ITERATIVE ALGORITHMS FOR UNBIASED FIR STATE ESTIMATION IN DISCRETE TIME

°*Yuriy S. Shmaliy*, ⋆*Dan J. Simon*

°Department of Electronics Engineering, Universidad de Guanajuato,
Ctra. Salamanca-Valle, 3.5+1.8km, Palo-Blanco, 36855, Salamanca, Mexico
phone: + 52 (464) 647-99-40, email: shmaliy@ugto.mx
⋆Department of Electrical and Computer Engineering, Cleveland State University,
Cleveland, OH 44115, USA
phone: +123-45-678910, email: d.j.simon@csuohio.edu

ABSTRACT

Various iterative unbiased finite impulse response (UFIR) algorithms are proposed for filtering, smoothing, and prediction of discrete-time state-space models in white Gaussian noise. The distinctive property of UFIR algorithms is that noise statistics are completely ignored. Instead, an optimal window size is required for optimal performance. Under real-world operating conditions with uncertainties, non-Gaussian noise, and unknown noise statistics, the UFIR estimator generally demonstrates better robustness than the Kalman filter, even with suboptimal window size.

1. INTRODUCTION

It is well known from practical experience [1] that implementation of the Kalman filter is often difficult due to the inability in getting a good estimate of the noise covariance matrices. Optimal estimators [2] may thus be less accurate than unbiased ones that are derived under the unbiasedness condition $E\{\hat{\mathbf{x}}_n\} = E\{\mathbf{x}_n\}$, where $\mathbf{x}_n$ indicates a state variable at discrete time n, $\hat{\mathbf{x}}_n$ its estimate, and $E\{x\}$ is the expected value of x. That means that unbiased finite impulse response (UFIR) structures that ignore noise statistics and initial errors are able to produce acceptable suboptimal estimates.

The basic operating principles of the optimal Kalman filter [3] and UFIR filter [4] are summarized in Fig. 1. At time n, the Kalman filter requires the noise statistics at time $n-1$, such as the process and measurement noise covariance matrices $\mathbf{Q}_{n-1}$ and $\mathbf{R}_{n-1}$ respectively, as well as the estimation error covariance $\mathbf{P}_{n-1}$. The optimal UFIR filter ignores these statistics. Instead, it requires the optimal averaging interval of N_{opt} points that can easily be found via measurement [5]. Applications of UFIR filters can be found in many papers [6–11]. Even so, UFIR estimators still remain somewhat beyond the typical range of traditional signal processing techniques.

Below, we discuss a family of iterative UFIR algorithms for filtering, smoothing, and prediction of discrete state-space models in white Gaussian noise. The following definitions will be used: UFIR estimator satisfies the unbiasedness condition, optimal FIR (OFIR) minimizes the mean square error (MSE), and optimal UFIR (OUFIR) estimator minimizes the MSE in the UFIR estimator by N_{opt}.

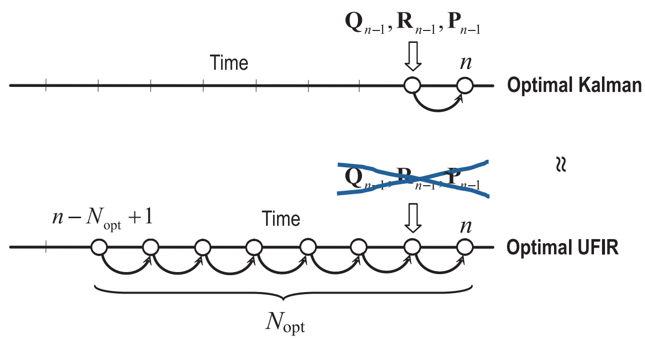

Figure 1: Operating diagrams of the optimal Kalman and OUFIR filters.

2. LINEAR MODEL AND UFIR ESTIMATOR

Consider a class of discrete TV linear models represented in state space with the state and observation equations

$$\mathbf{x}_n = \mathbf{F}_n\mathbf{x}_{n-1} + \mathbf{B}_n\mathbf{w}_n\,, \tag{1}$$

$$\mathbf{z}_n = \mathbf{H}_n\mathbf{x}_n + \mathbf{v}_n\,, \tag{2}$$

where $\mathbf{x}_n \in \Re^K$ and $\mathbf{z}_n \in \Re^M$ are the state and observation vectors, respectively. Here, $\mathbf{F}_n \in \Re^{K\times K}$, $\mathbf{B}_n \in \Re^{K\times P}$, and $\mathbf{H}_n \in \Re^{M\times K}$. The noise vectors, $\mathbf{w}_n \in \Re^P$ and $\mathbf{v}_n \in \Re^M$, have zero mean white Gaussian components, $E\{\mathbf{w}_n\} = \mathbf{0}$ and $E\{\mathbf{v}_n\} = \mathbf{0}$, are mutually uncorrelated, $E\{\mathbf{w}_i\mathbf{v}_j^T\} = \mathbf{0}$, for all i and j, and have covariances $\mathbf{Q}_n = E\{\mathbf{w}_n\mathbf{w}_n^T\}$ and $\mathbf{R}_n = E\{\mathbf{v}_n\mathbf{v}_n^T\}$ which may be unknown to the engineer.

