Hindawi Publishing Corporation
ISRN Applied Mathematics
Volume 2014, Article ID 417623, 10 pages
http://dx.doi.org/10.1155/2014/417623

Research Article

Iterative and Algebraic Algorithms for the Computation of the Steady State Kalman Filter Gain

Nicholas Assimakis[1,2] **and Maria Adam**[2]

[1] *Department of Electronic Engineering, Technological Educational Institute of Central Greece, 3rd km Old National Road Lamia-Athens, 35100 Lamia, Greece*
[2] *Department of Computer Science and Biomedical Informatics, University of Thessaly, 2-4 Papasiopoulou Street, 35100 Lamia, Greece*

Correspondence should be addressed to Maria Adam; madam@dib.uth.gr

Received 24 February 2014; Accepted 31 March 2014; Published 4 May 2014

Academic Editors: F. Ding, L. Guo, and H. C. So

The Kalman filter gain arises in linear estimation and is associated with linear systems. The gain is a matrix through which the estimation and the prediction of the state as well as the corresponding estimation and prediction error covariance matrices are computed. For time invariant and asymptotically stable systems, there exists a steady state value of the Kalman filter gain. The steady state Kalman filter gain is usually derived via the steady state prediction error covariance by first solving the corresponding Riccati equation. In this paper, we present iterative per-step and doubling algorithms as well as an algebraic algorithm for the steady state Kalman filter gain computation. These algorithms hold under conditions concerning the system parameters. The advantage of these algorithms is the autonomous computation of the steady state Kalman filter gain.

1. Introduction

The Kalman filter gain arises in Kalman filter equations in linear estimation and is associated with linear systems. State space systems have been widely used in estimation theory to describe discrete time systems [1–5]. It is known [1] for time invariant systems that if the signal process model is asymptotically stable, then there exists a steady state value of the Kalman filter gain. Thus, the steady state gain is associated with time invariant systems described by the following state space equations:

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

for $k \geq 0$, where x_k is the n-dimensional state vector at time k, z_k is the m-dimensional measurement vector at time k, F is the $n\times n$ system transition matrix, H is the $m\times n$ output matrix, w_k is the plant noise at time k, and v_k is the measurement noise at time k. Also, $\{w_k\}$ and $\{v_k\}$ are Gaussian zero-mean white random processes with covariance matrices Q and R, respectively.

The discrete time Kalman filter [1, 6] is the most well-known algorithm that solves the filtering problem. In fact, Kalman filter faces simultaneously two problems as follows.

(i) *Estimation*: the aim is to recover at time k information about the state vector at time k using measurements up till time k.

(ii) *Prediction*: the aim is to obtain at time k information about the state vector at time $k+1$ using measurements up till time k; it is clear that prediction is related to the forecasting side of information processing.

Kalman filter uses the measurements up till time k in order to produce the (one step) prediction of the state vector and the corresponding prediction error covariance matrix $P_{k+1/k}$, as well as producing the estimation of the state vector and the corresponding estimation error covariance matrix $P_{k/k}$. The Kalman filter equations needed for the computation of the prediction and estimation error covariance matrices are as follows:

$$K_k = P_{k/k-1}H^T\left[HP_{k/k-1}H^T + R\right]^{-1}, \tag{2}$$

$$P_{k/k} = \left(I - K_kH\right)P_{k/k-1}, \tag{3}$$

$$P_{k+1/k} = Q + FP_{k/k}F^T, \tag{4}$$

for $k \geq 0$, with initial condition $P_{0/-1} = P_0$ for the time instant, where there are no measurements given. Note that K_k is the *Kalman filter gain*.

From (2) to (4), we are able to derive the *Riccati equation*, which is an iterative equation with respect to the prediction error covariance:

$$\begin{aligned} P_{k+1/k} = {} & Q + FP_{k/k-1}F^T \\ & - FP_{k/k-1}H^T\left[HP_{k/k-1}H^T + R\right]^{-1}HP_{k/k-1}F^T. \end{aligned} \tag{5}$$

In the general case, where R and P_0 are positive definite matrices, using in (5) the matrix inversion lemma:

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B\left[C^{-1} + DA^{-1}B\right]^{-1}DA^{-1}, \tag{6}$$

the Riccati equation is formulated as

$$P_{k+1/k} = Q + F\left[P_{k/k-1}^{-1} + H^TR^{-1}H\right]^{-1}F^T. \tag{7}$$

The Riccati equation is a nonlinear iterative equation with respect to the prediction error covariance. For time invariant systems, it is well known [1] that if the signal process model is asymptotically stable, then there exists a steady state value $\widetilde{P}_p$ of the prediction error covariance matrix. In fact, the prediction error covariance tends to the steady state prediction error covariance.

The steady state prediction error covariance satisfies the *steady state Riccati equation*

$$\widetilde{P}_p = Q + F\left[\widetilde{P}_p^{-1} + H^TR^{-1}H\right]^{-1}F^T. \tag{8}$$

Then, from (2), it is clear that there also exists a steady state value $\widetilde{K}$ of the Kalman filter gain [7]. The steady state gain can be calculated by

$$\widetilde{K} = \widetilde{P}_pH^T\left[H\widetilde{P}_pH^T + R\right]^{-1}. \tag{9}$$

Also, from (3), it is clear that there also exists a steady state value $\widetilde{P}_e$ of the estimation error covariance matrix [7], which can be calculated by

$$\widetilde{P}_e = \widetilde{P}_p - \widetilde{P}_pH^T\left[H\widetilde{P}_pH^T + R\right]^{-1}H\widetilde{P}_p = \left[\widetilde{P}_p^{-1} + H^TR^{-1}H\right]^{-1}. \tag{10}$$

It is obvious from (9) that the steady state Kalman filter gain can be derived via the steady state prediction error covariance. The covariance matrix in Kalman filter plays an important role in many applications [1, 4, 6, 8–10]. The steady state prediction error covariance can be derived by solving the Riccati equation emanating from Kalman filter. The discrete time Riccati equation has attracted recent attention. In view of the importance of the Riccati equation, there exists considerable literature on its algebraic solutions; for example, in [1, 7, 11, 12], the authors have derived an eigenvector solution, while the author of [13] has included solving scalar polynomials. Other methods are based on the iterative solutions [1, 13–18] concerning per-step or doubling algorithms. The iterative algorithms that provide the steady state Kalman filter gain together with the prediction error covariance are the Chandrasekhar algorithms [1], as well as the iterative algorithm that calculate the Kalman gain only once for a period of the stationary channel, as opposed to each data sample in the conventional filter [19]. A geometric illustration of the Kalman filter gain is given in [20].

