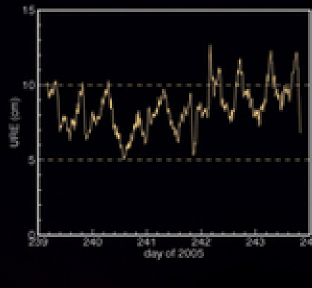
ADVANCED KALMAN FILTERING, LEAST-SQUARES AND MODELING



BRUCE P. GIBBS

WILEY

ftp:// SITE AVAILABLE

ADVANCED KALMAN FILTERING, LEAST-SQUARES AND MODELING

About the Cover

The watercolor portrait of Carl Friedrich Gauss was made by J.C.A. Schwartz in 1803 (photo courtesy of Axel Wittmann, copyright owner). This is the decade when Gauss first used the least squares method to determine orbits of asteroids and comets. The 1963 picture of Rudolf Kalman in his office at the Research Institute for Advanced Studies was taken after he first published papers describing the Kalman filter (photo courtesy of Rudolf Kalman, copyright owner). The GPS IIF spacecraft is the latest operational version of the spacecraft series (picture courtesy of The Boeing Company, copyright owner). The GPS ground system uses a Kalman filter to track the spacecraft orbits and clock errors of both the spacecraft and ground monitor stations. A least squares fit is used to compute the navigation message parameters that are uplinked to the spacecraft and then broadcast to user receivers. GPS receivers typically use a Kalman filter to track motion and clock errors of the receiver. The plot shows the root-mean-squared *user range error* (URE) for 29 GPS satellites operational in 2005. Those URE values were computed using smoothed GPS orbit and clock estimates as the "truth" reference (see Example 9.6 of Chapter 9).

ADVANCED KALMAN FILTERING, LEAST-SQUARES AND MODELING

A Practical Handbook

BRUCE P. GIBBS
Carr Astronautics, Inc.

WILEY

A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey
Published simultaneously in Canada

Library of Congress Cataloging-in-Publication Data:

Gibbs, Bruce. P., 1946–
Advanced kalman filtering, least-squares and modeling : a practical handbook / Bruce. P. Gibbs.
p. cm.
Includes bibliographical references.
ISBN 978-0-470-52970-6 (cloth)
1. Engineering–Statistical methods. 2. Kalman filtering. 3. Least squares. 4. Mathematical models. I. Title.
TA340.G49 2010
620.0072′7–dc22

2010013944

Printed in Singapore

oBook ISBN: 978-0-470-89004-2
ePDF ISBN: 978-0-470-89003-5

10 9 8 7 6 5 4 3 2 1

This book is dedicated to the memory of Gerald Bierman,
Larry Levy, and James Vandergraft.
They contributed much to the field, and are greatly missed.

CONTENTS

Most of the software used for the examples in this book is available online at ftp://ftp.wiley.com/public/sci_tech_med/least_squares/. Chapter 13, "Software Documentation," Appendix C, "Selected First-Principle Concepts," and Appendix D, "Discrete-Time (ARMAX-type) Models" are also available at this location.

PREFACE

Another book on Kalman filtering and least-squares estimation—are there not enough of them? Well, yes and no. Numerous books on the topic have been published, and many of them, both old and recent, are excellent. However, many practitioners of the field do not appear to fully understand the implications of theory and are not aware of lessons learned decades ago. It often appears that model structure was not carefully analyzed and that options for improving performance of the estimation were not considered. Available books present information on optimal estimation theory, standard implementations, methods to minimize numerical errors, and application examples, but information on model development, practical considerations, and useful extensions is limited.

Hence the first goal of this book is to discuss model development in sufficient detail so that the reader may design an estimator that meets all application requirements and is somewhat insensitive to modeling errors. Since it is sometimes difficult to *a priori* determine the best model structure, use of *exploratory data analysis* to define model structure is discussed. Methods for deciding on the "best" model are also presented. The second goal is to present little known extensions of least-squares estimation or Kalman filtering that provide guidance on model structure and parameters, or that reduce sensitivity to changes in system behavior. The third goal is discussion of implementation issues that make the estimator more accurate or efficient, or that make it flexible so that model alternatives can be easily compared. The fourth goal is to provide the designer/analyst with guidance in evaluating estimator performance and in determining/correcting problems. The final goal is to provide a subroutine library that simplifies implementation, and flexible general purpose high-level drivers that allow both easy analysis of alternative models and access to extensions of the basic estimation.