The p-shift estimate $\hat{\mathbf{x}}_{n+p|n}$ of $\mathbf{x}_n$ can be provided at time $n+p$ with the UFIR estimator proposed in [12, 13]. We suppose that $\hat{\mathbf{x}}_{n+p|n}$ is the estimate at $n+p$ via $\mathbf{z}_n$ from the past to n; $p=0$ corresponds to filtering, $p>0$ to p-step prediction, and $p<0$ to q-lag smoothing, where $q=-p$. Sometimes we simplify notation by using $\hat{\mathbf{x}}_{n+p} \triangleq \hat{\mathbf{x}}_{n+p|n}$.

Let us first write the iterative filtering estimate at n, by $p=0$, as

$$\hat{\mathbf{x}}_l = \mathbf{F}_l\hat{\mathbf{x}}_{l-1} + \mathbf{K}_l(\mathbf{z}_l - \mathbf{H}_l\mathbf{F}_l\hat{\mathbf{x}}_{l-1})\,, \tag{3}$$

$$\mathbf{K}_l = \mathbf{G}_l\mathbf{H}_l^T\,, \tag{4}$$

$$\mathbf{G}_l = [\mathbf{H}_l^T\mathbf{H}_l + (\mathbf{F}_l\mathbf{G}_{l-1}\mathbf{F}_l^T)^{-1}]^{-1}\,. \tag{5}$$

The initial values are given by

$$\hat{\mathbf{x}}_s = \mathscr{F}_{s,0}^{m+1}\mathscr{H}_{s,m}^{-1}\mathbf{Z}_{s,m}\,, \tag{6}$$

$$\mathbf{G}_s = \mathscr{F}_{s,0}^{m+1}(\mathbf{H}_{s,m}^T\mathbf{H}_{s,m})^{-1}\mathscr{F}_{s,0}^{m+1\,T}\,, \tag{7}$$

where $\mathscr{H}_{n,m}^{-1} = (\mathbf{H}_{n,m}^T \mathbf{H}_{n,m})^{-1} \mathbf{H}_{n,m}^T$ and

$$\mathbf{H}_{n,m} = \bar{\mathbf{H}}_{n,m} \mathbf{F}_{n,m}\,, \tag{8}$$

$$\mathbf{Z}_{n,m} = \left[\mathbf{z}_n^T \; \mathbf{z}_{n-1}^T \; \ldots \; \mathbf{z}_m^T \right]^T , \tag{9}$$

$$\mathbf{F}_{n,m} = [\underbrace{\mathscr{F}_{n,0}^{m+1^T} \; \mathscr{F}_{n,1}^{m+1^T} \; \ldots \; \mathbf{F}_{m+1}^T \; \mathbf{I}}_{n-m+1}]^T , \tag{10}$$

$$\bar{\mathbf{H}}_{n,m} = \mathrm{diag}(\underbrace{\mathbf{H}_n \; \mathbf{H}_{n-1} \; \ldots \mathbf{H}_m}_{n-m+1}), \tag{11}$$

$$\mathscr{F}_{r,h}^{r-g} = \prod_{i=h}^{g} \mathbf{F}_{r-i} = \mathbf{F}_{r-h} \mathbf{F}_{r-h-1} \ldots \mathbf{F}_{r-g}\,. \tag{12}$$

Here, $s = m + K - 1$ and l ranges from $m + K$ to n. The filter output is taken when $l = n$.

Given $\hat{\mathbf{x}}_n$ from (3) with $l = n$, the p-shift estimate can then be computed as

$$\hat{\mathbf{x}}_{n+p} = \mathscr{B}_{n,m}(p) (\mathscr{F}_{n,0}^{m+1})^{-1} \hat{\mathbf{x}}_n\,, \tag{13}$$

where

$$\mathscr{B}_{n,m}(p) = \begin{cases} \mathscr{F}_{n+p,0}^{m+1} & , \quad p > -N+1\,, \\ \mathbf{I} & , \quad p = -N+1\,, \\ \left(\mathscr{F}_{m,0}^{n+p+1} \right)^{-1} & , \quad p < -N+1\,, \end{cases} \tag{14}$$

As can be seen, the noise statistics are not required by this procedure.