In this paper, we present algorithms for the steady state Kalman filter gain autonomous computation. These algorithms hold under conditions concerning the system parameters. The paper is organized as follows: two new per-step iterative algorithms, a new doubling iterative algorithm, and an algebraic algorithm for the computation of the steady state Kalman filter gain are presented in Section 2. In Section 3, two examples verify the results. Finally, Section 4 summarizes the conclusions.

2. New Algorithms for the Steady State Kalman Filter Computation

2.1. Assumptions. We assume the general case, where R and P_0 are positive definite matrices.

The Kalman filter gain K_k is a matrix of dimension $n \times m$. We define the matrix

$$G_k = K_kH. \tag{11}$$

It is clear that G_k is a nonsymmetric matrix of dimension $n{\times}n$.

It is also clear that there exists a steady state value

$$\widetilde{G} = \widetilde{K}H. \tag{12}$$

Also, we define the matrix

$$S = H^TR^{-1}H. \tag{13}$$

Note that S is an $n \times n$ symmetric positive semidefinite matrix and S is a positive definite if $\operatorname{rank}(H) = n$; this means that S is a nonsingular matrix in the case, $\operatorname{rank}(H) = n$ with $m \geq n$, [21].

2.2. Indirect Steady State Kalman Filter Gain Computation. In this section, we present algorithms for $\widetilde{G}$ computation. Then, we show how to compute the steady state Kalman filter $\widetilde{K}$ through $\widetilde{G} = \widetilde{K}H$.

2.2.1. Iterative Algorithms for $\widetilde{G}$ Computation. In this section, we present two iterative per-step algorithms and an iterative doubling algorithm for $\widetilde{G}$ computation.

Per-Step Iterative Algorithm 1. Using (2) and (11), it is derived that

$$
\begin{aligned}
G_k &= K_k H \\
&= P_{k/k-1} H^T \left[H P_{k/k-1} H^T + R \right]^{-1} H \\
&= P_{k/k-1} H^T \left[H P_{k/k-1} H^T + R \right]^{-1} R R^{-1} H \\
&= P_{k/k-1} H^T \Big(\left[H P_{k/k-1} H^T + R \right]^{-1} \\
&\qquad \times \left[H P_{k/k-1} H^T + R - H P_{k/k-1} H^T \right] \Big) R^{-1} H \\
&= P_{k/k-1} H^T \Big(\left[H P_{k/k-1} H^T + R \right]^{-1} \left[H P_{k/k-1} H^T + R \right] \\
&\qquad - \left[H P_{k/k-1} H^T + R \right]^{-1} H P_{k/k-1} H^T \Big) R^{-1} H \\
&= P_{k/k-1} H^T \left(I - \left[H P_{k/k-1} H^T + R \right]^{-1} H P_{k/k-1} H^T \right) R^{-1} H \\
&= P_{k/k-1} H^T R^{-1} H \\
&\quad - P_{k/k-1} H^T \left[H P_{k/k-1} H^T + R \right]^{-1} H P_{k/k-1} H^T R^{-1} H \\
&= \left(P_{k/k-1} - P_{k/k-1} H^T \left[H P_{k/k-1} H^T + R \right]^{-1} H P_{k/k-1} \right) \\
&\quad \times H^T R^{-1} H \\
&= \left[P_{k/k-1}^{-1} + H^T R^{-1} H \right]^{-1} H^T R^{-1} H \\
&= \left[P_{k/k-1}^{-1} + S \right]^{-1} S.
\end{aligned}
\tag{14}
$$

Thus, arises

$$
K_k H = \left[P_{k/k-1}^{-1} + S \right]^{-1} S. \tag{15}
$$

Using the Riccati equation (7), (15), the nonsingularity of S, and some algebra we have

$$
\begin{aligned}
P_{k+1/k} &= Q + F \left[P_{k/k-1}^{-1} + H^T R^{-1} H \right]^{-1} F^T \\
&= Q + F \left[P_{k/k-1}^{-1} + S \right]^{-1} F^T = Q + F K_k H S^{-1} F^T \\
&\Longrightarrow P_{k+1/k} S = Q S + F K_k H S^{-1} F^T S \\
&= Q F^{-T} S \left[S^{-1} F^{-T} S \right]^{-1} + F K_k H \left[S^{-1} F^{-T} S \right]^{-1} \\
&\Longrightarrow P_{k+1/k} S \left(S^{-1} F^{-T} S \right) = Q F^{-T} S + F K_k H \\
&\Longrightarrow S^{-1} F^{-T} S = S^{-1} P_{k+1/k}^{-1} \left(Q F^{-T} S + F K_k H \right) \\
&\Longrightarrow K_{k+1} H \left(S^{-1} F^{-T} S \right) \\
&= K_{k+1} H S^{-1} P_{k+1/k}^{-1} \left(Q F^{-T} S + F K_k H \right).
\end{aligned}
\tag{16}
$$

Also, from (2) and (13), we can write

$$
\begin{aligned}
K_k \left(H P_{k/k-1} H^T + R \right) &= P_{k/k-1} H^T \\
&\Longrightarrow K_k H P_{k/k-1} H^T + K_k R = P_{k/k-1} H^T \\
&\Longrightarrow K_k R = P_{k/k-1} H^T - K_k H P_{k/k-1} H^T \\
&\Longrightarrow K_k R = \left(I - K_k H \right) P_{k/k-1} H^T \\
&\Longrightarrow K_k = \left(I - K_k H \right) P_{k/k-1} H^T R^{-1} \\
&\Longrightarrow K_k H = \left(I - K_k H \right) P_{k/k-1} H^T R^{-1} H \\
&\Longrightarrow K_k H = \left(I - K_k H \right) P_{k/k-1} S.
\end{aligned}
\tag{17}
$$

Since the matrices $P_{k/k-1}, S$ are nonsingular, the last equation yields

$$
K_k H S^{-1} P_{k/k-1}^{-1} = I - K_k H. \tag{18}
$$

Substituting in (16) the matrix $K_{k+1} H S^{-1} P_{k+1/k}^{-1}$ by (18), it follows

$$
\begin{aligned}
K_{k+1} H \left(S^{-1} F^{-T} S \right) &= K_{k+1} H S^{-1} P_{k+1/k}^{-1} \left(Q F^{-T} S + F K_k H \right) \\
&= \left(I - K_{k+1} H \right) \left(Q F^{-T} S + F K_k H \right) \\
&= Q F^{-T} S + F K_k H - K_{k+1} H Q F^{-T} S \\
&\quad - K_{k+1} H F K_k H,
\end{aligned}
\tag{19}
$$

whereby it is implied that

$$
\begin{aligned}
&K_{k+1} H \left(S^{-1} F^{-T} S \right) + K_{k+1} H Q F^{-T} S + K_{k+1} H F K_k H \\
&\quad = Q F^{-T} S + F K_k H \\
&\quad \Longrightarrow K_{k+1} H \left(S^{-1} F^{-T} S + Q F^{-T} S + F K_k H \right) \\
&\quad = Q F^{-T} S + F K_k H \\
&\quad \Longrightarrow K_{k+1} H = \left(Q F^{-T} S + F K_k H \right) \\
&\qquad \times \left[S^{-1} F^{-T} S + Q F^{-T} S + F K_k H \right]^{-1}.
\end{aligned}
\tag{20}
$$