It is unlikely that a book limited to the above goals would be widely read. To be useful, it must also include fundamental information for readers with limited knowledge of the subject. This book is intended primarily as a handbook for engineers who must design practical systems. Although it could be used as a textbook, it is not intended for that purpose since many excellent texts already exist. When discussions of theory are brief, alternate texts are mentioned and readers are encouraged to consult them for further information. Most chapters include real-world examples of topics discussed. I have tried to approach the topics from the practical implementation point of view used in the popular *Applied Optimal Estimation* (Gelb 1974) and *Numerical Recipes* (Press 2007) books.

Why am I qualified to offer advice on optimal estimation? I was very lucky in my career to have been in the right place at the right time, and had the great fortune of working with a number of very talented people. The reputation of these people allowed our organizations to consistently obtain contracts involving R&D for innovative state-of-the-art estimation. In several cases our implementations were—to our knowledge—the first for a particular application. Some of the approximately 40 estimation applications I have worked on include:

1. Spacecraft orbit determination (batch least squares, Kalman filtering, and smoothing) and error analysis
2. Spacecraft attitude determination
3. Determination of imaging instrument optical bias and thermal deformation
4. Combined spacecraft orbit and optical misalignment determination
5. Monitoring of crustal movement using spaceborne laser ranging
6. Ship and submarine tracking (active and passive)
7. Aircraft tracking (active and passive)
8. Missile tracking using radar
9. Real-time tank maneuvering target tracking using multiple hypothesis testing
10. Tank gun tube flexure prediction
11. Ship inertial navigation error modeling
12. Missile inertial guidance error modeling
13. Geomagnetic core field modeling
14. Nonlinear model predictive control for fossil power plants
15. Subsurface ground water flow modeling and calibration
16. Geophysical modeling
17. Optical image landmarking using shoreline correlation
18. Spacecraft bi-propellant propulsion modeling
19. Spacecraft thruster calibration using in-orbit data
20. Global Positioning System constellation orbit/clock/monitor station tracking (filtering, smoothing, and maximum likelihood estimation process noise optimization)
21. Atomic clock calibration

The wide range of applications has included filters with as few as three states and as many as 1000. The ground water flow modeling software has been used on problems with more than 200,000 nodes (400,000 unknowns), although the solution technique does not compute an error covariance matrix. Some models were based on first-principles, some models were empirically derived, and others were both. Because of this wide range in applications and model types, it was necessary to develop software and methods that could easily adapt to different models, and could be configured to easily change the selection of states to be estimated. The need to execute large scale problems on computers that were 200 times slower than today's PCs made it necessary to use reduced-order models of systems and to implement efficient code. This forced the development of tools and methods for simplifying the dynamic models, and for determining which states are important to the estimation. It also encouraged development of software that could easily adapt to different models and to different selections of estimated states. The lessons learned from these experiences are the focus of this book.

This book starts with introductory material and gradually expands on advanced topics. The first chapter briefly describes the estimation problem and the history of optimal estimation from Gauss to Kalman. Notation used in the book is discussed. Other background material on matrix properties, probability theory, and stochastic processes appears in Appendices.

Chapter 2 discusses different types of dynamic models, their use as the basis of estimation, and methods for computing the state transition and process noise matrices. Use of first-principles models, reduced-order models, and models for dynamic effects that are poorly known are addressed. Chapter 3 describes several real-world problems that demonstrate various modeling principles and are used as examples in later chapters.

Chapter 4 derives least-squares estimation from several different points of view (weighted least squares, Bayesian least squares, minimum mean-squared error, minimum variance, maximum likelihood, maximum *a posteriori*) and discusses various implementations. Chapter 5 discusses least-squares solution techniques such as Cholesky decomposition of the normal equations, QR decomposition, Singular Value Decomposition, and iterative Krylov space methods. Also addressed is the theory of orthogonal transformations, solution uniqueness, observability, condition number, and the pseudo-inverse. Numerous examples demonstrate the issues and performance.

Chapter 6 discusses methods to evaluate the validity of least-squares solutions, error analysis, selection of model order, and regression analysis for parameter estimation. Chapter 7 addresses the important topics of least-squares estimation for nonlinear systems, constrained estimation, robust estimation, data editing, and measurement preprocessing. Real-world examples are given.