For TI models, the filtering algorithm simplifies to

$$\hat{\mathbf{x}}_l = \mathbf{F} \hat{\mathbf{x}}_{l-1} + \mathbf{K}_l (\mathbf{z}_l - \mathbf{H} \mathbf{F} \hat{\mathbf{x}}_{l-1}) \tag{15}$$

$$\mathbf{K}_l = \mathbf{G}_l \mathbf{H}^T , \tag{16}$$

$$\mathbf{G}_l = [\mathbf{H}^T \mathbf{H} + (\mathbf{F} \mathbf{G}_{l-1} \mathbf{F}^T)^{-1}]^{-1} , \tag{17}$$

with the initial conditions computed by

$$\hat{\mathbf{x}}_s = \mathbf{F}^{s-m} \mathscr{H}_{s,m}^{-1} \mathbf{Z}_{s,m}\,, \tag{18}$$

$$\mathbf{G}_s = \mathbf{F}^{s-m} (\mathbf{H}_{s,m}^T \mathbf{H}_{s,m})^{-1} \mathbf{F}^{s-m^T} . \tag{19}$$

Accordingly, the p-shift estimate can be computed as

$$\hat{\mathbf{x}}_{n+p} = \mathbf{F}^p \hat{\mathbf{x}}_n\,. \tag{20}$$

One may conclude that the algorithm of (15)–(20) is simple from a programming perspective. As has been shown [12–14], it is also a strong rival to the Kalman filter if the noise covariances are not known exactly.

2.1 Estimation Errors

For the estimation error $\varepsilon_{n+p} = \mathbf{x}_{n+p} - \hat{\mathbf{x}}_{n+p}$, the MSE $\mathbf{P}_{n+p}$ at time $n + p$ can be defined as

$$\mathbf{P}_{n+p} = E\{\varepsilon_{n+p} \varepsilon_{n+p}^T\} \tag{21}$$

and the error lower bound (LB) can be shown to be [14]

$$\mathbf{P}_l^{\mathrm{LB}} = (\mathbf{I} - \mathbf{K}_l \mathbf{H}_l) \mathbf{F}_l \mathbf{P}_{l-1}^{\mathrm{LB}} \mathbf{F}_l^T (\ldots)^T + \mathbf{K}_l \mathbf{R}_l \mathbf{K}_l^T . \tag{22}$$

Then the p-shift LB can be computed for TV and TI models as, respectively,

$$\mathbf{P}_{n+p}^{\mathrm{LB}} = \mathscr{B}_{n,m}(p) \mathscr{F}_{n,0}^{m+1^{-1}} \mathbf{P}_n^{\mathrm{LB}} \mathscr{F}_{n,0}^{m+1^{-T}} \mathscr{B}_{n,m}^T(p)\,, \tag{23}$$

$$\mathbf{P}_{n+p}^{\mathrm{LB}} = \mathbf{F}^p \mathbf{P}_n^{\mathrm{LB}} \mathbf{F}^{p^T} , \tag{24}$$

where $\mathbf{P}_n^{\mathrm{LB}}$ is provided from (22) with $l = n$. The LB can also be computed in the three-sigma sense as shown in [4].

Table 1: Fixed-Horizon TV UFIR Filtering Algorithm

Stage	
Given:	$K, N, m = n - N + 1 \geqslant 0$,
	$s = m + K - 1,\ m + K \leqslant l \leqslant n$.
Set:	$\hat{\mathbf{x}}_s$ by (6) and $\mathbf{G}_s$ by (7) .
Update:	$\mathbf{G}_l = [\mathbf{H}_l^T \mathbf{H}_l + (\mathbf{F}_l \mathbf{G}_{l-1} \mathbf{F}_l^T)^{-1}]^{-1}$,
	$\hat{\mathbf{x}}_l = \mathbf{F}_l \hat{\mathbf{x}}_{l-1} + \mathbf{G}_l \mathbf{H}_l^T (\mathbf{z}_l - \mathbf{H}_l \mathbf{F}_l \hat{\mathbf{x}}_{l-1})$.
Instruction:	Use the estimate when $l = n$.

