International Journal of Engineering & Technology, 7 (4.38) (2018) 1655-1658

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Incorporation of Matrix Form in Time-Varying Finite Memory Structure Filter

Pyung Soo Kim[1*]

[1] *System Software Solution Lab., Department of Electronic Engineering, Korea Polytechnic University, 237 Sangidaehak-ro, Siheung-si, Gyeonggi-do, 429-793, KOREA*
**Corresponding author E-mail: pspeter.kim@gmail.com*

Abstract

This paper develops a computationally efficient algorithm for the time-varying finite memory filter with matrix form under a weighted least square criterion using only finite observations on the most recent window. Firstly, the time-varying finite memory filter is represented in matrix form as an alternative of recursive form. Secondly, a computationally efficient algorithm is derived to obtain the numerical stability for improving computational reliability and the amenability for the parallel and systolic implementation, which can reduce computational burden. The computationally efficient algorithm is derived from the recursive form of time-varying finite memory filter by applying a square-root strategy. Through computer simulations for a sinusoid signal and diverse window lengths, the proposed algorithm can be shown to be better than the infinite memory filtering based algorithm for the temporarily uncertain system.

Keywords: *Time-varying system, finite memory filter, infinite memory filter, computational efficiency, square-root strategy.*

1. Introduction

As an alternative to time-invariant infinite memory filter and smoother [1][2], the finite memory filter and smoother [3]-[7] have been developed for state estimation in time-invariant systems. These finite memory filter and smoother have been known there can be several inherent good properties such as unbiasedness and deadbeat while these properties cannot be obtained by infinite memory filter and smoother. The finite memory filter and smoother shows inherently BIBO stability and robustness for temporary uncertainties such as uncertain model parameters and round-off errors, whereas the infinite memory filter and smoother with the recursive structure that tends to accumulate the filtering error as time goes. Hence, these finite memory filter and smoother have been applied successfully for various applications such as positioning system, self-localization, fault detection, etc., as shown in [8]-[12].

However, as shown in [13]-[16], time-varying systems can be often used for many practical and real-time applications. Thus, the finite memory filter for time-varying systems is necessary. In addition, in order to improve computational reliability and overcome computational burden, the computational efficiency should be considered for the implementation of the time-varying finite memory filter.

Therefore, this paper proposes a time-varying finite memory filter with computational efficiency. The proposed finite memory filter is developed by optimizing a weighted least square criterion using only finite observations on the most recent window. It is noted that the recursive form of the finite memory filter for time-varying systems was developed already in [3]. Hence, in this paper, the proposed filter is represented in matrix form as an alternative. For the numerical stability and the amenability to parallel and systolic implementation, a computationally efficient algorithm is considered. The square-root strategy has been preferred for computationally efficient implementation of the infinite memory filtering and smoothing formulas. The square-root strategy is applied for the recursive form of the proposed filter to obtain the computationally efficient algorithm. The proposed filter can be shown to be better than the existing infinite memory filter for the temporarily uncertain system through computer simulations for a sinusoid signal.

This paper is organized as follows. In Section 2, a computationally efficient algorithm for time-varying finite memory filter is proposed. In Section 3, computer simulations are performed. Finally, concluding remarks are presented in Section 4.

2. Computationally Efficient Algorithm for Time-varying Finite Memory Filter

2.1 Time-Varying Finite Memory Filter

A time-varying system can be modeled by state-space model

$$\begin{aligned} x_{i+1} &= \Phi_i x_i + G_i w_i, \\ z_i &= H_i x_i + v_i, \end{aligned} \tag{1}$$

where x_i is unknown state and z_i is a measured observation. The system noise w_i and the observation noise v_i are random variable with zero-mean white Gaussian whose covariances Q_i and R_i.