Chapter 8 presents the basics of Kalman filtering, and shows that it is based on fundamental concepts presented in earlier chapters. Discrete and continuous versions of the Kalman filter are derived, and extensions to handle data correlations and certain types of model errors are presented. Other topics include steady-state filtering, outlier editing, model divergence, and model validation. The relationship to the Wiener filter is also discussed.

Since real-world problems are frequently nonlinear, methods for nonlinear filtering are discussed in Chapter 9. Also discussed are smoothing (fixed point, fixed interval, and fixed lag), analysis of various modeling errors, design of reduced-order models, and measurement preprocessing.

Chapter 10 discusses numerical issues and shows how use of factorized (square root) estimation can minimize the growth of numerical errors. The factorized U-D and SRIF algorithms and their various implementations are discussed at length, and smoothers designed to work with the factored filters are presented. An example based on an inertial navigation system error model is used to compare properties of the covariance and factored filters. Usefulness of the square-root information filter (SRIF) data equation concept as a general approach to estimation is explained, and a hydrological flow problem with soft spatial continuity constraints on hydraulic conductivity demonstrates application of the data equation to two-dimensional and three-dimensional spatial problems.

Chapter 11 presents several advanced topics. It shows how properties of the filter innovations (one-step filter measurement residuals) allow model jump detection/estimation, and calculation of the log likelihood function. The log likelihood is useful when determining which of several models is best, and in determining the "best" values of model dynamic parameters and magnitudes of process and measurement noise. Adaptive filtering techniques such as jump detection and multiple-model approaches are discussed. Other topics include constrained estimation, robust estimation, and newer nonlinear filtering approaches (unscented and particle filters).

Chapter 12 discusses empirical model development in cases where it is not possible to develop a complete model from first principles. It may seem odd that this chapter appears at the end of the book, but the methods used for exploratory analysis of stochastic time series data depend on model concepts discussed in Chapter 2, and on least-squares techniques presented in Chapters 4 through 6. Use of spectral analysis for determining model dynamics and order is explained, and methods for computing parameters of autoregressive (AR) or autoregressive moving average (ARMA) models are presented. Accurate determination of the model order and states is discussed. The theory is presented at a level where the reader can understand the implications of assumptions and limitations of the methods. Applications of the theory for real-world examples are mentioned, and the performance of the different methods is demonstrated using data generated from a fourth-order autoregressive moving average with exogenous input (ARMAX) model.

Most of the advanced methods presented in this book have appeared in previous literature. Unfortunately some methods are rarely used in practice because of difficulty in implementing flexible general-purpose algorithms that can be applied to different problems. Chapter 13 presents a framework for doing this and also describes software for that purpose. The goal is to structure code so that alternative models can be easily compared and enhancements can easily be implemented. Many algorithms and high-level drivers are available as Fortran 90/95 code (compatible with Fortran 2003), downloadable from the web site ftp://ftp.wiley.com/public/sci_tech_med/least_squares/. Software used for many examples described in various chapters, and drivers for advanced algorithms are also included. The various algorithms are implemented in software that I have used successfully for many years. Where the original code was written in Fortran 77, it has been upgraded to Fortran 90/95 usage and standards. In some cases the library functions and subroutines are enhanced versions of codes written by others. In particular, some modules for factorized estimation are an enhancement of Estimation Subroutine Package or Library subroutines written by Dr. G. Bierman and associates. (The original software is still available on the web at sources listed in Chapter 13.) Other libraries and algorithms are used by some drivers. One particularly useful library is the Linear Algebra PACKage (LAPACK). Others packages implement the LSQR and NL2SOL algorithms.

There were several reasons for choosing Fortran 90/95/2003—these are very different languages than the Fortran 77 familiar to many. The Fortran 90 enhancements most relevant to estimation problems are the matrix/vector notation used in

MATLAB® (registered trademark of The Mathworks, Inc.), and inclusion of matrix/vector operators such as multiply, dot product, and element-by-element operations. Other important enhancements include dynamic array allocation, code modules (encapsulation), limited variable scope, argument usage validation, and flexible string handling. Fortran 90 compilers can validate usage of variables and catch many bugs that would have previously been caught at execution time. Fortran 95 extends the capabilities and Fortran 2003 adds many object oriented capabilities. Fortran 90/95/2003 is a much more suitable language than C or C++ for linear algebra applications. Use of MATLAB was also considered, but rejected because it does not easily integrate into some production applications, and many previous estimation books have included MATLAB code. The reader should have little difficulty in converting Fortran 90/95 code to MATLAB because vector/matrix syntax is similar. Use of global variables and module features in the supplied software was deliberately limited so that code could be more easily ported to MATLAB.

