6

OPTIMAL SMOOTHERS

> Many difficulties which nature throws in our way may be smoothed away by the exercise of intelligence.
>
> —Livy [Titus Livius] (59BCE–17CE)

6.1 CHAPTER FOCUS

6.1.1 Smoothing and Smoothers

6.1.1.1 What We Mean by "Optimal Smoothing?" The term *smoothing* has many meanings in different contexts, dating from antiquity. The term *optimal smoothing* has been used from around at least since 1795, when Gauss discovered the method of least squares. It later became a part of Wiener–Kolmogorov filtering theory. Since the 1960s, most of the methods used in optimal smoothing have come from Kalman and the Kalman–Bucy filtering theory.

Modern optimal smoothing methods are derived from the same models as the Kalman filter, for solving the same sort of estimation problem—but not necessarily in real time. The Kalman filter optimizes the use of all measurements made at or before the time that the estimated state vector is valid. Smoothing can do a better job than the Kalman filter by using additional measurements made *after* the time of the estimated state vector.

The term *linear smoothing* is also used to indicate that linear dynamic models of the underlying processes are used as part of the smoother, but this terminology could

Kalman Filtering: Theory and Practice Using MATLAB®, Fourth Edition.
Mohinder S. Grewal and Angus P. Andrews.
© 2015 John Wiley & Sons, Inc. Published 2015 by John Wiley & Sons, Inc.

be interpreted to mean that some sort of straight-line fitting is used. In fact, there are many smoothing methods based on least-squares fitting of locally smooth functions (e.g., polynomial splines) to data—but these methods are not included here. The emphasis here is on extensions of Kalman filtering that use the same linear stochastic systems as models for the underlying dynamic processes, but relax the relative timing constraints on estimates and measurements.

Smoothers are the algorithmic or analog implementations of smoothing methods. If the application has input signal frequencies too high for sampling or computation rates too fast for digital processing, then the analogous Kalman–Bucy theory in continuous time can be used for designing the implementation circuitry—in much the same way as it was done using the Wiener filtering theory.

6.1.2 Kalman Filtering, Prediction, Interpolation and Smoothing

Kalman filtering theory was developed in Chapters 2 through 5. The same theoretical models have been used in developing optimal smoothing methods.

Prediction was also covered in Chapter 5. It is an integral part of Kalman filtering, because the estimated values and their associated covariances of uncertainty are always predicted ahead one time step as a priori variables. In practice, this sort of prediction is generally continued forward whenever anticipated measurements are unavailable or otherwise deemed unreliable.

Interpolation usually refers to estimating state variables during periods of time when no measurements are available, using measurements from adjoining time segments to fill in the "data gaps." Optimal smoothers are also be used for interpolation, even though they are usually defined for estimating state variables at those times when measurements are available.

For example, the Kalman filters in some global navigation satellite system (GNSS) receivers continue forward prediction of position and velocity estimates whenever the signals from satellites are unavailable. This is pure prediction, and it is done to aid in rapid reacquisition of signals when they next become available. Once the signals are again available, however, optimal smoothing can also be used to improve (after the fact) the predicted navigation estimates made during the signal outage. *That* is interpolation.

Smoothing is the subject of this chapter. Its relationship to optimal predictors and optimal filters is illustrated in Figure 6.1, the distinction depending on when an estimated value of the state vector $\hat{\mathbf{x}}$ is valid, and how that time is related to the times at which those measurements used in making that estimate were sampled.

- *Filtering* occupies the diagonal line in the figure, where each state vector estimate is based on those measurements made up to and including the time when the estimate is valid.
- *Prediction* occupies the region below that diagonal line. Each predicted estimate is based on measurement data taken strictly *before* the time that estimate is valid.
- *Smoothing* occupies the region above the diagonal. Each smoothed estimate is based on measurement data taken both *before and after* the time that estimate

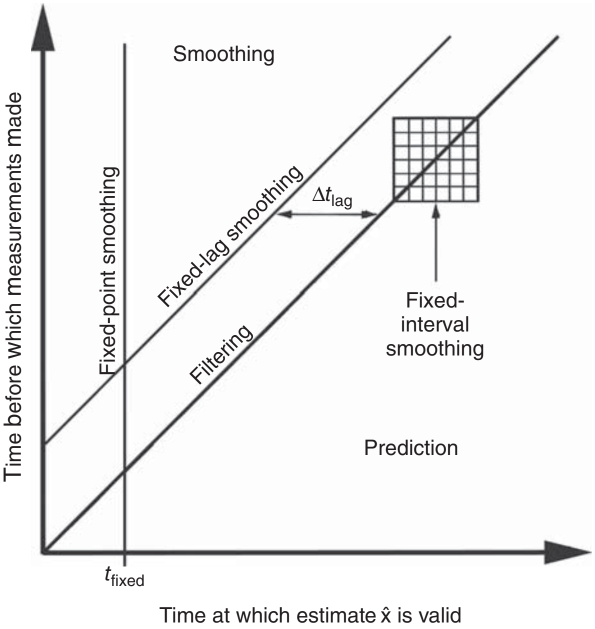

Figure 6.1 Estimate/measurement timing constraints.

is valid. The three different types of smoothers labeled in Figure 6.1 are briefly described in the following subsection.

6.1.3 Types of Smoothers

Most applications of smoothing have evolved into three distinct types, according to the measurement data dependencies of the estimated state vector, as illustrated in Figure 6.1. The following descriptions spell out the constraints on the times measurements are made relative to the times that the value of state vector is to be estimated and provide some examples of how the different smoothers have been used in practice.

6.1.3.1 Fixed-Interval Smoothers Fixed-interval smoothers use all the measurements made at times t_{meas} over a fixed time interval $t_{\mathrm{start}} \leq t_{\mathrm{meas}} \leq t_{\mathrm{end}}$ to produce an estimated state vector $\hat{\mathbf{x}}(t_{\mathrm{est}})$ at times $t_{\mathrm{start}} \leq t_{\mathrm{est}} \leq t_{\mathrm{end}}$ in the same fixed interval—as illustrated by the square area along the diagonal in Figure 6.1.

The time at which fixed-interval smoothing is done can be any time after all these measurements have been taken. The most common use of fixed-interval smoothing is for post-processing of all measurements taken during a test procedure. Fixed-interval smoothing is generally not a real-time process, because the information processing generally takes place after measurements ceased.

Technical descriptions, derivations, and performance analysis of fixed-interval smoothers are presented in Section 6.2.

6.1.3.2 Fixed-Lag Smoothers Fixed-lag smoothers use all measurements made over a time interval $t_{\text{start}} \leq t_{\text{meas}} \leq t_{\text{est}} + \Delta t_{\text{lag}}$ for the estimate $\hat{\mathbf{x}}(t_{\text{est}})$ at time t_{est}. That is, the estimate generated at time t is for the value of x at time $t - \Delta t_{\text{lag}}$, where Δt_{lag} is a fixed lag time, as illustrated by the sloped line above the "filtering" line in Figure 6.1.

Fixed-lag smoothers are used in communications for improving signal estimation. They sacrifice some delay in order to reduce bit error rates. Fixed-lag smoothers run in real time, using all measurement up to the current time, but generate an estimate in "lagged time."

Technical descriptions, derivations, and performance analysis of fixed-lag smoothers are presented in Section 6.3.

6.1.3.3 Fixed-point Smoothers Fixed-point smoothers generate an estimate $\hat{\mathbf{x}}(t_{\text{fixed}})$ of $\boldsymbol{x}$ at a fixed time t_{fixed}, based on all measurements $\mathbf{z}(t_{\text{meas}})$ up to the current time t (i.e., $t_{\text{start}} \leq t_{\text{meas}} \leq t$). These are illustrated by the vertical line in Figure 6.1. Fixed-point smoothers function as predictors when the current time $t < t_{\text{fixed}}$, as filters when $t = t_{\text{fixed}}$, and as smoothers when $t > t_{\text{fixed}}$.

Fixed-point smoothing is useful for estimation problems in which the system state is only of interest at some epochal event time t_{fixed}, which is often the initial state. They are used for initial alignment of inertial navigation systems (INS) [1] and for estimating the initial INS attitudes on vehicles using GNSS receivers as auxiliary navigation sensors. GNSS receivers provide no attitude information during initial INS alignment, but they can detect subsequent patterns of position errors from which initial attitude errors can be estimated using fixed-point smoothing.

Technical descriptions, derivations, and performance analysis of fixed-point smoothers are presented in Section 6.4.

6.1.4 Implementation Algorithms

We describe in this chapter the different types of optimal smoothers, their implementation algorithms, some example applications, and some indication of their relative advantage over optimal filters.

Many different smoothing algorithms have been derived and published, but not all turned out to be practical. Important criteria for smoothing algorithms are

1. Numerical stability. Some early algorithms were actually numerically unstable.
2. Computational complexity, which limits the data rates at which they can operate.
3. Memory requirements.

Some smoother implementations are in continuous time.

6.1.5 Smoothing Applications

In practice, smoothing is often overlooked as a practical method for getting better estimates than those attainable by filtering alone. Depending on attributes of the application, the improvement of smoothing over filtering can be many orders of magnitude in mean-squared estimation uncertainty.

6.1.5.1 Determining Whether and How to Use Smoothing Questions you should consider when deciding whether to use smoothing—and which type of smoothing to use—include the following:

1. What are the constraints imposed by the application on the relative times when the measurements are taken, when the resulting estimates of the system state vector should be valid, and when the results are needed?
   (a) If the estimated state vector values over an entire interval are required, then fixed-interval smoothing should be considered—especially if the results are not needed in near real time.
   (b) If the estimated state vector value is required to be valid only at one instant of time, then fixed-point smoothing is an appropriate estimation method.
   (c) If results are required in near real time (i.e., with limited delays), then fixed-lag smoothing may be appropriate.
2. What is the required, expected, or hoped for improvement (over filtering) of uncertainty in the estimated values of state variables for the intended application?
3. How might the reliability of results be affected by numerical stability of the estimation algorithm or by oversensitivity of performance to assumed model parameter values?
4. How might computational complexity compromise the cost and attainability of implementation? This issue is more important for "real-time" applications of fixed-lag and fixed-point smoothing. Smoothing uses more data for each estimated value, and this generally requires more computation than for filtering. One must always consider the computational resources required, whether they are attainable, and their cost.
5. Are there any serious memory storage limitations? This could be a problem for some applications of fixed-interval smoothing with very large data sets.

6.1.6 Improvement over Filtering

Theoretical limits on the asymptotic improvement of smoothing over filtering were published by Anderson [2] in 1969 for the case that the dynamic system is asymptotically exponentially stable. The limit in that case is a factor of two in mean-squared estimation uncertainty, but greater improvement is attainable for unstable systems.

6.1.6.1 Relative Improvement For *scalar* models (i.e., $n = 1$), the relative improvement of smoothers over filters can be characterized by the ratios of their respective variances of uncertainty in the resulting estimates.

For multidimensional problems (i.e., $n > 1$) with $P_{[s]}$ as the covariance matrix of smoothing uncertainty and $P_{[f]}$ the covariance matrix of filtering uncertainty, we would say that smoothing is an improvement over filtering if

$$P_{[s]} < P_{[f]}, \text{ or } [P_{[f]} - P_{[s]} \text{ is positive definite}].$$

This can generally be determined by implementing the smoother and filter and comparing their covariance matrices of estimation uncertainty.

6.1.6.2 Dependence on Model Parameters In order to gain some insight into the relative benefits of smoothing, and how they depend on attributes of the application, we develop in Sections 6.2 through 6.4 performance analysis models to evaluate how the improvement of smoothing over filtering depends on the parameters of the model. All these models are for linear time-invariant problems, and the models for which we can plot the dependence of performance on problem parameters are all for scalar [1] state variables (i.e., $n = 1$). Even for these scalar models, depending on the parameters of the model, the relative improvement of smoothing over filtering can range from hardly anything to orders of magnitude reduction in mean-squared uncertainty.

6.1.6.3 The Anderson–Chirarattananon Bound This result, due to Brian D. O. Anderson and Surapong Chirarattananon [3], was derived in the frequency domain and uses signal-to-noise ratio (SNR) as a parameter. This supposes that the dynamic process model representing the signal is *stable,* in order that the signal have finite steady-state power.

In Kalman filter models, mean-squared noise is represented by the parameter R, the covariance of measurement noise, and the *signal* by the noise-free part of the measurement $z(t)$ (i.e., Hx). This requires that the dynamic process representing the component of the state vector x in the rank space of H be stable, which implies that the corresponding dynamic coefficient matrix F have negative characteristic values. Sensor noise is assumed to be white, which has constant power spectral density across all frequencies. The peak SNR then occurs at that frequency where the signal power spectral density peaks.

The result, shown in Figure 6.2 in two forms, is a bound on the improvement of smoothing over filtering, defined by the ratio of smoothing error variance to filtering error variance. The curve in Figure 6.2(a) is a lower bound on the improvement ratio; that in Figure 6.2(b) is an upper bound on the equivalent percent improvement in mean-squared signal estimation error from using smoothing instead of filtering.

Additional bounds are derived in this chapter for the relative improvement of each type of smoothing over filtering, in terms of how the improvement depends on the values of the underlying linear stochastic system parameters.

6.2 FIXED-INTERVAL SMOOTHING

6.2.1 Performance Analysis in Continuous Time

6.2.1.1 Scalar Parametric Models

Linear Time-Invariant Models in Continuous Time Smoother performance models are developed in continuous time for pedagogical reasons: the resulting model

[1] Setting $n = 1$ has more to do with dimensionality of the independent parameters (for plotting) than the feasibility of the performance analysis.

