Shmaliy and Simon *EURASIP Journal on Advances in Signal Processing* 2013, **2013**:113
http://asp.eurasipjournals.com/content/2013/1/113

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

REVIEW Open Access

Iterative unbiased FIR state estimation: a review of algorithms

Yuriy S Shmaliy[1]* and Dan Simon[2]

Abstract

In this paper, we develop in part and review various iterative unbiased finite impulse response (UFIR) algorithms (both direct and two-stage) for the filtering, smoothing, and prediction of time-varying and time-invariant discrete state-space models in white Gaussian noise environments. The distinctive property of UFIR algorithms is that noise statistics are completely ignored. Instead, an optimal window size is required for optimal performance. We show that the optimal window size can be determined via measurements with no reference. UFIR algorithms are computationally more demanding than Kalman filters, but this extra computational effort can be alleviated with parallel computing, and the extra memory that is required is not a problem for modern computers. 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. In applications requiring large window size, the UFIR estimator is also superior to the best previously known optimal FIR estimators.

Keywords: Unbiased FIR estimator, Kalman filter, Iterative algorithm, Filtering, Smoothing, Prediction

1 Review

1.1 Introduction

In optimal estimation theory, *unbiasedness* is a key condition that is used to derive linear and nonlinear estimators. A classical example is the *ordinary least squares* (OLS) estimator proposed by Gauss in 1795 [1]. The Gauss-Markov theorem states that if the noise is white and has the same variance at each time step, the OLS estimator is also the *best linear unbiased estimator* (BLUE) [2]. In convolution-based optimal filtering, the unbiasedness constraint [2] leads to the *unbiased finite impulse response* (UFIR) estimator [3,4]. An extremely useful property of the BLUE and UFIR is that noise statistics are not required. Another example is the *maximum likelihood estimator* (MLE), which obtains the estimate at an extremum of the density function of the state conditioned on the measurements [5]. Like the BLUE and UFIR, the MLE is suboptimal for finite data. However, if the sample size (memory) increases to infinity, each of them are optimal.

For finite data, unbiasedness does not guarantee the *minimum mean square error* (MSE), which is comprised of standard deviation and bias:

$$\mathrm{MSE} = \mathrm{Var} + \mathrm{Bias}^2\,, \tag{1}$$

where 'Var' is the error variance. Since the minimum MSE is required by many applications, a minimization of (1) is often desired at the expense of a small increase in bias. That leads to different kinds of optimal solutions such as the *minimum variance unbiased estimator* (MVUE), the recursive *Kalman filter* [6], and the *optimal* FIR (OFIR) filter [7,8]. The common disadvantage of these filters is that noise statistics and initial errors are required. In view of the fact that noise statistics and initial errors are commonly not well known, especially for time-variant models, theoretically optimal estimators end up being suboptimal in practical applications. In this regard, engineering experience says the following [9]:

> *Practical implementation of the Kalman filter is often difficult due to the inability in getting a good estimate of the noise covariance matrices.*

That means that due to insufficient knowledge about noise statistics, optimal estimators that minimize (1) may be less accurate than unbiased ones that are derived under

*Correspondence: shmaliy@ugto.mx
[1] Department of Electronics Engineering, Universidad de Guanajuato, Salamanca, 36855, Mexico
Full list of author information is available at the end of the article

the invariant $E\{\mathbf{x}_n - \hat{\mathbf{x}}_n\} = \mathbf{0}$, which leads to the unbiasedness condition

$$E\{\hat{x}_n\} = E\{x_n\}\,, \tag{2}$$

where $\mathbf{x}_n$ indicates a state variable at discrete time step n, $\hat{\mathbf{x}}_n$ its estimate, and $E\{x\}$ is the expected value of x. Note that the cost of equipment that is required for the characterization of noise statistics cannot commonly be afforded by users, and methods for the estimation of noise covariance matrices via measurements are not well developed. On the other hand, noise statistics are not always necessary to get a good estimate that we illustrate below based on an example.

Example 1. *A linear signal x_n is measured as $z_n = x_n + v_n$ in the presence of zero mean white Gaussian noise v_n having variance $\sigma_v^2 = 1$. The p-shift ramp UFIR estimator matches this signal and is given by the convolution-based estimate*

$$\hat{x}_{n+p} = \sum_{i=0}^{N-1} h_{1i}(N,p) z_{n-i}\,, \tag{3}$$

where the impulse response function is (eq. (89) of [17])

$$h_{1n}(N,p) = \frac{2(2N-1)-6n}{N(N+1)} + \frac{6p(N-1-2n)}{N(N^2-1)}\,, \tag{4}$$

$p = 0$ corresponds to filtering, $p < 0$ corresponds to $|p|$-lag smoothing, and $p > 0$ corresponds to p-step prediction. The estimation variance is defined as $\sigma^2 = G(N,p)\sigma_v^2 = G(N,p)$, where $G(N,p) = \sum_{i=0}^{N-1} h_{1i}^2(N,p)$ is the noise power gain (NPG) [11],

$$G(N,p) = \frac{2(2N-1)}{N(N+1)} + \frac{12p(N-1+p)}{N(N^2-1)}\,. \tag{5}$$

The estimation variance is sketched in Figure 1 as a function of N. Here, the $1/N$ bound is obtained by simple averaging that is optimal in the sense of error variance, although with a 50% bias for a linear x_n. The case of $p = 0$ corresponds to the ramp UFIR filter with $h_{1n}(N,0)$ given by (4), and we notice that denoising is inherently less efficient in this case. If we set $p = -N/2$, the $(N/2)$-lag smoother estimation error variance rapidly converges to that of simple averaging. A similar effect can be observed in the one-step prediction error variance with $p = 1$. Here, a large error variance for small values of N reduces to that of the ramp filter as N increases. A common feature of these plots is that the error variances decrease with the reciprocal of N. That means that noise in UFIR estimators with large memory $N \gg 1$ may be very low and the estimate may be almost optimal. This leads to the following statement:

There is no need to use optimal estimators in many applications. UFIR structures that ignore noise statistics and initial estimation error statistics are able to produce acceptable suboptimal estimates.

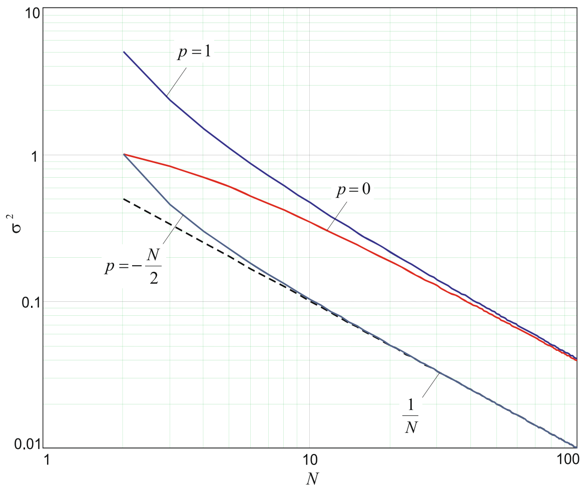

Figure 1 Estimation error variance σ^2 for different UFIR structures (Example 2). $1/N$ corresponds to simple averaging, **$p = 0$** to the ramp filter, **$p = 1$** to the one-step predictor, and **$p = -N/2$** to the **$(N/2)$**-lag smoother.

UFIR estimators have attracted researcher's attention for decades, beginning with the work of Johnson [12] and others, in which they extended the Wiener filter theory to discrete finite time. Further, the ability of UFIR estimators to produce nearly optimal estimates while ignoring noise statistics was greatly regarded in the development of estimators for polynomial signals [13-15]. Most recently UFIR methods were extended to state space in batch form [3,4,16-18] and in an iterative Kalman-like form [19,20]. The latter has made the UFIR estimator a significant rival of the Kalman filter and its applications can be found in [10,21-24]. Even so, UFIR estimators still remain somewhat beyond the typical range of traditional signal processing techniques.

The basic operating principles of the optimal Kalman and UFIR filters are summarized in Figure 2. 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 in order to be optimal.

In this paper, we develop in part the results achieved in the field of UFIR filtering and review a family of iterative UFIR algorithms for filtering, smoothing, and prediction of time-varying (TV) and time-invariant (TI) discrete state-space models in white Gaussian noise environments. The following definitions will be used: UFIR estimator satisfies the unbiasedness condition (2), OFIR estimator minimizes the MSE (1), and Optimal UFIR (OUFIR) estimator minimizes the MSE in the UFIR estimator by using a window size N_{opt}.

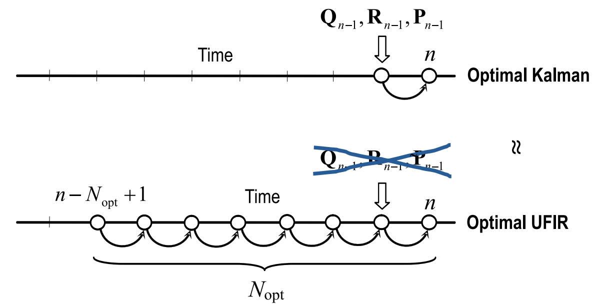
$\mathbf{Q}_{n-1}, \mathbf{R}_{n-1}, \mathbf{P}_{n-1}$
Time
n
Optimal Kalman
≈
$\mathbf{Q}_{n-1}, \mathbf{R}_{n-1}, \mathbf{P}_{n-1}$
$n-N_{\text{opt}}+1$
Time
n
Optimal UFIR
N_{opt}

Figure 2 Basic operating diagrams of the optimal Kalman and UFIR filters.

Section 1.2 presents the linear state-space model, formulates the problem, and considers the batch p-shift UFIR estimator along with the generalized NPG. Section 1.3 presents two forms of the p-shift iterative UFIR algorithm. Section 1.4 discusses the estimation errors of the UFIR estimators. Sections 1.5, 1.6, and 1.7 give the reader a number of practical algorithms for filtering, smoothing, and prediction. Section 1.8 considers an extension to nonlinear systems. Section 1.9 discusses methods for the determination of the optimal memory size N_{opt}. Finally, section 1.10 concludes with some useful generalizations.

1.2 Linear model and batch UFIR estimator

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

$$\mathbf{x}_n = \mathbf{F}_n\mathbf{x}_{n-1} + \mathbf{B}_n\mathbf{w}_n\,, \tag{6}$$

$$\mathbf{z}_n = \mathbf{H}_n\mathbf{x}_n + \mathbf{v}_n\,, \tag{7}$$

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 $\mathbf{H}_n \in \Re^{M\times K}$. Let us suppose that the state noise vector $\mathbf{w}_n \in \Re^P$ and measurement noise vector $\mathbf{v}_n \in \Re^M$ have zero mean white Gaussian components, $E\{\mathbf{w}_n\} = \mathbf{0}$ and $E\{\mathbf{v}_n\} = \mathbf{0}$. We also assume that these vectors are mutually uncorrelated, $E\{\mathbf{w}_i\mathbf{v}_j^T\} = \mathbf{0}$, for all i and j, and have the following covariances:

$$\mathbf{Q}_n = E\{\mathbf{w}_n\mathbf{w}_n^T\}\,, \tag{8}$$

$$\mathbf{R}_n = E\{\mathbf{v}_n\mathbf{v}_n^T\}\,, \tag{9}$$

where $\mathbf{Q}_n$ and $\mathbf{R}_n$ may be unknown to the engineer.

Now suppose that the p-shift estimate[a] $\hat{\mathbf{x}}_{n+p|n}$ of $\mathbf{x}_n$ is provided at time $n+p$ with the UFIR estimator proposed in [19,20]. We would like to modify this estimator and review engineering algorithms for different kinds of filtering, q-lag smoothing, and p-step prediction. We also wish to estimate the estimation errors and generalize the properties to facilitate a comparison with the OFIR and Kalman algorithms.

1.2.1 Time-variant models

In convolution-based filtering (3), we suppose that measurements $\mathbf{z}_n$ are available on a time horizon of N points (memory[b]), from time $m = n - N + 1$ to time n, that the estimator is causal, and that $m \geqslant 0$. In order to find $\hat{\mathbf{x}}_{n+p}$ in state space, the batch p-shift UFIR estimator [8,20] can be applied. For TV models, the p-shift UFIR estimator was derived in [8], assuming that the negative shift p is no smaller than $-N+1$. Below, we modify this estimator for arbitrary p, which is needed for one of the smoother forms.