After reading this book you may get the impression that development of "good" estimation models and algorithms is less of a science than expected—in some respects it is an art. Creation of universal "rules" for developing models and implementing estimation algorithms is a desirable, but probably unreachable, goal. After four decades of trying to find such rules, I have come to the conclusion that the best one can hope for is a set of "guidelines" rather than rules. A highly respected colleague once announced that "after years of trying to find rules for characterizing the effects of model errors on the estimation," he came to the conclusion that there are no rules. Every time he discovered a possible rule, he eventually discovered an exception. Mis-modeling in the estimation will, of course, usually increase estimation errors, but it is also possible that model errors can partially offset the effects of a particular noise sequence, and hence reduce errors in serendipitous cases. To determine bounds on estimation errors due to particular types of model errors, it is usually necessary to simulate both the estimation algorithm and the true system. This can be done using Monte Carlo or covariance methods, both of which are time-consuming.

With regard to algorithm accuracy, numerical analysts have extensively studied the growth of truncation and round-off errors (due to finite computer word length) for various algorithms, and certain algorithms have been identified as more stable than others. Convergence properties of iterative algorithms (for nonlinear problems) have also been studied and some algorithms have been found to converge faster or more reliably than others for most problems. However, there are always exceptions. Sometimes the algorithm that works best on a simple problem does not work well on a large-scale problem. One fault of many estimation books is the tendency to present "simple" low-order examples, leaving the reader with the impression that the described behavior is a general characteristic. My suggestion is to implement as many "reasonable" algorithms as you can, and to then test them thoroughly on simulated and real data. For example, try to include a "factorized" estimation algorithm (U-D or SRIF) option when a covariance formulation is the prime algorithm because you never know when numerical errors will cause problems. Also try to implement models and algorithms in multiple ways so that results can be compared. If it is important to develop an estimation algorithm that meets requirements under all possible conditions, expect much hard work—there are few shortcuts.

You may be tempted to give up and turn over development to "experts" in the field. I do not want to discourage potential consulting business, but please do not get discouraged. The estimation field needs new bright engineers, scientists, and mathematicians. One goal of this book is to pass on the lessons learned over several decades to a new generation of practitioners.

To provide feedback on the book or associated software, or to report errors, please send email to bgibbs00@ieee.org.

ACKNOWLEDGMENTS

I greatly appreciate the support, suggestions, and review from others. Thomas Englar and David Uetrecht reviewed the early chapters and offered suggestions. William Levine reviewed several chapters, suggested many improvements (particularly in Appendix B), and provided good publishing advice. James Carr reviewed most chapters and made many helpful suggestions. David Porter reviewed material on maximum likelihood parameter estimation and multiple model approaches, provided additional background, and suggested a number of improvements. Dan Simon reviewed several chapters, made many good suggestions, and provided helpful feedback. Catherine Thornton reviewed sections on error analysis and factorized estimation, discussed historical background, and provided additional papers by herself and Gerald Bierman. Wallace Larimore reviewed Chapter 12 discussions of canonical variate analysis (CVA), and provided additional background and references. Mark Pittelkau also reviewed Chapter 12 and made many suggestions. Axel Wittmann provided a picture of Gauss, while Rudolf Kalman provided a picture of himself and further explained filter history. Finally, I greatly appreciate the support and tolerance of my wife Edwina during the long book-writing process.

BRUCE P. GIBBS

CHAPTER 1

INTRODUCTION

Applications of estimation theory were limited primarily to astronomy, geodesy, and regression analysis up to the first four decades of the twentieth century. However, during World War II and in the following decades, there was an explosive growth in the number and types of estimation applications. At least four reasons were responsible for this growth. First, development of the new radar, sonar, and communication technology greatly expanded the interest in signal processing theory. Second, development of digital computers provided a means to implement complex math-based algorithms. Third, the start of space exploration and associated expansion in military technology provided a critical need for estimation and control, and also increased interest in state-space approaches. Finally, papers by Kalman (1960, 1961), Kalman and Bucy (1961), and others provided practical algorithms that were sufficiently general to handle a wide variety of problems, and that could be easily implemented on digital computers.