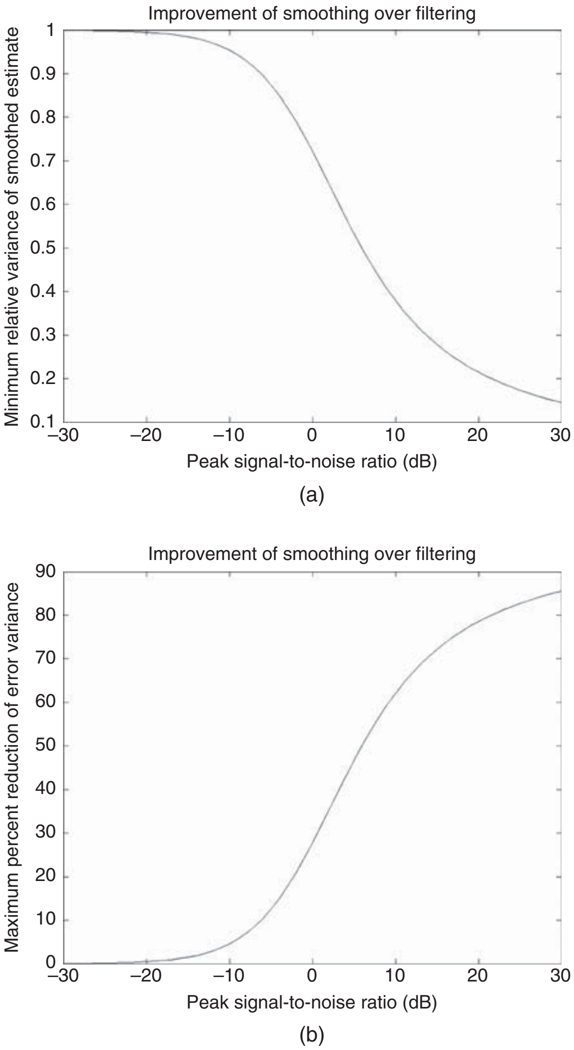

Figure 6.2 Anderson–Chirarattananon bound on smoothing variance improvement over filtering. (a) Relative improvement and (b) percent improvement.

equations are more easily solved in closed form. Smoother performance can also be determined for the discrete-time models, and for higher-dimensional state vectors—although solution may require numerical methods.

Performance Formulas We present some general methods for determining how the relative performance of smoothing with respect to filtering depends on the time-invariant parameters F, Q, H, and R of linear models. They are then demonstrated for the scalar state variable case, although the general approach works for arbitrary state vector and measurement vector dimension.

Stochastic System Model The parametric stochastic system model is

$$\begin{aligned}
\dot{x} &= F\,x + w\,\text{(dynamic system model)} \\
z &= H\,x + v\,\text{(measurement model)} \\
\dot{P} &= FP + PF - PHR^{-1}H^{\mathrm{T}}P + Q\,\text{(Riccati differential equation).}
\end{aligned} \tag{6.1}$$

Parameter Units for $n = 1$ When the state vector has dimension $n = 1$, the scalar Riccati equation variables and parameters have the following units:

$$\begin{array}{ll}
P & \text{has the same units as } x^2. \\
F & \text{has units of } 1/\text{time (e.g., s}^{-1}). \\
H & \text{has } z \text{ (measurement) units divided by } x \text{ (state variable) units.} \\
R & \text{has the units of time} \times z^2 \text{ (measurement units)}^2. \\
Q & \text{has the units of } x^2/\text{time.}
\end{array} \tag{6.2}$$

The seemingly strange units or R and Q are a consequence of stochastic integration using the stochastic differential equation model.

Two-Filter Model For fixed-interval smoothing, performance models are based on two Kalman–Bucy filters:

- A forward filter, running forward in time. At each instant of time, the estimate from the forward filter is based on all the measurements made up to that time, and the associated estimation uncertainty covariance characterizes estimation uncertainty based on all those measurements.
- A backward filter, running backward in time. At each instant of time, the estimate from the backward filter is based on all the measurements made after that time, and the associated estimation uncertainty covariance characterizes estimation uncertainty based on all those measurements. When time is reversed, the sign on the dynamic coefficient matrix F in Equation 6.1 changes, which can make the performance of the backward filter model different from that of the forward filter model. The covariance $P_{[b]}$ of backward filter uncertainty can be different from the covariance $P_{[f]}$ of forward filter uncertainty.

Why the Two-Filter Model? At each time t, the forward filter generates the covariance matrix $P_{[f]}(t)$ representing the mean-squared uncertainty in the estimate $\hat{x}_{[f]}(t)$ using all measurements $z(s)$ for $s \leq t$. Similarly, the backward filter generates the covariance matrix $P_{[b]}(t)$ representing the mean-squared uncertainty in the estimate $\hat{x}_{[b]}(t)$ using all measurements $z(s)$ for $s \geq t$. The optimal smoother combines $\hat{x}_{[f]}(t)$ and $\hat{x}_{[b]}(t)$, using $P_{[f]}(t)$ and $P_{[b]}(t)$ in the Kalman filter, to minimize the resulting covariance matrix $P_{[s]}(t)$ of smoother uncertainty. $P_{[s]}(t)$ will tell us how well the smoother performs.

Smoother Estimate and Uncertainty At each instant of time, the smoother estimate is obtained by optimally combining two-filter models (forward and backward); the estimates obtained by this way will be based on all the measurements, before and after the time of the estimate, over the entire fixed interval. Let

$\hat{x}_{[f]}(t)$ be the forward filter estimate at time t,
$P_{[f]}(t)$ be the associated covariance of estimation uncertainty,
$\hat{x}_{[b]}(t)$ be the backward filter estimate at time t,
$P_{[b]}(t)$ be the associated covariance of estimation uncertainty.

If we treat $\hat{x}_{[f]}(t)$ as the a priori smoother estimate and $\hat{x}_{[b]}(t)$ as an independent measurement, then the optimal smoothed estimate of x at time t will be

$$\hat{x}_{[s]}(t) = \hat{x}_{[f]}(t) + \underbrace{P_{[f]}(t)[P_{[f]}(t) + P_{[b]}(t)]^{-1}}_{\overline{K}}[\hat{x}_{[b]}(t) - \hat{x}_{[f]}(t)],$$

which is essentially the Kalman filter observational update formula with

$$\begin{aligned}
\hat{x}_{[f]}(t) &= \text{the smoother a priori estimate}\\
H &= I\,(\text{identity matrix})\\
z(t) &= \hat{x}_{[b]}(t)\,(\text{backward filter estimate})\\
R &= P_{[b]}(t)\\
\overline{K} &= P_{[f]}(t)[P_{[f]}(t) + P_{[b]}(t)]^{-1}\,(\text{Kalman gain})\\
\hat{x}_{[s]}(t) &= \text{the smoother a posteriori estimate}\\
P_{[s]}(t) &= \text{the a posteriori covariance of smoother estimation uncertainty}\\
&= P_{[f]}(t) - P_{[f]}(t)[P_{[f]}(t) + P_{[b]}(t)]^{-1}P_{[f]}(t)\\
&= P_{[f]}(t)\{I - [P_{[f]}(t) + P_{[b]}(t)]^{-1}P_{[f]}(t)\}\\
&= P_{[f]}(t)[P_{[f]}(t) + P_{[b]}(t)]^{-1}\{[P_{[f]}(t) + P_{[b]}(t)] - P_{[f]}(t)\}\\
&= P_{[f]}(t)[P_{[f]}(t) + P_{[b]}(t)]^{-1}P_{[b]}(t).
\end{aligned} \tag{6.3}$$

The last of these equations characterizes smoother covariance as a function of forward and backward filter covariances, which are the solutions of the appropriate Riccati equations.

6.2.1.2 Infinite-Interval Case This is the easiest model for solving for smoother performance. It generally requires solving two algebraic Riccati equations, but this can be done in closed form for the continuous-time model with scalar state vector.

General Linear Time-Invariant Model In the linear time-invariant case on the interval $-\infty < t < +\infty$, the Riccati equations for the forward and backward Kalman–Bucy filters will be at steady state:

$$\begin{aligned}0 &= \frac{d}{dt}P_{[f]}(t)\\ &= FP_{[f]} + P_{[f]}F^{\mathrm{T}} + Q - P_{[f]}H^{\mathrm{T}}R^{-1}HP_{[f]}\\ 0 &= \frac{d}{dt}P_{[b]}(t)\\ &= -FP_{[b]} - P_{[b]}F^{\mathrm{T}} + Q - P_{[b]}H^{\mathrm{T}}R^{-1}HP_{[b]},\end{aligned}$$

where the backward filter model has the sign on the dynamic coefficient matrix F reversed. In general, these algebraic Riccati equations can be solved by numerical methods and inserted into Equation 6.3 to determine smoother performance.

Scalar State Vector Case This reduces to two quadratic equations, both of which can be solved in closed form for the single positive scalar solution:

$$P_{[f]} = \sqrt{\left(\frac{FR}{H^2}\right)^2 + \left(\frac{QR}{H^2}\right)} + \left(\frac{FR}{H^2}\right) \tag{6.4}$$

$$P_{[b]} = \sqrt{\left(\frac{FR}{H^2}\right)^2 + \left(\frac{QR}{H^2}\right)} - \left(\frac{FR}{H^2}\right) \tag{6.5}$$

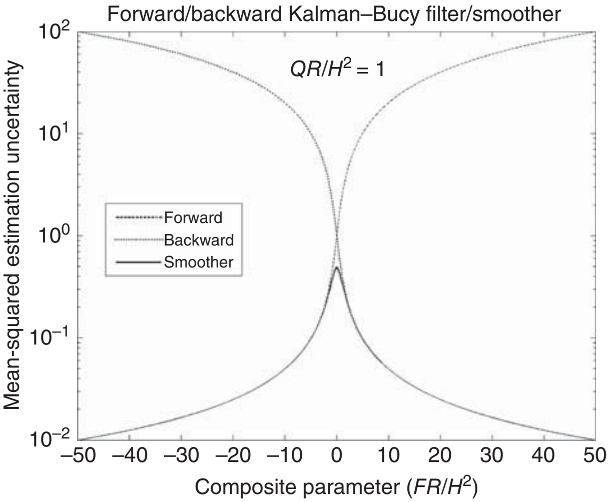

Figure 6.3 Dependence of Kalman–Bucy filter and smoother performance on FR/H^2.

$$P_{[s]} = \frac{\left(\frac{QR}{H^2}\right)}{2\sqrt{\left(\frac{FR}{H^2}\right)^2 + \left(\frac{QR}{H^2}\right)}}. \tag{6.6}$$

All these formulas depend on just two parameter expressions:

$$\left(\frac{FR}{H^2}\right) \text{ and } \left(\frac{QR}{H^2}\right),$$

meaning that mean-squared filter and smoother performance can be plotted as a function of these two parameter expressions.

Results Figures 6.3 and 6.4 are plots of infinite-interval smoothing performance relative to filtering performance as functions of the parameter expressions FR/H^2 and QR/H^2, using the formulas derived above.

Influence of the Sign of F on Relative Performance The sign of F makes a big difference in the benefits of smoothing over filtering. The parameters R, Q, and H^2 are all positive, but F can change sign. The influence of the sign and magnitude of F on filtering and smoothing performance is plotted in Figure 6.3, which shows the steady-state variance of forward and backward filtering and smoothing as a function of FR/H^2, with the other parameter $QR/H^2 = 1$. The forward filter is stable for $F < 0$, and the backward filter is stable for $F > 0$. For both filters, performance in the stable direction is quite close to smoother performance (solid line), but filter performance for the unstable direction is much worse than smoothing. In essence, smoothing provides a significant improvement over filtering when the backward filter performance is much better than the forward filter performance. For the range of values of FR/H^2 shown (±50), the variance of smoothing uncertainty is as much as 10,000 times smaller than that for filtering.

Influence of FR/H^2 and QR/H^2 over Large Dynamic Ranges This is shown with more detail in Figure 6.4 (a) and (b), which are 3D log–log–log plots of the "smoothing improvement ratio" $P_{[s]}/P_{[f]}$, as functions of both composite parameters, (FR/H^2) and (QR/H^2), ranging over ±4 orders of magnitude.

Unstable Systems For $F > 0$ (Figure 6.4(a)), the smoothing improvement ratio $P_{[s]}/P_{[f]} \rightarrow 0$ (arbitrarily better smoothing than filtering) as $(FR/H^2) \rightarrow -\infty$, due principally to degrading filter performance, not improving smoother performance. The improvement of smoothing over filtering also improves significantly as $(QR/H^2) \rightarrow 0$. In the left corner of the plot, smoothing variance is more than 10^{120} times smaller than filtering variance. These results are different from those of Moore and Teo [4], which assumed the dynamic system to have bounded signal power.

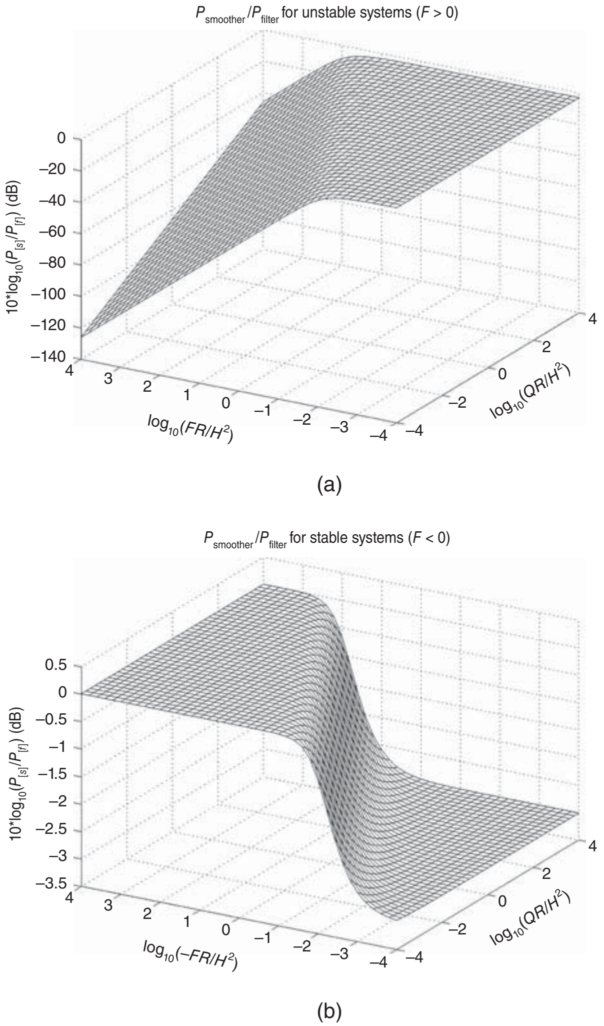

Figure 6.4 Smoothing improvement ratio versus FR/H^2 and QR/H^2. (a) Unstable system and (b) stable system.

