Estimation Algorithms via Kalman Filter Gain

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

Abstract: - Estimation algorithms using Kalman filter gain are proposed. Kalman filter gain is computed at each iteration through auxiliary quantities. The proposed estimation algorithm is faster than the traditional Kalman filter for time invariant systems, is equal fast to the traditional Kalman filter for steady state case and can be faster than the traditional Kalman filter for time varying systems, depending on the model dimensions.

Key-Words: - Estimation, Kalman filter, Kalman filter gain, Time Varying, Time Invariant, Steady State

Received: July 15, 2023. Revised: March 25, 2024. Accepted: June 7, 2024. Published: July 5, 2024.

1 Introduction

The importance of Kalman filter [1] is undoubtable: it has been widely and successfully used to solve estimation and prediction problems in a vast range of applications: object detection and tracking [2], robotic applications [3], electric load estimation [4], industrial applications [5], stock price prediction [6], weather forecasts [7], satellite orbit determination [8], power generation prediction [9], cases prediction of Covid-19 [10], multi-observation fusion applications related to timescale [11], structural parameter tracking [12], dynamic tracking of the uncertainties associated with the underlying data and prior knowledge [13], Kalman filter-based tracking-by-detection (KFTBD) tracker [14].

Consider the discrete time $(k = 0,1, ...)$ linear model of the form [15]:

$$x(k+1) = F(k+1,k)x(k) + w(k) \quad (1)$$

$$z(k) = H(k)x(k) + v(k) \quad (2)$$

where the state $x(k)$ has dimension n, the measurement has dimension m. The model describes the relation between two successive states through $F(k+1,k)$ and the relation between the state and the measurement through $H(k)$. In addition, the state noise $w(k)$ and measurement noise $v(k)$ are Gaussian with zero means and covariances $Q(k)$ and $R(k)$. The initial state $x(0)$ is Gaussian with mean x_0 and covariance P_0.

The traditional Kalman filter computes the estimation $x(k/k)$ with the corresponding covariance $P(k/k)$ and the prediction $x(k+1/k)$ with the corresponding covariance $P(k+1/k)$ using the Kalman filter gain $K(k)$, as it is shown in Figure 1.

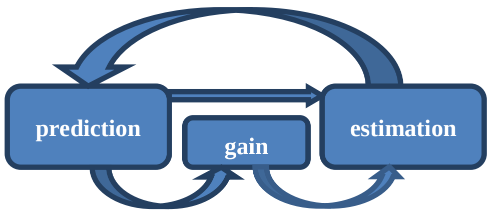

Fig. 1: Kalman filter iterative calculations

In this work, we derive estimation algorithms using Kalman filter gain but avoiding the computation of the estimation covariance matrix. The system parameters determine when these algorithms are valid.

The novelty of this paper consists in developing estimation algorithms (a) using Kalman filter gain computed through auxiliary quantities, (b) avoiding the estimation error covariance computation.

Kalman filter equations are briefly presented in section 2, the proposed estimation algorithms are derived in section 3, the algorithms are compared in section 4 and conclusions are presented in section 5.

2 Kalman Filter

For *time varying systems*, the Time Varying Kalman Filter (TVKF) is derived:

Time Varying Kalman Filter (TVKF)

$$K(k) = P(k/k-1)H^T(k) [H(k)P(k/k-1)H^T(k) + R(k)]^{-1}$$
$$x(k/k) = [I - K(k)H(k)]x(k/k-1) + K(k)z(k)$$
$$P(k/k) = [I - K(k)H(k)]P(k/k-1)$$
$$x(k+1/k) = F(k+1,k)x(k/k)$$
$$P(k+1/k) = Q(k) + F(k+1,k)P(k/k)F^T(k+1,k)$$

The initialization ($k = 0$) is:
$x(0/-1) = x_0$
$P(0/-1) = P_0$
Note that I is the identity matrix and M^T is the transpose of matrix M.