The time-varying system (1) can be represented in a matrix form on the most recent time interval $[i-M, i]$. This finite interval is called the observation window and M is called the window length. For simplicity, the window initial time $i-M$ will be denoted by i_M hereafter. On the most recent window $[i_M, i]$, the finite number of observations is represented by the state x_i at the present time i as

$$Z_{i-1} = \Gamma_{i-1} x_i + \Lambda_{i-1} W_{i-1} + V_{i-1} \tag{2}$$

where $Z_{i-1}, W_{i-1}, V_{i-1}$ are denoted by

$$Z_{i-1} \equiv \begin{bmatrix} z_{i-M}^T & z_{i-M+1}^T & \cdots & z_{i-1}^T \end{bmatrix}^T,$$
$$W_{i-1} \equiv \begin{bmatrix} w_{i-M}^T & w_{i-M+1}^T & \cdots & w_{i-1}^T \end{bmatrix}^T,$$
$$V_{i-1} \equiv \begin{bmatrix} v_{i-M}^T & v_{i-M+1}^T & \cdots & v_{i-1}^T \end{bmatrix}^T,$$

and matrices are denoted by

$$\Gamma_{i-1} \equiv \begin{bmatrix} H_{i_M} \Psi_{i,M}^{-1} \\ H_{i-M+1} \Psi_{i,M-1}^{-1} \\ \vdots \\ H_{i-1} \Psi_{i,2}^{-1} \\ H_{i-1} \Psi_{i,1}^{-1} \end{bmatrix}, \ \Psi_{i,j} \equiv \Phi_{i-1} \Phi_{i-2} \cdots \Phi_{i-j},$$

$$\Lambda_{i-1} \equiv \begin{bmatrix} H_{i-1}\Psi_{i,1}^{-1}G_{i-1} & H_{i-2}\Psi_{i,2}^{-1}G_{i-2} & \cdots & H_{i_M+1}\Psi_{i,M-1}^{-1}G_{i_M+1} & H_{i_M}\Psi_{i,M}^{-1}G_{i_M} \\ 0 & H_{i-1}\Psi_{i,1}^{-1}G_{i-1} & \cdots & H_{i_M+2}\Psi_{i,M-2}^{-1}G_{i_M+2} & H_{i_M+1}\Psi_{i,M-1}^{-1}G_{i_M+1} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & H_{i-1}\Psi_{i,1}^{-1}G_{i-1} \end{bmatrix}.$$

It is noted in (2) that since noises w_i and v_i are zero-mean white Gaussian random variables, the noise term $\Lambda_{i-1} W_{i-1} + V_{i-1}$ can be also a zero-mean with covariance Ξ_{i-1} given by

$$\Xi_{i-1} \equiv \Lambda_{i-1} \left[diag\left(\overbrace{Q_{i_M} \quad Q_{i_M+1} \quad \cdots \quad Q_{i-1}}^{M} \right) \right] \Lambda_{i-1}^T + \left[diag\left(\overbrace{R_{i_M} \quad R_{i_M+1} \quad \cdots \quad R_{i-1}}^{M} \right) \right]. \tag{3}$$

Now, to get the time-varying finite memory filter $\hat{x}_i$ given the finite observations Z_{i-1} on the most recent window $[i_M, i]$, the following cost function with weighted least square criteria must be minimized:

$$\left[Z_{i-1} - \Gamma_{i-1} x_i \right]^T \Xi_{i-1}^{-1} \left[Z_{i-1} - \Gamma_{i-1} x_i \right] \tag{4}$$

Taking a derivation of (4) with respect to x_i and setting it to zero, the time-varying finite memory filter $\hat{x}_i$ is given by following simple matrix form:

$$\hat{x}_i = \left(\Gamma_{i-1}^T \Xi_{i-1}^{-1} \Gamma_{i-1} \right)^{-1} \Gamma_{i-1}^T \Xi_{i-1}^{-1} Z_{i-1} \tag{5}$$

where Γ_{i-1} and Ξ_{i-1} are given by (4) and (5), respectively. Since the matrix Ξ_{i-1} is positive definite, the matrix $\Gamma_{i-1}^T \Xi_{i-1}^{-1} \Gamma_{i-1}$ is nonsingular if and only if the matrix Γ_{i-1} is of full rank. The matrix Γ_{i-1} is of full rank if $\{\Phi_i, H_i\}$ is observable for $M \ge n$.