Today applications of least-squares estimation and Kalman filtering techniques can be found everywhere. Nearly every branch of science or engineering uses estimation theory for some purpose. Space and military applications are numerous, and implementations are even found in common consumer products such as Global Positioning System (GPS) receivers and automotive electronics. In fact, the GPS system could not function properly without the Kalman filter. Internet searches for "least squares" produce millions of links, and searches for "Kalman filter" produce nearly a million at the time of this writing. Kalman filters are found in applications as diverse as process control, surveying, earthquake prediction, communications, economic modeling, groundwater flow and contaminant transport modeling, transportation planning, and biomedical research. Least-squares estimation and Kalman filtering can also be used as the basis for other analysis, such as error budgeting and risk assessment. Finally, the Kalman filter can be used as a unit Jacobian transformation that enables maximum likelihood system parameter identification.

With all this interest in estimation, it is hard to believe that a truly new material could be written on the subject. This book presents the theory, but sometimes limits detailed derivations. It emphasizes the various methods used to support

Advanced Kalman Filtering, Least-Squares and Modeling: A Practical Handbook, by Bruce P. Gibbs

batch and recursive estimation, practical approaches for implementing designs that meet requirements, and methods for evaluating performance. It focuses on model development, since it is generally the most difficult part of estimator design. Much of this material has been previously published in various papers and books, but it has not all been collected in a form that is particularly helpful to engineers, scientists, or mathematicians responsible for implementing practical algorithms.

Before presenting details, we start with a general explanation of the estimation problem and a brief history of estimation theory.

1.1 THE FORWARD AND INVERSE MODELING PROBLEM

Modeling of physical systems is often referred to as either *forward modeling* or *inverse modeling*. In forward modeling a set of known parameters and external inputs are used to model (predict) the measured output of a system. A forward model is one that can be used for simulation purposes. In inverse modeling (a term used by Gauss) a set of measured values are used to infer (estimate) the model states that best approximate the measured behavior of the true system. Hence "inverse modeling" is a good description of the estimation process.

Figure 1.1 shows a generic forward model: a set of j constant parameters $\mathbf{p}$, a deterministic time-varying set of l input variables $\mathbf{u}(\tau)$ defined over the time interval $t_0 \le \tau \le t$, and an unknown set of k random process inputs $\mathbf{q}(\tau)$—also defined over the time interval $t_0 \le \tau \le t$—are operated on by a linear or nonlinear operator $\mathbf{f}_t(\mathbf{p},\mathbf{u},\mathbf{q},t)$ to compute the set of n states $\mathbf{x}(t)$ at each measurement time t. (Bold lower case letters are used to represent vectors, for example, $\mathbf{p} = [p_1 \quad p_2 \quad \cdots \quad p_j]^T$. Bold upper case letters are used later to denote matrices. The subscript "t" on $\mathbf{f}$ denotes that it is a "truth" model.) The states included in vector $\mathbf{x}(t)$ are assumed to completely define the system at the given time. In control applications $\mathbf{u}(t)$ is often referred to as a *control* input, while in biological systems it is referred to as an *exogenous* input.

Noise-free measurements of the system output, $\mathbf{y}_t(t)$, are obtained from a linear or nonlinear transformation on the state $\mathbf{x}(t)$. Finally it is assumed that the actual

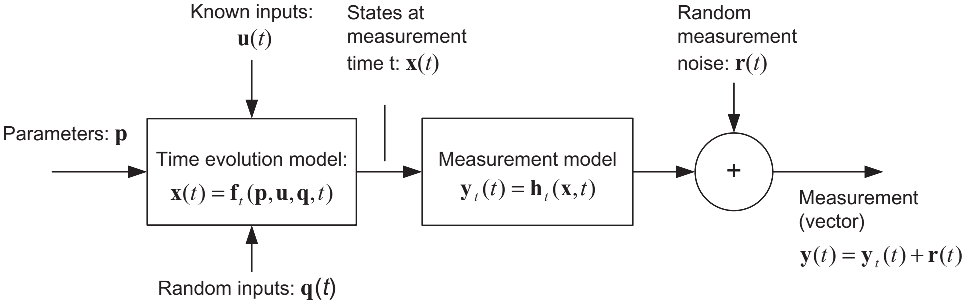


FIGURE 1.1: Generic forward model.

measurements are corrupted by additive random noise $\mathbf{r}(t)$, although measurement noise is often not considered part of the forward model.

