KALMAN FILTERS

Estimation problems arise in diverse fields, such as communications, control, econometrics, and signal processing. Underlying these are many general results in probability and statistics. What distinguishes the particular applications mentioned above is the fact that they have additional structure that can be used to further refine these general results. The proper exploration and exploitation of this structure leads to many problems and challenges.

In this article we focus on a certain rather narrowly defined class of problems. This is essentially the study of linear least-squares estimation problems for signals with known finite-dimensional linear state-space models. Despite its apparent narrowness, this is a rich subject with useful applications to some very different problems, such as those of quadratic control, adaptive filtering, H_∞-filtering and control, matrix theory, and linear algebra.

The most celebrated estimation tool in this context is the Kalman filter; it is an efficient procedure for the estimation of the states of a linear state-space model from noisy observations of the output process. Since its inception in the early 1960s, the Kalman filter has attracted considerable attention and has encountered numerous applications in diverse fields. We discuss it in some detail in this article, but first we review the fundamental problem that underlies the Kalman filter theory—that of estimating one random variable from another.

STOCHASTIC ESTIMATION

Consider two (scalar or column-vector) random variables x and y (possibly complex-valued) with joint probability density function $f_{\mathbf{x},\mathbf{y}}(\cdot,\cdot)$. If the random variables are independent, that is, if they assume values independently of each other, then there is little (if anything) that can be said about the value assumed by one random variable when the value assumed by the other is known or measured. Therefore, we assume that the random variables are dependent, and ask the following question: given that the variable y assumed the value y in a particular experiment, what can be said (or guessed) about the value assumed by the random variable x?

Such questions often arise when the quantity of interest is not directly observable or directly measurable while it is possible to monitor another related quantity. For example, we may only have available noisy measurements y of x, say $y = x + v$, where the random variable v represents additive noise or disturbance. With a proper formulation, reasonable information about x can be extracted from the noisy measurements of y.

To tackle the general question, an estimate of the value assumed by x, say $\hat{x}$, can be described as a function of the value assumed by y, say $\hat{x} = h(y)$. We refer to x as the *estimate.* Likewise, we refer to the random variable $\hat{x}$ defined by $\hat{x} = h(y)$, as the *estimator:* evaluating the estimator $\hat{x}$ at a particular value for y results in an estimate $\hat{x}$.

The challenge is to suitably choose the function $h(\cdot)$ to yield reasonable estimates. By reasonable we mean estimates that satisfy a desired optimality criterion. There are several criteria that can be used for estimation problems, but for signal processing, communications, and control, one of the most important, at least in the sense of having had the most applications, is the least-squares criterion.

Nonlinear Least-Mean-Squares Estimation

The least-mean-squares (*lms*) criterion determines the function $h(\cdot)$ by minimizing the variance of the error variable $\hat{x} = x - \hat{x}$; in other words,

$$\min_{h(\cdot)} E(\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^*) \tag{1}$$

where the symbol $*$ denotes complex conjugation and E denotes the expected value. Note that $\tilde{x}\tilde{x}^*$ is a matrix not a scalar, since $\tilde{x}$ is a column random variable. In this regard, the minimization is to be interpreted with respect to the partial ordering defined over the set of nonnegative definite matrices. That is, the optimal solution $h(\cdot)$ will be such that the matrix difference

$$E[\boldsymbol{x} - h'(\boldsymbol{y})][\boldsymbol{x} - h'(\boldsymbol{y})]^* - E[\boldsymbol{x} - h(\boldsymbol{y})][\boldsymbol{x} - h(\boldsymbol{y})]^*$$

is always nonnegative definite for all other choices $h'(\cdot)$. It turns out that the optimal $h(\cdot)$ is given by the conditional expectation of x given y,

$$h(\boldsymbol{y}) = E(\boldsymbol{x}|\boldsymbol{y}) \tag{2}$$

The Case of Jointly Gaussian Random Variables

For general random variables x and y, the conditional expectation in Eq. (3) is generally a nonlinear function of the observations. For the special case of jointly Gaussian random variables, however, the expression collapses to a linear function of the observations. Linear estimators, as we shall see, have several advantages: they are easier to compute and, more important, easier to update.

The probability density function (*pdf*) of two jointly Gaussian zero-mean circular random variables x and y is proportional to

$$f_{\boldsymbol{x},\boldsymbol{y}}(x, y) \propto \exp\left\{-[x^* \quad y^*]R^{-1}\begin{bmatrix} x \\ y \end{bmatrix}\right\} \tag{3}$$

where R denotes their (nonsingular) covariance matrix,

$$R = \begin{bmatrix} R_x & R_{xy} \\ R_{yx} & R_y \end{bmatrix}$$

with

$$R_x = E\boldsymbol{x}\boldsymbol{x}^*, R_y = E\boldsymbol{y}\boldsymbol{y}^* \quad R_{xy} = E\boldsymbol{x}\boldsymbol{y}^* = R_{yx}^*$$

For such jointly Gaussian variables, it can be verified by direct calculation that the expression $E(x|y)$ in Eq. (3) for the optimal estimator becomes

$$\hat{\boldsymbol{x}} = R_{xy}R_y^{-1}\boldsymbol{y} \tag{4}$$

which is completely specified by the auto- and cross-correlation quantities (i.e., by the second-order statistics) of the quantities involved.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 2007 John Wiley & Sons, Inc.

Linear Estimators and the Orthogonality Condition

But what if the random variables x and y are not jointly Gaussian? An estimator having the same linear structure as Eq. (7) can still be obtained for general zero-mean random variables x and y by restricting ourselves to linear functions $h(\cdot)$. In this case, we seek a linear estimator of the form $x = K_0 y$, and determine the coefficient matrix K_0 by minimizing the error covariance matrix, that is,

$$\min_{K} E[\boldsymbol{x} - K\boldsymbol{y}][\boldsymbol{x} - K\boldsymbol{y}]^* \tag{5}$$

It turns out that all K_0 that solve Eq. (8) are solutions to the so-called *normal equations*

$$K_0 R_y = R_{xy} \tag{6}$$

When $R_y > 0$, the solution K_0 is unique and given by $K_0 = R_{xy}R^{-1}{}_{y}$, in which case the expression for $\hat{x}$ is identical to Eq. (7). Note furthermore that the normal Eq. (9) is equivalent to

$$E(\boldsymbol{x} - K_0\boldsymbol{y})\boldsymbol{y}^* = 0 \tag{7}$$

This suggests that if we regard the random variables x and y as vectors (i.e., elements) in an inner product space, with inner product defined by $\langle x, y\rangle \triangleq Exy^*$, then the above condition has the geometric meaning that $(x - K_0 y)$ is orthogonal to y, written as $(x - K_0 y) \perp y$. This is a fundamental property that fully characterizes linear least-mean-squares estimators (*llmse*).

Usually, the variable y is vector-valued and composed of several observations, say $y = \text{col}\{y_0, \ldots, y_N\}$, where each y_i is itself a possibly vector-valued random variable. (The notation $\text{col}\{\cdot\}$ denotes a column vector with the specified entries.) We shall then say that $\hat{x} = K_0 y$ is the projection of x onto the linear space spanned by the random variables $\{y_i\}$, written $\mathscr{L}\{y_0, \ldots, y_N\}$.

THE INNOVATIONS PROCESS

We therefore see that the solution of the llms estimation problem in Eq. (8) requires that we solve the normal equations $K_0 R_y = R_{xy}$. Since the solution of linear equations is a much studied problem, it would seem that there is not much more to be said, except to refer to some books on the subject. However, there are at least two features of the problem that should give us some pause:

1. It takes proportional to N^3 operations (an operation may be taken as the multiplication or addition of two real numbers) to solve an $N \times N$ set of linear equations. This can be a substantial amount of work when N is large: N could be of the order 10 to 100 in several aerospace problems and 500 to 2000 to 4000 to 10,000 in many environmental, geodetic, power-system, econometric, and image-processing problems.
2. For large N, there may be a problem of data storage, especially because in many applications the data comes in sequentially, so that we have to solve the estimation problem for sequentially increasing values of N. The storage problem could be ameliorated if we could develop a *sequential* or *recursive* method of solving the equations; it would be nice if the new datum could be used to update the previous estimate, and then discarded, so that no data storage is necessary. Note that recursive solutions can be useful whenever N is large, whether or not it is growing.

Although general methods are known for the recursive solution of linear equations, the problem must have some special structure if the number of computations (and the amount of storage) is to be significantly reduced, to say $O(N^2)$ or even $O(N)$ from $O(N^3)$. Fortunately, such structure is present in the estimation and control problems of interest to us; in particular, we deal with stochastic processes that have a certain finite-dimensional (state-space) structure, which will be reflected into the structure of the linear equations.

The exploration of structure can be carried out by algebraic or geometric methods in several different ways. We pursue one particular route here, motivated by our interest in state-space models.

A Geometric Approach

Recall that we are not interested in linear equations as such, but in those that arise from the problem of computing the projection of a vector, say x, onto the linear space spanned by another set of vectors (or random variables) $\{y_0, y_1, \ldots, y_N\}$. As we have seen, this problem reduces to the solution of a simultaneous set of linear equations, say $K_0 R_y = R_{xy}$, where

$$R_y = [\langle \boldsymbol{y}_i, \boldsymbol{y}_j\rangle]_{i,j=0:N} \quad \text{and} \quad R_{xy} = [\langle \boldsymbol{x}, \boldsymbol{y}_i\rangle]_{i=0:N}$$

It is a pretty obvious remark that these equations would be easy to solve if R_y were a diagonal matrix, or equivalently if the $\{y_i\}$ were orthogonal to each other, in which case the projection would reduce to just the sum of the projections of x onto each orthogonal vector. Of course, in most problems, R_y would not be diagonal; in fact, it is the nature of the dependence between the vectors $\{y_i\}$ that distinguishes various physical problems from each other.

To begin with, we henceforth always assume that the variables $\{y_i\}$ are not an arbitrary collection, but belong to an indexed or ordered set, in the sense that y_{i+1} follows y_i. In other words, we assume that the $\{y_i\}$ constitute a *stochastic process,* where the index i will be assumed, for definiteness, to be a time index, though it could also be a space index if desired.

The fact that the generally nonorthogonal vectors $\{y_i\}$ arise from an indexed set may immediately remind one of the obvious (in retrospect) *recursive* Gram–Schmidt procedure for replacing a set of indexed vectors by an *equivalent* orthogonal set of vectors. Thus assume that we have transformed $\{y_0, \ldots, y_N\}$ to an equivalent set of orthogonal vectors $\{e_0, \ldots, e_N\}$, equivalent in the sense that they span the same linear (sub)space, written, say,

$$\mathscr{L}\{\boldsymbol{e}_0, \ldots, \boldsymbol{e}_N\} = \mathscr{L}\{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_N\} \triangleq \mathscr{L}_N \tag{8}$$

If now we have an additional vector, y_{N+1}, a natural way of proceeding is by projecting y_{N+1} onto $\mathcal{L}_N$ to get

$$\boldsymbol{e}_{N+1} = \boldsymbol{y}_{N+1} - \text{proj}\cdot\{\boldsymbol{y}_{N+1}|\mathcal{L}_N\} \tag{9}$$

Moreover, finding the above projection is aided by Eq. (12), which allows us to find the projection by separately projecting onto each of the previously found orthogonal vectors $\{e_i\}$,

$$\text{proj}\cdot\{\boldsymbol{y}_{N+1}|\mathcal{L}_N\} = \sum_{j=0}^{N} \langle \boldsymbol{y}_{N+1}, \boldsymbol{e}_j\rangle \|\boldsymbol{e}_j\|^{-2}\boldsymbol{e}_j$$

where the notation $\|e\|^2$ stands for $Eee*$. This then leads to the recursive formula

$$\boldsymbol{e}_{N+1} = \boldsymbol{y}_{N+1} - \sum_{j=0}^{N} \langle \boldsymbol{y}_{N+1}, \boldsymbol{e}_j\rangle \|\boldsymbol{e}_j\|^{-2}\boldsymbol{e}_j \tag{10}$$

which can be begun with $e_0 = y_0$. This is known as the Gram–Schmidt orthogonalization procedure.

