6-2013

Unified Forms for Kalman and Finite Impulse Response Filtering and Smoothing

Daniel J. Simon
Cleveland State University, d.j.simon@csuohio.edu

Yuriy S. Shmaliy
Universidad de Guanajuato

Publisher's Statement

NOTICE: this is the author's version of a work that was accepted for publication in Automatica. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Automatica, 49, 6, (06-01-2013); 10.1016/j.automatica.2013.02.026

Original Citation

D. Simon and Y. S. Shmaliy, "Unified forms for Kalman and finite impulse response filtering and smoothing," Automatica, vol. 49, pp. 1892-1899, 2013.

Repository Citation

Unified forms for Kalman and finite impulse response filtering and smoothing☆

Dan Simon[a,1], Yuriy S. Shmaliy[b]

[a] Cleveland State University, Cleveland, OH, USA
[b] Universidad de Guanajuato, Salamanca, Mexico

1. Introduction

We assume that we have a linear system described as

$$x_i = F_i x_{i-1} + w_i$$
$$y_i = H_i x_i + v_i \quad (1)$$

where the time index $i \geq 1$, x_i is the K-dimensional state vector, y_i is the M-dimensional measurement, $\{w_i\}$ is a process noise sequence, $\{v_i\}$ is a measurement noise sequence, and system matrices F_i and H_i are known. Our objective is to estimate x_i based on the measurements and our knowledge of the system dynamics.

We use the term *estimator* to refer to the class of algorithms that includes filtering, prediction, and smoothing. A *filter* estimates x_i based on measurements up to and including time i. A *predictor* estimates x_i based on measurements prior to time i. A *smoother* estimates x_i based on measurements prior to time i, at time i, and later than time i.

Kalman estimation

The Kalman smoother can be written in fixed-lag form, fixed-interval form, or fixed-point form. These algorithms can be described as follows (Anderson & Moore, 2005) and (Simon, 2006, ch. 9).

- A fixed-lag smoother estimates x_i for $i \geq 1$ using measurements up to and including time $i + q$ for a fixed value of $q > 0$.
- A fixed-interval smoother estimates x_i for $i \in [1, N]$ using measurements up to and including time N.
- A fixed-point smoother estimates x_i using measurements up to and including time $i+q$ for a fixed value of i and for $q = 1, 2, \ldots$.

As we will see in Section 2, the form of the Kalman smoother is much different than that of the Kalman filter. Section 2.1 derives a Kalman smoother that is in the same form as the predictor/corrector form of the Kalman filter.

The Kalman filter is an infinite impulse response (IIR) filter; that is, each measurement y_m affects each estimate $\hat{x}_i$ for all $m \leq i$. The IIR nature of the Kalman filter makes it sensitive to modeling errors (Heffes, 1966; Nishimura, 1966; Soong, 1965). Over the past few decades, researchers have proposed many methods of making the Kalman estimator more robust (Peña & Guttman, 1988). Kalman estimation with uncertainties in the system matrices has been considered by many authors (Kosanam & Simon, 2004; Theodor & Shaked, 1996; Xie, Lu, Zhang, & Zhang, 2004; Zhang, Heemink, & Van Eijkeren, 1995); this is often called adaptive or robust Kalman estimation (Hide, Moore, & Smith, 2003). Methods for identifying noise covariances are presented in Alspach (1974); Mehra (1972) and Myers and Tapley (1976).

Finite impulse response estimation

Whereas the research efforts mentioned above aimed to improve the Kalman estimator in the presence of mismodeling, we propose instead to use a finite impulse response (FIR) estimator. The advantages of transversal FIR estimators over Kalman estimators were recognized as far back as the 1960s, particularly in the areas of stability and robustness (Jazwinski, 1970). In spite of their history, FIR filters are not commonly used for state estimation. This is probably due to their analytical complexity and large computational burden. FIR smoothers can be used for polynomial models (Wang, 1991; Zhou & Wang, 2004). Order-recursive FIR smoothers were proposed for state space (Yuan & Stuller, 1994). General receding horizon FIR smoother theory has been developed (Ahn & Kim, 2008; Han & Kwon, 2007, 2008; Kwon, Han, Kwon, & Kwon, 2007). More recently, unbiased FIR (UFIR) smoothing of polynomial state space models has been considered (Shmaliy & Morales-Mendoza, 2010), and FIR smoothing was developed from the general p-shift estimator (Shmaliy, 2010, 2011; Shmaliy & Ibarra-Manzano, 2012). Iterative UFIR algorithms have also been developed (Shmaliy, 2010, 2011). These algorithms have the same predictor/corrector structure as the Kalman filter, often ignore the statistics of the noise and initial estimation errors, and become virtually optimal as the length of the FIR window increases.

Overview of the paper

Section 2 gives a brief review of Kalman filtering and smoothing, and derives a unified form for the two algorithms. Section 3 gives a review of UFIR filtering and smoothing, and derives two distinct but mathematically equivalent unified forms for the two algorithms. It also derives upper and lower bounds for the estimation error covariance. Section 4 presents some simulation results.

2. Kalman filtering and smoothing

If our estimate of x_i is based on measurements up to and including time t, we denote the estimate as $\hat{x}_{i|t}$. If $t = i$ then we have $\hat{x}_{i|i}$, which is called the *a posteriori* state estimate. If $t = i - 1$ then we have $\hat{x}_{i|i-1}$, which is called the *a priori* state estimate. If $t > i$, then we have a non-causal smoothed estimate. Suppose the following conditions hold:

(1) $\{w_i\}$ and $\{v_i\}$ are zero-mean, Gaussian, white, and uncorrelated, with known covariances Q_i and R_i respectively;
(2) We have an initial state estimate before any measurements are processed that we denote as $\hat{x}_{0|0}$;
(3) $(x_0 - \hat{x}_{0|0}) \sim N(0, P_{0|0})$, which means that the initial estimation error is Gaussian and zero-mean with covariance $P_{0|0}$.

Then the Kalman filter output is the mean of the state conditioned on measurements up to and including the current time:

$$\hat{x}_{i|i} = E(x_i|y_1, y_2, \ldots, y_i) \tag{2}$$

for $i \geq 1$. Furthermore, the Kalman filter estimate is the one that minimizes the trace of the covariance of the estimation error. The Kalman filter algorithm can be described as shown in Fig. 1, although there are also other equivalent formulations of the Kalman filter (Simon, 2006).

In the case of smoothing, we use future measurements to obtain the state estimate. One well-known smoothing algorithm is called the Rauch–Tung–Striebel (RTS) smoother, which is a type of fixed-interval smoother (Rauch, Tung, & Striebel, 1965) and (Simon, 2006, Section 9.4.2). Given measurements y_i for $i \in [1, N]$, the RTS smoother outputs $\hat{x}_{i|N}$ for all $i \in [0, N]$. The RTS smoother algorithm is summarized in Fig. 2.

