10

Kalman Filtering

Once you get the physics right, the rest is mathematics.
Rudolf E. Kalman (1930–2016) Kailath Lecture,
Stanford University, 11 May 2008

10.1 Chapter Focus

The primary purpose of this chapter is to provide a working familiarity with Kalman filtering – both the theoretical and practical aspects of it, and especially those features essential for global navigation satellite system (GNSS) navigation, inertial navigation performance analysis (in Chapter 11), and performance analysis of integrated GNSS/INS navigation (in Chapter 12).

The next section (10.2) is a heuristic introduction to Kalman filtering, primarily intended to give the uninitiated an appreciation of its place in the history of technology. This is followed by a section defining some mathematical notation needed in the rest of the chapter, beginning with a section on the mathematical foundations of the Kalman filter. Although people who like Kalman filters and sausages need not watch them being made, they should benefit from understanding some of their ingredients. In the case of the Kalman filter, a key ingredient is the linear least mean squares estimator (LLMSE) first developed by Carl Friedrich Gauss (1777–1855) in the 1820s [1] and extended to include real-time estimation (i.e. filtering) by the late Rudolf Emil Kalman in the 1950s [2]. Understanding the LLMSE is key to understanding many essential attributes of Kalman filtering, including the fact that it does not depend on any statistics of the underlying error probability density functions (PDFs) beyond their means and covariances. We start with the LLMSE to establish these attributes, then present a brief overview of how Kalman (and others) extended this to develop his famous filter.

We also present alternative but equivalent mathematical representations of the Kalman filter that have been found to be more robust against computer

Global Navigation Satellite Systems, Inertial Navigation, and Integration,
Fourth Edition. Mohinder S. Grewal, Angus P. Andrews, and Chris G. Bartone.

Companion website: www.wiley.com/go/grewal/gnss

roundoff errors, especially for applications with large numbers of unknowns to be estimated. This can be critical for maintaining accuracy in large-scale applications such as GNSS system state estimation at the constellation level.

Some of the more successful approaches used for approximating nonlinear transformations of means and covariances in Kalman filtering are also presented. Appendix C (on www.wiley.com/go/grewal/gnss) examines the "PDF ambiguity errors" this introduces with only means and covariances to start with.

For GNSS receiver applications, we show how to represent the random dynamical characteristics of various host vehicles in state-space form for using Kalman filters in vehicle tracking, and assess the relative statistical significance of linearization errors in those applications.

We conclude with some methods used for monitoring Kalman filter operation to detect and diagnose anomalous behaviors.

10.2 Frequently Asked Questions

1. **What is a Kalman filter?**
   In this context, a *filter* is a real-time estimator. The Kalman filter uses noisy measurements to track the status of dynamic systems with random disturbances.
   Its popularity is due, in large part, to the fact that so many practical estimation problems can be modeled and solved in this way, and the fact that the Kalman filter has performed so well in so many of these applications.
   It has been justifiably labeled "navigation's integration workhorse" [3] because it has become an essential part of modern navigation systems – especially for integrating navigation systems as disparate as GNSS and INS (inertial navigation system). Kalman filtering is the connective tissue binding satellite navigation systems together.
2. **What is it used for?**
   Its principal uses in navigation are:
   (a) *In performance-predictive design of sensor systems* used for estimating the current state of a particular dynamic system in a proposed application. A subset of the Kalman filter implementation, called the *matrix riccati equations*, determines how well the state of the dynamic system can be estimated, given certain attributes of the sensors to be used. For example, it has been used for comparing the relative performance of various configurations of GNSS satellite constellations, locations and operations of supportive ground stations, and receiver hardware and software configurations. It is also used to determine *if* the unknown state variables can be uniquely determined – and how well – from a given set of measurements. This is the issue of *observability*.

(b) *As an embedded real-time estimator* for a particular application. Kalman filters in one form or another are in current use throughout GNSS systems to maintain their accuracy, and in receivers with Kalman model parameters representing the host vehicle dynamics, receiver signal characteristics, and any auxiliary sensors used.

3. **How does it work?**

The Kalman filter has been called "ideally suited to digital computer implementation" [4], in part because it represents the estimation problem using algebraic expressions with a finite number of variables and parameters. It does, however, assume that these variables are *real numbers* – with *infinite* precision. Some of the problems encountered in its use arise from this distinction. These are all issues on the practical side of Kalman filtering that must be considered along with the theory:

(a) *Representing uncertainty.*

Generally, a *random number* or *variate* is a variable whose value may be random but whose probability distribution is known to some degree. The truly remarkable thing about Gauss's LLMSE is that the solution depends only on two parameters of the probability distributions of the random variates involved. It is otherwise independent of other attributes of the underlying error distributions.

i. The first of these parameters is the *mean value* of a probability distribution. If the variate is a vector, the mean will also be a vector.

ii. The second parameter is the *mean-squared deviation about the mean,* which will be a matrix if the variate is a vector. It is also called the *covariance* about the mean of the distribution or the *covariance of uncertainty* in the estimate, because it characterizes all the cross correlations among the components of vector variates.

These two parameters are used for characterizing each of three probability distributions in Kalman filtering:

i. The distributions of *measurement errors,* in which case the means are usually assumed to be zero, but can otherwise be treated as known measurement biases and subtracted from the sensor outputs.

ii. The distributions of *random dynamic disturbances,* the means of which are also assumed to be zero. Otherwise, any nonzero mean can be treated as a known dynamic disturbance.

iii. The distribution of *estimation errors,* in which case equations for maintaining means and covariances can be further subdivided into two classes:

A. Matrix equations for maintaining the covariance of estimation errors are collectively called *the riccati equations,* a reference to a nonlinear differential equation studied by the Italian mathematician Jacopo Riccati (1676–1754) and shown to be equivalent to a pair of linear differential equations – an attribute shared with its matrix counterpart. It was Kalman's colleague Richard S. Bucy who recognized that the nonlinear differential equations for covariance propagation in the Kalman filter had the form of Riccati's equation. In Kalman filtering, the solution to the riccati equations does not depend on the estimate, which is why the riccati equations can be used independently for assessing the expected performance of a proposed application defined by its stochastic dynamic model and the performance specifications and measurement schedules for the proposed sensor suite.

B. *The Kalman filter is essentially an estimator for the mean of the distribution of the unknown variate,* in which case the estimation errors will have zero mean. The filter uses the covariance of estimation errors from the solution to the riccati equations described earlier to maintain the linear least mean squares estimate of the unknown variate. This part of the Kalman filter implementation depends on the solution of the riccati equations, but *the riccati equations do not depend on the estimate.*

(b) *Dependence on linearity.*

Transformations of the means and covariances in its implementation depend on linearity of the dynamics and measurements. Although several approximations for not-quite-linear applications have been derived, applied, and evaluated, the resulting means and covariances from nonlinear transformations of the variates generally depend on other attributes of the starting probability distributions beyond just the mean and covariance. Even assuming that the starting distributions are Gaussian is inadequate, because nonlinear transformations of Gaussian distributions do not preserve gaussianity.

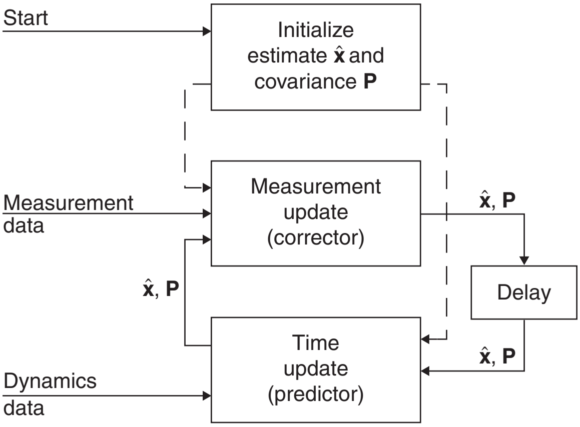
(c) *Linear algebra.*

The Kalman filter is based on linear algebraic procedures for the addition and multiplication of vectors and matrices, and inversion of matrices. This makes it particularly easy to implement in MATLAB®.

4. **How is it implemented?**

It is called a "*predictor–corrector*" algorithm because it propagates the linear least mean squares estimate $\hat{\mathbf{x}}$ and its covariance of estimation errors $\mathbf{P}$ (also called the covariance of estimation *uncertainty*) forward in the time between measurements, predicting the estimate and its covariance of estimation uncertainty before the next measurement is used. Then the results of the measurement(s) are used to correct the predicted values to reflect the influence of the information gained from the new measurements.

The top-level data flow structure of the implementation is shown in block form in Figure 10.1, where the box titled "Measurement update (corrector)" uses as its inputs the estimated state variable $\hat{\mathbf{x}}$ prior to using the measurements and the associated covariance matrix $\mathbf{P}$ of prior state



Start
Initialize estimate $\hat{\mathbf{x}}$ and covariance **P**
Measurement data
Measurement update (corrector)
$\hat{\mathbf{x}}$, **P**
Delay
$\hat{\mathbf{x}}$, **P**
$\hat{\mathbf{x}}$, **P**
Time update (predictor)
Dynamics data

Figure 10.1 Top-level date flow diagram of Kalman filter implementation.

estimation uncertainty. Inputs also include measurement data in the form of the vector of measurement values **z**, its associated matrix **H** of linear sensitivities to the unknown dynamic state variable **x**, and the associated covariance matrix **R** of measurement noise. Outputs include the estimated state variable $\hat{\mathbf{x}}$ after using the measurements and the covariance matrix **P** of state estimation uncertainty corrected to reflect the improvement in estimation uncertainty resulting from the used of measurements. If no usable measurements are available, no correction is made, which allows for testing and rejecting anomalous measurements.

The box titled "Delay" represents the time delay between one set of measurements and the next, which is not necessarily constant and can be zero.

The box titled "Time update[1] (predictor)" uses as its inputs the estimated state variable $\hat{\mathbf{x}}$ after using the last set of measurements, and the associated covariance matrix **P** of estimation uncertainty. Inputs also include parameters defining the changes in the estimate and its associated uncertainty due to the passage of time. These input parameters include the linear transformation matrix $\mathbf{\Phi}$ modeling the deterministic dynamics of the state vector **x** over the time period since the last measurement update and the matrix **Q** representing the covariance matrix of additional dynamic uncertainty resulting from random disturbances of the state vector during that period. Outputs of the predictor include the time-updated estimate $\hat{\mathbf{x}}$ of the dynamic state vector **x** and its associated covariance matrix **P** of estimation uncertainty.

Activities required for starting the Kalman filter are represented by the box title "Initialize estimate $\hat{\mathbf{x}}$ and covariance **P**," which gives initial values of the estimate $\hat{\mathbf{x}}$ and its associated covariance matrix of estimation uncertainty **P** to either the corrector or predictor – but not both. After initialization, program control loops between the predictor and corrector, the derivations of which are outlined in the following sections.

[1] Also called the "temporal update."

10.3 Notation

The Kalman filter makes extensive use of linear algebra, so it will be necessary to define some related mathematical conventions before we can start to use them.

10.3.1 Real Vectors and Matrices

10.3.1.1 Notation

We will use font, case, and face to distinguish between printed symbols used for different types of mathematical objects:

1. The symbol $\mathbb{R}$ represents the field of *real numbers*, also called *scalars*. Real numbers or scalars $r \in \mathbb{R}$ will be denoted by lowercase unbolded letters.
2. The symbol $\mathbb{R}^n$ represents n-dimensional Euclidean space, the set of n-dimensional real *vectors*, also called n-vectors. Unless specified otherwise, they are assumed to be *column vectors*. Vectors $\mathbf{v} \in \mathbb{R}^n$ will be denoted by lowercase **bold** letters.
   (a) The scalar element in the ith row of a column vector $\mathbf{v}$ is represented as v_i, unbolded and post-subscripted. If $\mathbf{v}$ is a row vector, v_i represents the element in the ith column of $\mathbf{v}$.
   (b) The post-superscript ^{T} denotes the *transpose* of a vector (represented by the post-apostrophe' in MATLAB). The transpose of a column vector is a row vector, and *vice versa*.
   (c) The letter $\mathbf{x}$ will be used to denote the state vector of a dynamic system being tracked by a Kalman filter.
   (d) The overbar $\overline{\mathbf{x}}$ will be used to denote the mean of the probability distribution of $\mathbf{x}$.
   (e) The "hatted" letter $\hat{\mathbf{x}}$ will be used to denote the *estimate* (or estimated mean) of the probability distribution of the state vector.
3. The symbol $\mathbb{R}^{\ell \times n}$ represents the set of $\ell \times n$ real matrices, where ℓ and n are positive integers. ℓ representing the number of rows and n representing the number of columns. Matrices $\mathbf{M} \in \mathbb{R}^{\ell \times n}$ will be denoted by uppercase **bold** letters.
   (a) The letter $\mathbf{P}$ will be used to denote the covariance of uncertainty (defined hereafter) of the estimate $\hat{\mathbf{x}}$ (defined earlier).
   (b) The post-superscript ^{T} denotes the *transpose* (apostrophe' in MATLAB) of a matrix:
   $$\{\mathbf{M}^T\}_{ij} = \{\mathbf{M}\}_{ji} = m_{ji}$$
   (c) The post-superscript $^{-1}$ denotes the *inverse* of a matrix – if it exists.
   (d) The post-superscript$^{1/2}$ in $\mathbf{P}^{1/2}$ denotes a *square root* of a symmetric positive-definite matrix $\mathbf{P}$, defined as a solution to the matrix equation
   $$\mathbf{P}^{1/2}\mathbf{P}^{T/2} = \mathbf{P} \quad \text{for} \quad \mathbf{P}^{T/2} \overset{\text{def}}{=} [\mathbf{P}^{1/2}]^T \tag{10.1}$$

(e) The post-superscript$^{-1/2}$ in $\mathbf{P}^{-1/2}$ denotes a *square root* of $\mathbf{P}^{-1}$.
(f) The scalar element in the ith row and jth column of a matrix **M** is represented as m_{ij}, where the m is lowercase and unbolded.
(g) The "dot" $(\cdot)$ will be used to indicate the full range of a subscript, so $\mathbf{m}_{\cdot j}$ will denote the jth column vector of the matrix **M** and $\mathbf{m}_{i\cdot}$ will denote the ith row vector of **M**. (Its equivalent in MATLAB is the colon, ":")

4. Vectors and matrices in mathematical expressions are assumed to be *conformably dimensioned* for the operations shown. A matrix product **AB**, for example, assumes the number of columns in **A** equals the number of rows in **B**.

10.3.1.2 Vector and Matrix Properties

Properties useful in Kalman filtering include the following:

Unit vectors **u** have unit length

$$\sqrt{\sum_i u_i^2} = 1$$

Zero matrices have zero (0) in all elements and are denoted by the zero symbol (0).

Square matrices have the same number of rows and columns. Non-square matrices are called *rectangular*.

Symmetric matrices are square matrices **M** such that $\mathbf{M}^T = \mathbf{M}$.

Positive definite matrices are symmetric $n \times n$ matrices **M** such that, for any nonzero column n-vector **v**,

$$\mathbf{v}^T\mathbf{M}\mathbf{v} > 0$$

For **M** *positive semi-definite*, $\mathbf{v}^T\mathbf{M}\mathbf{v} \geq 0$.

The **main diagonal** of a matrix **M** is that subset of elements m_{ii} with the same row and column index i. (Note that this applies to non-square matrices.)

The **trace** of a square matrix **M** is defined as the sum of its diagonal elements and denoted as $\mathrm{Tr}(\mathbf{M})$. The trace operator is linear, in that

$$\mathrm{Tr}(a\mathbf{A} + b\mathbf{B}) = a\mathrm{Tr}(\mathbf{A}) + b\mathrm{Tr}(\mathbf{B})$$

The trace of a matrix product is also invariant under commutation:

$$\mathrm{Tr}(\mathbf{AB}) = \mathrm{Tr}(\mathbf{BA}) \tag{10.2}$$

The trace of a matrix **M** is computed in MATLAB by the function `trace(M)`.

Diagonal matrices are square matrices **D** in which only the diagonal elements d_{ii} are nonzero. The notation $\mathrm{diag}(\mathbf{v})$ will denote the square diagonal matrix with diagonal elements specified by the vector **v**.

Identity matrices are diagonal matrices in which all diagonal elements equal 1. The symbol **I** will be used to denote an identity matrix.

Triangular matrices have zeros on one side or the other of the main diagonal.
Upper triangular matrices have zeros below the main diagonal.
Lower triangular matrices have zeros above the main diagonal.
Strictly triangular matrices have zeros on the main diagonal, as well.
Unit triangular matrices have ones on their main diagonal.

Determinants of $n \times n$ matrices $\mathbf{M}$ are real numbers equal to their *volumetric scaling*, which is the volume of the unit n-cube after transformation by $\mathbf{M}$. The determinant of $\mathbf{M}$ will be denoted by $\det \mathbf{M}$ and is computed in MATLAB by the function `det(M)`.

Eigenvalues or **characteristic values** of a matrix $\mathbf{M}$ are the the solutions λ of its *characteristic polynomial*

$$\det(\mathbf{M} - \lambda \mathbf{I}) = 0$$

Eigenvectors of a matrix $\mathbf{M}$ are the unit vector solutions of $\mathbf{Mu} = \lambda_i \mathbf{u}_i$ for the eigenvalues λ_i of $\mathbf{M}$.

Decomposition – in mathematics – refers to representing something as a sum or product. A square matrix $\mathbf{M}$, for example, can be represented as a sum of its symmetric and antisymmetric parts as

$$\mathbf{A} = \underbrace{(\mathbf{M} + \mathbf{M}^T)/2}_{\text{symmetric}} + \underbrace{(\mathbf{M} - \mathbf{M}^T)/2}_{\text{antisymmetric}}$$

Factorization refers to matrix decomposition as a product, perhaps done to distinguish it from biological terminology.

The **eigenvalue–eigenvector decomposition** of a symmetric matrix $\mathbf{M}$ is the representation

$$\mathbf{M} = \sum_i \lambda_i \mathbf{u}_i \mathbf{u}_i^T$$

where the λ_i are the eigenvalues of $\mathbf{M}$ and the $\mathbf{u}_i$ are the corresponding unit eigenvectors.

Eigendecomposition – or spectral decomposition – of a square matrix $\mathbf{M}$ refers to its factorization in the form

$$\mathbf{M} = \mathbf{E}\mathbf{D}_\lambda \mathbf{E}^{-1},$$

where the columns of $\mathbf{E}$ are the eigenvectors of $\mathbf{M}$ and the diagonal entries of $\mathbf{D}_\lambda$ are the corresponding eigenvalues.

Orthogonal matrices are nonsingular square matrices Ω such that

$$\mathbf{\Omega}^T = \mathbf{\Omega}^{-1} \quad \text{and} \quad \mathbf{\Omega}^T\mathbf{\Omega} = \mathbf{\Omega}\mathbf{\Omega}^T = \mathbf{I}$$

These have properties which are very useful in matrix computations:

- Products of $n \times n$ orthogonal matrices are also $n \times n$ orthogonal matrices and

$$[\mathbf{\Omega}_1\mathbf{\Omega}_2\mathbf{\Omega}_3 \ \cdots \ \mathbf{\Omega}_m]^T = \mathbf{\Omega}_m^{-1}\mathbf{\Omega}_{m-1}^{-1}\mathbf{\Omega}_{m-2}^{-1} \cdots \mathbf{\Omega}_1^{-1}$$

- If $\mathbf{P}^{1/2}$ is a square root of $\mathbf{P}$ (defined in Eq. (10.1)) and Ω is a conformable orthogonal matrix, then $\mathbf{P}^{1/2}\boldsymbol{\Omega}$ is also a square root of $\mathbf{P}$, because

$$\mathbf{P}^{1/2}\boldsymbol{\Omega}[\mathbf{P}^{1/2}\boldsymbol{\Omega}]^T = \mathbf{P}^{1/2}\boldsymbol{\Omega}\boldsymbol{\Omega}^T\mathbf{P}^{T/2} = \mathbf{P}^{1/2}\mathbf{I}\mathbf{P}^{T/2} = \mathbf{P} \tag{10.3}$$

  This implies that matrix square roots are not unique and are related by orthogonal matrices, which will have important consequences in square root implementations of the Kalman filter.
- Determinants of orthogonal matrices are ±1 and the absolute values of all eigenvalues are +1.

Singular values of a matrix $\mathbf{M}$ are the square roots of the eigenvalues of $\mathbf{M}\mathbf{M}^T$ (left singular values) and $\mathbf{M}^T\mathbf{M}$ (right singular values).

Singular value decomposition of an $n \times n$ symmetric positive-definite matrix $\mathbf{M}$ is a representation in the form

$$\mathbf{M} = \boldsymbol{\Omega}\mathbf{D}\boldsymbol{\Omega}^T$$

where the unit eigenvectors of $\mathbf{M}$ are the columns of the $n \times n$ orthogonal matrix $\boldsymbol{\Omega}$ and $\mathbf{D}$ is a diagonal matrix with the corresponding eigenvalues on its diagonal, ordered such that

$$d_{11} \geq d_{22} \geq d_{33} \geq \cdots \geq d_{nn} > 0$$

For any symmetric positive-definite matrix $\mathbf{M}$ the MATLAB function `[OmegaL,D,OmegaR] = svd(M)` returns $\boldsymbol{\Omega}_L$, $\mathbf{D}$, $\boldsymbol{\Omega}_R$ such that $\boldsymbol{\Omega}_L = \boldsymbol{\Omega}_R$.