A polynomial in time is a simple example of a forward model. For example, the linear position of an object might be modeled as $x(t) = p_1 + p_2 t + p_3 t^2$ where t represents the time difference from a fixed epoch. A sensor may record the position of the object as a function of time and random noise may corrupt the measurements, that is, $y(t) = x(t) + r(t)$. This is a one-dimensional example where neither process noise ($\mathbf{q}$) nor forcing inputs ($\mathbf{u}$) affect the measurements. The state will be multidimensional for most real-world problems.

The inputs $\mathbf{p}$, $\mathbf{u}$, $\mathbf{q}$, and $\mathbf{r}$ may not exist in all models. If the random inputs represented by $\mathbf{q}(t)$ are present, the model is called *stochastic*; otherwise it is *deterministic*. In some cases $\mathbf{q}(t)$ may not be considered part of the forward model since it is unknown to us. Although the model of Figure 1.1 is shown to be a function of time, some models are time-invariant or are a function of one, two, or three spatial dimensions. These special cases will be discussed in later chapters. It is generally assumed in this book that the problem is time-dependent.

Figure 1.2 graphically shows the inverse modeling problem for a deterministic model. We are given the time series (or possibly a spatially distributed set) of "noisy" measurements $\mathbf{y}(t)$, known system inputs $\mathbf{u}(t)$, and models (time evolution and measurement) of the system. These models, $\mathbf{f}_m(\mathbf{p},\mathbf{u},t)$ and $\mathbf{h}_m(\mathbf{x})$, are unlikely to exactly match the true system behavior (represented by $\mathbf{f}_t(\mathbf{p},\mathbf{u},t)$ and $\mathbf{h}_t(\mathbf{x})$), which are generally unknown to us. To perform the estimation, actual measurements $\mathbf{y}(t)$ are differenced with model-based predictions of the measurements $\mathbf{y}_m(t)$ to compute measurement residuals. The set of measurement residuals for the entire data span is processed by an optimization algorithm to compute a new set of parameter values that minimize some function of the measurement residuals. In least-squares estimation the "cost" or "loss" function to be minimized is the sum-of-squares, possibly weighted, of all residuals. Other optimization criteria will be discussed later. The new parameter values are passed to the time evolution and measurement models

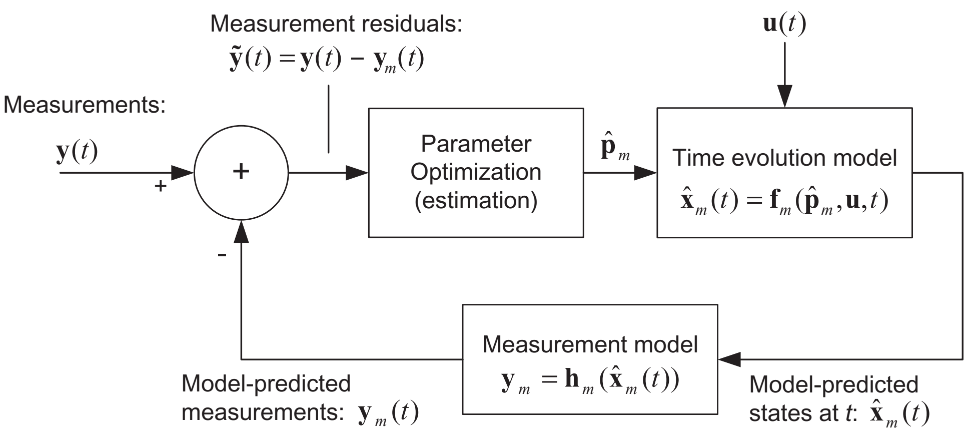


FIGURE 1.2: Deterministic inverse modeling.

to compute another set of model-based predicted measurements. A new set of measurement residuals is computed for the entire data span and a new cost function is computed. If the models are linear, only one iteration is normally required to converge on parameters that minimize the cost function. If the models are nonlinear, multiple iterations will be required to compute the optimum.