Remark. In the case where no measurement is exact, the inverse of $H(k)P(k/k-1)H^T(k) + R(k)$ exist, since R(k) is p.d. (positive definite).
For *time invariant systems* where $F = F(k+1,k), H = H(k), Q = Q(k), R = R(k)$, the Time Invariant Kalman filter (TIKF) is derived:

Time Invariant Kalman Filter (TIKF)

$$\boxed{\begin{aligned}&K(k) = P(k/k-1)H^T[HP(k/k-1)H^T + R]^{-1}\\&x(k/k) = [I - K(k)H]x(k/k-1) + K(k)z(k)\\&P(k/k) = [I - K(k)H]P(k/k-1)\\&x(k+1/k) = Fx(k/k)\\&P(k+1/k) = Q + FP(k/k)F^T\end{aligned}}$$

The initialization ($k = 0$) is:
$x(0/-1) = x_0$
$P(0/-1) = P_0$.

It is known [15] that in the steady state case the prediction and estimation covariances remain constant; then the Steady State Kalman Filter (SSKF) is derived:

Steady State Kalman Filter (SSKF)

$$\boxed{x(k/k) = (I - KH)Fx(k-1/k-1) + Kz(k)}$$

The initialization ($k = 1$) is:
$x(0/-1) = x_0$
$P(0/-1) = P_0$
$K(0) = P_0H^T[HP_0H^T + R]^{-1}$
$x(0/0) = [I - K(0)H]x_0 + K(0)z(0)$
The coefficients $[(I - KH)F]$ and $[K]$ are off-line computed by solving the algebraic Riccati equation:

$$P = Q + FPF^T - FPH^T[HPH^T + R]^{-1}HPF^T \quad (3)$$

where P is the steady state prediction covariance and then computing the steady state Kalman filter gain K as:

$$K = PH^T[HPH^T + R]^{-1} \quad (4)$$

Remark. There exists significant bibliography concerning the algebraic and the iterative solutions of the Riccati equation; see for example [15]-[20].

3 Estimation Algorithms based on Kalman Filter Gain

The main idea is the use of the Kalman filter gain and the non-use of the estimation covariance. We are going to derive the Kalman filter gain via auxiliary quantities as in [21] and we are not going to compute the estimation error covariance.

3.1 Time Varying Algorithms

3.1.1 Time varying algorithm with $m > n$

Working as in [21] we use the auxiliary matrices

$$G(k) = K(k)H(k) \quad (5)$$

and

$$\xi(k) = [H^T(k)H(k)]^{-1}H^T(k) \quad (6)$$

and we derive the following algorithm:

Time Varying Kalman Filter using Gain (TVKFG)

$$\boxed{\begin{aligned}&S(k) = H^T(k)R^{-1}(k)H(k)\\&A(k) = [Q(k) + S^{-1}(k)]F^{-T}(k+1,k)S(k)\\&B(k) = F(k+1,k)\\&C(k) = Q(k)F^{-T}(k+1,k)S(k)\\&D(k) = F(k+1,k)\\&G(k+1) = [C(k) + D(k)G(k)]\\&\qquad [A(k) + B(k)G(k)]^{-1}\\&\xi(k+1) = [H^T(k+1)H(k+1)]^{-1}H^T(k+1)\\&K(k+1) = G(k+1)\xi(k+1)\\&x(k+1/k+1) = [I - G(k+1)]\\&\qquad F(k+1,k)x(k/k)\\&\qquad +K(k+1)z(k+1)\end{aligned}}$$

The initialization ($k = 0$) is:
$x(0/-1) = x_0$
$P(0/-1) = P_0$
$K(0) = P_0H^T(0)[H(0)P_0H^T(0) + R(0)]^{-1}$
$G(0) = K(0)H(0)$
$x(0/0) = [I - G(0)]x_0 + K(0)z(0)$

Remark. S(k) is symmetric positive semidefinite matrix of dimension $n \times n$ and S(k) is p.d. (positive definite) matrix if $\mathrm{rank}(H(k)) = n$. This means that S(k) is nonsingular when $\mathrm{rank}(H(k)) = n$ with $m \geq n$ [22].