Condition numbers for computational procedures generally refer to sensitivities of outputs to inputs. The condition number for inverting a square matrix $\mathbf{M}$ is the ratio of the largest singular value of $\mathbf{M}$ to the smallest singular value of $\mathbf{M}$. Larger condition numbers generally indicate poorer conditioning for inversion. The condition number for an orthogonal matrix is 1 (one), which is as good as it gets.

10.3.2 Probability Essentials

Kalman filtering is not about probabilities. It is about expected values of estimation errors and their squared values, but we need to use probabilities to explain what it is about.

10.3.2.1 Basic Concepts

Derivation of the Kalman filter equations requires only a few concepts from probability theory. To wit:

1. All probability distributions are assumed to be defined on n-dimensional Euclidean spaces $\mathbb{R}^n$ by integrable PDFs $p(\cdot)$ such that
   (a) $p(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$
   (b) $\int_{\mathbb{R}^n} p(\mathbf{x})dx = 1$

2. The *expected value* $\mathrm{E}_p\langle F\rangle$ of any real-, vector-, or matrix-valued function F of $\mathbf{x} \in \mathbb{R}^n$ is defined as

$$\mathrm{E}_p\langle F\rangle \overset{\text{def}}{=} \int_{\mathbb{R}^n} F(\mathbf{x})\, p(\mathbf{x})\, d\mathbf{x}$$

   where $d\mathbf{x} = dx_1\, dx_2\, dx_3\, \cdots\, dx_n$ and the expected value of a vector or matrix is the vector or matrix of the expected values of its elements. The subscript p on E represents the probability density function defining the operative expectancy operator, although this notation can be dropped after the quasi-independence of the LLMSE from full specification of PDFs has been established.
3. *Raw moments* of probability distributions on $\mathbb{R}^n$ are the expected values of outer-product powers of the variate. The Nth such raw moment is an N-dimensional data structures $\mathscr{D}$ with elements

$$\{\mathscr{D}\}_{j_1, j_2, j_3, \ldots, j_N} \overset{\text{def}}{=} \mathrm{E}\langle x_{j_1}\, x_{j_2}\, x_{j_3}\, \cdots\, x_{j_N}\rangle, \quad 1 \le j_i \le n, \quad 1 \le i \le N$$

4. The first-order raw moment is called the *mean* of the probability distribution defined by a PDF $p(\cdot)$ as the vector

$$\overline{\mathbf{x}} = \mathrm{E}_p\langle \mathbf{x}\rangle$$

5. *Central moments* are defined with respect to the mean as the raw moments of $\mathbf{x} - \overline{\mathbf{x}}$.
6. The second central moment of the probability distribution defined by $p(\cdot)$ would then be the $n \times n$ matrix
   (a) $\mathbf{P} = \mathrm{E}_p\langle[\mathbf{x} - \overline{\mathbf{x}}]\ [\mathbf{x} - \overline{\mathbf{x}}]^T\rangle$, and its trace (from Eq. (10.2))
   (b) $\mathrm{Tr}(\mathbf{P}) = \mathrm{E}_p\langle|\mathbf{x} - \overline{\mathbf{x}}|^2\rangle$, the mean squared value.

It should be noted that there are so-called "pathological" probability distributions (e.g. the Cauchy distribution) for which the mean and covariance are not defined. Otherwise, in the LLMSE, the underlying probability distribution p does not need to be specified beyond its mean and covariance about the mean. This is because the theory depends only on the mean and covariance about the mean and is otherwise independent of any other particulars of the probability distribution.

10.3.2.2 Linearity of the Expectancy Operator E⟨·⟩

An important property of the expectancy operator $\mathrm{E}\langle\cdot\rangle$ is that it is *linear*, in the sense that for any real numbers a and b, scalar functions f and g, vector-valued functions $\mathbf{f}$ and $\mathbf{g}$, real matrices $\mathbf{A}$ and $\mathbf{B}$, and matrix-valued functions $\boldsymbol{F}$ and $\boldsymbol{G}$,

$$\mathrm{E}\langle af + bg\rangle = a\mathrm{E}\langle f\rangle + b\mathrm{E}\langle g\rangle \tag{10.4}$$

$$\mathrm{E}\langle \mathbf{Af} + \mathbf{Bg}\rangle = \mathbf{A}\mathrm{E}\langle \mathbf{f}\rangle + \mathbf{B}\mathrm{E}\langle \mathbf{g}\rangle \tag{10.5}$$

$$\mathrm{E}\langle \mathbf{A}\boldsymbol{F} + \mathbf{B}\boldsymbol{G}\rangle = \mathbf{A}\mathrm{E}\langle \boldsymbol{F}\rangle + \mathbf{B}\mathrm{E}\langle \boldsymbol{G}\rangle \tag{10.6}$$

10.3.2.3 Means and Covariances of Linearly Transformed Variates

If $\mathbf{A}$ is any matrix, $\overline{\mathbf{x}}$ is the mean of a probability distribution on $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{P}$ is its covariance, then the mean and covariance of $\mathbf{Ax}$ will be

$$\begin{aligned} \mathrm{E}\langle \mathbf{Ax} \rangle &= \mathbf{A}\mathrm{E}\langle \mathbf{x} \rangle \\ &= \mathbf{A}\overline{\mathbf{x}} \quad \text{(mean)} \qquad (10.7) \\ \mathrm{E}\langle [\mathbf{Ax} - \mathbf{A}\overline{\mathbf{x}}][\mathbf{Ax} - \mathbf{A}\overline{\mathbf{x}}]^T \rangle &= \mathbf{A}\mathrm{E}\langle [\mathbf{x} - \overline{\mathbf{x}}][\mathbf{x} - \overline{\mathbf{x}}]^T \rangle \mathbf{A}^T \\ &= \mathbf{APA}^T \quad \text{(covariance)} \qquad (10.8) \end{aligned}$$

where $\mathbf{P}$ is the covariance of $\mathbf{x}$. *This will be very important in the derivation of the Kalman filter.*

10.3.3 Discrete Time Notation

10.3.3.1 Subscripting

The Kalman filter is defined in terms of a discrete time sequence of events,

$$t_0,\ t_1,\ t_2,\ t_3, \ldots,$$

where the real time sequence is nondecreasing,

$$t_0 \le t_1 \le t_2 \le t_3 \le \cdots$$

and the time interval between successive events is not necessarily constant. These discrete time values t_k generally represent the times at which measurements can be made but may also represent times when there is no measurement used but estimates are needed. These conditions may occur when a measurement is deemed to be in error and not usable, for example.

As a measure for mitigating subscript confusion, sub-arrays or elements of time-subscripted arrays can be represented by braces, so that $\{\mathbf{M}_k\}_{ij}$ denotes the element in the ith row and jth column of the time-subscripted matrix $\mathbf{M}_k$.

10.3.3.2 A Priori and A Posteriori Values

In the discrete time instance t_k of each Kalman filter measurement update there can be two values of the estimate $\hat{\mathbf{x}}$ and its associated covariance matrix of estimation uncertainty $\mathbf{P}$:

1. The *a priori* values of $\hat{\mathbf{x}}$ and $\mathbf{P}$ before the information in the measurement(s) is used, which will be denoted as $\hat{\mathbf{x}}_{k(-)}$ and $\mathbf{P}_{k(-)}$.
2. The *a posteriori* values of $\hat{\mathbf{x}}$ and $\mathbf{P}$ after the information in the measurement(s) has been used, which will be denoted as $\hat{\mathbf{x}}_{k(+)}$ and $\mathbf{P}_{k(+)}$.

10.3.3.3 Allowing for Testing and Rejecting Measurements

There is the possibility that a measurement vector $\mathbf{z}_k$ or some elements thereof at time t_k can be tested and rejected as being too anomalous to be acceptable without significant risk.

In the case that the entire measurement vector $\mathbf{z}_k$ is unacceptable, the a posteriori values of the estimate $\hat{\mathbf{x}}_{k(+)}$ and its covariance of uncertainty $\mathbf{P}_{k(+)}$ will be equal to their respective a priori values, $\hat{\mathbf{x}}_{k(-)}$ and $\mathbf{P}_{k(-)}$.

In the case that only some components of a measurement vector are deemed unusable, the suspect components of the measurement vector $\mathbf{z}_k$, the corresponding rows of $\mathbf{H}_k$, and the corresponding rows and columns of the associated covariance matrix $\mathbf{R}_k$ of measurement noise can be eliminated and their dimension ℓ_k reduced accordingly.

Methods for monitoring and testing are discussed in Section 10.7.

10.4 Kalman Filter Genesis

Kalman played a major role in a mid-twentieth century paradigm shift in engineering mathematics for estimation and control problems – from probability theory and spectral characteristics in the frequency domain to linear first-order differential equations in “state space” (Euclidean space) and matrix theory. In 1955, when the Glenn L. Martin Company founded its Research Institute for Advanced Studies (RIAS) just outside Baltimore, one of its major research areas would be in the mathematical theory for control of dynamic systems. Kalman joined RIAS in 1956, and in November of 1958 he posed for himself the problem of transforming Wiener–Kolmogorov “filtering” (real-time estimation) in the frequency domain to state-space form in the time domain. The effort would come to rely on mathematics beyond historical engineering curricula, but its result – about a year later – would be the Kalman filter [2], followed by major discoveries about the relationships between linear estimation and control problems in what has come to be called *mathematical systems theory* [5].

A brief historical overview of the mathematical origins of that transformation is presented in the following subsections 10.4.1–10.4.3, the main purpose of which is to demonstrate the special properties of Gauss’s LLMSE that have been inherited by Kalman filtering. This also honors some of the other major contributors to the mathematical development of Kalman filtering.

10.4.1 Measurement Update (Corrector)

The measurement update (also called the “observational update”) is where measurements related to the state vector are used to update the estimated state vector.

To really understand why the Kalman filter depends only on the first two moments of probability distributions it is necessary to start with one of its progenitors, the *LLMSE* of Gauss [1], and follow through some of its transformations to become the so-called “corrector” implementation of the Kalman filter.

10.4.1.1 Linear Least Mean Squares Estimation: Gauss to Kalman

Least Squares Gauss discovered his *method of least squares* in 1795, at the age of 18 [6], but he did not apply it to an estimation problem until 1801 and did not publish his method until 1809 [7]. In the meantime, his method of least squares had been discovered independently and published by Andrien-Marie Legendre (1752–1833) [8] in France and Robert Adrian (1775–1843) in the United States [9].

Gauss's method of least squares finds a solution to the overdetermined linear equation $\mathbf{z} = \mathbf{H}\mathbf{x}$ for $\mathbf{x}$, given $\mathbf{z}$ and $\mathbf{H}$. The solution for $\mathbf{x}$ that minimizes the squared error $|\mathbf{z} - \mathbf{H}\mathbf{x}|^2$ can be found by setting its derivative to 0:

$$\frac{d}{d\mathbf{x}}|\mathbf{z} - \mathbf{H}\mathbf{x}|^2 = 0$$

and solving the resulting *normal equation*

$$\underbrace{\mathbf{H}^T\mathbf{H}}_{\text{Gramian}}\,\mathbf{x} = \mathbf{H}^T\mathbf{z} \tag{10.9}$$

for the least-squares estimate

$$\hat{\mathbf{x}} = [\mathbf{H}^T\mathbf{H}]^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{z} \tag{10.10}$$

if the *Gramian*[2] *matrix* $\mathbf{H}^T\mathbf{H}$ is nonsingular.

Gauss–Markov Theorem and Homoscedasticity In [1], Gauss showed that the least squares solution is also the linear least mean squares solution if the *measurement error*

$$\mathbf{v} \overset{\text{def}}{=} \mathbf{z} - \mathbf{H}\mathbf{x} \tag{10.11}$$

is zero-mean and *homoskedastic*. That is, it has covariance

$$\begin{aligned} \mathrm{E}\langle \mathbf{v}\mathbf{v}^T \rangle &= \begin{bmatrix} \sigma^2 & 0 & 0 & \cdots & 0 \\ 0 & \sigma^2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma^2 \end{bmatrix} \\ &= \sigma^2\mathbf{I} \end{aligned} \tag{10.12}$$

where $\mathbf{I}$ is an identity matrix. This implies that the individual components of the measurement error vector $\mathbf{v}$ are statistically independent and all have the same variance σ^2. Note that *homoskedasticity* is a weaker constraint than that of *independent identically distributed*, a not-uncommon assumption in statistics.

[2] Named for Jørgen Pedersen Gram (1850–1916), also codiscoverer (with Erhard Schmidt (1876–1959)) of Gram–Schmidt orthogonalization, a modified version of which is used in the unit triangular diagonal (UD) square root implementation of the Kalman filter.

Gauss showed that under the constraints on the unknown measurement error vector $\mathbf{v}$ to be zero-mean and satisfy Eq. (10.12), the least squares solution also minimizes the mean squared error

$$\mathrm{E}\langle|\mathbf{z}-\mathbf{Hx}|^2\rangle$$

Gauss published his results in New Latin,[3] which eventually fell out of favor as a language for mathematical discourse. As a consequence, his work was largely forgotten until essentially the same result was obtained by Andrey Andreyevich Markov (1856–1922) nearly a century later [11]. It was called the *Markov theorem* until the earlier publication by Gauss was rediscovered. It is now called the *Gauss–Markov theorem.*

Generalization by Aitken Distributions that are not homoskedastic are called *heteroskedastic.* The Gauss–Markov theorem was generalized to the heteroskedastic (i.e. general) case by Alexander Craig Aitken (1895–1967) in a 1935 publication [12]. In this case, the covariance of the zero-mean measurement errors

$$\mathrm{E}\langle\mathbf{v}\mathbf{v}^T\rangle = \mathbf{R} \tag{10.13}$$

which is not necessarily in the form of a scalar matrix, as in Eq. (10.12).

A weighted least squares solution. The trick is to convert the least squares problem of $\mathbf{z} = \mathbf{Hx} + \mathbf{v}$ to a *weighted least squares* problem,

$$\underbrace{\mathbf{Wz}}_{\mathbf{z}^\star} = \underbrace{\mathbf{WHx}}_{\mathbf{H}^\star} + \underbrace{\mathbf{Wv}}_{\mathbf{v}^\star} \tag{10.14}$$

such that the weighted noise vector $\mathbf{v}^\star$ is homoskedastic. That is, such that a scalar matrix

$$\begin{aligned}\sigma^2\mathbf{I} &= \mathrm{E}\langle\mathbf{v}^\star\mathbf{v}^{\star T}\rangle\\ &= \mathrm{E}\langle(\mathbf{Wv})(\mathbf{Wv})^T\rangle\\ &= \mathbf{W}\mathrm{E}\langle\mathbf{v}\mathbf{v}^T\rangle\mathbf{W}^T\\ &= \mathbf{WRW}^T \end{aligned}\tag{10.15}$$

or, for $\sigma = 1$,

$$\begin{aligned}\mathbf{R} &= \mathbf{W}^{-1}\mathbf{I}\mathbf{W}^{-T}\\ &= \mathbf{W}^{-1}\mathbf{W}^{-T}\end{aligned}$$

[3] Also called Modern Latin or Neoclassical Latin, it was a revival of Latin as a language for scientific publications between the late fourteenth and late nineteenth centuries. Another of Gauss's discoveries, the fast Fourier transform, was also published in New Latin, in 1805, later to be discovered independently by James W. Cooley (1926–2016) and John W. Tukey (1915–2000) and published in 1965 [10].

$$= [\mathbf{W}^T\mathbf{W}]^{-1} \tag{10.16}$$

$$\mathbf{R}^{-1} = \mathbf{W}^T\mathbf{W} \tag{10.17}$$

which constrains the values of the weighting matrix $\mathbf{W}$ such that the weighted least squares problem of Eq. (10.14) is homoskedastic. In that case, the *Gauss–Markov theorem* guarantees that the linear least mean squares solution for the unknown $\mathbf{x}$ will be the least squares solution of Eq. (10.14), given by Eq. (10.10) with $\mathbf{H}$ replaced by $\mathbf{H}^\star$ and $\mathbf{z}$ replaced by $\mathbf{z}^\star$:

$$\begin{aligned}\hat{\mathbf{x}} &= [\mathbf{H}^{\star T}\mathbf{H}^\star]^{-1}\mathbf{H}^{\star T}\mathbf{z}^\star\\ &= [(\mathbf{W}\mathbf{H})^T\mathbf{W}\mathbf{H}]^{-1}(\mathbf{W}\mathbf{H})^T\mathbf{W}\mathbf{z}\\ &= [\mathbf{H}^T\mathbf{W}^T\mathbf{W}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{W}^T\mathbf{W}\mathbf{z}\\ &= [\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{z}\end{aligned} \tag{10.18}$$

– the general linear least mean squares solution. Note that it does not depend on choosing a particular solution $\mathbf{W}$ to Eq. (10.17), but only on the covariance matrix $\mathbf{R}$ of measurement errors.

Means and Mean Squared Estimation Error If $\hat{\mathbf{x}}$ is an estimated value of a variate with mean $\overline{\mathbf{x}}$ and covariance $\mathbf{P}$, then the squared estimation error for $\hat{\mathbf{x}}$ as a function of $\mathbf{x} \in \mathbb{R}^n$ is $|\hat{\mathbf{x}} - \mathbf{x}|^2$ and the mean squared estimation error over the probability distribution of $\mathbf{x}$ is its expected value,

$$\begin{aligned}\mathrm{E}\langle|\hat{\mathbf{x}} - \mathbf{x}|^2\rangle &= \mathrm{E}\langle[\hat{\mathbf{x}} - \mathbf{x}]^T[\hat{\mathbf{x}} - \mathbf{x}]\rangle\\ &= |\hat{\mathbf{x}}|^2 - 2\hat{\mathbf{x}}^T\mathrm{E}\langle\mathbf{x}\rangle + \mathrm{E}\langle|\mathbf{x}|^2\rangle\\ &= |\hat{\mathbf{x}}|^2 - 2\hat{\mathbf{x}}^T\mathrm{E}\langle\mathbf{x}\rangle + \mathrm{E}\langle|\overline{\mathbf{x}} + (\mathbf{x} - \overline{\mathbf{x}})|^2\rangle\\ &= |\hat{\mathbf{x}}|^2 - 2\hat{\mathbf{x}}^T\overline{\mathbf{x}} + |\overline{\mathbf{x}}|^2 + \mathrm{E}\langle|\mathbf{x} - \overline{\mathbf{x}}|^2\rangle\\ &= |\hat{\mathbf{x}}|^2 - 2\hat{\mathbf{x}}^T\overline{\mathbf{x}} + |\overline{\mathbf{x}}|^2 + \sum_i \mathrm{E}\langle(x_i - \overline{x}_i)^2\rangle\end{aligned} \tag{10.19}$$

$$\mathrm{E}\langle|\hat{\mathbf{x}} - \mathbf{x}|^2\rangle = |\hat{\mathbf{x}} - \overline{\mathbf{x}}|^2 + \mathrm{Tr}(\mathbf{P}) \tag{10.20}$$

Equation (10.20) is a quadratic equation in the estimated value $\hat{\mathbf{x}}$ that reaches its minimum value at $\hat{\mathbf{x}} = \overline{\mathbf{x}}$, its mean, and that minimum value is the trace of $\mathbf{P}$, which the sum of the diagonal elements of $\mathbf{P}$. That is, in the LLMSE, *the least mean squares estimate is always the mean of the probability distribution* – independent of any other attributes of the probability distribution in question. That is why – in the LLMSE – only the means and covariances of the distributions matter. The linear least mean squares estimate is otherwise independent of other attributes of the probability distributions. This is a very important attribute of Kalman filtering, as well.

See Appendix C on www.wiley.com/go/grewal/gnss of a demonstration showing that Eq. (10.20) is independent of the PDF of the variates involved.

Covariance of Estimation Error *Estimation error* is defined as the difference between the estimate $\hat{\mathbf{x}}$ and $\mathbf{x}$:

$$\begin{aligned}
\delta &\overset{\text{def}}{=} \hat{\mathbf{x}} - \mathbf{x} \\
&= [\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{z} - \mathbf{x} \\
&= [\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}(\mathbf{H}\mathbf{x} + \mathbf{v}) - \mathbf{x} \\
&= \underbrace{\left[\underbrace{[\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}}_{\mathbf{I}} - \mathbf{I}\right]}_{0}\mathbf{x} + [\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{v} \\
&= [\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{v}
\end{aligned} \tag{10.21}$$

Hence its mean,

$$\begin{aligned}
\underset{\mathbf{v}}{\mathrm{E}}\langle\delta\rangle &= \mathrm{E}_\mathbf{v}\langle[\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{v}\rangle \\
&= [\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathrm{E}_\mathbf{v}\langle\mathbf{v}\rangle \\
&= 0
\end{aligned} \tag{10.22}$$

if $\mathbf{v}$ is zero-mean, and its covariance about the mean

$$\begin{aligned}
\mathbf{P} &\overset{\text{def}}{=} \mathrm{E}\langle\delta\delta^T\rangle \\
&= \mathrm{E}_\mathbf{v}\langle[\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{v}\,\mathbf{v}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}\,[\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1}\rangle \\
&= [\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathrm{E}_\mathbf{v}\langle\mathbf{v}\,\mathbf{v}^T\rangle\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}\,[\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1} \\
&= [\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\underbrace{\mathbf{R}_\mathbf{v}^{-1}\mathbf{R}_\mathbf{v}}_{\mathbf{I}}\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}\,[\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1} \\
&= \underbrace{[\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}}_{\mathbf{I}}[\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1} \\
&= [\mathbf{H}^T\mathbf{R}_\mathbf{v}^{-1}\mathbf{H}]^{-1}
\end{aligned} \tag{10.23}$$

the covariance of estimation errors from the linear least-mean-squared estimator.