Table 2: Full-Horizon TV UFIR Filtering Algorithm

Stage	
Given:	$K, n \geqslant K$.
Set:	$\hat{\mathbf{x}}_{K-1}$ by (6) and $\mathbf{G}_{K-1}$ by (7) for $m = 0$.
Update:	$\mathbf{G}_n = [\mathbf{H}_n^T \mathbf{H}_n + (\mathbf{F}_n \mathbf{G}_{n-1} \mathbf{F}_n^T)^{-1}]^{-1}$,
	$\hat{\mathbf{x}}_n = \mathbf{F}_n \hat{\mathbf{x}}_{n-1} + \mathbf{G}_n \mathbf{H}_n^T (\mathbf{z}_n - \mathbf{H}_n \mathbf{F}_n \hat{\mathbf{x}}_{n-1})$.

3. UFIR ALGORITHMS

For filtering, smoothing, and prediction, the UFIR algorithms can be represented as in the following.

3.1 Filtering

The filtering UFIR estimate is obtained by (3) or (15) with the estimation error LBs given by (22). There can be recognized several particular solutions.

Fixed-Horizon Filtering: The fixed-horizon (fixed memory size N) iterative UFIR filtering algorithm is summarized for TV models in Table 1. It implies that $N = \mathrm{const}$. Note that the estimation error is minimal if one sets $N = N_{\mathrm{opt}}$ [5]. A simplification for the TI model is straightforward. One must just let all of the matrices be TI in Table 1.

Full-Horizon Filtering: This algorithm given in Table 2 is most simple. It utilizes all the data with $N = n + 1$ and requires only the number of the states K. A natural extension to the TI case is provided by removing the time dependencies from the matrices. The error LB can be computed by (22)–(24) if one substitutes l with n. Note that the full-horizon UFIR filter may demonstrate substantial decrease in the output noise as n becomes large.

Tricky-Horizon Filtering: The tricky-horizon (variable memory size N) algorithm implies an individual N_{opt} at each n. Such flexibility allows for better system tracking with minimum residuals [12] in adaptive systems [15]. To implement tricky-horizon filtering, the algorithm (Table 1) can be used if to let $N = \mathrm{var}$.

3.2 Smoothing

Provided the filtering estimate (3), the TV and TI UFIR smoothers become by (13) and (20) respectively [14]

$$\hat{\mathbf{x}}_{n-q} = \bar{\mathscr{B}}_{n,m}(q) (\mathscr{F}_{n,0}^{m+1})^{-1} \hat{\mathbf{x}}_n\,, \tag{25}$$

$$\hat{\mathbf{x}}_{n-q} = \mathbf{F}^{-q} \hat{\mathbf{x}}_n\,, \tag{26}$$

Table 3: Fixed-Interval TV OUFIR Smoothing Algorithm

Stage	
Given:	K, $N = N_{\text{opt}}$, q, $m = n - N + 1 \geqslant 0$, $s = m + K - 1$, $m + K \leqslant l \leqslant n$.
Set:	$\hat{\mathbf{x}}_s$ by (6) and $\mathbf{G}_s$ by (7) .
Update:	$\mathbf{G}_l = [\mathbf{H}_l^T\mathbf{H}_l + (\mathbf{F}_l\mathbf{G}_{l-1}\mathbf{F}_l^T)^{-1}]^{-1}$, $\hat{\mathbf{x}}_l = \mathbf{F}_l\hat{\mathbf{x}}_{l-1} + \mathbf{G}_l\mathbf{H}_l^T(\mathbf{z}_l - \mathbf{H}_l\mathbf{F}_l\hat{\mathbf{x}}_{l-1})$. Use $\hat{\mathbf{x}}_n$ when $l = n$ and compute $\hat{\mathbf{x}}_{n-q} = \bar{\mathscr{B}}_{n,m}(q)(\mathscr{F}_{n,0}^{m+1})^{-1}\hat{\mathbf{x}}_n$.
Instruction:	Valid for any $n \geqslant N - 1$. The fixed interval of $M = N_{\text{opt}}$ points is from time index m to n.

where

$$\bar{\mathscr{B}}_{n,m}(q) = \begin{cases} \prod_{i=0}^{N-2-q} \mathbf{F}_{n-q-i}\,, & q < N-1\,, \\ \mathbf{I}\,, & q = N-1\,, \\ \left(\prod_{i=1}^{q-N} \mathbf{F}_{m-i}\right)^{-1}, & q > N-1\,. \end{cases} \quad (27)$$

The error LBs become, respectively,

$$\mathbf{P}_{n-q}^{\text{LB}} = \bar{\mathscr{B}}_{n,m}(q)\mathscr{F}_{n,0}^{m+1^{-1}}\mathbf{P}_n^{\text{LB}}\mathscr{F}_{n,0}^{m+1^{-T}}\bar{\mathscr{B}}_{n,m}^T(q)\,, \quad (28)$$

$$\mathbf{P}_{n-q}^{\text{LB}} = \mathbf{F}^{-q}\mathbf{P}_n^{\text{LB}}\mathbf{F}^{-q^T}\,, \quad (29)$$

where $\mathbf{P}_n^{\text{LB}}$ is provided by (22) at $l = n$. As in filtering, here the LB can serve well in the three-sigma sense [4].