Let $p = -N+1$ and consider the estimate (eq. (21)) of [19]) at the initial point m that gives us

$$\hat{\mathbf{x}}_m = \mathcal{H}_{n,m}^{-1}\mathbf{Z}_{n,m}\,, \tag{10}$$

where $\mathcal{H}_{n,m}^{-1} = (\mathbf{H}_{n,m}^T\mathbf{H}_{n,m})^{-1}\mathbf{H}_{n,m}^T$ is the generalized left inverse, and

$$\mathbf{H}_{n,m} = \bar{\mathbf{H}}_{n,m}\mathbf{F}_{n,m}\,, \tag{11}$$

$$\mathbf{Z}_{n,m} = \left[\, \mathbf{z}_n^T \; \mathbf{z}_{n-1}^T \; \ldots \; \mathbf{z}_m^T \,\right]^T\,, \tag{12}$$

$$\mathbf{F}_{n,m} = \big[\underbrace{\mathcal{F}_{n,0}^{m+1^T} \; \mathcal{F}_{n,1}^{m+1^T} \; \ldots \; \mathbf{F}_{m+1}^T \; \mathbf{I}}_{n-m+1}\big]^T\,, \tag{13}$$

$$\bar{\mathbf{H}}_{n,m} = \text{diag}\big(\underbrace{\mathbf{H}_n \; \mathbf{H}_{n-1} \; \ldots \; \mathbf{H}_m}_{n-m+1}\big)\,, \tag{14}$$

$$\mathcal{F}_{r,h}^{r-g} = \prod_{i=h}^{g}\mathbf{F}_{r-i} = \mathbf{F}_{r-h}\mathbf{F}_{r-h-1}\ldots\mathbf{F}_{r-g}\,. \tag{15}$$

One can notice that (10) is reminiscent of the familiar OLS or BLUE, although the matrices are different.

To provide the estimate for any p, we find the state transition matrix $\mathcal{B}_{n,m}(N,p)$ by writing (10) as $\hat{\mathbf{x}}_{n+p|n} = \mathcal{B}_{n,m}(N,p)\hat{\mathbf{x}}_{m|n}$. By combining the forward-time and backward-time solutions [26], $\mathcal{B}_{n,m}(p) \triangleq \mathcal{B}_{n,m}(N,p)$ becomes

$$\mathcal{B}_{n,m}(p) = \begin{cases} \mathcal{F}_{n+p,0}^{m+1} = \prod\limits_{i=0}^{N-2+p}\mathbf{F}_{n+p-i} & ,\; p > N_1 \\ \mathbf{I} & ,\; p = N_1\,. \\ \left(\mathcal{F}_{m,0}^{n+p+1}\right)^{-1} = \left(\prod\limits_{i=0}^{|p|-N}\mathbf{F}_{m-i}\right)^{-1} & ,\; p < N_1 \end{cases} \tag{16}$$

where $N_1 = -N+1$. The most general batch form of the p-shift UFIR estimator for TV models is thus

$$\hat{\mathbf{x}}_{n+p} = \mathbf{A}_{n,m}(p)\mathbf{Z}_{n,m}\,, \tag{17a}$$

$$= \mathcal{B}_{n,m}(p)\mathcal{H}_{n,m}^{-1}\mathbf{Z}_{n,m}\,, \tag{17b}$$

where $\mathbf{A}_{n,m}(p)$ is the UFIR estimator gain; and p can be arbitrary, $-\infty \leqslant p \leqslant \infty$. In the case of $-N+1 < p < 0$, one may also use a particular form of (17b) shown in [19], (21) with $\mathcal{B}_{n,m}(p) = \mathcal{F}_{n+p,0}^{m+1}$.

If we now observe that the filter estimate with $p = 0$ is

$$\hat{\mathbf{x}}_n = \mathcal{F}_{n,0}^{m+1}\mathcal{H}_{n,m}^{-1}\mathbf{Z}_{n,m}\,, \tag{18}$$

then (17b) can alternatively be written as

$$\hat{\mathbf{x}}_{n+p} = \mathcal{B}_{n,m}(p)(\mathcal{F}^{m+1}_{n,0})^{-1}\hat{\mathbf{x}}_n \,. \tag{19}$$

This suggests that prediction and smoothing can be organized based on the filtering estimate (18) if we use an auxiliary p-shift gain matrix. We will show below that (19) plays an important part in the design of UFIR algorithms.

1.2.2 Time-invariant models

In the special TI case, we have $\mathcal{B}_{n,m}(p) = \mathbf{F}^{n-m+p} = \mathbf{F}^{N-1+p}$ and the estimator becomes

$$\hat{\mathbf{x}}_{n+p} = \mathbf{A}(N,p)\mathbf{Z}_{n,m} \tag{20a}$$

$$= \mathbf{F}^{N-1+p}\bar{\mathcal{H}}^{-1}_{N-1}\mathbf{Z}_{n,m} \,, \tag{20b}$$

where $\bar{\mathcal{H}}^{-1}_{N-1} = (\bar{\mathbf{H}}^T_{N-1}\bar{\mathbf{H}}_{N-1})^{-1}\bar{\mathbf{H}}^T_{N-1}$ and

$$\bar{\mathbf{H}}_{N-1} = \hat{\mathbf{H}}_{N-1}\bar{\mathbf{F}}_{N-1} \,, \tag{21}$$

$$\bar{\mathbf{F}}_{N-1} = \big(\underbrace{\mathbf{F}^{N-1^T} \ \ldots \ \mathbf{F}^T \ \mathbf{I}}_{N}\big)^T \,, \tag{22}$$

$$\hat{\mathbf{H}}_{N-1} = \mathrm{diag}\big(\underbrace{\mathbf{H}\ \mathbf{H}\ \ldots\ \mathbf{H}}_{N}\big) \,. \tag{23}$$

Following (19), the estimate (20b) can alternatively be written as follows:

$$\hat{\mathbf{x}}_{n+p} = \mathbf{F}^p\hat{\mathbf{x}}_n \,, \tag{24}$$

where the TI filter estimate is given by

$$\hat{\mathbf{x}}_n = \mathbf{F}^{N-1}\bar{\mathcal{H}}^{-1}_{N-1}\mathbf{Z}_{n,m} \,. \tag{25}$$

A distinctive feature of both TV and TI batch UFIR estimators is that they can be applied to models with noise having arbitrary distributions and covariances. They can also be represented in fast iterative Kalman-like forms using an auxiliary matrix called the *generalized NPG* (GNPG), which will be discussed next.

1.2.3 Generalized noise power gain

It follows from (5) that the NPG is a measure of how much the measurement noise is suppressed at the FIR estimator output. In state space, the NPG is defined via the MSE [27], with the assumption that $\mathbf{B}_n = \mathbf{0}$:

$$\begin{aligned}\bar{\mathbf{P}}_{n+p} &= E\{(\mathbf{x}_{n+p} - \hat{\mathbf{x}}_{n+p})(\ldots)^T\}\\ &= E\{[\,\mathbf{x}_{n+p} - \mathbf{A}_{n,m}(p)\mathbf{Z}_{n,m}]\,[\,\ldots\,]^T\,\}\\ &= E\{[\,\mathbf{x}_{n+p} - \mathbf{A}_{n,m}(p)\mathbf{H}_{n,m}\mathbf{X}_{n,m}\\ &\qquad - \mathbf{A}_{n,m}(p)\mathbf{V}_{n,m}]\,[\,\ldots\,]^T\,\} \,,\end{aligned} \tag{26}$$

where

$$\mathbf{X}_{n,m} = \Big[\, \mathbf{x}^T_n \ \mathbf{x}^T_{n-1} \ \ldots \ \mathbf{x}^T_m \,\Big]^T \,, \tag{27}$$

$$\mathbf{V}_{n,m} = \Big[\, \mathbf{v}^T_n \ \mathbf{v}^T_{n-1} \ \ldots \ \mathbf{v}^T_m \,\Big]^T \,. \tag{28}$$

In view of the fact that the estimate is unbiased, two first-two terms in the brackets of (26) are zero by (2), which gives

$$\bar{\mathbf{P}}_{n+p} = \mathbf{A}_{n,m}E\{\mathbf{V}_{n,m}\mathbf{V}^T_{n,m}\}\mathbf{A}^T_{n,m} \,, \tag{29}$$

where $E\{\mathbf{V}_{n,m}\mathbf{V}^T_{n,m}\}$ is the measurement noise covariance on the averaging interval. A simplification follows instantly if one lets $p = 0$ and supposes that the model is one-state and time-invariant one. That leads to the estimation variance as follows:

$$\sigma^2_{\mathrm{est}} = \sigma^2_v\mathbf{A}(N)\mathbf{A}^T(N) \,, \tag{30}$$

where the product $\mathbf{A}(N)\mathbf{A}^T(N)$ is known as the NPG [11].

More generally, the GNPG can thus be written as

$$\mathbf{G}_{n,m}(p) = \mathbf{A}_{n,m}(p)\mathbf{A}^T_{n,m}(p) \tag{31}$$

to characterize the noise strength at the estimator output. In particular, if the GNPG is an identity matrix, then no noise reduction is provided by the estimator. If the GNPG has components that are equal to zero, then the noise is fully suppressed by the estimator.

By transforming (31) and utilizing (17b), (18), and (19), one can find two equivalent GNPG forms corresponding to TV models:

$$\mathbf{G}_{n,m}(p) = \mathcal{B}_{n,m}(p)(\mathbf{H}^T_{n,m}\mathbf{H}_{n,m})^{-1}\mathcal{B}^T_{n,m}(p) \,, \tag{32a}$$

$$= \mathcal{B}_{n,m}(p)(\mathcal{F}^{m+1}_{n,0})^{-1}\mathbf{G}_{n,m}(\mathcal{F}^{m+1}_{n,0})^{-T}\mathcal{B}^T_{n,m}(p) \,, \tag{32b}$$

where the GNPG $\mathbf{G}_{n,m} = \mathbf{G}_{n,m}(0)$ associated with filtering is given as follows:

$$\mathbf{G}_{n,m} = \mathcal{F}^{m+1}_{n,0}(\mathbf{H}^T_{n,m}\mathbf{H}_{n,m})^{-1}\mathcal{F}^{m+1^T}_{n,0} \,. \tag{33}$$

Similarly, using (20a), (24), and (25), the GNPG for TI models can be represented as follows:

$$\mathbf{G}(N,p) = \mathbf{F}^{N-1+p}(\mathbf{H}^T_{N-1}\mathbf{H}_{N-1})^{-1}\mathbf{F}^{N-1+p^T} \,, \tag{34a}$$

$$= \mathbf{F}^p\mathbf{G}(N)\mathbf{F}^{p^T} \,, \tag{34b}$$

where the GNPG $\mathbf{G}(N) = \mathbf{G}(N,0)$ for filtering is

$$\mathbf{G}(N) = \mathbf{F}^{N-1}(\mathbf{H}^T_{N-1}\mathbf{H}_{N-1})^{-1}\mathbf{F}^{N-1^T} \,. \tag{35}$$

Summarizing the generalizations provided for the batch UFIR estimator, we notice again that this estimator ignores noise statistics and initial errors in solving the problems of smoothing, filtering, and prediction in a unified scheme. Its important applied property is that the estimate becomes virtually optimal when $N \gg 1$ [20]. On the other hand, large N leads to computational problems owing to the large dimensions of the augmented matrices and vectors. For fast computation, iterative Kalman-like UFIR forms can be used, which will be discussed next.

1.3 Iterative Kalman-like UFIR estimation

Similar to the recursive OLS [28], the UFIR estimator can also be represented in a fast iterative form similar to the Kalman filter as shown in [19,20]. The iterative UFIR estimator requires that we start with initial values that are available from the batch algorithm, which typically requires matrix computations on the order of $K \times K$ dimensions, and then we iteratively update the estimator output. The state estimate is taken when an iterative variable reaches the current time n.

1.3.1 Time-varying models

For TV models, the estimates (17b) and (19) suggest two forms of iterative UFIR computation.