3.1.2 Time varying algorithm with $m = n$

When $m = n$ we derive the following algorithm:

Time Varying Kalman Filter using Gain with equal dimensions (TVKFGe)

$$\boxed{\begin{aligned}&S(k) = H^T(k)R^{-1}(k)H(k)\\&a(k) = H(k)[Q(k) + S^{-1}(k)]\\&\qquad F^{-T}(k+1,k)H^T(k)R^{-1}(k)\\&b(k) = H(k)F(k+1,k)\\&c(k) = Q(k)F^{-T}(k+1,k)H^T(k)R^{-1}(k)\\&d(k) = F(k+1,k)\\&K(k+1) = [c(k) + d(k)K(k)]\\&\qquad [a(k) + b(k)K(k)]^{-1}\\&x(k+1/k+1) = [I - K(k+1)H(k+1)]\\&\qquad F(k+1,k)x(k/k)\\&\qquad +K(k+1)z(k+1)\end{aligned}}$$

The initialization ($k = 0$) is:

$x(0/-1) = x_0$
$P(0/-1) = P_0$
$K(0) = P_0 H^T(0)[H(0)P_0 H^T(0) + R(0)]^{-1}$
$x(0/0) = [I - K(0)H]x_0 + K(0)z(0)$

3.2 Time Invariant Algorithms

3.2.1 Time invariant algorithm with $m > n$

Working as in [23] we use the auxiliary matrices

$$G(k) = K(k)H \quad (7)$$

and

$$\xi = [H^T H]^{-1} H^T \quad (8)$$

and we derive the following algorithm:

Time Invariant Kalman Filter using Gain (TIKFG)

$$G(k+1) = [C + DG(k)][A + BG(k)]^{-1}$$
$$K(k+1) = G(k+1)\xi$$
$$x(k+1/k+1) = [I - G(k+1)]Fx(k/k) + K(k+1)z(k+1)$$

The initialization $(k = 0)$ is:
$x(0/-1) = x_0$
$P(0/-1) = P_0$
$K(0) = P_0 H^T[HP_0 H^T + R]^{-1}$
$G(0) = K(0)$
$x(0/0) = [I - K(0)H]x_0 + K(0)z(0)$
Here, the following matrices are computed off-line:
$\xi = [H^T H]^{-1} H^T$
$S = H^T R^{-1} H$
$A = [Q + S^{-1}]F^{-T} S$
$B = F$
$C = QF^{-T} S$
$D = F$

3.2.2 Time invariant algorithm with $m = n$

When $m = n$ we derive the following algorithm:

Time Invariant Kalman Filter using Gain with equal dimensions (TIKFGe)

$$K(k+1) = [c + dK(k)][a + bK(k)]^{-1}$$
$$x(k+1/k+1) = [I - K(k+1)H]Fx(k/k) + K(k+1)z(k+1)$$

The initialization $(k = 0)$ is:
$x(0/-1) = x_0$
$P(0/-1) = P_0$
$K(0) = P_0 H^T[HP_0 H^T + R]^{-1}$
$x(0/0) = [I - K(0)H]x_0 + K(0)z(0)$
Here, the following matrices are computed off-line:
$S = H^T R^{-1} H$
$a = H[Q + S^{-1}]F^{-T} H^T R^{-1}$
$b = HF$
$c = QF^{-T} H^T R^{-1}$
$d = F$

3.3 Steady State Algorithms

3.3.1 Steady state algorithm with $m > n$

Steady State Kalman Filter using Gain (SSKFG)

$$x(k/k) = [I - G]Fx(k-1/k-1) + Kz(k)$$

The initialization $(k = 1)$ is:
$x(0/-1) = x_0$
$P(0/-1) = P_0$
$K(0) = P_0 H^T[HP_0 H^T + R]^{-1}$
$x(0/0) = [I - K(0)H]x_0 + K(0)z(0)$
The coefficients $[(I - G)F]$ and $[K]$ are off-line calculated by solving the equation

$$G = [C + DG][A + BG]^{-1} \quad (9)$$

using techniques derived in [21]
and then calculating K as:

$$K = G\xi \quad (10)$$

using (8).

3.3.2 Steady state algorithm with $m = n$

When $m = n$ we derive the following algorithm:

Steady State Kalman Filter using Gain with equal dimensions (SSKFGe)

$$x(k/k) = [I - KH]Fx(k-1/k-1) + Kz(k)$$

The initialization $(k = 1)$ is:
$x(0/-1) = x_0$
$P(0/-1) = P_0$
$K(0) = P_0 H^T[HP_0 H^T + R]^{-1}$
$x(0/0) = [I - K(0)H]x_0 + K(0)z(0)$
The coefficients $[(I - KH)F]$ and $[K]$ are off-line calculated by computing K solving the equation

$$K = [c + dK][a + bK]^{-1} \quad (11)$$

using techniques derived in [21].