When the $\{y_i\}$ are random variables, a suggestive terminology can be associated with the orthogonal variables $\{e_i\}$. Thus recall that in the stochastic case,

$$\begin{aligned} \text{proj}\cdot\{\boldsymbol{y}_{N+1}|\mathcal{L}_N\} &= \text{the l.l.m.s. estimator of } \boldsymbol{y}_{N+1} \text{ given} \\ &\qquad \mathcal{L}\{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_N\} \\ &\overset{\Delta}{=} \hat{\boldsymbol{y}}_{N+1}, \quad \text{say} \end{aligned}$$

This is the part of the random variable y_{N+1} that is determined by knowledge of the previous random variables $\{y_0, \ldots, y_N\}$. The remainder is the random variable e_{N+1},

$$\boldsymbol{e}_{N+1} \overset{\Delta}{=} \boldsymbol{y}_{N+1} - \hat{\boldsymbol{y}}_{N+1} \tag{11}$$

which we can regard as the new information or the innovation in y_{N+1} given $\{y_0, \ldots, y_N\}$. Therefore, we shall call

$$\{\boldsymbol{e}_i\} = \text{the innovations process associated with } \{\boldsymbol{y}_i\}$$

As befits the name, each vector e_i brings new information, because e_i is uncorrelated with all other vectors $\{e_j\}_{j\neq i}$; in other words, the innovations process is a white noise process. However, the white noise property by itself is not enough to characterize the innovations. It is important that there is a causal relationship between the indexed collections $\{y_i\}$ and $\{e_i\}$: for every $i \geq 0$,

$$\boldsymbol{e}_i \in \mathcal{L}\{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_i\} \quad \text{and} \quad \boldsymbol{y}_i \in \mathcal{L}\{\boldsymbol{e}_0, \ldots, \boldsymbol{e}_i\}$$

In other words, the processes $\{y_i\}$ and $\{e_i\}$ are related by a causal and causally invertible linear transformation. This causality restriction makes the white-noise process $\{e_i\}$ unique (apart from scaling).

The Modified Gram–Schmidt Procedure

While the innovations process $\{e_i\}$ is unique, this does not mean that there is only one way of constructing them. Here we describe another alternative—the so-called modified Gram–Schmidt (*MGS*) procedure:

1. Set $e_0 = y_0$
2. Form $\tilde{y}_{i|0} = y_i - \langle y_i, e_0\rangle \|e_0\|^{-2}\, e_0$, and then set $e_1 = \tilde{y}_{1|0}$
3. Form $\tilde{y}_{i|1} = \tilde{y}_{i|0} - \langle \tilde{y}_{i|0}, e_1\rangle \|e_1\|^{-2}\, e_1$, and then set $e_2 = \tilde{y}_{2|1}$

and so on. The partial residuals $\{y_i, \tilde{y}_{i|0}, \tilde{y}_{i|1}, \ldots\}$ can be rearranged in a triangular array, the diagonal entries of which are the innovations $\{e_i\}$:

$$\begin{array}{lllll} \boldsymbol{y}_0 & & & & \\ \boldsymbol{y}_1 & \tilde{\boldsymbol{y}}_{1|0} & & & \\ \boldsymbol{y}_2 & \boldsymbol{y}_{2|0} & \tilde{\boldsymbol{y}}_{2|1} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \boldsymbol{y}_N & \tilde{\boldsymbol{y}}_{N|0} & \tilde{\boldsymbol{y}}_{N|1} & \cdots & \tilde{\boldsymbol{y}}_{N|N-1} \end{array}$$

This and other methods for determining the innovations all have special features of interest, but for the moment the point we wish to make is that they all take essentially the same order of elementary computations, that is, $O(N^3)$ for N innovations. In applications, however, we often have special structures, for example, *stationarity* of the process or the availability of *state-space* or *difference equation* models for it, that enable *fast* ways of obtaining the innovations. In this article, our focus is on state-space structure. If we have an n-dimensional state-space model for the observation process $\{y_i\}$, then it turns out that the innovations can be found with $O(Nn^3)$ operations, which can be very much less than $O(N^3)$ if $n < N$. The details are given further ahead.

Estimation Given the Innovations Process

The reason for seeking to determine the innovations is that we can now replace the problem of estimation given the process $\{y_i\}$, with the simpler one of estimation given the orthogonal innovations process $\{e_i, i \leq k\}$. Thus

$$\hat{\boldsymbol{x}}_{|N} \overset{\Delta}{=} \text{the llms estimator of } \boldsymbol{x} \text{ given } \{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_N\}$$

can also be expressed as

$$\hat{\boldsymbol{x}}_{|N} = \text{the llms estimator of } \boldsymbol{x} \text{ given } \{\boldsymbol{e}_0, \ldots, \boldsymbol{e}_N\}$$

which, due to the orthogonality of the $\{e_j\}$, is given by

$$\hat{\boldsymbol{x}}_{|N} = \sum_{j=0}^{N} \langle \boldsymbol{x}, \boldsymbol{e}_j\rangle \|\boldsymbol{e}_j\|^{-2}\boldsymbol{e}_j \tag{12}$$

Moreover, if we now have an additional observation y_{N+1}, then the estimator $\hat{x}_{|N}$ can be readily updated by using the innovation e_{N+1},

$$\begin{aligned} \hat{\boldsymbol{x}}_{|N+1} &= \hat{\boldsymbol{x}}_{|N} + (\text{llmse of } \boldsymbol{x} \text{ given } \boldsymbol{e}_{N+1}) \\ &= \hat{\boldsymbol{x}}_{|N} + \langle \boldsymbol{x}, \boldsymbol{e}_{N+1}\rangle \|\boldsymbol{e}_{N+1}\|^{-2}\boldsymbol{e}_{N+1} \quad \hat{\boldsymbol{x}}_{|-1} = 0 \end{aligned} \tag{13}$$

where

$$\boldsymbol{e}_{N+1} = \boldsymbol{y}_{N+1} - \hat{\boldsymbol{y}}_{N+1|N} = \boldsymbol{y}_{N+1} - \sum_{j=0}^{N} \langle \boldsymbol{y}_{N+1}, \boldsymbol{e}_j\rangle \|\boldsymbol{e}_j\|^{-2}\boldsymbol{e}_j \quad \boldsymbol{e}_0 = \boldsymbol{y}_0 \tag{14}$$

The simple formulas in Eqs. (23), (24), and (25) are the key to many results in linear least-squares estimation theory.

We may remark that we are often interested in estimating not just a single variable, x, but actually another stochastic process, say $\{x_i\}$, from observations of a process $\{y_i\}$. The way to tackle this apparently more difficult problem is to regard it as a collection of problems in each of which we estimate one of the variables in the $\{x_i\}$ process from the observations of the process $\{y_i\}$. We encounter this procedure in the sequel. Here we note again that for the basic formulas in Eqs. (23), (24), and (25) to be really useful, we must be able to determine the innovations $\{e_i\}$ in some way that requires less work than determining $\hat{x}_{|N}$ directly by solving N linear equations in N unknowns (strictly speaking, with our numbering convention, determining $\hat{x}_{|N}$ requires solving $N + 1$ linear equations in $N + 1$ unknowns; we are often guilty of this minor inconsistency). This is possible when the observation process arises from a finite-dimensional linear state-space model.

THE STANDARD STATE-SPACE MODEL

The assumption of a finite-dimensional state-space model for the observations process allows the innovations to be recursively and efficiently computed, with $O(Nn^3)$ computations as opposed to $O(N^3)$, where n is the state dimension and N is the number of observations. There are also many problems, especially in aerospace applications, where the state variables have a direct physical significance and where estimates of the state variables, or of some linear combinations of these variables, are needed. As noted earlier, once we have the innovations, the estimation of related quantities (states, inputs, and linear combinations thereof) is straightforward. We first set up a standard state-space model.

Since the early 1960s, much effort has been devoted to modeling processes $\{y_i\}$ in state-space form, that is,

$$\boldsymbol{y}_i = \boldsymbol{H}_i\boldsymbol{x}_i + \boldsymbol{v}_i \quad i \geq 0 \tag{15}$$

where the $n \times 1$ state-vector x_i obeys the recursion

$$\boldsymbol{x}_{i+1} = \boldsymbol{F}_i\boldsymbol{x}_i + \boldsymbol{G}_i\boldsymbol{u}_i \quad i \geq 0 \tag{16}$$

The processes v_i and u_i are assumed to be $(p \times 1)$- and $(m \times 1)$-vector zero-mean white noise processes, with

$$\left\langle \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{v}_i \end{bmatrix}, \begin{bmatrix} \boldsymbol{u}_j \\ \boldsymbol{v}_j \end{bmatrix} \right\rangle = \begin{bmatrix} Q_i & S_i \\ S_i^* & R_i \end{bmatrix} \delta_{ij} \tag{17}$$

whereas the initial state x_0 is assumed to have zero mean, covariance matrix Π_0, and to be uncorrelated with the $\{u_i\}$ and $\{v_i\}$, that is,

$$\langle \boldsymbol{x}_0, \boldsymbol{x}_0 \rangle = \Pi_0 \quad \text{and} \quad \langle \boldsymbol{u}_i, \boldsymbol{x}_0 \rangle = 0 \quad \langle \boldsymbol{v}_i, \boldsymbol{x}_0 \rangle = 0 \quad i \geq 0 \tag{18}$$

These assumptions can be compactly restated as

$$\left\langle \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{v}_i \\ \boldsymbol{x}_0 \end{bmatrix}, \begin{bmatrix} \boldsymbol{u}_j \\ \boldsymbol{v}_j \\ \boldsymbol{x}_0 \\ 1 \end{bmatrix} \right\rangle = \begin{bmatrix} \begin{bmatrix} Q_i & S_i \\ S_i^* & R_i \end{bmatrix} \delta_{ij} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 0 & \quad 0 \end{matrix} & \begin{matrix} \Pi_0 & 0 \end{matrix} \end{bmatrix} \tag{19}$$

It is also assumed that the matrices F_i [of dimension $(n \times n)$], $G_i(n \times m), H_i(p \times n), Q_i(m \times m), R_i(p \times p), S_i(m \times p)$, and $\Pi_0(n \times n)$ are known a priori. The process v_i is often called *measurement noise* and the process u_i *plant noise.* They are often uncorrelated (i.e., $S_i = 0$), but the more general assumption is necessary to handle problems where there may be feedback from the output to the states.

We do not discuss here how the state equations have been obtained. In many situations, the definitions of the state variables are naturally suggested by the physical problem; linearization may often have to be used to actually obtain linear equations as in Eqs. (26) and (27). As a result, the state-space model can be set up in slightly different forms, for example, with different assumptions on the correlation between $\{u_i, v_i\}$. These models can be analyzed in ways quite similar to the ones we are going to describe here. The model specified above will be henceforth called the *standard* model.

THE KALMAN FILTER

Now we go on to the problem of whether we can conveniently find the innovations, $e_i \triangleq y_i - \hat{y}_{i|i-1}$, when the $\{y_i\}$ have the state-space structure described above. It turns out that the recursive construction of the innovations combines nicely with the recursive evolution of the state variables to give a recursion for the innovations in terms of the parameters of the model and a pair of other matrices $\{K_{p,i}, R_{e,i}\}$. These can be computed in different ways, one of which we present here.

Recursion for the Innovations

Starting with $y_i = H_i x_i + v_i$, and *projecting* onto the linear subspace spanned by $\{y_0, \ldots, y_{i-1}\}$ yields

$$\hat{\boldsymbol{y}}_{i|i-1} = \boldsymbol{H}_i\hat{\boldsymbol{x}}_{i|i-1} + \hat{\boldsymbol{v}}_{i|i-1} \tag{20}$$

Our standard notational convention is that $\hat{x}_{i|j}$ = the projection of x_i on the linear subspace spanned by $\{y_0, \ldots, y_j\}$, $\mathscr{L}\{y_0, \ldots, y_j\}$. Now the assumptions on our state-space model imply that $v_i \perp y_j$ for $j \leq i - 1$, so that $\hat{v}_{i|i-1} = 0$ and

$$\boldsymbol{e}_i = \boldsymbol{y}_i - \hat{\boldsymbol{y}}_{i|i-1} = \boldsymbol{y}_i - \boldsymbol{H}_i\hat{\boldsymbol{x}}_{i|i-1} \tag{21}$$

Therefore, we see that the problem of finding the innovations reduces to one of finding a convenient way of determining the one-step predictions of the state-vector. For this purpose, we can try to use the basic formula for estimation given the (uncorrelated) innovations process

$$\hat{\boldsymbol{x}}_{i+1|i} = \sum_{j=0}^{i} \langle \boldsymbol{x}_{i+1}, \boldsymbol{e}_j \rangle R_{e,j}^{-1} \boldsymbol{e}_j \tag{22}$$

where $R_{e,j} = \langle e_j, e_j \rangle$.