$\hat{x}_{0|0} = E(x_0)$
$P_{0|0} = E\left[(x_0 - \hat{x}_{0|0})(x_0 - \hat{x}_{0|0})^T\right]$
For $i = 1, 2, \cdots$
  $\hat{x}_{i|i-1} = F_i \hat{x}_{i-1|i-1}$
  $P_{i|i-1} = F_i P_{i-1|i-1} F_i^T + Q_i$
  $K_i = P_{i|i-1} H_i^T (H_i P_{i|i-1} H_i^T + R_i)^{-1}$
  $\hat{x}_{i|i} = \hat{x}_{i|i-1} + K_i(y_i - H_i \hat{x}_{i|i-1})$
  $P_{i|i} = (I - K_i H_i) P_{i|i-1}$
Next i

Fig. 1. The Kalman filter. K_i is the Kalman gain, $P_{i|i}$ is the *a posteriori* estimation error covariance, and $P_{i|i-1}$ is the *a priori* estimation error covariance.

Execute the Kalman filter for $i = 1, 2, \cdots, n$ (See Fig. 1)
Initialize $P_n^s = P_{n|n}$
For $q = 1, 2, \cdots, n$
  $i = n - q$
  $K_i^s = P_{i|i} F_{i+1}^T (P_{i+1|i})^{-1}$
  $P_i^s = P_{i|i} - K_i^s (P_{i+1|i} - P_{i+1}^s)(K_i^s)^T$
  $\hat{x}_{i|n} = \hat{x}_{i|i} + K_i^s (\hat{x}_{i+1|n} - \hat{x}_{i+1|i})$
Next i

Fig. 2. The RTS smoother. K_i^s is the Kalman smoother gain, and P_i^s is the covariance of the error of the smoothed estimate at time i.

2.1. Unified Kalman filtering and smoothing

Fig. 1 shows that the Kalman filter estimate can be written in the form

$$\hat{x}_{i|i} = \gamma_i \hat{x}_{i-1|i-1} + K_i y_i$$
$$\text{where } \gamma_i = (I - K_i H_i) F_i \tag{3}$$

for $i \geq 1$. This is called a predictor/corrector form. However, the smoothed estimate in Fig. 2 does not have this form. We would like to find a similar form for the smoothed estimate:

$$\hat{x}_{n-q|n} = \gamma_{n,q} \hat{x}_{n-1|n-1} + \sum_{m=n-q+1}^{n} \beta_{n,q,m} y_m \tag{4}$$

where the smoother lag $q > 0$. Such a form could serve at least two purposes.

First, we find it mathematically attractive to obtain unified forms for different algorithms. We see this in many areas of science and engineering (Fonseca & Fleming, 1998; Guerreiro & Trigueiros, 2010; Miller & Boxer, 1999), so the parallel form of (3) and (4) is intuitively appealing.

Second, the smoother form of (4) may have practical benefits because it directly shows the additional sensitivity of the smoothed estimate to each measurement, beyond the sensitivity already incorporated in $\hat{x}_{n-1|n-1}$. $\beta_{n,q,m}$ is the sensitivity of $\hat{x}_{n-q|n}$ to y_m for $m \in [n-q+1, n]$ beyond the sensitivity that is implicit in $\hat{x}_{n-1|n-1}$. These sensitivities could be used to process measurements in order of decreasing sensitivity so that the most important measurements are processed first, in case the timeliness of the smoothed estimate is important.

Note that all of the measurements up to and including time $n-1$ are incorporated in the filtered estimate $\hat{x}_{n-1|n-1}$ in (4). However, the additional contribution of those measurements to obtain the smoothed estimate $\hat{x}_{n-q|n}$ is determined by the $\beta_{n,q,m}$ coefficients. We suppose that the estimate $\hat{x}_{n-1|n-1}$ is available and that the user may want to process only a subset of the measurements to obtain the smoothed estimate.

To be more specific, (4) can be written algorithmically by computing

$$\mu(l) = \text{value of } m \text{ in the } l\text{-th largest value of } \beta_{n,q,m} \tag{5}$$

for $m \in [n-q+1, n]$ and $l \in [1, q]$. When we say "l-th largest value of $\beta_{n,q,m}$", we implicitly assume some matrix or vector norm. After

$\hat{x}^{(0)}_{n-q|n} = \gamma_{n,q}\hat{x}_{n-1|n-1}$
For $l = 1$ to q
$\quad \hat{x}^{(l)}_{n-q|n} = \hat{x}^{(l-1)}_{n-q|n} + \beta_{n,q,\mu(l)}y_{\mu(l)}$
Next l

Fig. 3. This algorithm gives a smoothed estimate that improves with each iteration.

computing (5), we perform the algorithm of Fig. 3. This algorithm gives a smoothed estimate that improves with each iteration so that at the end of the loop, $\hat{x}^{(q)}_{n-q|n} = \hat{x}_{n-q|n}$ is the optimal Kalman smoother output. More importantly, the algorithm processes the measurements in an optimal order; they are processed in the order of their influence on the estimate. We give an example in Section 2.2.

As a counterpoint to the above discussion, we note that it is $\beta_{n,q,m}y_m$ rather than only $\beta_{n,q,m}$ that contributes to the estimate $\hat{x}_{n-q|n}$ in (4). Therefore, the user may prefer to prioritize larger $\beta_{n,q,m}y_m$ terms rather than larger $\beta_{n,q,m}$ terms. According to this perspective, (5) would be rewritten as

$$\mu(l) = \text{value of } m \text{ in } l\text{-th largest value of } \beta_{n,q,m}y_m \tag{6}$$

before performing the smoothing algorithm that follows (5). However, if the measurements are not available to the user before designing the smoothing algorithm, then the best approach that the user can take is given by (5).

Another possible use for the form of (4) is sensor selection or design. Before we implement a filtering or smoothing algorithm, we need to select or design measurement sensors. We can use the Kalman filter algorithm of Fig. 1 to see how much a given sensor contributes to the filtered state estimate, and we can use the unified smoother form of (4) to see how much the sensor contributes to the further improvement of the smoothed estimate.

To write the RTS smoother algorithm of Fig. 2 in the form of (4), we note from Fig. 2 that

$$\hat{x}_{n-1|n} = \hat{x}_{n-1|n-1} + K^s_{n-1}(\hat{x}_{n|n} - \hat{x}_{n|n-1}). \tag{7}$$

Proceeding inductively, it can be shown that

$$\hat{x}_{n-q|n} = \hat{x}_{n-q|n-q} + \sum_{l=1}^{q} L_{n,q,l}(\hat{x}_{n-q+l|n-q+l} - \hat{x}_{n-q+l|n-q+l-1})$$

$$\text{where } L_{n,q,l} = \prod_{r=0}^{l-1} K^s_{n-q+r} \tag{8}$$

for $q \geq 0$. Now notice from Fig. 1 that

$$\begin{aligned}\hat{x}_{n|n} &= (I - K_nH_n)F_n\hat{x}_{n-1|n-1} + K_ny_n \\ &= (I - K_nH_n)F_n(I - K_{n-1}H_{n-1})F_{n-1}\hat{x}_{n-2|n-2} \\ &\quad + (I - K_nH_n)F_nK_{n-1}y_{n-1} + K_ny_n.\end{aligned} \tag{9}$$