Random Walk Conditions The smoothing improvement ratio $P_{[s]}/P_{[f]} \to 1/2$ (-3 dB) as $(FR/H^2) \to 0$, the random walk model. That is, the improvement of smoothing over filtering for a *random walk* is a factor of two decrease in mean-squared estimation uncertainty—or a factor of $\sqrt{2}$ decrease in root-mean-square (RMS) uncertainty.

Stable Systems For $F < 0$ (Figure 6.4 (b)), the smoothing improvement ratio $P_{[s]}/P_{[f]} \to 1/2$ as $(FR/H^2) \to 0$, the random walk model. For $F < 0$, $P_{[s]}/P_{[f]} \to 1/2$ as $(QR/H^2) \to +\infty$, as well. That is, a factor of 2 improvement in mean-squared smoothing uncertainty over filtering uncertainty is the best one can hope for when $F < 0$. This agrees with the results of Moore and Teo [4].

These plots may provide some insight into how smoothing and filtering performance depend on the parameters of the dynamic system. In more complex models, with matrix-valued parameters F, Q, H, and R, the matrix F can have both positive and negative characteristic values. In that case, the dynamic system model can be stable in some subspaces of state space and unstable in others, and the dependence of filter covariance $P_{[f]}$ on F can be much more complicated.

6.2.1.3 Finite Interval Case In this case, the measurements $z(t)$ are confined to a finite interval $t_{\text{start}} \leq t \leq t_{\text{end}}$. This introduces some additional "end effects" on filter and smoother performance. We derive here some formulas for the scalar case ($n = 1$), although the same methods apply to the general case.

This requires the transient solution (as opposed to the steady-state solution) of the Riccati differential equation. For the forward filter, this is

$$P_{[f]}(t) = A_{[f]}(t)B_{[f]}^{-1}(t) \tag{6.7}$$

$$\begin{bmatrix} A_{[f]}(t) \\ B_{[f]}(t) \end{bmatrix} = \exp\left(\left(t - t_{\text{start}}\right) \begin{bmatrix} F & Q \\ H^2/R & -F \end{bmatrix} \right) \begin{bmatrix} A_{[f]}(t_{\text{start}}) \\ B_{[f]}(t_{\text{start}}) \end{bmatrix} \tag{6.8}$$

for initial conditions $P_{[f]}(t_{\text{start}}) = A_{[f]}(t_{\text{start}})B_{[f]}^{-1}(t_{\text{start}})$ at the beginning of the finite interval. Setting $B_{[f]}(t_{\text{start}}) = 0$ and $A_{[f]}(t_{\text{start}}) = 1$ is equivalent to assuming no initial information about the estimate. In that case, the solution will be

$$\tau = \frac{R}{\sqrt{R(F^2R + H^2Q)}} \tag{6.9}$$

$$P_{[f]}(t) = \frac{\sqrt{R(F^2R + H^2Q)}}{H^2} \frac{(e^{(t-t_{\text{start}})/\tau} + e^{-(t-t_{\text{start}})/\tau})}{(e^{(t-t_{\text{start}})/\tau} - e^{-(t-t_{\text{start}})/\tau})} + \frac{FR}{H^2}. \tag{6.10}$$

For the backward filter, this Riccati equation solution is

$$P_{[b]}(t) = A_{[b]}(t)B_{[b]}^{-1}(t) \tag{6.11}$$

$$\begin{bmatrix} A_{[b]}(t) \\ B_{[b]}(t) \end{bmatrix} = \exp\left(\left(t - t_{\text{end}}\right) \begin{bmatrix} -F & Q \\ H^2/R & F \end{bmatrix} \right) \begin{bmatrix} A_{[b]}(t_{\text{end}}) \\ B_{[b]}(t_{\text{end}}) \end{bmatrix}, \tag{6.12}$$

with the sign of F reversed. As for the forward filter, setting $B_{[b]}(t_{\text{end}}) = 0$ and $A_{[b]}(t_{\text{end}}) = 1$ is equivalent to assuming no initial information about the estimate at the end of the interval. In that case, the solution will be

$$P_{[b]}(t) = \frac{\sqrt{R(F^2R + H^2Q)}}{H^2} \frac{(e^{(t_{\text{end}}-t)/\tau} + e^{-(t_{\text{end}}-t)/\tau})}{(e^{(t_{\text{end}}-t)/\tau} - e^{-(t_{\text{end}}-t)/\tau})} - \frac{FR}{H^2}. \tag{6.13}$$

The plot in Figure 6.5 was generated using these formulas in the MATLAB® m-file `FiniteFIS.m` on the companion Wiley web site. Performance is flat in the mid-section

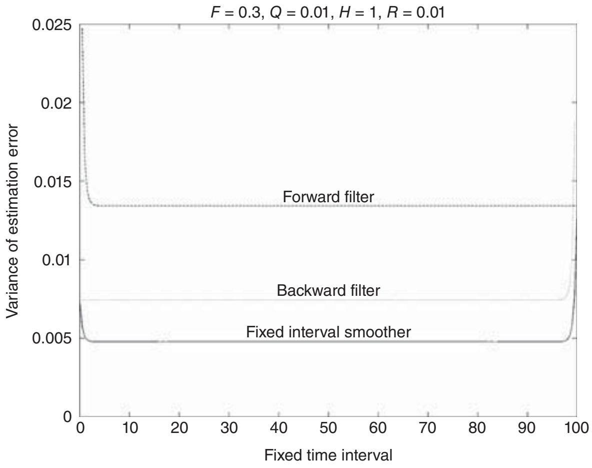

Figure 6.5 Fixed-interval smoothing on finite interval.

because the model is time invariant. The plot shows the "end effects" at both the ends of the interval, where the smoothing variance kicks up when one of its filters has no information.

You can modify the model parameters in the m-file `FiniteFIS.m` to see how the changes affect smoother and filter performance.

6.2.2 Three-pass Fixed-interval Smoothing

This is perhaps the oldest of the smoothing methods based on the Kalman filter. It follows the same game plan as the smoother discussed on Section 6.2.1.3, except that it operates in discrete time and it generates the filtered and smoothed estimates as well as their respective covariances of estimation uncertainty.

The general idea is illustrated in Figure 6.6, in which the zigzag paths represent filter outputs of the estimated state vector $\hat{x}$ and its computed covariance matrix P over the discrete time interval from t_1 to t_N. The plot shows the outputs of two such filters, one running forward in time, starting at t_1 on the left and moving to the right, and the other running backward in time, starting at t_N on the right and moving toward the left. At each filter time step (running either forward or backward) the output estimate of each filter represents its best estimate based on all the measurements it has processed up to that time. When the forward and backward filters meet, all the measurements have been used up to produce two estimates. One is based on all the measurements to the left and the other is based on all measurements to the right. If these two independent estimates can be combined optimally, then the resulting *smoothed estimate* will be based on all the measurements taken throughout the entire fixed interval.

The implementation problem is to do this at every discrete time t_k in the entire fixed interval. It requires three passes through the measurements and data derived therefrom:

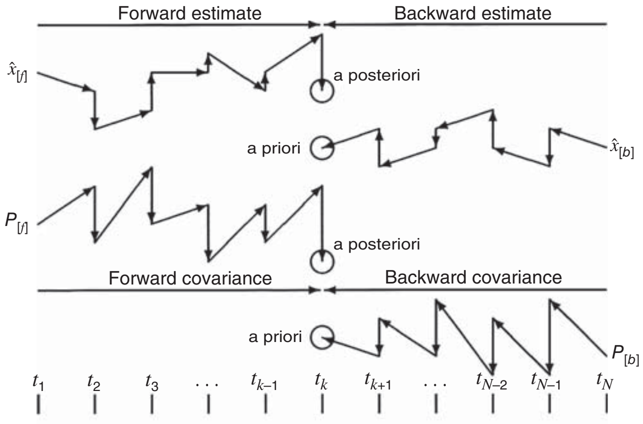

Figure 6.6 Forward and backward filter outputs.

1. A complete filter pass in the forward direction (i.e., with measurement time increasing), saving the values of the a posteriori estimate $\hat{x}_{[f],k(+)}$ and associated covariance of estimation uncertainty $P_{[f],k(+)}$ (the subscript "$[f]$" stands for "forward"). The values of the state-transition matrices Φ_k can also be saved on this pass.
2. A complete filter pass in the backward direction (time decreasing), saving the a priori estimates and associated covariance of estimation uncertainties,

$$\hat{x}_{[b],k-1(-)} = \Phi^{-1}_{[f],k-1}\hat{x}_{[b],k(+)} \,\text{(predictor)} \tag{6.14}$$

$$P_{[b],k-1(-)} = \Phi^{-1}_{[f],k-1}P_{[b],k(+)}\Phi^{-\mathrm{T}}_{[f],k-1} + Q_k \tag{6.15}$$

   (the subscript "$[b]$" is for "backward"), using the inverses $\Phi^{-1}_{[f],k-1}$ of the state-transition matrices from the forward pass (analogous to changing the sign of F in continuous time). The Kalman filter corrector step must also implemented, although the resulting *a posteriori* estimates and covariances are not used in the third (smoother) pass.
3. A third, smoother pass (which can actually be included with the second pass) combining the forward and backward data to obtain the smoothed estimate:

$$\hat{x}_{[s],k(+)} = \hat{x}_{[f],k(+)} + \underbrace{P_{[f],k(+)}[P_{[f],k(+)} + P_{[b],k(-)}]^{-1}}_{\overline{K}}[\hat{x}_{[b],k(-)} - \hat{x}_{[f],k(+)}], \tag{6.16}$$

   where $\hat{x}_{s,k}(+)$ is the smoothed estimate.

   One can also compute the covariance of smoother uncertainty,

$$P_{[s],k} = \overline{K}P_{[b],k(-)}. \tag{6.17}$$

It is not required in the smoother operation, but it is useful for understanding how good the smoothed estimate is supposed to be.

6.2.2.1 The smoother Pass is Not Recursive That is, the variables computed at each time t_k depend only on results saved from the two filter passes, and not on smoother-derived values at adjoining times t_{k-1} or t_{k+1}. The covariance matrix $P_{s,k(+)}$ of the smoothed estimate can also be computed (without recursion) as an indicator of expected smoother performance, although it is not used in obtaining the smoothed estimate values.

It is important to combine the a priori data from one direction with the a posteriori data from the other direction, as illustrated in Figure 6.6. This is to make sure that the forward and backward estimates at each discrete time in the smoothing operation are truly *independent,* in that they do not depend on any common measurements.

6.2.2.2 Optimal Combination of the Two Filter Outputs Equation 6.16 has the form of the Kalman filter observational update (corrector) for combining two statistically independent estimates using their associated covariances of uncertainty, where

$\hat{x}_{[f],k(+)}$ is the effective a priori estimate,

$P_{[f],k(+)}$ is its covariance of uncertainty,

$\hat{x}_{[b],k(-)}$ is the effective "measurement" (independent of $\hat{x}_{[f],k(+)}$), with $H = I$ (the identity matrix) its effective measurement sensitivity matrix,

$P_{[b],k(-)}$ is its analogous "measurement noise" covariance (i.e., its covariance of uncertainty),

$\overline{K} = P_{[f],k(+)}[P_{[f],k(+)} + P_{[b],k(-)}]^{-1}$ is the effective Kalman gain, and

$\hat{x}_{[s],k(+)}$ is the resulting optimal smoother estimate.

Computational issues Equation 6.16 requires the inverse of the covariance matrix P from the forward pass, and Equations 6.14 and 6.15 require the inverse of the state-transition matrix Φ from the forward pass. The latter (i.e., computing Φ^{-1}) is not a serious problem for time-invariant problems, because Φ would have to be computed and inverted only once. Inversion of the covariance matrix is an issue that has been addressed by developing alternative "information smoothing" methods based on the information matrix (inverse of P), rather than the covariance matrix P.

The MATLAB m-file `sim3pass.m` on the companion Wiley web site generates a trajectory for a scalar linear time-invariant model driven by simulated noise for measurement noise and dynamic disturbance noise, applies a three-pass fixed-interval smoother to the resulting measurement sequence, and plots the resulting estimates along with the true (simulated) values and computed variances of estimation uncertainty. A sample output is shown in Figure 6.7. As an empirical measure of smoothing performance relative to the performance of the two filter, the RMS estimation error, computed as the deviation from the "true" simulated trajectory, is calculated for the forward filter, backward filter and smoother each time the program is run. However, this smoother is included here primarily to explain the theoretical basis for

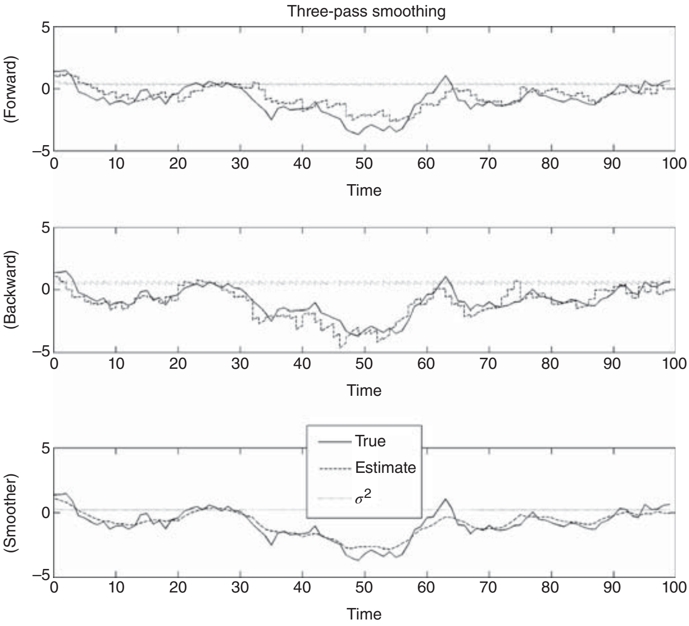

Figure 6.7 Simulated three-pass fixed-interval smoother.

optimal smoothing, to show that a relatively straightforward implementation method exists, and to derive the following example of smoothing performance for very simple models.