The direct form Following the derivations given in Appendices I and II of [20], the direct form of the iterative algorithm corresponding to (17b) is the following:

$$\hat{\mathbf{x}}_{l+p} = \mathbf{F}_{l+p}\hat{\mathbf{x}}_{l+p-1} + \mathbf{K}_l(\mathbf{z}_l - \mathbf{H}_l\mathcal{Y}_l\hat{\mathbf{x}}_{l+p-1})\,, \tag{36}$$

where $\mathcal{Y}_l \triangleq \mathcal{Y}_l(p) = \bar{\mathcal{Y}}_l(p)\mathbf{F}_{l+p}$ and

$$\bar{\mathcal{Y}}_l = \begin{cases} \mathcal{F}_{l,0}^{l+p+1} = \prod\limits_{i=0}^{|p|-1} \mathbf{F}_{l-i} & ,\ p < 0 \\ \mathbf{I} & ,\ p = 0\,. \\ \left(\mathcal{F}_{l+p,0}^{l-1}\right)^{-1} = \left(\prod\limits_{i=0}^{p-1} \mathbf{F}_{l+p-i}\right)^{-1} & ,\ p > 0 \end{cases} \tag{37}$$

The bias correction gain $\mathbf{K}_l \triangleq \mathbf{K}_{l,m}(p)$ is given here as

$$\mathbf{K}_l = \mathbf{G}_l\bar{\mathcal{Y}}_l^T\mathbf{H}_l^T\,, \tag{38}$$

where the GNPG $\mathbf{G}_l \triangleq \mathbf{G}_{l,m}(p)$ is computed iteratively by

$$\mathbf{G}_l = \mathbf{F}_{l+p}(\mathcal{Y}_l^T\mathbf{H}_l^T\mathbf{H}_l\mathcal{Y}_l + \mathbf{G}_{l-1}^{-1})^{-1}\mathbf{F}_{l+p}^T\,, \tag{39a}$$

$$=[\,\bar{\mathcal{Y}}_l^T\mathbf{H}_l^T\mathbf{H}_l\bar{\mathcal{Y}}_l + (\mathbf{F}_{l+p}\mathbf{G}_{l-1}\mathbf{F}_{l+p}^T)^{-1}]^{-1}\,, \tag{39b}$$

where $\bar{\mathcal{Y}}_l = \mathcal{Y}_l\mathbf{F}_{l+p}^{-1}$. The initial values, $\hat{\mathbf{x}}_{s+p}$ and $\mathbf{G}_s$, are computed in short batch forms as

$$\hat{\mathbf{x}}_{s+p} = \mathcal{B}_{s,m}(p)\mathcal{H}_{s,m}^{-1}\mathbf{Z}_{s,m}\,, \tag{40}$$

$$\mathbf{G}_s = \mathcal{B}_{s,m}(p)(\mathbf{H}_{s,m}^T\mathbf{H}_{s,m})^{-1}\mathcal{B}_{s,m}^T(p)\,, \tag{41}$$

where $s = m + K - 1$; and the iterative variable l ranges from $m + K$ to n. The estimator output is taken when $l = n$. Since the UFIR estimate does not require initial conditions, one may approximately set (40) to zero and let (41) be the identity when $N \gg 1$. However, this simplification may not always lead to good estimates for smaller values of N.

The estimate at time $n + p$ appears in (36) from an iterative update beginning with time step $m + K + p - 1$. Its flaw is that $|p|$ past points before the N-point estimator window are formally required for smoothing. This disadvantage is overcome in the two-stage form, which will be discussed next.

The two-stage form The batch estimate (19) suggests that another iterative UFIR form can be found if we first set $p = 0$ in (36) and find the filter estimate:

$$\hat{\mathbf{x}}_l = \mathbf{F}_l\hat{\mathbf{x}}_{l-1} + \mathbf{K}_l(\mathbf{z}_l - \mathbf{H}_l\mathbf{F}_l\hat{\mathbf{x}}_{l-1})\,, \tag{42}$$

in which

$$\mathbf{K}_l = \mathbf{G}_l\mathbf{H}_l^T\,, \tag{43}$$

$$\mathbf{G}_l =[\,\mathbf{H}_l^T\mathbf{H}_l + (\mathbf{F}_l\mathbf{G}_{l-1}\mathbf{F}_l^T)^{-1}]^{-1}\,, \tag{44}$$

and the initial values are given as follows:

$$\hat{\mathbf{x}}_s = \mathcal{F}_{s,0}^{m+1}\mathcal{H}_{s,m}^{-1}\mathbf{Z}_{s,m}\,, \tag{45}$$

$$\mathbf{G}_s = \mathcal{F}_{s,0}^{m+1}(\mathbf{H}_{s,m}^T\mathbf{H}_{s,m})^{-1}\mathcal{F}_{s,0}^{m+1\,T}\,. \tag{46}$$

Here, $s = m + K - 1$ and l ranges from $m + K$ to n, as before.

Given $\hat{\mathbf{x}}_n$ from (42) with $l = n$, the p-shift estimate can then be computed utilizing (19) as

$$\hat{\mathbf{x}}_{n+p} = \mathcal{B}_{n,m}(p)(\mathcal{F}_{n,0}^{m+1})^{-1}\hat{\mathbf{x}}_n\,. \tag{47}$$

As can be seen, this second form available does not require extra data points before the filtering window. However, it requires two computational steps, unlike the direct form (36).

1.3.2 Time-invariant models

Employing (20b) and (24), the p-shift estimate for TI models can also be found in two equivalent iterative forms.

The direct form If all of the model matrices are TI, we have $\mathcal{Y}_l = \mathbf{F}^{1-p}$. Accordingly, (36) becomes

$$\hat{\mathbf{x}}_{l+p} = \mathbf{F}\hat{\mathbf{x}}_{l+p-1} + \mathbf{K}_l(\mathbf{z}_l - \mathbf{H}\mathbf{F}^{1-p}\hat{\mathbf{x}}_{l+p-1}) \tag{48}$$

and the bias correction gain (38) attains the form

$$\mathbf{K}_l = \mathbf{G}_l\mathbf{F}^{-p^T}\mathbf{H}^T\,, \tag{49}$$

where $\mathbf{G}_l$ is computed iteratively as

$$\mathbf{G}_l =[\,\mathbf{F}^{-p^T}\mathbf{H}^T\mathbf{H}\mathbf{F}^{-p} + (\mathbf{F}\mathbf{G}_{l-1}\mathbf{F}^T)^{-1}]^{-1}\,. \tag{50}$$

The initial values are computed as

$$\hat{\mathbf{x}}_{s+p} = \mathbf{F}^{s-m+p}\bar{\mathcal{H}}_{s-m}^{-1}\mathbf{Z}_{s,m}\,, \tag{51}$$

$$\mathbf{G}_s = \mathbf{F}^{s-m+p}(\mathbf{H}_{s-m}^T\mathbf{H}_{s-m})^{-1}\mathbf{F}^{s-m+p^T}\,, \tag{52}$$

where $s = m + K - 1$ and l ranges from $m + K$ to n. The desired state estimate is taken at $l = n$.

The two-stage form Provided the TI filtering estimate

$$\hat{\mathbf{x}}_l = \mathbf{F}\hat{\mathbf{x}}_{l-1} + \mathbf{K}_l(\mathbf{z}_l - \mathbf{H}\mathbf{F}\hat{\mathbf{x}}_{l-1}) \tag{53}$$

with

$$\mathbf{K}_l = \mathbf{G}_l\mathbf{H}^T\,, \tag{54}$$

$$\mathbf{G}_l =[\,\mathbf{H}_l^T\mathbf{H}_l + (\mathbf{F}_l\mathbf{G}_{l-1}\mathbf{F}_l^T)^{-1}]^{-1}\,, \tag{55}$$

and the initial conditions

$$\hat{\mathbf{x}}_s = \mathbf{F}^{s-m}\mathcal{H}_{s,m}^{-1}\mathbf{Z}_{s,m}\,, \tag{56}$$

$$\mathbf{G}_s = \mathbf{F}^{s-m}(\mathbf{H}_{s,m}^T\mathbf{H}_{s,m})^{-1}\mathbf{F}^{s-m^T}\,, \tag{57}$$

where $s = m + K - 1$ and the iterative variable l ranges from $m + K$ to n, the p-shift estimate can alternatively be computed by (24) as follows:

$$\hat{\mathbf{x}}_{n+p} = \mathbf{F}^p\hat{\mathbf{x}}_n\,. \tag{58}$$

One may conclude that the algorithm of (53) and (58) is very simple from a programming perspective. As was shown in [8,19,20] and in many other studies, the UFIR estimator is a strong rival to the Kalman filter if the noise covariances are not known exactly.

1.4 Estimation errors

Next, we discuss errors in UFIR estimators assuming white Gaussian noise. Given the instantaneous error

$$\epsilon_{n+p} = \mathbf{x}_{n+p} - \hat{\mathbf{x}}_{n+p}\,, \tag{59}$$

where the estimate $\hat{\mathbf{x}}_{n+p}$ comes from either a TV or TI model, the MSE $\mathbf{P}_{n+p}$ at time $n + p$ can be defined as

$$\mathbf{P}_{n+p} = E\{\epsilon_{n+p}\epsilon_{n+p}^T\}\,. \tag{60}$$

In spite of the fact that the UFIR estimator has two equivalent forms (batch and iterative), the MSE can rigorously be determined only via the batch form. Finding closed analytical solutions for the MSE via (19) and (25) implies a large mathematical burden and is still a challenging problem. On the other hand, a rigorous error computation may be unnecessary since estimation error covariances are not used in the UFIR algorithms, and so reasonable approximations can serve us well in practical applications. Such an approximation provided following [23] is given in the Appendix.

The MSE upper bound (UB) $\mathbf{P}_{n+p}^{\text{UB}}$ can be obtained from an iterative computation of (114) for the general TV model. Equation (114) implies that process noise covariances are accumulated at each iteration. Therefore, the predicted value from (114) is a bit larger than the actual estimation error covariance for small N and significantly larger for $N \gg 1$. For the same reason, the estimate of (114) also diverges as p increases. The UB can thus be very useful for filtering ($p = 0$) when N is not large and for smoothing with small lags. In the case of prediction, the future noise is neglected in (114) so it can serve as a tight upper bound even for very large p.

The MSE lower bound (LB) can be found if we take into consideration the fact that if $N \leqslant N_{\text{opt}}$ the UFIR estimator order fits the system order. Therefore the system noise can be neglected in (114) and the LB $\mathbf{P}_{n+p}^{\text{LB}}$ can be found by iterating

$$\mathbf{P}_{l+p}^{\text{LB}} = (\mathbf{I} - \mathbf{K}_l\mathbf{H}_l\bar{\mathcal{Y}}_l)\mathbf{F}_{l+p}\mathbf{P}_{l+p-1}^{\text{LB}}\mathbf{F}_{l+p}^T(\ldots)^T + \mathbf{K}_l\mathbf{R}_l\mathbf{K}_l^T \tag{61}$$

until l reaches n. For TI models, (61) becomes

$$\mathbf{P}_{l+p}^{\text{LB}} = (\mathbf{I} - \mathbf{K}_l\mathbf{H}\mathbf{F}^{-p})\mathbf{F}\mathbf{P}_{l+p-1}^{\text{LB}}\mathbf{F}^T(\ldots)^T + \mathbf{K}_l\mathbf{R}_l\mathbf{K}_l^T\,. \tag{62}$$

Equations (61) and (62) correspond to the direct estimator forms of (36) and (48) respectively.

If one employs the two-stage forms of (47) and (58) and the MSE LB for filtering ($p = 0$), this is defined using (61) as follows:

$$\mathbf{P}_l^{\text{LB}} = (\mathbf{I} - \mathbf{K}_l\mathbf{H}_l)\mathbf{F}_l\mathbf{P}_{l-1}^{\text{LB}}\mathbf{F}_l^T(\ldots)^T + \mathbf{K}_l\mathbf{R}_l\mathbf{K}_l^T\,, \tag{63}$$

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

$$\mathbf{P}_{n+p}^{\text{LB}} = \mathcal{B}_{n,m}(p)\mathcal{F}_{n,0}^{m+1^{-1}}\mathbf{P}_n^{\text{LB}}\mathcal{F}_{n,0}^{m+1^{-T}}\mathcal{B}_{n,m}^T(p)\,, \tag{64}$$

$$\mathbf{P}_{n+p}^{\text{LB}} = \mathbf{F}^p\mathbf{P}_n^{\text{LB}}\mathbf{F}^{p^T}\,, \tag{65}$$

where $\mathbf{P}_n^{\text{LB}}$ is provided from (63) with $l = n$. Note that the LB is associated with the NPG and serves well in the three-sigma sense [27].