Recursive Linear Least Mean Squares Estimation If $\mathbf{z}_{k-1}$ and $\mathbf{z}_k$ are two independent measurements of the same variate $\mathbf{x}$ with statistically independent zero-mean measurement errors, then their respective measurement errors $\mathbf{v}_{k-1}$ and $\mathbf{v}_k$ are zero-mean with covariances

$$\mathrm{E}\langle\mathbf{v}_{k-1}\mathbf{v}_{k-1}^T\rangle = \mathbf{R}_{k-1} \tag{10.24}$$

$$\mathrm{E}\langle\mathbf{v}_k\mathbf{v}_k^T\rangle = \mathbf{R}_k \tag{10.25}$$

$$\mathrm{E}\langle\mathbf{v}_{k-1}\mathbf{v}_k^T\rangle = 0, \quad \text{the zero matrix} \tag{10.26}$$

and respective measurement sensitivity matrices $\mathbf{H}_{k-1}$ and $\mathbf{H}_k$.

Consequently the first measurement $\mathbf{z}_{k-1}$ will yield the initial estimate

$$\hat{\mathbf{x}}_{k-1(+)} = (\mathbf{H}_{k-1}^T \mathbf{R}_{k-1}^{-1} \mathbf{H}_{k-1})^{-1} \mathbf{H}_{k-1}^T \mathbf{R}_{k-1}^{-1} \mathbf{z}_{k-1} \tag{10.27}$$

with covariance of estimation error

$$\mathbf{P}_{k-1(+)} = (\mathbf{H}_{k-1}^T \mathbf{R}_{k-1}^{-1} \mathbf{H}_{k-1})^{-1} \tag{10.28}$$

from which we can extract two useful formulas:

$$\mathbf{H}_{k-1}^T \mathbf{R}_{k-1}^{-1} \mathbf{z}_{k-1} = \mathbf{P}_{k-1(+)}^{-1} \hat{\mathbf{x}}_{k-1(+)} \tag{10.29}$$

$$\mathbf{H}_{k-1}^T \mathbf{R}_{k-1}^{-1} \mathbf{H}_{k-1} = \mathbf{P}_{k-1(+)}^{-1} \tag{10.30}$$

The combined first and second measurements should then yield the linear least mean squares solution to

$$\begin{bmatrix} \mathbf{z}_{k-1} \\ \mathbf{z}_k \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{k-1} \\ \mathbf{H}_k \end{bmatrix} \mathbf{x}_{k(+)} + \begin{bmatrix} \mathbf{v}_{k-1} \\ \mathbf{v}_k \end{bmatrix} \tag{10.31}$$

with measurement noise covariance

$$\mathrm{E}\left\langle \begin{bmatrix} \mathbf{v}_{k-1} \\ \mathbf{v}_k \end{bmatrix} \begin{bmatrix} \mathbf{v}_{k-1} \\ \mathbf{v}_k \end{bmatrix}^T \right\rangle = \begin{bmatrix} \mathbf{R}_{k-1} & 0 \\ 0 & \mathbf{R}_k \end{bmatrix} \tag{10.32}$$

and solution

$$\begin{aligned} \hat{\mathbf{x}}_{k(+)} &= \left\{ \begin{bmatrix} \mathbf{H}_{k-1} \\ \mathbf{H}_k \end{bmatrix}^T \begin{bmatrix} \mathbf{R}_{k-1} & 0 \\ 0 & \mathbf{R}_k \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{H}_{k-1} \\ \mathbf{H}_k \end{bmatrix} \right\}^{-1} \\ &\quad \times \begin{bmatrix} \mathbf{H}_{k-1} \\ \mathbf{H}_k \end{bmatrix}^T \begin{bmatrix} \mathbf{R}_{k-1} & 0 \\ 0 & \mathbf{R}_k \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{z}_{k-1} \\ \mathbf{z}_k \end{bmatrix} \\ &= \{\mathbf{H}_{k-1}^T \mathbf{R}_{k-1}^{-1} \mathbf{H}_{k-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k\}^{-1} \\ &\quad \times \{\mathbf{H}_{k-1}^T \mathbf{R}_{k-1}^{-1} \mathbf{z}_{k-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{z}_k\} \end{aligned} \tag{10.33}$$

Using Eqs. (10.29) and (10.30), this last result can be transformed into

$$\begin{aligned} \hat{\mathbf{x}}_{k(+)} &= \{\mathbf{P}_{k-1(+)}^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k\}^{-1} \\ &\quad \times \{\mathbf{P}_{k-1(+)}^{-1} \hat{\mathbf{x}}_{k-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{z}_k\} \end{aligned} \tag{10.34}$$

Equation (10.34) expresses the kth linear least mean squares estimate $\hat{\mathbf{x}}_{k(+)}$ in terms of the $(k-1)$ th estimate $\hat{\mathbf{x}}_{k-1(+)}$ and the kth measurement $\mathbf{z}_k$. This can be put into the form used in Kalman filtering by using a matrix inversion formula from the mid-1940s.

Duncan–Guttman Formula This is a matrix inversion formula published first by William Jolly Duncan (1984–1960) in 1945 [13], followed by Louis Eliyahu Guttman (1916–1987) in 1946 [14]:

$$(\mathbf{A} - \mathbf{U}\mathbf{D}^{-1}\mathbf{V})^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{U}\,(\mathbf{D} - \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1} \tag{10.35}$$

which, with the substitutions

$$\mathbf{A} = \mathbf{P}^{-1}_{k-1(+)} \tag{10.36}$$

$$\mathbf{U} = \mathbf{H}^T_k \tag{10.37}$$

$$\mathbf{D} = -\mathbf{R}_k \tag{10.38}$$

$$\mathbf{V} = \mathbf{H}_k \tag{10.39}$$

becomes

$$\begin{aligned}(\mathbf{P}^{-1}_{k-1(+)} + \mathbf{H}^T_k\mathbf{R}^{-1}_k\mathbf{H}_k)^{-1} = \mathbf{P}_{k-1(+)} \\ -\underbrace{\mathbf{P}_{k-1(+)}\mathbf{H}^T_k(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k-1(+)}\mathbf{H}^T_k)^{-1}}_{\mathbf{K}_k} \\ \mathbf{H}_k\mathbf{P}_{k-1(+)}\end{aligned} \tag{10.40}$$

changing Eq. (10.34) to

$$\begin{aligned}\hat{\mathbf{x}}_{k(+)} &= \{\mathbf{P}_{k-1(+)} - \mathbf{P}_{k-1(+)}\mathbf{H}^T_k(\mathbf{R}_k + \mathbf{H}_{k(+)}\mathbf{P}_{k-1(+)}\mathbf{H}^T_k)^{-1}\mathbf{H}_k\mathbf{P}_{k-1(+)}\} \\ &\quad \times\{\mathbf{P}^{-1}_{k-1(+)}\hat{\mathbf{x}}_{k-1(+)} + \mathbf{H}^T_k\mathbf{R}^{-1}_k\mathbf{z}_k\}. \\ &= \{\mathbf{I} - \mathbf{P}_{k-1(+)}\mathbf{H}^T_k(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k-1(+)}\mathbf{H}^T_k)^{-1}\mathbf{H}_k\} \\ &\quad \times\{\hat{\mathbf{x}}_{k-1(+)} + \mathbf{P}_{k-1(+)}\mathbf{H}^T_k\mathbf{R}^{-1}_k\mathbf{z}_k\} \\ &= \hat{\mathbf{x}}_{k-1(+)} + \mathbf{P}_{k-1(+)}\mathbf{H}^T_k(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k-1(+)}\mathbf{H}^T_k)^{-1} \\ &\quad \times \{[(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k-1(+)}\mathbf{H}^T_k) - \mathbf{H}_k\mathbf{P}_{k-1(+)}\mathbf{H}^T_k]\mathbf{R}^{-1}_k\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k-1(+)}\} \\ &= \hat{\mathbf{x}}_{k-1(+)} + \underbrace{\mathbf{P}_{k-1(+)}\mathbf{H}^T_k(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k-1(+)}\mathbf{H}^T_k)^{-1}}_{\mathbf{K}_k} \\ &\quad \times \{[\mathbf{R}_k]\mathbf{R}^{-1}_k\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k-1(+)}\} \\ &= \hat{\mathbf{x}}_{k-1(+)} + \mathbf{K}_k\{\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k-1(+)}\}\end{aligned} \tag{10.41}$$

Kalman Gain Matrix Equation (10.41) is almost the form used in the Kalman measurement update, except for subscription substitutions to correct the assumptions that

1. The discrete times $t_k = t_{k-1}$.
2. The state vector $\mathbf{x}$ remained unmolested by random dynamics between t_{k-1} and t_k, so that

$$\hat{\mathbf{x}}_{k(-)} = \hat{\mathbf{x}}_{k-1(+)}$$

$$\mathbf{P}_{k(-)} = \mathbf{P}_{k-1(+)}$$

With these substitutions, the measurement update has the form

$$\mathbf{K}_k = \mathbf{P}_{k(-)}\mathbf{H}^T_k(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k(-)}\mathbf{H}^T_k)^{-1} \tag{10.42}$$

$$\hat{\mathbf{x}}_{k(+)} = \hat{\mathbf{x}}_{k(-)} + \mathbf{K}_k\{\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k(-)}\} \tag{10.43}$$

$\mathbf{K}_k$ is called the *Kalman gain matrix*. It is the weighting matrix applied to the difference between the kth measurement $\mathbf{z}_k$ and the *expected measurement* $\mathbf{H}_k\hat{\mathbf{x}}_{k(-)}$ based on the a priori estimate $\hat{\mathbf{x}}_{k(-)}$. It is considered to be the *crown jewel* of the Kalman filter derivation.

The fact that the Kalman gain matrix does not depend on the measurement can be exploited by computing it in the time between measurements.

10.4.1.2 Kalman Measurement Update Equations

The Kalman measurement correction equations account for the changes in the estimate $\hat{\mathbf{x}}_k$ and its associated covariance matrix of estimation uncertainty to reflect the information gained from the most recent measurement $\mathbf{z}_k$. It can be computed more efficiently by reusing the Kalman gain in the covariance update:

$$\mathbf{K}_k \overset{\text{def}}{=} \mathbf{P}_{k(-)}\mathbf{H}_k^T(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k(-)}\mathbf{H}_k^T)^{-1} \tag{10.44}$$

$$\hat{\mathbf{x}}_{k(+)} = \hat{\mathbf{x}}_{k(-)} + \mathbf{K}_k\{\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k(-)}\} \tag{10.45}$$

$$\mathbf{P}_{k(+)} = \mathbf{P}_{k(-)} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k(-)} \tag{10.46}$$

where the subscripting indicates the sequencing of the measurements $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \dots$.

This completes the derivation of the Kalman corrector equations.

10.4.2 Time Update (Predictor)

The time update is also called the “temporal update.”

The Kalman predictor uses the same unknown vector $\mathbf{x}$ from the corrector, but now addresses the question of how the estimate $\hat{\mathbf{x}}$ and its associated covariance of estimation uncertainty $\mathbf{P}$ are affected by known and unknown (i.e. random) dynamics. Like the measurement update, it also uses linearity to characterize the estimation problem using only means and covariances.

10.4.2.1 Continuous-Time Dynamics

State space is a representation of a dynamic system of n variables x_i as components of a vector $\mathbf{x}$ in n-dimensional Euclidean space $\mathbb{R}^n$. The dimension of this state space is determined by the number of degrees of freedom of the dynamic system. In the Kalman *predictor*, the same vector $\mathbf{x} \in \mathbb{R}^n$ used in the Kalman corrector becomes the vector of variables subject to changes over time – as governed by linear differential equations of the sort

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(t)\ \mathbf{x}(t) + \mathbf{w}(t) \tag{10.47}$$

where

$\mathbf{x}(t)$ is a real n-vector-valued function of time (t), where n is the dimension of $\mathbf{x}$.

$\mathbf{F}(t)$ is an $n \times n$ real matrix-valued function of time.
$\mathbf{w}(t)$ is a real vector-valued function of time (t).

Homogeneous and Nonhomogeneus Differential Equations If the function $\mathbf{w}(t) = 0$, the resulting differential equation (10.47) assumes the form

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(t)\ \mathbf{x}(t) \tag{10.48}$$

which is called *homogeneous,* meaning that the dependent variable $\mathbf{x}$ appears in every term of the equation. Otherwise, the differential equation (10.47) is considered to be *nonhomogeneous.*

Example 10.1 ***(mth Order Nonhomogeneous Ordinary Linear Differential Equation)***

This is a differential equation of the sort

$$\sum_{i=0}^{m} a_i(t)\left(\frac{d}{dt}\right)^i y(t) = b(t)$$

in the scalar function $y(t)$. For pedagogical purposes it can be rewritten in the form

$$\left(\frac{d}{dt}\right)^m y(t) = \frac{b(t)}{a_m(t)} - \sum_{i=0}^{m-1} \frac{a_i(t)}{a_m(t)}\left(\frac{d}{dt}\right)^i y(t) \tag{10.49}$$

so that the first-order linear differential equation for the vectorized dependent variable

$$\mathbf{x}(t) \overset{\text{def}}{=} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ \vdots \\ x_m(t) \end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix} y(t) \\ \frac{d}{dt}y(t) \\ \left(\frac{d}{dt}\right)^2 y(t) \\ \vdots \\ \left(\frac{d}{dt}\right)^{m-1} y(t) \end{bmatrix}$$

$$\frac{d}{dt}\mathbf{x}(t) = \begin{bmatrix} \frac{d}{dt}y(t) \\ \left(\frac{d}{dt}\right)^2 y(t) \\ \left(\frac{d}{dt}\right)^3 y(t) \\ \vdots \\ \left(\frac{d}{dt}\right)^m y(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ x_3(t) \\ x_4(t) \\ \vdots \\ \frac{b(t)}{a_m(t)} - \sum_{i=0}^{m-1} \frac{a_i(t)}{a_m(t)}\left(\frac{d}{dt}\right)^i y(t) \end{bmatrix},$$

where we have used Eq. (10.49) so that the result can be expressed in state-space form as

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{w}(t)$$

$$\mathbf{F} \overset{\text{def}}{=} \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\dfrac{a_0(t)}{a_m(t)} & -\dfrac{a_1(t)}{a_m(t)} & -\dfrac{a_2(t)}{a_m(t)} & \cdots & -\dfrac{a_{m-1}(t)}{a_m(t)} \end{bmatrix}$$

$$\mathbf{w}(t) \overset{\text{def}}{=} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \dfrac{b(t)}{a_m(t)} \end{bmatrix}$$

Homogeneous Equation Solutions The key to solving the nonhomogeneous differential equation (10.47) depends on the solution to the corresponding homogeneous equation (10.48). In either case, the solution on a time interval $t_{k-1} \leq t \leq t_k$ can be expressed in terms of its initial value $\mathbf{x}(t_{k-1})$ on that interval, and the solution will involve the matrix exponential function.

Matrix Exponentials The exponential function exp ($\mathbf{M}$) of an $n \times n$ square matrix $\mathbf{M}$ is defined by the exponential power series

$$\exp(\mathbf{M}) \overset{\text{def}}{=} \sum_{m=0}^{\infty} \frac{1}{m!}\mathbf{M}^m \tag{10.50}$$

which always converges – but not fast enough to compute it that way. It is implemented more efficiently in MATLAB as the function `expm`.

If the square matrix $\mathbf{M}(t)$ is a differentiable function of t, the derivative of its exponential will be

$$\begin{aligned} \frac{d}{dt}\exp(\mathbf{M}(t)) &= \frac{d\,\mathbf{M}(t)}{dt}\exp(\mathbf{M}(t)) \\ &= \exp(\mathbf{M}(t))\frac{d\,\mathbf{M}(t)}{dt} \end{aligned} \tag{10.51}$$

Consequently, if we let the matrix $\mathbf{M}$ be defined as an integral of $\mathbf{F}(t)$ from the homogeneous equation (10.48), then

$$\mathbf{M}(s,t) \overset{\text{def}}{=} \int_s^t \mathbf{F}(\tau)d\tau \tag{10.52}$$

$$\mathbf{\Phi}(s,t) \overset{\text{def}}{=} \exp(\mathbf{M}(s,t)) \tag{10.53}$$

$$\mathbf{\Phi}(s,s) = \mathbf{\Phi}(t,t) = \mathbf{I} \tag{10.54}$$

$$\mathbf{\Phi}(t,\tau)\mathbf{\Phi}(s,t) = \mathbf{\Phi}(s,\tau) \tag{10.55}$$

$$\mathbf{\Phi}^{-1}(s,t) = \mathbf{\Phi}(t,s) \tag{10.56}$$

$$\frac{d}{dt}\mathbf{\Phi}(s,t) = \mathbf{F}(t)\ \mathbf{\Phi}(s,t) \tag{10.57}$$

$$\frac{d}{ds}\mathbf{\Phi}(s,t) = -\mathbf{F}(s)\ \mathbf{\Phi}(s,t) \tag{10.58}$$

The matrix $\mathbf{\Phi}(s,\ t)$ is what is called the *fundamental solution matrix* for Eq. (10.48), in that it solves the associated boundary value problem as

$$\mathbf{x}(t) = \mathbf{\Phi}(s,\ t)\ \mathbf{x}(s) \tag{10.59}$$

State Transition Matrices Kalman filtering uses the discrete-time notation

$$\mathbf{\Phi}_k \overset{\text{def}}{=} \mathbf{\Phi}(t_{k-1},\ t_k) \tag{10.60}$$

for the *state transition matrix* $\mathbf{\Phi}_k$ between the $(k-1)$ th and kth discrete time epochs.

Nonhomogeneous Solutions Equation (10.47) with a known integrable dynamic disturbance function $\mathbf{w}(t)$ has a closed-form solution for the initial value problem in terms of the state transition matrix function as

$$\mathbf{x}(t) = \mathbf{\Phi}(s,\ t)\ \mathbf{x}(s) + \int_s^t \mathbf{\Phi}(\tau,\ t)\ \mathbf{w}(\tau)d\tau \tag{10.61}$$

where $\mathbf{x}(s)$ is the initial value on the interval $[s,\ t]$ and the integral is the ordinary (Riemann) integral.

Linear Stochastic Differential Equations One might like to generalize Eq. (10.61) for a *white noise process* $\mathbf{w}(t)$ such that

$$\mathrm{E}\langle \mathbf{w}(t)\rangle = 0 \quad \text{for all} \quad t \quad \text{(zero mean)}$$

$$\mathrm{E}\langle \mathbf{w}(t)\mathbf{w}^T(s)\rangle = \begin{cases} 0, & s \neq t \quad \text{(uncorrelated in time)} \\ \mathbf{Q}(t), & s = t \end{cases}$$

where the covariance matrix $\mathbf{Q}(t)$ of random noise uncertainty is an $n \times n$ symmetric positive definite matrix and n is the dimension of $\mathbf{w}(t)$. If that could be done, then a solution might be obtained as

$$\mathbf{x}_k = \mathbf{\Phi}_k \mathbf{x}_{k-1} + \mathbf{w}_k \tag{10.62}$$

where the discrete-time additive random disturbance vector

$$\mathbf{w}_k = \int_{t_{k-1}}^{t_k} \mathbf{\Phi}(\tau,\ t_k)\ \mathbf{w}(\tau)d\tau \tag{10.63}$$

has zero mean

$$\mathrm{E}\langle \mathbf{w}_k\rangle = \int_{t_{k-1}}^{t_k} \mathbf{\Phi}(\tau,\ t_k)\ \mathrm{E}\langle \mathbf{w}(\tau)\rangle d\tau = 0 \tag{10.64}$$

and covariance

$$\begin{aligned}\mathbf{Q}_k &= \mathrm{E}\langle \mathbf{w}_k \mathbf{w}_k^T \rangle \\ &= \mathrm{E}\left\langle \left[\int_{t_{k-1}}^{t_k} \boldsymbol{\Phi}(\tau,\ t_k)\ \mathbf{w}(\tau) d\tau\right] \left[\int_{t_{k-1}}^{t_k} \boldsymbol{\Phi}(\sigma,\ t_k)\ \mathbf{w}(\sigma) d\sigma\right]^T \right\rangle \\ &= \int_{t_{k-1}}^{t_k} \int_{t_{k-1}}^{t_k} \boldsymbol{\Phi}(\tau,\ t_k)\ \mathrm{E}\langle \mathbf{w}(\tau)\ \mathbf{w}^T(\sigma)\rangle \boldsymbol{\Phi}^T(\sigma,\ t_k) d\tau\ d\sigma \\ &= \int_{t_{k-1}}^{t_k} \boldsymbol{\Phi}(\tau,\ t_k)\ \mathbf{Q}(\tau) \boldsymbol{\Phi}^T(\tau,\ t_k) d\tau \end{aligned} \tag{10.65}$$

and the corresponding discrete time update for the covariance of state estimation uncertainty could then be expressed as

$$\mathbf{P}_{k(-)} = \boldsymbol{\Phi}_k \mathbf{P}_{k-1(+)} \boldsymbol{\Phi}_k^T + \mathbf{Q}_k \tag{10.66}$$

The problem with this approach is that Eq. (10.47) then becomes what is called a *stochastic*[4] *differential equation* because it includes a white noise process, which is not an integrable function in the Riemann calculus. A new calculus would be required for modeling random disturbance noise $\mathbf{w}(t)$ in Eq. (10.47).