4 Comparison of Algorithms

4.1. Per Iteration Calculation Burden

It is established that the estimation algorithms are derived using the Kalman filter equations. Thus the estimation algorithms and the traditional Kalman filter are equivalent filters with respect to their behavior, since they calculate theoretically the same estimates. Since all the filters are iterative, we assume that they compute the estimation executing the same number of iterations. So, the algorithms will be compared with respect to their per iteration calculation burden (CB) required for the (on-line) calculations. Table 1 summarizes the CB of needed matrix operations; see details in [23].

Table 1. Matrices Operations Calculation Burden

Matrix Operation	Calculation Burden
$M1 + M2 = M$ $(d_1 \times d_2) + (d_1 \times d_2)$	$d_1 d_2$
$M1 + M2 = S$ $(d \times d) + (d \times d)$ S symmetric	$\frac{1}{2}d^2 + \frac{1}{2}d$
$I + M1 = M$ $(d \times d) + (d \times d)$	d
$M1 \cdot M2 = M$ $(d_1 \times d_2) \cdot (d_2 \times d_3)$	$2d_1 d_2 d_3 - d_1 d_3$
$M1 \cdot M2 = S$ $(d_1 \times d_2) \cdot (d_2 \times d_1)$ S symmetric	${d_1}^2 d_2 + d_1 d_2 - \frac{1}{2}{d_1}^2 - \frac{1}{2}d_1$
M^{-1} $(d \times d)$	$\begin{cases} \frac{1}{6}(16d^3 - 3d^2 - d), & d \geq 2 \\ 1, & d = 1 \end{cases}$

The per iteration calculation burdens of the traditional Kalman Filter – KF (time varying, time invariant and steady state) have been calculated in [23]. The per iteration calculation burdens of the proposed estimation algorithms – KFG (time varying, time invariant and steady state) are calculated in the Appendix and summarized in Table 2.

Table 2. Per iteration CB of estimation algorithms

Time Varying Algorithms		
KF	$m \geq n \geq 2$	$CB_{TVKF} = \frac{1}{2}(8n^3 + 7n^2 - 3n)$ $+4n^2m + nm + 3nm^2$ $+\frac{1}{6}(16m^3 - 3m^2 - m)$
	$m > n = 1$	$CB_{TVKF1} = \frac{1}{6}(16m^3 + 15m^2 + 29m + 36)$
	$m = n \geq 2$	$CB_{TVKFe} = \frac{1}{3}(41n^3 + 12n^2 - 5n)$
	$m = n = 1$	$CB_{TVKF1} = 15$
KFG	$m \geq n \geq 2$	$CB_{TVKFG} = \frac{1}{6}(136n^3 - 15n^2 - 13n)$ $+6n^2m + nm + 2nm^2$ $+\frac{1}{6}(16m^3 - 3m^2 - m)$
	$m > n = 1$	$CB_{TVKFG1} = \frac{1}{6}(84m^3 + 9m^2 + 41m + 84)$
	$m = n \geq 2$	$CB_{TVKFGe} = \frac{1}{3}(95n^3 - 9n^2 - 5n)$
	$m = n = 1$	$CB_{TVKFG1} = 23$
Time Invariant Algorithms		
KF	$m \geq n \geq 2$	$CB_{TIKF} = \frac{1}{2}(8n^3 + 7n^2 - 3n)$ $+4n^2m + nm + 3nm^2$ $+\frac{1}{6}(16m^3 - 3m^2 - m)$
	$m > n = 1$	$CB_{TIKF1} = \frac{1}{6}(16m^3 + 15m^2 + 29m + 36)$
	$m = n \geq 2$	$CB_{TIKFe} = \frac{1}{3}(41n^3 + 12n^2 - 5n)$
	$m = n = 1$	$CB_{TIKF1} = 15$
KFG	$m \geq n \geq 2$	$CB_{TIKFG} = \frac{1}{6}(52n^3 + 15n^2 - 7n)$ $+2n^2m + nm$
	$m > n = 1$	$CB_{TVKFG1} = 3m + 9$
	$m = n \geq 2$	$CB_{TIKFGe} = \frac{1}{6}(64n^3 + 21n^2 - 7n)$
	$m = n = 1$	$CB_{TIKFG1} = 12$
Steady State Algorithms		
KF	$m \geq n$	$CB_{SSKF} = 2n^2 + 2nm - n$
	$m = n$	$CB_{SSKFe} = 4n^2 - n$
KFG	$m \geq n$	$CB_{SSKFG} = 2n^2 + 2nm - n$
	$m = n$	$CB_{SSKFGe} = 4n^2 - n$