1.5 Filtering

Filtering is used when the state estimate is required at the current time point n. It can also be projected to the future (prediction), or smoothed by combining several filtering estimates from the past. By letting $p = 0$ in (36), the UFIR filtering estimate becomes

$$\hat{\mathbf{x}}_l = \mathbf{F}_l\hat{\mathbf{x}}_{l-1} + \mathbf{K}_l(\mathbf{z}_l - \mathbf{H}_l\mathbf{F}_l\hat{\mathbf{x}}_{l-1})\,, \tag{66}$$

where the bias correction gain is

$$\mathbf{K}_l = \mathbf{G}_l\mathbf{H}_l^T \tag{67}$$

and the GNPG $\mathbf{G}_l$ is computed iteratively by

$$\mathbf{G}_l = \left[\mathbf{H}_l^T\mathbf{H}_l + (\mathbf{F}_l\mathbf{G}_{l-1}\mathbf{F}_l^T)^{-1}\right]^{-1}\,. \tag{68}$$

The initial values for (66) and (67) are respectively specified in short batch forms as follows:

$$\hat{\mathbf{x}}_s = \mathcal{F}_{s,0}^{m+1}\mathcal{H}_{s,m}^{-1}\mathbf{Z}_{s,m}\,, \tag{69}$$

$$\mathbf{G}_s = \mathcal{F}_{s,0}^{m+1}(\mathbf{H}_{s,m}^T\mathbf{H}_{s,m})^{-1}\mathcal{F}_{s,0}^{m+1^T}\,, \tag{70}$$

where $s = m + K - 1$, $\mathcal{F}_{s,0}^{m+1} = \prod_{i=0}^{K-2}\mathbf{F}_{s-i}$, the iteration index l ranges from $m + K$ to n, and the estimate of the current state is taken when $l = n$.

The MSE UB can be found for (66) by setting $p = 0$ in (114), which gives

$$\mathbf{P}_l^{\text{UB}} = (\mathbf{I} - \mathbf{K}_l\mathbf{H}_l)\mathbf{P}_l^-(\mathbf{I} - \mathbf{K}_l\mathbf{H}_l)^T + \mathbf{K}_l\mathbf{R}_l\mathbf{K}_l^T\,, \tag{71}$$

where the predicted (*a priori*) estimate covariance $\mathbf{P}_l^- \triangleq \mathbf{P}_{l|l-1}$ is given by

$$\mathbf{P}_l^- = \mathbf{F}_l\mathbf{P}_{l-1}^{\mathrm{UB}}\mathbf{F}_l^T + \mathbf{B}_l\mathbf{Q}_l\mathbf{B}_l^T \qquad (72)$$

with the initial value $\mathbf{P}_{l-1}^{\mathrm{LB}}$ specified as for the Kalman filter.

The LB appears from (72) by neglecting $\mathbf{Q}_l$, which gives

$$\mathbf{P}_l^{\mathrm{LB}} = (\mathbf{I}-\mathbf{K}_l\mathbf{H}_l)\mathbf{F}_l\mathbf{P}_{l-1}^{\mathrm{LB}}\mathbf{F}_l^T(\mathbf{I}-\mathbf{K}_l\mathbf{H}_l)^T+\mathbf{K}_l\mathbf{R}_l\mathbf{K}_l^T\,, \qquad (73)$$

where the initial value $\mathbf{P}_{l-1}^{\mathrm{LB}}$ can also be specified as in the Kalman filter.

It can easily be shown that (71) is the Kalman *a posteriori* estimate covariance, if we substitute $\mathbf{K}_l$ with the Kalman gain. However, unlike the Kalman filter, (66) can be applied to deterministic models. If that is the case ($\mathbf{R}_l = \mathbf{0}$ and $\mathbf{Q}_l = \mathbf{0}$), then the estimation error is zero.

Several particular filtering solutions can now be discussed, which will be done in the following sections.

1.5.1 Fixed-horizon filtering

The fixed-horizon (fixed memory size N) iterative UFIR filtering algorithm is summarized for TV models in Table 1.

It is implied that measurements are available beginning at time index 0, and the time index n starts at $N-1$. The initial values $\hat{\mathbf{x}}_s$ and $\mathbf{G}_s$ are computed using (69) and (70), respectively. For each n, the iterative variable l increments from $m+K$ to n, and the desired estimate is taken when $l = n$. Note that the estimation error computed by (71) is minimal if one sets $N = N_{\mathrm{opt}}$. A simplification for the TI model is straightforward. One must just let all of the matrices be TI in Table 1.

1.5.2 Full-horizon filtering

Full-horizon filtering can be applied to highly stable or highly predictable systems such as those in astronomy, precise clocks and oscillators [27], and others associated with near deterministic state-space models. The full-horizon TV algorithm is given in Table 2.

This algorithm is the most simple. It requires only the number of the states K since the filter memory window size changes with time; so, $N = n+1$. No additional information is needed, and the algorithm thus has extremely desirable engineering features. A natural extension of the TV algorithm (Table 2) to the TI case is provided by removing the time dependencies from the matrices.

The MSE UB and LB can be computed by (71) and (73) if we substitute l with n. Note that the full-horizon UFIR filter may demonstrate substantial decrease in the output noise as n becomes large.

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 (69) and $\mathbf{G}_s$ by (70) .
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 (69) and $\mathbf{G}_{K-1}$ by (70) for $m = 0$.
Update:	$\mathbf{G}_n = [\mathbf{H}_n^T\mathbf{H}_n + (\mathbf{F}_n\mathbf{G}_{n-1}\mathbf{F}_n^T)^{-1}]^{-1}\,,$
	$\hat{\mathbf{x}}_n = \mathbf{F}_n\hat{\mathbf{x}}_{n-1} + \mathbf{G}_n\mathbf{H}_n^T(\mathbf{z}_n - \mathbf{H}_n\mathbf{F}_n\hat{\mathbf{x}}_{n-1})\,.$

1.5.3 Tricky-horizon filtering

The tricky-horizon (time-variant memory size N) algorithm can be used in adaptive filtering [29,30] and whenever some reference information about the process behavior is available. It implies an individual N_{opt} at each time index n. Such flexibility allows better system tracking with minimum residuals [19]. To implement tricky-horizon filtering, the algorithm (Table 1) can be used if we allow N to be variable.

1.6 Smoothing

Smoothing is commonly associated with a lag $q > 0$ relating the estimate at a given time index to measurements up to and including some past index. By combining 'future' and past estimates, it becomes possible to obtain better noise reduction for many practical applications. Note that an infinity of smoother solutions exists [31]. We will discuss two basic schemes for UFIR smoothers in this section.

The direct form By letting $q = -p > 0$ in (36), the smoothing estimate at $n - q$ can be found iteratively as follows [23]:

$$\hat{\mathbf{x}}_{l-q} = \mathbf{F}_{l-q}\hat{\mathbf{x}}_{l-q-1} + \mathbf{K}_l\left(\mathbf{z}_l - \mathbf{H}_l\mathcal{Y}_l\hat{\mathbf{x}}_{l-q-1}\right)\,, \qquad (74)$$

where $\mathcal{Y}_l = \mathcal{F}_{l,0}^{l-q} = \prod_{i=0}^{q}\mathbf{F}_{l-i}$ and l ranges from $m+K$ to n. The estimate $\hat{\mathbf{x}}_{n-q}$ is traditionally taken at $l = n$ in each iterative cycle.

The bias correction gain can be computed here using the following:

$$\mathbf{K}_l = \bar{\mathcal{Y}}_l^{-1}\mathbf{G}_l\mathbf{H}_l^T\,,$$

where $\bar{\mathcal{Y}}_l = \mathcal{Y}_l \mathbf{F}_{l-q}^{-1}$ and $\mathbf{G}_l$ is given by (39b). The initial values $\hat{\mathbf{x}}_s$ and $\mathbf{G}_l$ can be defined at $s = m + K - 1$ as, respectively,

$$\hat{\mathbf{x}}_s = \bar{\mathcal{B}}_{s,m}(q)\mathcal{H}_{s,m}^{-1}\mathbf{Z}_{s,m}\,, \tag{75}$$

$$\mathbf{G}_s = \bar{\mathcal{B}}_{s,m}(q)(\mathcal{H}_{s,m}^T\mathbf{H}_{s,m})^{-1}\bar{\mathcal{B}}_{s,m}^T(q\,, \tag{76}$$

where $\bar{\mathcal{B}}_{s,m}(q) \triangleq \bar{\mathcal{B}}_{s,m}(K,q)$ is given by

$$\bar{\mathcal{B}}_{n,m}(N,q) = \begin{cases} \prod\limits_{i=0}^{N-2-q} \mathbf{F}_{n-q-i}\,, & q < N-1\,, \\ \mathbf{I}\,, & q = N-1\,, \\ \left(\prod\limits_{i=1}^{q-N} \mathbf{F}_{m-i}\right)^{-1}, & q > N-1\,. \end{cases} \tag{77}$$

Following (114), the MSE UB for (74) can be found to be

$$\mathbf{P}_{l-q}^{\mathrm{UB}} = (\mathbf{I} - \mathbf{K}_l\mathbf{H}_l\bar{\mathcal{Y}}_l)\mathbf{P}_{l-q}^{-}(\ldots)^T + \mathbf{K}_l\mathbf{R}_l\mathbf{K}_l^T + \mathbf{K}_l\mathbf{H}_l\tilde{\mathcal{Q}}_l\mathbf{H}_l^T\mathbf{K}_l^T \tag{78}$$

where $\mathbf{P}_{l-q}^{-}$ is given by (72) and

$$\tilde{\mathcal{Q}}_l = \breve{\mathcal{Q}}_l + \mathbf{B}_l\mathbf{Q}_l\mathbf{B}_l^T\,, \tag{79}$$

$$\breve{\mathcal{Q}}_l = \prod_{i=2}^{q} \mathcal{Y}_l(q-i)\mathbf{B}_{l-q-1+i}\mathbf{Q}_{l-q-1+i}\mathbf{B}_{l-q-1+i}^T\mathcal{Y}_l^T(q-i)\,. \tag{80}$$

The MSE LB is obtained by neglecting $\mathbf{Q}_n$ as

$$\mathbf{P}_{l-q}^{\mathrm{LB}} = (\mathbf{I} - \mathbf{K}_l\mathbf{H}_l\bar{\mathcal{Y}}_l)\mathbf{F}_{l-q-1}\mathbf{P}_{l-q-1}^{\mathrm{LB}}\mathbf{F}_{l-q-1}^T(\ldots)^T + \mathbf{K}_l\mathbf{R}_l\mathbf{K}_l^T\,. \tag{81}$$

The two-stage form Provided the filtering estimate (66), the second form for the TV and TI UFIR smoothers become by (19) and (24) respectively

$$\hat{\mathbf{x}}_{n-q} = \bar{\mathcal{B}}_{n,m}(q)(\mathcal{F}_{n,0}^{m+1})^{-1}\hat{\mathbf{x}}_n\,, \tag{82}$$

$$\hat{\mathbf{x}}_{n-q} = \mathbf{F}^{-q}\hat{\mathbf{x}}_n\,, \tag{83}$$

where $\bar{\mathcal{B}}_{n,m}(q) \triangleq \bar{\mathcal{B}}_{n,m}(N,q)$ is given by (77).