Stochastic Integrals and Markov Processes The problem of integrating white noise processes had been studied for more than half a century [15] until the first stochastic integral was defined by Kyosi Itô (1915–2008) in 1944 [16]. The resulting stochastic integral equation is not in the same form, but it does result in what is called a **Markov process** in discrete time. This is a sequence of random vectors

$$\cdots\ , \mathbf{w}_{k-1},\ \mathbf{w}_k,\ \mathbf{w}_{k+1},\ \cdots$$

such that

$$\mathrm{E}\langle \mathbf{w}_k \rangle = 0 \tag{10.67}$$

$$\mathrm{E}\langle \mathbf{w}_j \mathbf{w}_k^T \rangle = \begin{cases} 0, & j \neq k \\ \mathbf{Q}_k, & j = k \end{cases} \tag{10.68}$$

where the $\mathbf{Q}_k$ are $n \times n$ symmetric positive definite covariance matrices and n is the dimension of $\mathbf{x}_k$ and $\mathbf{w}_k$. This is a somewhat generalized Markov process, in that we only care about the sequence of means and covariances of the $\mathbf{w}_k$, and their covariances can vary with k. For a derivation including the stochastic calculus step, see [17]. That is all that is needed for Kalman filtering.

10.4.2.2 Discrete-Time Dynamics

From Eqs. (10.62) and (10.68),

$$\hat{\mathbf{x}}_k = \boldsymbol{\Phi}_k\ \hat{\mathbf{x}}_{k-1(+)} \tag{10.69}$$

[4] From the Greek word for *guess* or *aim at*.

$$\mathbf{P}_{k(-)} = \mathbf{\Phi}_k \mathbf{P}_{k-1(+)} \mathbf{\Phi}_k^T + \mathbf{Q}_k \tag{10.70}$$

where the parameters $\mathbf{\Phi}_k$ and $\mathbf{Q}_k$, and the inputs $\hat{\mathbf{x}}_{k-1(+)}$ and $\mathbf{P}_{k-1(+)}$ are given.

10.4.3 Basic Kalman Filter Equations

The top-level processing flow of the Kalman filter is shown in Figure 10.1.

The essential implementation equations for the Kalman filter are summarized in Table 10.1. In most cases, these follow the same symbolic notation used by Kalman in his original paper [2]. The main exception is that Kalman used the symbol Δ for what is now called the Kalman gain matrix and – in his honor – represented by the letter **K**.

10.4.4 The Time-Invariant Case

If the set of Kalman filter model parameters $\{\mathbf{\Phi}_k, \mathbf{Q}_k, \mathbf{H}_k, \mathbf{R}_k\}$ have the same values for every time t_k, the system is called *time-invariant* or *linear time-invariant* (LTI).

Perhaps the most common use of time-invariant Kalman filtering is for solving the steady-state riccati equations to calculate the steady-state Kalman gain $\mathbf{K}_\infty$, which can be used as the measurement feedback gain without having to solve the riccati equation. This approach can be much more efficient than full-blown Kalman filtering. Time-invariance is otherwise not that common in practice.

10.4.5 Observability and Stability Issues

Observability is the issue of whether an unknown variable can be determined from the relationships between the unknowns and knowns (i.e. the data) of an estimation problem. For linear problems, this does not generally depend on the data, but only on its relationship to the unknowns.

Least squares problems are deemed observable if and only if the associated *Gramian matrix* of Eq. (10.9) is nonsingular. This is a *qualitative* characterization of observability that does not depend on the data **z**, but may require infinite precision to be calculated precisely.

Kalman filtering observability is characterized by the resulting covariance matrix **P** of estimation uncertainty, which is a *quantitative* measure of observability depending on the problem parameters $\mathbf{H}_k$, $\mathbf{R}_k$, $\mathbf{\Phi}_k$, and $\mathbf{Q}_k$. The final value of $\mathbf{P}_k$ characterizes *how well* the state vector **x** can be determined, and it does not depend on the actual measurements $\mathbf{z}_k$. This is very important in Kalman filtering problems, because it allows the degree of observability of the estimation problem to be determined just from the problem parameters, without requiring the actual measured values $\mathbf{z}_k$.

Stability of a linear stochastic system is the issue of whether its solutions are bounded over time. For time-invariant systems, this can be characterized by the eigenvalues of the dynamic coefficient matrix **F** being in the left half of the complex plane (i.e. having negative real parts) or those of the associated state transition matrix $\mathbf{\Phi}$ being inside the unit circle in the complex plane. However, the accuracy of numerically calculated eigenvalues near the boundaries of stability may leave room for ambiguity.

Table 10.1 Basic Kalman filter equations.

Implementation equations

Inputs	Processing	Outputs
Time update (predictor)		
$\hat{\mathbf{x}}_{k-1(+)}$, $\mathbf{P}_{k-1(+)}$	$\hat{\mathbf{x}}_{k(-)} = \mathbf{\Phi}_k \hat{\mathbf{x}}_{k-1(+)}$	$\hat{\mathbf{x}}_{k(-)}$, $\mathbf{P}_{k(-)}$
$\mathbf{\Phi}_k$, $\mathbf{Q}_k$	$\mathbf{P}_{k(-)} = \mathbf{\Phi}_k \mathbf{P}_{k-1(+)} \mathbf{\Phi}_k^T + \mathbf{Q}_k$	
Measurement update (corrector)		
$\hat{\mathbf{x}}_{k(-)}$, $\mathbf{P}_{k(-)}$	$\mathbf{K}_k = \mathbf{P}_{k(-)} \mathbf{H}_k^T (\mathbf{R}_k + \mathbf{H}_k \mathbf{P}_{k(-)} \mathbf{H}_k^T)^{-1}$	$\hat{\mathbf{x}}_{k(+)}$, $\mathbf{P}_{k(+)}$
$\mathbf{z}_k$, $\mathbf{H}_k$, $\mathbf{R}_k$	$\hat{\mathbf{x}}_{k(+)} = \hat{\mathbf{x}}_{k(-)} + \mathbf{K}_k \{\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k(-)}\}$	
	$\mathbf{P}_{k(+)} = \mathbf{P}_{k(-)} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k(-)}$	

Parameters and variables

Symbol	Description	Dimension
n	Dimension of state vector (integer)	1
ℓ_k	Dimension of kth measurement vector (integer)	1
k	Discrete measurement time index (integer)	1
t_k	Discrete time of kth measurement (scalar)	1
$\hat{\mathbf{x}}_{k-1(+)}$	A posteriori estimate at time t_{k-1}	$n \times 1$
$\mathbf{P}_{k-1(+)}$	A posteriori estimation covariance at timet_{k-1}	$n \times n$
$\mathbf{\Phi}_k$	State transition matrix from t_{k-1} to t_k	$n \times n$
$\mathbf{Q}_k$	Uncertainty accumulated between t_{k-1} and t_k	$n \times n$
$\hat{\mathbf{x}}_{k(-)}$	A priori estimate at time t_k	$n \times 1$
$\mathbf{P}_{k(-)}$	A priori estimation covariance at time t_k	$n \times n$
$\mathbf{z}_k$	Measurement vector at time t_k	$\ell_k \times 1$
$\mathbf{H}_k$	Measurement sensitivity matrix at time t_k	$\ell_k \times n$
$\mathbf{R}_k$	Measurement error covariance at timet_k	$\ell_k \times \ell_k$
$\mathbf{K}_k$	Kalman gain matrix at time t_k	$n \times \ell_k$
$\hat{\mathbf{x}}_{k(+)}$	A posteriori estimate at time t_k	$n \times 1$
$\mathbf{P}_{k(+)}$	A posteriori estimation covariance at time t_k	$n \times n$

One of the triumphs of Kalman filtering was showing that *dynamically unstable* estimation problems can still be observable. That is, the riccati equations (characterized by $\mathbf{H}_k$, $\mathbf{R}_k$, $\mathbf{\Phi}_k$, and $\mathbf{Q}_k$) can be stable even if the dynamic equations (characterized by $\mathbf{\Phi}_k$ and $\mathbf{Q}_k$ only) are unstable.

10.5 Alternative Implementations

The first working implementation of a Kalman filter was directed by Stanley F. Schmidt (1926–2015) when he was chief of the Dynamics Analysis Branch at the NASA Ames Research Center (ARC) in Mountain View, California in the late 1950s and early 1960s [18]. Schmidt was searching for navigation methods suitable for what would become the Apollo project to send American astronauts to the moon and back. Schmidt's approach included linearization of the navigation problem about nominal trajectories, and he consulted with Kalman about Kalman's "new approach to linear filtering and prediction problems" [2]. Schmidt's efforts resolved many of the essential implementation issues, but they also exposed some remaining vulnerabilities related to accumulated computer roundoff errors, especially in solving the riccati equations for updating the covariance matrix **P** of estimation uncertainty. Something better would be needed for shoehorning the implementation into a space-qualified flight computer for the Apollo missions.

10.5.1 Implementation Issues

Performance-related issues for the implementation of Kalman filters on digital computers include:

1. *Computational complexity,* which determines the numbers of arithmetic operations required to execute one cycle of the predictor and corrector equations. This generally scales as a polynomial in the dimensions of the state vector and measurement vector. Computational complexity formulas for most of the algorithms mentioned in this chapter can be found in the book [19] by Gilbert Stewart. Complexity is often quoted as an "order," $O(n^p \ell_k^q)$, in terms of the highest powers (p and q) of the dimensions (n and ℓ_k) of the matrices involved. Inversion of an $n \times n$ square matrix, for example, is of $O(n^3)$ complexity, as is multiplication of two $n \times n$ matrices.
2. *Wordlength requirements,* which determine the number of bits of precision needed to control to acceptable levels the roundoff errors in the digital implementation of a Kalman filters. This may depend on such model attributes as the condition numbers of matrices used in the Kalman filter model, or other algebraic properties of the matrices used. This issue has been exploited greatly by what are called "square root" implementations

of the Kalman filter. Better algorithms for matrix methods robust against computer roundoff may be found in the book by Golub and Van Loan [20].

Although digital processing technologies have improved by orders of magnitude since the introduction of the Kalman filter in 1960, these issues are still important for very large-scale implementations such as GNSS system state tracking at the constellation level.

10.5.2 Conventional Implementation Improvements

These are minor modifications of the implementation shown in Table 10.1 that have been shown to reduce computational complexity.

10.5.2.1 Measurement Decorrelation by Diagonalization

Kaminski [21] discovered that the computational complexity of the conventional Kalman measurement update implementation can be reduced significantly by processing the components of vector-valued observations sequentially after factoring the measurement covariance matrix $\mathbf{R}_k$ into a form such as

$$\mathbf{R}_k = \mathbf{U}_k \mathbf{D}_k \mathbf{U}_k^T \quad \text{(factorization)} \tag{10.71}$$

$$\mathbf{D}_k = \mathbf{U}_k^{-1} \mathbf{R}_k \mathbf{U}_k^{-T} \quad \text{(diagonalization)} \tag{10.72}$$

where $\mathbf{D}_k$ is a diagonal matrix with positive elements and $\mathbf{U}_k$ is easily inverted.

The modified Cholesky decomposition algorithm implemented by the MATLAB m-file `modchol.m`, for example, results in an $\ell_k \times \ell_k$ unit upper triangular $\mathbf{U}_k$. Inversion of $\ell_k \times \ell_k$ triangular matrices is of $O(\ell_k^2)$ computational complexity, whereas general $\ell_k \times \ell_k$ matrix inversion is of $O(\ell_k^3)$ complexity. Then the alternative measurement vector and model parameters

$$\mathbf{z}_k^\star \overset{\text{def}}{=} \mathbf{U}_k^{-1} \mathbf{z}_k \tag{10.73}$$

$$\mathbf{H}_k^\star \overset{\text{def}}{=} \mathbf{U}_k^{-1} \mathbf{H}_k \tag{10.74}$$

$$\mathbf{R}_k^\star \overset{\text{def}}{=} \mathbf{D}_k \tag{10.75}$$

represent a measurement vector $\mathbf{z}_k^\star$ with uncorrelated errors, with corresponding measurement sensitivity matrices equal to the rows of $\mathbf{H}_k^\star$ and corresponding variances equal to the diagonal elements of $\mathbf{D}_k$. Individual elements of the vector $\mathbf{z}_k^\star$ can then be processes serially as uncorrelated scalar measurements.

The computational advantage of Kaminski's decorrelation approach [21] is

$$\frac{1}{3}\ell_k^3 - \frac{1}{2}\ell_k^2 + \frac{7}{6}\ell_k - \ell_k n + 2\ell_k^2 n + \ell_k n^2 \text{ floating-point operations (flops)}$$

where ℓ_k is the dimension of $\mathbf{z}_k$ and n is the dimension of the state vector. That is, it requires that many fewer flops to decorrelate vector-valued measurements and process the components serially.

10.5.2.2 Exploiting Symmetry

Computation requirements for computing symmetric matrices can be reduced by computing only the diagonal elements and the off-diagonal elements on one side of the main diagonal. For example, this would apply to the following terms from the conventional Kalman filter implementation in Table 10.1:

$$\mathbf{P}_{k(-)} = \mathbf{\Phi}_k \mathbf{P}_{k-1(+)} \mathbf{\Phi}_k^T + \mathbf{Q}_k,\ (\mathbf{R}_k + \mathbf{H}_k \mathbf{P}_{k(-)} \mathbf{H}_k^T)^{-1},\ \mathbf{P}_{k(+)} = \mathbf{P}_{k(-)} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k(-)}$$

Furthermore, Verhaegen and Van Dooren [22] have shown that this "forced symmetry" also mitigates some cumulative effects of computer roundoff errors.

10.5.2.3 Information Filter

The inverse of the covariance matrix of estimation uncertainty is called an *information matrix*:

$$\mathbf{Y} \overset{\text{def}}{=} \mathbf{P}^{-1} \tag{10.76}$$

a concept first defined in a broader context by Ronald Aylmer Fisher (1890–1962) in 1925 [23]. This alternative representation can be useful in some situations – such as those in which there is essentially *no* information in some subspace of state space. This can occur during start-up with incomplete state estimates, for example. In that case the linear least mean squares estimate $\hat{\mathbf{x}}$ satisfying

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}$$
$$\mathrm{E}\langle \mathbf{v}\mathbf{v}^T \rangle = \mathbf{R}$$

is *underdetermined* because its Gramian $\mathbf{H}^T\mathbf{H}$ is singular. However, the associated information matrix

$$\mathbf{Y} = \mathbf{H}\mathbf{R}^{-1}\mathbf{H}^T$$

is computable, even though some of its eigenvalues are 0. In this case, the corresponding eigenvalues of $\mathbf{P}$ would be infinite.

The Kalman filter implementation in terms of $\mathbf{Y}$ greatly simplifies the measurement update but complicates the time update. It is also less transparent because the information matrix needs to be inverted to understand and evaluate estimation uncertainties quantitatively, and because the state estimate for the information filter is redefined as

$$\hat{\mathbf{y}} \overset{\text{def}}{=} \mathbf{Y}\hat{\mathbf{x}} \tag{10.77}$$

The information filter implementation in terms of $\hat{\mathbf{y}}$ and $\mathbf{Y}$ is listed in Table 10.2. Note that this implementation assumes the state transition matrices $\mathbf{\Phi}_k$ and dynamic disturbance covariance matrices $\mathbf{Q}_k$ are nonsingular.

Table 10.2 Information filter equations.

Information filter variables

$$\mathbf{Y} \overset{\text{def}}{=} \mathbf{P}^{-1} \qquad \mathbf{y} \overset{\text{def}}{=} \mathbf{Y}\,\mathbf{x}$$

Implementation equations

Predictor	Corrector
$\mathbf{C}_k = \mathbf{\Phi}_k^{-1}\mathbf{Y}_{k-1(+)}\mathbf{\Phi}_k^{-T}$	$\mathbf{A}_k = \mathbf{H}_k^T\mathbf{R}_k^{-1}$
$\mathbf{D}_k = \mathbf{C}_k[\mathbf{C}_k + \mathbf{Q}_k^{-1}]^{-1}$	$\mathbf{B}_k = \mathbf{A}_k^T\mathbf{H}_k^T$
$\mathbf{E}_k = \mathbf{I} - \mathbf{D}_k$	$\hat{\mathbf{y}}_{k(+)} = \hat{\mathbf{y}}_{k(-)} + \mathbf{A}_k\mathbf{z}_k$
$\hat{\mathbf{y}}_{k(-)} = \mathbf{E}_k\mathbf{\Phi}_k^{-T}\hat{\mathbf{y}}_{k-1(+)}$	$\mathbf{Y}_{k(+)} = \mathbf{Y}_{k(-)} + \mathbf{B}_k$
$\mathbf{Y}_{k(-)} = \mathbf{E}_k\mathbf{C}_k\mathbf{E}_k^T + \mathbf{D}_k\mathbf{Q}_k^{-1}\mathbf{D}_k^T$	

10.5.2.4 Sigma Rho Filtering

This implementation by Grewal and Kain [24] uses the alternative covariance factorization

$$\mathbf{C} = \mathbf{D}_\sigma \boldsymbol{P} \mathbf{D}_\sigma \tag{10.78}$$

$$\mathbf{D}_\sigma = \mathrm{diag}(\sigma_1,\ \sigma_2,\ \sigma_3,\ \dots\ ,\ \sigma_n) \tag{10.79}$$

$$\sigma_i > 0 \quad \text{for all} \quad 1 \le i \le n \tag{10.80}$$

$$\boldsymbol{P} = \begin{bmatrix} 1 & \rho_{12} & \rho_{13} & \cdots & \rho_{1n} \\ \rho_{21} & 1 & \rho_{23} & \cdots & \rho_{2n} \\ \rho_{31} & \rho_{32} & 1 & \cdots & \rho_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{n1} & \rho_{n2} & \rho_{n3} & \cdots & 1 \end{bmatrix} \tag{10.81}$$

where $\mathbf{C} = \mathbf{R}$ or $\mathbf{P}$, not to be confused with the capital of ρ ("rho"), which is $\boldsymbol{P}$ ("Rho").

The σ_i are the RMS uncertainties of the measurement or state vector components and the ρ_{ij} are the correlation coefficients between the errors of the ith and jth components. For more detail see [25].

10.5.3 James E. Potter (1937–2005) and Square Root Filtering

A major breakthrough came in 1962, when Potter was a staff member at the MIT Instrumentation Laboratory (later reorganized and renamed *The Charles Stark Draper Laboratory*), working on the MIT-designed navigation system for the Apollo missions. Studies on mainframe computers with 36-bit floating point arithmetic had shown that a Kalman filter implemention was adequate for the task, but it had to be implemented on the Apollo Computer, a beyond-state-of-the-art special-purpose flight computer with 15-bit fixed-point arithmetic. Potter took the problem home with him on a Friday and returned with a solution the following Monday.

Potter's idea was to redefine the Kalman filter implementation in terms of a square root $\mathbf{P}^{1/2}$ of the covariance matrix $\mathbf{P}$ of estimation uncertainty,

$$\mathbf{P} = \mathbf{P}^{1/2}\mathbf{P}^{T/2} \tag{10.82}$$

Potter's definition of a matrix square root is at odds with standard mathematical nomenclature, which defines the square of a matrix $\mathbf{S}$ as $\mathbf{S}^2 = \mathbf{SS}$ – without the second $\mathbf{S}$ being transposed. After the success of Potter's approach, however, the term "square root" for a solution $\mathbf{P}^{1/2}$ of Eq. (10.82) has become common usage in Kalman filtering.

Potter's implementation of the Kalman filter in terms of $\mathbf{P}^{1/2}$ has come to be called *the Potter square root filter*. It has been characterized as "achieving the same accuracy with half as many bits" [26], and it has spawned a host of alternative Kalman filter implementations[5] with improved robustness against computer roundoff errors. These have come to utilize a variety of methods for creating and manipulating matrix square roots.

10.5.4 Square Root Matrix Manipulation Methods

These are digital computation methods for linear algebra problems in square root filtering that have been found to be particularly robust against computer roundoff error.

10.5.4.1 Cholesky Decomposition

Factorization or **decomposition** refers to methods for factoring matrices as products of two or more matrices with useful properties. "Decomposition" is the older established term in mathematics, but some objected to its meaning outside mathematics and substituted "factorization" instead.