This seems puzzling (in fact, circular), because so far we have only defined the innovations $\{e_i\}$ in terms of the one-step predictions, which are the things we are trying to estimate. The reason Eq. (33) can make sense is that on the right-hand-side we have the quantities $\{e_j, j \leq i\}$, so that in trying to find $\hat{x}_{i+1|i}$ from Eq. (32), we are only using earlier one-step predictions $\{\hat{x}_{j|j-1}, j \le i\}$. This suggests that what we should try to find is a recursive solution, with the present value $\hat{x}_{i+1|i}$ being computed from the most recent past value $\hat{x}_{i|i-1}$ and the new information $e_i = y_i - H_i\hat{x}_{i|i-1}$. To see if this is possible, let us first rewrite Eq. (33) in a form more indicative of a recursion

$$\begin{aligned}\hat{x}_{i+1|i} &= \left(\sum_{j=0}^{i-1}\langle x_{i+1}, e_j\rangle R_{e,j}^{-1} e_j\right) + \langle x_{i+1}, e_i\rangle R_{e,i}^{-1} e_i \\ &= \hat{x}_{i+1|i-1} + \langle x_{i+1}, e_i\rangle R_{e,i}^{-1}(y_i - H_i\hat{x}_{i|i-1})\end{aligned} \tag{23}$$

This is almost in the desired form, and would be exactly so if the term $\hat{x}_{i+1|i-1}$ could be expressed in terms of just $\hat{x}_{i|i-1}$ and e_i. At this point, no more general statements can be made; to go further we must have more information about the way the states change with time.

In our problem we know that x_{i+1} obeys the state equation $x_{i+1} = F_i x_i + G_i u_i$. But then projecting onto the linear subspace spanned by $\{y_j, j \le i - 1\}$ shows that

$$\hat{x}_{i+1|i-1} = F_i\hat{x}_{i|i-1} + G_i\hat{u}_{i|i-1} = F_i\hat{x}_{i|i-1} + 0 \tag{24}$$

since by the assumptions on our model, $u_i \perp y_j, j \le i - 1$. But a relation as in Eq. (35) is exactly what we were seeking. In other words, by combining Eqs. (32) to (24) we have the following recursive set of equations for determining the innovations:

$$e_i = y_i - H_i\hat{x}_{i|i-1} \tag{25}$$

$$\hat{x}_{i+1|i} = F_i\hat{x}_{i|i-1} + K_{p,i}e_i \quad i \ge 0 \tag{26}$$

with initial conditions

$$\hat{x}_{0|-1} = 0 \quad \text{or equivalently, } e_0 = y_0 \tag{27}$$

and where we have defined

$$K_{p,i} = \langle x_{i+1}, e_i\rangle R_{e,i}^{-1} \tag{28}$$

The subscript p indicates that $K_{p,i}$ is used to update a predicted estimator.

The $\{K_{p,i}, R_{e,i}\}$ are nonrandom quantities that should be completely determinable from our knowledge of the means and covariances of the model, and in fact we shall show how this can be done; once the $\{K_{p,i}, R_{e,i}\}$ have been specified, we see that the innovations $\{e_i\}$ can be computed in a nice recursive way via Eqs. (25) to (27).

We can combine Eqs. (25) and (26) as

$$\hat{x}_{i+1|i} = F_{p,i}\hat{x}_{i|i-1} + K_{p,i}y_i \quad F_{p,i} \triangleq F_i - K_{p,i}H_i \quad \hat{x}_{0|-1} = 0 \quad i \ge 0 \tag{29}$$

which emphasizes that in finding the innovations, we actually also have a complete recursion for the state estimators $\{\hat{x}_{i|i-1}\}$.

The Error-Variance Matrices

To complete the computation of the innovations, let us describe one way of computing the coefficients $\{K_{p,i}, R_{e,i}\}$ needed for the basic recursions shown by Eqs. (25) and (26). The formulas we present here were first explicitly given by Kalman in 1960. Some important alternative methods (the so-called square-root and fast equation methods) for computing $\{K_{p,i}, R_{e,i}\}$ will be presented later.

Kalman began by introducing the quantity

$$P_i \triangleq \langle\tilde{x}_{i|i-1}, \tilde{x}_{i|i-1}\rangle \quad \tilde{x}_{i|i-1} \triangleq x_i - \hat{x}_{i|i-1} \tag{30}$$

which is of course of independent interest as the covariance matrix of the error in the predicted state estimator, and noting that the quantities $\{K_{p,i}, R_{e,i}\}$ in the basic recursions in Eqs. (25) to (28) could be expressed in terms of the $\{P_i\}$. It remains only to specify the $\{P_i\}$ in terms of the model parameters, and he showed that they could be described via a discrete-time Riccati recursion,

$$P_{i+1} = F_iP_iF_i^* + G_iQ_iG_i^* - K_{p,i}R_{e,i}K_{p,i}^* \quad i \ge 0 \tag{31}$$

with initial condition $P_0 = \Pi_0$. The recursion was so named by Kalman as an analog of a famous quadratically nonlinear differential equation attributed to Jacopo Francesco, Count Riccati (ca. 1700), and first ingenuously exploited in the calculus of variations by Legendre (1786). It was reintroduced into control theory by Bellman in 1957, and then in general matrix form by Kalman in 1960.

It is important to note that since one-step predicted quantities are encountered often, we use the following briefer notations (except when necessary for emphasis) $\hat{x}_i \triangleq \hat{x}_{i|i-1}$, and $\tilde{x}_i \triangleq \tilde{x}_{i|i-1}$.

The Gain Matrix and the Innovations Variance

Returning to Eq. (42), and to see how P_i enters into the computation of $\{K_{p,i}, R_{e,i}\}$, note first that since

$$e_i = y_i - H_i\hat{x}_i = H_ix_i - H_i\hat{x}_i + v_i = H_i\tilde{x}_i + v_i \tag{32}$$

and $v_i \perp \tilde{x}_i$, we can express the covariance matrix of e_i in terms of P_i,

$$R_{e,i} \triangleq \langle e_i, e_i\rangle = H_iP_iH_i^* + R_i \tag{33}$$

It turns out that this is also true of $K_{p,i}$. For we have

$$\langle x_{i+1}, e_i\rangle = F_i\langle x_i, e_i\rangle + G_i\langle u_i, e_i\rangle \tag{34}$$

and it can be checked that $\langle x_i, e_i\rangle = P_iH^*{}_i$ and $\langle u_i, e_i\rangle = S_i$. Therefore,

$$K_{p,i} \triangleq \langle x_{i+1}, e_i\rangle R_{e,i}^{-1} = (F_iP_iH_i^* + G_iS_i)R_{e,i}^{-1} \tag{35}$$

so we see that $\{K_{p,i}, R_{e,i}\}$ can be determined once we have the error covariance matrices $\{P_i\}$. These, we show soon, can be successively computed via the previously mentioned *discrete Riccati recursion* shown in Eq. (42).

It is important to note that the quantities $\{P_i, K_{p,i}, R_{e,i}\}$ depend only upon the prior assumptions on the model and not on the actual observations $\{y_i\}$; therefore, these quantities can be precomputed (or computed off-line) and stored for use in the actual prediction calculations. However, the above formulas do allow these quantities to be updated as needed (in real time), thus eliminating the need for extensive storage.

Recursion for the State-Error Variance

The covariance matrix of the state-vector of a white-noise driven process, $x_{\mathrm{i}+1} = F_{\mathrm{i}}x_{\mathrm{i}} + G_{\mathrm{i}}u_{\mathrm{i}}$, obeys the easily derived recursion

$$\Pi_{i+1} = \boldsymbol{F}_i \Pi_i \boldsymbol{F}_i^* + \boldsymbol{G}_i \boldsymbol{Q}_i \boldsymbol{G}_i^*, \quad \Pi_i \triangleq \langle \boldsymbol{x}_i, \boldsymbol{x}_i \rangle \tag{36}$$

Now we note that the estimator equation is also one driven by a white-noise process, namely the innovations

$$\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{F}_i \hat{\boldsymbol{x}}_i + \boldsymbol{K}_{p,i} \boldsymbol{e}_i, \quad \langle \boldsymbol{e}_i, \boldsymbol{e}_j \rangle \triangleq \boldsymbol{R}_{e,i} \delta_{ij}$$

Therefore, if we define the covariance matrix of the state estimators as $\Sigma_{\mathrm{i}} \triangleq \langle \hat{x}_{\mathrm{i}}, \hat{x}_{\mathrm{i}} \rangle$, then (as for Π_{i}) we can write

$$\Sigma_{i+1} = \boldsymbol{F}_i \Sigma_i \boldsymbol{F}_i^* + \boldsymbol{K}_{p,i} \boldsymbol{R}_{e,i} \boldsymbol{K}_{p,i}^* \tag{37}$$

with initial condition $\Sigma_0 = 0$. But the orthogonal decomposition $x_{\mathrm{i}} = \hat{x}_{\mathrm{i}} + \tilde{x}_{\mathrm{i}}$, with $\hat{x}_{\mathrm{i}} \perp \tilde{x}_{\mathrm{i}}$, shows that $\Pi_{\mathrm{i}} = \Sigma_{\mathrm{i}} + P_{\mathrm{i}}$. It is now immediate that

$$\boldsymbol{P}_{i+1} = \Pi_{i+1} - \Sigma_{i+1} = \boldsymbol{F}_i(\Pi_i - \Sigma_i)\boldsymbol{F}_i^* + \boldsymbol{G}_i \boldsymbol{Q}_i \boldsymbol{G}_i^* - \boldsymbol{K}_{p,i} \boldsymbol{R}_{e,i} \boldsymbol{K}_{p,i}^*$$

which is indeed the Riccati recursion shown in Eq. (42). This is perhaps the most direct route to the Riccati recursion.

Statement of the Kalman Filter

In summary, given the state-space model shown in Eqs. (<xref target="W7210-mdis-0026 W7210-mdis-0027" style="unformatted"/>), and (30), the innovations of the process $\{y_{\mathrm{i}}\}$ can be recursively computed via the equations

$$\boldsymbol{e}_i = \boldsymbol{y}_i - \boldsymbol{H}_i \hat{\boldsymbol{x}}_i, \quad \hat{\boldsymbol{x}}_{i+1} = \boldsymbol{F}_i \hat{\boldsymbol{x}}_i + \boldsymbol{K}_{p,i} \boldsymbol{e}_i, \quad \hat{\boldsymbol{x}}_0 = 0, \quad \boldsymbol{e}_0 = \boldsymbol{y}_0 \tag{38}$$

where $K_{\mathrm{p},i} = (F_{\mathrm{i}}P_{\mathrm{i}}H^{*}{}_{\mathrm{i}} + G_{\mathrm{i}}S_{\mathrm{i}})R^{-1}{}_{\mathrm{e},i}$, $R_{\mathrm{e},i} = R_{\mathrm{i}} + H_{\mathrm{i}}P_{\mathrm{i}}H^{*}{}_{\mathrm{i}}$, and P_{i} is computed using Eq. (42).

Note that the number of computations required for going from e_{i} to $e_{\mathrm{i}+1}$ is $O(n^3)$, because the most expensive step is the computation of the triple product $F_{\mathrm{i}}P_{\mathrm{i}}F^{*}{}_{\mathrm{i}}$, of $n \times n$ matrices.

Measurement and Time Updates

In addition to the predicted estimators $\{\hat{x}_{\mathrm{i}|i-1}\}$, we may be interested in the so-called *filtered estimators* $\hat{x}_{\mathrm{i}|i}$, or in going from $\hat{x}_{\mathrm{i}|i-1}$ to $\hat{x}_{\mathrm{i}|i}$ (a so-called *measurement-update* step), or from $\hat{x}_{\mathrm{i}|i}$ to $\hat{x}_{\mathrm{i}+1}$ (a so-called *time-update* step). These are readily obtained using the innovations.

For the measurement-update step we can verify that

$$\hat{\boldsymbol{x}}_{i|i} = \hat{\boldsymbol{x}}_i + \boldsymbol{K}_{f,i} \boldsymbol{e}_i, \quad \boldsymbol{K}_{f,i} \triangleq \boldsymbol{P}_i \boldsymbol{H}_i^* \boldsymbol{R}_{e,i}^{-1} \tag{39}$$

with

$$\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_{i|i}\|^2 \triangleq \boldsymbol{P}_{i|i} = \boldsymbol{P}_i - \boldsymbol{K}_{f,i} \boldsymbol{R}_{e,i} \boldsymbol{K}_{f,i}^* = \boldsymbol{P}_i - \boldsymbol{P}_i \boldsymbol{H}_i^* \boldsymbol{R}_{e,i}^{-1} \boldsymbol{H}_i \boldsymbol{P}_i \tag{40}$$

Likewise, for the time-update step we have

$$\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{F}_i \hat{\boldsymbol{x}}_{i|i} + \boldsymbol{G}_i \boldsymbol{S}_i \boldsymbol{R}_{e,i}^{-1} \boldsymbol{e}_i \tag{41}$$

with

$$\begin{aligned} \boldsymbol{P}_{i+1} = {} & \boldsymbol{F}_i \boldsymbol{P}_{i|i} \boldsymbol{F}_i^* + \boldsymbol{G}_i(\boldsymbol{Q}_i - \boldsymbol{S}_i \boldsymbol{R}_{e,i}^{-1} \boldsymbol{S}_i^*)\boldsymbol{G}_i^* \\ & - \boldsymbol{F}_i \boldsymbol{K}_{f,i} \boldsymbol{S}_i^* \boldsymbol{G}_i^* - \boldsymbol{G}_i \boldsymbol{S}_i \boldsymbol{K}_{f,i}^* \boldsymbol{F}_i^* \end{aligned} \tag{42}$$

These results suggest another useful way of carrying out the Kalman filter recursions. Thus note that the estimators $\{\hat{x}_{\mathrm{i}}\}$ and $\{\hat{x}_{\mathrm{i}|i}\}$ can be sequentially computed, starting with $\hat{x}_{0|-1} = 0$, and using first the measurement-update equation followed by the time-update equation. That is, starting with the given initial estimator $\hat{x}_{0|-1}$, we can successively compute the estimators as indicated below

$$0 = \hat{\boldsymbol{x}}_{0|-1} \xrightarrow{\text{mu}} \hat{\boldsymbol{x}}_{0|0} \xrightarrow{\text{tu}} \hat{\boldsymbol{x}}_1 \xrightarrow{\text{mu}} \hat{\boldsymbol{x}}_{1|1} \xrightarrow{\text{tu}} \hat{\boldsymbol{x}}_2 \xrightarrow{\text{mu}} \hat{\boldsymbol{x}}_{2|2} \xrightarrow{\text{tu}} \hat{\boldsymbol{x}}_3 \cdots$$

where the abbreviations mu and tu stand for measurement and time updates, respectively.