Proceeding inductively, it can be shown that

$$\begin{aligned}\hat{x}_{n|n} &= \prod_{j=0}^{q} J_{n-j}F_{n-j}\hat{x}_{n-q-1|n-q-1} \\ &\quad + \sum_{l=1}^{q}\left(\prod_{j=0}^{l-1} J_{n-j}F_{n-j}\right)K_{n-l}y_{n-l} + K_ny_n\end{aligned}$$

$$\text{where } J_n = I - K_nH_n \tag{10}$$

for $q \geq 0$. Now replace n with $n-1$ and replace q with $q-2$ in (10) to obtain

$$\begin{aligned}\hat{x}_{n-1|n-1} &= M_{q-2}\hat{x}_{n-q|n-q} \\ &\quad + \sum_{l=1}^{q-2} M_{l-1}K_{n-1-l}y_{n-1-l} + K_{n-1}y_{n-1}\end{aligned}$$

$$\text{where } M_r = \begin{cases} \prod_{j=0}^{r} J_{n-1-j}F_{n-1-j} & \text{if } r \geq 0 \\ I & \text{if } r = -1 \\ (J_nF_n)^{-1} & \text{if } r = -2. \end{cases} \tag{11}$$

Now write (8) as

$$\begin{aligned}\hat{x}_{n-q|n} &= \hat{x}_{n-q|n-q} + \sum_{l=1}^{p} L_{n,q,l}\left(\hat{x}_{n-q+l|n-q+l}\right. \\ &\quad \left. - J^{-1}_{n-q+l}\hat{x}_{n-q+l|n-q+l} + J^{-1}_{n-q+l}K_{n-q+l}y_{n-q+l}\right) \\ &= \hat{x}_{n-q|n-q} + \sum_{l=1}^{q} L_{n,q,l}\left((I - J^{-1}_{n-q+l})\hat{x}_{n-q+l|n-q+l}\right. \\ &\quad \left. + J^{-1}_{n-q+l}K_{n-q+l}y_{n-q+l}\right).\end{aligned} \tag{12}$$

Now use (11) to find $\hat{x}_{n-q+l|n-q+l}$ for $l \in [0, q]$ in terms of $\hat{x}_{n-1|n-1}$ and y_j for $j \in [n-q+l+1, n-1]$, and substitute for $\hat{x}_{n-q+l|n-q+l}$ in the above equation. After some lengthy algebra, we obtain

$$\hat{x}_{n-q|n} = \gamma_{n,q}\hat{x}_{n-1|n-1} + \sum_{m=n-q+1}^{n} \beta_{n,q,m}y_m \quad (q \geq 0)$$

$$\text{where } \gamma_{n,q} = M^{-1}_{q-2} + \sum_{l=1}^{q} L_{n,q,l}(I - J^{-1}_{n-q+l})M^{-1}_{q-l-2}$$

$$\beta_{n,q,m} = L_{n,q,q}K_n \quad \text{if } m = n \tag{13}$$

$$\begin{aligned}\beta_{n,q,m} = \Bigg[&L_{n,q,q+m-n}J^{-1}_m - M^{-1}_{q-2}M_{n-2-m} \\ &- \sum_{l=1}^{q+m-n-1} L_{n,q,l}(I - J^{-1}_{n-q+l})M^{-1}_{q-l-2}M_{n-2-m}\Bigg]K_m\end{aligned}$$

if $m < n$.

This algorithm, which we call the *unified Kalman filter/smoother*, is mathematically identical to the Kalman filter of Fig. 1 for $q = 0$, and mathematically identical to the RTS smoother of Fig. 2 for $q > 0$. Some similarities and differences between the RTS and unified forms of the smoother are as follows.

(1) Both forms require that the Kalman filter execute before smoothing. Both forms require the forward error covariances $P_{i|i}$ and $P_{i+1|i}$, for all $i \in [0, n-1]$.
(2) The RTS smoother requires saving the forward state estimates $\hat{x}_{i+1|i}$ for $i \in [0, n-1]$. The unified Kalman smoother requires saving the measurements y_i for $i \in [1, n]$. Therefore, if the measurement dimension is much smaller than the state dimension, the unified form may require less memory.

2.2. Unified Kalman filter/smoother example

Consider the time-invariant system of (1) with

$$F = \begin{bmatrix}1 & \Delta t \\ 0 & 1\end{bmatrix} \qquad H = \begin{bmatrix}1 & 0\end{bmatrix}$$

$$Q = \begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix} \qquad R = 10$$

$$x_0 = \begin{bmatrix}1 & 0\end{bmatrix}^T \qquad P_{0|0} = \begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix} \tag{14}$$

where $\Delta t = 0.1$. Suppose the system runs until time index $n = 41$ and that we want to find the smoothed estimate with a lag $q = 20$. Eq. (13) gives the following values for $\|\beta_{n,q,m}\|_2$ for $m \in [22, 41]$:

$$\begin{aligned}\|\beta\| = \{&0.10, 0.21, 0.34, 0.50, 0.71, 0.98, 1.34, 1.83, \\ &2.49, 3.37, 4.56, 6.15, 8.30, 11.20, 15.10, 20.34, \\ &27.40, 36.91, 49.71, 0.05\}.\end{aligned} \tag{15}$$

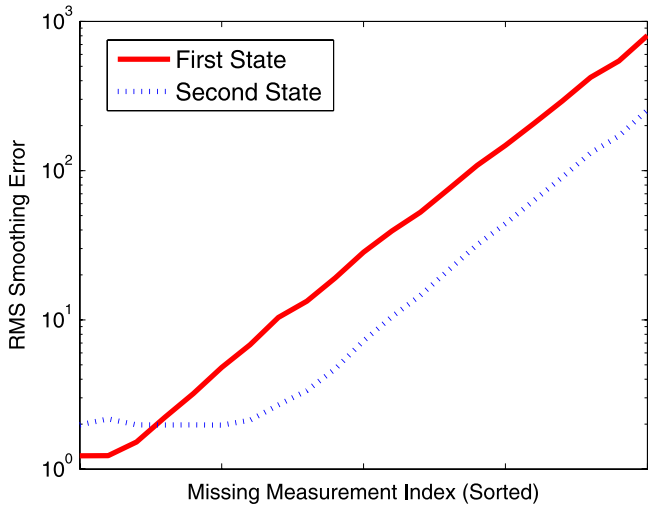

Fig. 4. This figure illustrates the effect of neglecting increasingly important measurements in a smoothed estimate.

This means that y_{40} is the most important measurement for the $\hat{x}_{n-q}$ estimate, y_{39} is the second most important measurement, and so on. Suppose we need to neglect one of the measurements in (13) due to computational constraints. It stands to reason that neglecting y_{41} would result in the smallest degradation of $\hat{x}_{n-q}$ from its optimal value, neglecting y_{22} would result in the second smallest degradation, and so on. We can numerically confirm this by using (13) to calculate

$$\hat{x}_{n-q|n,r} = \gamma_{n,q}\hat{x}_{n-1|n-1} + \sum_{m=n-q+1}^{r-1} \beta_{n,q,m} y_m + \sum_{m=r+1}^{n} \beta_{n,q,m} y_m \tag{16}$$

for $r \in [22, 41]$. That is, $\hat{x}_{n-q|n,r}$ is calculated in the same way as the optimal smoothed estimate, except it does not use y_r. Fig. 4 shows the RMS estimation error of 100 Monte Carlo simulations of $\hat{x}_{n-q|n,r}$ as a function of the missing measurement index r. The left-most point corresponds to the optimal smoothed estimate, and the other elements on the horizontal axis are in order of increasing $\|\beta\|$. As predicted, the estimation error gets worse as we leave out more important measurements. We further see that if we neglect one of the few least important measurements, then smoothing performance degrades only slightly relative to the optimal performance.