André-Louis Cholesky (1875–1918) was a French mathematician, geodesist, and artillery officer killed in WWI and credited [29] with the discovery of an improved solution to the linear least-squares problem in geodesy (determining the shape of geopotential surfaces). Cholesky's algorithms factored the $n \times n$ Gramian matrix

$$\mathbf{G} = \mathbf{H}^T\mathbf{H} \tag{10.83}$$

[5] See, e.g. [27] and [28] for a sampling of alternative implementations, including some with cross correlation between $\mathbf{w}_k$ and $\mathbf{v}_k$.

of Eq. (10.9) as a symmetric product of a triangular[6] matrix $\mathbf{\Delta}$ *with reversal of the order of the transposed factor*[7]:

$$\mathbf{G} = \mathbf{\Delta}\mathbf{\Delta}^T \tag{10.84}$$

$$\mathbf{\Delta} = \begin{bmatrix} \delta_{11} & \delta_{12} & \delta_{13} & \cdots & \delta_{1n} \\ 0 & \delta_{22} & \delta_{23} & \cdots & \delta_{2n} \\ 0 & 0 & \delta_{33} & \cdots & \delta_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \delta_{nn} \end{bmatrix} \quad \text{(upper triangular)} \tag{10.85}$$

or

$$= \begin{bmatrix} \delta_{11} & 0 & 0 & \cdots & 0 \\ \delta_{21} & \delta_{22} & 0 & \cdots & 0 \\ \delta_{31} & \delta_{32} & \delta_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \delta_{n1} & \delta_{n2} & \delta_{n3} & \cdots & \delta_{nn} \end{bmatrix} \quad \text{(lower triangular)} \tag{10.86}$$

Cholesky's algorithm for computing the triangular factors in Eq. (10.84) came to be called *Cholesky decomposition* or *Cholesky factorization*, and the triangular factors $\mathbf{\Delta}$ are called *Cholesky factors.*

The built-in MATLAB function `chol` returns the upper triangular Cholesky factor $\mathbf{\Delta}^\star$ for the alternative factorization

$$\mathbf{G} = \mathbf{\Delta}^{\star T}\mathbf{\Delta}^\star$$

which is the lower triangular solution to Eq. (10.84). Therefore, `Delta = chol(G)';` should return the **lower triangular** solution of Eq. (10.84).

The **upper triangular** solution of Eq. (10.84) is returned by the MATLAB function `utchol` in the m-file `utchol.m` on www.wiley.com/go/grewal/gnss.

10.5.4.2 Modified Cholesky Decomposition

This is a factoring of the form

$$\mathbf{P} = \mathbf{U}\mathbf{D}\mathbf{U}^T \tag{10.87}$$

where $\mathbf{D}$ is a diagonal matrix with positive diagonal elements and $\mathbf{U}$ is a *unit triangular* matrix.

[6] Linear solutions of triangular systems require $O(n^2)$ arithmetic operations, compared with $O(n^3)$ for non-triangular systems, which tends to favor using triangular square roots of matrices wherever it is possible.

[7] Reversal of the order of the transposed factor is required here for cleaving to Potter's definition of his matrix square root. Cholesky had things the other way around.

10.5.4.3 Nonuniqueness of Matrix Square Roots

Matrix square roots (defined in Eq. (10.1)) are not unique and can be transformed into one another by orthogonal transformations (Eq. (10.3)). This relationship with orthogonal transformations is exploited in square root filter implementations.

10.5.4.4 Triangularization by QR Decomposition

This is a factorization of a matrix **M** into a matrix product **M** = **QR**, where **Q** is orthogonal and **R**[8] is triangular. It is usually traced to the Gram–Schmidt orthogonalization methods of Jørgen Pedersen Gram [30] and Erhard Schmidt [31], although it may have had its roots in works by Cauchy and Laplace, and its numerical stability has been improved by methods due to Wallace Givens [32] and Alston Householder [33]. Also, a modified form of Gram–Schmidt orthogonalization was used by Catherine L. Thornton [34] for implementing the time update of the Bierman–Thornton UD filter (described in the following text).

QR decomposition was originally derived with the orthogonal matrix factor coming first but has been generalized to allow it to come last – the form needed in square root filtering. The MATLAB m-file `housetri.m` on www.wiley.com/go/grewal/gnss performs the triangularization of a matrix **M** through a series of Householder transformations $\mathbf{\Omega}_j$ such that the result

$$\begin{bmatrix} \mathbf{\Delta} & 0 \end{bmatrix} = \mathbf{M} \times \mathbf{\Omega}_1 \times \mathbf{\Omega}_2 \times \cdots \times \mathbf{\Omega}_N \tag{10.88}$$

where $\mathbf{\Delta}$ is a triangular matrix. Applied to the matrix $\mathbf{P}^{1/2}_{k(-)}$ from Eq. (10.89), it returns the upper triangular square root of $\mathbf{P}_{k(-)}$ for the Potter square root filter with dynamic disturbance noise.

10.5.4.5 Householder Triangularization

Alston S. Householder (1904–1993) devised a robust method for triangularizing rectangular matrices using symmetric orthogonal matrices of the sort

$$\mathbf{\Omega}_H(\mathbf{u}) = \mathbf{I} - \frac{2\mathbf{u}\mathbf{u}^T}{|\mathbf{u}|^2}$$

with carefully selected vectors **u** [33]. His algorithm is implemented in the m-file `housetri.m`, on www.wiley.com/go/grewal/gnss.

10.5.5 Alternative Square Root Filter Implementations

10.5.5.1 Potter Implementation

Potter's derivation of the Kalman filter in terms of a square root of the covariance matrix assumed scalar measurements and did not include dynamic disturbance noise (not an issue for space travel without Klingons). Measurement vector decorrelation (described in Section 10.5.2.1) takes care of

[8] We will use different notation to avoid confusion with Kalman's use of **Q** for the covariance of dynamic disturbance noise and **R** for the covariance of measurement noise.

the scalar measurement issue and triangularization methods (Section 10.5.4.4) take care of the dynamic disturbance noise issue.

Measurement Update Potter's algorithm is implemented in the m-file `potter.m` on www.wiley.com/go/grewal/gnss. It does not preserve triangularity of the matrix square root, but that is corrected by the time update.

Time Update This starts with the block matrix[9]

$$\mathbf{B}_{k(-)} = \left[\boldsymbol{\Phi}_k \mathbf{P}^{1/2}_{k-1(+)} \ \mathbf{Q}^{1/2}_k \right] \tag{10.89}$$

the "Potter square" of which

$$\begin{aligned} \mathbf{B}_{k(-)}\mathbf{B}^T_{k(-)} &= \boldsymbol{\Phi}_k \mathbf{P}^{1/2}_{k-1(+)} \mathbf{P}^{T/2}_{k-1(+)} \boldsymbol{\Phi}^T_k + \mathbf{Q}^{1/1}_k \mathbf{C}^{T/2}_k \\ &= \boldsymbol{\Phi}_k \mathbf{P}_{k-1(+)} \boldsymbol{\Phi}^T_k + \mathbf{Q}_k \\ &= \mathbf{P}_{k(-)} \end{aligned} \tag{10.90}$$

Therefore, triangularization of $\mathbf{B}_{k(-)}$ by multiplying it on the right by orthogonal transformations will produce

$$\mathbf{B}_{k(-)}\boldsymbol{\Omega} = \left[\mathbf{P}^{1/2}_{k(-)} \ 0 \right] \tag{10.91}$$

where $\mathbf{P}^{1/2}_{k(-)}$ is triangular. The implementation in the m-file `housetri.m` on www.wiley.com/go/grewal/gnss produces an upper triangular result.

10.5.5.2 Carlson "Fast Triangular" Square Root Filter

This algorithmic implementation by Neal A. Carlson [36] maintains the square root matrix in triangular form, which reduces the computational requirements relative to that of the Potter implementation. It essentially modifies the Potter "rank-one modification" algorithm for the measurement update so that it maintains triangularity of the square root matrix. This cuts the computational requirements nearly in half. The MATLAB m-file `carlson.m` on www.wiley.com/go/grewal/gnss implements Carlson's measurement update. The time update is the same as Potter's.

10.5.5.3 Bierman–Thornton UD Filter

An alternative square root filter implementation that avoids the numerical calculation of square roots was developed by Gerald J. Bierman (1941–1987) and Catherine L. Thornton. It uses the alternative factorization

$$\mathbf{P} = \mathbf{U}\mathbf{D}\mathbf{U}^T \tag{10.92}$$

where the matrix **U** is *unit triangular* and **D** is a diagonal matrix with positive diagonal elements. Unit triangular matrices have ones along their main

[9] Equation (10.89) is due to Schmidt [35].

diagonal and zeros on one side of it. In this case, the zeros are below the main diagonal and **U** is called *unit upper triangular*. The time update was derived by Thornton [34], based on what is called *modified Gram–Schmidt orthogonalization*. It is implemented in the m-file `thornton.m` on www.wiley.com/go/grewal/gnss. The measurement update is implemented in the m-file `bierman.m`. See [26] for a more detailed examination of its performance characteristics.

The m-file `udu.m` factors a symmetric positive definite matrix into its **U** and **D** factors. The m-file `UD_decomp.m` does this in-place, storing **D** on the main diagonal (which is otherwise known to contain only ones) and the other elements of **U** above the main diagonal.

Avoidance of scalar square roots can make a difference in implementations on processors without arithmetic square root instructions or microcoded square roots. The condition number of **U** in this decomposition tends to be relatively favorable for numerical stability, but that of **D** may scale up as that of **P**. This issue can be addressed in fixed-point implementations by rescaling.

10.5.5.4 Unscented Square Root Filter

The unscented Kalman filter (UKF) is a sampling-based approximation for nonlinear propagation means and square roots of covariance matrices ($\mathbf{P}^{1/2}$ or $\mathbf{R}^{1/2}$) using a set of sample values structured around the starting values of the estimate and its covariance. The square-root implementation of this filter approximates the square root of the nonlinearly transformed covariance using similar sampling strategies. This is discussed in Sections 10.6 and 10.6.5.3.

10.5.5.5 Square Root Information Filter (SRIF)

The *square root information filter* is usually abbreviated as *SRIF*. (The conventional square root filter is often abbreviated as *SRCF*, which stands for *square-root covariance filter*.) Like the SRCF, the SRIF is more robust against roundoff errors than the inverted covariance form of the filter.

Like most square root filters using triangularization, it admits more than one solution form. A complete formulation (i.e. including both updates) of the SRIF was developed by Dyer and McReynolds [37], using triangularization methods developed by Golub and Van Loan [20] and applied to sequential least-squares estimation by Lawson and Hanson [38]. The form developed by Dyer and McReynolds is shown in Table 10.3 [Ref. [18] in Chapter 4].

10.6 Nonlinear Approximations

For many of the estimation problems of practical interest, linearity tends to be more of an exception than the rule. However, this has not dissuaded

Table 10.3 Square-root information filter using triangularization. $\mathbf{\Omega}_{\Delta\mathrm{meas}}$ and $\mathbf{\Omega}_{\Delta\mathrm{temp}}$ are orthogonal matrices which lower triangularize the left-hand side matrices. Submatrices represented by "■" on the right-hand sides are extraneous.

Measurement update

$$\begin{bmatrix} \mathbf{Y}_{k(-)} & \mathbf{H}_k^T\mathbf{R}_k^{-1/2} \\ \hat{\mathbf{y}}_{k(-)}^T & \mathbf{z}_k^T\mathbf{R}_k^{-1/2} \end{bmatrix} \mathbf{\Omega}_{\Delta\mathrm{meas}} = \begin{bmatrix} \mathbf{Y}_{k(+)} & 0 \\ \hat{\mathbf{y}}_{k(+)}^T & \blacksquare \end{bmatrix}$$

Time update

$$\begin{bmatrix} \mathbf{Q}_k^{-1/2} & -\Phi_k^{-\mathrm{T}}\mathbf{Y}_{k(+)}^{1/2} \\ 0 & \Phi_k^{-\mathrm{T}}\mathbf{Y}_{k(+)}^{1/2} \\ 0 & \hat{\mathbf{y}}_{k(+)}^T \end{bmatrix} \mathbf{\Omega}_{\Delta\mathrm{temp}} = \begin{bmatrix} \blacksquare & 0 \\ \blacksquare & \mathbf{Y}_{k+1(-)}^{1/2} \\ \blacksquare & \hat{\mathbf{y}}_{k+1(-)}^T \end{bmatrix}$$

problem-solvers from using approximations to apply the more successful linear estimation methods to less-than-linear problems.

In the case of the Kalman filter, this generally involves transformations of the means and covariances of error distributions, the bread-and-butter of Kalman filtering. This approximation problem is "ill posed" in the mathematical sense. That is, there may be no unique solution. This is because nonlinear transformations of different PDFs with identical means and covariances can result in PDFs with distinctly different means and covariances. This issue of what might be called "ambiguity errors" is addressed in Appendix C (on www.wiley.com/go/grewal/gnss).

A method for assessing the magnitude of linearization errors is presented in the subsection 10.6.1, followed by some nonlinear implementations and approximation methods that have been used in Kalman filtering with relative success.

10.6.1 Linear Approximation Errors

The essential idea behind linear approximation of nonlinear PDF transformations is that, within reasonably expected variations of the state vector from its estimated value (as determined by the covariance of state estimation uncertainty), the mean-squared errors due to linearization should be dominated by the modeled uncertainties due to other sources. For measurement nonlinearities, the modeled uncertainties are characterized by the measurement noise covariance **R**. For dynamic nonlinearities, the modeled uncertainties are partly characterized by the covariance matrix **P** of estimation errors and by the dynamic disturbance noise covariance **Q**. The range of perturbations under which these conditions need to be met can be specified in terms of the expected magnitude of uncertainty in the estimate (e.g. within $\pm 3\sigma$).

The resulting statistical conditions for linearization can be stated in the following manner:

1. For the time state transition function $\boldsymbol{\phi}(\mathbf{x})$, the linear approximation error should be insignificant compared with $\mathbf{Q}$ when the state vector variations $\boldsymbol{\delta}_x$ of $\hat{\mathbf{x}}$ are statistically significant. This condition can be met if the values of $\boldsymbol{\delta}_x$ are smaller than the $N\sigma$-values of the estimated distribution of uncertainty in the estimate $\hat{\mathbf{x}}$, which is characterized by the covariance matrix $\mathbf{P}$, for $N \geq 3$. That is, for
$$(\boldsymbol{\delta}_x)^T \mathbf{P}^{-1} (\boldsymbol{\delta}_x) \leq N^2 \tag{10.93}$$
the linear approximation error
$$\varepsilon_{\phi}(\boldsymbol{\delta}_x) \overset{\text{def}}{=} \underbrace{\boldsymbol{\phi}(\hat{\mathbf{x}} + \boldsymbol{\delta}_x) - \left[\boldsymbol{\phi}(\hat{\mathbf{x}}) + \left.\frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}} \boldsymbol{\delta}_x\right]}_{\text{approximation error}} \tag{10.94}$$
should be bounded by
$$\varepsilon_{\phi}^T(\boldsymbol{\delta}_x) \mathbf{Q}^{-1} \varepsilon_{\phi}(\boldsymbol{\delta}_x) \ll 1 \tag{10.95}$$
for $\boldsymbol{\delta}_x$ covering most of its expected range, in which case the nonlinear approximation errors should be dominated by modeled dynamic uncertainty.
2. *For the measurement/sensor transformation* $\boldsymbol{h}(\mathbf{x})$. for $N\sigma \geq 3\sigma$ perturbations of $\hat{\mathbf{x}}$, the linear approximation error is insignificant compared with $\mathbf{R}$. That is, for some $N \geq 3$, for all perturbations $\boldsymbol{\delta}_x$ of $\hat{\mathbf{x}}$ such that
$$(\boldsymbol{\delta}_x)^T \mathbf{P}^{-1} (\boldsymbol{\delta}_x) \leq N^2 \tag{10.96}$$
$$\varepsilon_{h}(\boldsymbol{\delta}_x) \overset{\text{def}}{=} \underbrace{h(\hat{\mathbf{x}} + \boldsymbol{\delta}_x) - \left[\boldsymbol{h}(\hat{\mathbf{x}}) + \left.\frac{\partial \boldsymbol{h}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}} \boldsymbol{\delta}_x\right]}_{\text{approximation error}} \tag{10.97}$$
$$\varepsilon_{h}^T(\boldsymbol{\delta}_x) \mathbf{R}^{-1} \varepsilon_{h}(\boldsymbol{\delta}_x) \ll 1 \tag{10.98}$$
for $\boldsymbol{\delta}_x$ covering most of its expected range, in which case the nonlinear approximation errors should be dominated by modeled measurement uncertainty.
The value of estimation uncertainty covariance $\mathbf{P}$ used in Eq. (10.96) would ordinarily be the a priori value, calculated before the measurement is used. If the measurement update uses what is called the *iterated extended Kalman filter* (IEKF), however, the a posteriori value can be used.
The range of variation $\boldsymbol{\delta}_x$ used in calculating the linear approximation error should encompass the "Nσ points" defined in terms of the

eigenvalue–eigenvector decomposition of **P** as

$$\mathbf{P} = \sum_{i=1}^{n} \sigma_i^2 \mathbf{e}_i \mathbf{e}_i^T \tag{10.99}$$

$$\sigma_i^2 \overset{\text{def}}{=} i\text{th eigenvalue of } \mathbf{P}$$

$$\mathbf{e}_i \overset{\text{def}}{=} i\text{th eigenvector of } \mathbf{P}$$

$$\boldsymbol{\delta}_{xi} \overset{\text{def}}{=} \begin{cases} N\sigma_i \mathbf{e}_i, & 1 \leq i \leq n \\ 0, & i = 0 \\ -N\sigma_{-i}\mathbf{e}_{-i}, & -n \leq i \leq -1 \end{cases} \tag{10.100}$$

which would be located at the major axes on the manifold of the $N\sigma$ equiprobability hyperellipse if the PDF were Gaussian, but otherwise represent the principal $N\sigma$ values of $\boldsymbol{\delta}$ associated with **P**.

The mean-squared linearization errors calculated using the Nσ points could then be approximated as

$$\mathrm{E}\langle|\varepsilon_\phi|^2\rangle \approx \frac{1}{2N+1} \sum_{i=-N}^{N} |\varepsilon_\phi(\boldsymbol{\delta}_{xi})|^2 \tag{10.101}$$

$$\mathrm{E}\langle|\varepsilon_h|^2\rangle \approx \frac{1}{2N+1} \sum_{i=-N}^{N} |\varepsilon_h(\boldsymbol{\delta}_{xi})|^2 \tag{10.102}$$

Example 10.2 ***(Nonlinearity of satellite pseudorange measurements)***
The GNSS measurement sensitivity matrix is based on the partial derivatives of the measured pseudorange $\rho_\psi = |\boldsymbol{\psi}|$ with respect to the location of the receiver antenna, with the partial derivative evaluated at the estimated receiver antenna location.

One can use Eq. (10.97) to determine whether the pseudorange measurement is sufficiently linear, given the uncertainty in the receiver antenna position, the nonlinear pseudorange measurement model, and the uncertainty in the pseudorange measurement.

This was evaluated for satellite elevation angles of 0°, 30°, 60°, and 90° versus RMS position error on the three axes. The resulting 3σ nonlinearity errors for these perturbations are plotted versus position error in Figure 10.2 as a function of σ_{pos} for each of the four satellite elevation angles above the horizon. The four barely distinguishable solid diagonal lines in the plot are for these four different satellite elevation angles, which apparently have little influence on nonlinearity errors. The horizontal line represents the RMS pseudorange noise, indicating that nonlinear approximation errors are dominated by pseudorange noise for 3σ position uncertainties less than ~2 km.

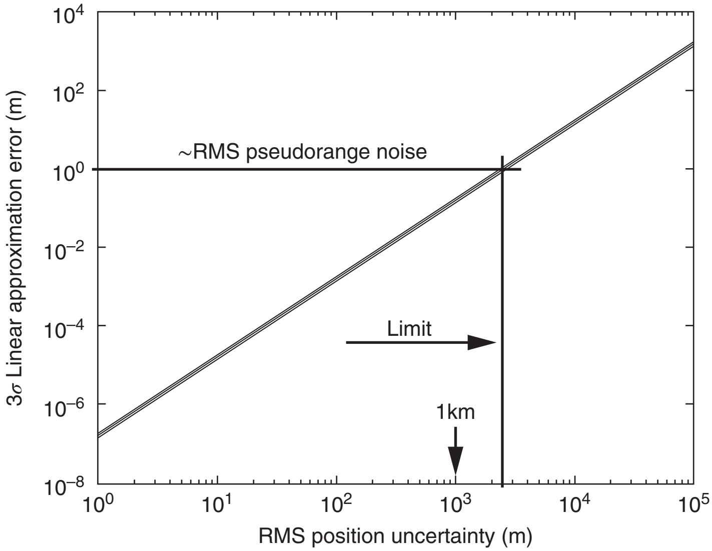


Figure 10.2 Linearization error analysis of pseudorange measurements.

10.6.2 Adaptive Kalman Filtering

This is a special class of nonlinear Kalman filtering problems in which some elements of the state vector are also parameters of the Kalman filter model. For example, making the scale factor of a sensor a part of the state vector makes the measurement a product of state vector components. This kind of nonlinearity, in particular, has sometimes been difficult to accommodate using extended Kalman filtering.

10.6.3 Taylor–Maclauren Series Approximations

Taylor–Maclauren series approximations for nonlinear Kalman filtering was suggested by Bucy [39] in 1965, after first-order (linear) approximations had been used in estimation for more than a century. Gauss first used his method of linear least squares in 1801 for solving a nonlinear problem: estimating the trajectory in space of the dwarf planet Ceres, and a first-order linearization of the Kalman filter had been introduced by Stanley F. Schmidt around 1960.