4.2. The Faster Algorithm Determination

Concerning **Time Varying Algorithms**, from Table 2 it results that the proposed algorithm may be faster than the traditional Kalman filter, depending on the model dimensions. Figure 2 depicts the faster algorithm with respect to the state dimension n and the measurement dimension m. Recall that the proposed algorithm holds for $m \geq n$. It is clear that the proposed algorithm is faster than the traditional Kalman filter when $m \geq 5.5n$, while the the traditional Kalman filter is faster that the proposed algorithm when $5.5n > m \geq n$. Thus, the proposed algorithm is faster than the traditional Kalman filter when the rule of thumb holds:

$$m \geq 5.5n \tag{12}$$

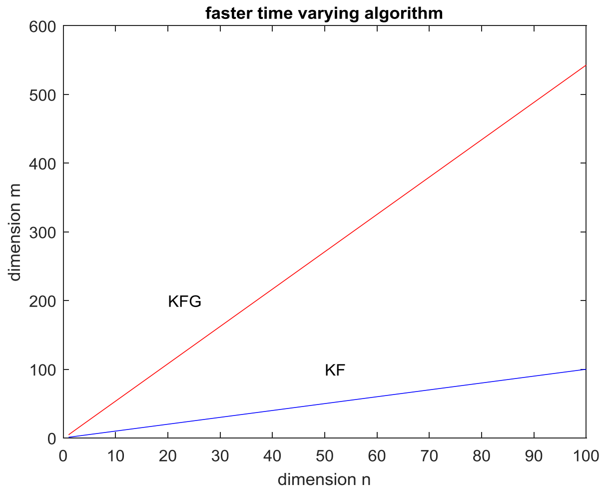

Fig. 2: Faster time varying algorithm determination

Concerning **Time Invariant Algorithms**, from Table 2 it results that the proposed algorithm is always faster than the traditional Kalman filter.

Concerning **Steady State Algorithms**, from Table 2 it results that the proposed algorithm is as fast is the traditional Kalman filter.

Example. A seismic deconvolution example taken from [24] is considered. The (time invariant) wavelet used to describe the signal received by the seismic sensors is of dimension $n = 4$. In order to capture the seismic trace, $m = 1000$ sensors are used. Then the proposed algorithm is faster than the traditional one: speedup $= \frac{CB_{TIKF}}{CB_{TIKFG}} = 70.909 \cdot 10^3$

5 Conclusion

The traditional Kalman filter uses the Kalman filter gain in order to compute the estimation of the state. We proposed a new Kalman filter variant which holds when the measurements dimension m is greater or equal to the state dimension n, i.e. $m \geq n$.

Estimation algorithms have been developed for the cases: time varying, time invariant and steady state. The proposed estimation time varying and time invariant algorithms compute the Kalman filter gain at each iteration through auxiliary quantities, without computing the estimation error covariance. The proposed estimation algorithm for the steady state case, requires the off-line computation of the steady state Kalman filter gain.

Comparing the developed estimation algorithm to the traditional Kalman filter with respect to their computational burdens, we have concluded that: (a) in the time invariant case, the developed estimation algorithm is always faster than the traditional Kalman filter, (b) in the steady state case, the two algorithms have equal calculation burdens, (c) in the time varying case, the developed estimation algorithm is faster than the traditional Kalman filter when $m \geq 5.5n$. Thus, the proposed estimation algorithm outperforms the classical Kalman filter (1) for time invariant problems with $m \geq n$, (2) for time varying problems with $m \geq 5.5n$. This means that the proposed estimation algorithms take advantage of multi-sensors problems.

Future research may investigate how to apply the derived algorithms to LQR (Linear Quadratic Regulator) [25]. Another subject of future research is to use the main idea of this work in order to derive information filters that use the inverses of the covariance matrices.

Appendix

The calculation burdens of the proposed algorithms are analytically presented.