Thus, the above equation can be written as

$$
K_{k+1} H = \left(Q F^{-T} S + F K_k H \right) \left[S^{-1} F^{-T} S + Q F^{-T} S + F K_k H \right]^{-1}. \tag{21}
$$

Combining (21) with (11), the following nonlinear iterative equation with respect to G_k is derived:

$$
\begin{aligned}
G_{k+1} &= \left(\left(Q F^{-T} S \right) + (F) G_k \right) \\
&\qquad \times \left[\left(S^{-1} F^{-T} S + Q F^{-T} S \right) + (F) G_k \right]^{-1} \\
&= \left(C + D G_k \right) \left[A + B G_k \right]^{-1},
\end{aligned}
\tag{22}
$$

where

$$\begin{aligned} A &= \left(Q + S^{-1}\right) F^{-T} S, \\ B &= F, \\ C &= Q F^{-T} S, \\ D &= F. \end{aligned} \tag{23}$$

The algorithm uses the initial condition $G_0 = K_0 H = P_0 H^T [H P_0 H^T + R]^{-1} H$. It is known [1] that the prediction error covariance tends to the steady state prediction error covariance and that the convergence is independent of the initial uncertainty, that is, independent of the value of the initial condition P_0. Thus, we are able to assume zero initial condition $P_0 = 0$ and so we are to use the initial condition $G_0 = 0$.

It is clear that G_k tends to a steady state value $\widetilde{G}$ and by (22) $\widetilde{G}$ satisfies

$$\widetilde{G} = \left(C + D\widetilde{G}\right) \left[A + B\widetilde{G}\right]^{-1}. \tag{24}$$

Per-Step Iterative Algorithm 2. We rewrite (22) as

$$\begin{aligned} G_{k+1} = {} & (C + D G_k) \left[A + B G_k\right]^{-1} \\ = {} & \left(C A^{-1} A + \left(D + C A^{-1} B - C A^{-1} B\right) G_k\right) \left[A + B G_k\right]^{-1} \\ = {} & \left(C A^{-1} (A + B G_k) + \left(D - C A^{-1} B\right) G_k\right) \left[A + B G_k\right]^{-1} \\ = {} & C A^{-1} + \left(D - C A^{-1} B\right) G_k \left[A + B G_k\right]^{-1} \\ = {} & C A^{-1} + \left(D - C A^{-1} B\right) \left[A G_k^{-1} + B\right]^{-1} \\ = {} & C A^{-1} + \left(D - C A^{-1} B\right) \left[G_k^{-1} + A^{-1} B\right]^{-1} A^{-1}. \end{aligned} \tag{25}$$

Thus, the following nonlinear iterative equation with respect to G_k is derived:

$$G_{k+1} = c + a\left[G_k^{-1} + b\right]^{-1} d, \tag{26}$$

where

$$\begin{aligned} a &= D - C A^{-1} B = F - Q\left[Q + S^{-1}\right]^{-1} F, \\ b &= A^{-1} B = S^{-1} F^T \left[Q + S^{-1}\right]^{-1} F, \\ c &= C A^{-1} = Q\left[Q + S^{-1}\right]^{-1}, \\ d &= A^{-1} = S^{-1} F^T \left[Q + S^{-1}\right]^{-1}. \end{aligned} \tag{27}$$

The algorithm uses the initial condition $G_0 = K_0 H = P_0 H^T [H P_0 H^T + R]^{-1} H$. It is known [1] that the prediction error covariance tends to the steady state prediction error covariance and that the convergence is independent of the initial uncertainty, that is, independent of the value of the initial condition P_0. Thus, we are able to assume zero initial condition $P_0 = 0$. In this case, in order to avoid G_0^{-1}, we are to use the initial condition $G_1 = c$.

It is clear that G_k tends to a steady state value $\widetilde{G}$ and by (26) $\widetilde{G}$ satisfies

$$\widetilde{G} = c + a\left[\widetilde{G}^{-1} + b\right]^{-1} d. \tag{28}$$

Doubling Iterative Algorithm. In (22), setting

$$G_k = K_k H = Y_k X_k^{-1}, \tag{29}$$

we take

$$\begin{aligned} Y_{k+1} X_{k+1}^{-1} &= \left(C + D Y_k X_k^{-1}\right) \left[A + B Y_k X_k^{-1}\right]^{-1} \\ &= \left(C X_k + D Y_k\right) \left[A X_k + B Y_k\right]^{-1} \end{aligned} \tag{30}$$

or

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \end{bmatrix} = \Phi \begin{bmatrix} X_k \\ Y_k \end{bmatrix}, \tag{31}$$

where

$$\Phi = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \left(Q + S^{-1}\right) F^{-T} S & F \\ Q F^{-T} S & F \end{bmatrix} = \begin{bmatrix} d^{-1} & d^{-1} b \\ c d^{-1} & c d^{-1} b + a \end{bmatrix} \tag{32}$$

is a matrix of dimension $2n \times 2n$ and A, B, C, D as in (23).

We are able to use zero initial condition $P_0 = 0$, so $G_0 = K_0 H = Y_0 X_0^{-1} = 0$; that is,

$$\begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix}, \tag{33}$$

and hence

$$\begin{bmatrix} X_k \\ Y_k \end{bmatrix} = \Phi^k \begin{bmatrix} I \\ 0 \end{bmatrix}. \tag{34}$$

We define

$$\Phi^{2^k} = \begin{bmatrix} d_k^{-1} & d_k^{-1} b_k \\ c_k d_k^{-1} & c_k d_k^{-1} b_k + a_k \end{bmatrix} \tag{35}$$

with initial condition

$$\Phi^{2^0} = \begin{bmatrix} d_0^{-1} & d_0^{-1} b_0 \\ c_0 d_0^{-1} & c_0 d_0^{-1} b_0 + a_0 \end{bmatrix} = \Phi = \begin{bmatrix} d^{-1} & d^{-1} b \\ c d^{-1} & c d^{-1} b + a \end{bmatrix}. \tag{36}$$