6.2.3 Rauch–Tung–Striebel (RIS) Two-pass Smoother

This fixed-interval two-pass implementation is the fastest fixed-interval smoother [5], and it has been used quite successfully for decades. The algorithm was published by Herbert E. Rauch, F. Tung, and Charlotte T. Striebel in 1965 [6], and the implementation formulas have had some tweaking since then. The first (forward) pass uses a Kalman filter but saves the intermediate results $\hat{x}_{k(-)}, \hat{x}_{k(+)}, P_{k(-)}$, and $P_{k(+)}$ at each measurement time t_k. The second pass runs backward in time in a sequence from the time t_N of the last measurement, computing the smoothed state estimate from the intermediate results stored on the forward pass. The smoothed estimate (designated by the subscript $[s]$) is initialized with the value

$$\hat{x}_{[s]N} = \hat{x}_{N(+)}, \tag{6.18}$$

then computed recursively by the formulas

$$\hat{x}_{[s]k} = \hat{x}_{k(+)} + A_k(\hat{x}_{[s]k+1} - \hat{x}_{k+1(-)}), \tag{6.19}$$

$$A_k = P_{k(+)}\Phi_k^{\mathrm{T}} P_{k+1(-)}^{-1}. \tag{6.20}$$

6.2.3.1 Computational Complexity Issues Inverting the filter covariance matrices P_k is the major added computational burden. It could also compromise numerical stability when the conditioning of the P_k for inversion is questionable.

6.2.3.2 Performance The covariance of uncertainty of the smoothed estimate can also be computed on the second pass:

$$P_{[s]k} = P_{k(+)} + A_k(P_{[s]k+1} - P_{k+1(-)})A_k^{\mathrm{T}}, \tag{6.21}$$

although this is not a necessary part of the smoother implementation. It can be computed if the estimated smoother performance is of some interest.

The MATLAB m-file `RTSvsKF.m`, described in Appendix A, demonstrates (using simulated data) the performance of this smoother relative to that of the Kalman filter.

6.3 FIXED-LAG SMOOTHING

6.3.1 Stability Problems of Early Methods

Rauch [7] published the earliest fixed-lag smoothing method based on the Kalman filter model in 1963. Kailath and Frost [8] later discovered that the Kalman filter and the Rauch fixed-lag smoother are an "adjoint pair," and Kelly and Anderson [9] showed that this implies that the Rauch fixed-lag smoother is not asymptotically stable. Many of the early fixed-lag smoothing methods were plagued by similar instability problems.

Biswas and Mahalanabis [10] first introduced the idea of augmenting the state vector to recast the smoothing problem as a filtering problem. Biswas and Mahalanabis also proved stability of their fixed-lag smoother implementation [11] and determined its computational requirements [12].

We present in Section 6.3.3 an augmented state vector implementation from Biswas and Mahalanabis [10], Premier and Vacroux [13], and Moore [14], which has become standard. Prasad and Mahalanabis [15] and Tam and Moore [16] have also developed stable fixed-lag smoothers in continuous time for analog implementations in communications signal processing.

We first introduce a simpler (but less stable) example for the purpose of analyzing how "generic" fixed-lag smoothing performs relative to filtering and how relative performance depends on the lag time and other parameters of the application.

6.3.2 Performance Analysis

6.3.2.1 Analytical Model in Continuous Time We can use the same scalar Kalman–Bucy filter model of Equations 6.1 and 6.2 from Section 6.2.1 to characterize the relative improvement of fixed-lag smoothing over filtering as a function of lag time and the parameters of the filter model.

How this model is used for the fixed-lag smoother is illustrated in Figure 6.8. At any time t, the effect of the lag time Δt_{lag} on the estimate at time t, given measurements

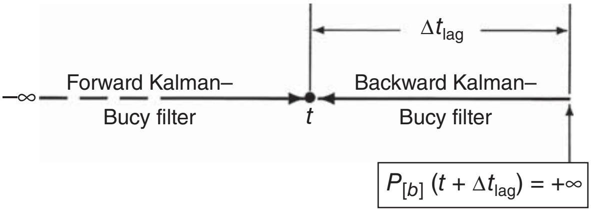

Figure 6.8 Kalman–Bucy fixed-lag smoother model.

out to time $t + \Delta t_{\text{lag}}$, is modeled by an additional filter operating backward from time $t + \Delta t_{\text{lag}}$ to time t, starting with no information at all at time $t + \Delta t_{\text{lag}}$ (i.e., $P(t + \Delta t_{\text{lag}}) = +\infty$). The improvement of fixed-lag smoothing over filtering can then be evaluated as it was in Section 6.2.1, except that this example has as an additional parameter the lag time Δt_{lag}.

6.3.2.2 Forward Kalman–Bucy Filter Performance In Section 6.2.1, the Kalman–Bucy forward filter performance at steady state ($\dot{P}_{[f]} = 0$) was shown to be

$$P_{[f]}(t) = \sqrt{\left(\frac{FR}{H^2}\right)^2 + \left(\frac{QR}{H^2}\right)} + \left(\frac{FR}{H^2}\right), \tag{6.22}$$

which is a formula for filtering performance as a function of the two composite parameters (FR/H^2) and (QR/H^2).

6.3.2.3 Backward Kalman–Bucy Filter Performance For the nonsteady-state problem of the backward filter, we can use the linearized Riccati equation solution (originally published by Jacopo Riccati in 1724 [17]) from Section 5.8.3. The solution for the backward filter covariance $P_{[b]}(t)$ at time t, having started at time $t + \Delta t_{\text{lag}}$ with initial value $P_{[b]}(t + \Delta t_{\text{lag}}) = +\infty$ (i.e., with no initial information, modeled by setting $P(0) = 1$ and the vector component below it is equal to 0 in Equation 5.70) is

$$P_{[b]}(t) = \sqrt{\left(\frac{FR}{H^2}\right)^2 + \left(\frac{QR}{H^2}\right)}\frac{e^{\rho} + e^{-\rho}}{e^{\rho} - e^{-\rho}} - \left(\frac{FR}{H^2}\right) \tag{6.23}$$

$$\rho = \left(\frac{H^2 \Delta t_{\text{lag}}}{R}\right)\sqrt{\left(\frac{FR}{H^2}\right)^2 + \left(\frac{QR}{H^2}\right)}, \tag{6.24}$$

which introduces a third composite parameter, $(H^2 \Delta t_{\text{lag}}/R)$.

6.3.2.4 Kalman–Bucy Fixed-lag Smoother Performance The formula for two-filter smoothing (from Section 6.2.1) is

$$P_{[s]}(t) = \frac{P_{[b]}(t)P_{[f]}(t)}{P_{[f]}(t) + P_{[b]}(t)},$$

for the values of $P_{[f]}(t)$ and $P_{[b]}(t)$ derived above as functions of the three composite parameters (FR/H^2), (QR/H^2), and $(H^2 \Delta t_{\text{lag}}/R)$.

6.3.2.5 Improvement of Fixed-lag Smoothing over Filtering The smoothing improvement ratio over filtering,

$$\frac{P_{[s]}(t)}{P_{[f]}(t)} = \frac{P_{[b]}(t)}{P_{[f]}(t) + P_{[b]}(t)},$$

now depends on three composite parameters: (FR/H^2), (QR/H^2), and $(H^2\ \Delta t_{\text{lag}}/R)$. It cannot be plotted on a paper as a function of three parameters, as the infinite-interval smoother performance was plotted as a function of two parameters in Section 6.2.1, but it can still be "plotted" using time as the third dimension. The MATLAB m-file `FLSmovies.m` creates two short video clips of the smoothing improvement ratio $P[s]/P[f]$ as a function of QR/H^2 and FR/H^2 as the lag time parameter $\Delta t_{\text{lag}}\ H^2/R$ varies from frame to frame from 10^{-6} to about 0.06. The created file `FixedLagU.avi` is for $F > 0$. The created file `FixedLagS.avi` is for $F < 0$. These video clips show how fixed-lag smoothing improvement over filtering varies as a function of the three composite parameters (FR/H^2), (QR/H^2), and $(H^2\ \Delta t_{\text{lag}}/R)$. These files are in audio video interleave format for Windows PCs, and each requires more than a megabyte of storage.

The asymptotic values for fixed-lag smoother performance are

$$\begin{array}{llllrll}
\text{as} & \Delta t_{\text{lag}} & \to & 0, & \frac{P_{[s]}(t)}{P_{[f]}(t)} & \to & 1 \text{ (no improvement);} \\
\text{as} & \Delta t_{\text{lag}} & \to & +\infty, & \rho & \to & +\infty, \\
 & & & & e^{-\rho} & \to & 0, \\
 & & & & e^{\rho} & \to & +\infty, \\
 & & & & \frac{e^{\rho}+e^{-\rho}}{e^{\rho}-e^{-\rho}} & \to & 1, \\
 & & & & \frac{P_{[s]}(t)}{P_{[f]}(t)} & \to & \dfrac{\left(\frac{QR}{H^2}\right)}{2\left(\frac{FR}{H^2}\right)^2+2\left(\frac{QR}{H^2}\right)+2\left(\frac{FR}{H^2}\right)\sqrt{\left(\frac{FR}{H^2}\right)^2+\left(\frac{QR}{H^2}\right)}},
\end{array}$$

the improvement ratio of the fixed infinite-interval Kalman–Bucy smoother of Section 6.2.1

6.3.3 Biswas–Mahalanabis Fixed-lag Smoother (BMFLS)

The earliest implementations for a fixed-lag smoother were mathematically correct but were found to be poorly condition for numerical implementation. We demonstrate here the relative numerical stability of an implementation published by Moore [14], based on an approach by Premier and Vacroux [13], which is the "state augmentation" filtering approach published earlier by Biswas and Mahalanabis [10]. It is essentially the Kalman filter using an augmented state vector made up from the successive values of the original system state vector over a discrete-time window of fixed width, as illustrated in Figure 6.9. If

$$\Delta t_{\text{lag}} = \ell \Delta t$$

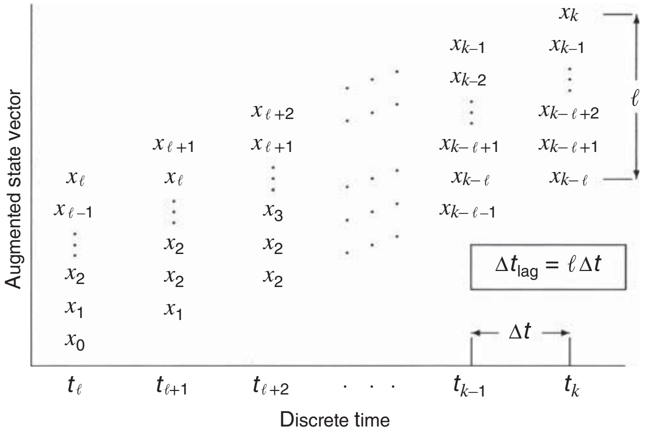

Figure 6.9 Biswas–Mahalanabis augmented state vector.

is the time lag, which is ℓ discrete time steps, then the augmented state vector at time t_k is of length $n(\ell + 1)$, with (ℓ_1) n-dimensional subvectors x_k, x_{k-1}, x_{k-2}, … , $x_{k-\ell}$—as shown in Figure 6.9.

6.3.3.1 State Vector In order to distinguish between the Kalman filter state vector and the BMFLS fixed-lag smoother state vector, we will use the following notations:

$\mathbf{x}_{k,\text{KF}}$ to denote the true value of the state vector of Kalman filter at the k^{th} epoch in discrete time.

$\hat{\mathbf{x}}_{k,\text{KF}}$ to denote the estimated value of the state vector of Kalman filter at the kth epoch in discrete time.

$\mathbf{x}_{k,\text{BMFLS}}$ to denote the true value of the state vector of the Biswas–Mahalanabis fixed-lag smoother (BMFLS) at the kth epoch in discrete time.

$\hat{\mathbf{x}}_{k,\text{BMFLS}}$ to denote the estimated value of the state vector of the BMFLS at the kth epoch in discrete time.

The BMFLS uses $\ell + 1$ lagged values $\mathbf{x}_{k,\text{KF}}$, $\mathbf{x}_{k-1,\text{KF}}$, $\mathbf{x}_{k-2,\text{KF}}$, … , $\mathbf{x}_{k-\ell,\text{KF}}$ of the Kalman filter state vector $\mathbf{x}_{\text{KF}}$, stacked into a single vector of dimension $n\ (\ell + 1) \times 1$, where n is the number of state variables in the Kalman filter model and ℓ is the fixed number of lag steps in the fixed-lag smoother. That is, the BMFLS state vector

$$\mathbf{x}_{k,\text{BMFLS}} \overset{\text{def}}{=} \begin{bmatrix} \mathbf{x}_{k,\text{KF}} \\ \mathbf{x}_{k-1,\text{KF}} \\ \mathbf{x}_{k-2,\text{KF}} \\ \vdots \\ \mathbf{x}_{k-\ell,\text{KF}} \end{bmatrix}. \tag{6.25}$$

The BMFLS is the Kalman filter using this augmented state vector. If this state vector is estimated using the same sequence of measurements $\{\mathbf{z}_k\}$ as the conventional Kalman filter, then each augmented subvector of the resulting estimate $\hat{\mathbf{x}}_{k,\mathrm{BMFLS}}$ corresponding to $\mathbf{x}_{k-\ell,\mathrm{KF}}$ represents the smoothed estimate of $\mathbf{x}_{k-\ell}$ using the measurements $\mathbf{z}_j$ for $j \leq k$. That is, the resulting BMFLS estimates the *smoothed* values of

$$\mathbf{x}_{k-\ell,\mathrm{KF}},\ \mathbf{x}_{k-\ell+1,\mathrm{KF}},\ \mathbf{x}_{k-\ell+2,\mathrm{KF}},\ \cdots\ ,\mathbf{x}_{k-1,\mathrm{KF}}$$

plus the *filtered* value $\mathbf{x}_{k,\mathrm{KF}}$, given the measurements

$$\cdots\ ,\ \mathbf{z}_{k-\ell},\ \mathbf{z}_{k-\ell+1},\ \mathbf{z}_{k-\ell+2},\ \cdots\ ,\mathbf{z}_k.$$

In other words, the estimated state vector of the BMFLS will be equivalent to

$$\hat{\mathbf{x}}_{k,\mathrm{BMFLS}} \equiv \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{k-1|k} \\ \hat{\mathbf{x}}_{k-2|k} \\ \vdots \\ \hat{\mathbf{x}}_{k-\ell|k} \end{bmatrix}, \tag{6.26}$$

where $\hat{\mathbf{x}}_{k-\ell|k}$ is the notation for the smoothed estimate of $\mathbf{x}_{k-\ell}$ given all measurements up to $\mathbf{z}_k$.