TVKFG	
Matrix Operation	**Calculation Burden**
$R^{-1}(k)$	$\frac{1}{6}(16m^3-3m^2-m), m \geq 2$ $1, \quad m=1$
$H^T(k)R^{-1}(k)$	$2nm^2-nm$
$S(k)=H^T(k)R^{-1}(k)H(k)$	$n^2m+nm-\frac{1}{2}n^2-\frac{1}{2}n$
$S^{-1}(k)$	$\frac{1}{6}(16n^3-3n^2-n), n \geq 2$ $1, \quad n=1$
$Q(k)+S^{-1}(k)$	$\frac{1}{2}n^2+\frac{1}{2}n$
$F^{-T}(k+1,k)$	$\frac{1}{6}(16n^3-3n^2-n)$
$F^{-T}(k+1,k)S(k)$	$2n^3-n^2$
$A(k)=[Q(k)+S^{-1}(k)]F^{-T}(k+1,k)S(k)$	$2n^3-n^2$
$C(k)=Q(k)F^{-T}(k+1,k)S(k)$	$2n^3-n^2$
$D(k)G(k)$	$2n^3-n^2$
$C(k)+D(k)G(k)$	n^2
$B(k)G(k)$	$2n^3-n^2$
$A(k)+B(k)G(k)$	n^2
$[A(k)+B(k)G(k)]^{-1}$	$\frac{1}{6}(16n^3-3n^2-n), n \geq 2$ $1, \quad n=1$
$G(k+1)=[C(k)+D(k)G(k)]$ $[A(k)+B(k)G(k)]^{-1}$	$2n^3-n^2$
$H^T(k+1)H(k+1)$	$n^2m+nm-\frac{1}{2}n^2-\frac{1}{2}n$
$[H^T(k+1)H(k+1)]^{-1}$	$\frac{1}{6}(16n^3-3n^2-n), n \geq 2$ $1, \quad n=1$
$\xi(k+1)=[H^T(k+1)H(k+1)]^{-1}$ $H^T(k+1)$	$2n^2m-nm$
$K(k+1)=G(k+1)\xi(k+1)$	$2n^2m-nm$
$I-G(k+1)$	n
$F(k+1,k)x(k/k)$	$2n^2-n$
$[I-G(k+1)]F(k+1,k)x(k/k)$	$2n^2-n$
$K(k+1)z(k+1)$	$2nm-n$
$x(k+1/k+1)=K(k+1)z(k+1)$ $+[I-G(k+1)]F(k+1,k)x(k/k)$	n

TVKFGe	
Matrix Operation	**Calculation Burden**
$R^{-1}(k)$	$\frac{1}{6}(16m^3-3m^2-m), m \geq$ $1, \quad m=1$
$H^T(k)R^{-1}(k)$	$2n^3-n^2$
$S(k)=H^T(k)R^{-1}(k)H(k)$	$n^3+\frac{1}{2}n^2-\frac{1}{2}n$
$S^{-1}(k)$	$\frac{1}{6}(16n^3-3n^2-n), n \geq 2$ $1, \quad n=1$
$Q(k)+S^{-1}(k)$	$\frac{1}{2}n^2+\frac{1}{2}n$
$F^{-T}(k+1,k)$	$\frac{1}{6}(16n^3-3n^2-n), n \geq 2$ $1, \quad n=1$
$F^{-T}(k+1,k)S(k)$	$2n^3-n^2$
$[Q(k)+S^{-1}(k)]F^{-T}(k+1,k)S(k)$	$2n^3-n^2$
$a(k)=H(k)[Q(k)+S^{-1}(k)]$ $F^{-T}(k+1,k)S(k)$	$2n^3-n^2$
$b(k)=H(k)F(k+1,k)$	$2n^3-n^2$
$c(k)=Q(k)F^{-T}(k+1,k)S(k)$	$2n^3-n^2$
$d(k)K(k)$	$2n^3-n^2$
$c(k)+d(k)K(k)$	n^2
$b(k)K(k)$	$2n^3-n^2$
$a(k)+b(k)K(k)$	n^2
$[a(k)+b(k)K(k)]^{-1}$	$\frac{1}{6}(16n^3-3n^2-n), n \geq 2$ $1, \quad n=1$
$K(k+1)=[c(k)+d(k)K(k)]$ $[a(k)+b(k)K(k)]^{-1}$	$2n^3-n^2$
$K(k+1)H(k+1)$	$2n^3-n^2$
$I-K(k+1)H(k+1)$	n
$F(k+1,k)x(k/k)$	$2n^2-n$
$[I-K(k+1)H(k+1)]F(k+1,k)x(k/k)$	$2n^2-n$
$K(k+1)z(k+1)$	$2n^2-n$
$x(k+1/k+1)=K(k+1)z(k+1)$ $+[I-K(k+1)H(k+1)]F(k+1,k)x(k/k)$	n