The relevant estimation error covariance LBs become, respectively,

$$\mathbf{P}_{n-q}^{\mathrm{LB}} = \bar{\mathcal{B}}_{n,m}(q)\mathcal{F}_{n,0}^{m+1^{-1}}\mathbf{P}_n^{\mathrm{LB}}\mathcal{F}_{n,0}^{m+1^{-T}}\bar{\mathcal{B}}_{n,m}^T(q)\,, \tag{84}$$

$$\mathbf{P}_{n-q}^{\mathrm{LB}} = \mathbf{F}^{-q}\mathbf{P}_n^{\mathrm{LB}}\mathbf{F}^{-q^T}\,, \tag{85}$$

where $\mathbf{P}_n^{\mathrm{LB}}$ is provided by (63) at $l = n$. As in filtering, here, the LB can serve well in the three-sigma sense [27].

1.6.1 Fixed-interval smoothing

Among various smoothing problems, the fixed-interval one is basic and often associated with smoothing [25,32-34]. The fixed-interval UFIR smoother is intended to provide an estimate $\hat{\mathbf{x}}_{n-q|n}$ with any lag $0 < q < M$ utilizing measurement on a fixed interval of M points, from time index $n - M + 1$ to n. Although M may not be equal to N_{opt}, UFIR smoothing is most efficient when $M = N_{\mathrm{opt}}$. In fact, If $M > N_{\mathrm{opt}}$, smoothing is inefficient when $N_{\mathrm{opt}} < q < M$, because q exceeds the length of the averaging interval and smoothing virtually provides the backward prediction, which has an estimation error larger than in filtering. On the other hand, N_{opt} should not be larger than M, because M is commonly associated with an available database.

Provided $M = N_{\mathrm{opt}}$, two traditional forms can be suggested for fixed-interval UFIR smoothing.

The direct form This form implements (74) as listed in Table 3. A special peculiarity is that n starts at $N-1+q$ in order for the smoother to process only nonnegative time indices. For TI models, the modification of this algorithm is straightforward.

The two-stage form The two-stage form implementing (82) can be used as shown in Table 4. To apply this algorithm to TI models, one must compute $\hat{\mathbf{x}}_{n-q} = \mathbf{F}^{-q}\hat{\mathbf{x}}_n$. A common feature of this algorithm is that two stages are required: first filtering must be provided to get the estimate at time n, then the obtained filter estimate is projected to time $n - q$.

1.6.2 Fixed-lag smoothing

Fixed-lag smoothing is commonly used for denoising if a time delay of q points is allowed [31,32,35,36]. Two basic fixed-lag algorithms can be designed based on the UFIR technique.

Fixed-lag OUFIR smoothing Provided N_{opt}, the fixed lag q can be specified based on the process properties to obtain the best denoising. Intuition indicates that smoothing is best if the estimation time is the center of the observation interval. This holds true if the polynomial describing the process

Table 3 Direct fixed-interval TV OUFIR smoothing algorithm

Stage	
Given:	$K, N =$ constant, $q, m = n - N + 1 \geqslant 0$, $s = m + K - 1, m + K \leqslant l \leqslant n\,.$
Set:	$\hat{\mathbf{x}}_s$ by (75) and $\mathbf{G}_s$ by (76) .
Update:	$\mathbf{G}_l = [\mathbf{H}_l^T\mathbf{H}_l + (\mathbf{F}_l\mathbf{G}_{l-1}\mathbf{F}_l^T)^{-1}]^{-1}\,,$ $\mathbf{K}_l = \left(\prod_{i=0}^{q-1}\mathbf{F}_{l-i}\right)^{-1}\mathbf{G}_l\mathbf{H}_l^T\,,$ $\hat{\mathbf{x}}_{l-q} = \mathbf{F}_{l-q}\hat{\mathbf{x}}_{l-q-1} + \mathbf{K}_l\big(\mathbf{z}_l - \mathbf{H}_l\prod_{i=0}^{q}\mathbf{F}_{l-i}\hat{\mathbf{x}}_{l-q-1}\big)\,.$
Instruction:	The algorithm is valid for any $n \geqslant N - 1 + q$. Use the smoothed estimate when $l = n$. The fixed interval of $M = N = N_{\mathrm{opt}}$ points is from time index m to n.

Table 4 Two-stage fixed-interval TV OUFIR smoothing algorithm

Stage	
Given:	$K, N =$ constant, $q, m = n - N + 1 \geqslant 0$, $s = m + K - 1, m + K \leqslant l \leqslant n$.
Set:	$\hat{\mathbf{x}}_s$ by (75) and $\mathbf{G}_s$ by (76) .
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 $\hat{\mathbf{x}}_{n-q} = \bar{\boldsymbol{\mathcal{B}}}_{n,m}(q)(\boldsymbol{\mathcal{F}}_{n,0}^{m+1})^{-1}\hat{\mathbf{x}}_n$.
Instruction:	This algorithm is valid for any $n \geqslant N - 1$. The fixed interval of $M = N = N_{\text{opt}}$ points is from time index m to n.

behavior on the observation interval is of odd degree. Otherwise, if the degree is even, denoising may be better with shorter lags as shown in Figure eight in [17]. The fixed-lag OUFIR smoothing algorithm is listed in Table 4 if one sets $N = N_{\text{opt}}$ and $q =$ constant. Its extension to the TI case can be provided by replacing the $\hat{\mathbf{x}}_{n-q}$ equation in Table 4 with $\hat{\mathbf{x}}_{n-q} = \mathbf{F}^{-q}\hat{\mathbf{x}}_n$.

Fixed-lag full-horizon UFIR smoothing This approach implies that the filter window includes all the available data, but the lag is fixed. Examples can be found in highly predictable or quasi deterministic slow dynamics, for which the estimates at time n and $n - q$ do not significantly vary from each other in terms of bias. The relevant algorithm for TV models is listed in Table 5. Its extension to the TI case can be obtained by replacing the $\hat{\mathbf{x}}_{n-q}$ equation in Table 5 with $\hat{\mathbf{x}}_{n-q}$ as $\hat{\mathbf{x}}_{n-q} = \mathbf{F}^{-q}\hat{\mathbf{x}}_n$.

1.6.3 Fixed-point smoothing

Fixed-point smoothing implies that measurements are available from time index 0 up to the current time point n, but the estimate is required at some fixed past point $0 \leqslant \nu < n$, where ν is a constant [32,37]. The time-varying lag is thus $q = n - \nu$. In such a formulation, the UFIR smoother is always full-horizon as shown by the TV algorithm in Table 6. 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. Probably the most interesting application for such algorithms is initial state estimation with $\nu = 0$. Note that the same problem can be solved using the fixed-interval smoother (Table 4) if we set the initial interval point to zero.

1.7 Prediction

State prediction plays a key role in many applications. The one-step predictor is fundamental for digital feedback control systems [38]. It is also commonly provided when measurements are unavailable at some points [39] and when estimates of long-term future behavior are required [40]. Predictive estimation is necessary for global positioning system (GPS)-based applications when the GPS receiver temporarily fails or when a signal is temporarily unavailable [27]. Predictive estimation is required for holdover in digital communication networks [41], for maintaining normal functioning of certain systems during down time [42,43], and for astronomy and climate forecasting. The predictor goal is thus to compensate for unavailable information. In many cases, linear predictors do perform better than nonlinear ones [44].

To develop UFIR prediction, two algorithms [27] can be used as shown in Figure 3. It is supposed that measurements at each point are either available (○) or not (×). Utilizing N_{opt} available points from the immediate past, the estimator provides a one-step ahead projection (●) from each point of this interval: from point 1 to 2a, from 2 to 3a, etc. At point 4, the measurement is unavailable. Therefore, the predicted value 4a is utilized at point 4. Further predictor equations can be organized either with *fixed steps* or *variable steps* in the direct and two-stage forms. It has been shown in [27] that the variable-step approach is more precise in the short term, and that there is not a large difference between the estimates in the long term.

Table 5 Fixed-lag full-horizon TV UFIR smoothing algorithm

Stage	
Given:	$K, q =$ constant, $n \geqslant K$.
Set:	$\hat{\mathbf{x}}_{K-1}$ by (75) and $\mathbf{G}_{K-1}$ by (76) 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 $\hat{\mathbf{x}}_{n-q} = \bar{\boldsymbol{\mathcal{B}}}_{n,m}(q)(\boldsymbol{\mathcal{F}}_{n,0}^{m+1})^{-1}\hat{\mathbf{x}}_n$.

Table 6 Fixed-point TV UFIR smoothing algorithm

Stage	
Given:	$K, \nu =$ constant $\geqslant 0, q = n - \nu, n \geqslant K$.
Set:	$\hat{\mathbf{x}}_{K-1}$ by (75) and $\mathbf{G}_{K-1}$ by (76) 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 $n > \nu$ as follows: $\hat{\mathbf{x}}_{n-q} = \bar{\boldsymbol{\mathcal{B}}}_{n,m}(q)(\boldsymbol{\mathcal{F}}_{n,0}^{m+1})^{-1}\hat{\mathbf{x}}_n$.

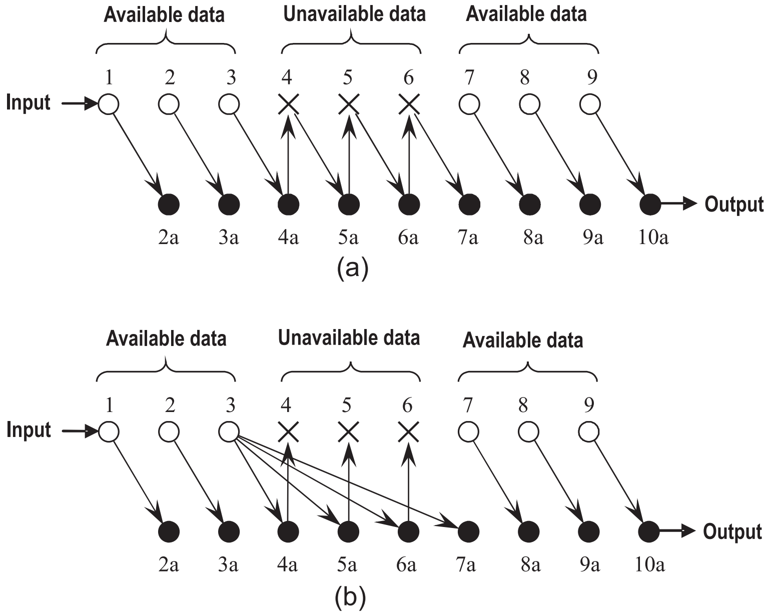

Figure 3 Basic UFIR prediction algorithms: (a) fixed-step and (b) variable-step, after [27].

1.7.1 Fixed-step prediction

In the fixed-step case shown in Figure 3a, p is often unity, but in general may be arbitrary ($p > 0$). With $p = 1$, prediction can permanently substitute for unavailable measurements with predicted values.

The direct form The one-step predictor appears from (36) by setting $p = 1$:

$$\hat{\mathbf{x}}_{l+1} = \mathbf{F}_{l+1}\hat{\mathbf{x}}_l + \mathbf{G}_l\mathbf{F}_{l+1}^{-T}\mathbf{H}_l^T(\mathbf{z}_l - \mathbf{H}_l\hat{\mathbf{x}}_l)\,, \tag{86}$$

where $\mathbf{G}_l$ is computed iteratively by

$$\mathbf{G}_l = \mathbf{F}_{l+1}(\mathbf{H}_l^T\mathbf{H}_l + \mathbf{G}_{l-1}^{-1})^{-1}\mathbf{F}_{l+1}^T \tag{87}$$

and the initial values are determined as

$$\hat{\mathbf{x}}_{s+1} = \mathcal{F}_{s+1,0}^{m+1}\mathcal{H}_{s,m}^{-1}\mathbf{Z}_{s,m}\,, \tag{88}$$

$$\mathbf{G}_s = \mathcal{F}_{s+1,0}^{m+1}(\mathbf{H}_{s,m}^T\mathbf{H}_{s,m})^{-1}\mathcal{F}_{s+1,0}^{m+1\;T}\,, \tag{89}$$

where $\mathcal{F}_{s+1,0}^{m+1} = \prod_{i=0}^{K-1}\mathbf{F}_{s+1-i}$, $s = m + K - 1$, and l ranges from $m + K$ to n. The desired estimate is obtained when $l = n$.