The optimization process may update the estimates of $\mathbf{p}$ after each new measurement is received. This is called recursive estimation and is particularly useful when process noise $\mathbf{q}(t)$ is present in the system. This topic is discussed in Chapter 8 (Kalman filtering) and later chapters.

This inverse modeling summary was intended as a high-level description of estimation. It intentionally avoided mathematical rigor so that readers unfamiliar with estimation theory could understand the concepts before being swamped with mathematics. Those readers with significant estimation experience should not be discouraged: the math will quickly follow.

1.2 A BRIEF HISTORY OF ESTIMATION

Estimation theory started with the least-squares method, and earliest applications modeled motion of the moon, planets, and comets. Work by Johannes Kepler (1619) established the geometric laws governing motion of heavenly bodies, and Sir Isaac Newton (1687) demonstrated that universal gravitation caused these bodies to move in conic sections. However, determination of orbits using astronomical observations required long spans of data and results were not as accurate as desired—particularly for comets. In the mid-1700s it was recognized that measurement errors and hypothetical assumptions about orbits were partly responsible for the problem. Carl Friedrich Gauss claims to have first used the least-squares technique in 1795, when he was only 18, but he did not initially consider it very important. Gauss achieved wide recognition in 1801 when his predicted return of the asteroid Ceres proved to be much more accurate than the predictions of others. Several astronomers urged him to publish the methods employed in these calculations, but Gauss felt that more development was needed. Furthermore, he had "other occupations." Although Gauss's notes on the Ceres calculations appear contradictory, he probably employed an early version of the least-squares method. Adrien-Marie Legendre independently invented the method—also for modeling planetary motion—and published the first description of the technique in a book printed in 1806. Gauss continued to refine the method, and in 1809 published a book (*Theoria Motus*) on orbit determination that included a detailed description of least squares. He mentioned Legendre's work, but also referred to his earlier work. A controversy between Gauss and Legendre ensued, but historians eventually found sufficient evidence to substantiate Gauss's claim as the first inventor.

Gauss's (1809) explanation of least squares showed great insight and may have been another reason that he is credited with the discovery. Gauss recognized that observation errors could significantly affect the solution, and he devised a method for determining the orbit to "satisfy all the observations in the most accurate manner possible." This was accomplished "by a suitable combination of more observations than the number absolutely requisite for the determination of the unknown quantities." He further recognized that "… the most probable system of

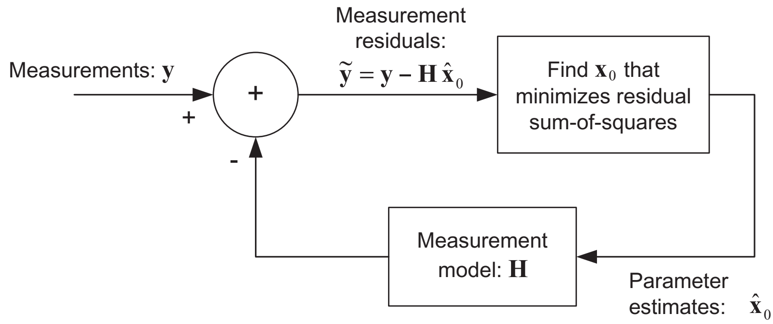


FIGURE 1.3: Least-squares solution to inverse modeling.

values of the quantities … will be that in which the sum of the squares of the differences between the actually observed and computed values multiplied by numbers that measure the degree of precision is a minimum."

Gauss's work may have been influenced by the work of others, but he was the first to put all the pieces together to develop a practical method. He recognized the need for redundancy of observations to eliminate the influence of measurement errors, and also recognized that determining the most probable values implied minimizing observation residuals. He computed these measurement residuals $\tilde{y}_1, \tilde{y}_2, \ldots \tilde{y}_m$ using equations (models) of planetary motion and measurements that were based on an "approximate knowledge of the orbit." This approach allowed iterative solution of nonlinear problems. Gauss reasoned that measurement errors would be independent of each other so that the joint probability density function of the measurement residuals would be equal to the product of individual density functions. Further he claimed that the normal density function would be

$$p_Y(\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_m) = \frac{\prod_{i=1}^{m} w_i}{(2\pi)^{m/2}} \exp\left[-\frac{1}{2}\sum_{i=1}^{m} \tilde{y}_i^2 w_i^2\right],$$

where w_i are weighting factors that take into account measurement errors. Gauss noted that the maximum of the probability density function is determined by maximizing the logarithm of the density function. For the assumed conditions, this is equivalent to minimizing the measurement residual sum-of-squares. Figure 1.3 shows the general structure of the least-squares method devised by Gauss, where vector **y** includes all measurements accumulated over a fixed period of time (if time is an independent variable of the problem).