Similarly, starting with the given value $P_{0|-1} = \Pi_0$, we can successively compute

$$\Pi_0 = \boldsymbol{P}_{0|-1} \xrightarrow{\text{mu}} \boldsymbol{P}_{0|0} \xrightarrow{\text{tu}} \boldsymbol{P}_{1|0} \xrightarrow{\text{mu}} \boldsymbol{P}_{1|1} \cdots$$

A nice feature of this two-step (measurement and time update) form of the equations is that it makes clear how to proceed if we have a variable time between measurements or if, for some reason, certain measurements are lost. Therefore, digital computer implementations of the Kalman filter tend to be of this form. Analog (or hybrid) computer realizations usually use the prediction estimator equation.

Sequential Processing

The measurement update formulation can be used to motivate another scheme that is widely used in practical applications and is based on the reduction of a vector measurements problem to a sequence of scalar measurement problems. Indeed, when $R_{\mathrm{i}} > 0$, we can rewrite the measurement update equations in the equivalent form:

$$\boldsymbol{P}_{i|i}^{-1} = \boldsymbol{P}_i^{-1} + \boldsymbol{H}_i^* \boldsymbol{R}_i^{-1} \boldsymbol{H}_i \quad \text{and} \quad \boldsymbol{P}_{i|i}^{-1} \hat{\boldsymbol{x}}_{i|i} = [\boldsymbol{P}_i^{-1} \hat{\boldsymbol{x}}_i + \boldsymbol{H}_i^* \boldsymbol{R}_i^{-1} \boldsymbol{y}_i] \tag{43}$$

Moreover, while $K_{\mathrm{f},i}$ does not appear explicitly in these formulas, it will be useful to note that it can be rewritten as

$$\boldsymbol{K}_{f,i} = \boldsymbol{P}_{i|i} \boldsymbol{H}_i^* \boldsymbol{R}_i^{-1} = (\boldsymbol{P}_i^{-1} + \boldsymbol{H}_i^* \boldsymbol{R}_i^{-1} \boldsymbol{H}_i)^{-1} \boldsymbol{H}_i^* \boldsymbol{R}_i^{-1} \tag{44}$$

Since the inverse of the variance of a parameter is a (rough) measure of the information in the parameter, i.e., large variance means high uncertainty or less information, these formulas are often described as *information-form* measurement-update formulas.

A very useful application of these formulas is to reduce the problem of vector measurements (i.e., y_{i} a $p \times 1$ vector, $p > 1$) to that of a sequence of scalar measurements. Doing this would reduce computations because inversion of the $p \times p$ matrices $R_{\mathrm{e},i}$ would be trivialized.

If the measurements are in fact nonscalar, but the additive noise covariance matrices R_{i} are strictly positive definite, then by preliminary operations we can arrange that the entries of the output noise vector be uncorrelated. More specifically, let $R_i = L_i D_i L^*_i$ denote a triangular factorization of R_i, say, and scale the output equation $y_i = H_i x_i + v_i$ by $L^{-1}{}_i$, that is, $L^{-1}{}_i y_i = L^{-1}{}_i H_i x_i + L^{-1}{}_i v_i$. Then the new noise sequence $\bar{v}_i = L^{-1}{}_i v_i$ is such that

$$E\bar{\boldsymbol{v}}_i\bar{\boldsymbol{v}}_j^* = D_i\delta_{ij} \quad D_i = \mathrm{diag}\{d_i^1, d_i^2, \ldots, d_i^p\}$$

for some positive numbers $\{d^j{}_i\}$.

We further partition the entries of the scaled output vector $L^{-1}{}_i y_i$, and of the scaled matrix $L^{-1}{}_i H_i$, as follows:

$$L_i^{-1}\boldsymbol{y}_i = \mathrm{col}\{\boldsymbol{y}^1(i), \boldsymbol{y}^2(i), \ldots, \boldsymbol{y}^p(i)\} \quad L_i^{-1}H_i = \mathrm{col}\{h_i^1, h_i^2, \ldots, h_i^p\}$$

where $\{y^k(i)\}$ are scalars and $\{h^k{}_i\}$ are row vectors.

Now the p measurement processes $\{y^1(i), \ldots, y^p(i)\}$ will be mutually uncorrelated and we should be able to incorporate them one at a time, essentially by making a series of measurement updates, first with $y^1(i)$, then with $y^2(i)$, $\ldots$, and finally with $y^p(i)$.

To do this, we successively compute a sequence of matrices

$$\begin{aligned}
&P_i^1 = (I - K_{f,i}^1 h_i^1)P_i && K_{f,i}^1 = P_i h_i^{1*}[h_i^1 P_i h_i^{1*} + d_i^1]^{-1} \\
&P_i^2 = (I - K_{f,i}^2 h_i^2)P_i^1 && K_{f,i}^2 = P_i^1 h_i^{2*}[h_i^2 P_i^1 h_i^{2*} + d_i^2]^{-1} \\
&\vdots && \vdots \\
&P_i^p = (I - K_{f,p}^p h_i^p)P_i^{p-1} && K_{f,i}^p = P_i^{p-1} h_i^{p*}[h_i^p P_i^{p-1} h_i^{p*} + d_i^p]^{-1}
\end{aligned} \tag{45}$$

Then $P^p{}_i$ will be the updated covariance matrix $P_{i|i}$ based on all the measurements. Note that all the inversions required here are trivial, that is, scalar. As far as the estimators go, sequential incorporation of the new information in the components $\{y^1(i), \ldots, y^p(i)\}$ will lead to the equations $\hat{x}_{i|i} = \hat{x}^p{}_i$, where by the basic measurement update formula

$$\hat{\boldsymbol{x}}_i^k = \hat{\boldsymbol{x}}_i^{k-1} + K_{f,i}^k[\boldsymbol{y}^k(i) - h_i^k\hat{\boldsymbol{x}}_i^{k-1}], \quad k = 1, \ldots, p, \quad \hat{\boldsymbol{x}}_i^0 = \hat{\boldsymbol{x}}_i \tag{46}$$

and

$$K_{f,i}^k = \langle\boldsymbol{x}_i, \boldsymbol{e}^k(i)\rangle\|\boldsymbol{e}^k(i)\|^{-2}, \quad \boldsymbol{e}^k(i) = \boldsymbol{y}^k(i) - h_i^k\hat{\boldsymbol{x}}_i^{k-1} \tag{47}$$

Now if we define $P^k{}_i = \|\tilde{x}^k{}_i\|^2$, $\tilde{x}^k{}_i = x_i - \hat{x}^k{}_i$, then we can readily see that

$$K_{f,i}^k = P_i^k h_i^{k*}[h_i^k P_i^k h_i^{k*} + d_i^k]^{-1} \tag{48}$$

It should be noted that this is the same as the expression for $K^k{}_{f,i}$ in Eq. (62), where we did not make explicit the stochastic meaning of $P^k{}_i$ and $K^k{}_{f,i}$.

STEADY-STATE BEHAVIOR

The Kalman filter Eqs. (<xref target="W7210-mdis-0042 W7210-mdis-0044" style="unformatted"/>), and (45) have the interesting feature that even when the underlying state-space model is time-invariant, say described by the constant matrices $\{F, G, H, Q, R, S\}$, the equations for recursively predicting x_i using $\{y_j\}^{i-1}{}_{j=0}$ are time-variant, since both $K_{p,i}$ and $R_{e,i}$ depend on the time-variant Riccati variable P_i.

A natural question of interest would be to clarify whether in the time-invariant case the gain matrices $\{K_{p,i}\}$ might tend to a constant matrix, say K_p, as $i \to \infty$. Furthermore, we might wonder if the steady-state value depends upon the particular initial condition $P_0 = \Pi_0$, that is, $K_{p,0} = (F\Pi_0 H^* + GS)(R + H\Pi_0 H^*)^{-1}$. These are important questions, with several different and important consequences. The most obvious is that if K_p is constant, the optimum filter will be time invariant and generally easier to implement. Less obvious is the fact that if $K_{p,i}$ tends to the same value K_p no matter what $\Pi_0 \geq 0$ is, then the effects of unavoidable round-off errors introduced at each stage of the computation will tend to die off as time progresses; if this was not true, then errors would accumulate and the results would soon become meaningless.

In fact, it turns out that convergence of the Riccati recursion [Eq. (42)] can be guaranteed for some indefinite, and even negative semidefinite, initial matrices Π_0 (provided they are bounded below by a certain negative semidefinite matrix). This has certain implications for the numerical stability of the Riccati recursion, because it shows that even if the Riccati variable P_i loses its positive semidefiniteness (due to say, numerical errors) it may still converge.

The general convergence result states that given the Riccati recursion [Eq. (42)] and the following assumptions:

1. $\{F, H\}$ is detectable There are various characterizations of detectability. For example, a pair $\{F, H\}$ is detectable if, and only, if there exists a constant matrix K such that $F - KH$ is stable. Likewise, a pair $\{F, G\}$ is said to be stabilizable if $\{F^*, G^*\}$ is detectable.
2. $\{F - GSR^{-1}H, G(Q - SR^{-1}S^*)^{1/2}\}$ is stabilizable The notation $Q^{1/2}$ denotes any matrix satisfying $Q = Q^{1/2}Q^{*/2}$, where Q is nonnegative definite and $Q^{*/2}$ further denotes $[Q^{1/2}]^*$.
3. The initial condition P_0 is such that $I + (P^a)^{*/2}P_0(P^a)^{1/2} > 0$, where P^a is the unique positive semidefinite solution to the so-called dual Riccati equation,

$$P^a = F^{s*}P^aF^s + H^*R^{-1}H - F^{s*}P^aG(Q^{-s} + G^*P^aG)^{-1}G^*P^aF^s$$

with $F^s = F - GSR^{-1}H$, $Q^s = Q - SR^{-1}S^*$.

Then P_i converges to the unique positive semidefinite matrix, P, that satisfies the discrete-time algebraic Riccati equation (*DARE*)

$$P = FPF^* + GQG^* - (FPH^* + GS)(R + HPH^*)^{-1}(FPH^* + GS)^* \tag{49}$$

Furthermore, the limiting matrix P is such that the matrix determining the observer dynamics is a stable matrix, that is, all the eigenvalues of $F - K_pH$, $K_p = (FPH^* + GS)(R + HPH^*)^{-1}$, are less than unity in magnitude. Moreover, in z-transform notation we have,

$$\hat{\boldsymbol{x}}(z) = (zI - F + K_pH)^{-1}K_p\boldsymbol{y}(z) \tag{50}$$

In other words, the steady-state (or asymptotic) estimators of the state vector can be found by passing the observations through a linear-time-invariant (*LTI*) filter with transfer matrix $(zI - F + K_pH)^{-1}K_p$. This is the so-called Wiener filter for the problem of estimating the state vector from a stationary output process.

It can also be verified that the rate of convergence of P_i to P is exponential. That is, $\rho(P_i - P) \le c\lambda^{2i}$, where $\rho(\cdot)$ denotes the spectral radius of its argument, $\lambda = \rho(F - K_p H)$ is less than unity ($\lambda < 1$), and c is a bounded constant.