TIKFG	
Matrix Operation	**Calculation Burden**
$D(k)G(k)$	$2n^3 - n^2$
$C(k) + D(k)G(k)$	n^2
$B(k)G(k)$	$2n^3 - n^2$
$A(k) + B(k)G(k)$	n^2
$[A(k) + B(k)G(k)]^{-1}$	$\frac{1}{6}(16n^3 - 3n^2 - n), n \geq 2$ $1, \quad n = 1$
$G(k+1) = [C(k) + D(k)G(k)]$ $[A(k) + B(k)G(k)]^{-1}$	$2n^3 - n^2$
$K(k+1) = G(k+1)\xi$	$2n^2m - nm$
$I - G(k+1)$	n
$Fx(k/k)$	$2n^2 - n$
$[I - G(k+1)]Fx(k/k)$	$2n^2 - n$
$K(k+1)z(k+1)$	$2nm - n$
$x(k+1/k+1) = [I - G(k+1)]Fx(k/k)$ $+K(k+1)z(k+1)$	n

TIKFGe	
Matrix Operation	**Calculation Burden**
$dK(k)$	$2n^3 - n^2$
$c + dK(k)$	n^2
$bK(k)$	$2n^3 - n^2$
$a + bK(k)$	n^2
$[a + bK(k)]^{-1}$	$\frac{1}{6}(16n^3 - 3n^2 - n), n \geq 2$ $1, \quad n = 1$
$K(k+1) = [c + dK(k)]$ $[a + bK(k)]^{-1}$	$2n^3 - n^2$
$K(k+1)H$	$2n^3 - n^2$
$I - K(k+1)H$	n
$Fx(k/k)$	$2n^2 - n$
$[I - K(k+1)H]Fx(k/k)$	$2n^2 - n$
$K(k+1)z(k+1)$	$2n^2 - n$
$x(k+1/k+1) = [I - K(k+1)H]Fx(k/k)$ $+K(k+1)z(k+1)$	n

SSKFG	
Matrix Operation	**Calculation Burden**
$[I - KH]Fx(k-1/k-1)$	$2n^2 - n$
$Kz(k)$	$2nm - n$
$x(k/k) = [I - KH]Fx(k-1/k-1) + Kz(k)$	n

SSKFGe	
Matrix Operation	**Calculation Burden**
$[I - KH]Fx(k-1/k-1)$	$2n^2 - n$
$Kz(k)$	$2n^2 - n$
$x(k/k) = [I - KH]Fx(k-1/k-1) + Kz(k)$	n

References:

[1] Kalman R. E., A new approach to linear filtering and prediction problems, *Journal of Basic Engineering*, vol. 82, Issue 1, 1960, pp. 35–45.

[2] Ali A, Terada K., Object detection and tracking using Kalman filter and fast mean shift algorithm, *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*, 2009, pages 585-589, DOI: 10.5220/0001787705850589

[3] Cadena C., Neira J., SLAM in O(log n) with the Combined Kalman-Information Filter, *Robotics and Autonomous Systems*, vol. 58, 2010, pp. 1207-1219.

[4] Shankar R., Chatterjee K., Chatterjee T. K., A Very Short-Term Load forecasting using Kalman filter for Load Frequency Control with Economic Load Dispatch, J*ournal of Engineering Science and Technology Review*, Vol. 5, No. 1, 2012, pp. 97-103.

[5] Auger, F., Hilairet, M., Guerrero, J. M., Monmasson, E., Orlowska-Kowalska, T., & Katsura, S., Industrial applications of the Kalman filter: a review, *IEEE Transactions on Industrial Electronics*, Vol. 60, 2013, https://doi.org/10.1109/TIE.2012.2236994

[6] Xu Y., Zhang G., Application of Kalman Filter in the Prediction of Stock Price, *International Symposium on Knowledge Acquisition and Modeling (KAM 2015)*, 2015.