6.3.3.2 State-Transition Matrix If the Kalman filter model has state-transition matrices $\mathbf{\Phi}_{k,\mathrm{KF}}$, then the BMFLS state vector has the property that

$$\mathbf{x}_{k+1,\mathrm{BMFLS}} = \begin{bmatrix} \mathbf{x}_{k+1,\mathrm{KF}} \\ \mathbf{x}_{k,\mathrm{KF}} \\ \mathbf{x}_{k-1,\mathrm{KF}} \\ \vdots \\ \mathbf{x}_{k-\ell+2,\mathrm{KF}} \\ \mathbf{x}_{k-\ell+1,\mathrm{KF}} \end{bmatrix} \tag{6.27}$$

$$= \begin{bmatrix} \mathbf{\Phi}_{k,\mathrm{KF}} & 0 & 0 & \cdots & 0 & 0 \\ \mathbf{I} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{I} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \mathbf{I} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{I} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k,\mathrm{KF}} \\ \mathbf{x}_{k-1,\mathrm{KF}} \\ \mathbf{x}_{k-2,\mathrm{KF}} \\ \vdots \\ \mathbf{x}_{k-\ell+1,\mathrm{KF}} \\ \mathbf{x}_{k-\ell,\mathrm{KF}} \end{bmatrix}. \tag{6.28}$$

That is, the equivalent state-transition matrix for the BMFLS is

$$\mathbf{\Phi}_{k,\mathrm{BMFLS}} = \begin{bmatrix} \mathbf{\Phi}_{k,\mathrm{KF}} & 0 & 0 & \cdots & 0 & 0 \\ \mathbf{I} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{I} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \mathbf{I} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{I} & 0 \end{bmatrix}. \tag{6.29}$$

6.3.3.3 Process Noise Covariance If $\mathbf{Q}_{k,\mathrm{KF}}$ is the process noise covariance for the Kalman filter model, then

$$\mathbf{Q}_{k,\mathrm{BMFLS}} = \begin{bmatrix} \mathbf{Q}_{k,\mathrm{KF}} & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{6.30}$$

6.3.3.4 Measurement Sensitivity Matrix Because the measurement $\mathbf{z}_k$ is sensitive only to the subvector $\mathbf{x}_{k,\mathrm{KF}}$, the appropriate measurement sensitivity matrix for the BMFLS will be

$$\mathbf{H}_{k,\mathrm{BMFLS}} = \begin{bmatrix} \mathbf{H}_{k,\mathrm{KF}} & 0 & 0 & \cdots & 0 \end{bmatrix}, \tag{6.31}$$

where $\mathbf{H}_{k,\mathrm{KF}}$ is the measurement sensitivity matrix from the Kalman filter model.

6.3.3.5 Measurement Noise Covariance Because the measurements for the BMFLS are the same as those for the Kalman filter, the equivalent measurement noise covariance matrix will be

$$\mathbf{R}_{k,\mathrm{BMFLS}} = \mathbf{R}_{k,\mathrm{KF}}, \tag{6.32}$$

the same as for the Kalman filter.

6.3.3.6 Implementation Equations

Start-up Transition The BMFLS is actually a filter, but with state augmentation for lagged estimates. During start-up, it must transition from a filter with no augmentation, through augmentation with increasing numbers of lagged estimates, until it reaches the specified fixed number of lags.

At the initial time t_0, this fixed-lag smoother starts with a single initial estimate $\hat{x}_0$ of the system state vector and an initial value P_0 of its covariance of uncertainty.

Let n denote the length of $\hat{x}_0$, so that P_0 will have dimensions $n \times n$, and let ℓ denote the ultimate number of time lags (discrete-time steps of delay) desired for fixed-lag smoothing.

State Vector Augmentation As the first $\ell + 1$ measurements $z_1, z_2, z_3, \ldots, z_{\ell+1}$ come in, the smoother incrementally augments its state vector at each time step to build up to $\ell + 1$ time-lagged state vector values in its state vector.

Throughout the smoothing process, the topmost n-component subvector of this augmented state vector will always equal the filtered estimate, and subsequent n-dimensional subvectors are the smoothed estimates with increasing numbers of time lags. The last n components of the smoother state vector will be the smoothed estimate with the current longest lag.

The length of the smoothed estimate state vector increases by n components for the first $\ell + 1$ measurement times t_k, $1 \leq k \leq \ell + 1$. At time $t_{\ell+1}$, the specified fixed lag time is reached, the incremental augmentation of the state vector ceases, and the smoother state vector length remains steady at $n(\ell + 1)$ thereafter.

The start-up sequence of state vector augmentation is illustrated in Figure 6.10, where each subvector of length n in the smoother state vector is represented by a separate symbol. Note that the topmost subvector is always the filter estimate of the current state.

If the smoother starts receiving measurements at discrete time t_1, then the length of the smoother state vector will be

$$\text{length }[\hat{x}_{[s]}(t_k)] = \begin{cases} nk, & 1 \le k \le \ell + 1 \\ n(\ell + 1), & k \ge \ell + 1, \end{cases} \tag{6.33}$$

and the associated covariance matrix of smoother state vector uncertainty will grow as $(nk \times nk)$ until $k = \ell + 1$.

Covariance Matrix Augmentation The corresponding buildup of smoother covariance matrices is shown in Table 6.1—until the size of the matrix stabilizes at $t_{\ell+1}$.

As with the state vector, there are two values of the covariance matrix at each time step t_k: the a priori value $P_{[s](-)}(t_k)$ and the *a posteriori* value $P_{[s](+)}(t_k)$. A reshuffling of the subblocks of the covariance matrix first occurs with each a priori value, with an upper-left subblock of the previous a posteriori value becoming the lower-right subblock of the succeeding a priori value.

This sort of matrix operation—a shifting of the matrix down along its own diagonal—is known as *displacement*, a concept developed by Professor Thomas Kailath at the Stanford University, who discovered that the rank of the difference between the original and displacement matrices (called *displacement rank*) characterizes how certain algorithmic operations on matrices can be performed in parallel.

Steady State Given the model parameters as defined above, the implementation equations for the BMFLS are the same as those for the conventional Kalman filter:

$$\begin{aligned}
\mathbf{K}_{k,\text{BMFLS}} &= \mathbf{P}_{k,\text{BMFLS}}\mathbf{H}^{\text{T}}_{k,\text{BMFLS}}(\mathbf{H}_{k,\text{BMFLS}}\mathbf{P}_{k,\text{BMFLS}}\mathbf{H}^{\text{T}}_{k,\text{BMFLS}} + \mathbf{R}_k)^{-1} \quad \text{Gain matrix} \\
\hat{\mathbf{x}}_{k,\text{BMFLS}} &\leftarrow \hat{\mathbf{x}}_{k,\text{BMFLS}} + \mathbf{K}_{k,\text{BMFLS}}(\mathbf{z}_k - \mathbf{H}_{k,\text{BMFLS}}\hat{\mathbf{x}}_{k,\text{BMFLS}}) \quad \text{Observational state update} \\
\mathbf{P}_{k,\text{BMFLS}} &\leftarrow \mathbf{P}_{k,\text{BMFLS}} - \mathbf{K}_{k,\text{BMFLS}}\mathbf{H}_{k,\text{BMFLS}}\mathbf{P}_{k,\text{BMFLS}} \quad \text{Observational covariance update} \\
\hat{\mathbf{x}}_{k+1,\text{BMFLS}} &\leftarrow \mathbf{\Phi}_{k,\text{BMFLS}}\hat{\mathbf{x}}_{k,\text{BMFLS}} \quad \text{Temporal state update} \\
\hat{\mathbf{P}}_{k+1,\text{BMFLS}} &\leftarrow \mathbf{\Phi}_{k,\text{BMFLS}}\hat{\mathbf{P}}_{k,\text{BMFLS}}\mathbf{\Phi}^{\text{T}}_{k,\text{BMFLS}} + \mathbf{Q}_{k,\text{BMFLS}} \quad \text{Temporal covariance update,}
\end{aligned}$$

where $\mathbf{P}_{k,\text{BMFLS}}$ is the covariance matrix of smoother estimation uncertainty. In the following application to exponentially correlated noise, an initial value for $\mathbf{P}_{k,\text{BMFLS}}$ is determined from the steady-state value of the Riccati equation without measurements.

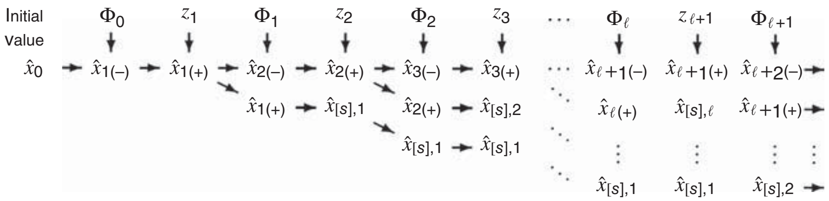

Figure 6.10 Initial state vector transition for BMFLS.

MATLAB Implementation The MATLAB function `BMFLS.m` on the companion Wiley web site implements the complete BMFLS, from the first measurement, through the transition to steady-state state augmentation, and after that.

Example 6.1 (MATLAB Example Using BMFLS.m) Figure 6.11 is a plot of a sample application of `BMFLS.m` using simulated data. The model in this case us for a sinusoidal signal being demodulated and sampled in-phase and in quadrature with phase slip rate ω. The problem is time invariant. The state vector and measurement vector both have two components, and the signal model is

$$\Phi = \exp(-\Delta t/\tau) \begin{bmatrix} \cos(\omega\ \Delta t) & \sin(\omega\ \Delta t) \\ -\sin(\omega\ \Delta t) & \cos(\omega\ \Delta t) \end{bmatrix} \tag{6.34}$$

$$x_k = \Phi x_{k-1} + w_{k-1} \tag{6.35}$$

TABLE 6.1 BMFLS Fixed-Lag Smoother Covariance Transition

<table>
<tr><th>$P_{[s],k}$</th><th>A Priori Value</th><th>A Posteriori Value $\langle$dim.$\rangle$</th></tr>
<tr><td>$P_{[s],1}$</td><td>$P_{[f],1(-)} = \Phi_0 P_0 \Phi_0^{\mathrm{T}} + Q_0$</td><td>$P_{[f],1(+)}\ \langle n \times n \rangle$</td></tr>
<tr><td>$P_{[s],2}$</td><td>$\begin{bmatrix} \Phi_1 P_{[f],1(+)} \Phi_1^{\mathrm{T}} + Q_1 & \Phi_1 P_{[f],1(+)} \\ P_{[f],1(+)} \Phi_1^{\mathrm{T}} & P_{[f],1(+)} \end{bmatrix}$</td><td>$\begin{bmatrix} P_{[f],2(+)} & P_{[s],1,2(+)} \\ P_{[s],1,2(+)}^{\mathrm{T}} & P_{[s],2,2(+)} \end{bmatrix}$</td></tr>
<tr><td>$P_{[s],3}$</td><td>$\begin{bmatrix} \Phi_2 P_{[f],2(+)} \Phi_2^{\mathrm{T}} + Q_2 & \Phi_2 P_{[f],2(+)} & \Phi_2 P_{[s],1,2(+)} \\ P_{[f],2(+)} \Phi_2^{\mathrm{T}} & P_{[f],2(+)} & P_{[s],1,2(+)} \\ P_{[s],1,2(+)}^{\mathrm{T}} \Phi_2^{\mathrm{T}} & P_{[s],1,2(+)}^{\mathrm{T}} & P_{[s],2,2(+)} \end{bmatrix}$</td><td>$\vdots$</td></tr>
<tr><td>$\vdots$</td><td>$\vdots$</td><td>$\vdots$</td></tr>
<tr><td>$P_{[s],\ell}$</td><td>$\vdots$</td><td>$P_{[s](+)}(t_\ell)\ \langle n\ell \times n\ell \rangle$</td></tr>
<tr><td>$P_{[s],\ell+1}$</td><td>$\left[\begin{array}{c|c} \Phi_{\ell-1}\ P_{[f],\ell-1(+)} \Phi_{\ell-1}^{\mathrm{T}} & \cdots \\ \hline \vdots & P_{[s](+)}(t_\ell) \end{array}\right]$</td><td>$P_{[s](+)}(t_{\ell+1})$
$\langle n(\ell+1) \times n(\ell+1) \rangle$</td></tr>
<tr><td>$P_{[s],\ell+2}$</td><td>$\left[\begin{array}{c|c} \Phi_\ell P_{[f],\ell(+)} \Phi_\ell^{\mathrm{T}} & \cdots \\ \hline \cdot & \text{upper-left } \langle n\ell \times n\ell \rangle \\ \cdot & \text{submatrix of} \\ \cdot & P_{[s](+)}(t_{\ell+1}) \end{array}\right]$</td><td>$P_{[s](+)}(t_{\ell+2})$
$\langle n(\ell+1) \times n(\ell+1) \rangle$</td></tr>
</table>