Fixed-Interval Smoothing: This algorithm is intended to estimate $\hat{\mathbf{x}}_{n-q|n}$ with any lag $0 < q < M$ utilizing measurement from $n - M + 1$ to n. It is most efficient if $M = N_{\text{opt}}$ as implemented in Table 3. To apply this algorithm to TI models, one must compute $\hat{\mathbf{x}}_{n-q} = \mathbf{F}^{-q}\hat{\mathbf{x}}_n$.

Fixed-Lag Smoothing: Two basic fixed-lag smoothing algorithms can be recognized. Provided N_{opt}, the fixed lag q OUFIR smoothing algorithm is listed in Table 3 if one sets $N = N_{\text{opt}}$ and $q = \text{const}$. Its extension to the TI case can be provided by replacing the $\hat{\mathbf{x}}_{n-q}$ equation with $\hat{\mathbf{x}}_{n-q} = \mathbf{F}^{-q}\hat{\mathbf{x}}_n$. Fixed-lag full-horizon UFIR smoothing implies that the filter window includes all the available data, but the lag is fixed. The relevant algorithm is listed in Table 4. Its extension to the TI case can be obtained by replacing the $\hat{\mathbf{x}}_{n-q}$ equation with $\hat{\mathbf{x}}_{n-q}$ as $\hat{\mathbf{x}}_{n-q} = \mathbf{F}^{-q}\hat{\mathbf{x}}_n$.

Fixed-Point Smoothing: This algorithm implies that measurements are available from 0 up to n, but the estimate is required at some fixed past point $0 \leqslant v < n$, where v is a constant [16]. The time-varying lag is $q = n - v$ and the UFIR smoother is thus always full-horizon (Table 5). By replacing the $\hat{\mathbf{x}}_{n-q}$ equation with $\hat{\mathbf{x}}_{n-q} = \mathbf{F}^{-q}\hat{\mathbf{x}}_n$, it becomes applicable for TI models.

3.3 Prediction

State prediction plays a key role in many applications. Two basic UFIR prediction algorithms can be found in [4].

Table 4: Fixed-Lag Full-Horizon TV UFIR Smoothing Algorithm

Stage	
Given:	K, $q = \text{constant}$, $n \geqslant K$.
Set:	$\hat{\mathbf{x}}_{K-1}$ by (6) and $\mathbf{G}_{K-1}$ by (7) for $m = 0$.
Update:	$\mathbf{G}_n = [\mathbf{H}_n^T\mathbf{H}_n + (\mathbf{F}_n\mathbf{G}_{n-1}\mathbf{F}_n^T)^{-1}]^{-1}$, $\hat{\mathbf{x}}_n = \mathbf{F}_n\hat{\mathbf{x}}_{n-1} + \mathbf{G}_n\mathbf{H}_n^T(\mathbf{z}_n - \mathbf{H}_n\mathbf{F}_n\hat{\mathbf{x}}_{n-1})$. Compute $\hat{\mathbf{x}}_{n-q}$ for $n \geqslant q$ as $\hat{\mathbf{x}}_{n-q} = \bar{\mathscr{B}}_{n,m}(q)(\mathscr{F}_{n,0}^{m+1})^{-1}\hat{\mathbf{x}}_n$.

Table 5: Fixed-Point TV UFIR Smoothing Algorithm

Stage	
Given:	K, $v = \text{constant} \geqslant 0$, $q = n - v$, $n \geqslant K$.
Set:	$\hat{\mathbf{x}}_{K-1}$ by (6) and $\mathbf{G}_{K-1}$ by (7) for $m = 0$.
Update:	$\mathbf{G}_n = [\mathbf{H}_n^T\mathbf{H}_n + (\mathbf{F}_n\mathbf{G}_{n-1}\mathbf{F}_n^T)^{-1}]^{-1}$, $\hat{\mathbf{x}}_n = \mathbf{F}_n\hat{\mathbf{x}}_{n-1} + \mathbf{G}_n\mathbf{H}_n^T(\mathbf{z}_n - \mathbf{H}_n\mathbf{F}_n\hat{\mathbf{x}}_{n-1})$. Compute $\hat{\mathbf{x}}_{n-q}$ for $n > v$ as follows: $\hat{\mathbf{x}}_{n-q} = \bar{\mathscr{B}}_{n,m}(q)(\mathscr{F}_{n,0}^{m+1})^{-1}\hat{\mathbf{x}}_n$.