FAST ALGORITHMS FOR TIME-INVARIANT SYSTEMS

Another feature of the Kalman filter equations is that their computational requirements are indifferent to whether the coefficient matrices $\{F_i, G_i, H_i, Q_i, R_i, S_i\}$ are constant (time-invariant) or not. In particular, it takes $O(n^3)$ operations (additions and multiplications of real numbers) to update P_i to P_{i+1} via the Riccati Eq. (42), whether the matrices $\{F_i, G_i, H_i, Q_i, R_i, S_i\}$ are constant or not. This is a strength—the algorithms are general; but also a weakness, because we would expect that in some way constant-parameter problems should be easier to handle than similar time-variant problems.

It turns out that estimation for a constant parameter state-space model $\{F, G, H, Q, R, S\}$ can be achieved by replacing the Riccati recursions used in the Kalman filter by a different set of fast recursions. These equations can be solved with less effort than those of the Riccati-equation-based Kalman filter: $O(n^2)$ rather than $O(n^3)$. The difference can be very significant for large n.

The fast equations can be described as follows. Introduce the difference

$$\Delta(\Pi_0) \triangleq F\Pi_0F^* + GQG^* - (F\Pi_0H^* + GS)(R + H\Pi_0H^*)^{-1}(F\Pi_0H^* + GS)^* - \Pi_0 \tag{51}$$

and factor it (nonuniquely) as $\Delta(\Pi_0) = L_0M_0L^*{}_0$, where L_0 and M_0 are $n \times \alpha$ and $\alpha \times \alpha$ matrices, $\alpha = \text{rank}\ \Delta(\Pi_0)$, and $M_0 = \text{diag}\{1, \ldots, 1, -1, \ldots, -1\}$ is a signature matrix with as many ±1's as $\Delta(\Pi_0)$ has strictly positive and strictly negative eigenvalues. In other words, M_0 describes what is called the inertia of the matrix $\Delta(\Pi_0)$. Then the gain matrix $K_{p,i}$ can be computed as follows. We write $K_{p,i} = K_iR^{-1}{}_{e,i}$, and generate $\{K_i, R_{e,i}\}$ via recursions involving certain auxiliary sequences $\{L_i, R_{r,i}\}$:

$$K_{i+1} = K_i - FL_iR_{r,i}^{-1}L_i^*H^* \tag{52}$$

$$L_{i+1} = (F - K_iR_{e,i}^{-1}H)L_i \tag{53}$$

$$R_{e,i+1} = R_{e,i} - HL_iR_{r,i}^{-1}L_i^*H^* \tag{54}$$

$$R_{r,i+1} = R_{r,i} - L_i^*H^*R_{e,i}^{-1}HL_i \tag{55}$$

with initial conditions $K_0 = F\Pi_0H^* + GS, R_{e,0} = R + H\Pi_0H^*$, and $R_{r,0} = -M^{-1}{}_0$. Moreover, the error covariance matrix P_i can be computed as $P_{i+1} = -\Sigma^i{}_{j=0}\ L_jR^{-1}{}_{r,j}L^*{}_j$. This algorithm is of interest when the parameter α is significantly smaller than n, which happens in several important cases.

The above recursions are sometimes said to be of Chandrasekhar-type because they are generalizations of equations introduced by Chandrasekhar (1947, 1950) in certain radiative transfer problems.

ARRAY ALGORITHMS

As mentioned earlier, the largest amount of computation in the Kalman filter recursions arises in propagating the error covariance matrix P_i. However, more is at stake than the amount of computation. One consequence of round-off error is that the computed P_i may be non-Hermitian. This is sometimes compensated for by averaging the computed P_i and its Hermitian transpose. A better solution is only to propagate half the elements in P_i—say the ones on and below the main diagonal.

A more serious consequence arises from the fact that the P_i being covariance matrices have to be nonnegative definite. But round-off errors in the computation might destroy this property. Moreover, this is not an easy property to check; a matrix may be indefinite even if all its diagonal entries are nonnegative. The diagonal entries are the mean-square errors in the estimates of each of the components of the state vector and, of course, the computation would be seriously off if these diagonal entries turned out to be negative.

Nevertheless, it has been observed that such situations need not always be catastrophic—it can happen that the computation recovers, and that some iterations later the P_i are nonnegative definite. One explanation arises from the previously mentioned result that convergence of P_i to a constant matrix can happen even for certain indefinite Π_0 (provided their smallest eigenvalue is not too negative). Because the system is time-invariant, the P_i at each i can be regarded as the initial value for the Riccati recursion, and the result just mentioned therefore allows for P_i to be indefinite for some values of i without affecting the ultimate convergence.

Despite these possibilities, it is desirable to try to ensure that P_i is always nonnegative-definite. It turns out that an important step in this direction is to propagate not P_i but a square-root factor, that is, a matrix A_i such that $P_i = A_iA^*{}_i$. There will be of course round-off errors in propagating A_i, just as for P_i, but the point is that the product of the computed factors, say $\hat{P}_i = \hat{A}_i\hat{A}^*{}_i$, is almost certainly nonnegative-definite. In theory, $\hat{A}_i\hat{A}^*{}_i$ always is nonnegative-definite, but of course again round-off effects may arise; however, they are much easier to control, and in fact, it is easy to see that the diagonal elements will never be negative. Such algorithms are called array algorithms and are briefly discussed next. They have the following general form:

1. We form a certain *prearray* of numbers based on the given data at time i.
2. This array is reduced to a specified form (often triangular) by *a sequence of elementary unitary operations* (rotations or reflections).
3. The desired quantities at time i + 1 can be immediately read-off from the resulting so-called *postarray.*

No explicit equations are necessary. Such array algorithms are often much simpler to describe and implement (in software or hardware) than explicit sets of equations: they are becoming the algorithms of choice in many applications, including state-space estimation.

In the sequel, we shall make for simplicity of presentation, the standing assumption that $S_i = 0$. We may remark that when $R_i > 0$, a circumstance to be favored in setting up the state-space model, nonzero S_i can always be accommodated by replacing $\{F_i, Q_i\}$ by

$$F_i^s = F_i - G_i S_i R_i^{-1} H_i \quad \text{and} \quad Q_i^s = Q_i - S_i R_i^{-1} S_i^* \tag{56}$$

Square-Root Factors

As noted above, a matrix A such that $P = AA^*$ is called a square-root factor of P. Such factors are not unique, since $A\Theta$, for any unitary matrix Θ (i.e., one that satisfies $\Theta\Theta^* = \Theta^*\Theta = I$), is clearly also a square-root factor. We can choose Θ to make the factor unique, for example, by making $A\Theta$ Hermitian, or as we choose, by making it lower triangular with positive diagonal elements. For notational convenience, we denote a square-root factor of a matrix P by $P^{1/2}$, and almost always understand it as the unique triangular square-root factor. We also write

$$P = (P^{1/2})(P^{1/2})^* = P^{1/2}P^{*/2}$$

and

$$P^{-1} = (P^{*/2})^{-1}(P^{1/2})^{-1} = P^{-*/2}P^{1/2}$$

Array Algorithm for the Time-Update Problem

An array algorithm is fairly evident for the time-update problem. Indeed, Eq. (55) for the error covariance matrix, assuming $S_i = 0$, is

$$P_{i+1} = F_i P_{i|i} F_i^* + G_i Q_i G_i^* \quad i \geq 0 \tag{57}$$

Hence,

$$P_{i+1} = [F_i P_{i|i}^{1/2} \quad G_i Q_i^{1/2}][F_i P_{i|i}^{1/2} \quad G_i Q_i^{1/2}]^* \tag{58}$$

This gives a factorization of P_{i+1}, but unfortunately the dimensions of the factor $[F_i P^{1/2}{}_{i|i} \; G_i Q^{1/2}{}_i]$ are too large, $n \times (n + m)$ rather than $n \times n$. However, here we could take advantage of the nonuniqueness of square root factors and introduce a unitary matrix Θ,

$$P_{i+1} = [F_i P_{i|i}^{1/2} \quad G_i Q_i^{1/2}]\Theta\Theta^*[F_i P_{i|i}^{1/2} \quad G_i Q_i^{1/2}]^* \tag{59}$$

and try to choose Θ so that

$$[F_i P_{i|i}^{1/2} \quad G_i Q_i^{1/2}]\Theta = [X \quad 0_{n\times m}] \tag{60}$$

where $0_{n\times m}$ denotes an $n \times m$ matrix of all zero elements and X denotes a presently undetermined $n \times n$ matrix. If we can find such a Θ, then it must hold by squaring that

$$[F_i P_{i|i}^{1/2} \quad G_i Q_i^{1/2}]\underbrace{\Theta\Theta^*}_{I}[F_i P_{i|i}^{1/2} \quad G_i Q_i^{1/2}]^* = [X \quad 0_{n\times m}][X \quad 0_{n\times m}]^* \tag{61}$$

and, hence,

$$F_i P_{i|i} F_i^* + G_i Q_i G_i^* = XX^* \tag{62}$$

But since the left-hand side is equal to P_{i+1}, X can be identified as $P^{1/2}{}_{i+1}$, a square-root factor of P_{i+1}. So we have the following algorithm. Form a so-called *prearray*

$$\mathcal{A}_1 = [F_i P_{i|i}^{1/2} \quad G_i Q_i^{1/2}]$$

and unitarily (block) triangularize it to yield a *postarray* of the form

$$\mathcal{A}_1\Theta = [X \quad 0_{n\times n}], \quad \Theta\Theta^* = I_{n+m} = \Theta^*\Theta$$

We can identify X as a square-root of P_{i+1}. Uniqueness could be ensured by assuming that X is, say, lower triangular.

In summary, the array algorithm for the time-update problem takes the following form (assuming $S_i = 0$):

$$[F_i P_{i|i}^{1/2} \quad G_i Q_i^{1/2}]\Theta = [P_{i+1}^{1/2} \quad 0] \tag{63}$$

where Θ is any unitary matrix that triangularizes the prearray. The matrix Θ can be found in several ways via well-known methods in numerical linear algebra, including those based on Givens rotations and Householder reflections.

Array Algorithm for the Measurement-Update Problem

We now wish to go from $P^{1/2}{}_i$ to $P^{1/2}{}_{i|i}$ in accordance with the measurement update equation $P_{i|i} = P_i - P_i H^*{}_i R^{-1}{}_{e,i} H_i P_i$. For this purpose, we form the prearray

$$\mathcal{A}_2 = \begin{bmatrix} R_i^{1/2} & H_i P_i^{1/2} \\ 0 & P_i^{1/2} \end{bmatrix} \tag{64}$$

and then triangularize it via a unitary transformation Θ:

$$\begin{bmatrix} R_i^{1/2} & H_i P_i^{1/2} \\ 0 & P_i^{1/2} \end{bmatrix}\Theta = \begin{bmatrix} X & 0 \\ Y & Z \end{bmatrix} \tag{65}$$

The entries $\{X, Y, Z\}$ in the postarray can be identified by squaring both sides as above. So we shall be brief and simply state the final form of the array algorithm:

$$\begin{bmatrix} R_i^{1/2} & H_i P_i^{1/2} \\ 0 & P_i^{1/2} \end{bmatrix}\Theta = \begin{bmatrix} R_{e,i}^{1/2} & 0 \\ \bar{K}_{f,i} & P_{i|i}^{1/2} \end{bmatrix} \tag{66}$$

where $\bar{K}_{f,i} = P_i H^*{}_i R^{-*/2}{}_{e,i}$.

Array Algorithm for the Predicted Estimates

By combining the measurement-and-time update steps we can obtain the following algorithm. Form the prearray

$$\mathcal{A}_3 = \begin{bmatrix} R_i^{1/2} & H_i P_i^{1/2} & 0 \\ 0 & F_i P_i^{1/2} & G_i Q_i^{1/2} \end{bmatrix}$$

and triangularize it via a unitary transformation Θ to get, say,

$$\mathcal{A}_3\Theta = \begin{bmatrix} X & 0 & 0 \\ Y & Z & 0 \end{bmatrix} \tag{67}$$

By squaring, we can identify the entries $\{X, Y, Z\}$ and obtain the array equations

$$\begin{bmatrix} R_i^{1/2} & H_i P_i^{1/2} & 0 \\ 0 & F_i P_i^{1/2} & G_i Q_i^{1/2} \end{bmatrix} \Theta = \begin{bmatrix} R_{e,i}^{1/2} & 0 & 0 \\ \bar{K}_{p,i} & P_{i+1}^{1/2} & 0 \end{bmatrix} \tag{68}$$

where $\bar{K}_{p,i} = K_i R^{1/2}{}_{e,i}$.

Operation Counts and Condensed Forms

The number of operations needed in going from step i to step $(i + 1)$ in the array algorithm for the predicted estimators is $O(n^3)$, the same order as the Riccati-based algorithm. In general, though, the actual number of computations in the array method would tend to be somewhat larger than in the direct Riccati equation method. However, there are of course important compensatory numerical advantages, and with proper programming it appears that the computational efforts can be made essentially the same. It is also useful to first transform the given model parameters $\{F_i, G_i, H_i\}$ by unitary operations to so-called condensed forms, which help reduce the operations count further.