However, in the gain matrix $\left(\Gamma_{i-1}^T \Xi_{i-1}^{-1} \Gamma_{i-1} \right)^{-1} \Gamma_{i-1}^T \Pi_{i-1}^{-1}$ in (5), the inversion computation of matrices Ξ_{i-1} and $\Gamma_{i-1}^T \Xi_{i-1}^{-1} \Gamma_{i-1}$ is required. The dimension of these matrices becomes large as the window length M increases. In this case, the computation amount for the filter gain matrix increases. Therefore, since the filter gain must be computed newly for every windows, computational load might be very burdensome for time-varying systems. Hence, the time-varying finite memory filter $\hat{x}_i$ (5) with a matrix form is required to be represented in a recursive form on the window for computational advantage.

Define

$$\Sigma_{i_M+j+1} \equiv \Gamma_{i_M+j}^T \Xi_{i_M+j}^{-1} \Gamma_{i_M+j}, \tag{6}$$

then (6) can be represented by

$$\begin{aligned} \Sigma_{i_M+j+1} \equiv & \Big[I + \Phi_{i_M+j}^{-T} \left(\Sigma_{i_M+j} + H_{i_M+j}^{-T} R_{i_M+j}^{-1} H_{i_M+j} \right) \\ & \Phi_{i_M+j}^{-1} G_{i_M+j} Q_{i_M+j} G_{i_M+j}^T \Big] \\ & \Phi_{i_M+j}^{-T} \left(\Sigma_{i_M+j} + H_{i_M+j}^{-T} R_{i_M+j}^{-1} H_{i_M+j} \right) \Phi_{i_M+j}^{-1}, \end{aligned} \tag{7}$$

with $\Sigma_{i_M} = 0$. Note that $\Sigma_{i_M} = 0$ should be satisfied to obtain the same Σ_{i_M+1} in (6) and (7). Using (7), the estimate $\hat{x}_i$ (5) can be expressed by

$$\Sigma_i \hat{x}_i \equiv \Gamma_{i_M+j}^T \Xi_{i_M+j}^{-1} Z_{i-1},$$

where this can be obtained from the following subsidiary filter $\hat{\mu}_{i_M+1}$ at time $i_M + 1$ defined as

$$\begin{aligned} \hat{\mu}_{i_M+j} & \equiv \Sigma_{i_M+j} \hat{x}_{i_M+j} \\ & \equiv \Gamma_{i_M+j-1}^T \Xi_{i_M+j-1}^{-1} Z_{i_M+j-1}. \end{aligned}$$

Then, the subsidiary filter $\hat{\mu}_{i_M+j}$ can be represented in the recursive form on the window as follows:

$$\begin{aligned} \hat{\mu}_{i_M+j+1} & = \Gamma_{i_M+j}^T \Xi_{i_M+j}^{-1} Z_{i_M+j} \\ & = \Big[I + \Phi_{i_M+j}^{-T} \left(\Sigma_{i_M+j} + H_{i_M+j}^T R_{i_M+j}^{-1} H_{i_M+j} \right) \\ & \quad \Phi_{i_M+j}^{-1} G_{i_M+j} Q_{i_M+j} G_{i_M+j}^T \Big] \\ & \quad \Phi_{i_M+j}^{-T} \left[\hat{\mu}_{i_M+j} + H_{i_M+j}^T R_{i_M+j}^{-1} z_{i_M+j} \right], \ 0 \le j \le M-1. \end{aligned}$$

Then, the time-varying finite memory filter $\hat{x}_i$ is obtained from the recursive form (5) on the window $[i_M, i]$ as follow:

$$\hat{x}_i = \Sigma_i^{-1} \hat{\theta}_i, \tag{8}$$

where Σ_i is given by (7). Note that the recursive form in (8) does not require the inversion computation of matrices with large dimension. Therefore, the recursive form has the computational advantage for the matrix form.