3. Unified UFIR filtering and smoothing

This section presents two forms for unified UFIR filtering and smoothing based on the p-shift UFIR estimator (Shmaliy, 2010; Shmaliy & Ibarra-Manzano, 2012). We begin with some preliminaries, and then derive the two unified forms in Section 3.1. We derive estimation error bounds in Section 3.2.

Suppose we have the linear system (1) where y_n is the most recent measurement. The UFIR estimator uses the N most recent measurements to obtain the filtered estimate $\hat{x}_n$, or the smoothed estimate $\hat{x}_{n-q}$ for some value of lag $q \in [1, N-1]$, where N is a user-specified smoothing interval. We often set $q = \lfloor N/2 \rfloor$, where $\lfloor\cdot\rfloor$ is the *floor* function. However, other values of q may provide lower estimation errors (Shmaliy & Morales-Mendoza, 2010). Since $\hat{x}_n$ and $\hat{x}_{n-q}$ are functions of N, we sometimes write them as $\hat{x}_n(N)$ and $\hat{x}_{n-q}(N)$.

The UFIR filter/smoother in this section ignores noise statistics and initial estimation errors. The UFIR estimator requires an optimal averaging interval of N_{opt} points in order for the mean square error (MSE) of the estimate to be minimal. It was recently shown in Shmaliy (2012) that N_{opt} can be estimated with high accuracy by minimizing the derivative of $E\{[y_n - H_n\hat{x}_n(N)][y_n - H_n\hat{x}_n(N)]^T\}$ with respect to N. An example of this approach is given in Section 4.1.

Like the Kalman filter, the UFIR estimator can be given in a fast iterative form for filtering and prediction (Shmaliy, 2011). For smoothing, however, we need to modify the estimator to obtain an iterative form. To provide this modification, we write the smoothed estimate (Shmaliy & Ibarra-Manzano, 2012, Eq. (27)) at the first time point in the smoothing interval as

$$\hat{x}_m = \mathcal{H}_{n,m}^{-1} Y_{n,m} \tag{17}$$

where $m = n - N + 1$, and $\mathcal{H}_{n,m}^{-1} = (H_{n,m}^T H_{n,m})^{-1} H_{n,m}^T \in \mathcal{R}^{K\times M(n-m+1)}$ is the generalized left inverse of $H_{n,m}$, and

$$H_{n,m} = \bar{H}_{n,m} F_{n,m} \in \mathcal{R}^{M\alpha\times K}$$

$$Y_{n,m} = \begin{bmatrix} y_n^T & y_{n-1}^T & \cdots & y_m^T \end{bmatrix}^T \in \mathcal{R}^{M\alpha\times 1}$$

$$F_{n,m} = \Big[\underbrace{\mathcal{F}_{n,0}^{m+1^T} \quad \mathcal{F}_{n,1}^{m+1^T} \quad \cdots \quad F_{m+1}^T}_{n-m+1} \ \mathbf{I} \Big]^T \in \mathcal{R}^{K\alpha\times K}$$

$$\bar{H}_{n,m} = \operatorname{diag}\Big(\underbrace{H_n \ H_{n-1} \ \cdots \ H_m}_{n-m+1} \Big) \in \mathcal{R}^{M\alpha\times K\alpha}$$

$$\mathcal{F}_{r,h}^{r-g} = \prod_{i=h}^{g} F_{r-i} \in \mathcal{R}^{K\times K} \tag{18}$$

where $\alpha = n - m + 1$. To provide a unified UFIR smoothing equation for arbitrary lag q, the transition matrix $\mathcal{B}_{n,m}(q)$ must be specified such that $\hat{x}_{n-q} = \mathcal{B}_{n,m}(q)\hat{x}_m$, where $\hat{x}_m$ is given in (17). By combining the forward-time and backward-time solutions (Stark & Woods, 1994), this matrix can be found as

$$\mathcal{B}_{n,m}(q) = \begin{cases} \mathcal{F}_{n-q,0}^{m+1}, & q \leqslant n-m-1 \\ I, & q = n-m \\ \left(\mathcal{F}_{m,0}^{n-q+1}\right)^{-1}, & q \geqslant n-m+1. \end{cases} \tag{19}$$

The most general batch form of the unified UFIR filter ($q = 0$) and smoother ($q > 0$) is thus

$$\hat{x}_{n-q} = \mathcal{B}_{n,m}(q)\mathcal{H}_{n,m}^{-1} Y_{n,m} \tag{20}$$

where $q \geqslant 0$. Assuming $0 \leqslant q < N-1$, one may also use the form of (20) given in Shmaliy (2011, Eq. (20)), although that form has the limitations discussed in Section 3.1.1 below. Note that (20) is similar to Shmaliy (2010, Eq. (42)), although that result is for time-invariant systems; and it is also similar to Shmaliy and Ibarra-Manzano (2012, Eq. (29)), although that result uses noise statistics.

If we observe that the filter estimate ($q = 0$) is

$$\hat{x}_n = \mathcal{F}_{n,0}^{m+1}\mathcal{H}_{n,m}^{-1} Y_{n,m} \tag{21}$$

then (20) can be written as

$$\hat{x}_{n-q} = \mathcal{B}_{n,m}(q)(\mathcal{F}_{n,0}^{m+1})^{-1}\hat{x}_n. \tag{22}$$

This equation plays a key role in the derivation of the second form of the UFIR estimator discussed in Section 3.1.2.

3.1. Iterative UFIR filtering and smoothing

This section derives two forms of the iterative unified UFIR filter/smoother, which are similar in form to the unified Kalman filter/smoother.

3.1.1. The first form of the unified UFIR filter/smoother

Following the derivations given in Shmaliy (2010, Appendices I and II), (20) becomes

$$\hat{x}_{l-q} = F_{l-q}\hat{x}_{l-q-1} + K_l(y_l - H_l\mathcal{Y}_l\hat{x}_{l-q-1}) \tag{23}$$

for $l \in [m+K, n]$, where

$$\bar{\mathcal{Y}}_l = \begin{cases} \mathcal{F}_{l,0}^{l-q+1} = \prod_{i=0}^{q-1} F_{l-i} & \text{if } q > 0 \\ I & \text{if } q = 0 \end{cases}$$

$$\mathcal{Y}_l = \bar{\mathcal{Y}}_l F_{l-q}. \tag{24}$$

Since $\bar{\mathcal{Y}}_l$ and $\mathcal{Y}_l$ are functions of q, we sometimes write them as $\bar{\mathcal{Y}}_l(q)$ and $\mathcal{Y}_l(q)$. Note that (23) defines an iterative procedure, so $\hat{x}_{l-q}$ is not optimal for $l < n$. The iteration (23) leads to the optimal UFIR q-lag smoothed estimate (assuming that $N = N_{\text{opt}}$) when $l = n$. When $l = n$, (23) gives $\hat{x}_{n-q}$ (or, to be more explicit, $\hat{x}_{n-q|n}$). However, when $l < n$, $\hat{x}_{l-q}$ in (23) is simply an intermediate variable that we use to eventually obtain $\hat{x}_{n-q}$. This is because $\hat{x}_{l-q}$ in (23) is not equal to the batch form of (20) unless $l = n$.