For TI models, the one-step predictor becomes:

$$\hat{\mathbf{x}}_{l+1} = \mathbf{F}\hat{\mathbf{x}}_l + \mathbf{G}_l\mathbf{F}^{-T}\mathbf{H}^T(\mathbf{z}_l - \mathbf{H}\hat{\mathbf{x}}_l)\,, \tag{90}$$

$$\mathbf{G}_l = \mathbf{F}(\mathbf{H}^T\mathbf{H} + \mathbf{G}_{l-1}^{-1})^{-1}\mathbf{F}^T\,, \tag{91}$$

$$\hat{\mathbf{x}}_{s+1} = \mathbf{F}^K\mathcal{H}_{s,m}^{-1}\mathbf{Z}_{s,m}\,, \tag{92}$$

$$\mathbf{G}_s = \mathbf{F}^K(\mathbf{H}_{s,m}^T\mathbf{H}_{s,m})^{-1}\mathbf{F}^{K^T}\,. \tag{93}$$

Both predictors can be implemented in the algorithm (Figure 3a) to satisfy the following condition: if $\mathbf{z}_n$ is unavailable at time n, then set $\mathbf{z}_n = \mathbf{H}_n\hat{\mathbf{x}}_n$ for a TV model and $\mathbf{z}_n = \mathbf{H}\hat{\mathbf{x}}_n$ for a TI one.

The two-stage form By (47) and (58), the second form of the one-step predictor for TV and TI models, respectively, are the following:

$$\hat{\mathbf{x}}_{n+1} = \mathbf{F}_{n+1}\hat{\mathbf{x}}_n\,, \tag{94}$$

$$\hat{\mathbf{x}}_{n+1} = \mathbf{F}\hat{\mathbf{x}}_n\,, \tag{95}$$

where $\hat{\mathbf{x}}_n$ is the filter estimate. This is the most widely used prediction scheme.

1.7.2 Variable-step prediction

In the variable-step case illustrated in Figure 3b, the predicted estimates still compensate for unavailable measurements (points 4, 5, 6), but they are not involved to produce predictions, which is unlike the case of Figure 3a. Instead, p continues to increment until the measurement becomes available. At point 7, all measured and predicted values on a horizon of N_{opt} past points are used to produce a prediction at point 8a. There are no other differences between fixed-step and variable-step prediction, and the estimates (36), (47), (48), and (58) can be used in a straightforward manner, along with the relevant error bounds.

1.8 Nonlinear models and extended filtering

For many applications, nonlinear systems can be modeled in additive white Gaussian noise environments with the state and observation equations as follows:

$$\mathbf{x}_n = \mathbf{f}_n(\mathbf{x}_{n-1}) + \mathbf{B}_n\mathbf{w}_n\,, \tag{96}$$

$$\mathbf{z}_n = \mathbf{h}_n(\mathbf{x}_n) + \mathbf{v}_n\,, \tag{97}$$

where $\mathbf{f}_n(\mathbf{x}_{n-1})$ and $\mathbf{h}_n(\mathbf{x}_n)$ are time-varying nonlinear vector functions and all other notations are given in (6) and (7). If $\mathbf{f}_n(\mathbf{x}_{n-1})$ and $\mathbf{h}_n(\mathbf{x}_n)$ are smooth enough, then the first-order Taylor series expansion is often applied to make the model approximately linear between two neighboring points.

An expansion of $\mathbf{f}_n(\mathbf{x}_{n-1})$ around $\hat{\mathbf{x}}_{n-1}$ and $\mathbf{h}_n(\mathbf{x}_n)$ around the prior estimate $\hat{\mathbf{x}}_n^- = \hat{\mathbf{x}}_{n|n-1}$ leads to

$$\mathbf{f}_n(\mathbf{x}_{n-1}) \cong \mathbf{f}_n(\hat{\mathbf{x}}_{n-1}) + \mathbf{F}_n\varepsilon_{n-1}\,, \tag{98}$$

$$\mathbf{h}_n(\mathbf{x}_n) \cong \mathbf{h}_n(\hat{\mathbf{x}}_n^-) + \mathbf{H}_n\varepsilon_n^-\,, \tag{99}$$

where $\mathbf{F}_n = \frac{\partial \mathbf{f}_n}{\partial \mathbf{x}}\Big|_{\hat{\mathbf{x}}_{n-1}}$ and $\mathbf{H}_n = \frac{\partial \mathbf{h}_n}{\partial \mathbf{x}}\Big|_{\hat{\mathbf{x}}_n^-}$ are Jacobians and $\varepsilon_n^- = \mathbf{x}_n - \hat{\mathbf{x}}_n^-$ is the prior estimation error. Unbiasedness implies $E\{\varepsilon_n\} = \mathbf{0}$, and a first-order approximation of the expectation of $\mathbf{f}_n(\mathbf{x}_{n-1})$ leads to the prior estimate

$$\hat{\mathbf{x}}_n^- = \bar{\mathbf{f}}_n(\mathbf{x}_{n-1}) = \mathbf{f}_n(\hat{\mathbf{x}}_{n-1})\,. \tag{100}$$

The expectation of the prior error can be written as $E\{\varepsilon_n^-\} = E\{\mathbf{x}_n - \hat{\mathbf{x}}_n^-\} = E\{\mathbf{F}_n\varepsilon_{n-1} + \mathbf{B}_n\mathbf{w}_n\} = \mathbf{0}$. A first-order approximation of the average of $\mathbf{h}_n(\mathbf{x}_n)$ is

$$\bar{\mathbf{h}}_n(\mathbf{x}_n) = \mathbf{h}_n(\hat{\mathbf{x}}_n^-)\,. \tag{101}$$

With (96) and (97) linearized in this way, UFIR filtering can be applied as shown below.

1.8.1 Iterative EFIR filtering

Following the Kalman filter extension [45], the extended unbiased FIR (EUFIR) filter estimate is shown in [46] to be

$$\hat{\mathbf{x}}_l = \hat{\mathbf{x}}_l^- + \mathbf{K}_l[\,\mathbf{z}_l - \mathbf{h}_l(\hat{\mathbf{x}}_l^-)]\,, \tag{102}$$

where the prior estimate is $\hat{\mathbf{x}}_n^- = \mathbf{f}_n(\hat{\mathbf{x}}_{n-1})$, the bias correction gain is $\mathbf{K}_l = \mathbf{G}_l\mathbf{H}_l^T$, and $\mathbf{G}_l$ is computed iteratively as

$$\mathbf{G}_l =[\,\mathbf{H}_l^T\mathbf{H}_l + (\mathbf{F}_l\mathbf{G}_{l-1}\mathbf{F}_l^T)^{-1}]^{-1}\,. \tag{103}$$

The iterative EUFIR filtering algorithm is given in Table 7. As can be seen, it has the same structure as Table 1, except for the nonlinear functions in the estimate. Although the EUFIR algorithm traditionally does not use noise statistics or initial error statistics, the estimation error covariance may be required to characterize the performance. An analysis of error covariances is given in [46]. Note that, in contrast to the first-order expansion, (98) and (99), the second-order expansion involves noise statistics. However, as in the extended Kalman filter [28], the higher order expansion typically does not lead to a definitive advantage [46].

Table 7 EUFIR filtering algorithm for TV models

Stage	
Given:	$K, N, m = n - N + 1,$
	$s = m + K - 1, m + K \leqslant l \leqslant n.$
Set:	$\hat{\mathbf{x}}_s$ by (69) and $\mathbf{G}_s$ by (70)
Update:	$\mathbf{F}_l$ by (100), $\mathbf{H}_l$ by (101),
	$\mathbf{G}_l =[\,\mathbf{H}_l^T\mathbf{H}_l + (\mathbf{F}_l\mathbf{G}_{l-1}\mathbf{F}_l^T)^{-1}]^{-1},$
	$\mathbf{K}_l = \mathbf{G}_l\mathbf{H}_l^T,$
	$\hat{\mathbf{x}}_l = \mathbf{f}_l(\hat{\mathbf{x}}_{l-1}) + \mathbf{K}_l[\,\mathbf{z}_l - \mathbf{h}_l(\hat{\mathbf{x}}_l^-)]\ .$
Remark:	Use the estimate when $l = n$.

1.9 Memory for OUFIR estimators

The window size N plays an important role in UFIR estimators. If $N < N_{\text{opt}}$, denoising appears to be inefficient: the random error dominates, although bias is negligible. If $N > N_{\text{opt}}$, the random error is small, but bias affects the estimate.

Estimation of N_{opt} is still a challenging mathematical problem that requires finding the derivative of the estimate with respect to the convolution length N. Even so, there are several available approaches. For low-degree polynomial models, N_{opt} was found analytically in [47]. A more general approach employing the bandlimited property of signals was developed in [20]. Finally, an efficient algorithm exploiting measurements was recently proposed in [48]. In any case, it is much simpler to estimate a scalar N_{opt}, rather than accurately estimating matrices $\mathbf{Q}_n$ and $\mathbf{R}_n$ as is required in the Kalman filter.

1.9.1 Bandlimited signals

In real applications, a measured signal is causal and bandlimited with some maximum frequency W. By the Shannon theorem, the maximum sampling interval that prevents aliasing is $T = 1/2W$. If measurements are obtained with sampling interval T, then only $N = K$ points are available for averaging by the K-state FIR estimator. If we use larger N, then the estimate will be biased. In order to avoid bias, we would need the model to be represented with a larger number of states, and such a model may not be acceptable or available.

Typically, measurements are provided at time steps $\tau < T$ or even $\tau \ll T$ and N_{opt} can be calculated as follows [20]:

$$N_{\text{opt}} = \left\lfloor (2W\tau)^{-1} + 1 \right\rfloor\,, \tag{104}$$

where $\lfloor x \rfloor$ means the maximum integer that is less than or equal to x. A simple analysis shows that if $N > N_{\text{opt}}$, aliasing causes a bias. If $N < N_{\text{opt}}$, noise reduction is inefficient.

1.9.2 Known reference model

If the reference model for $\mathbf{x}_n$ is known, then the full-horizon UFIR filter (Table 2) with window size $N = n + 1$

can be applied to produce the estimate $\hat{\mathbf{x}}_n \triangleq \hat{\mathbf{x}}_{n|n}$ via measurements taken from time index 0 to n. Following (60), the N-variant MSE $\mathbf{P}_n$ can thus be defined by

$$\mathbf{P}_n = E\{\epsilon_n \epsilon_n^T\}, \tag{105}$$

where $\epsilon_n = \mathbf{x}_n - \hat{\mathbf{x}}_n$. The MSE (105) will be minimal if we minimize it to obtain n_{opt} and let $N_{\text{opt}} = n_{\text{opt}} + 1$. In doing so, one can either minimize the trace $\text{tr}(\mathbf{P}_n)$ if N_{opt} needs to be applied to all of the states, or the (kk)th component $P_{(kk)n}$ of $\mathbf{P}_n$ corresponding to the kth state, respectively,

$$N_{\text{opt}} = \arg\min_n (\text{tr}\,\mathbf{P}_n) + 1\,, \tag{106}$$

$$N_{k\text{opt}} = \arg\min_n (P_{(kk)n}) + 1\,. \tag{107}$$

It has been shown in [48] that by increasing n, the first minimum in both (106) and (107) corresponds to N_{opt}. The problem, however, arises when the reference model $\mathbf{x}_n$ is unknown, as it usually is.

1.9.3 Unknown reference model

The case of unknown model for $\mathbf{x}_n$ is typical. In this case, we estimate N_{opt} via the mean square value (MSV):

$$\mathbf{V}_n = E\{(\mathbf{z}_n - \mathbf{H}\hat{\mathbf{x}}_n)(\mathbf{z}_n - \mathbf{H}\hat{\mathbf{x}}_n)^T\}\,, \tag{108}$$

in which $\mathbf{z}_n$ and $\hat{\mathbf{x}}_n$ are both known. It has been shown in [48] that the estimate $\hat{N}_{\text{opt}}$ of N_{opt} can be found via (108) to minimize the estimation error of all of the states or the kth one as, respectively,

$$\hat{N}_{\text{opt}} \cong \arg\min_n \left(\frac{\partial}{\partial n}\text{tr}\mathbf{V}_n\right) + 1\,, \tag{109}$$

$$\hat{N}_{k\text{opt}} \cong \arg\min_n \left(\frac{\partial}{\partial n} V_{(kk)n}\right) + 1\,, \tag{110}$$

where $V_{(kk)n}$ is the (kk)th component of $\mathbf{V}_n$. The minimization is performed by increasing n, starting with $K-1$, until the first minimum. To avoid ambiguities when minimizing these functions, the number of points used in the expected value must be sufficiently large, and smoothing of the objective function may be desirable.