4. SOME GENERALIZATIONS AND CONCLUSIONS

Based on extensive investigations provided by many authors, now we provide some generalizations, compare the trade-off between the OUFIR, OFIR and Kalman filters, and summarize the results in Table 6.

4.1 OUFIR vs. OFIR

Beginning with the early limited memory filter of Jazwinski [2], OFIR filtering has been under development for several decades. In control theory, fundamental progress was achieved by Kwon *et al.* and his followers [18, 20, 25–28]. In signal processing, solutions were found by Shmaliy *et al.* [4, 13, 19]. It was shown in [28] that different kinds of limited memory filters [2, 17] are equivalent to the OFIR one. The most serious flaws of this technique are high computational complexity and high memory consumption. With such poor engineering features, OFIR estimators still have not gained currency and their development remains mostly at a theoretical level.

On the other hand, OFIR estimators do not result in estimation errors that are substantially smaller than OUFIR ones, especially when $N \gg 1$. The rule of thumb here is as shown in Fig. 2: *The error difference between the OFIR and OUFIR estimates diminishes as N increases*. Note that the boundary value 10...30 in Fig. 2 is flexible and depends on the model. However, recalling that FIR filters require a large order (window size $N \gg 1$) to guarantee good performance, we obtain the following conclusion:

Fast and low-complexity iterative OUFIR algorithms that ignore noise statistics and initial error statistics are practically superior to the best known OFIR ones.

Note that this deduction often holds even if N is small.

Table 6: Critical Evaluation of the Kalman, OFIR, and OUFIR Filters

	Kalman [23]	Batch OFIR [13, 18, 19]	Iterative OFIR [18, 20]	Batch OUFIR [21, 22]	Iterative OUFIR [12, 13]
Optimality:	Optimal	Optimal	Optimal	Unbiased	Unbiased
Initial conditions:	*A priori*	*A posteriori*	*A posteriori*	Ignored	*A posteriori*
Noise statistics:	Required	Required	Required	Ignored	Ignored
Noise:	White	Arbitrary	White	Arbitrary	Arbitrary
System model:	Stochastic	Arbitrary	Arbitrary	Arbitrary	Arbitrary
Memory (points):	2	N_{opt}	N_{opt}	N_{opt}	N_{opt}
Stability:	May diverge	BIBO	BIBO	BIBO	BIBO
Operation:	Fast	Slow	Medium	Medium	$\sim N_{\text{opt}}$ times slower than Kalman; Fast in parallel comp.
Total memory:	Small	Large	Medium	Large	$\sim N_{\text{opt}}$ times more than Kalman
Complexity:	Low	High	Medium	Medium	Low

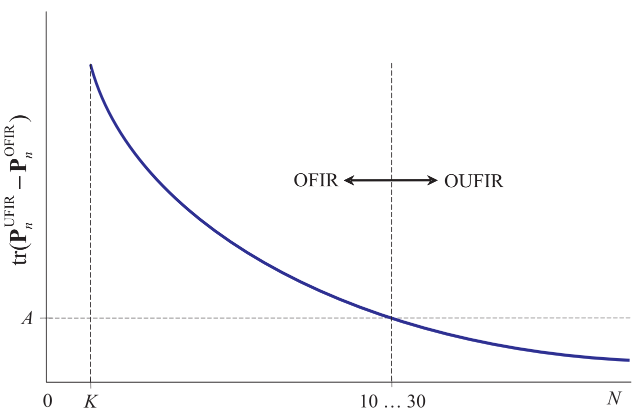

Figure 2: Effect of the estimator window size N on the error difference between the OUFIR and OFIR estimators. Threshold A indicates where the difference becomes visually indistinguishable.

But in some applications, OFIR filters can be more appropriate because of their better accuracy.

4.2 OUFIR vs. Kalman Filter

The well-known features of the Kalman filter are optimality, fast computation, and low memory consumption. However, the Kalman filter requires *a priori* initial condition and noise statistics, and this is recognized as the most annoying flaw of the Kalman filter. Because of this requirement, the Kalman filter is suboptimal for all practical purposes. Moreover, its optimality is guaranteed only if the noise sources are white, which is not the case for many applications. Finally, the Kalman filter applies only to stochastic models.

In turn, the iterative OUFIR filter ignores noise statistics, allows the noise to have any distribution and covariance, exhibits BIBO stability, and serves for both stochastic and deterministic models. However, it does not guarantee optimality, especially when N_{opt} is small. It requires $(N_{\text{opt}} - 1)$-times more computational time and needs about N_{opt} times more memory than the Kalman filter.