The bias correction gain $K_l \triangleq K_{l,m}(q)$ is

$$K_l = G_l\bar{\mathcal{Y}}_l^T H_l^T \tag{25}$$

where the generalized noise power gain (GNPG) $G_l \triangleq G_{l,m}(q)$ can be computed iteratively using

$$\begin{aligned} G_l &= F_{l-q}\left(\mathcal{Y}_l^T H_l^T H_l \mathcal{Y}_l + G_{l-1}^{-1}\right)^{-1} F_{l-q}^T \\ &= \left[\bar{\mathcal{Y}}_l^T H_l^T H_l \bar{\mathcal{Y}}_l + (F_{l-q}G_{l-1}F_{l-q}^T)^{-1}\right]^{-1}. \end{aligned} \tag{26}$$

The initial values for this iteration, $\hat{x}_{m+K-1-q}$ and G_{m+K-1}, are

$$\hat{x}_{s-q} = \mathcal{B}_{s,m}(q)\mathcal{H}_{s,m}^{-1}Y_{s,m} \tag{27}$$

$$G_s = \mathcal{B}_{s,m}(q)(H_{s,m}^T H_{s,m})^{-1}\mathcal{B}_{s,m}^T(q) \tag{28}$$

where $s = m + K - 1$. The index variable l in (23) ranges from $m+K$ to n, and the smoothed estimate $\hat{x}_{n-q}$ is obtained when $l = n$ in (23).

The UFIR estimator does not depend strongly on initial conditions. This is similar to the Kalman filter where the effect of initial conditions decays as we process more measurements. Therefore, in many practical applications, we can approximately set (27) to zero and (28) to the identity matrix. This simplification gives relatively good estimates if $N \gg 1$, although it may not be accurate otherwise.

The first unified UFIR filter/smoother form (23) is summarized in Fig. 5. If $q = 0$ the algorithm becomes the UFIR filter, and if $q > 0$ the algorithm becomes the UFIR smoother. For time-invariant systems the algorithm is greatly simplified, though we do not show the simplified version here.

3.1.2. The second form of the unified UFIR filter/smoother

We see from (23) that when $q = 0$,

$$\begin{aligned} &\hat{x}_l = F_l\hat{x}_{l-1} + K_l(y_l - H_l F_l \hat{x}_{l-1}) \\ &K_l = G_l H_l^T \\ &G_l = \left[H_l^T H_l + \left(F_l G_{l-1} F_l^T\right)^{-1}\right]^{-1} \end{aligned} \tag{29}$$

for $l \in [m+K, n]$, with initial values

$$\hat{x}_s = \mathcal{F}_{s,0}^{m+1}\mathcal{H}_{s,m}^{-1}Y_{s,m} \tag{30}$$

$$G_s = \mathcal{F}_{s,0}^{m+1}\left(H_{s,m}^T H_{s,m}\right)^{-1}\left(\mathcal{F}_{s,0}^{m+1}\right)^T \tag{31}$$

Given: K = number of states
n = most recent measurement time index
N = number of points in estimator interval
q = smoother time lag ($q = 0$ for filtering)
Define: $m = n - N + 1$
= time index of earliest measurement used
$s = m + K - 1$
Eq. (27): $\hat{x}_{s-q} = \mathcal{B}_{s,m}(q)\mathcal{H}_{s,m}^{-1}Y_{s,m}$
Eq. (28): $G_s = \mathcal{B}_{s,m}(q)(H_{s,m}^T H_{s,m})^{-1}\mathcal{B}_{s,m}^T(q)$
For $l = m + K$ to n
$G_l = \left[H_l^T H_l + (F_l G_{l-1} F_l^T)^{-1}\right]^{-1}$
$K_l = \prod_{i=0}^{q-1} F_{l-i} G_l H_l^T$
$\hat{x}_{l-q} = F_{l-q}\hat{x}_{l-q-1} + K_l(y_l - H_l\mathcal{Y}_l\hat{x}_{l-q-1})$
Next l

Fig. 5. The first form of the iterative unified UFIR filter/smoother. The optimal value of N can be obtained as shown in Section 4.1.

Given K = number of states
n = most recent measurement time index
N = number of points in estimator interval
q = smoother time lag ($q = 0$ for filtering)
Define $m = n - N + 1$
= time index of earliest measurement used
$s = m + K - 1$
Eq. (30): $\hat{x}_s = \mathcal{F}_{s,0}^{m+1}\mathcal{H}_{s,m}^{-1}Y_{s,m}$
Eq. (31): $G_s = \mathcal{F}_{s,0}^{m+1}\left(H_{s,m}^T H_{s,m}\right)^{-1}\left(\mathcal{F}_{s,0}^{m+1}\right)^T$
For $l = m + K$ to n
$G_l = \left[H_l^T H_l + \left(F_l G_{l-1} F_l^T\right)^{-1}\right]^{-1}$
$\hat{x}_l = F_l\hat{x}_{l-1} + G_l H_l^T(y_l - F_l\hat{x}_{l-1})$
Next l
$\hat{x}_{n-q} = \mathcal{B}_{n,m}(q)\left(\mathcal{F}_{n,0}^{m+1}\right)^{-1}\hat{x}_n$

Fig. 6. The second form of the iterative unified UFIR filter/smoother. The optimal value of N can be obtained as shown in Section 4.1.

where $s = m+K-1$. After the index l finishes iterating from $m+K$ to n, (29) provides the filtered estimate $\hat{x}_n$, and the q-lag smoother estimate is computed by retrodicting $\hat{x}_n$ to time $n-q$ using (22).

The second unified UFIR filter/smoother form is summarized in Fig. 6. As with the first form, if $q = 0$ the algorithm is the UFIR filter, and if $q > 0$ the algorithm is the UFIR smoother. As with the first form, the algorithm is greatly simplified for time-invariant systems, although we do not show the simplified version here.

3.1.3. Comparison between the two unified UFIR forms

There are slight but definite differences between the two unified UFIR filter/smoother forms. The first form of Fig. 5 computes smoothed estimates $\hat{x}_{n-N+K-q}, \hat{x}_{n-N+K-q+1}, \ldots, \hat{x}_{n-q}$, and only the final smoothed estimate is optimal. The second form of Fig. 6 computes filtered estimates $\hat{x}_{n-N+K}, \hat{x}_{n-N+K+1}, \ldots, \hat{x}_n$, only the last one of which is optimal, and the optimal smoothed estimate $\hat{x}_{n-q}$ is retrodicted from the filtered estimate $\hat{x}_n$. The second form requires slightly more computational effort because of the retrodiction, but also allows optimal smoothed estimates to be retrodicted from $\hat{x}_n$ for any smoothing lag $q \in [0, N]$. In summary, the most distinctive differences between the two forms are the following: (1) the first form calculates some intermediate smoothed estimates prior to the averaging interval, but the second form does not; and (2) the second form involves two stages of processing (an iterative loop and a retrodiction), but the first form produces the smoothed estimate in only one stage of processing (an iterative loop). The preferred form therefore depends on the application.