1.10 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 8.

1.10.1 OUFIR vs. OFIR

Beginning with the early limited memory filter of Jazwinski [5], OFIR filtering has been under development for several decades. In control theory, fundamental progress was achieved by Kwon et al. and his followers [7,35,49-53]. In signal processing, solutions were found by Shmaliy et al. [8,20,27]. It was shown in [52] that different kinds of limited memory filters [5,54] 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 Figure 4: *The error difference between the OFIR and OUFIR estimates diminishes as N increases.* Note that the boundary value 10 . . . 30 in Figure 4 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. But in some applications, OFIR filters can be more appropriate because of their better accuracy.

1.10.2 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 (except for the zero-mean assumption), allows

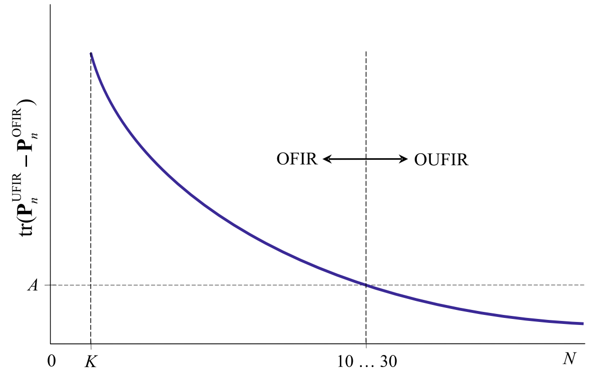

Figure 4 Effect of the estimator window size *N* on the error difference between OUFIR and OFIR estimators. Threshold *A* indicates where the difference becomes visually indistinguishable.

Table 8 Critical evaluation of the Kalman, OFIR, and OUFIR filters

	Kalman	**Batch OFIR**	**Iterative OFIR**	**Batch OUFIR**	**Iterative OUFIR**
	[6]	**[7,8,20]**	**[7,49]**	**[3,4]**	**[19,20]**
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 characteristics:	White	Arbitrary	White	Arbitrary	Arbitrary
System model:	Stochastic	Arbitrary	Arbitrary	Arbitrary	Arbitrary
Filter memory (points):	2	N_{opt}	N_{opt}	N_{opt}	N_{opt}
Stability:	May diverge	BIBO	BIBO	BIBO	BIBO
Operation:	Fast	Slow	Medium	Medium	Approximately N_{opt} times slower than Kalman; Fast with parallel computing
Memory consumption:	Small	Large	Medium	Large	Approximately N_{opt} times more than Kalman
Computational complexity:	Low	High	Medium	Medium	Low

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.

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 Figure 5. 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 (Figure 6).

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.*

Simulation results confirming these observations can be found in [19,23,46].

2 Conclusions

The UFIR algorithms discussed in this paper cover many applied problems associated with filtering, smoothing, and prediction of discrete-time state-space models. The most general conclusions one may arrive at by analyzing these estimators are the following: 1) UFIR algorithms are able to provide nice suboptimal estimates that are acceptable for many applications; 2) The optimal window size N_{opt} can easily be estimated experimentally; 3) The extra time required by the UFIR iterations can be alleviated with parallel computing; and 4) The extra memory required by the UFIR estimators is not a problem for modern computers. So, we conclude that UFIR algorithms are strong rivals

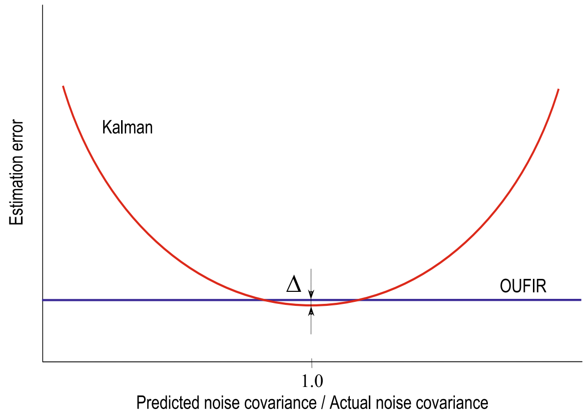

Figure 5 Effect of errors in the noise covariances of the Kalman and OUFIR estimates. The value Δ depends on N and becomes insignificant when $N_{\text{opt}} \gg 1$.

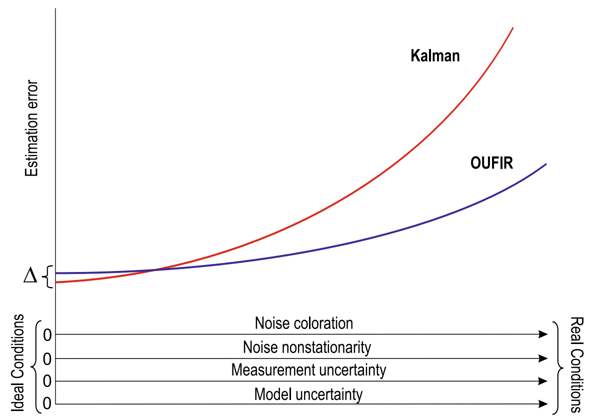

Figure 6 Effect of operating conditions on the Kalman and OUFIR estimates. The value Δ depends on N and becomes insignificant when $N_{\text{opt}} \gg 1$.

to the Kalman filter for real-world applications. The iterative UFIR estimator commonly outperforms the OFIR one even if N_{opt} is not large, and it is able to outperform the Kalman filter under real-world operating conditions and when the noise statistics are not known exactly. That makes UFIR algorithms highly attractive for applications. We see the following main trends in further developments of FIR estimators. Optimal FIR filtering and smoothing strongly require fast Kalman-like algorithms which are similar to those developed for UFIR estimators and considered in this paper. Such algorithms are required for small N_{opt}. In turn, iterative UFIR algorithms need further optimization and robustification in non-Gaussian environments and under the uncertainties. Special attention should also be paid to fast algorithms for the determination of N_{opt}. Provided such modifications, one may expect new efficient FIR solutions.

Endnotes

[a] $\hat{\mathbf{x}}_{n+p|n}$ means the estimate at time $n+p$ given measurements up to and including time n. Here, $p=0$ corresponds to filtering, $p>0$ corresponds to p-step prediction, and $p<0$ corresponds to q-lag smoothing, where $q=-p$. We simplify notation by using $\hat{\mathbf{x}}_{n+p} \triangleq \hat{\mathbf{x}}_{n+p|n}$.

[b] In different applications, the FIR estimator memory is also called the *receding horizon* [53], *sliding window* [55], *averaging interval* [56], etc.

Appendix

The covariance upper bound for TV models

Consider the MSE $\mathbf{P}_{l+p} = E\{\epsilon_{l+p}\epsilon_{l+p}^T\}$ in which we substitute the estimate $\hat{\mathbf{x}}_{n+p}$ with (36),

$$\begin{aligned}\mathbf{P}_{l+p} = E\{[\,\mathbf{F}_{l+p}\epsilon_{l+p-1} + \mathbf{B}_{l+p}\mathbf{w}_{l+p} \\ - \mathbf{K}_l(\mathbf{z}_l - \mathbf{H}_l\mathcal{Y}_l\hat{\mathbf{x}}_{l+p-1})]\,[\,\ldots\,]^T\}\,.\end{aligned} \tag{111}$$

To find an iterative computation of (111), measurement $\mathbf{z}_l$ needs to be expressed via $\mathbf{x}_{l+p-1}$. That can be done by combining the forward and backward solutions as follows:

$$\begin{aligned}
p > 1, \mathbf{x}_v : \mathbf{z}_l &= \mathbf{H}_l\mathcal{Y}_l\mathbf{x}_v + \mathbf{v}_l \\
&\vdots - \mathbf{H}_l\sum_{j=0}^{p-2}\mathcal{Y}_l(p-j)\mathbf{B}_{v-j}\mathbf{w}_{v-j} \\
p = 2, \mathbf{x}_{l+1} : \mathbf{z}_l &= \mathbf{H}_l\mathbf{F}_{l+1}^{-1}(\mathbf{x}_{l+1} - \mathbf{B}_{l+1}\mathbf{w}_{l+1}) + \mathbf{v}_l \\
p = 1, \mathbf{x}_l : \mathbf{z}_l &= \mathbf{H}_l\mathbf{x}_l + \mathbf{v}_l \\
p = 0, \mathbf{x}_{l-1} : \mathbf{z}_l &= \mathbf{H}_l(\mathbf{F}_l\mathbf{x}_{l-1} + \mathbf{B}_l\mathbf{w}_l) + \mathbf{v}_l \\
p = -1, \mathbf{x}_{l-2} : \mathbf{z}_l &= \mathbf{H}_l\mathbf{F}_l(\mathbf{F}_{l-1}\mathbf{x}_{l-2} + \mathbf{B}_{l-1}\mathbf{w}_{l-1}) \\
&\vdots + \mathbf{H}_l\mathbf{B}_l\mathbf{w}_l + \mathbf{v}_l \\
p < 0, \mathbf{x}_v : \mathbf{z}_l &= \mathbf{H}_l\mathcal{Y}_l\mathbf{x}_v + \mathbf{v}_l + \mathbf{H}_l\mathbf{B}_l\mathbf{w}_l \\
&+ \mathbf{H}_l\sum_{j=1}^{|p|}\mathcal{Y}_l(p+j)\mathbf{B}_{v+j}\mathbf{w}_{v+j}\,,
\end{aligned}$$

where $v = l + p - 1$. Then deductive reasoning gives us

$$\mathbf{z}_l = \mathbf{H}_l(\mathcal{Y}_l\mathbf{x}_{l+p-1} + \mathcal{M}_l) + \mathbf{v}_l\,, \tag{112}$$

where $\mathcal{M}_l \triangleq \mathcal{M}_l(p)$ depends on p as

$$\mathcal{M}_l = \begin{cases} \sum\limits_{j=1}^{|p|}\mathcal{Y}_l(p+j)\mathbf{B}_{l+p-1+j}\mathbf{w}_{l+p-1+j} + \mathbf{B}_l\mathbf{w}_l\,, & p < 0\,, \\ \mathbf{B}_l\mathbf{w}_l\,, & p = 0\,, \\ \mathbf{0}\,, & p = 1\,, \\ -\sum\limits_{j=0}^{p-2}\mathcal{Y}_l(p-j)\mathbf{B}_{l+p-1-j}\mathbf{w}_{l+p-1-j}\,, & p > 1 \end{cases}$$

By (112), the MSE becomes

$$\begin{aligned}\mathbf{P}_{l+p} = E\{[\,(\mathbf{F}_{l+p} - \mathbf{K}_l\mathbf{H}_l\mathcal{Y}_l)\epsilon_{l+p-1} + \mathbf{B}_{l+p}\mathbf{w}_{l+p} \\ - \mathbf{K}_l(\mathbf{H}_l\mathcal{M}_l + \mathbf{v}_l)]\,[\,\ldots\,]^T\}\,.\end{aligned} \tag{113}$$