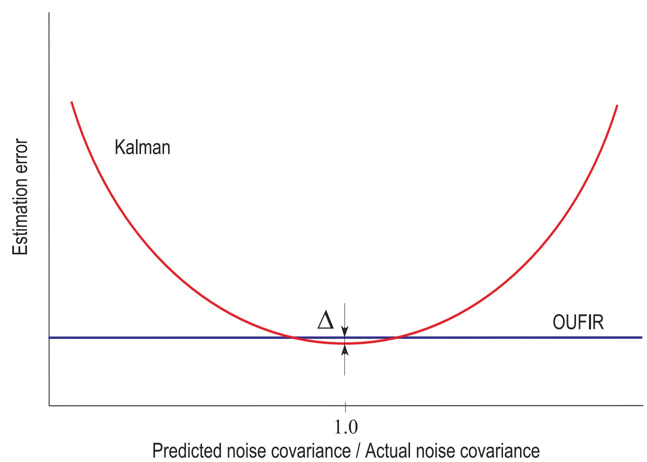

Figure 3: Effect of errors in the noise covariances of the Kalman and OUFIR estimates. The value Δ depends on N and becomes insignificant when $N_{\text{opt}} \gg 1$.

The Kalman filter is thus best when the noise is white and its statistics are exactly known. Otherwise, one may follow the rule of thumb sketched in Fig. 3. As can be seen, it is only within a narrow range around the actual noise covariances that the OUFIR filter falls a bit short of the Kalman filter. Otherwise, the OUFIR filter demonstrates smaller errors. The Kalman filter is also the best filter under the ideal conditions. Otherwise, its error grows more rapidly than the OUFIR, meaning that the latter is more robust in real-world applications (Fig. 4). Note that the error difference Δ between the two filters decreases with increasing N_{opt}. These observations by diverse authors who have investigated uncertainties, different kinds of noise sources, and other irregular perturbations, result in the following important inference:

Under the real world operating conditions, and when noise statistics and initial error statistics are not known exactly, the OUFIR estimator is able to outperform the Kalman filter even if N_{opt} is not large.

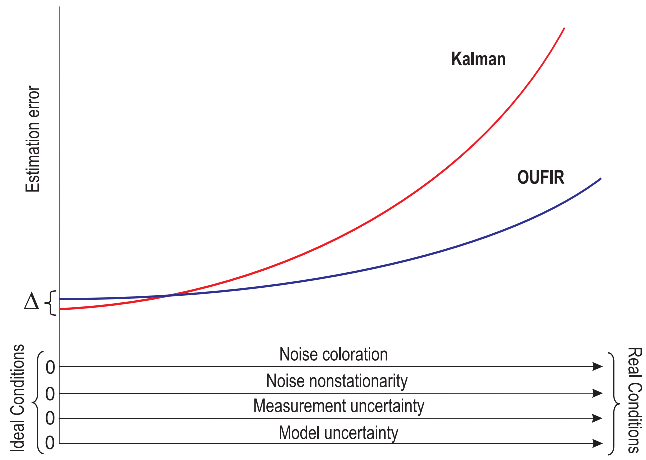

Figure 4: Effect of operating conditions on the Kalman and OUFIR estimates. The value Δ depends on N and becomes insignificant when $N_{\text{opt}} \gg 1$.

Simulation results confirming these observations can be found in [12, 14, 24]. More details about the iterative UFIR estimation algorithms can be found in [29].

REFERENCES

[1] B. Gibbs, *Advanced Kalman Filtering, Least-Squares and Modeling*, New York: John Wiley & Sons, 2011.

[2] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, New York: Academic Press, 1970.

[3] D. Simon, *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches*, Hoboken, NJ: J. Wiley & Sons, 2006.

[4] Y. S. Shmaliy, *GPS-based Optimal FIR Filtering of Clock Models*, New York: Nova Science Publ., 2009.

[5] F. Ramirez-Echeverria, A. Sarr, Y. S. Shmaliy, "Optimal memory for discrete-time FIR filters in state-space", *IEEE Trans. Signal Process.* (to be published).

[6] J. Levine, "Invited Review Article: The statistical modeling of atomic clocks and the design of time scales," *Rev. Sc. Instr.*, vol. 83, pp. 021101-1–021101-28, Feb. 2012.

[7] W.Q. Wang, "Phase noise suppression in GPS-disciplined frequency synchronization systems," *Fluctuation Noise L.*, vol. 10, pp. 303–313, Mar. 2011.