3.2. Estimation error covariance bounds

This section derives bounds for the UFIR filter and smoother estimation errors. Define the instantaneous error and its covariance as

$$\epsilon_{l-q} = x_{l-q} - \hat{x}_{l-q}$$

$$P_{l-q} = E\{\epsilon_{l-q}\epsilon_{l-q}^T\}. \tag{32}$$

If we try to calculate (32) iteratively using (23), then the calculation of P_{l-q} will require continually expanding matrix operations at each iteration. Finding a rigorous closed-form analytical solution for the UFIR smoother covariance via (20) is a topic of current research. Instead, reasonably accurate covariance bounds may be sufficient for practical applications. We discuss upper and lower UFIR smoother covariance bounds below.

3.2.1. Upper bound of UFIR covariance

This section derives an upper bound (UB) for the UFIR smoother covariance P_{n-q}^{UB}. We start by substituting $\hat{x}_{l-q}$ from (23) in (32) to obtain

$$\begin{aligned} P_{l-q} &= E\{\epsilon_{l-q}\epsilon_{l-q}^T\} \\ &= E\left\{\left[F_{l-q}\epsilon_{l-q-1} + w_{l-q} - K_l(y_l - H_l\mathcal{Y}_l\hat{x}_{l-q-1})\right]\right. \\ &\quad \left.[\cdots]^T\right\}. \end{aligned} \tag{33}$$

To find an iterative computation of (33), we express y_l in terms of x_{l-q-1} as

$$\begin{aligned} y_l &= H_l(F_l x_{l-1} + w_l) + v_l \\ &= H_l F_l(F_{l-1}x_{l-2} + w_{l-1}) + H_l w_l + v_l \\ &\vdots \\ &= H_l\mathcal{Y}_l x_{l-q-1} + H_l\sum_{j=1}^{q}\mathcal{Y}_l(q-j)w_{l-q-1+j} \\ &\quad + H_l w_l + v_l \end{aligned} \tag{34}$$

which can be written as

$$y_l = H_l(\mathcal{Y}_l x_{l-q-1} + \mathcal{M}_l) + v_l \tag{35}$$

where $\mathcal{M}_l \triangleq \mathcal{M}_l(q)$

$$= \begin{cases} \sum_{j=1}^{q}\mathcal{Y}_l(q-j)w_{l-q-1+j} + w_l & \text{if } q > 0 \\ w_l & \text{if } q = 0. \end{cases}$$

Now we substitute (35) in (33) to get an iteration of P_l from $l = m + K$ to $l = n$. This iteration gives an upper bound for P_n because it is based on the iteration of (23), but $\hat{x}_{l-q}$ in (23) is not optimal for any value of $l < n$. Substituting (35) in (33) gives the following:

$$\begin{aligned} P_{l-q}^{\text{UB}} &= E\{[F_{l-q}\epsilon_{l-q-1} + w_{l-q} - K_l y_l + K_l H_l\mathcal{Y}_l\hat{x}_{l-q-1}][\cdots]^T\} \\ &= E\{[F_{l-q}\epsilon_{l-q-1} + w_{l-q} - K_l H_l\mathcal{Y}_l x_{l-q-1} \\ &\quad - K_l H_l\mathcal{M}_l - K_l v_l + K_l H_l\mathcal{Y}_l\hat{x}_{l-q-1}][\cdots]^T\} \\ &= E\{[F_{l-q}\epsilon_{l-q-1} - K_l H_l\mathcal{Y}_l\epsilon_{l-q-1} + w_{l-q} \\ &\quad - K_l H_l\mathcal{M}_l - K_l v_l][\cdots]^T\} \\ &= E\{[(F_{l-q} - K_l H_l\mathcal{Y}_l)\epsilon_{l-q-1} + w_{l-q} \\ &\quad - K_l(H_l\mathcal{M}_l + v_l)][\cdots]^T\}. \end{aligned} \tag{36}$$

Expanding this equation for $q > 0$ gives

$$\begin{aligned} P_{l-q}^{\text{UB}} &= (I - K_l H_l\bar{\mathcal{Y}}_l)F_{l-q}P_{l-q-1}^{\text{UB}}F_{l-q}^T(\cdots)^T \\ &\quad + Q_{l-q} + K_l R_l^T K_l^T + K_l H_l E\{\mathcal{M}_l\mathcal{M}_l^T\}H_l^T K_l^T \\ &\quad - E\{w_{l-q}\mathcal{M}_l^T\}H_l^T K_{l-q}^T - K_{l-q}H_l E\{\mathcal{M}_l w_{l-q}^T\} \\ &= (I - K_l H_l\bar{\mathcal{Y}}_l)F_{l-q}P_{l-q-1}^{\text{UB}}F_{l-q}^T(\cdots)^T \\ &\quad + Q_{l-q} + K_l R_l^T K_l^T + K_l H_l(\bar{\mathcal{Q}}_l + Q_l)H_l^T K_l^T \\ &\quad - Q_{l-q}\bar{\mathcal{Y}}_l^T H_l^T K_l^T - K_l H_l\bar{\mathcal{Y}}_l Q_{l-q} \\ &= (I - K_l H_l\bar{\mathcal{Y}}_l)(F_{l-q}P_{l-q-1}^{\text{UB}}F_{l-q}^T + Q_{l-q})(\cdots)^T \\ &\quad + K_l R_l^T K_l^T + K_l H_l[\bar{\mathcal{Q}}_l + Q_l - \bar{\mathcal{Y}}_l Q_{l-q}\bar{\mathcal{Y}}_l^T]H_l^T K_l^T \end{aligned} \tag{37}$$

where $\bar{\mathcal{Q}}_l \triangleq \bar{\mathcal{Q}}_l(q)$ can be written as

$$\bar{\mathcal{Q}}_l = \sum_{j=1}^{q}\mathcal{Y}_l(q-j)Q_{l-q-1+j}\mathcal{Y}_l^T(q-j). \tag{38}$$

This upper bound involves accumulating process noise covariances at each iteration. Therefore, the upper bound is relatively tight for small N, and is more conservative when $N \gg 1$. In a similar way, the conservativeness of the upper bound increases with q. This means that the upper bound is most useful (that is, most strict) for applications in which both N and q are small.