2.2 Computationally Efficient Algorithm

The matrix form (5) and the recursive form (8) of the proposed time-varying finite memory filter have been derived. It can be seen that the matrix form (5) of the time-varying finite memory filter for large M requires a large number of multiplications and thus a large memory for the filter gain matrix $\left(\Gamma_{i-1}^T \Xi_{i-1}^{-1} \Gamma_{i-1}\right)^{-1} \Gamma_{i-1}^T \Xi_{i-1}^{-1}$. On the other hand, the recursive form (8) of the time-varying finite memory filter can be more useful for computational burden and memory requirement than the matrix form (5).

A more computationally efficient algorithm is derived to obtain the numerical stability for improving computational reliability and the amenability for the parallel and systolic implementation, which can reduce computational burden. The computationally efficient algorithm is derived from the recursive form of time-varying finite memory filter by applying a square-root strategy. The square-root strategy for state estimation have been preferred for computationally efficient implementation of the infinite memory filtering and smoothing formulas [13]-[15]. Since the matrix Σ_{i_M+j} propagated by the time-varying Riccati equation, this matrix can lose the positive-definiteness theoretically required due to the accumulation of numerical errors such as coefficient quantization and roundoff errors. In special case, the diagonal entries of Σ_{i_M+j} can become negative and thus this can cause absolutely meaningless state estimates. Hence, it is widely recommended to propagate square-root factors $\Sigma_{i_M+j}^{1/2}$ in order to resolve this issue. The matrix product $\Sigma_{i_M+j}^{1/2} \Sigma_{i_M+j}^{T/2}$ is much more likely to lead to be positive definite because the diagonal elements of this matrix product can always be positive. Of course, there can be numerical effects in this case. Thus, from this observation, a square-root algorithm can be shown to provide the numerical stability and thus improve computational reliability. As shown in the recursive form (10), the computation of the state estimate $\hat{x}_i$ requires the time-consuming computation of Σ_{i_M+j} in (8). Hence, the propagation of square-root factors $\Sigma_{i_M+j}^{1/2}$ can be advantageous in the amenability to easier parallel and systolic implementation and thus overcome computational burden.

As mentioned before, the time-varying finite memory filter can be often robust for numerical errors because of its finite memory structure. Hence, when applying the time-varying finite memory filter, the square-root algorithm can be more required to obtain the amenability for parallel and systolic implementation. The alternative subsidiary filter is defined by

$$\hat{\mu}_{i_M+j} \equiv \Sigma_{i_M+j}^{1/2} \hat{x}_{i_M+j},$$

on the most recent window $[i_M, i]$ and applying both inner and cross products of the array rows using the square-root factor $\Sigma_{i_M+j}^{1/2}$ and the state estimate $\hat{x}_{i_M+j}$. Then, the square-root algorithm can be derived for the recursive form (8) of the time-varying finite memory filter on the most recent window $[i_M, i]$ as

$$\begin{bmatrix} -\Phi_{i_M+j}^{-T} H_{i_M+j}^T R_{i_M+j}^{-1/2} & \Phi_{i_M+j}^{-T} \Sigma_{i_M+j}^{T/2} & 0 \\ Q_{i_M+j}^{1/2} G_{i_M+j}^T \Phi_{i_M+j}^{-T} H_{i_M+j}^T R_{i_M+j}^{-1/2} & -Q_{i_M+j}^{1/2} G_{i_M+j}^T \Phi_{i_M+j}^{-T} \Sigma_{i_M+j}^{-1/2} & I \\ --- & --- & -- \\ -z_{i_M+j}^T & \mu_{i_M+j}^T & 0 \end{bmatrix} \times \Theta_{i_M+j} = \begin{bmatrix} 0 & \Sigma_{i_M+j+1}^{T/2} & * \\ 0 & 0 & * \\ -- & --- & -- \\ * & \mu_{i_M+j+1}^T & * \end{bmatrix},$$