Fast Array Algorithms

The fast recursions [Eqs. (70) to (73)] also admit an array form. For this purpose, we assume that we are given a constant-parameter state-space model and, at any time instant i, we consider the difference $\delta P_i = (P_{i+1} - P_i)$ and introduce a (nonunique) factorization $P_{i+1} - P_i = \bar{L}_i J_i \bar{L}^*{}_i$, where $\bar{L}_i$ is an $n \times \alpha_i$ matrix, J_i is an $\alpha_i \times \alpha_i$ signature matrix with as many ± 1's as $(P_{i+1} - P_i)$ has positive and negative eigenvalues, and $\alpha_i = \text{rank}(P_{i+1} - P_i)$. The time subscript i is used in both J_i and α_i to indicate, for now, that the inertia of δP_i may vary with time. It will follow, however, that the inertia of δP_i does not vary with time.

The array algorithm follows by forming the prearray

$$A = \begin{bmatrix} R_{e,i}^{1/2} & H\bar{L}_i \\ \bar{K}_{p,i} & F\bar{L}_i \end{bmatrix} \tag{69}$$

and triangularizing it via an $(I \oplus J_i)$ unitary matrix Θ, that is,

$$A\Theta = \begin{bmatrix} R_{e,i}^{1/2} & H\bar{L}_i \\ \bar{K}_{p,i} & F\bar{L}_i \end{bmatrix} \Theta = \begin{bmatrix} X & 0 \\ Y & Z \end{bmatrix} \tag{70}$$

for some Θ such that

$$\Theta \begin{bmatrix} I & 0 \\ 0 & J_i \end{bmatrix} \Theta^* = \begin{bmatrix} I & 0 \\ 0 & J_i \end{bmatrix}$$

We can identify the $\{X, Y, Z\}$ terms by comparing the $(I \oplus J_i)$ norms on both sides of Eq. (93). We omit the details and only state that this calculation allows us to make the identifications $X = R^{1/2}{}_{e,i+1}$, $Y = \bar{K}_{p,i+1}$, $Z = \bar{L}_{i+1}$, and to conclude also that $J_{i+1} = J_i \equiv J$.

In summary, the quantities $\{\bar{K}_{p,i}, R^{1/2}{}_{e,i}\}$ can be recursively updated via the array algorithm

$$\begin{bmatrix} R_{e,i}^{1/2} & H\bar{L}_i \\ \bar{K}_{p,i} & F\bar{L}_i \end{bmatrix} \Theta = \begin{bmatrix} R_{e,i+1}^{1/2} & 0 \\ \bar{K}_{p,i+1} & \bar{L}_{i+1} \end{bmatrix} \tag{71}$$

where Θ is any $(I \oplus J)$ unitary matrix that produces the block zero entry in the postarray. Moreover, the initial conditions are

$$R_{e,0} = R + H\Pi_0 H^* \quad K_0 = F\Pi_0 H^* + GS$$

with $(\bar{L}_0, J)$ obtained via the factorization

$$P_1 - \Pi_0 = [F\Pi_0 F^* + GQG^* - K_0 R_{e,0}^{-1} K_0^* - \Pi_0] = \bar{L}_0 J \bar{L}_0^*$$

SMOOTHING ALGORITHMS

The Kalman filter and its variants give us recursive algorithms for computing the predicted and filtered state estimators, $\hat{x}_{i|i-1}$ and $\hat{x}_{i|i}$. It is not hard to compute higher order predicted estimates $\hat{x}_{i+m|i}$, $m > 0$. In fact

$$\hat{x}_{i+m|i} = F_{i+m-1} \dots F_i \hat{x}_{i|i} \quad m > 0$$

However the determination of smoothed estimators, say $\hat{x}_{i|N}$ for $i < N$, requires more effort. We state here some smoothing algorithms.

The Bryson-Frazier Formulas

Consider again the standard state-space model and assume further that $S_i = 0$, for simplicity. The so-called Bryson–Frazier (*BF*) algorithm finds the smoothed estimators $\hat{x}_{i|N}$ by

$$\hat{x}_{i|N} = \hat{x}_i + P_i \lambda_{i|N} \quad 0 \le i \le N \tag{72}$$

where $\lambda_{i|N}$ is found via the backward recursion

$$\lambda_{i|N} = F_{p,i}^* \lambda_{i+1|N} + H_i^* R_{e,i}^{-1} e_i, \quad \lambda_{N+1|N} = 0 \tag{73}$$

The corresponding error-covariance matrix can be found as $P_{i|N} = P_i - P_i \Lambda_{i|N} P_i$, where

$$\Lambda_{i|N} \triangleq \|\lambda_{i|N}\|^2 = F_{p,i}^* \Lambda_{i+1} F_{p,i} + H_i^* R_{e,i}^{-1} H_i \tag{74}$$

The quantities $\{\hat{x}_{i,i}, R_{e,i}, F_{p,i}, P_i\}$ are as in the Kalman filter formulas.

The BF formulas give us a two-pass algorithm. On a *forward pass,* we compute the innovations and the predicted and filtered state estimators; then a *backward pass* uses the innovations to compute the so-called adjoint variables $\{\lambda_{i|N}\}$. Finally, an appropriate combination gives the smoothed estimators.

The Rauch–Tung–Striebel Formulas

Assuming that F_i is invertible and $P_i > 0$, the so-called Rauch–Tung–Striebel (*RTS*) formulas compute the smoothed estimators as follows:

$$\hat{x}_{i|N} = F_{s,i} \hat{x}_{i+1|N} + F_i^{-1} G_i Q_i G_i^* P_{i+1}^{-1} \hat{x}_{i+1} \tag{75}$$

where

$$F_{s,i} \triangleq F_i^{-1}(I - G_i Q_i G_i^* P_{i+1}^{-1}) = P_i F_{p,i}^* P_{i+1}^{-1} = P_{i|i} F_i^* P_{i+1}^{-1} \tag{76}$$

The error covariance matrix obeys

$$P_{i|N} = F_{s,i} P_{i+1|N} F_{s,i}^* + F_i^{-1} G_i Q_i^r G_i^* F_i^{-*} \tag{77}$$

where

$$Q_i^r \triangleq Q_i - Q_i G_i^* P_{i+1}^{-1} G_i Q_i \tag{78}$$

There is also an alternative set of discrete-time formulas that also goes by the name RTS formulas. They are slightly more general than the above formulas in that they do not require the invertibility of the F_{i}. Defining $F_{\mathrm{s},i}$ = $P_{\mathrm{i}}F^*{}_{\mathrm{p},i}P^{-1}{}_{\mathrm{i+1}}$, it can be shown that

$$\hat{\boldsymbol{x}}_{i|N} = F_{s,i}\hat{\boldsymbol{x}}_{i+1|N} + (\hat{\boldsymbol{x}}_{i|i} - F_{s,i}\hat{\boldsymbol{x}}_{i+1}) \tag{79}$$

and

$$P_{i|N} = F_{s,i} P_{i+1|N} F_{s,i}^* + P_{i|i} - F_{s,i} P_{i+1} F_{s,i}^* \tag{80}$$

where the boundary conditions $\hat{x}_{\mathrm{N}|N}$ and $P_{\mathrm{N}|N}$ can be obtained by applying the appropriate Kalman filter recursions to the data $\{y_0, y_1, \ldots, y_{\mathrm{N}}\}$.

The RTS algorithm is also a two-pass algorithm, with all smoothed estimators being directly obtained at the end of the backward pass; note that we need only the estimators $\{\hat{x}_{\mathrm{i}}\}$ and $\{\hat{x}_{\mathrm{i}|i}\}$ for the second pass, the original data $\{y_{\mathrm{i}}\}$ and even the innovations $\{e_i\}$ need not be retained. The fact that the $P^{-1}{}_{\mathrm{i+1}}$ are required at every step perhaps increases the computational burden somewhat over that required for the original BF formulas. However, the differences are small and much will depend on the actual codes and machines on which the algorithms are run.

Note that for fixed-interval smoothing problems, the direction of time is not important, and we should be able to process the data both forward and backward in time, say starting with y_{N} and ending with y_0. Smoothing algorithms that are based on combinations of forward estimators and backward estimators involve so-called two-filter formuals. We omit the details for space limitations.

The Hamiltonian Equations

Using Eq. (100), the backward recursion for $\lambda_{\mathrm{i}|N}$, and Eq. (99) which gives $\hat{x}_{\mathrm{i}|N}$ as a linear combination of $\hat{x}_{\mathrm{i}}$ and $\lambda_{\mathrm{i}|N}$, as well as the Riccati Eq. (42) for P_{i}, we can derive a slightly different backward recursion for $\lambda_{\mathrm{i}|N}$, that is,

$$\lambda_{i|N} = F_i^* \lambda_{i+1|N} - H_i^* R_i^{-1} H_i \hat{\boldsymbol{x}}_{i|N} + H_i^* R_i^{-1} \boldsymbol{y}_i \tag{81}$$

Likewise, we can show that

$$\hat{\boldsymbol{x}}_{i+1|N} = F_i \hat{\boldsymbol{x}}_{i|N} + G_i Q_i G_i^* \lambda_{i+1|N} \tag{82}$$

Combining this equation with Eq. (108) in matrix form we get

$$\begin{bmatrix} \hat{\boldsymbol{x}}_{i+1|N} \\ \lambda_{i|N} \end{bmatrix} = \begin{bmatrix} F_i & G_i Q_i G_i^* \\ -H_i^* R_i^{-1} H_i & F_i^* \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{x}}_{i|N} \\ \lambda_{i+1|N} \end{bmatrix} + \begin{bmatrix} 0 \\ H_i^* R_i^{-1} \end{bmatrix} \boldsymbol{y}_i \tag{83}$$

where from Eqs. (100) and (99), the boundary conditions are found to be

$$\hat{\boldsymbol{x}}_{0|N} = \Pi_0 \lambda_{0|N}, \quad \lambda_{N+1|N} = 0 \tag{84}$$

These are the so-called Hamiltonian equations. They have several interesting features. First of all, they are called Hamiltonian equations because equations of this type are encountered in certain classical (deterministic) variational problems associated with famous names such as Euler, Lagrange, Hamilton, and so on.

Another interesting fact is that the Hamiltonian equations are only a representation for the smoothed estimators—they do not (directly) provide an algorithm for finding $\{\hat{x}_{\mathrm{i}|N}\}$. The reason is that the boundary conditions Eq. (111) are mixed: one variable is specified at $i = N + 1$ ($\lambda_{\mathrm{N+1}|N} = 0$), but the other one only at $i = 0$ (and that too only implicitly, $\hat{x}_{0|N} = \Pi_0\lambda_{0|N}$).

Yet another fascinating and fruitful feature of the equations is a physical picture in terms of a (generalized) transmission-line. This physical picture (called the scattering picture) turns out to be very useful in understanding many aspects of the state-space estimation problem.

Equation (110) can be graphically depicted as shown in Fig. 1, which suggests that we can regard $\hat{x}_{\cdot|N}$ as a forward wave and $\lambda_{\cdot|N}$ as a *backward* wave traveling through a section of a scattering medium that is specified by the quantities:

$$\begin{aligned} F_i &= \text{the forward transmission coefficient} \\ F_i^* &= \text{the backward transmission coefficient} \\ -H_i^* R_i^{-1} H_i &= \text{the left reflection coefficient} \\ G_i Q_i G_i^* &= \text{the right reflection coefficient} \\ H_i^* R_i^{-1} \boldsymbol{y}_i &= \text{the internal backward source excitation} \end{aligned}$$

We can put together many such sections to get a macroscopic section. By studying the propagation of signals through such scattering sections, we can derive all the filtering and smoothing formulas described so far in the article, in addition to several change-in-initial-conditions formulas that are not as immediate to derive through other methods.

CONTINUOUS-TIME STATE-SPACE FILTERING

All our discussion so far has been on discrete-time signals and systems. Now many physical systems evolve continuously in time, as do many physical signals. In this section we provide a brief overview of results on state-space filtering for continuous-time state-space models. Much of the discussion is patterned along our derivation for the discrete-time case in the earlier sections.