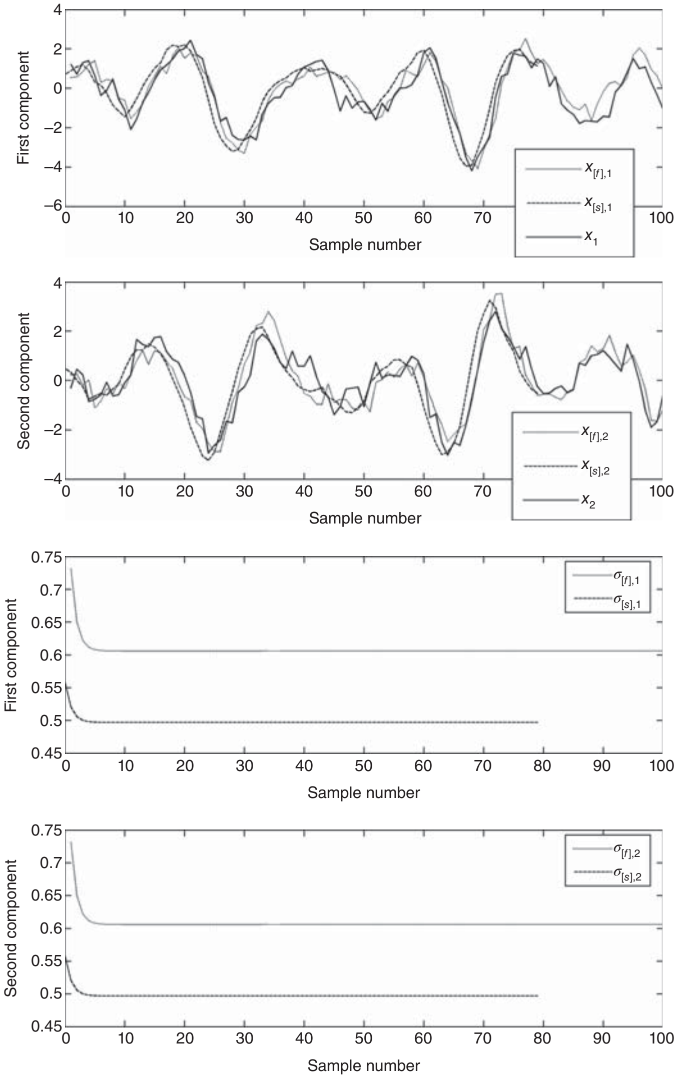

Figure 6.11 Example application of BMFLS.m.

$$z_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k + v_k, \tag{6.36}$$

with other parameters as specified in Problem 6.8. The plots include both phase components, and the estimates and associated RMS uncertainties.

Note that RMS smoothing uncertainty for this example (which is stable) is about 20% less than filtering uncertainty.

Example 6.2 (Algebraic Riccati Equation Solution) The algebraic Riccati equation for the BMFLS with scalar state vector has a relatively regular structure, from which it is possible to obtain a closed-form solution.

The smoother state-transition matrix in this case has the form

$$\Phi_{[s]} = \begin{bmatrix} \phi & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{6.37}$$

$$\phi = \exp\ (\Delta t/\tau), \tag{6.38}$$

where ϕ is a scalar.

Let the a posteriori smoother covariance matrix

$$P_{[s],\ k-1(+)} = \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & \cdots & p_{1,\ell} & p_{1,\ell+1} \\ p_{1,2} & p_{2,2} & p_{2,3} & \cdots & p_{2,\ell} & p_{2,\ell+1} \\ p_{1,3} & p_{2,3} & p_{3,3} & \cdots & p_{3,\ell} & p_{3,\ell+1} \\ p_{1,4} & p_{2,4} & p_{3,4} & \cdots & p_{3,\ell} & p_{3,\ell+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ p_{1,\ell} & p_{2,\ell} & p_{3,\ell} & \cdots & p_{\ell,\ell} & p_{\ell,\ell+1} \\ p_{1,\ell+1} & p_{2,\ell+1} & p_{3,\ell+1} & \cdots & p_{\ell,\ell+1} & p_{\ell+1,\ell+1} \end{bmatrix}, \tag{6.39}$$

so that the matrix product

$$\Phi_{[s]}P_{[s],\ k-1(+)}$$

$$= \begin{bmatrix} \phi p_{1,1} & \phi p_{1,2} & \phi p_{1,3} & \cdots & \phi p_{1,\ell} & \phi p_{1,\ell+1} \\ p_{1,1} & p_{1,2} & p_{1,3} & \cdots & p_{1,\ell} & p_{1,\ell+1} \\ p_{1,2} & p_{2,2} & p_{2,3} & \cdots & p_{2,\ell} & p_{2,\ell+1} \\ p_{1,3} & p_{2,3} & p_{3,3} & \cdots & p_{3,\ell} & p_{3,\ell+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ p_{1,\ell-1} & p_{2,\ell-1} & p_{3,\ell-1} & \cdots & p_{\ell-1,\ell} & p_{\ell-1,\ell+1} \\ p_{1,\ell} & p_{2,\ell} & p_{3,\ell} & \cdots & p_{\ell,\ell} & p_{\ell,\ell+1} \end{bmatrix} \tag{6.40}$$

and the a priori smoother covariance matrix

$$P_{[s],\ k(-)} = \Phi_{[s]}P_{[s],\ k-1(+)}\Phi_{[s]}^{\mathrm{T}} + Q_{[s]} \tag{6.41}$$

$$= \begin{bmatrix} \phi^2 p_{1,1} + q & \phi p_{1,1} & \phi p_{1,2} & \cdots & \phi p_{1,\ell-1} & \phi p_{1,\ell} \\ \phi p_{1,1} & p_{1,1} & p_{1,2} & \cdots & p_{1,\ell-1} & p_{1,\ell} \\ \phi p_{1,2} & p_{1,2} & p_{2,2} & \cdots & p_{2,\ell-1} & p_{2,\ell} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \phi p_{1,\ell-1} & p_{1,\ell-1} & p_{2,\ell-1} & \cdots & p_{\ell-1,\ell-1} & p_{\ell-1,\ell} \\ \phi p_{1,\ell} & p_{1,\ell} & p_{2,\ell} & \cdots & p_{\ell-1,\ell} & p_{\ell,\ell} \end{bmatrix}. \tag{6.42}$$

The measurement sensitivity matrix

$$H_{[s]} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}, \tag{6.43}$$

and the matrix product

$$P_{[s],\,k}(-)H_{[s]}^{\mathrm{T}} \tag{6.44}$$

$$= \begin{bmatrix} \phi^2 p_{1,1} + q \\ \phi p_{1,1} \\ \phi p_{1,2} \\ \vdots \\ \phi p_{1,\ell-1} \\ \phi p_{1,\ell} \end{bmatrix} \tag{6.45}$$

and the innovations variance (a scalar)

$$H_{[s]}P_{[s],\,k}(-)H_{[s]}^{\mathrm{T}} + R_{[s]} \tag{6.46}$$

$$= \phi^2 p_{1,1} + q + r \tag{6.47}$$

so that the matrix product

$$P_{[s],\,k(-)}H_{[s]}^{\mathrm{T}}[H_{[s]}P_{[s],\,k(-)}H_{[s]}^{\mathrm{T}} + R]^{-1}H_{[s]}P_{[s],\,k(-)} = \frac{1}{p_{1,1}\phi^2+q+r}$$

$$\begin{bmatrix} \left(p_{1,1}\phi^2 + q\right)^2 & (p_{1,1}\phi^2 + q)p_{1,1}\phi & (p_{1,1}\phi^2 + q)p_{1,2}\phi & \cdots & (p_{1,1}\phi^2 + q)p_{1,\ell}\phi \\ (p_{1,1}\phi^2 + q)p_{1,1}\phi & {p_{1,1}}^2\phi^2 & p_{1,1}\phi^2 p_{1,2} & \cdots & p_{1,1}\phi^2 p_{1,\ell} \\ (p_{1,1}\phi^2 + q)p_{1,2}\phi & p_{1,1}\phi^2 p_{1,2} & {p_{1,2}}^2\phi^2 & \cdots & p_{1,2}\phi^2 p_{1,\ell} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (p_{1,1}\phi^2 + q)p_{1,\ell}\phi & p_{1,1}\phi^2 p_{1,\ell} & p_{1,2}\phi^2 p_{1,\ell} & \cdots & {p_{1,\ell}}^2\phi^2 \end{bmatrix} \tag{6.48}$$

and the a posteriori smoother covariance

$$P_{[s],\,k(+)} = P_{[s],\,k(-)} - P_{[s],\,k(-)}H_{[s]}^{\mathrm{T}}[H_{[s]}P_{[s],\,k(-)}H_{[s]}^{\mathrm{T}} + R]^{-1}H_{[s]}P_{[s],\,k(-)} \tag{6.49}$$

$$= \begin{bmatrix} \phi^2 p_{1,1} + q & \phi p_{1,1} & \phi p_{1,2} & \cdots & \phi p_{1,\ell} \\ \phi p_{1,1} & p_{1,1} & p_{1,2} & \cdots & p_{1,\ell} \\ \phi p_{1,2} & p_{1,2} & p_{2,2} & \cdots & p_{2,\ell} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi p_{1,\ell} & p_{1,\ell} & p_{2,\ell} & \cdots & p_{\ell,\ell} \end{bmatrix} - \frac{1}{p_{1,1}\phi^2 + q + r}$$

$$\cdot\begin{bmatrix} \left(p_{1,1}\phi^2+q\right)^2 & (p_{1,1}\phi^2+q)p_{1,1}\phi & (p_{1,1}\phi^2+q)p_{1,2}\phi & \cdots & (p_{1,1}\phi^2+q)p_{1,\ell}\phi \\ (p_{1,1}\phi^2+q)p_{1,1}\phi & {p_{1,1}}^2\phi^2 & p_{1,1}\phi^2p_{1,2} & \cdots & p_{1,1}\phi^2p_{1,\ell} \\ (p_{1,1}\phi^2+q)p_{1,2}\phi & p_{1,1}\phi^2p_{1,2} & {p_{1,2}}^2\phi^2 & \cdots & p_{1,2}\phi^2p_{1,\ell} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (p_{1,1}\phi^2+q)p_{1,\ell}\phi & p_{1,1}\phi^2p_{1,\ell} & p_{1,2}\phi^2p_{1,\ell} & \cdots & {p_{1,\ell}}^2\phi^2 \end{bmatrix}. \tag{6.50}$$

At steady state, the a posteriori smoother covariance matrix of Equation 6.50 will equal the a posteriori smoother covariance matrix in Equation 6.39. By equating the matrix elements term by term, one gets the $\ell\ (\ell+1)/2$ equations:

$$p_{1,1} = p_{1,1}\phi^2 + q - \frac{(p_{1,1}\phi^2+q)^2}{p_{1,1}\phi^2+q+r} \tag{6.51}$$

$$p_{1,i+1} = p_{1,i}\phi - \frac{(p_{1,1}\phi^2+q)p_{1,i}\phi}{p_{1,1}\phi^2+q+r} \tag{6.52}$$

$$p_{i+1,i+1} = p_{i,i} - \frac{{p_{1,i}}^2\phi^2}{p_{1,1}\phi^2+q+r} \tag{6.53}$$

$$p_{j+1,j+k+1} = p_{j,j+k} - \frac{p_{1,j}\phi^2p_{1,j+k}}{p_{1,1}\phi^2+q+r}, \tag{6.54}$$

for $1 \le i \le \ell$, $1 \le j \le \ell - 1$, and $1 \le k \le \ell - j$.

These can all be solved recursively, starting with the variance of steady-state filtering uncertainty,

$$p_{1,1} = \frac{\phi^2 r - q - r + \sqrt{r^2(1-\phi^2)^2 + 2\ rq(\phi^2+1) + q^2}}{\phi^2}, \tag{6.55}$$

then with the cross-covariances of filter error with the smoother estimates,

$$p_{1,i+1} = \frac{p_{1,i}\phi\ r}{p_{1,1}\phi^2+q+r}, \tag{6.56}$$

for $1 \le i \le \ell$. Next, the variances of smoother stage estimates,

$$p_{i+1,i+1} = \frac{(p_{i,i}p_{1,1} - p_{1,i}^2)\phi^2 + p_{i,i}(q+r)}{p_{1,1}\phi^2+q+r}, \tag{6.57}$$

and finally the covariances between different smoother stage errors,

$$p_{j+1,j+k+1} = \frac{(p_{j,j+k}p_{1,1} - p_{1,j}p_{1,j+k})\phi^2 + p_{j,j+k}(q+r)}{p_{1,1}\phi^2+q+r} \tag{6.58}$$

for $1 \le j \le \ell - 1$ and $1 \le k \le \ell - j$.

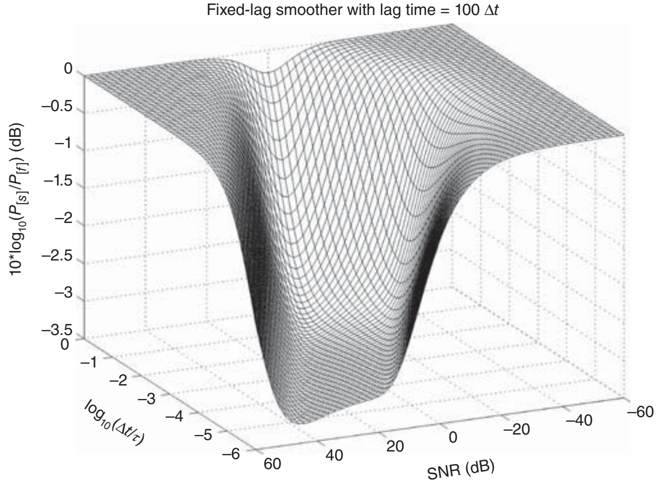

Figure 6.12 Improvement of smoothing over filtering versus SNR and $\Delta t/\tau$.

