H]^{-1}$	$2n^3 - n^2$
$Y(k+1) = F[I + \Lambda(k)H]^{-1}Y(k)$	$2n^2r - nr$
$O^{-1}(k+1)$	$\frac{1}{6}(16m^3 - 3m^2 - m)$
$W_6(k) = HW_1(k)$	$nmr - mr$
$W_7(k) = O^{-1}(k+1)W_6(k)$	$2m^2r - mr$
$W_8(k) = W_6^T(k)W_7(k)$	$r^2m + rm - \frac{1}{2}r^2 - \frac{1}{2}r$
$S(k+1) = S(k) - W_8(k)$	$\frac{1}{2}r^2 + \frac{1}{2}r$
$P(k+1/k) = P(k/k-1) + W_2(k)$	$\frac{1}{2}n^2 + \frac{1}{2}n$
$CB_{CTAGE2} = 2n^3 - n^2 + n + \frac{1}{6}(16n^3 - 3n^2 - n) + \frac{1}{6}(16m^3 - 3m^2 - m) + 6n^2m - nm + nm^2 + 3n^2r - nr + 2nr^2 + 2m^2r - mr + r^2m + nmr$	

Table 8. Lyapunov equation

Matrix Operation	Calculation Burden
$FP(k/k-1)$	$2n^3 - n^2$
$FP(k/k-1)F^T$	$n^3 + \frac{1}{2}n^2 - \frac{1}{2}n$
$P(k+1/k) = Q + FP(k/k-1)F^T$	$\frac{1}{2}n^2 + \frac{1}{2}n$
$CB_{LE} = 3n^3$	

Table 9. Chandrasekhar-type algorithm – Lyapunov equation

Matrix Operation	Calculation Burden
$\Psi(k+1) = F\Psi(k)$	$2n^2r - nr$
$\Psi(k)\Psi^T(k)$	$n^2r + nr - \frac{1}{2}n^2 - \frac{1}{2}n$
$P(k+1/k) = P(k/k-1) + \Psi(k)\Psi^T(k)$	$\frac{1}{2}n^2 + \frac{1}{2}n$
$CB_{LE} = 3n^2r$	