where $0 \le j \le M-1$, $\hat{\mu}_{i_M} = \Sigma_{i_M}^{1/2} \hat{x}_{i_M} = 0$. It is noted that "*" indicates a redundant entry and Θ_{i_M+j} is any orthogonal rotation and satisfies, $\Theta_{i_M+j} \Theta_{i_M+j}^T = I$, that upper-triangularizes the first two rows of the pre-array. Thus, an intermediate variable for the actual state estimate $\hat{x}_i$ at the present time i can be given by the entries of the post array with solving the triangular system system $\hat{\mu}_{i_M+j+1}$. When the available value $\Sigma_i^{-1/2}$ is given, the ultimate actual state estimate $\hat{x}_i = \Sigma_i^{-1/2} \hat{\mu}_i$ at the present time i can be obtained. Since ultimate actual state estimate $\hat{x}_i$ is obtained from products of quantities, the square-root algorithm for the time-varying finite memory filter can be shown to be amenable to parallel and systolic implementation.

3. Computer Simulations

To verify the proposed algorithm and to compare with the existing infinite memory filtering based algorithm, computer simulations are performed for a sinusoid signal model with an uncertain model parameter as follows:

$$\Phi_i = \begin{bmatrix} \cos(\pi/32) + \delta_i & \sin(\pi/32) \\ -\sin(\pi/32) & \cos(\pi/32) + \delta_i \end{bmatrix}$$
$$G_i = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \ H_i = \begin{bmatrix} 1 + 0.25\delta_i & 0.25\delta_i \end{bmatrix} \quad (9)$$

For the noisy sinusoid signal model (9), system and observation noise covariances are taken as $Q_i = 0.1^2$ and $R_i = 0.2^2$, respectively.

The uncertain model parameter is taken as $\delta_i = 0.04$ for the interval $100 \le i \le 150$ for the sinusoid signal model (9). As shown in Figure 1, the estimation errors of the proposed algorithm with three kinds of window lengths $M = 10$, $M = 15$, and $M = 20$ are smaller than that of the existing infinite memory filtering based algorithm on the interval where modeling uncertainty exists. Moreover, the convergence rate of the estimation error of the proposed algorithm is much faster than that of the existing infinite memory filtering based algorithm after temporary modeling uncertainty disappears. Of course, the existing infinite memory filtering based algorithm can outperform the proposed algorithm after the effect of temporary modeling uncertainty is completely gone. Therefore, the proposed algorithm can be more robust than the existing infinite memory filtering based algorithm for temporarily uncertain systems, although the proposed algorithm is designed with no consideration of robustness.

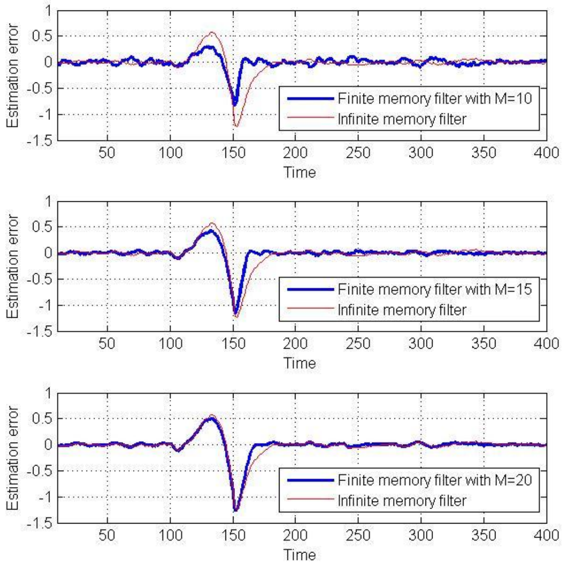

Figure 1: Estimation error for sinusoid signal model with temporary modeling uncertainty for diverse window lengths

4. Conclusion

This paper has proposed a computationally efficient algorithm for the time-varying finite memory filter is developed by optimizing a weighted least square criterion utilizing only finite observations on the most recent window. Firstly, the time-varying finite memory filter has been represented in matrix form as well as recursive form. Secondly, a computationally efficient algorithm has been derived for the numerical stability and the amenability to parallel and systolic implementation. To obtain the computationally efficient algorithm, a square-root strategy has been applied for the recursive form of the time-varying finite memory filter. The square-root strategy has been preferred for computationally efficient implementation of the infinite memory filtering and smoothing formulas. The proposed algorithm has been shown to outperform the existing algorithm when there is temporarily uncertainty through computer simulations for a sinusoid signal.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2017R1D1A1B03033024).