[8] P. Ferrari, A. Flammini, S. Rinaldi, A. Bondavalli, and F. Brancati, "Experimental characterization of uncertainty sources in a software-only synchronization system," *IEEE Trans. Instrum. Meas.*, vol. 61, pp. 1512–1521, May 2012.

[9] Y. Kou, Y. Jiao, D. Xu, M. Zhang, Y. Liu, and X. Li, "Low-cost precise measurement of oscillator frequency instability based on GNSS carrier observation," *Advances in Space Research*, vol. 51, pp. 969–977, Mar. 2013.

[10] S. Mahotra, C. Patlolla, and N. Kehtarnavaz, "Real-time computation of disparity for hand-pair gesture recognition using a stereo webcam," *J. Real-Time Image Process.*, vol. 7, pp. 257–266, Dec. 2012.

[11] L. Morales-Mendoza, R. Vazquez-Bautista, E. Moralez-Mendoza, and Y. S. Shmaliy, "Reduction speckle noise in ultrasound imaging using the key points in the low degree unbiased FIR filters," *Computacin y Sistamas*, vol. 16, pp. 287–295, Jul-Sep. 2012.

[12] Y. S. Shmaliy, "An iterative Kalman-like algorithm ignoring noise and initial conditions," *IEEE Trans. Signal Process.*, vol. 59, pp. 2465–2473, Jun. 2011.

[13] Y. S. Shmaliy, "Linear optimal FIR estimation of discrete time-invariant state-space models," *IEEE Trans. Signal Process.*, vol. 58, pp. 3086–3096, Jun. 2010.

[14] D. Simon and Y. S. Shmaliy, "Unified Forms for Kalman and Finite Impulse Response Filtering and Smoothing," *Automatica*, vol. 49, pp. 1892–1899, Jun. 2013.

[15] F. Gustafsson, *Adaptive Filtering and Change Detection*, New York: J. Wiley & Sons, 2000.

[16] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*, New York: Wiley-Interscience, 2001.

[17] F. C. Schweppe, *Uncertain Dynamic Systems*, Englewood Cliffs: Prentice-Hall, 1973.

[18] O. K. Kwon, W. H. Kwon, and K. S. Lee, "FIR filters and recursive forms for discrete-time state-space models," *Automatica*, vol. 25, no.5, pp. 715–728, Sep. 1989.

[19] Y. S. Shmaliy and O. Ibarra-Manzano, "Time-variant linear optimal finite impulse response estimator for discrete-time state-space models," *Int. J. Adapt. Contr. Signal Process.*, vol. 26, pp. 95–104, Feb. 2012.

[20] S. H. Han, W. H. Kwon, P. S. Kim, "Quasi-deadbeat minimax filters for deterministic state-space models," *IEEE Trans. Automat.Contr.*, vol. 47, pp. 1904–1908, Nov. 2002.

[21] Y. S. Shmaliy, "An unbiased FIR filter for TIE model of a local clock in applications to GPS-based timekeeping," *IEEE Trans. Ultrason., Ferroel. Freq. Contr.*, vol. 53, pp. 862–870, May 2006.

[22] P. S. Kim and M. E. Lee, "A new FIR filter for state estimation and its applications," *J. Comput. Sc. Techn.*, vol. 22, pp. 779–784, Sep. 2007.

[23] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, pp. 35–45, Mar. 1960.

[24] Y. S. Shmaliy, "Suboptimal FIR filtering of nonlinear models in additive white Gaussian noise," *IEEE Trans. Signal Process.*, vol. 60, pp. 5519–5527, Oct. 2012.

[25] C. K. Ahn and P. S. Kim, "Fixed-lag maximum likelihood FIR smoother for state-space models," *IEICE Electonics Express*, vol. 5, pp. 11–16, Jan. 2008.

[26] C. K. Ahn and S. H. Han, "New H_∞ FIR smoother for linear discrete-time state-space models," *IEICE Trans. on Communications*, vol. E91.B, pp. 896–899, Mar. 2010.

[27] W. H. Kwon and S. Han, *Receding horizon control: Model predictive control for state models*, London: Springer-Berlag, 2005.

[28] W. H. Kwon, Y. S. Suh, Y. I. Lee, and O. K. Kwon, "Equivalence of finite memory filters," *IEEE Trans. on Aerospace and Electron. Syst.*, vol. 30, pp. 968–972, Jul. 1994.

[29] Y. S. Shmaliy and D. Simon, "Iterative unbiased FIR state estimation: a review of algorithms," *EURASIP J. Advances Signal Process.* (to be published).