The standard continuous-time state-space model is of the form

$$\dot{\boldsymbol{x}}(t) = F(t)\boldsymbol{x}(t) + G(t)\boldsymbol{u}(t) \tag{85}$$

$$\boldsymbol{y}(t) = H(t)\boldsymbol{x}(t) + \boldsymbol{v}(t) \quad t \geq 0 \tag{86}$$

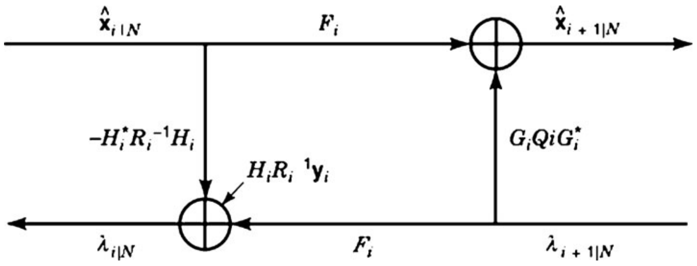

Figure 1. A scattering layer for the fixed-interval smoothing problem.

where $\{u(\cdot), v(\cdot)\}$ are white noise processes such that

$$\left\langle \begin{bmatrix} u(t) \\ v(t) \\ x(0) \end{bmatrix}, \begin{bmatrix} u(s) \\ v(s) \\ x(0) \\ 1 \end{bmatrix} \right\rangle = \begin{bmatrix} Q(t)\delta(t-s) & S(t)\delta(t-s) & 0 & 0 \\ S^*(t)\delta(t-s) & R(t)\delta(t-s) & 0 & 0 \\ 0 & 0 & \Pi_0 & 0 \end{bmatrix} \tag{87}$$

where $\langle a(t), b(s)\rangle = Ea(t)b^*(s)$, for zero-mean random processes $\{a(\cdot), b(\cdot)\}$. These equations are clearly quite analogous to those for the standard discrete-time model. The major difference is the presence of continuous-time white noise processes $\{u(\cdot), v(\cdot)\}$. Engineers use these obviously nonphysical processes as approximations to white band noise processes. Here we note only that there are also certain mathematical issues in the treatment of white noise processes in the standard theory of stochastic processes, which mean also that the process $x^2(t)$ in Eq. (113) cannot be directly handled in the conventional theory. Special definitions of stochastic integrals have to be introduced and equations such as Eq. (113) have to be regarded as a shorthand for a more formal (integral) version. For linear least-mean-squares estimation problems, there is no need to introduce this more formal theory, and one can proceed quite satisfactorily with the now-usual methods of working with white noise processes. The problem is analogous to avoiding the need for delta functions in deterministic system theory by first working with step functions and then taking (formal) derivatives—engineers (and others) have long since learned to work comfortably with delta functions. The same holds for studies of stochastic linear systems.

Filtered Estimators

To proceed here, we note that there are a couple of ways of approaching the study of continuous-time problems—directly or by reduction to an equivalent, or more often an approximate, discrete-time problem. Here we proceed directly. Direct continuous-time solutions are quite feasible and in fact quite straightforward when we use the innovations,

$$e(t) = y(t) - H(t)\hat{x}(t) = H(t)\tilde{x}(t) + v(t)$$

where $\hat{x}(t) \triangleq \hat{x}(t|t_-)$ the linear least-mean-squares estimator of $x(t)$ given $\{y(\tau), 0 \le \tau < t\}$
$\tilde{x}(t)\ x(t) - \hat{x}(t)$

The innovations can be shown to be computed via

$$e(t) = y(t) - H(t)\hat{x}(t), \quad e(0) = y(0) \tag{88}$$

$$\dot{\hat{x}}(t) = F(t)\hat{x}(t) + K(t)e(t), \quad t \ge 0 \tag{89}$$

where

$$K(t) = [P(t)H^*(t) + G(t)S(t)]R^{-1}(t) \tag{90}$$

and

$$\begin{aligned} \dot{P}(t) &= F(t)P(t) + P(t)F^*(t) + G(t)Q(t)G^*(t) - K(t)R(t)K^*(t) \\ P(0) &= \Pi_0 \end{aligned} \tag{91}$$

We should note that the above formulas require that $R(\cdot)$, the intensity of the measurement noise process $v(\cdot)$, is invertible, that is, strictly positive-definite. This is in contrast to the discrete-time case, where it was only needed that $R_i \ge 0$, where $R_i\delta_{ij} = Ev_i v^*_j$; the quantity that needed to be invertible was $R_{e,i} = R_i + H_iP_iH^*_i$. In the continuous-time case, it is an interesting and important fact that

$$Ee(t)e^*(s) = R(t)\delta(t-s) = Ev(t)v^*(s)$$

While one can of course study continuous-time problems where $R(t)$ is not strictly positive-definite, the solution will generally involve derivatives of the observed process $y(t)$ and will therefore tend to be more sensitive to errors of various kinds.

Moreover, the nonlinear matrix Riccati differential Eq. (120) can rarely be solved analytically when $n \neq 1$ (the state dimension). In general, it will have to be solved numerically, which is facilitated to some extent by the fact that it is an initial value equation.

Also, the continuous-time formulas are somewhat simpler than those in discrete-time. For one thing, in continuous time, there is no distinction between predicted and filtered estimators.

Smoothed Estimators

The continuous-time version of the BF smoothing formulas take the following form. The smoothed estimator $\hat{x}(t|T)$ can be found via

$$\hat{x}(t|T) = \hat{x}(t) + P(t)\lambda(t|T), \quad 0 \le t \le T \tag{92}$$

where $\lambda(t|T)$ satisfies the backward time recursion

$$\begin{aligned} \dot{\lambda}(t|T) &= -[F(t) - K(t)H(t)]^*\lambda(t|T) - H^*(t)R^{-1}(t)e(t) \\ \lambda(T|T) &= 0 \end{aligned} \tag{93}$$

An alternative equation is

$$\begin{aligned} \dot{\lambda}(t|T) = {} & -F(t)^*\lambda(t|T) - H^*(t)R^{-1}(t)H(t)\hat{x}(t) \\ & - H^*(t)R^{-1}(t)y(t), \quad \lambda(T|T) = 0 \end{aligned} \tag{94}$$

Moreover, the smoothed error variance can be computed as

$$P(t|T) = P(t) - P(t)\int_t^T \Phi^*(s,t)H^*(s)R^{-1}(s)H(s)\Phi(s,t)\,ds \tag{95}$$

where $\phi(s,t)$ is the state transition matrix of the closed-loop filter $F(s) - K(s)H(s)$. The quantities $\{\hat{x}(t), e(t), K(t), P(t)\}$ are found in a forward pass by running the Kalman filter equations over the interval $[0, T]$.

Likewise, the RTS version of the smoothing filter is the following. We find the smoothed estimator $\hat{x}(t|T)$ by solving, backward in time, the equation

$$\begin{aligned} \dot{\hat{\boldsymbol{x}}}(t|T) &= F_s(t)\hat{\boldsymbol{x}}(t|T) + G(t)Q(t)G^*(t)P^{-1}(t)\hat{\boldsymbol{x}}(t|T) \\ \hat{\boldsymbol{x}}(T|T) &= \hat{\boldsymbol{x}}(T) \end{aligned} \tag{96}$$

where $F_s(t) = F(t) + G(t)Q(t)G^*(t)P^{-1}(t)$. The smoothing errors variance obeys the equation

$$\begin{aligned} \frac{dP(t|T)}{dt} &= F_s(t)P(t|T) + P(t|T)F_s^*(t) - G(t)Q(t)G^*(t) \\ P(T|T) &= P(T) \end{aligned} \tag{97}$$

Fast Algorithms

As in the discrete-time case, the effort required to solve the Riccati differential equation [Eq. (120)] is the same whether the model is time-variant or not. By one measure, we have to solve $n(n + 1)/2$ [since $P(\cdot)$ is Hermitian] coupled nonlinear differential equations for the entries of $P(\cdot)$.

To exploit the constancy of the state-space model we have to find a way of computing the gain function $K(\cdot)$ in Eq. (119) that does not require the computation of $P(\cdot)$. This is possible using a fast algorithm.

We factor $P^2(0)$ as $P^2(0) = L_0JL^*{}_0$, where $J = (I_p \oplus -I_q)$ is the signature of $P^2(0)$, that is, p is the number of positive eigenvalues of $P^2(0)$, whereas q is the number of negative eigenvalues. Then

$$\begin{aligned} \alpha \triangleq p + q &= \text{the rank of } \dot{P}(0) \\ &= \text{rank of } (F\Pi_0 + \Pi_0 F^* + GQG^* - \Pi_0 H^* R^{-1} H \Pi_0) \end{aligned}$$

and the gain matrix $K(\cdot)$ of the Kalman filter can be computed by solving the following set of $n(p + \alpha)$ coupled nonlinear equations

$$\dot{K}(t) = L(t)JL^*(t)H^*R^{-1} \tag{98}$$

$$\dot{L}(t) = [F - K(t)H]L(t) \tag{99}$$

with initial conditions $K(0) = \Pi_0 H^* R^{-1}$ and $L(0) = L_0$.

APPROXIMATE NONLINEAR FILTERING

Most practical systems are nonlinear to some extent and sometimes an idealized linear model suffices to describe the system. But very often the nonlinearities cannot be disregarded. Examples are nonlinear plant dynamics in control problems, perhaps due to actuator saturation or to a nonlinear measurement process. Another example is in the communication task of the demodulation of frequency- or phase-modulated signals in additive Gaussian white noise, with the modulating signal assumed Gaussian.

Let us consider the case of frequency modulation (*FM*) where the message $\lambda(t)$ has a first-order Butterworth spectrum, being modeled as the output of a first-order, time-invariant linear system with one real pole driven by continuous-time white noise. This message is then passed through an integrator to yield $\theta(t) = \int^{t}_{0} \lambda(\tau)d\tau$, which then is employed to phase modulate a carrier signal with carrier frequency ω_c rad/s. The model state equations can then be written as

$$\begin{bmatrix} \dot{\lambda}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} -1/\beta & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda(t) \\ \theta(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \boldsymbol{u}(t) \tag{100}$$

$$\boldsymbol{y}(t) = \sqrt{2}\sin[\omega_c t + \theta(t)] + \boldsymbol{v}(t) \tag{101}$$

for some noise disturbances $v(t)$ and $u(t)$ and some $\beta > 0$. The equation for the state is linear, but the measurement equation is nonlinear.

A more general nonlinear state-space model in continuous time can be one of the form

$$\dot{\boldsymbol{x}}(t) = f_t[\boldsymbol{x}(t)] + g_t[\boldsymbol{x}(t)]\boldsymbol{u}(t) \tag{102}$$

$$\boldsymbol{y}(t) = h_t[\boldsymbol{x}(t)] + \boldsymbol{v}(t) \tag{103}$$

where $u(t)$ and $v(t)$ are white noise signals, and $\{f_t(\cdot), g_t(\cdot), h_t(\cdot)\}$ are time-variant nonlinear functions. Regardless of the model, the least-mean-squares estimator of the state vector $x(t)$, at any particular time instant t, is given by the conditional mean

$$E[\boldsymbol{x}(t)|\boldsymbol{Y}(t)],\ \boldsymbol{Y}(t) = \{\boldsymbol{y}(\sigma),\ 0 < \sigma < t\} \tag{104}$$

In general, the result is too complicated to be of practical interest, with rare cases where implementations are possible. For this reason, we often resort to approximations on two levels. First, we discretize the continuous system, thus leading to a nonlinear discrete-time model of the general form

$$\boldsymbol{x}_{i+1} = f_i(\boldsymbol{x}_i) + g_i(\boldsymbol{x}_i)\boldsymbol{u}_i \tag{105}$$

$$\boldsymbol{y}_i = h_i(\boldsymbol{x}_i) + \boldsymbol{v}_i \tag{106}$$

with $\{f_i(\cdot), h_i(\cdot), g_i(\cdot)\}$ nonlinear in general, u_i, v_i are zero mean, white processes, and x_0 is a random variable with mean $\bar{x}_0$. We shall assume $\{u_i\}$, $\{v_i\}$, and x_0 are mutually uncorrelated, and that

$$\left\langle \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{v}_i \\ \boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0 \end{bmatrix}, \begin{bmatrix} \boldsymbol{u}_j \\ \boldsymbol{v}_j \\ \boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0 \end{bmatrix} \right\rangle = \begin{bmatrix} Q_i\delta_{ij} & 0 & 0 \\ 0 & R_i\delta_{ij} & 0 \\ 0 & 0 & \Pi_0 \end{bmatrix} \tag{107}$$

Second, we employ linear estimation techniques to develop estimators for the discretized state vector x_i. The Kalman solution offers an attractive alternative if the model could be further linearized. This creates several possibilities but we only discuss here two of the most widely used.

A Linearized Kalman Filter

The first possibility is to linearize the state-space equations [Eqs. (136) and (137)] around a known nominal trajectory $x^{\text{nom}}{}_{\text{i}}$. A common choice is the unforced solution,

$$x_{i+1}^{\text{nom}} = f_i(x_i^{\text{nom}}), \quad x_0^{\text{nom}} = \bar{x}_0 \tag{108}$$

This defines a deterministic sequence and we can write

$$\boldsymbol{x}_i = x_i^{\text{nom}} + \boldsymbol{\Delta x}_i \tag{109}$$

where Δx_{i} measures the perturbation away from the nominal trajectory and is a random variable.