Then,

$$\begin{bmatrix} X\left(2^k\right) \\ Y\left(2^k\right) \end{bmatrix} = \Phi^{2^k} \begin{bmatrix} I \\ 0 \end{bmatrix}, \tag{37}$$

and, using the doubling principle [1] $\Phi^{2^{k+1}} = \Phi^{2^k}\Phi^{2^k}$, we have

$$\begin{bmatrix} d_{k+1}^{-1} & d_{k+1}^{-1}b_{k+1} \\ c_{k+1}d_{k+1}^{-1} & c_{k+1}d_{k+1}^{-1}b_{k+1} + a_{k+1} \end{bmatrix} = \begin{bmatrix} d_k^{-1} & d_k^{-1}b_k \\ c_k d_k^{-1} & c_k d_k^{-1} b_k + a_k \end{bmatrix} \begin{bmatrix} d_k^{-1} & d_k^{-1}b_k \\ c_k d_k^{-1} & c_k d_k^{-1} b_k + a_k \end{bmatrix}. \tag{38}$$

Then we are able to derive, after some algebra, the following nonlinear iterative equations:

$$\begin{aligned} a_{k+1} &= a_k \left(I - c_k [I + b_k c_k]^{-1} b_k\right) a_k, \\ b_{k+1} &= b_k + d_k [I + b_k c_k]^{-1} b_k a_k, \\ c_{k+1} &= c_k + a_k c_k [I + b_k c_k]^{-1} d_k, \\ d_{k+1} &= d_k [I + b_k c_k]^{-1} d_k, \end{aligned} \tag{39}$$

with initial conditions

$$\begin{aligned} a_1 &= D - CA^{-1}B = F - Q\left[Q + S^{-1}\right]^{-1} F, \\ b_1 &= A^{-1}B = S^{-1}F^T\left[Q + S^{-1}\right]^{-1} F, \\ c_1 &= CA^{-1} = Q\left[Q + S^{-1}\right]^{-1}, \\ d_1 &= A^{-1} = S^{-1}F^T\left[Q + S^{-1}\right]^{-1}. \end{aligned} \tag{40}$$

Then, since

$$\begin{bmatrix} X\left(2^k\right) \\ Y\left(2^k\right) \end{bmatrix} = \Phi^{2^k} \begin{bmatrix} I \\ 0 \end{bmatrix} = \begin{bmatrix} d_k^{-1} \\ c_k d_k^{-1} \end{bmatrix}, \tag{41}$$

it is clear that $c_k = Y_{2^k} X_{2^k}^{-1} = G_{2^k}$ tends to a steady state value $\widetilde{G}$.

2.2.2. Algebraic Algorithm for $\widetilde{G}$ Computation. In this section, we present an algebraic algorithm for $\widetilde{G}$ computation. As in (29), setting

$$G_k = K_k H = Y_k X_k^{-1} \tag{42}$$

and using the parameters A, B, C, D by (23), we derive

$$\Phi = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \left(Q + S^{-1}\right) F^{-T} S & F \\ QF^{-T}S & F \end{bmatrix}, \tag{43}$$

which is a matrix of dimension $2n \times 2n$. Since

$$\begin{aligned} \det \Phi &= \det(A) \det\left(D - CA^{-1}B\right) \\ &= \det\left(\left(Q + S^{-1}\right) F^{-T} S\right) \\ &\quad \times \det\left(F - QF^{-T}S\left[\left(Q + S^{-1}\right)F^{-T}S\right]^{-1} F\right) \\ &= \det\left(Q + S^{-1}\right) \det\left(F^{-1}\right) \det(S) \\ &\quad \times \det\left(F - QF^{-T}SS^{-1}F^T\left[Q + S^{-1}\right]^{-1} F\right) \\ &= \det(S) \det\left(F - Q\left[Q + S^{-1}\right]^{-1} F\right) \\ &\quad \times \det\left(F^{-1}\right) \det\left(Q + S^{-1}\right) \\ &= \det(S) \det\left(I - Q\left[Q + S^{-1}\right]^{-1}\right) \\ &\quad \times \det(F) \det\left(F^{-1}\right) \det\left(Q + S^{-1}\right) \\ &= \det(S) \det\left(\left(Q + S^{-1}\right) - Q\right) \\ &= \det(S) \det\left(S^{-1}\right) = 1, \end{aligned} \tag{44}$$

it is evident that Φ is a nonsingular matrix and its eigenvalues occur in reciprocal pairs.

Thus, (43) can be written

$$\Phi = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \left(Q + S^{-1}\right) F^{-T} S & F \\ QF^{-T}S & F \end{bmatrix} = WLW^{-1}, \tag{45}$$

where

$$L = \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda^{-1} \end{bmatrix} \tag{46}$$

is a diagonal matrix containing the eigenvalues of Φ, with Λ diagonal matrix with all the eigenvalues of Φ lying outside the unit circle, and

$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \tag{47}$$

is the matrix containing the corresponding eigenvectors of Φ, with

$$V = W^{-1} = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}. \tag{48}$$

We are able to use zero initial condition $P_0 = 0$, so $G_0 = K_0 H = Y_0 X_0^{-1} = 0$; that is,

$$\begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix}, \tag{49}$$

and hence

$$\begin{bmatrix} X_k \\ Y_k \end{bmatrix} = \Phi^k \begin{bmatrix} I \\ 0 \end{bmatrix}. \tag{50}$$

Then, from (50) and (45)–(48), we are able to write

$$\begin{aligned}\begin{bmatrix} X_k \\ Y_k \end{bmatrix} &= \Phi^k \begin{bmatrix} I \\ 0 \end{bmatrix} = \left[WLW^{-1}\right]^k \begin{bmatrix} I \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \begin{bmatrix} \Lambda^k & 0 \\ 0 & \Lambda^{-k} \end{bmatrix} \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \begin{bmatrix} I \\ 0 \end{bmatrix}; \end{aligned} \tag{51}$$

that is,

$$\begin{bmatrix} X_k \\ Y_k \end{bmatrix} = \begin{bmatrix} W_{11}\Lambda^k V_{11} + W_{12}\Lambda^{-k} V_{21} \\ W_{21}\Lambda^k V_{11} + W_{22}\Lambda^{-k} V_{21} \end{bmatrix}. \tag{52}$$

Substituting in (42) the matrices X_k, Y_k from (52), we have that

$$\begin{aligned} G_k = Y_k X_k^{-1} = &\left(W_{21}\Lambda^k V_{11} + W_{22}\Lambda^{-k} V_{21}\right) \\ &\times \left[W_{11}\Lambda^k V_{11} + W_{12}\Lambda^{-k} V_{21}\right]^{-1}. \end{aligned} \tag{53}$$