The ratio of ℓ-stage smoother variance to filter variance is then

$$\frac{\sigma^2_{[s]}}{\sigma^2_{[f]}} = \frac{p_{\ell+1,\ell+1}}{p_{1,1}}. \tag{6.59}$$

If the scalar $|\phi| < 1$, the dynamic system model is stable and models an exponentially correlated random process with finite steady-state mean-squared signal. Then the Riccati equation solution can be computed as a function of SNR and the ratio of discrete time step to correlation time. An example plot is shown in Figure 6.12, which plots the improvement of fixed-lag smoothing over filtering as a function of SNR and $\Delta t/\tau$, where τ is the process correlation time. Note that the best improvement is a factor of two (−3 dB), which is as good as it gets for stable systems.

6.4 FIXED-POINT SMOOTHING

6.4.1 Performance Analysis

Performance Models. "Performance models" are just that. They model estimator performance. They are based on an estimator model, but do not include the estimator itself. They only include the equations which govern the propagation of the covariance of estimation uncertainty.

Scalar Continuous Linear Time-Invariant Model. The following analytical performance model is for the scalar linear time-invariant case in continuous time. The stochastic system is modeled in continuous time because that approach is generally easier to manipulate and solve mathematically. The resulting formulas define how fixed-point smoothing performs as a function of time t; the scalar parameters F, Q,

H, and R; and the time t_{fixed} at which the smoothed estimated is required. Performance is defined by the variance of estimation uncertainty for the fixed-point smoother.

Comparisons between Smoothing and Filtering. Because there is no comparable filtering method that estimates the system state at a fixed time, we cannot compare fixed-lag smoothing performance with filter performance. Also, because we have added yet another model parameter (t_{fixed}), we can no longer reduce the number of independent parameters and plot smoother performance as a function of the model parameters, as was done for fixed-interval smoothing and fixed-lag smoothing (by resorting to videos).

MATLAB Implementations. This same model was used in generating the figures of RMS fixed-point smoother estimation uncertainty versus time. The enclosed MATLAB m-file `FPSperformance.m` implements the formulas used in generating the figures. You can modify this script to vary the model parameters and observe how that changes smoother performance.

6.4.1.1 End-to-End Model for Fixed-Point Smoothing The all-inclusive end-to-end fixed-point smoother model is made up of three different models, as illustrated in Figure 6.13. This shows the *forward Kalman–Bucy filter,* the *Kalman–Bucy predictor* from times before the designated fixed point in time t_{fixed}, and the *backward Kalman–Bucy filter* for times $t > t_{\text{fixed}}$. Different submodels are used over different time intervals:

1. For $t_{\text{start}} \leq t < t_{\text{fixed}}$, the variance of estimation uncertainty from the forward filter at time t is shown as $P_{[f]}(t)$. This is determined by the forward Kalman–Bucy filter model. The result must be projected ahead to the fixed time t_{fixed} by the prediction model, as $P_{[s]}(t_{\text{fixed}}) = P_{[p]}(t_{\text{fixed}})$, the variance of fixed-point smoothing uncertainty during that time period. The prediction uncertainty variance $P_{[p]}(t_{\text{fixed}})$ is a function of the filter uncertainty variance $P_{[f]}(t)$, as determined by the predictor model. The transformation of the filter uncertainty variance $P_{[f]}(t)$ at time t to the prediction uncertainty variance $P_{[p]}(t_{\text{fixed}})$ at time t_{fixed} uses the linearized Riccati equation models of Section 5.8.3 with $H = 0$ (i.e., no measurements).
2. When $t = t_{\text{fixed}}$, the variance of fixed-point smoothing uncertainty will be $P_{[s]}(t_{\text{fixed}}) = P_{[f]}(t_{\text{fixed}})$, the variance of the forward filter estimation uncertainty.

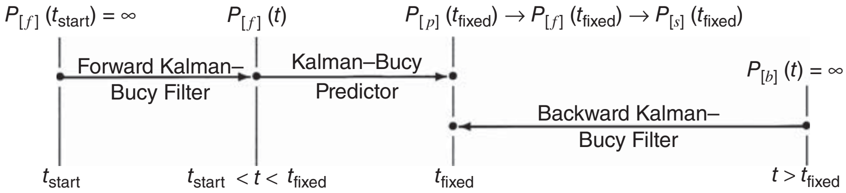
$P_{[f]}(t_{\text{start}}) = \infty$
$P_{[f]}(t)$
$P_{[p]}(t_{\text{fixed}}) \rightarrow P_{[f]}(t_{\text{fixed}}) \rightarrow P_{[s]}(t_{\text{fixed}})$
Forward Kalman–Bucy Filter
Kalman–Bucy Predictor
$P_{[b]}(t) = \infty$
Backward Kalman–Bucy Filter
t_{start}
$t_{\text{start}} < t < t_{\text{fixed}}$
t_{fixed}
$t > t_{\text{fixed}}$

Figure 6.13 Kalman–Bucy fixed-point smoother model.

3. When $t > t_{\text{fixed}}$, the variance of fixed-point smoothing uncertainty will be

$$P_{[s]}(t_{\text{fixed}}) = \frac{P_{[f]}(t_{\text{fixed}})P_{[b]}(t_{\text{fixed}})}{P_{[f]}(t_{\text{fixed}}) + P_{[b]}(t_{\text{fixed}})},$$

a combination of the forward ($P_{[f]}$) and backward ($P_{[b]}$) filter variances.

These submodels are further described and derived below in subsections 6.4.1.2 through 6.4.1.5.

Simpler Models for Special Cases Not all fixed-point smoothing problems require all the parts of this model:

1. If $t_{\text{fixed}} = t_{\text{start}}$, the initial time, then the forward filter and predictor are not required. This would be the case, for example, for estimating initial conditions using later measurements—as in the aforementioned problem of determining initial inertial navigation alignment errors.
2. If $t_{\text{fixed}} \geq t$ for the entire mission, then the backward filter is not required. This would be the case for missile intercept, for example, in which the predicted positions of the interceptor and target at t_{fixed} = the predicted time of impact are the state variables of primary interest for intercept guidance.

6.4.1.2 Forward Filter Submodel The subscript $[f]$ is used for parameters and variables of the forward Kalman–Bucy filter model. If there is no information about the state variable at the beginning, then the linearized Riccati equation for the Kalman–Bucy filter has the form

$$\begin{aligned}
P_{[f]}(t) &= A_{[f]}(t)/B_{[f]}(t) \\
\begin{bmatrix} A_{[f]}(t) \\ B_{[f]}(t) \end{bmatrix} &= \exp\left(\begin{bmatrix} F & Q \\ H^2/R & -F \end{bmatrix} (t - t_{\text{start}}) \right) \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
P_{[f]}(t) &= \frac{E_{[f]}^{(+)}\,\rho_{[f]} - RFE_{[f]}^{(-)} + RFE_{[f]}^{(+)} + E_{[f]}^{(-)}\,\rho_{[f]}}{H^2(-E_{[f]}^{(-)} + E_{[f]}^{(+)})} \\
\rho_{[f]} &= \sqrt{R(F^2R + QH^2)} \\
E_{[f]}^{(-)} &= e^{-\rho_{[f]}\,(t - t_{\text{start}})/R} \\
E_{[f]}^{(+)} &= e^{\rho_{[f]}\,(t - t_{\text{start}})/R}.
\end{aligned}$$

As part of the model design assumptions, as $t \to t_{\text{start}}$, $P_{[f]}(t) \to +\infty$. Therefore, for this model, RMS estimation uncertainty for the forward filter is valid only for evaluation times $t_{\text{start}} < t \leq t_{\text{fixed}}$.

6.4.1.3 Prediction Submodel The subscript $[p]$ is used for parameters and variables of the Kalman–Bucy predictor model. The predictor carries forward the filtered variance $P_{[f]}(t)$ to time $t_{\text{fixed}} > t$, assuming no measurements. The model in this case becomes

$$\begin{aligned}
P_{[p]}(t_{\text{fixed}}) &= \frac{A_{[p]}(t_{\text{fixed}})}{B_{[p]}(t_{\text{fixed}})} \\
\begin{bmatrix} A_{[p]}(t_{\text{fixed}}) \\ B_{[p]}(t_{\text{fixed}}) \end{bmatrix} &= \exp\left(\begin{bmatrix} F & Q \\ 0 & -F \end{bmatrix} (t_{\text{fixed}} - t) \right) \begin{bmatrix} P_{[f]}(t) \\ 1 \end{bmatrix} \\
P_{[p]}(t_{\text{fixed}}) &= e^{2\ (t_{\text{fixed}}-t)F} P_{[f]}(t) + \frac{Q(e^{2\ (t_{\text{fixed}}-t)F} - 1)}{2\ F} \\
&= P_{[f]}(t) + Q(t_{\text{fixed}} - t) \text{ if } F = 0 \\
&= P_{[f]}(t_{\text{fixed}}) \text{ if } t = t_{\text{fixed}}.
\end{aligned}$$

6.4.1.4 Backward Filter Submodel Fixed-point smoothing is not implemented using a backward Kalman filter, but the performance of the fixed-point smoother can be characterized by such a model.

The subscript $[b]$ is used for parameters and variables of the backward Kalman–Bucy filter model. This works backward from $t > t_{\text{fixed}}$ to t_{fixed}, starting with no state variable information ($P_{[b]}(t) = \infty$ at t). Reversing the direction of time changes the Riccati differential equation by reversing the sign of F, so that the linearized Riccati equation model becomes

$$\begin{aligned}
P_{[b]}(t_{\text{fixed}}) &= \frac{A_{[b]}(t_{\text{fixed}})}{B_{[b]}(t_{\text{fixed}})} \\
\begin{bmatrix} A_{[b]}(t_{\text{fixed}}) \\ B_{[b]}(t_{\text{fixed}}) \end{bmatrix} &= \exp\left(\begin{bmatrix} -F & Q \\ H^2/R & F \end{bmatrix} (t - t_{\text{fixed}}) \right) \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
P_{[b]}(t_{\text{fixed}}) &= \frac{E_{[b]}^{(+)}\ \rho_{[b]} + RFE_{[b]}^{(-)} - RFE_{[b]}^{(+)} + E_{[b]}^{(-)}\ \rho_{[b]}}{H^2(-E_{[b]}^{(-)} + E_{[b]}^{(+)})} \\
\rho_{[b]} &= \sqrt{R(F^2R + QH^2)} \\
E_{[b]}^{(-)} &= e^{-\rho_{[b]}\ (t-t_{\text{fixed}})/R} \\
E_{[b]}^{(+)} &= e^{\rho_{[b]}\ (t-t_{\text{fixed}})/R}.
\end{aligned}$$

6.4.1.5 Fixed-Point Smoother Submodel The resulting fixed-point smoother model uses the forward filter, predictor, and backward filter models, depending on where t is relative to t_{fixed}:

$$P_{[s]}(t_{\text{fixed}}) = \begin{cases} P_{[p]}\left(t_{\text{fixed}}\right), & t < t_{\text{fixed}} \\ P_{[f]}(t_{\text{fixed}}), & t = t_{\text{fixed}} \\ \dfrac{P_{[f]}(t_{\text{fixed}})P_{[b]}(t_{\text{fixed}})}{P_{[f]}(t_{\text{fixed}})+P_{[b]}(t_{\text{fixed}})}, & t > t_{\text{fixed}} \end{cases},$$

where the subscript $[s]$ on $P_{[s]}$ indicates the variance of estimation uncertainty of the fixed-point smoother.

6.4.1.6 Fixed-Point Smoothing Performance Figure 6.14 is a plot of modeled RMS fixed-point smoother uncertainty for a stochastic process model with

$$\begin{aligned} F &= 0.01\,(\text{dynamic coefficient}), \\ Q &= 0.01\ (\text{variance of dynamic process noise}), \\ H &= 1\ (\text{measurement sensitivity}), \\ R &= 0.01\ (\text{variance of measurement noise}), \\ t_{\text{start}} &= 0\ (\text{starting time}), \\ t_{\text{fixed}} &= 100\ (\text{fixed time at which system state is to be estimated}), \\ t_{\text{end}} &= 300\ (\text{ending time}), \end{aligned}$$

The plot also shows the RMS uncertainties in the forward filter estimate and the backward filter estimate. Notice how slowly the RMS uncertainty in the smoothed estimate

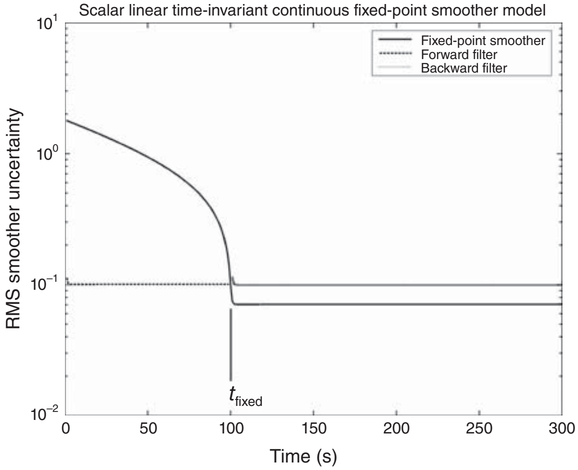

Figure 6.14 Sample performance of fixed-point smoother with $t_{\text{fixed}} = 100$.

falls at first, then much faster as the current time approaches the fixed time. For this particular set of parameters, it is the predictor which limits the RMS uncertainty of the smoother, which falls by another $1/\sqrt{2}$ as time passes t_{fixed} and the backward filter kicks in. The plot also indicates that—for these particular parameter values—the backward filter does not continue to reduce smoother uncertainty by much beyond a few seconds past t_{fixed}.

This plot was generated by the MATLAB m-file `FPSperformance.m`. You can vary these parameter values to see how they influence performance.

The effects of varying the model parameters are illustrated in Figures 6.15 (varying Q), 6.16 (varying R), and 6.17 (varying F), which are multiplots of the results with different parameter values.