3.2.2. Lower bound of UFIR covariance

We can find the lower bound (LB) of the UFIR smoother covariance P_{n-q}^{LB} in two different ways. First, we use (37), which was derived from the first UFIR form of Section 3.1.1. We neglect the process noise covariances in (37) to obtain

$$P_{l-q}^{\text{LB}} = (I - K_l H_l\bar{\mathcal{Y}}_l)F_{l-q}P_{l-q-1}^{\text{LB}}F_{l-q}^T(\cdots)^T + K_l R_l^T K_l^T. \tag{39}$$

For time-invariant models, (39) becomes

$$P_{l-q}^{\text{LB}} = (I - K_l H F^q)FP_{l-q-1}^{\text{LB}}F^T(\cdots)^T + K_l R^T K_l^T. \tag{40}$$

Both (39) and (40) correspond to the first UFIR estimator form (23).

We can also use the second UFIR form of Section 3.1.2 to find the LB of the UFIR smoother covariance. We first find the LB for filtering ($q = 0$) as

$$P_l^{\text{LB}} = (I - K_l H_l)F_l P_{l-1}^{\text{LB}}F_l^T(\cdots)^T + K_l R_l^T K_l^T \tag{41}$$

and then use (22) to compute the LB for smoothing:

$$P_{n-q}^{\text{LB}} = \mathcal{B}_{n,m}(q)\left(\mathcal{F}_{n,0}^{m+1}\right)^{-1}P_n^{\text{LB}}\left(\mathcal{F}_{n,0}^{m+1}\right)^{-T}\mathcal{B}_{n,m}^T(q) \tag{42}$$

for time-varying models, and

$$P_{n-q}^{\text{LB}} = F^{-q}P_n^{\text{LB}}F^{-q^T} \tag{43}$$

for time-invariant models, where P_n^{LB} is provided by (41) when $l = n$. Note that the LBs can also be computed via the noise power gain to serve well in the three-sigma sense (Shmaliy & Ibarra-Manzano, 2011).

4. Simulation results

This section presents simulation results to illustrate the theory of the preceding sections.

4.1. Example of N_{opt} estimation

Recall from the UFIR algorithms of Figs. 5 and 6 that the user needs to select N, which is the number of measurements used in the UFIR estimator. If N is too small, then there is too little information to form a reliable estimate of the state. However, if N is too large, then bias errors enter the estimate. As described in Shmaliy (2012),

$$N_{\text{opt}} \approx \arg\min_N \frac{\partial V(N)}{\partial N} \tag{44}$$

where $V(N) = E\left\{[y_n - H_n\hat{x}_n(N)][y_n - H_n\hat{x}_n(N)]^T\right\}$.

We can numerically estimate the above derivative for various values of N, and then use any optimization algorithm to minimize it. In this example we use

$$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \qquad R = (5/6)^2 \tag{45}$$

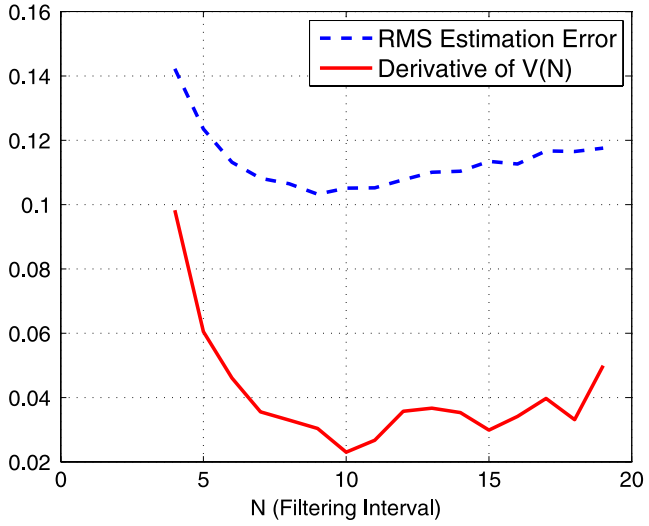

Fig. 7. This figure illustrates the strategy of finding the optimal measurement interval N for the UFIR filter. The estimation error is minimized at about the same value of N as the derivative of $V(N)$.

Table 1
RMS estimation errors for the first state. The true value of $Q(2,2)$ is 1. For each value of assumed $Q(2,2)$, the smaller error between the Kalman and UFIR filter, and the smaller error between the Kalman and UFIR smoother, is shown in bold font.

	Assumed value of $Q(2,2)$				
	0.01	0.1	1	10	100
Kalman filter	1.56	0.71	**0.53**	0.60	0.71
UFIR filter	**0.56**	**0.56**	0.56	**0.56**	**0.56**
Kalman smoother	0.84	0.40	**0.30**	0.35	0.47
UFIR smoother	**0.34**	**0.34**	0.34	**0.34**	**0.34**

where $\Delta t = 0.1$. Fig. 7 shows $\partial V/\partial N$ along with the RMS value of the estimation error, which was obtained from 10,000 UFIR filter time steps. We see that $\partial V/\partial N$ is minimum at $N = 10$, while the estimation error is minimum at $N = 9$. The estimation error is relatively flat near its minimum, which indicates that the UFIR filter is robust for this example. A more accurate estimate of N_{opt} can be obtained by using more time steps to approximate $V(N)$.

4.2. Comparisons between Kalman and UFIR estimators

Next we compare Kalman and UFIR estimator performance. We use the same system parameters as in the previous example. For the UFIR estimators, we set $N = 12$ to simulate an error in our approximation of N_{opt} and to explore its robustness. This gives a UFIR smoother lag $q = N/2 = 6$. We vary the value of $Q(2,2)$ assumed by the Kalman estimator. Process noise is often difficult to characterize, so errors in the Kalman estimator value of Q are common. We run the estimator simulations for $n = 10,000$ time steps. The Kalman smoother uses all n measurements to find the smoothed estimate $\hat{x}_{i|n}$ for $i \in [1, 10000]$ (see Fig. 2). The UFIR smoother uses y_k for $k \in [i-11, i]$ to find the smoothed estimate $\hat{x}_{i-6}$, where $i \in [7, 10000]$ (see Figs. 5 and 6). Tables 1 and 2 compare the performance of the Kalman and UFIR estimators based on 20 Monte Carlo simulations.

We make the following observations about Tables 1 and 2.

(1) When the Kalman estimator assumes the correct statistics of the system, it is optimal and outperforms the UFIR filter/smoother.
(2) The Kalman smoother always outperforms the Kalman filter, and the UFIR smoother always outperforms the UFIR filter.

Table 2
RMS estimation errors for the second state. The true value of $Q(2,2)$ is 1. For each value of assumed $Q(2,2)$, the smaller error between the Kalman and UFIR filter, and the smaller error between the Kalman and UFIR smoother, is shown in bold font.

	Assumed value of $Q(2,2)$				
	0.01	0.1	1	10	100
Kalman filter	3.15	2.42	**2.07**	2.74	5.35
UFIR filter	**2.26**	**2.26**	2.26	**2.26**	**2.26**
Kalman smoother	1.58	1.20	**1.01**	1.40	3.18
UFIR smoother	**1.10**	**1.10**	1.10	**1.10**	**1.10**

(3) As the Kalman estimator assumed value of $Q(2,2)$ deviates from the true value, performance suffers. For this example, performance suffers more when $Q(2,2)$ is smaller than the true value. This agrees with intuition. If we know that we have modeling errors, we should put more emphasis on the measurements rather than on the model information. However, when $Q(2,2)$ decreases, the Kalman gain increases.
(4) UFIR estimator performance is invariant with respect to errors in the Kalman estimator assumed value of $Q(2,2)$. Tables 1 and 2 show that the UFIR estimator clearly outperforms the Kalman estimator when the assumed noise statistics are incorrect.