The Taylor–Maclauren series expansion of an infinitely continuously differentiable scalar function f of a scalar variable x in the vicinity of $x = x_0$ has the form

$$f(x) = \sum_{\ell=0}^{\infty} \frac{1}{\ell!} \left. \frac{\partial^\ell f}{\partial x^\ell} \right|_{x=x_0} (x - x_0)^\ell \tag{10.103}$$

However, when the variable and the function are n-vector-valued the equivalent mathematical representations for the ℓth-order derivatives become multidimensional structures with dimensions

$$\underbrace{n \times n \times n \times \cdots \times n}_{\ell+1 \text{ times}}$$

and the complexity of the resulting formulas soon become overwhelming. Therefore, the summed series is usually truncated after the first-order or second-order term.

10.6.3.1 First-Order: Extended Kalman Filter

Schmidt used first-order partial derivatives *evaluated at the estimated value of the state vector* for what is now called the *extended* Kalman filter. The possibly altered implementation equations are listed in Table 10.4. However, only the nonlinear parts need be altered. That is, if only the measurement is nonlinear, then only that part needs to be altered, and similarly for the time update.

Iterated Extended Kalman Filtering The measurement update can be iterated, using the partial derivative of $\mathbf{h}_k$ reevaluated at $\mathbf{x}_{k(+)}$ for a second guess at the approximation to $\mathbf{H}_k$.

10.6.3.2 Second-Order: Bass–Norum–Schwartz Filter

A second-order Taylor–Maclauren series approximation for nonlinear Kalman filtering was derived by Bass et al. [40, 41] in 1965. Its computational complexity and potential numerical instability have limited its acceptability, however [41].

10.6.4 Trajectory Perturbation Modeling

In the earliest application of the Kalman filter, Stanley F. Schmidt had linearized the space navigation problem by using Newton's laws to compute the first-order variations in the future position and velocity of a spacecraft due to variations

Table 10.4 Extended Kalman filter modifications

Extended Kalman filter: altered implementation equations	
Measurement update	Time update
$\boldsymbol{\nu}_k \approx \mathbf{z}_k - \mathbf{h}_k(\mathbf{x}_{k(-)})$	$\hat{\mathbf{x}}_{k(-)} \approx \mathbf{f}_k(\hat{\mathbf{x}}_{k-1(+)})$
$\mathbf{H}_k \approx \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}}\right\vert_{\mathbf{x}=\hat{\mathbf{x}}_{k(-)}}$	$\boldsymbol{\Phi}_k \approx \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{x}}\right\vert_{\mathbf{x}=\hat{\mathbf{x}}_{k-1(+)}}$

in the position and velocity at the current time. This produced a 6 × 6 state transition matrix Φ for a Kalman filter model in which the state vector elements are the six perturbations of position and velocity from the reference trajectory.

In the case of orbital trajectories, his first-order linearized model could be simplified to the two-body models of Kepler, and the state transition matrices could then be approximated by analytical partial derivatives based on Kepler's equations. There is no equivalent analytical model for more than two bodies, however.

10.6.5 Structured Sampling Methods

These could be characterized as *extended Kalman filtering*, in that they extended the idea of nonlinear transformation of the estimate to include nonlinear transformation of the estimate and structured samples representing an approximation of its covariance matrix of estimation uncertainty.

Sampling-based methods got their start in 1946 when mathematician Stanislaw Ulam (1909–1984) was playing Canfield solitaire while recovering from viral encephalitis, and pondered what were his odds of winning. He soon found the problem too complicated to work out mathematically with pencil and paper, thought of the way gamblers determine odds by playing with many independently shuffled decks, imagined doing it faster using computers, and discovered what is now called Monte Carlo analysis [42]. This became a powerful tool for numerical approximation of nonlinear transformation of PDFs. It was originally based on using independent random samples from known probability distributions and used for estimating what happens to those probability distributions when the variates undergo nonlinear transformations.

Sampling-based estimation methods have since come to include the use of samples structured in such a way as to reduce the numbers of samples required, but these still generally depend on knowing more about the probability distributions involved than just their means and covariances. Nevertheless, some using only means and covariances have performed well compared with extended Kalman filtering in nonlinear applications.

The term "particle filter" generally refers to Kalman filters using structured sampling of some sort, with the terms "sigma point" or "unscented" referring to specific forms of sampling and processing. When used in Kalman filtering, the samples represent a PDF with mean equal to the estimate and covariance representing the second central moment about the mean. Therefore, they will generally be of the form

$$\boldsymbol{\sigma}_i = \hat{\mathbf{x}} + \boldsymbol{\delta}_i$$

where $\boldsymbol{\delta}$ is a sample from a zero-mean PDF. Then the transformed mean is approximated as a weighted sum of the transformed samples and the transformed covariance is approximated by a weighted sum of the relevant outer products:

$$\mathbf{y} \overset{\text{def}}{=} \mathbf{f}(\mathbf{x}) + \mathbf{w} \tag{10.104}$$

$$\begin{aligned}\hat{\mathbf{y}} &\overset{\text{def}}{=} \mathrm{E}\langle \mathbf{y} \rangle \\ &\approx \sum_i \zeta_i \mathbf{f}(\boldsymbol{\sigma}_i)\end{aligned} \tag{10.105}$$

$$\begin{aligned}\mathbf{P}_{yy} &\overset{\text{def}}{=} \mathrm{E}\langle (\mathbf{y} - \hat{\mathbf{y}})(\mathbf{y} - \hat{\mathbf{y}})^T \rangle \\ &\approx \sum_i \zeta_i [\mathbf{f}(\boldsymbol{\sigma}_i) - \hat{\mathbf{y}}][\mathbf{f}(\boldsymbol{\sigma}_i) - \hat{\mathbf{y}}]^T + \mathrm{E}\langle \mathbf{w}\mathbf{w}^T \rangle\end{aligned} \tag{10.106}$$

where the ζ_i are positive weighting factors, generally constrained so that the correct linear results are obtained when $\mathbf{y}$ is linear.

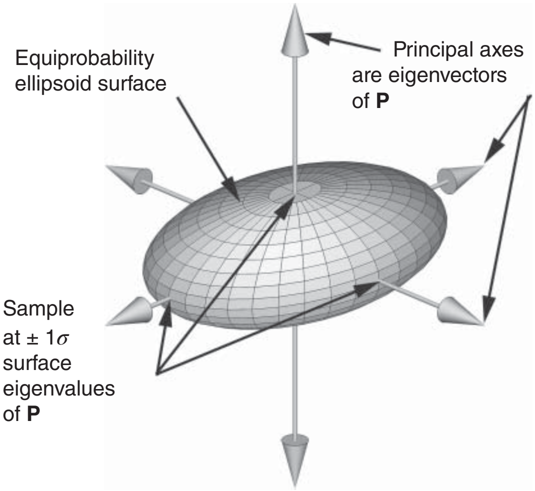
These generally perform better than the extended Kalman filter. However, they do introduce errors that are not accounted for in the covariance of estimation uncertainty.

In practice, Kalman filtering tends to be conservative in its treatment of covariance matrices, biasing its estimates of estimation uncertainty to the high side just a bit to avoid filter divergence due to low Kalman gains.

10.6.5.1 Sigma-Point Filters

Sigma-points of Gaussian distributions are on the 1σ equiprobability ellipsoid at its principal axes, which are the eigenvectors of its covariance matrix – as illustrated in Figure 10.3 for a three-dimensional Gaussian distribution. In sigma-point filters, another sample is added at the center of the distribution, which is its mean.

Figure 10.3 Sigma-points for $n = 3$.


Equiprobability ellipsoid surface
Principal axes are eigenvectors of **P**
Sample at ± 1σ surface eigenvalues of **P**

The samples used for a PDF with mean $\hat{\mathbf{x}}_{k-1(+)}$ and $n \times n$ covariance matrix $\mathbf{P}_{k-1(+)}$ with eigenvalue–eigenvector decomposition

$$\mathbf{P}_{k-1(+)} = \sum_{i=1}^{n} \lambda_i \mathbf{e}_i \mathbf{e}_i^T \tag{10.107}$$

are at the $2n + 1$ vectors

$$\boldsymbol{\sigma}_0 = \hat{\mathbf{x}}_{k-1(+)} \tag{10.108}$$

$$\boldsymbol{\sigma}_i = \hat{\mathbf{x}}_{k-1(+)} + \sqrt{\lambda_i}\mathbf{e}_i \tag{10.109}$$

$$\boldsymbol{\sigma}_{-i} = \hat{\mathbf{x}}_{k-1(+)} - \sqrt{\lambda_i}\mathbf{e}_i \tag{10.110}$$

for $1 \leq i \leq n$, where the λ_i are the eigenvalues of $\mathbf{P}_{k-1(+)}$ and the $\mathbf{e}_i$ are the corresponding unit eigenvectors.

Sigma-Point Transform For any transformation $\mathbf{f}(\mathbf{x})$ of the variate $\mathbf{x}$, the sigma-point approximated mean is

$$\hat{\mathbf{x}}_{k(-)} \approx \frac{1}{2n+1} \sum_{i=-n}^{+n} \mathbf{f}(\boldsymbol{\sigma}_i) \tag{10.111}$$

and the approximated transformed covariance about the approximated mean is

$$\mathbf{P}_{k(-)} \approx \frac{1}{2n+1} \sum_{i=-n}^{+n} [\mathbf{f}(\boldsymbol{\sigma}_i) - \hat{\mathbf{x}}_{k(-)}][\mathbf{f}(\boldsymbol{\sigma}_i) - \hat{\mathbf{x}}_{k(-)}]^T \tag{10.112}$$

Note that Eqs. (10.111) and (10.112) produce the basic Kalman filter time update equations when $\mathbf{f}(\mathbf{x}) = \boldsymbol{\Phi}\mathbf{x}$ for $\boldsymbol{\Phi}$ an $n \times n$ matrix. That is, the sigma-point filter becomes the basic Kalman filter when the transformation is linear.

10.6.5.2 Particle Filters

There are several generalizations of the sigma-point filter, using different factorizations (e.g. Cholesky decomposition) of the covariance matrix and different weightings of the samples. Some are still called "sigma-point" filters, although they do not use conventional sigma points as samples.

10.6.5.3 The Unscented Kalman Filter

The term "unscented" was proposed by Jeffrey Uhlmann as an alternative to naming his approach the *Uhlman Kalman filter*. Uhlmann had begun development of the UKF in the 1990s by analyzing the performance of different sigma-point sampling and weighting strategies for large-scale nonlinear estimation problems. His analysis showed no significant variation in performance related to the particular square root factorization of the initial covariance matrix, and established some weighting strategies that can be "tuned" to improve performance for different applications. The results of this

Table 10.5 Unscented transform samples and weights.

Sampling strategy	Sample size[a)]	Sample values[b)]	Sample weights[c)]	Index values
Simplex	$n + 2$	$\boldsymbol{\sigma}_0 = \hat{\mathbf{x}}$	$0 \leq \zeta_0 \leq 1$(can be varied)	
		$\boldsymbol{\sigma}_i = \boldsymbol{\sigma}_0 + \boldsymbol{\gamma}_i$	$\zeta_1 = 2^{-n}(1 - \zeta_0)$	
		$\boldsymbol{\Gamma} = \mathbf{C}\boldsymbol{\Psi}$	$\zeta_2 = \zeta_1$	
		$\boldsymbol{\Psi} = [\boldsymbol{\psi}_1 \ \boldsymbol{\psi}_2 \ \cdots \ \boldsymbol{\psi}_{n+1}]$	$\zeta_i = 2^{i-1}\zeta_1$	$3 \leq i \leq n + 1$
Symmetric	$2n$	$\boldsymbol{\sigma}_i = \hat{\mathbf{x}} + \sqrt{n}\ \mathbf{c}_i$	$\zeta_i = 1/(2n)$	$1 \leq i \leq n$
		$\boldsymbol{\sigma}_{i+n} = \hat{\mathbf{x}} - \sqrt{n}\ \mathbf{c}_i$	$\zeta_{i+n} = 1/(2n)$	$1 \leq i \leq n$
	$2n + 1$	$\boldsymbol{\sigma}_0 = \hat{\mathbf{x}}$	$\zeta_0 = \kappa/\sqrt{n + \kappa}$	
		$\boldsymbol{\sigma}_i = \hat{\mathbf{x}} + \sqrt{n + \kappa}\ \mathbf{c}_i$	$\zeta_i = 1/[2(n + \kappa)]$	$1 \leq i \leq n$
		$\boldsymbol{\sigma}_{i+n} = \hat{\mathbf{x}} - \sqrt{n + \kappa}\ \mathbf{c}_i$	$\zeta_{i+n} = 1/[2(n + \kappa)]$	$1 \leq i \leq n$
Scaled	$2n + 1$	$\boldsymbol{\sigma}_0 = \hat{\mathbf{x}}$	$\zeta_0 = \lambda/(n + \lambda)$	
		$\lambda = \alpha^2(n + \kappa) - n$	$\zeta_0^\star = \zeta_0 + (1 - \alpha^2 + \beta)$	
		$\boldsymbol{\sigma}_i = \hat{\mathbf{x}} + \sqrt{n + \lambda}\ \mathbf{c}_i$	$\zeta_i = 1/[2(n + \kappa)]$	$1 \leq i \leq n$
		$\boldsymbol{\sigma}_{n+i} = \hat{\mathbf{x}} - \sqrt{n + \lambda}\ \mathbf{c}_i$	$\zeta_{n+i} = 1/[2(n + \kappa)]$	$1 \leq i \leq n$

a) n = dimension of state space.

b) $\mathbf{c}_i$ = ith column of a Cholesky factor $\mathbf{C}$ of $\mathbf{P}$. $\boldsymbol{\gamma}_i$ is the ith column of $\boldsymbol{\Gamma}$.

c) α, β, κ, and λ are "tuning" parameters, and $\zeta_0^\star$ is a separate weighting for covariance.

and collaborative studies is a suite of sample-based transforms, collectively called "unscented transforms," for approximating the means and covariances of PDFs after nonlinear transformations of the variate.

Examples of unscented transform strategies, in terms of the samples used and their associated weightings, are presented in Table 10.5. These are sample values $\boldsymbol{\sigma}_i$ and weightings ζ_i for estimating the nonlinearly transformed means and covariances using Eqs. (10.105) and (10.106) – with the exception that

$$\begin{aligned} \mathbf{P}_{yy} \approx & \sum_{i \neq 0} \zeta_i [\mathbf{f}(\boldsymbol{\sigma}_i) - \hat{\mathbf{y}}][\mathbf{f}(\boldsymbol{\sigma}_i) - \hat{\mathbf{y}}]^T + \zeta_0^\star [\mathbf{f}(\boldsymbol{\sigma}_0) - \hat{\mathbf{y}}][\mathbf{f}(\boldsymbol{\sigma}_0) - \hat{\mathbf{y}}]^T \\ & + \mathrm{E}\langle \mathbf{w}\mathbf{w}^T \rangle \end{aligned} \tag{10.113}$$

for the scaled unscented transform.

These methods have about the same computational complexity as the extended Kalman filter and have generally performed better for many nonlinear applications.

10.7 Diagnostics and Monitoring

10.7.1 Covariance Matrix Diagnostics

By definition, covariance matrices $\mathbf{P}$ are symmetric and positive semi-definite, meaning that $\mathbf{P} = \mathbf{P}^T$ and the eigenvalues of $\mathbf{P}$ are nonnegative. Neither of these is an issue with square root filtering, but they can be a problem in covariance or information filtering.

10.7.1.1 Symmetry Control

This is not a problem for square root filtering, although it was one of the symptoms of numerical instability leading to its discovery.

Verhaegen and Van Dooren [22] cite covariance matrix asymmetry as a destabilizing symptom, easily eliminated by computing only the upper or lower triangular submatrix of a covariance matrix, then forcing symmetry by storing results in the remaining symmetric positions. This also cuts the amount of computation nearly in half.

10.7.1.2 Eigenanalysis

Zero eigenvalues of a covariance matrix would imply perfect knowledge in some subspace of variates, which is practically impossible for Kalman filter applications. Therefore – if everything is working properly – **P** should have positive eigenvalues. The MATLAB function `eig` returns the eigenvalues and associated eigenvectors of a matrix. Eigenvectors corresponding to the largest eigenvalues are the directions in state space (i.e. combinations of state variables) with the greatest uncertainty.

10.7.1.3 Conditioning

The MATLAB function `cond` returns the ratio of the largest and smallest singular values of a matrix, which is a measure of its conditioning for inversion. The reserved variable `eps` is the smallest number ϵ for which $1 + \epsilon \neq 1$ in machine precision. The number `-log2(cond(P)*eps)` is about how many bits of precision will be useful in `inv(P)`. The only matrix inversion in Kalman filtering is $(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k(-)}\mathbf{H}_k^T)^{-1}$, but the conditioning of **P** may also be of interest in Kalman filter monitoring.

10.7.2 Innovations Monitoring

10.7.2.1 Kalman Filter Innovations

The intermediate ℓ_k-vector result

$$\boldsymbol{\nu}_k \overset{\text{def}}{=} \{\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k(-)}\} \tag{10.114}$$

from Eq. (10.45) is called "the *k*th innovation."[10] It is the difference between the current measurement $\mathbf{z}_k$ and the *expected measurement* $\mathbf{H}_k\hat{\mathbf{x}}_{k(-)}$ based on the a priori estimate $\hat{\mathbf{x}}_{k(-)}$. Its theoretical covariance

$$\mathrm{E}\langle\boldsymbol{\nu}_k\boldsymbol{\nu}_k^T\rangle = \mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k(-)}\mathbf{H}_k^T \tag{10.115}$$

[10] This notation is from Thomas Kailath, who used the name *innovation* and gave it the Greek letter $\boldsymbol{\nu}$ (nu) because it represents "what is new" about the *k*th measurement.

includes the influence of measurement noise ($\mathbf{R}_k$) and prior estimation errors ($\mathbf{H}_k\mathbf{P}_{k(-)}\mathbf{H}_k^T$). Its inverse

$$\boldsymbol{\Upsilon}_k = (\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k(-)}\mathbf{H}_k^T)^{-1} \tag{10.116}$$

is computed as a partial result in the computation of the Kalman gain in Eq. (10.44).

10.7.2.2 Information-Weighted Innovations Monitoring

This is a statistic used by Fred C. Schweppe [43] for signal detection using Kalman filter signal and noise models, and examined by Kailath and Poor [28] in a broader setting. Schweppe used the fact that these zero-mean innovations sequences, if they were Gaussian, would have the PDF

$$p(\boldsymbol{v}_k) = \frac{\det \boldsymbol{\Upsilon}_k}{\sqrt{(2\pi)^{\ell_k}}} \exp\left(-\frac{1}{2}\boldsymbol{v}_k^T\boldsymbol{\Upsilon}_k\boldsymbol{v}_k\right) \tag{10.117}$$

$$\log_e[p(\boldsymbol{v}_k)] = -\frac{1}{2}\boldsymbol{v}_k^T\boldsymbol{\Upsilon}_k\boldsymbol{v}_k + \log_e(\det\boldsymbol{\Upsilon}_k) - \frac{\ell_k}{2}\log_e(2\pi) \tag{10.118}$$

making $-\boldsymbol{v}_k^T\boldsymbol{\Upsilon}_k\boldsymbol{v}_k$ a likelihood function (or log-likelihood function) of $\boldsymbol{v}_k$.

However, this same statistic – without assuming gaussianity – can also be used to monitor the health of Kalman filters in terms of how well the model fits the data. It may be useful for monitoring overall Kalman filter goodness-of-fit, although it is generally not specific about likely sources of mis-modeling.

Given the innovations and their modeled information matrices, one can easily calculate the quadratic statistic

$$\chi_k^2 \overset{\text{def}}{=} \boldsymbol{v}_k^T\boldsymbol{\Upsilon}_k\boldsymbol{v}_k \tag{10.119}$$

which has the expected value

$$\mathrm{E}\langle\chi_k^2\rangle = \mathrm{E}\langle\boldsymbol{v}_k^T\boldsymbol{\Upsilon}_k\boldsymbol{v}_k\rangle \tag{10.120}$$

By using the properties of the matrix trace from Eq. (10.2), this can be maneuvered into the forms

$$\begin{aligned}\mathrm{E}\langle\chi_k^2\rangle &= \mathrm{E}\langle\mathrm{Tr}(\boldsymbol{v}_k^T\boldsymbol{\Upsilon}_k\boldsymbol{v}_k)\rangle\\ &= \mathrm{E}\langle\mathrm{Tr}(\boldsymbol{\Upsilon}_k\boldsymbol{v}_k\boldsymbol{v}_k^T)\rangle\\ &= \mathrm{Tr}(\boldsymbol{\Upsilon}_k\mathrm{E}\langle\boldsymbol{v}_k\boldsymbol{v}_k^T\rangle)\\ &= \mathrm{Tr}(\boldsymbol{\Upsilon}_k\boldsymbol{\Upsilon}_k^{-1})\\ &= \mathrm{Tr}(\mathbf{I}_{\ell_k})\\ &= \ell_k\end{aligned} \tag{10.121}$$

the dimension of the kth measurement vector. Even though we have not assumed gaussianity from Eq. (10.119) onward, this result is also the mean of a chi-squared distribution with ℓ_k degrees of freedom.