Assuming the functions $\{f_{\text{t}}, g_{\text{t}}, h_{\text{t}}\}$ are smooth enough, and making a first-order Taylor expansion, we obtain

$$f_i(\boldsymbol{x}_i) \approx f_i(x_i^{\text{nom}}) + F_i\boldsymbol{\Delta x}_i, h_i(\boldsymbol{x}_i) \approx h_i(x_i^{\text{nom}}) + H_i\boldsymbol{\Delta x}_i \tag{110}$$

where the matrices F_{i} and H_{i} are defined by

$$F_i = \left.\frac{\partial f_i(x)}{\partial x}\right|_{x=x^{\text{nom}}} \quad H_i = \left.\frac{\partial h_i(x)}{\partial x}\right|_{x=x^{\text{nom}}} \tag{111}$$

This means that the (k, j)th component of F_{i} is the partial derivative of the kth component of $f_{\text{i}}(\cdot)$ with respect to the j-th component of x, and similarly for H_{i}, each derivative being evaluated at $x^{\text{nom}}{}_{\text{i}}$. Likewise, taking a zeroth order expansion leads to

$$g_i(\boldsymbol{x}_i) \approx g_i(x_i^{\text{nom}}) \overset{\Delta}{=} G_i \tag{112}$$

Then, it can be shown that an approximate estimator for the state x_{i} can be recursively computed as follows. Start with $\hat{x}_{0|-1} = \bar{x}_0, P_{0|-1} = \Pi_0$ and repeat:

$$\hat{\boldsymbol{x}}_{i+1|i} = F_i(\hat{\boldsymbol{x}}_{i|i} - x_i^{\text{nom}}) + f_i(x_i^{\text{nom}}) \tag{113}$$

$$\hat{\boldsymbol{x}}_{i|i} = \hat{\boldsymbol{x}}_{i|i-1} + K_{f,i}[\boldsymbol{y}_i - h_i(x_i^{\text{nom}}) - H_i\hat{\boldsymbol{x}}_{i|i-1} + H_i x_i^{\text{nom}}] \tag{114}$$

$$K_{f,i} = P_{i|i-1}H_i^*(H_iP_{i|i-1}H_i^* + R_i)^{-1} \tag{115}$$

$$P_{i|i} = (I - K_{f,i}H_i)P_{i|i-1} \tag{116}$$

$$P_{i+1|i} = F_iP_{i|i}F_i^* + G_iQ_iG_i^* \tag{117}$$

The performance of the linearized filter is clearly dependent on the quality of the approximation in Eqs. (141) to (143). If Eqs. (141) to (143) are exact, the linearized filter produces the true conditional mean estimator. Moreover, for small i, or small $\|g(x_{\text{i}})u_{\text{i}}\|$, the nominal solution may be close to the true trajectory. However, with time the two will depart, often resulting in a breakdown of Eqs. (141) to (143) and filter divergence.

The Extended Kalman Filter

A second possibility for the estimation of the state vector of the nonlinear model [Eqs. (136) and (137)] is to linearize the model around the most recent available estimate, that is, around $\hat{x}_{\text{i}|i}$ or $\hat{x}_{\text{i}|i-1}$ [here, $\hat{x}_{\text{i}|i}$ denotes the estimate while the boldface notation $\hat{x}_{\text{i}|i}$ denotes the estimator.] Hence, we define

$$f_i(\boldsymbol{x}_i) \approx f_i(\hat{x}_{i|i}) + F_i(\boldsymbol{x}_i - \hat{x}_{i|i}) \tag{118}$$

$$h_i(\boldsymbol{x}_i) \approx h_i(\hat{x}_{i|i-1}) + H_i(\boldsymbol{x}_i - \hat{x}_{i|i-1}) \tag{119}$$

$$g_i(\boldsymbol{x}_i) \approx g_i(\hat{x}_{i|i}) \overset{\Delta}{=} G_i \tag{120}$$

$$F_i = \left.\frac{\partial f_i(x)}{\partial x}\right|_{x=\hat{x}_{i|i}} \quad H_i = \left.\frac{\partial h_i(x)}{\partial x}\right|_{x=\hat{x}_{i|i-1}} \tag{121}$$

Then, it can be shown that an approximate estimator for the state x_{i} can be recursively computed by using the so-called Extended Kalman Filter (*EKF*). We start with $\hat{x}_{0|-1} = \bar{x}_0, P_{0|-1} = \Pi_0$ and repeat

$$\hat{\boldsymbol{x}}_{i+1|i} = f_i(\hat{\boldsymbol{x}}_{i|i}) \tag{122}$$

$$\hat{\boldsymbol{x}}_{i|i} = \hat{\boldsymbol{x}}_{i|i-1} + K_{f,i}[\boldsymbol{y}_i - h_i(\hat{\boldsymbol{x}}_{i|i-1})] \tag{123}$$

$$K_{f,i} = P_{i|i-1}H_i^*(H_iP_{i|i-1}H_i^* + R_i)^{-1} \tag{124}$$

$$P_{i|i} = (I - K_{f,i}H_i)P_{i|i-1} \tag{125}$$

$$P_{i+1|i} = F_iP_{i|i}F_i^* + G_iQ_iG_i^* \tag{126}$$

Contrary to the linearized Kalman filter, observe now that the matrices $(F_{\text{i}}, H_{\text{i}}, G_{\text{i}})$ depend on the measurements and, therefore, the quantities $(P_{\text{i}}, K_{\text{f},i})$ cannot be precomputed. This represents an increased computational load. Moreover, while the linearized Kalman filter depended linearly on the $\{y_{\text{i}}\}$, this is not the case any more for the extended Kalman filter since $K_{\text{f},i}$ also depends nonlinearly on prior measurements.

The convergence of the both filter variants cannot be guaranteed in general and it needs to be verified by simulation. Table 1 provides some indication as to which implementation to choose under conditions on the duration of the estimation interval, the size of the process noise, the signal-to-noise (*SNR*) ratio, and the uncertainty in the initial state vector.

Table 1. Rules of Thumb for Choosing Between EKF and Linearized KF

Condition	Large	Small
Estimation interval	EKF	Linearized
Process noise $g_i\mathbf{u}_i$	EKF	Linearized
SNR	EKF	Linearized
Initial uncertainty Π_0	EKF	Linearized

Of course, higher order filters can be obtained by retaining more terms in the Taylor series. However, they are not necessarily better than an EKF. Also, more sophisticated filters can be developed that are based on Gaussian sum approximations, statistical linearization, spline approximations, and so on.

CONCLUDING REMARKS

The intent of this article was to provide an overview of some of the fundamentals of state-space estimation, with emphasis on array formulations of the varied algorithms (slow or fast) that are available for this purpose. More details and related discussion can be found in several of the references indicated at the end of this article. The references are not intended to be complete but rather indicative of the work in the different areas. More complete lists can be found in several of the textbooks mentioned therein.

Practical Issues

In concluding, we briefly comment on several issues that are relevant in practical implementations of the Kalman filtering algorithm, and which often lead to erroneous behavior. By erroneous we mean that the performance of the filter diverges from the optimal performance that is predicted by the underlying theory. In particular, the observed state-error covariance matrix may tend to assume values that are either considerably larger than the values predicted by the solution of the Riccati equation or even negative-definite. Either case can lead to practical results that are far from ideal and therefore deserve closer examination.

There are many causes for the difficulties that arise when Kalman filtering algorithms are implemented in practice and numerous studies have appeared in the literature. Here, we mention some of the more significant issues in our opinion.

Recall that the Kalman solution allows us to estimate the state vector of a given state-space model under certain assumptions on the measurement and process noise sequences. The solution is optimal in the least-mean squares sense as long as the matrices that describe the dynamics of the model and the statistics of the noise processes are known exactly. Any modeling errors in these matrices can lead to a filter design whose actual performance does not agree with the theoretical performance. This is because the Kalman filter does not include any mechanism that allows it to compensate for such inaccuracies in the model.

Modeling errors can occur in many different forms. Apart from actual errors in the model dynamics and in the noise statistics, as mentioned above, we may also face errors that are due to unmodeled bias terms. This situation arises when either the state equation or the output equation is driven by unknown terms. By employing a Kalman filter that simply ignores the presence of these terms, we may obtain estimation errors that are unacceptably large.

A third example of modeling errors arises when some modes of the actual system are ignored, either deliberately or not. While reduced-order models lead to filter structures that are less demanding computationally than a full scale model, they nevertheless can still lead to erroneous performance.

In addition to modeling errors, a second cause of suboptimal performance that may occur in practical implementations of Kalman filtering algorithms is the use of suboptimal gain matrices. Although the Kalman filter is an optimal estimator, it is nevertheless inherently time-variant even when the underlying state-space model itself is time-invariant. This means that the associated time-variant gain matrices need to be repeatedly computed, and also stored in the case of off-line implementations. A significant reduction in computations and complexity can be achieved if the optimal gain matrices are replaced by a constant gain matrix, at the expense of suboptimal performance.

A related issue of practical relevance is how to estimate the value of the optimal steady-state Kalman gain, especially when the noise covariance matrices are unknown, and how to tune the filter so that its performance approaches the theoretical limit in the presence of these uncertainties.

Another cause of erroneous performance which is not immediately related to modeling errors, is the occurrence of round-off errors when Kalman filtering algorithms are implemented in finite-precision arithmetic. Since binary representations of real numbers cannot cover the entire range of real values, numerical errors are bound to occur due to overflow and underflow. Moreover, the larger the matrix dimensions, the larger the number of floating point operations that are needed and, consequently, the larger the possibility of round-off errors. These errors can cause Kalman filtering implementations to diverge away from their ideal behavior. In particular, numerical errors may cause the computed state-error covariance matrix to become negative definite, a situation that motivated us to discuss the class of square-root algorithms. These array-based algorithms help ameliorate numerical problems by working with square-root factors of covariance matrices, but may not resolve them completely for ill-conditioned problems. Studies on numerically reliable implementations exist in the literature, but we omit the details here for brevity.

ACKNOWLEDGMENT

This research was supported in part by the Army Research Office under grant DAAH04-93-G-0029 and by the National Science Foundation under grants MIP-9796147 and CCR-9732376.

BIBLIOGRAPHY

Kalman published his first paper on the discrete-time problem in 1960:

R. E. Kalman A new approach to linear filtering and prediction problems, *Trans. ASME J. Basic Eng.*, **82**: 34–45, 1960.

The continuous-time version of the state-space estimation problem was published in 1961:

R. E. Kalman R. S. Bucy New results in linear filtering and prediction theory, *Trans. ASME J. Basic Eng.*, **83**: 95–107, 1961.

Several textbooks and publications exist that now treat Kalman filtering and its variants in some detail:

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

A. P. Sage J. L. Melsa *Estimation theory with Applications to Comuniations and Control*, New York: McGraw-Hill, 1971.

T. Kailath A view of three decades of linear filtering theory, *IEEE Trans. Inf. Theory*, **20**: 146–181, 1974.

B. D. O. Anderson J. B. Moore *Optimal Filtering*, Englewood Cliffs, NJ: Prentice-Hall, 1979.

T. Kailath *Lectures on Wiener and Kalman Filtering*, 2nd ed., New York: Springer-Verlag, 1981.

M. S. Grewal A. P. Andrews *Kalman Filtering: Theory and Practice*, Englewood Cliffs, NJ: Prentice-Hall, 1993.

T. Kailath A. H. Sayed B. Hassibi *Linear Estimation*, Upper Saddle River, NJ: Prentice-Hall, 2000.

A treatment of array algorithms can be found in:

M. Morf T. Kailath Square root algorithms for least squares estimation, *IEEE Trans. Autom. Control*, **20**: 487–497, 1975.

Articles addressing several practical issues of Kalman filtering implementations, as well as applications, can be found in the following book:

H. W. Sorenson, (ed.) *Kalman Filtering: Theory and Application*, New York: IEEE Press, 1985.

There are several other problems, in addition to the state-space estimation problem of the earlier sections, that lend themselves rather directly to the methods discussed in this article. Some of these include applications in adaptive filtering, quadratic control, and H_∞ filtering and control. Discussions in this regard and numerous related references on this rich subject can be found in the following:

T. Kailath A. H. Sayed B. Hassibi *Linear Estimation*, Upper Saddle River, NJ: Prentice-Hall, 2000.

B. Hassibi A. H. Sayed T. Kailath *Indefinite Quadratic Estimation and Control: A Unified Approach to H_2 and H_∞*, PA: SIAM, 1999.

A. H. Sayed T. Kailath A state-space approach to adaptive RLS filtering, *IEEE Signal Proc. Mag.*, **11** (3): 18–60, 1994.

THOMAS KAILATH
ALI H. SAYED
BABAK HASSIBI
Stanford University, Stanford, CA
University of California at Los Angeles, Los Angeles, CA
California Institute of Technology, Stanford, CA