[7] Giunta G., Vernazza R., Salerno R., Ceppi A., Ercolani G., Mancini M., Hourly weather forecasts for gasturbine power generation, *Meteorol. Z.*, Vol. 26, 2017, pp. 307–317.

[8] Ren X., Yang Y., Zhu J., Xu T., Comparing satellite orbit determination by batch processing and extended Kalman filtering using inter-satellite link measurements of the next-generation beidou satellites, *Gps Solutions* Vol. 23, 2019.

[9] Yang Y., Yu T., Zhao W., Zhu X., Kalman Filter Photovoltaic Power Prediction Model Based on Forecasting Experience, *Front. Energy Res.*, *Sec. Smart Grids*, Vol. 9, 2021, https://doi.org/10.3389/fenrg.2021.682852.

[10] Rao V. C. S., Devi B. G., Pratapagiri S., Srinivas C., Venkatramulu S., Raghavakumari, D., Prediction of Covid-19 using Kalman filter algorithm", AIP Conference Proceedings, Vol. 2418, Issue 1, 2022, id.030067, 8 pp., DOI: 10.1063/5.0081995.

[11] Wang X., Yang Y., Wang B., Lin Y., Han C., Resilient timekeeping algorithm with multi-

observation fusion Kalman filter, *Satellite Navigation*, Vol. 4, 2023.
[12] Diaz M., Charbonnel P.-É., Chamoin L., A new Kalman filter approach for structural parameter tracking: Application to the monitoring of damaging structures tested on shaking-tables, *Mechanical Systems and Signal Processing*, Vol.182, 2023.
[13] Agrahar C., Poole W., Bianco S., El-Samad H., Pathspace Kalman Filters with Dynamic Process Uncertainty for Analyzing Time-course Data, 2024, arXiv:2402.04498v1 [stat.ML] 7 Feb 2024
[14] Jung H., Kang S., Kim T., Kim H., ConfTrack: Kalman Filter-based Multi-Person Tracking by Utilizing Confidence Score of Detection Box, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024, pp. 6583-6592
[15] Anderson B. D. O., Moore J. B., *Optimal filtering*, New York: Dover Publications, 2005.
[16] Zhang J., Liu J., 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.
[17] Wang L., An improved iterative method for solving the discrete algebraic Riccati equation, *Mathematical Problems in Engineering*, Vol. 2020, Article ID 3283157, 2020, https://doi.org/10.1155/2020/3283157
[18] Şuşcă M., Mihaly V., Stănese M., Dobra P., Iterative Refinement Procedure for Solutions to Algebraic Riccati Equations, *2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, Cluj-Napoca, Romania, 2020, pp. 1-5, doi: 10.1109/AQTR49680.2020.9130027.
[19] Sima, V., Benner, P. (2020). Computational Experience with a Modified Newton Solver for Discrete-Time Algebraic Riccati Equations. In: Gusikhin, O., Madani, K. (eds) Informatics in Control, Automation and Robotics. ICINCO 2018. *Lecture Notes in Electrical Engineering*, vol. 613. Springer, Cham.
[20] Liu J., Wang Z., Xie Z., Iterative algorithms for reducing inversion of discrete algebraic Riccati matrix equation, *IMA Journal of Mathematical Control and Information*, Vol. 39, 2022, pp. 985–1007.
[21] Assimakis N., Adam M., Iterative and algebraic algorithms for the computation of the steady state Kalman filter gain, *ISRN Applied Mathematics*, Vol. 2014, Article ID 417623, 2014, http://dx.doi.org/10.1155/2014/417623
[22] Horn R.A., Johnson C.R., *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 2005.
[23] Assimakis N., Adam M., Discrete time Kalman and Lainiotis filters comparison, Int. Journal of Mathematical Analysis (IJMA), Vol. 1, No. 13, 2007, pp. 635-659.
[24] Lainiotis D.G., Katsikas S.K., Likothanassis S.D., Optimal seismic deconvolution, Signal Processing, Vol. 15, 1988, pp. 375–404.
[25] Augustine M.T., A note on linear quadratic regulator and Kalman filter, 2023, DOI: 10.48550/arXiv.2308.15798

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

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

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

No funding was received for conducting this study.

Conflict of Interest

The author has no conflict of interest to declare that is relevant to the content of this article.