Example 10.3 ***(Simulated Innovations Monitoring)***

The m-file `ChiSqTest.m` on www.wiley.com/go/grewal/gnss calls the function `InnovChisqSim.m` (also on the website) to generate simulated innovations sequences from a Kalman filter model with

$$\boldsymbol{\Phi} = \begin{bmatrix} \exp(-1/3) & 0 & 0 \\ 0 & \exp(-1/9) & 0 \\ 0 & 0 & \exp(-1/27) \end{bmatrix} \tag{10.122}$$

$$\mathbf{Q} = (\mathbf{I} - \boldsymbol{\Phi}\boldsymbol{\Phi}^T) \tag{10.123}$$

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \tag{10.124}$$

$$\mathbf{R} = \mathbf{I} \tag{10.125}$$

This is a hypothetical stochastic model for demonstrating how underestimating noise covariance influences innovations. It models the state variable by three independent exponentially correlated processes with different correlation times, and noisy measurements of the sums of independent pairs, using the above Kalman filter model. The simulations use Gaussian pseudorandom noise with covariances equal to or greater than those assumed by the Kalman filter model (i.e. **Q** and **R**) to show the resulting relative shifts in the χ^2 statistic defined in Eq. (10.119) scaled relative to its expected value (ℓ_k, the dimension of the measurement vector). Results shown in Figure 10.4 include simulated values of χ^2_k/ℓ_k with all noise sources properly modeled (top plot), and the χ^2_k/ℓ_k values smoothed by a 100-sample moving average filter (bottom plot) for three cases:

1. All (simulated) noise sources modeled correctly.
2. Simulated dynamic disturbance noise doubled.
3. Simulated measurement noise doubled.

These results are intended to demonstrate that:

1. Raw values of χ^2_k/ℓ_k can be noisy (top plot).
2. Because χ^2_k/ℓ_k is theoretically a white noise process (i.e. uncorrelated) when the problem is properly modeled, averaged values are more steady (bottom plot).
3. Averaged values of χ^2_k/ℓ_k are close to 1 (one) if the problem is well-modeled (bottom plot).
4. Averaged values of χ^2_k/ℓ_k tend to increase if noise sources are underrated (bottom plot).

Innovations monitoring may offer a useful form of feedback for "tuning" Kalman filters by adjusting the model parameters.

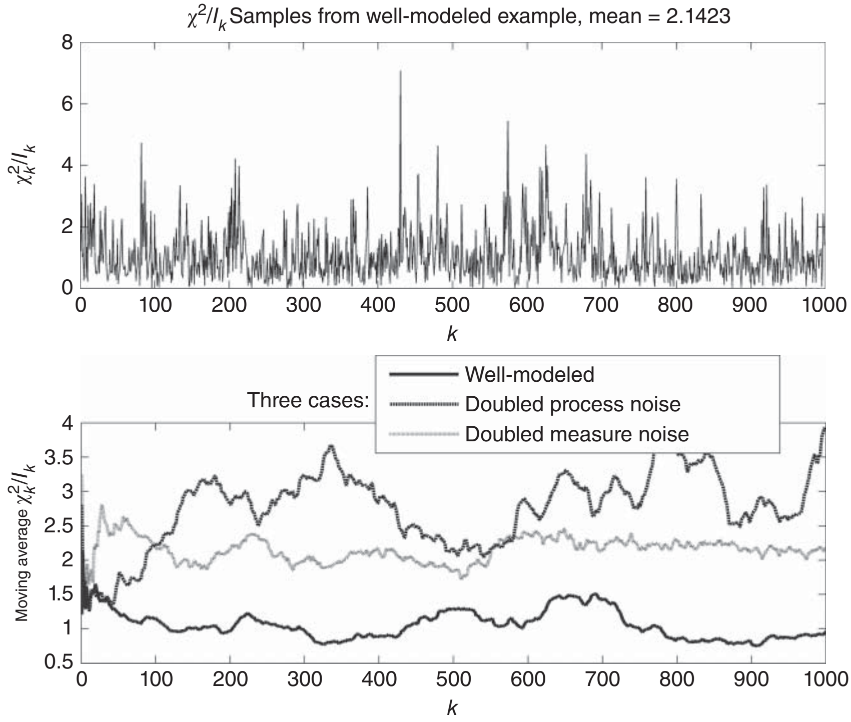


Figure 10.4 Simulated innovations monitoring.

10.8 GNSS-Only Navigation

Kalman filtering models required for any application include the following basic parameters:

Dynamic models defining the parameters
$\mathbf{\Phi}$, the state transition matrix
$\mathbf{Q}$, the covariance of dynamic disturbance noise

Measurement models defining the parameters
$\mathbf{H}$, the measurement sensitivity matrix
$\mathbf{R}$, the covariance of measurement noise

This section is about the ways in which these Kalman filter parameters can be defined for representing the GNSS navigation problem in different applications. These model parameter distinctions are important enough that many handheld GNSS receivers include separate built-in Kalman filter dynamic model parameters for walking, bicycling, or driving.

10.8.1 GNSS Dynamic Models

10.8.1.1 Receiver Clock Bias Dynamics

GNSS is a time-based navigation system depending on the accuracy of clocks in the satellites and those in the receivers. Satellite clock synchronization is maintained at the system level, but individual receivers are responsible for synchronizing their internal clocks to GNSS time. The receiver clock synchronization problem is common to all GNSS navigation applications.

Clocks are frequency sources. Those on satellites are generally expensive atomic clocks with tight environmental controls, but those in receivers are mostly lower-cost crystal oscillators with less stringent environmental controls. Maintaining synchronization with the GNSS satellite clocks is part of the receiver's Kalman filter implementation. This generally adds a few state variables to the Kalman filter.

These clock state variables include time and frequency offsets from GNSS time, but may also include frequency drift rate, which tends to be more stable over short time periods. In the latter case, a potential dynamic model for the clock bias "state" $\varepsilon_{\text{clock}}$ would be of the form

$$\left.\begin{aligned} \varepsilon_{\text{clock}} &= \begin{bmatrix} \varepsilon_{\text{bias}} \\ \varepsilon_{\text{freq}} \\ \varepsilon_{\text{drift}} \end{bmatrix} \\ \frac{d}{dt}\varepsilon_{\text{clock}} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1/\tau_{\text{drift}} \end{bmatrix} \varepsilon_{\text{clock}} + \begin{bmatrix} 0 \\ 0 \\ w_{\text{drift}}(t) \end{bmatrix} \end{aligned}\right\} \tag{10.126}$$

where τ_{drift} is the exponential correlation time of frequency drift rate and $w_{\text{drift}}(t)$ is the drift rate dynamic disturbance noise, with

$$\left.\begin{aligned} \mathrm{E}\langle w_{\text{drift}}(t)\rangle &= 0 \\ \mathrm{E}\langle w^2_{\text{drift}}(t)\rangle &= q_{\text{drift}} \\ \sigma^2_{\text{drift}} &\overset{\text{def}}{=} \lim_{t\to\infty} \mathrm{E}\langle \varepsilon^2_{\text{drift}}(t)\rangle \\ &= \frac{\tau_{\text{drift}} q_{\text{drift}}}{2} \\ \mathrm{E}\langle \varepsilon_{\text{drift}}(t)\varepsilon_{\text{drift}}(t+\Delta t)\rangle &= \sigma^2_{\text{drift}} e^{-|\Delta t|/\tau_{\text{drift}}} \end{aligned}\right\} \tag{10.127}$$

This dynamic system is unstable in the variables $\varepsilon_{\text{bias}}$ and $\varepsilon_{\text{freq}}$, but stable in $\varepsilon_{\text{drift}}$. However, the Kalman filter estimate of all three variables can still be stable – depending on the measurements used.

The critical clock parameters for this part of the Kalman filter implementation are the mean-squared frequency drift rate σ^2_{drift} and its correlation time τ_{drift}, which are generally determined by comparing test clocks to references standard clocks over long time periods.

10.8.1.2 Discrete Time Models

We have used the dynamic coefficient matrix **F** for continuous-time dynamics here because GNSS measurement times t_k are the GNSS signal time-mark arrival times, which may not be predictable and regularly spaced in time. The times between such events,

$$\Delta t_k = t_k - t_{k-1} \tag{10.128}$$

are not always predictable, and the equivalent state transition matrices $\mathbf{\Phi}$ in discrete time,

$$\mathbf{\Phi}_k = \exp\left(\int_{t_{k-1}}^{t_k} \mathbf{F}(t)dt\right) \tag{10.129}$$

depend on Δt_k.

10.8.1.3 Exponentially Correlated Random Processes

Equations (10.127) define the linear time-invariant dynamic model for what is called an *exponentially correlated random process*, which is used extensively in Kalman filtering as a model for time-correlated zero-mean random processes with bounded mean-squared values. Its equivalent form in discrete time is

$$\left.\begin{array}{rcl} \varepsilon_k &=& \exp(-\Delta t/\tau)\varepsilon_{k-1} + w_k \\ \tau &>& 0 \\ \mathrm{E}\langle w_k\rangle &=& 0 \\ \mathrm{E}\langle w_k^2\rangle &=& q \\ \sigma_\infty^2 &\overset{\text{def}}{=}& \lim\limits_{k\to\infty} \mathrm{E}\langle \varepsilon_k^2\rangle \\ &=& \dfrac{q}{1-\exp(-2\Delta t/\tau)} \\ \mathrm{E}\langle \varepsilon_k \varepsilon_j\rangle &=& \sigma_\infty^2 \exp(-|t_k - t_j|/\tau) \end{array}\right\} \tag{10.130}$$

where the essential design parameters are the correlation time-constant τ and σ_∞^2, the steady-state variance of error. The process noise variance q can be determined from τ and σ_∞^2, the steady-state variance of the exponentially correlated errors.

10.8.1.4 Host Vehicle Dynamics for Standalone GNSS Navigation

This is for navigation using only GNSS without an INS, in which case the dynamics of the host vehicle is defined only in terms of random process models.

Besides the clock correction variables, the other essential state variables for GNSS navigation are those determining the location of the receiver antenna relative to navigation coordinates – which are generally earth-fixed for terrestrial applications. The receiver antenna is the "holy point" for GNSS navigation. It is the point where all satellite signals are received, and its location relative to those of the satellite antennas determines the relative timing of the satellite signals at that point. Any signal delays between the antenna and the signal processor, plus processing delays, are common to all signals. These delays may not affect the navigation solution, but they do contribute to the solution time-lag.

For most applications, the dynamics of the antenna are determined by the dynamics of the host vehicle.[11] For applications in which the host vehicle has significant rotational freedom and the offsets between the antenna and the center of gravity of the host vehicle are large compared to measurement error, rotational dynamics of the host vehicle may also be part of the model. One of the advantages of GNSS/INS integration for this case is that an INS is at least an attitude sensor and – for strapdown INS – an attitude rate sensor.

10.8.1.5 Point Mass Dynamic Models

State vector dimension is determined by the degrees of freedom of movement of the host vehicle and by its mechanics of movement. GNSS measurements can only track the location of the receiver antenna, so the simplest models for host vehicle tracking are based on point-mass dynamics.

Velocity Random Walk Models The simplest usable tracking models are for point-mass dynamics including translational momentum mechanics, which adds one state variable (velocity) for each degree of freedom. In this case, the dynamic disturbance model has the form

$$\left.\begin{aligned} \mathbf{x}_{\text{dim}} &= \begin{bmatrix} x_{\text{pos}} \\ x_{\text{vel}} \end{bmatrix} \\ \frac{d}{dt}\mathbf{x}_{\text{dim}} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}_{\text{dim}} + \begin{bmatrix} 0 \\ w_{\text{acc}}(t) \end{bmatrix} \end{aligned}\right\} \tag{10.131}$$

for each dimension of travel.

Exponentially Correlated Velocity Models A point-mass stochastic model for random time-correlated host vehicle velocity with RMS value σ_{vel} and exponential correlation time τ has the form

$$\frac{d}{dt}\begin{bmatrix} x_{\text{pos}} \\ x_{\text{vel}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1/\tau \end{bmatrix} \begin{bmatrix} x_{\text{pos}} \\ x_{\text{vel}} \end{bmatrix} + \begin{bmatrix} 0 \\ w_{\text{acc}} \end{bmatrix} \tag{10.132}$$

$$\mathrm{E}\langle w_{\text{acc}} \rangle = 0 \tag{10.133}$$

$$\mathrm{E}\langle w_{\text{acc}}^2 \rangle = \frac{2\sigma_{\text{vel}}^2}{\tau} \tag{10.134}$$

for each axis of freedom, but with potentially different values for τ and v_{rms} for different axes.

[11] There are scientific applications in which the "host vehicle" is a bit of bedrock or glacier, in which case the state vector dynamic rates might be centimeters per year or meters per day.

Exponentially Correlated Acceleration Models These resemble the clock bias model of Eqs. (10.126) and (10.127) for each dimension of travel:

$$\left.\begin{aligned}
\mathbf{x}_{\text{dim}} &= \begin{bmatrix} x_{\text{pos}} \\ x_{\text{vel}} \\ x_{\text{acc}} \end{bmatrix} \\
\frac{d}{dt}\mathbf{x}_{\text{dim}} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1/\tau_{\text{acc}} \end{bmatrix} \mathbf{x}_{\text{dim}} + \begin{bmatrix} 0 \\ 0 \\ w_{\text{jerk}}(t) \end{bmatrix} \\
\mathrm{E}\langle w_{\text{jerk}}(t)\rangle &= 0 \\
\mathrm{E}\langle w^2_{\text{jerk}}(t)\rangle &= q_{\text{jerk}} \\
\sigma^2_{\text{acc}} &\overset{\text{def}}{=} \lim_{t\to\infty} \mathrm{E}\langle x^2_{\text{acc}}(t)\rangle \\
&= \frac{\tau_{\text{acc}} q_{\text{jerk}}}{2} \\
\mathrm{E}\langle x_{\text{acc}}(t) x_{\text{acc}}(t+\Delta t)\rangle &= \sigma^2_{\text{acc}}\ e^{-|\Delta t|/\tau_{\text{acc}}}
\end{aligned}\right\} \qquad (10.135)$$

where the subscript “jerk” refers to the derivative of acceleration, which is called “jerk.” This model may be more realistic for vehicles occupied by humans, who tend to become uncomfortable at acceleration levels greater than 3g (∼30 m/s/s). It is also appropriate for wheeled vehicles, whose acceleration capabilities are limited by tire-to-surface contact stiction/friction to be generally < 1g. (An exception is in auto racing, where downward aerodynamic force is used to boost the tire limit to the driver’s limit of ∼3G.)

Models for Bounded RMS Velocity and Acceleration In this case, a linear drag coefficient d is used for bounding the RMS velocity in the above model, in which case the lower-right 2 × 2 submatrices of the steady-state covariance matrix $\mathbf{P}_\infty$ and its associated dynamic coefficient matrix

$$\mathbf{P}_{\infty\ 2\times2} = \begin{bmatrix} v^2_{\text{rms}} & p_{1,2} \\ p_{1,2} & a^2_{\text{rms}} \end{bmatrix} \qquad (10.136)$$

$$\mathbf{F}_{2\times2} = \begin{bmatrix} -d & 1 \\ 0 & -\tau^{-1} \end{bmatrix} \qquad (10.137)$$

satisfy the steady-state equation in continuous time:

$$\begin{aligned} 0 &= \frac{d}{dt}\mathbf{P}_{\infty\ 2\times2} \\ &= \mathbf{F}_{2\times2}\mathbf{P}_{\infty\ 2\times2} + \mathbf{P}_{\infty\ 2\times2}\mathbf{F}^T_{2\times2} + \begin{bmatrix} 0 & 0 \\ 0 & q \end{bmatrix} \end{aligned} \qquad (10.138)$$

which can be solved for

$$q = \frac{2\ a^2_{\text{rms}}}{\tau} \qquad (10.139)$$

$$d = \frac{\sqrt{4\ \tau^2 a_{\mathrm{rms}}^2 + v_{\mathrm{rms}}^2} - v_{\mathrm{rms}}}{2\ \tau\ v_{\mathrm{rms}}} \tag{10.140}$$

$$p_{1,2} = v_{\mathrm{rms}}^2\ d \tag{10.141}$$

where the q in Eq. (10.139) is for the dynamic model in continuous time. The equivalent $\mathbf{Q}_k$ in discrete time is given by Eq. (10.65).

This model has been converted to its equivalent implementation in discrete time in the MATLAB m-files `avRMS.m` and `avRMStestfunction.m`, a sample output from which is shown in Figure 10.5. This is from an example with 0.5g RMS acceleration with one minute correlation time, 8 m/s RMS velocity, one second discrete time steps, and 1000 simulated steps. Figure 10.5 shows the RMS position, velocity, and acceleration from the riccati equation solution, demonstrating steady-state behavior in velocity and acceleration, and unstable RMS uncertainty in position.

10.8.2 GNSS Measurement Models

10.8.2.1 Measurement Event Timing

GNSS receivers process satellite signals to compute the differences between the times the signals left their respective satellite antennas (encoded in each satellite signal) and the times when the same signals arrived at the receiver antenna

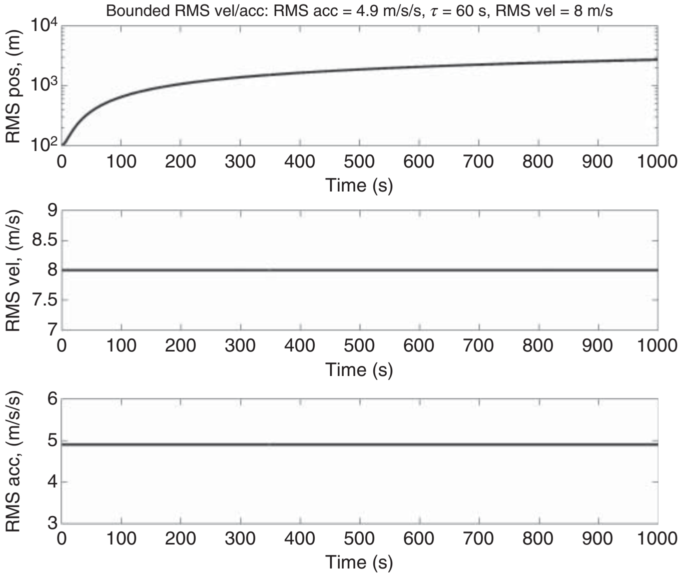


Figure 10.5 Sample 3 × 1 subplot output from `avRMStestfunction.m`, showing RMS uncertainties.

(determined by the receiver's clock). The effective measurement time t_k in this case is the time at which the signal timing mark arrived at the receiver antenna. The discrete time intervals between measurements are not necessarily uniform or predictable.

10.8.2.2 Pseudoranges

Those timing differences, multiplied by the propagation speed of the signal, are the *pseudoranges* between the satellite antennas at the transmission times and the receiver antenna at the reception times – as illustrated in Figure 10.6. For GPS, these total time differences are generally in the order of $\sim 75 \pm 10$ ms, with each nanosecond representing about 0.3 m of pseudorange distance.

10.8.2.3 Time and Distance Correlation

Some of the system error sources contributing to measurement error in GNSS pseudoranges are time-correlated, as well as distance correlated. However, we may not have good data on this for GPS III until after the next-generation upgrades are completed. Early on in the evolution of GPS, data collected worldwide showed a dominant exponentially time-correlation in the order of an hour and distance correlation in the order of 100 m [44]. We will not know any relevant correlation model parameters for upgraded and developing GNSS until some time after these upgrades are finished. The candidate dynamic model for GNSS navigation given below includes a single exponential time-correlation parameter, just in case it is still significant.

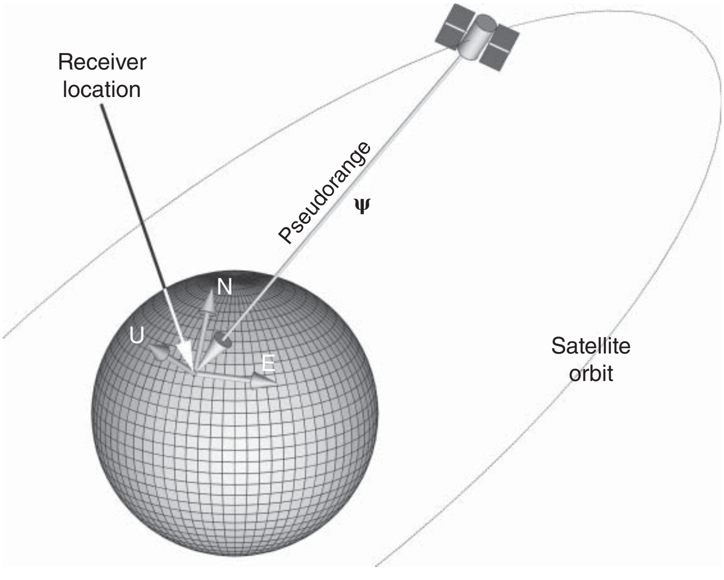


Figure 10.6 Pseudorange measurement geometry.

10.8.2.4 Measurement Sensitivity Matrix

Pseudoranges are scalar measurements, so the associated measurement sensitivity matrices **H** have but one row. At most four of the elements h_{1j} of **H** are nonzero, corresponding to

$\varepsilon_{\mathbf{bias}}$, the receiver clock time offset from GNSS time. This element of **H** will have the value $1/c$, where c is the signal propagation velocity.