Taking into account that $\mathbf{P}_{l+p-1} = E\{\epsilon_{l+p-1}\epsilon_{l+p-1}^T\}$, $\mathcal{Y}_l(p+1) = \bar{\mathcal{Y}}_l(p)$ and analyzing products of the noise terms leads to the following:

$$\begin{aligned}
\mathbf{P}_{l+p} &= (\mathbf{I} - \mathbf{K}_l\mathbf{H}_l\bar{\mathcal{Y}}_l)\mathbf{F}_{l+p}\mathbf{P}_{l+p-1}\mathbf{F}_{l+p}^T(\ldots)^T \\
&\quad + \mathbf{B}_{l+p}\mathbf{Q}_{l+p}\mathbf{B}_{l+p}^T + \mathbf{K}_l\mathbf{R}_l^T\mathbf{K}_l^T \\
&\quad + \mathbf{K}_l\mathbf{H}_lE\{\mathcal{M}_l\mathcal{M}_l^T\}\mathbf{H}_l^T\mathbf{K}_l^T - \mathbf{B}_{l+p}E\{\mathbf{w}_{l+p}\mathcal{M}_l^T\}\mathbf{H}_l^T\mathbf{K}_{l+p}^T \\
&\quad - \mathbf{K}_{l+p}\mathbf{H}_lE\{\mathcal{M}_l\mathbf{w}_{l+p}^T\}\mathbf{B}_{l+p}^T \\
&= (\mathbf{I} - \mathbf{K}_l\mathbf{H}_l\bar{\mathcal{Y}}_l)\mathbf{F}_{l+p}\mathbf{P}_{l+p-1}\mathbf{F}_{l+p}^T(\ldots)^T \\
&\quad + \mathbf{B}_{l+p}\mathbf{Q}_{l+p}\mathbf{B}_{l+p}^T + \mathbf{K}_l\mathbf{R}_l\mathbf{K}_l^T \\
&\quad - \hat{\mathcal{Q}}_{l+p}\mathbf{H}_l^T\mathbf{K}_l^T - \mathbf{K}_l\mathbf{H}_l\hat{\mathcal{Q}}_{l+p} + \mathbf{K}_l\mathbf{H}_l\bar{\mathcal{Q}}_l\mathbf{H}_l^T\mathbf{K}_l^T\,,
\end{aligned} \tag{114}$$

where

$$\hat{\mathcal{Q}}_{l+p} = \begin{cases} \mathbf{B}_{l+p}\mathbf{Q}_{l+p}\mathbf{B}_{l+p}^T\bar{\mathcal{Y}}_l^T\,, & p \leqslant 0\,, \\ \mathbf{0}\,, & p > 0 \end{cases} \tag{115}$$

and in view of the fact that future noise is unknown and is commonly estimated as 0, $\bar{\mathbf{Q}}_l \triangleq \bar{\mathbf{Q}}_l(p)$ can be written as

$$\bar{\mathcal{Q}}_l = \begin{cases} \sum\limits_{j=2}^{|p|} \mathcal{Y}_l(p+j)\mathbf{B}_{l+p-1+j}\mathbf{Q}_{l+p-1+j} & \\ \times\mathbf{B}_{l+p-1+j}^T\mathcal{Y}_l^T(p+j) + \mathbf{B}_l\mathbf{Q}_l\mathbf{B}_l^T\,, & p \leqslant 0\,, \\ \mathbf{0}\,, & p > 0\,. \end{cases} \tag{116}$$

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Electronics Engineering, Universidad de Guanajuato, Salamanca, 36855, Mexico. [2]Department of Electrical and Computer Engineering, Cleveland State University, Cleveland, OH 44115, USA.

Received: 6 November 2012 Accepted: 28 April 2013
Published: 30 May 2013

References

1. CF Gauss, *Theory of the Combination of Observations Least Subject to Errors*. (SIAM Publ, Philadelphia, 1995). Transl. by Stewart, GW
2. SM Kay, *Fundamentals of Statistical Signal Processing*. (Prentice Hall, New York, 2001)
3. PS Kim, ME Lee, A new FIR filter for state estimation and its applications. J. Comput. Sc. Techn. **22**, 779–784 (2007)
4. YS Shmaliy, An unbiased FIR filter for TIE model of a local clock in applications to GPS-based timekeeping. IEEE Trans. Ultrason., Ferroel. Freq. Contr. **53**, 862–870 (2006)
5. AH Jazwinski, *Stochastic Processes and Filtering Theory*. (Academic Press, New York, 1970)
6. RE Kalman, A new approach to linear filtering and prediction problems. J. Basic Eng. **82**, 35–45 (1960)
7. OK Kwon, WH Kwon, KS Lee, FIR filters and recursive forms for discrete-time state-space models. Automatica. **25**, 715–728 (1989)
8. YS Shmaliy, O Ibarra-Manzano, Time-variant linear optimal finite impulse response estimator for discrete-time state-space models. Int. J. Adapt. Contr. Signal Process. **26**, 95–104 (2012)
9. B Gibbs, *Advanced Kalman Filtering, Least-Squares and Modeling*. (Wiley, New York, 2011)
10. P Ferrari, A Flammini, S Rinaldi, A Bondavalli, F Brancati, Experimental characterization of uncertainty sources in a software-only synchronization system. IEEE Trans. Instrum. Meas. **61**, 1512–1521 (2012)
11. M Blum, On the mean square noise power of an optimum linear discrete filter operating on polynomial plus white noise input. IRE Trans. Inform. Theory. **3**, 225–231 (1957)
12. KR Johnson, Optimum, linear, discrete filtering of signals containing a nonrandom component. IRE Trans. Inform. Theory. **2**, 49–55 (1956)
13. P Heinonen, Y Neuvo, FIR-median hybrid filters with predictive FIR structures. IEEE Trans. Acoust. Speech Signal, Process. **36**, 892–899 (1988)
14. SJ Ovaska, O Vainio, TI Laakso, Design of predictive IIR filters via feedback extension of FIR forward predictors. IEEE Trans. Instrum. Meas. **46**, 1196–1201 (1997)
15. S Samadi, A Nishihara, Explicit formula for predictive FIR filters and differentiators using Hahn orthogonal polynomials. IEICE Trans. Fundamentals. **E90-A**, 1511–1518 (2007)
16. PS Kim, An alternative FIR filter for state estimation in discrete-time systems. Digital Signal Process. **20**, 935–943 (2010)
17. YS Shmaliy, LJ Morales-Mendoza, FIR smoothing of discrete-time polynomial signals in state space. IEEE Trans. Signal Process. **58**, 2544–2555 (2010)
18. YS Shmaliy, Unbiased FIR filtering of discrete-time polynomial state-space models. IEEE Trans. Signal Process. **57**, 1241–1249 (2009)
19. YS Shmaliy, An iterative Kalman-like algorithm ignoring noise and initial conditions. IEEE Trans. Signal Process. **59**, 2465–2473 (2011)
20. YS Shmaliy, Linear optimal FIR estimation of discrete time-invariant state-space models. IEEE Trans. Signal Process. **58**, 3086–3096 (2010)
21. DH Balbuena, O Sergiyenko, V Tyrsa, L Burtseva, MR Lopez, Signal frequency measurement by rational approximations. Measurement. **42**, 136–144 (2009)
22. J Levine, Invited Review Article: The statistical modeling of atomic clocks and the design of time scales. Rev. Sc. Instr. **83**, 021101-1–021101-28 (2012)
23. D Simon, YS Shmaliy, Unified forms for Kalman and finite impulse response filtering and smoothing. Automatica. **49**, 1892–1899 (2013)
24. WQ Wang, Phase noise suppression in GPS-disciplined frequency synchronization systems. Fluctuation Noise L. **10**, 303–313 (2011)
25. HE Rauch, F Tung, CT Striebel, Maximum likelihood estimates of linear dynamic systems. AIAA J. **3**, 1445–1450 (1965)
26. H Stark, JW Woods, *Probability, Random Processes, and Estimation Theory for Engineers*, 2nd edn. (Prentice Hall, Upper Saddle River, 1994)
27. YS Shmaliy, *GPS-based Optimal FIR Filtering of Clock Models*. (Nova Science Publ., New York, 2009)
28. D Simon, *Optimal State Estimation: Kalman, H_∞ and Nonlinear Approaches*. (Wiley, New York, 2006)
29. F Gustafsson, *Adaptive Filtering and Change Detection*. (Wiley, New York, 2000)
30. S Haykin, Lessons on adaptive systems for signal processing, communications, and control. IEEE Signal Process. Mag. **16**, 39–48 (1999)
31. JB Moore, Discrete-time fixed lag smoothing algorithms. Automatica. **9**, 163–173 (1973)
32. Y Bar-Shalom, XR Li, T Kirubarajan, *Estimation with Applications to Tracking and Navigation*. (Wiley, New York, 2001)
33. RE Helmick, WD Blair, SA Hoffman, Fixed-interval smoothing for Markovian switching systems. IEEE Trans. Inform. Theory. **41**, 1845–1855 (1995)
34. JH Kim, J Lyou, Target state estimator design using FIR filter and smoother. Trans. Control, Automation, and Syst. Eng. **4**, 305–310 (2002)
35. CK Ahn, PS Kim, Fixed-lag maximum likelihood FIR smoother for state-space models. IEICE Electonics Express. **5**, 11–16 (2008)
36. KK Biswas, AK Mahalanabis, An approach to fixed-lag smoothing problems. IEEE Trans. Aerospace Electron. Syst. **AES-8**, 676–682 (1972)
37. Y Theodor, U Shaked, Game theory approach to H_∞-optimal discrete-time fixed-point and fixed-lag smoothing. IEEE Trans. Autom. Contr. **39**, 1944–1948 (1994)
38. RJ Vaccaro, *Digital Control: A State-Space Approach*. (McGraw-Hill, New York, 1995)
39. BP Carlin, NG Polson, DS Stoffer, A Monte-Carlo approach to nonnormal and nonlinear state-space modeling. J. Am. Statist. Assoc. **87**, 493–500 (1992)
40. M Lamothe, M Auclair, C Hamzaoui, S Huot, Towards a prediction of long-term anomalous fading of feldspar IRSL. Radiat. Meas. **37**, 493–498 (2003)
41. ITU, *Timing Characteristics of Primary Reference Clocks*. (ITU-T Recommendation G.811, 1997)
42. J Levine, Time synchronization over the Internet using an adaptive frequency-locked loop. IEEE Trans. Ultrason., Ferroelect., Freq. Contr. **46**, 888–896 (1999)
43. T Iwata, M Imae, T Suzuyama, H Murakami, Y Kawasaki, Simulation and ground experiments of remote synchronization system for on-board crystal oscillator of Quazi-Zenith Satelite System. J. Inst. Navig. **53**, 231–235 (2006)
44. J Makhoul, Linear prediction: A tutorial review. Proc. IEEE. **63**, 561–580 (1975)
45. H Cox, On the estimation of state variables and parameters for noisy dynamic systems. IEEE Trans. Autom. Contr. **9**, 5–12 (1964)
46. Shmaliy Y S, Suboptimal FIR filtering of nonlinear models in additive white Gaussian noise. IEEE Trans. Signal Process. **60**, 5519–5527 (2012)
47. YS Shmaliy, J Muñoz-Diaz, L Arceo-Miquel, Optimal horizons for a one-parameter family of unbiased FIR filters. Digital Signal Process. **18**, 739–750 (2008)

48. F Ramirez-Echeverria, A Sarr, YS Shmaliy, Optimal memory for discrete-time FIR filters in state-space. IEEE Trans. Signal Process (2013). in press
49. SH Han, WH Kwon, PS Kim, Quasi-deadbeat minimax filters for deterministic state-space models. IEEE Trans. Automat. Contr. **47**, 1904–1908 (2002)
50. CK Ahn, Strictly passive FIR filtering for state-space models with external disturbance. Int. J. Electron. Commun. **66**, 944–948 (2012)
51. CK Ahn, SH Han, New H_∞ FIR smoother for linear discrete-time state-space models. IEICE Trans. Commun. **E91.B**, 896–899 (2010)
52. WH Kwon, YS Suh, YI Lee, OK Kwon, Equivalence of finite memory filters. IEEE Trans. Aerosp. Electron. Syst. **30**, 968–972 (1994)
53. Kwon W H, S Han, *Receding Horizon, Control: Model Predictive Control for State Models*. (Springer, London, 2005)
54. FC Schweppe, *Uncertain Dynamic Systems*. (Prentice-Hall, New York, 1973)
55. CB Papadias, DTM Slock, Normalized sliding window constant modulus and decision-directed algorithm: a link between blind equalization and classical adaptive filtering. IEEE Trans. Signal Process. **45**, 231–235 (1997)
56. J Treicher, M Larimore, New processing technique based on the constant modulus adaptive algorithm. IEEE Trans. Acoust., Speech, Signal Process. **33**, 420–431 (1985)

doi:10.1186/1687-6180-2013-113
Cite this article as: Shmaliy and Simon: **Iterative unbiased FIR state estimation: a review of algorithms.** *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:113.