Furthermore, the diagonal matrix Λ^{-1} contains all the eigenvalues of Φ lying inside the unit circle, which follows that $\lim_{k\to\infty}\Lambda^{-k} = 0$. Then, G_k tends to a steady state value $\widetilde{G}$ with $\widetilde{G} = \lim_{k\to\infty} G_k$, and from (53) arises

$$\widetilde{G} = W_{21} W_{11}^{-1}. \tag{54}$$

2.2.3. Steady State Kalman Filter Gain Computation. All algorithms presented in Sections 2.2.1 and 2.2.2 compute the steady state value $\widetilde{G}$. Taking into account the assumptions of Section 2.1, we are able to conclude that, under the condition $\operatorname{rank}(H) = n$, the steady state gain is

$$\widetilde{K} = \widetilde{G}\left[H^T H\right]^{-1} H^T. \tag{55}$$

2.3. Direct Steady State Kalman Filter Gain Computation. In this section, we present algorithms for the direct computation of the steady state Kalman filter $\widetilde{K}$. The proposed algorithms compute directly the steady state Kalman filter gain, that is, without using $\widetilde{G} = \widetilde{K}H$. All these algorithms hold under the assumption that $n = m$. Note that, since $\operatorname{rank}(H) = n$, H and S are nonsingular matrices.

2.3.1. Iterative Algorithms for $\widetilde{K}$ Computation. In this section, we present two iterative per-step algorithms and an iterative doubling algorithm for $\widetilde{K}$ computation.

Per-Step Iterative Algorithm 1. Using (11), (22), and (13), we are able to derive the following nonlinear iterative equation with respect to the Kalman filter gain K_k:

$$\begin{aligned} K_{k+1} &= G_{k+1} H^{-1} \\ &= \left(QF^{-T}S + FG_k\right)\left[S^{-1}F^{-T}S + QF^{-T}S + FG_k\right]^{-1} H^{-1} \\ &= \left(QF^{-T}H^T R^{-1} H + FG_k\right) \\ &\quad \times \left[H^{-1}RH^{-T}F^{-T}H^T R^{-1} H \right. \\ &\qquad \left. + QF^{-T}H^T R^{-1} H + FG_k\right]^{-1} H^{-1} \\ &= \left(QF^{-T}H^T R^{-1} H + FK_k H\right) \\ &\quad \times \left[RH^{-T}F^{-T}H^T R^{-1} H \right. \\ &\qquad \left. + HQF^{-T}H^T R^{-1} H + HFK_k H\right]^{-1} \\ &= \left(QF^{-T}H^T R^{-1} + FK_k\right) \\ &\quad \times H\left[RH^{-T}F^{-T}H^T R^{-1} H \right. \\ &\qquad \left. + HQF^{-T}H^T R^{-1} H + HFK_k H\right]^{-1} \\ &= \left(QF^{-T}H^T R^{-1} + FK_k\right) \\ &\quad \times \left[RH^{-T}F^{-T}H^T R^{-1} \right. \\ &\qquad \left. + HQF^{-T}H^T R^{-1} + HFK_k\right]^{-1} \\ &= \left(QF^{-T}H^T R^{-1} + FK_k\right) \\ &\quad \times \left[HQF^{-T}H^T R^{-1} + RH^{-T}F^{-T}H^T R^{-1} + HFK_k\right]^{-1}. \end{aligned} \tag{56}$$

The nonsingularity of S and (13) allow us to write the equality in (56) as

$$\begin{aligned} K_{k+1} = &\left(QF^{-T}H^T R^{-1} + FK_k\right) \\ &\times \left[HQF^{-T}H^T R^{-1} + HS^{-1}F^{-T}H^T R^{-1} + HFK_k\right]^{-1} \\ = &\left(C + DK_k\right)\left[A + BK_k\right]^{-1}, \end{aligned} \tag{57}$$

where

$$\begin{aligned} A &= HQF^{-T}H^T R^{-1} + HS^{-1}F^{-T}H^T R^{-1} \\ &= H\left(Q + S^{-1}\right)F^{-T}H^T R^{-1}, \\ B &= HF, \\ C &= QF^{-T}H^T R^{-1}, \\ D &= F. \end{aligned} \tag{58}$$

The initial condition is $K_0 = P_0 H^T [H P_0 H^T + R]^{-1}$. It is known [1] that the prediction error covariance tends to the steady state prediction error covariance and that the convergence is independent of the initial uncertainty, that is, independent of the value of the initial condition P_0. Thus, we are able to assume zero initial condition $P_0 = 0$ and so we are to use the initial condition $K_0 = 0$.

It is clear that K_k tends to a steady state value $\widetilde{K}$ satisfying

$$\widetilde{K} = \left(C + D\widetilde{K}\right)\left[A + B\widetilde{K}\right]^{-1}. \tag{59}$$

Per-Step Iterative Algorithm 2. Using (57), we are able to derive the following nonlinear iterative equation with respect to the Kalman filter gain K_k:

$$\begin{aligned} K_{k+1} &= (C + DK_k)[A + BK_k]^{-1} \\ &= \left(CA^{-1}A + \left(D + CA^{-1}B - CA^{-1}B\right)K_k\right)[A + BK_k]^{-1} \\ &= \left(CA^{-1}(A + BK_k) + \left(D - CA^{-1}B\right)K_k\right)[A + BK_k]^{-1} \\ &= CA^{-1} + \left(D - CA^{-1}B\right)K_k[A + BK_k]^{-1} \\ &= CA^{-1} + \left(D - CA^{-1}B\right)\left[AK_k^{-1} + B\right]^{-1} \\ &= CA^{-1} + \left(D - CA^{-1}B\right)\left[K_k^{-1} + A^{-1}B\right]^{-1}A^{-1} \\ &= c + a\left[K_k^{-1} + b\right]^{-1}d, \end{aligned} \tag{60}$$

where A, B, C, D are given by (58) and

$$\begin{aligned} a &= D - CA^{-1}B, \\ b &= A^{-1}B, \\ c &= CA^{-1}, \\ d &= A^{-1}. \end{aligned} \tag{61}$$

The algorithm uses the initial condition $K_0 = P_0 H^T [H P_0 H^T + R]^{-1}$. It is known [1] that the prediction error covariance tends to the steady state prediction error covariance and that the convergence is independent of the initial uncertainty, that is, independent of the value of the initial condition P_0. Thus, we are able to assume zero initial condition $P_0 = 0$. In this case, in order to avoid K_0^{-1}, we are to use the initial condition $K_1 = c$.