$x_{\text{dim }1}$, the first dimension of receiver antenna location in navigation coordinates.

$x_{\text{dim }2}$, the second dimension of receiver antenna location in navigation coordinates.

$x_{\text{dim }3}$, the third dimension of receiver antenna location in navigation coordinates.

The last three of these elements of **H** will be the three components of the unit vector

$$\mathbf{u}_\psi = \frac{\boldsymbol{\psi}}{|\boldsymbol{\psi}|}$$

in the direction from the satellite antenna toward the receiver antenna, representing the sensitivity of calculated pseudorange to estimated receiver antenna location errors. This calculation requires that the satellite antenna position and receiver antenna position be represented in a common coordinate frame, which can be navigation coordinates (earth-centered earth-fixed).

10.8.2.5 Noise Model

The only noise statistics that matter in Kalman filtering are means and covariances of the Kalman model after time-correlated errors have been converted to white noise sources driving dynamic systems.

This can vary somewhat from one GNSS constellation to another, may be different from one satellite to another within the same GNSS constellation (as is the measurement sensitivity matrix **H**), and may include time-correlated and satellite-to-satellite correlated errors. The modeling of time-correlated measurement errors requires that at least one state variable be included for each satellite being used. The simplest time-correlated model for one satellite would be one in which there is a single exponentially time-correlated noise component ξ, with continuous-time and discrete-time models

$$\frac{d}{dt}\xi(t) = \frac{-1}{\tau}\xi + w(t) \tag{10.142}$$

$$\xi_{k(-)} = \phi_k \xi_{k-1(+)} + w_k \tag{10.143}$$

$$\phi_k = \exp\left(-\frac{t_k - t_{k-1}}{\tau}\right) \tag{10.144}$$

$$\mathrm{E}\langle w_k^2 \rangle = q_k \tag{10.145}$$

where τ is the correlation time-constant. In the time-invariant case in which $t_k - t_{k-1} = \Delta t$ is constant, the steady-state mean-squared noise will be the solution of the steady-state riccati equation

$$\begin{aligned} \sigma_\infty^2 &= e^{-2\ \Delta t/\tau}\sigma_\infty^2 + q \\ &= \frac{q}{1 - e^{-2\Delta t/\tau}} \end{aligned} \tag{10.146}$$

This might be a suitable model for system-level errors such as satellite clock errors or ephemeris errors, and it requires adding the variables ξ for each satellite to the state vector. There will still be uncorrelated measurement noise, as well, due to such sources as background noise, processing roundoff errors, or electromagnetic interference.

Example 10.4 ***(State Vector Example for Standalone GNSS Navigation)***
This is an example for what the Kalman filter state vector might look like for 3D GNSS navigation with a clock drift rate model, a host vehicle with jerk-noise dynamics and exponentially correlated pseudorange measurement noise on each of 30 satellites. Table 10.6 identifies the 42 elements of the resulting state vector in earth-fixed locally-level east-north-up coordinates. The first three state variables model receiver clock errors, the next nine

Table 10.6 Example state vector for standalone GNSS navigation.

Number	Symbol	Description
1	$x^\star_{\text{bias}}$	Receiver clock bias
2	x_{freq}	Receiver clock frequency
3	x_{drift}	Receiver clock drift rate
4	$x^\star_{\text{pos}\ E}$	Receiver antenna easting
5	$x_{\text{vel}\ E}$	Receiver antenna east velocity
6	$x_{\text{acc}\ E}$	Receiver antenna east acceleration
7	$x^\star_{\text{pos}\ N}$	Receiver antenna northing
8	$x_{\text{vel}\ N}$	Receiver antenna north velocity
9	$x_{\text{acc}\ N}$	Receiver antenna north acceleration
10	$x^\star_{\text{pos}\ U}$	Receiver antenna altitude
11	$x_{\text{vel}\ U}$	Receiver antenna vertical velocity
12	$x_{\text{acc}\ U}$	Receiver antenna vertical acceleration
13	$x^\star_{\text{sat}\ 1}$	Satellite #1 signal delay
14	$x^\star_{\text{sat}\ 2}$	Satellite #2 signal delay
⋮	⋮	⋮
42	$x^\star_{\text{sat}\ 30}$	Satellite #30 signal delay

model host vehicle dynamics, and the remaining 30 model time-correlated measurement noise. The starred variables (e.g. $x^{\star}_{\text{bias}}$) are the only components with nonzero entries at the corresponding positions in the associated 1 × 42 measurement sensitivity matrix **H**. This example gives some idea of what time-correlated measurement errors can do to the dimension of the state vector.

10.9 Summary

1. In navigation problems, the *navigation solution* must include values for all those variables needed for getting the host vehicle to its destination. These must include position of the vehicle being navigated but may also include its velocity, orientation, or any other attributes of interest.
2. Kalman filtering is used in navigation for estimating the current value of the navigation solution, given all measurements made up to the current time. It is also used for a host of other tracking problems related to navigation, including:
   (a) Keeping track of the precise orbits of all the GNSS satellites in a constellation, using measurements made from ground stations and between the satellites in orbit.
   (b) Using signals received from GNSS satellites to track the motions of the receiver antenna.
   (c) Using GNSS and INS together to obtain a better navigation solution and to attain some degraded level of navigational accuracy when GNSS signals are not available.
3. The Kalman filter itself is an algorithm based on:
   (a) The LLMSE of Carl Friedrich Gauss, which minimizes the mean squared error in the estimated navigation solution, using only
      (i) The linear relationships between measured variables and the navigation solution variables
      (ii) The means and variances (or covariances) of all error distributions, including
         A. Measurement noise and
         B. Estimation errors.
   (b) A recursive form of the linear least mean squared estimator, which
      (i) Allows the measurements to be processed one at a time, as they come in.
      (ii) Maintains the current value of the linear-least-mean-squares estimate, based on all measurements made up to the current time.
      (iii) Maintains the current value of estimation error covariance, which is needed for updating the estimate as measurements come in.

    (iv) Minimizes the mean-square estimation error, given all measurements up to the current time.

  (c) Linear stochastic process models (due to Andrey Andreyevich Markov) for representing
    (i) The known linear dynamic model for the host vehicle being navigated, plus
    (ii) Mean-squared contributions to navigation uncertainty due to random disturbances of the host vehicle.
  (d) Linear algebra (lots of it).

4. The Kalman filter does not depend on any attributes of probability distributions beyond their first and second moments. As a consequence, it cannot be used for calculating probabilities – unless one further assumes a particular parametric form of the PDF (e.g. Gaussian).
5. The basic implementation equations for the Kalman filter – as derived by Kalman [2] – are summarized in Table 10.1.
6. Those nonlinear matrix equations related to the propagation of the covariance matrix of estimation uncertainty are called *riccati equations*, named after the Venetian mathematician who first showed that they can be converted to linear equations by a change of variables. These can be solved independently from the estimation equations and used for predicting the expected estimation accuracy for a given measurement scenario.
7. Digital implementation of the Kalman filter in finite precision introduce roundoff errors that can degrade the accuracy of the solution to some degree, depending on the digital wordlength of the implementation and certain algebraic properties of the matrices involved – especially covariance matrices.
8. James E. Potter introduced the idea of using "square roots" of covariance matrices in alternative representations of Kalman filtering, which effectively cut digital wordlength requirements in half for achieving the same level of numerical accuracy.
9. Square root Kalman filter implementations also make use of orthogonal matrix transformations to improve robustness against roundoff errors, which is especially important for problems with large numbers of unknowns to be estimated. This includes Kalman filter implementations for keeping track of GNSS constellations.
10. Linearity of the dynamic equations and measurements is essential for Kalman filtering. However, it has been used with considerable success in quasi-linear applications, in which the dynamics and measurements are dominated by linearity within the essential range of the probability distributions involved. (See Appendix C for additional coverage.)
11. Kalman filters for GNSS navigation generally require stochastic models for random dynamics of the receiver antenna, the receiver clock error, and any time-correlated signal delay errors.

12. Monitoring the run-time health of a Kalman filter may include statistical assessment of measurement *innovations* (differences between predicted and actual measurements) and checking on algebraic properties of covariance matrices.

Problems

10.1 Can you name a PDF without a defined mean or variance? Why would this distinction be important in Kalman filtering?

10.2 What are the main reasons why the Kalman filter depends only on the means and covariances of PDFs?

10.3 Why are means and covariances of PDFs not sufficient when nonlinear transformations are used?

10.4 (Random Walk) Let the discrete stochastic process $\{x_k | k = 1,2,3,\dots\}$ be defined by

$$x_{k+1} = x_k + w_k, \quad \mathrm{E}\langle w_k \rangle = 0 \quad \text{for all} \quad k, \quad \mathrm{E}\langle w_k^2 \rangle = 1$$

(a) What is the equivalent state transition matrix for this system?
(b) If $\mathrm{E}\langle x_1 \rangle = 0$, what is $\mathrm{E}\langle x_k \rangle$ for $k = 2, 3, 4, 5$?
(c) If $\mathrm{E}\langle x_1^2 \rangle = 1$, what is $\mathrm{E}\langle x_k^2 \rangle$ for $k = 2, 3, 4, 5$?
(d) Are random walks stable?

10.5 (Random Walk with Measurements) In Problem 10.9.4 let there be measurements $z_k = h_k x_k + v_k$ with

$$Q = \mathrm{E}\langle w_k^2 \rangle = 1, \quad h = 1, \quad \mathrm{E}\langle v_k \rangle = 0, \quad \mathrm{E}\langle v_k^2 \rangle = 2 \quad \text{for all} \quad k$$

(a) If $P_{1(-)} = \mathrm{E}\langle x_1^2 \rangle = 2$, what will be the value of $P_{1(+)}$ after the measurement z_1 has been processed?
(b) What will be the value of $P_{2(-)}$ before the next measurement z_2 is processed?
(c) What will be the value of $P_{2(+)}$ after the measurement z_2 has been processed?
(d) What will be the values of $P_{k(-)}$ and $P_{k(+)}$ for $k = 1, 2, 3, 4$?
(e) In this example, is the estimation uncertainty of an (unstable) random walk stable?

10.6 Let

$$\mathbf{F} \overset{\text{def}}{=} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

(a) What is

$$\mathbf{M}(t) = \int_0^t \mathbf{F}(s)ds?$$

(b) What is $\mathbf{M}^2(t)$?

(c) What is $\mathbf{M}^m(t)$ for $m > 1$?

(d) What is its matrix exponential

$$\exp(\mathbf{M}(t)) \overset{\text{def}}{=} \sum_{m=0}^{\infty} \frac{1}{m!}\mathbf{M}^m(t)$$

also known as a fundamental solution matrix for $\dot{\mathbf{x}} = \mathbf{F}\mathbf{x}$?

10.7 Prove that Eqs. (10.111) and (10.112) produce the basic Kalman filter temporal update equations when $\mathbf{f}(\mathbf{x}) = \mathbf{\Phi}\mathbf{x}$ for $\mathbf{\Phi}$ an $n \times n$ matrix.

10.8 Consider a scalar time-invariant dynamic system model or the sort

$$\begin{aligned} x_k = \phi x_{k-1} + w_{k-1}, \quad &\text{where} \quad \mathrm{E}\langle w_m\rangle = 0 \\ \text{and} \quad \mathrm{E}\langle w_m^2\rangle = q > 0 \quad &\text{for all} \quad m \end{aligned} \tag{10.147}$$

Under what conditions on ϕ is this dynamic system stable or unstable? (See Section 10.4.5.)

10.9 Can the Kalman estimates of the state of an unstable dynamic system be stable (i.e. bounded)?

10.10 If $\phi = \frac{1}{2}$ and $q = \frac{1}{2}$ in Eq. (10.147), the initial variance of uncertainty

$$p_0 \overset{\text{def}}{=} \mathrm{E}\langle x_0^2\rangle = 1$$

and there are no measurements, what is p_1?

10.11 Under the same conditions as in Problem 10.10, is there a steady-state solution for p_∞? If so, what is it?

10.12 Run the MATLAB function `avRMStestfunction.m` on www.wiley.com/go/grewal/gnss with the following parameter values:

Units	m/s/s	s	m/s	s	#	Possible
Inputs	`aRMS`	`tau`	`vRMS`	`Delta`	`Steps`	applications?
`avRMStestfunction(`	`2,`	`10,`	`10,`	`1,`	`1000)`	OFF-ROAD?
`avRMStestfunction(`	`4,`	`30,`	`20,`	`1,`	`1000)`	PAVEMENT-LIMITED?
`avRMStestfunction(`	`10,`	`30,`	`30,`	`1,`	`1000)`	HUMAN OCCUPANT?
`avRMStestfunction(`	`20,`	`10,`	`100,`	`1,`	`1000)`	ROCKET?

Observe the resulting multi-plots of RMS values and simulated examples for this model with bounded RMS acceleration and velocity. Note, however, that RMS position relative to the starting point is constantly increasing.

Here, the RMS (1σ) values can be used to make physical limits the $\sim 3\sigma$ values. For example,

- Off-road traction limits acceleration to be considerably less than $1g$ (9.8 m/s/s).
- Except for racing cars, pavement contact limits horizontal acceleration to about $1g$. Racing cars can utilize aerodynamic forces to boost that to the driver limit of about $3g$.
- Humans start to feel uncomfortable at around $3g$ for more than a few seconds.
- Unmanned hit-to-kill missile defense rockets can accelerate to dozens of *G*s and many times the speed of sound.

10.13 Run the m-file `ChiSqTest.m` on www.wiley.com/go/grewal/gnss to generate plots of the innovations statistic χ_k^2/ℓ_k for simulated measurement noise (Eq. (10.125)) and simulated dynamic disturbance noise (Eq. (10.123)) scaled up a factor of 2. The first of the three plots generated is for a well-modeled Kalman filter, and the second and third plots are for mis-scaled noise covariances. If everything is working properly, the outputs in all cases should be close to 1. Are these levels of mis-modeling apparent from the (simulated) values of χ_k^2/ℓ_k?

10.14 Run the m-file `GNSSonly.m` on the www.wiley.com/go/grewal/gnss. It simulates GNSS-only navigation of a vehicle traveling 100 kph on a 100 km figure-8 test track with a tunnel under the overpass where the tracks cross over one another, so that GNSS signals are lost briefly at that time. It also compares the RMS Monte Carlo simulation errors of the Kalman filter implementation with the riccati equation solution. The error buildup in the tunnel is largely determined by the stochastic model representing vehicle dynamics, which is rather simplistic in this case.

References

1 Gauss, C.F. (1995). In: *Theoria Combinationis Observationum Erribus Minimus Obnoxiae* (trans. G.W. Stewart). Philadelphia, PA: SIAM.

2 Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *ASME Transactions, Series D: Journal of Basic Engineering* 82: 35–45.

3 Levy, L.J. (1977). The Kalman filter: integrations's workhorse. *GPS World* (September 1977), pp. 65–71.

4 Gelb, A. (ed.) (1974). *Applied Optimal Estimation.* Cambridge, MA: MIT Press.

5 Antoulas, A.C (ed.) (1991). *Mathematical System Theory: The Influence of R.E. Kalman.* Berlin: Springer-Verlag.

6 Stigler, S.M. (1981). Gauss and the invention of least squares. *Annals of Statistics* 9 (3): 465–474.

7 Gauss, C.F. (1963). In: *Theoria Motus Corporum Coelestium* (trans. C.H. Davis). New York: Dover.

8 Legendre, A.M. (1805). *Nouvelles méthodes pour la détermination des orbites des comètes.* Paris: Didot.

9 Adrian, R. (1808). Research concerning the probabilities of the errors which happen in making observations, &c. *Analyst* 1 (4): 93–109.

10 Cooley, J.W. and Tukey, J.W. (1865). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation* 19: 297–301.

11 Markov, A.A. (1898). The law of large numbers and the method of least squares. *Izvestia Fizika-Mathematika Obschestva Kazan University* 8 (2): 110–128 (in Russian).

12 Aitken, A.C. (1935). On least squares and linear combinations of observations. *Proceedings of the Royal Society of Edinburgh* 55: 42–48.

13 Duncan, W.J. (1944). Some devices for the solution of large sets of simultaneous linear equations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, Series 7* 35: 660–670.

14 Guttman, L. (1946). Enlargement methods for computing the inverse matrix. *Annals of Mathematical Statistics* 17 (3): 336–343.

15 Jarrow, R. and Protter, P. (2004). A short history of stochastic integration and mathematical finance: the early years, 1880–1970. *Institute of Mathematical Statistics, Monograph Series* 45: 75–91.

16 Itô, K. (1946). Stochastic integral. *Proceedings of the Imperial Academy Tokyo* 20 (1944): 519–524.

17 Jazwinski, A.H. (2009). *Stochastic Processes and Filtering Theory.* Dover Publications (reprint of 1970 edition by Academic Press).

18 Schmidt, S.F. (1981). The Kalman filter – its recognition and development for aerospace applications. *AIAA Journal of Guidance, Control, and Dynamics* 4 (1): 4–7.

19 Stewart, G.W. (1998). *Matrix Algorithms: Basic Decompositions,* vol. 1, 1e. Philadelphia, PA: SIAM.

20 Golub, G.H. and Van Loan, C.F. (2013). *Matrix Computations,* 4e. Baltimore, MD: Johns Hopkins University Press.

21 Kaminski, P.G. (1971). Square root filtering and smoothing for discrete processes. PhD thesis. Stanford, CA: Stanford University.

22 Verhaegen, M. and Van Dooren, P. (1986). Numerical aspects of different Kalman filter implementations. *IEEE Transactions on Automatic Control* AC-31: 907–917.

23 Fisher, R.A. (1925). Theory of statistical estimation. *Mathematical Proceedings of the Cambridge Philosophical Society* 22 (5): 700–725.

24 Grewal, M.S. and Kain, J. (2010). Kalman filter implementation with improved numerical properties. *IEEE Transactions on Automatic Control* 55 (9): 2058–2068.

25 Yu, W., Keegan, R.G., and Hatch, R.R. (2018). Satellite navigation receiver with fixed point sigma Rho filter. US Patent 20180313958, 1 November 2018.

26 Bierman, G.J. (2017). *Factorization Methods for Discrete Sequential Estimation*. New York: Dover.

27 Morf, M. and Kailath, T. (1975). Square root algorithms for least squares estimation. *IEEE Transactions on Automatic Control* AC-20: 487–497.

28 Kailath, T. and Poor, H.V. (1998). Detection of stochastic processes. *IEEE Transactions on Information Theory* 44 (6): 2230–2259.

29 Commendat Benoit (1924). Sur une méthode de résolution des équations normales provenant de l'application de la méthode des moindes carrés a un système d'équations linéaires en numbre inférieur a celui des inconnues—application de la m.éthode a la resolution d'un système defini d'équations linéaires (Procédé du Commandant Cholesky). *Bulletin Géodêsique et Géophysique Internationale* 2: 67–77.

30 Gram, J.P. (1883). Uber die Entwickelung reeler Funtionen in Reihen mittelst der Methode der kleinsten Quadrate. *Journal fur die reine und angewandte Mathematik* 94: 71–73.

31 Schmidt, E. (1907). Zur Theorie der linearen und nichtlinearen Integralgleichungen. I. Teil: Entwicklung willkulicher Funktionen nach Systemen vorgeschriebener. *Mathematische Annalen* 63: 433–476.

32 Givens, W. (1958). Computation of plane unitary rotations transforming a general matrix to triangular form. *Journal of the Society for Industrial and Applied Mathematics* 6 (1): 26–50.

33 Householder, A.S. (1958). Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM* 5 (4): 339–342.

34 Thornton, C.L. (1976). Triangular covariance factorizations for Kalman filtering. PhD thesis. University of California at Los Angeles, School of Engineering.

35 Schmidt, S.F. (1970). Computational techniques in Kalman filtering. In: *Theory and Applications of Kalman Filtering*, AGARDograph 139. London: NATO Advisory Group for Aerospace Research and Development.

36 Carlson, N.A. (1973). Fast triangular formulation of the square root filter. *AIAA Journal* 11 (9): 1259–1265.

37 Dyer, P. and McReynolds, S. (1969). Extension of square-root filtering to include process noise. *Journal of Optimization Theory and Applications* 3: 444–458.

38 Lawson, C.L. and Hanson, R.J. (1974). *Solving Least Squares Problems.* Englewood Cliffs, NJ: Prentice-Hall.

39 Bucy, R.S. (1965). Nonlinear filtering theory. *IEEE Transactions on Automatic Control* 10 (2): 198.

40 Bass, R., Norum, V., and Schwartz, L. (1966). Optimal multichannel nonlinear filtering. *J. Math. Anal. Appl.* 16: 152–164.

41 Bucy, R.S. and Lo, J.T. (1998). Seminal contributions of Bob Bass to control. Proceedings of the 1998 American Control Conference. IEEE.

42 Ulam, S.M. (1976). *Adventures of a Mathematician.* New York: Charles Scribner's Sons.

43 Schweppe, F.C. (1965). Evaluation of likelihood functions for Gaussian signals. *IEEE Transactions on Information Theory* 11: 61–70.

44 Bierman, G.S. (1995). Error modeling for differential GPS. MS thesis. MIT.