6.4.2 Discrete-Time Fixed-Point Smoother

This type of smoother includes the Kalman filter to estimate the state at the current time t_k using the measurements up to time t_k, then adds the following equations to obtain a smoothed estimate of the state at a fixed time $t_i < t_k$:

$$\hat{x}_{[s]i|k} = \hat{x}_{[s]i|k-1} + B_k \overline{K}_k (z_k - H\hat{x}_{k(-)}), \tag{6.60}$$

$$B_k = B_{k-1} P_{k-1(+)} \Phi^{\mathrm{T}}_{k-1} P^{-1}_{k(-)}, \tag{6.61}$$

where the subscript notation $[s]i|k$ refers to the smoothed estimate of the state at time t_i, given the measurements up to time t_k. (A derivation and application of this technique to the analysis of INS test data may be found in Reference 1.) The values of

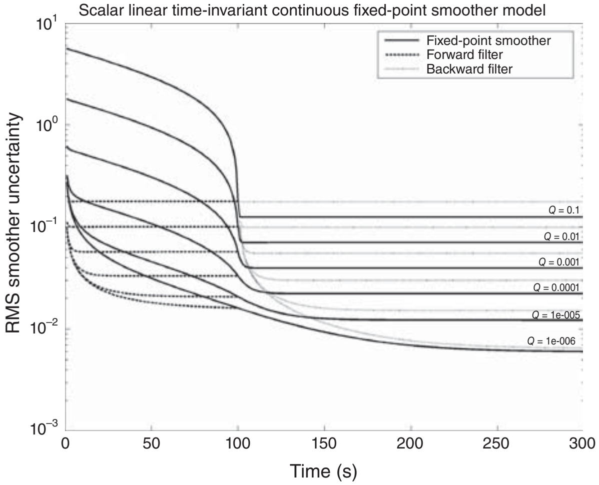

Figure 6.15 Fixed-point smoother performance with varying Q.

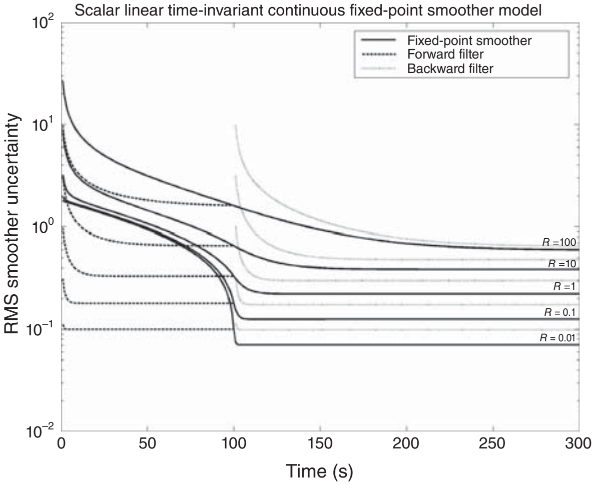

Figure 6.16 Fixed-point smoother performance with varying R.

$\hat{x}_{k(-)}, \overline{K}_k, z_k, H_kP$, and P are computed in the Kalman filter and the initial value $B_i = I$, the identity matrix. The covariance of uncertainty of the smoothed estimate can also be computed by the formula

$$P_{[s]i|k} = P_{[s]i|k-1} + B_k(P_{k(+)} - P_{k(-)})B_k^{\mathrm{T}}, \tag{6.62}$$

although this is not a necessary part of the smoother implementation.

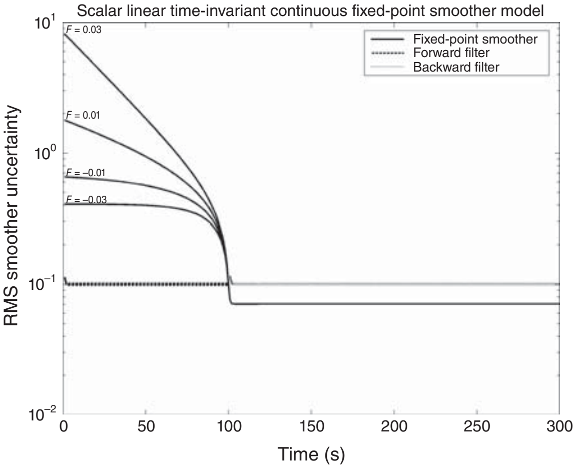

Figure 6.17 Fixed-point smoother performance with varying F.

6.5 SUMMARY

6.5.1 Smoothing

Modern optimal smoothing methods have been derived from the same types of models used in Kalman filtering. As a rule, the smoothed estimate has smaller mean-squared uncertainty than the corresponding filter estimate, but it requires measurements made *after* the time that the estimate is valid. Smoothing implementations are called *smoothers*. Several implementation algorithms have been derived, some with lower computational requirements or better numerical stability than others. Some of the more stable methods have been presented and implemented in MATLAB m-files.

6.5.1.1 Types of Smoothers Different types of smoothers have been developed for different types of applications. The most common types are

fixed-interval smoothers, which obtain the optimal estimate each time a measurement is sampled, using all the measurements sampled in the entire interval. It is most commonly used in post-processing.

fixed-lag smoothers, which typically run in real time but generate an estimated in delayed time. That is, when a measurement is sampled at time t, it is used to generate an estimate of the system state vector at time $t - \Delta t_{\text{lag}}$. Some of the first fixed-lag smoother implementations were numerically unstable, a problem that has been solved by more recent implementation methods.

fixed-point smoothers, which generate estimates of the system state vector at a fixed time t_{fixed}, using measurements made at times before and after t_{fixed}.

6.5.1.2 Implementations Smoothers can be implemented in discrete time (as algorithms) or in continuous time (as analog circuits) If the input signal bandwidth is too high for sampling, or computational requirements preclude digital implementation, then Kalman–Bucy models in continuous time can be used for designing the implementation circuitry.

In either case, there can be alternative realizations of the same smoothing function with different stability characteristics and different levels of complexity. Those implementation methods with less sensitivity to model parameter values and roundoff errors—or to analog component variability and noise—are preferred.

6.5.2 Improvement of Smoothing over Filtering

The metric used for the improvement of smoothing over filtering is the ratio of mean-squared estimation uncertainties.

- The relative improvement of smoothing over filtering depends on the values of the parameters of the stochastic system model (Q and F or Φ) and sensor model (H and R).
- For stable dynamic system models, the improvement of smoothing over filtering is at most a factor of two.

- For unstable observable dynamic system models, the improvement of smoothing over filtering can be orders of magnitude. This "improvement in the improvement" for unstable systems is due, in part, to the smoother using (in effect) a filter in reverse time, for which the effective dynamic system model is stable.

6.5.3 Sources for Additional Information

For historical and technical accounts of seminal works in optimal smoothing, see the surveys by Meditch [18], Kailath [19], and Park and Kailath [5]. The survey by Meditch includes many of the better-known fixed-lag smoother implementation methods.

For a summary of additional smoothing methods and concise technical overviews of smoothing algorithms and their history of development, see the survey by McReynolds [20].

PROBLEMS

6.1 For a scalar system with $\Phi = 1$, $H = 1$, $R = 2$, and $Q = 1$, find the steady-state value of the covariance P for a fixed-point smoother estimate.

6.2 Find the solution to Problem 5.1 for P_{10} when $P_0 = 1$, using the fixed-interval smoother on the interval $0 \leq k \leq 10$.

6.3 Solve Problem 5.1 with a fixed-lag smoother with the lag equal to two time steps.

6.4 Repeat Problem 5.1 with $R = 15$.

6.5 Let the model parameters of the Kalman filter be

$$\Phi_{[f]} = \begin{bmatrix} 0.9 & 0.3 \\ -0.3 & 0.9 \end{bmatrix} \tag{6.63}$$

$$H_{[f]} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{6.64}$$

$$Q_{[f]} = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \tag{6.65}$$

$$R_{[f]} = 1. \tag{6.66}$$

For a compatible BMFLS implementation with a lag of $\ell = 3$ time steps,

(a) What is the dimension of $\hat{x}_{[s]}$, the smoother state vector?

(b) Write down the corresponding smoother state-transition matrix $\Phi_{[s]}$.

(c) Write down the corresponding smoother measurement sensitivity matrix $H_{[s]}$.

(d) Write down the corresponding smoother disturbance noise covariance matrix $Q_{[s]}$.

6.6 For the 2×2 matrix $\Phi_{[f]}$ of Equation 6.63,

(a) Use the MATLAB function `eigs` to find its eigenvalues.

(b) Find their magnitudes using `abs`. Are they < 1, $= 1$, or > 1?

(c) What can you say about the eigenvalues of the coefficient matrix $F_{[f]}$ for the analogous model in continuous time? Are their real parts positive (i.e., in the right half-plane) or negative (in the left half-plane)?

(d) Is the dynamic system modeled as $x_k = \Phi_{[f]} x_{k-1}$ stable or unstable? Why?

(e) Let

$$x_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Using MATLAB, generate and plot the components of

$$x_k = \Phi[f]^k x_0 \text{ for } k = 1,\ 2,\ 3,\ \dots,\ 100.$$

Do they spiral inward toward the origin or outward toward ∞? How would you interpret that behavior?

(f) Will any fixed-lag smoother based on the same Kalman filter model make more than a factor of two improvementa in mean-squared estimation uncertainty? Justify your answer.

6.7 Repeat the exercises of the previous problem with

$$\Phi_{[f]} = \begin{bmatrix} 0.9 & 0.9 \\ -0.9 & 0.9 \end{bmatrix}. \tag{6.67}$$

6.8 Use the parameter values of Problem 6.5 and the MATLAB function `BMFLS` on the companion Wiley web site to demonstrate fixed-lag smoothing on simulated measurements. Remember to use the MATLAB code

```
x        = Phi*x+[sqrt(Q(1,1))*randn,sqrt(Q(2,2))*randn];
z        = H*x +  [sqrt(R(1,1))*randn,sqrt(R(2,2))*randn];
```

to simulate the dynamic noise and measurement noise. Plot

1. The simulated state vector.
2. The filter estimate.
3. The smoother estimate.
4. RMS filter uncertainties.
5. RMS smoother uncertainties.

Compare your results to those shown in Figure 6.11 of Example 6.1.

6.9 Use the parameter values of Problem 6.7 and the MATLAB function BMFLS on the companion Wiley web site to demonstrate fixed-lag smoothing on simulated measurements. Perform all the same tasks as in Problem 6.8.

REFERENCES

[1] M. S. Grewal, R. S. Miyasako, and J. M. Smith, "Application of fixed point smoothing to the calibration, alignment, and navigation data of inertial navigation systems," in *Proceedings of IEEE PLANS '88—Position Location and Navigation Symposium*, Orlando, FL, Nov. 29–Dec. 2, 1988, IEEE, New York, pp. 476–479, 1988.

[2] B. D. O. Anderson, "Properties of optimal linear smoothing," *IEEE Transactions on Automatic Control*, Vol. AC-14, pp. 114–115, 1969.

[3] B. D. O. Anderson and S. Chirarattananon, "Smoothing as an improvement on filtering: a universal bound," *Electronics Letters*, Vol. 7, No. 18, pp. 524–525, 1971.

[4] J. B. Moore and K. L. Teo, "Smoothing as an improvement on filtering in high noise," *System & Control Letters*, Vol. 8, pp. 51–54, 1986.

[5] P. G. Park and T. Kailath, "New square-root smoothing algorithms," *IEEE Transactions on Automatic Control*, Vol. 41, pp. 727–732, 1996.

[6] A. Gelb, J. F. Kasper Jr., R. A. Nash Jr., C. F. Price, and A. A. Sutherland Jr., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.

[7] H. E. Rauch, "Solutions to the linear smoothing problem," *IEEE Transactions on Automatic Control*, Vol. AC-8, pp. 371–372, 1963.

[8] T. Kailath and P. A. Frost, "An innovations approach to least squares estimation, Part II: Linear smoothing in additive white noise," *IEEE Transactions on Automatic Control*, Vol. AC - 13, pp. 646–655, 1968.

[9] C. N. Kelly and B. D. O. Anderson, "On the stability of fixed-lag smoothing algorithms," *Journal of the Franklin Institute*, Vol. 291, No. 4, pp. 271–281, 1971.

[10] K. K. Biswas and A. K. Mahalanabis, "Optimal fixed-lag smoothing for time-delayed systems with colored noise," *IEEE Transactions on Automatic Control*, Vol. AC-17, pp. 387–388, 1972.

[11] K. K. Biswas and A. K. Mahalanabis, "On the stability of a fixed-lag smoother," *IEEE Transactions on Automatic Control*, Vol. AC-18, pp. 63–64, 1973.

[12] K. K. Biswas and A. K. Mahalanabis, "On computational aspects of two recent smoothing algorithms," *IEEE Transactions on Automatic Control*, Vol. AC-18, pp. 395–396, 1973.

[13] R. Premier and A. G. Vacroux, "On smoothing in linear discrete systems with time delays," *International Journal on Control*, Vol. 13, pp. 299–303, 1971.

[14] J. B. Moore, "Discrete-time fixed-lag smoothing algorithms," *Automatica*, Vol. 9, No. 2, pp. 163–174, 1973.

[15] S. Prasad and A. K. Mahalanabis, "Finite lag receivers for analog communication," *IEEE Transactions on Communications*, Vol. 23, pp. 204–213, 1975.

[16] P. Tam and J. B. Moore, "Stable realization of fixed-lag smoothing equations for continuous-time signals," *IEEE Transactions on Automatic Control*, Vol. AC-19, No. 1, pp. 84–87, 1974.

[17] J. F. Riccati, “Animadversationnes in aequationes differentiales secundi gradus,” *Acta Eruditorum Quae Lipside Publicantur Supplementa*, Vol. 8, pp. 66–73, 1724.

[18] J. S. Meditch, “A survey of data smoothing for linear and nonlinear dynamic systems,” *Automatica*, Vol. 9, pp. 151–162, 1973.

[19] T. Kailath, “Correspondence item,” *Automatica*, Vol. 11, pp. 109–111, 1975.

[20] S. R. McReynolds, “Fixed interval smoothing: revisited,” *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 13, pp. 913–921, 1990.