It is clear that K_k tends to a steady state value $\widetilde{K}$ satisfying

$$\widetilde{K} = c + a\left[\widetilde{K}^{-1} + b\right]^{-1}d. \tag{62}$$

Doubling Iterative Algorithm. In (57), setting

$$K_k = Y_k X_k^{-1}, \tag{63}$$

we take

$$\begin{aligned} Y_{k+1}X_{k+1}^{-1} &= \left(C + DY_kX_k^{-1}\right)\left[A + BY_kX_k^{-1}\right]^{-1} \\ &= (CX_k + DY_k)[AX_k + BY_k]^{-1} \end{aligned} \tag{64}$$

or

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \end{bmatrix} = \Phi \begin{bmatrix} X_k \\ Y_k \end{bmatrix}, \tag{65}$$

where

$$\Phi = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} d^{-1} & d^{-1}b \\ cd^{-1} & cd^{-1}b + a \end{bmatrix} \tag{66}$$

is a matrix of dimension $2n \times 2n$ and A, B, C, D as in (58).

Working as in the doubling iterative algorithm of Section 2.2.1 and using zero initial condition $P_0 = 0$, so $K_0 = Y_0 X_0^{-1} = 0$; we are able to derive the following nonlinear iterative equations:

$$\begin{aligned} a_{k+1} &= a_k\left(I - c_k[I + b_kc_k]^{-1}b_k\right)a_k, \\ b_{k+1} &= b_k + d_k[I + b_kc_k]^{-1}b_ka_k, \\ c_{k+1} &= c_k + a_kc_k[I + b_kc_k]^{-1}d_k, \\ d_{k+1} &= d_k[I + b_kc_k]^{-1}d_k, \end{aligned} \tag{67}$$

with initial conditions

$$\begin{aligned} a_1 &= a, \\ b_1 &= b, \\ c_1 &= c, \\ d_1 &= d. \end{aligned} \tag{68}$$

It is clear that $c_k = Y_{2^k} X_{2^k}^{-1} = K_{2^k}$ tends to a steady state value $\widetilde{K}$.

2.3.2. Algebraic Algorithm for $\widetilde{K}$ Computation. In this section, we present an algebraic algorithm for $\widetilde{K}$ computation. Working as in the algebraic algorithm of Section 2.2.2 and using the parameters A, B, C, D by (58), we derive

$$\Phi = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda^{-1} \end{bmatrix} \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}^{-1}, \tag{69}$$

which is a matrix of dimension $2n \times 2n$.

Then, the steady state Kalman filter is

$$\widetilde{K} = W_{21} W_{11}^{-1}. \tag{70}$$

2.4. Advantages of the Proposed Algorithms. All algorithms for the computation of the steady state Kalman filter gain $\widetilde{K}$, presented in Section 2, are summarized in Table 1. It is clear that the direct computation of the Kalman filter gain

Table 1: Algorithms for the computation of the steady state Kalman filter gain $\widetilde{K}$.

Computation of $\widetilde{K}$ through $\widetilde{G} = \widetilde{K}H$	Direct computation of $\widetilde{K}$
System parameters F, H, Q, R	System parameters F, H, Q, R
Restriction $m \geq n$	Restriction $m = n$
Algorithms parameters	*Algorithms parameters*
$A = (Q + (H^T R^{-1} H)^{-1}) F^{-T} H^T R^{-1} H$	$A = H(Q + (H^T R^{-1} H)^{-1}) F^{-T} H^T R^{-1}$
$B = F$	$B = HF$
$C = QF^{-T} H^T R^{-1} H$	$C = QF^{-T} H^T R^{-1}$
$D = F$	$D = F$
$a = D - CA^{-1} B$	$a = D - CA^{-1} B$
$b = A^{-1} B$	$b = A^{-1} B$
$c = CA^{-1}$	$c = CA^{-1}$
$d = A^{-1}$	$d = A^{-1}$
Per-step iterative algorithm 1	*Per-step iterative algorithm 1*
$G_{k+1} = (C + DG_k)[A + BG_k]^{-1}$	$K_{k+1} = (C + DK_k)[A + BK_k]^{-1}$
$G_0 = 0$	$K_0 = 0$
$G_k \rightarrow \widetilde{G}$	$K_k \rightarrow \widetilde{K}$
$\widetilde{K} = \widetilde{G}[H^T H]^{-1} H^T$	
Per-step iterative algorithm 2	*Per-step iterative algorithm 2*
$G_{k+1} = c + a[G_k^{-1} + b]^{-1} d$	$K_{k+1} = c + a[K_k^{-1} + b]^{-1} d$
$G_1 = c$	$K_1 = c$
$G_k \rightarrow \widetilde{G}$	$K_k \rightarrow \widetilde{K}$
$\widetilde{K} = \widetilde{G}[H^T H]^{-1} H^T$	
Doubling iterative algorithm	*Doubling iterative algorithm*
$a_{k+1} = a_k \left(I - c_k [I + b_k c_k]^{-1} b_k\right) a_k$	$a_{k+1} = a_k \left(I - c_k [I + b_k c_k]^{-1} b_k\right) a_k$
$b_{k+1} = b_k + d_k [I + b_k c_k]^{-1} b_k a_k$	$b_{k+1} = b_k + d_k [I + b_k c_k]^{-1} b_k a_k$
$c_{k+1} = c_k + a_k c_k [I + b_k c_k]^{-1} d_k$	$c_{k+1} = c_k + a_k c_k [I + b_k c_k]^{-1} d_k$
$d_{k+1} = d_k [I + b_k c_k]^{-1} d_k$	$d_{k+1} = d_k [I + b_k c_k]^{-1} d_k$
$a_1 = a, b_1 = b, c_1 = c, d_1 = d$	$a_1 = a, b_1 = b, c_1 = c, d_1 = d$
$c_k \rightarrow \widetilde{G}$	$c_k \rightarrow \widetilde{K}$
$\widetilde{K} = \widetilde{G}[H^T H]^{-1} H^T$	
Algebraic algorithm	*Algebraic algorithm*
$\Phi = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$	$\Phi = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$
$= \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda^{-1} \end{bmatrix} \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}^{-1}$	$= \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda^{-1} \end{bmatrix} \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}^{-1}$
$\widetilde{G} = W_{21} W_{11}^{-1}$	$\widetilde{K} = W_{21} W_{11}^{-1}$
$\widetilde{K} = \widetilde{G}[H^T H]^{-1} H^T$	

is feasible only if the following restriction holds: $n = m$. The advantage of the presented algorithms is the autonomous computation of the steady state Kalman filter gain. Especially, the steady state Kalman filter gain is important, when we want to compute the parameters of the steady state Kalman filter

$$\begin{aligned} x_{k+1/k+1} &= \left(I - \widetilde{K}H\right) F x_{k/k} + \widetilde{K} z_{k+1} \\ &= \left(I - \widetilde{G}\right) F x_{k/k} + \widetilde{K} z_{k+1}. \end{aligned} \tag{71}$$

It is obvious from (71) that the parameters of the steady state Kalman filter are related to the steady state Kalman filter gain.