5. Conclusion

We derived a unified form for the Kalman filter and smoother that explicitly shows the sensitivity of the smoothed estimate to each measurement between the smoothing time point and the end of the smoothing interval. We derived two unified forms for the UFIR filter and smoother, along with bounds for their estimation error covariances. We have seen that although the Kalman estimator is optimal when the system matrices are known, the UFIR estimator can provide better robustness in the case of modeling errors. The UFIR estimator does not require any knowledge of noise statistics, which makes it an attractive alternative to the Kalman estimator. The UFIR estimator requires more computational effort than the Kalman estimator, although this could be circumvented through parallel processing. MATLAB® source code is available on the internet to replicate our examples (Simon, 2012).

Although the unified Kalman estimator is mathematically equivalent to the standard form, we have observed numerical difficulties under certain conditions, which future research should focus on characterizing and correcting. Other future research should focus on obtaining an exact equation for the UFIR estimation error covariance.

References

Ahn, C., & Kim, P. (2008). Fixed-lag maximum likelihood FIR smoother for state-space models. *IEICE Electronics Express*, *5*(1), 11–16.
Alspach, D. (1974). Gaussian sum approximations in nonlinear filtering and control. *Information Sciences*, *7*, 271–290.
Anderson, B., & Moore, J. (2005). *Optimal filtering*. Mineola, New York: Dover (chapter 7).
Fonseca, C., & Fleming, P. (1998). Multiobjective optimization and multiple constraint handling with evolutionary algorithms, I. A unified formulation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, *28*(1), 26–37.
Guerreiro, J., & Trigueiros, D. (2010). A unified approach to the extraction of rules from artificial neural networks and support vector machines. In L. Cao, J. Zhong, & Y. Feng (Eds.), *Advanced data mining and applications* (pp. 34–42). Berlin, Germany: Springer-Verlag.
Han, S., & Kwon, W. (2007). L_2-E FIR smoothers for deterministic discrete-time state-space signal models. *IEEE Transactions on Automatic Control*, *52*(5), 927–932.
Han, S., & Kwon, W. (2008). A note on two-filter smoothing formulas. *IEEE Transactions on Automatic Control*, *53*(3), 849–854.
Heffes, H. (1966). The effect of erroneous models on the Kalman filter response. *IEEE Transactions on Automatic Control*, *11*(3), 541–543.

Hide, C., Moore, T., & Smith, M. (2003). Adaptive Kalman filtering for low-cost INS/GPS. *The Journal of Navigation*, *56*(1), 143–152.
Jazwinski, A. (1970). *Stochastic processes and filtering theory*. New York: Academic.
Kosanam, S., & Simon, D. (2004). Kalman filtering with uncertain noise covariances. *Intelligent systems and control* (pp. 375–379). Honolulu, Hawaii.
Kwon, B., Han, S., Kwon, O., & Kwon, W. (2007). Minimum variance FIR smoothers for discrete-time state space models. *IEEE Signal Processing Letters*, *14*(8), 557–560.
Mehra, R. (1972). Approaches to adaptive filtering. *IEEE Transactions on Automatic Control*, *17*(5), 693–698.
Miller, R., & Boxer, L. (1999). *Algorithms sequential and parallel: a unified approach*. Upper Saddle River, New Jersey: Pearson Education.
Myers, K., & Tapley, B. (1976). Adaptive sequential estimation with unknown noise statistics. *IEEE Transactions on Automatic Control*, *21*(4), 520–523.
Nishimura, T. (1966). On the a priori information in sequential estimation problems. *IEEE Transactions on Automatic Control*, *11*(2), 197–204.
Peña, D., & Guttman, I. (1988). Bayesian approach to robustifying the Kalman filter. In J. Spall (Ed.), *Bayesian analysis of time series and dynamic models* (pp. 227–253). New York: Marcel Dekker.
Rauch, H., Tung, F., & Striebel, C. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, *3*(8), 1445–1450.
Shmaliy, Y. S. (2010). Linear optimal FIR estimation of discrete time-invariant state-space models. *IEEE Transactions on Signal Processing*, *58*(6), 3086–3096.
Shmaliy, Y. S. (2011). An iterative Kalman-like algorithm ignoring noise and initial conditions. *IEEE Transactions on Signal Processing*, *59*(6), 2465–2473.
Shmaliy, Y. S. (2012). Suboptimal FIR filtering of nonlinear models in additive white Gaussian noise. *IEEE Transactions on Signal Processing*, *60*(10), 5519–5527.
Shmaliy, Y. S., & Ibarra-Manzano, O. (2011). Noise power gain for discrete-time FIR estimators. *IEEE Signal Processing Letters*, *18*(4), 207–210.
Shmaliy, Y. S., & Ibarra-Manzano, O. (2012). Time-variant linear optimal finite impulse response estimator for discrete state-space models. *International Journal of Adaptive Control and Signal Processing*, *26*(2), 95–104.
Shmaliy, Y. S., & Morales-Mendoza, L. (2010). FIR smoothing of discrete-time polynomial signals in state space. *IEEE Transactions on Signal Processing*, *58*(5), 2544–2555.
Simon, D. (2006). *Optimal state estimation: Kalman, H_∞, and nonlinear approaches*. Hoboken, New Jersey: John Wiley & Sons.
Simon, D. (2012). *Unified forms for kalman and finite impulse response filtering and smoothing* [Online]. http://academic.csuohio.edu/simond/UnifiedKalmanFIR.
Soong, T. (1965). On a priori statistics in minimum-variance estimation problems. *ASME Journal of Basic Engineering*, *87*(1), 109–112.
Stark, H., & Woods, J. (1994). *Probability, random processes, and estimation theory for engineers* (2nd ed.). Upper Saddle River, New Jersey: Prentice Hall.
Theodor, Y., & Shaked, U. (1996). Robust discrete-time minimum-variance filtering. *IEEE Transactions on Signal Processing*, *44*(2), 181–189.
Wang, X. (1991). NFIR nonlinear filter. *IEEE Transactions on Signal Processing*, *39*(7), 1705–1708.
Xie, L., Lu, L., Zhang, D., & Zhang, H. (2004). Improved robust H_2 and H_∞ filtering for uncertain discrete-time systems. *Automatica*, *40*(5), 873–880.
Yuan, J.-T., & Stuller, J. (1994). Order-recursive FIR smoothers. *IEEE Transactions on Signal Processing*, *42*(5), 1242–1246.
Zhang, X., Heemink, A., & Van Eijkeren, J. (1995). Performance robustness analysis of Kalman filter for linear discrete-time systems under plant and noise uncertainty. *International Journal of System Science*, *26*(2), 257–275.
Zhou, X., & Wang, X. (2004). FIR-median hybrid filters with polynomial fitting. *Digital Signal Processing*, *39*(2), 112–124.