References

[1] R. Faragher, “Understanding the basis of the Kalman filter via a simple and intuitive derivation,” IEEE Signal Processing Magazine, vol. 29, no. 5, pp. 128–132, 2012.
[2] X. Lu, H. Wang, and X. Wang, “On Kalman smoothing for wireless sensor networks systems with multiplicative noises,” Journal of Applied Mathematics, vol. 2012, pp. 1–19, 2012.
[3] P. S. Kim and W. H. Kwon, “Receding horizon FIR filter and its square-root algorithm for discrete time-varying systems,” Transaction on Control, Automation, and System Engineering, vol. 2, no. 2, 2000.
[4] W. H. Kwon, P. S. Kim, S. H. Han, A receding horizon unbiased FIR filter for discrete-time state space models, Automatica 38 (3) (2002) 545–551.
[5] Y. S. Shmaliy, “Linear optimal FIR estimation of discrete time-invariant state-space models”, IEEE Transactions on Signal Processing, vol. 58, no. 6, 3086-3096, 2010
[6] Y. S. Shmaliy, L. J. Morales-Mendoza, “FIR smoothing of discrete-time polynomial signals in state space,” IEEE Transactions on Signal Processing, vol. 58, no. 5, 2544-2555, 2010
[7] P. S. Kim, “A computationally efficient fixed-lag smoother using recent finite measurements,” Measurement, vol. 11, no. 1, pp. 206–210, 2013.
[8] P. S. Kim, “An alternative FIR filter for state estimation in discrete-time systems,” Digital Signal Processing, vol. 20, no. 3, pp. 935–943, 2010.
[9] J. J. Pomarico-Franquiz, M. Granados-Cruz, and Y. S. Shmaliy, “Self-localization over RFID tag grid excess channels using extended filtering techniques,” J. Sel. Topics Signal Processing, vol. 9, no. 2, pp. 229–238, 2015.
[10] P. S. Kim, E. H. Lee, M. S. Jang, S. Y. Kang, “A finite memory structure filtering for indoor positioning in wireless sensor networks with measurement delay,” International Journal of Distributed Sensor Networks 13 (1) (2017) 1–8.
[11] P. S. Kim, “A design of finite memory residual generation filter for sensor fault detection,” Measurement Science Review 17 (2) (2017) 75–81.
[12] M. Vazquez-Olguin, Y. Shmaliy, O. Ibarra-Manzano, “Distributed Unbiased FIR Filtering with Average Consensus on Measurements for WSNs,” IEEE Transactions on Industrial Informatics, 2017.
[13] P. Park, “New square-root algorithms for Kalman filtering,” IEEE Trans. on Automatic Control, vol. 40, no. 5, pp. 895–899, 1995.
[14] P. Wu, X. Li, and Y. Bo, “Iterated square root unscented Kalman filter for maneuvering target tracking using TDOA measurements,” International Journal of Control, Automation and Systems, vol. 11, no. 3, pp. 761–767, 2013.
[15] J. L. Steward, A. Aksoy, Z. S. Haddad, “Parallel direct solution of the ensemble square root Kalman filter equations with observation principal components,” Journal of Atmospheric and Oceanic Technology, vol. 34, no. 9, pp. 1867–1884, 2017.
[16] P. S. Kim, “Time-varying finite memory structure filter to incorporate time-delayed measurements”, Engineering Letters, vol. 26, no. 4, pp. 410–414, 2018.