In particular, the steady state prediction error covariance can be computed via the steady state gain and is given by

$$\widetilde{P}_p = \left[I - \widetilde{K}H\right]^{-1} \widetilde{K} R H \left[H^T H\right]^{-1}. \tag{72}$$

Indeed, from (2), arises $\widetilde{K} = \widetilde{P}_p H^T [H \widetilde{P}_p H^T + R]^{-1}$, which leads to

$$\begin{aligned} \widetilde{K}\left(H \widetilde{P}_p H^T + R\right) = \widetilde{P}_p H^T &\Longrightarrow \widetilde{K} H \widetilde{P}_p H^T + \widetilde{K} R = \widetilde{P}_p H^T \\ &\Longrightarrow \widetilde{P}_p H^T - \widetilde{K} H \widetilde{P}_p H^T = \widetilde{K} R \end{aligned}$$

$$\begin{gathered}\Longrightarrow \left(I-\widetilde{K}H\right)\widetilde{P}_pH^T=\widetilde{K}R\\ \Longrightarrow \widetilde{P}_pH^T=\left[I-\widetilde{K}H\right]^{-1}\widetilde{K}R\\ \Longrightarrow \widetilde{P}_pH^TH=\left[I-\widetilde{K}H\right]^{-1}\widetilde{K}RH.\end{gathered}\tag{73}$$

Since rank$(H)=n$, the matrix H^TH is nonsingular [21]; thus from the last equation arises immediately the formula of the steady state prediction error covariance in (72).

Also, by (3), the steady state estimation error covariance can be computed via the steady state prediction error covariance

$$\widetilde{P}_e=\left[I-\widetilde{K}H\right]\widetilde{P}_p.\tag{74}$$

3. Examples

In this section, two examples verify the results of Section 2.

Example 1. A model of dimensions $n=1$ and $m=2$ is assumed with parameters:

$$\begin{gathered}F=0.8,\\ H=\begin{bmatrix}1\\2\end{bmatrix},\\ Q=5,\\ R=\begin{bmatrix}0.1&0\\0&0.4\end{bmatrix}.\end{gathered}\tag{75}$$

In this example, we have rank$(H)=n=1$ with $m>n$.

Using all algorithms presented in Section 2.2, we computed

$$\widetilde{G}=0.9902.\tag{76}$$

Then, using (55), we computed the steady state gain

$$\widetilde{K}=\left[0.1980\ \ 0.3961\right].\tag{77}$$

Example 2. A model of dimensions $m=n=2$ is assumed with parameters:

$$\begin{gathered}F=\begin{bmatrix}-0.9&0.7\\-0.3&0.1\end{bmatrix},\\ H=\begin{bmatrix}1&3\\2&2\end{bmatrix},\\ Q=\begin{bmatrix}1&0\\0&3\end{bmatrix},\\ R=\begin{bmatrix}0.1&0\\0&0.4\end{bmatrix}.\end{gathered}\tag{78}$$

In this example, we have rank$(H)=2$.

Using all algorithms presented in Section 2.2, we computed

$$\widetilde{G}=\begin{bmatrix}0.8382&0.0317\\0.0647&0.9841\end{bmatrix}.\tag{79}$$

Then, using (55), we computed the steady state gain

$$\widetilde{K}=\begin{bmatrix}-0.4033&0.6207\\0.4597&-0.1975\end{bmatrix}.\tag{80}$$

We also computed the same steady state gain, using all algorithms presented in Section 2.3, since $m=n=2$.

4. Conclusions

The Kalman filter gain arises in Kalman filter equations in linear estimation and is associated with linear systems. The gain is a matrix through which the estimation and the prediction of the state as well as the corresponding estimation and prediction error covariance matrices are computed. For time invariant and asymptotically stable systems, there exist steady state values of the estimation and prediction error covariance matrices. There exists also a steady state value of the Kalman filter gain.

The steady state Kalman filter gain is usually derived via the steady state prediction error covariance by first solving the corresponding Riccati equation. In view of the importance of the Riccati equation, there exists considerable literature on its algebraic or iterative solutions, including the Chandrasekhar algorithms, which are the only iterative algorithms that provide the steady state Kalman filter gain together with the prediction error covariance.

Iterative per-step and doubling algorithms as well as an algebraic algorithm for the steady state Kalman filter computation were presented. These algorithms hold under conditions concerning the system parameters. The advantage of these algorithms is the autonomous computation of the steady state Kalman filter gain. This is important if we want to compute only the steady state Kalman filter gain or to compute the parameters of the steady state Kalman filter, which are related to the steady state Kalman filter gain.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

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

[2] N. Assimakis and M. Adam, “Global systems for mobile position tracking using Kalman and Lainiotis filters,” *The Scientific World Journal*, vol. 2014, Article ID 130512, 8 pages, 2014.

[3] F. Ding, “Combined state and least squares parameter estimation algorithms for dynamic systems,” *Applied Mathematical Modelling*, vol. 38, no. 1, pp. 403–412, 2014.

[4] F. Ding and T. Chen, “Hierarchical identification of lifted state-space models for general dual-rate systems,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 6, pp. 1179–1187, 2005.

[5] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter*, Artech House, Boston, Mass, USA, 2004.

[6] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, John Wiley & Sons, Hoboken, NJ, USA, 3rd edition, 2008.

[7] N. Assimakis and M. Adam, "Kalman filter Riccati equation for the prediction, estimation and smoothing error covariance matrices," *ISRN Computational Mathematics*, vol. 2013, Article ID 249594, 7 pages, 2013.

[8] J. R. P. de Carvalho, E. D. Assad, and H. S. Pinto, "Kalman filter and correction of the temperatures estimated by PRECIS model," *Atmospheric Research*, vol. 102, no. 1-2, pp. 218–226, 2011.

[9] R. Furrer and T. Bengtsson, "Estimation of high-dimensional prior and posterior covariance matrices in Kalman filter variants," *Journal of Multivariate Analysis*, vol. 98, no. 2, pp. 227–255, 2007.

[10] Y. Teruyama and T. Watanabe, "Effectiveness of variable-gain Kalman filter based on angle error calculated from acceleration signals in lower limb angle measurement with inertial sensors," *Computational and Mathematical Methods in Medicine*, vol. 2013, Article ID 398042, 12 pages, 2013.

[11] S. Han, "A closed-form solution to the discrete-time Kalman filter and its applications," *Systems & Control Letters*, vol. 59, no. 12, pp. 799–805, 2010.

[12] D. R. Vaughan, "A nonrecursive algebraic solution for the disczte Riccati equation," *IEEE Transactions on Automatic Control*, vol. 15, no. 5, pp. 597–599, 1970.

[13] R. Leland, "An alternate calculation of the discrete-time Kalman filter gain and Riccati equation solution," *Transactions on Automatic Control*, vol. 41, no. 12, pp. 1817–1819, 1996.

[14] N. Assimakis, "Discrete time Riccati equation recursive multiple steps solutions," *Contemporary Engineering Sciences*, vol. 2, no. 7, pp. 333–354, 2009.

[15] N. D. Assimakis, D. G. Lainiotis, S. K. Katsikas, and F. L. Sanida, "A survey of recursive algorithms for the solution of the discrete time riccati equation," *Nonlinear Analysis: Theory, Methods and Applications*, vol. 30, no. 4, pp. 2409–2420, 1997.

[16] N. Assimakis, S. Roulis, and D. Lainiotis, "Recursive solutions of the discrete time Riccati equation," *Neural, Parallel and Scientific Computations*, vol. 11, no. 3, pp. 343–350, 2003.

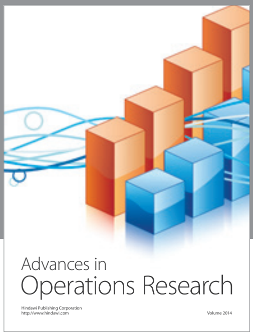
[17] D. G. Lainiotis, N. D. Assimakis, and S. K. Katsikas, "A new computationally effective algorithm for solving the discrete Riccati equation," *Journal of Mathematical Analysis and Applications*, vol. 186, no. 3, pp. 868–895, 1994.

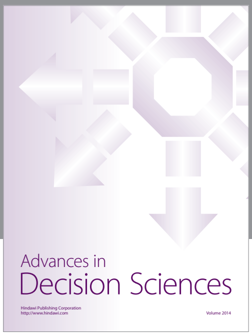
[18] D. G. Lainiotis, N. D. Assimakis, and S. K. Katsikas, "Fast and numerically robust recursive algorithms for solving the discrete time Riccati equation: the case of nonsingular plant noise covariance matrix," *Neural, Parallel and Scientific Computations*, vol. 3, no. 4, pp. 565–583, 1995.

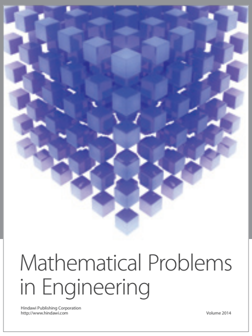
[19] M. Liyanage and I. Sasase, "Steady-state Kalman filtering for channel estimation in OFDM systems utilizing SNR," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–6, Dresden, Germany, June 2009.

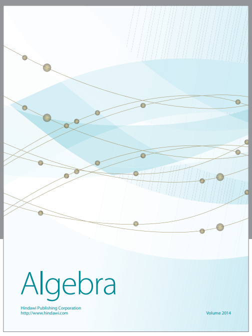
[20] T. R. Kronhamn, "Geometric illustration of the Kalman filter gain and covariance update algorithms," *IEEE Control Systems Magazine*, vol. 5, no. 2, pp. 41–43, 1985.

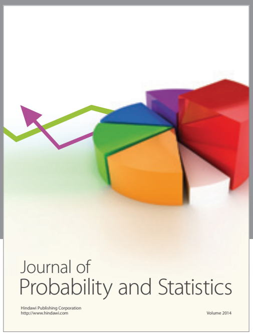
[21] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 2005.


Advances in Operations Research

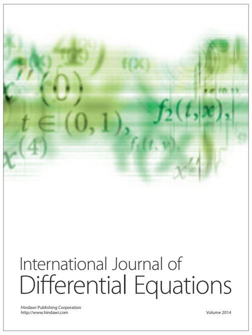

Advances in Decision Sciences

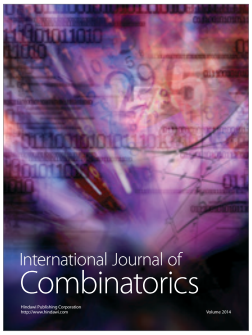

Mathematical Problems in Engineering

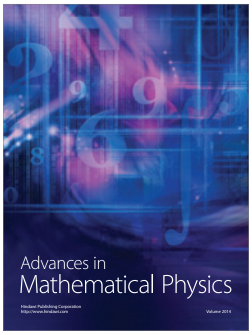

Algebra

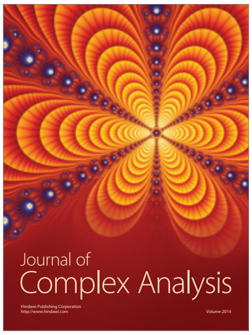

Journal of Probability and Statistics

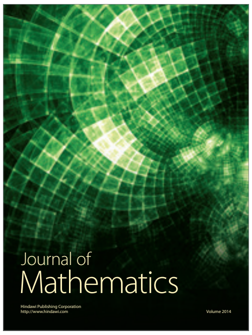
The Scientific World Journal

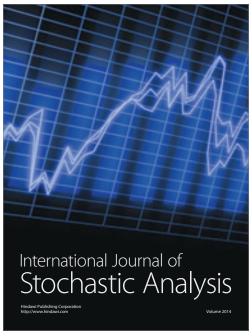

International Journal of Differential Equations

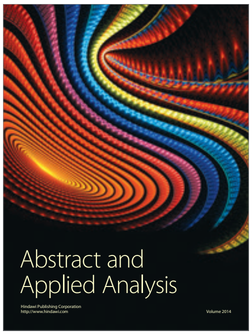

International Journal of Combinatorics

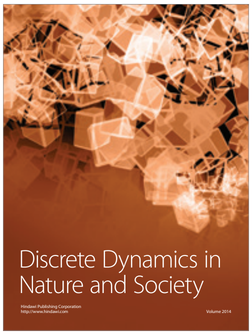

Advances in Mathematical Physics

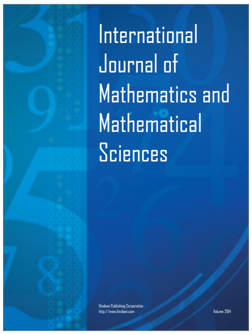

Journal of Complex Analysis


Journal of Mathematics


International Journal of Stochastic Analysis

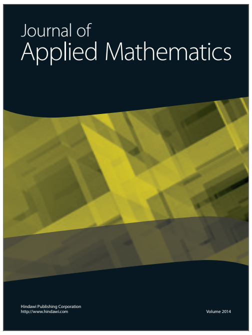

Abstract and Applied Analysis


Discrete Dynamics in Nature and Society


International Journal of Mathematics and Mathematical Sciences

Journal of Discrete Mathematics

Journal of Function Spaces


Journal of Applied Mathematics

Journal of Optimization