Improvements in the least-squares computational approach, statistical and probability foundations, and extensions to other applications were made by Pierre-Simon Laplace, Andrey Markov, and Friedrich Helmert after 1809. In the early 1900s Sir Ronald Fisher (1918, 1925) developed the maximum likelihood and analysis of variance methods for parameter estimation. However, no fundamental extensions of estimation theory were made until the 1940s. Until that time applications generally involved parameter estimation using deterministic models. Andrey Kolmogorov and Norbert Wiener were concerned with modeling unknown stochastic signals corrupted by additive measurement noise. The presence of random dynamic noise made the stochastic problem significantly different from least-squares

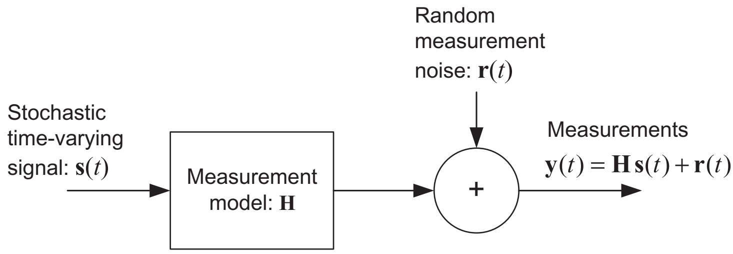


FIGURE 1.4: Simplified forward system model for Wiener filtering.

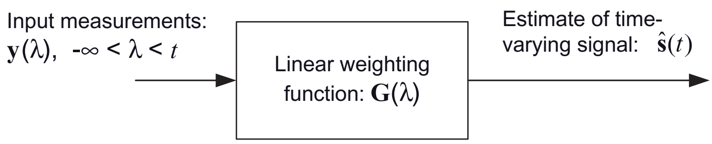


FIGURE 1.5: Wiener filter.

parameter estimation. Kolmogorov (1941) analyzed transition probabilities for a Markov process, and the discrete-time linear least-squares smoothing and prediction problem for stationary random processes. Wiener (1949) independently examined a similar prediction problem for continuous systems. Wiener's work was motivated by physical applications in a variety of fields, but the need to solve fire control and signal processing problems was a driving factor. This led him to study filtering and fixed-lag smoothing. Figure 1.4 shows a simplified version of the general problem addressed by Wiener filtering for multidimensional $\mathbf{y}(t)$ and $\mathbf{s}(t)$.

Message and measurement error processes are described by correlation functions or equivalently *power spectral densities* (PSD), and the minimum mean-squared error solution for the optimal weighting matrix $\mathbf{G}(t)$ is computed using the Wiener-Hopf integral equation in the time domain:

$$\mathbf{R}_{sy}(\tau) = \int_0^\infty \mathbf{G}(\lambda)\,\mathbf{R}_{ss}(\tau - \lambda)\,d\lambda \qquad 0 < \tau < \infty \tag{1.2-1}$$

where $\mathbf{R}_{sy}(\tau) = E[\mathbf{s}(t)\mathbf{y}^T(t-\tau)]$, $\mathbf{R}_{ss}(\tau) = E[\mathbf{s}(t)\mathbf{s}^T(t-\tau)]$, $E[\cdot]$ denotes the expected value of random variables and it is assumed that either $\mathbf{s}(t)$, $\mathbf{y}(t)$, or both are zero mean. The correlation functions $\mathbf{R}_{sy}(\tau)$ and $\mathbf{R}_{ss}(\tau)$ are empirically obtained from sampled data, or computed analytically if the signal characteristics are known. The steady-state filter gain, $\mathbf{G}(t)$, is computed by factorization of the power spectral density function—a frequency domain approach. (Details of the method are presented later in Chapter 8.) The resulting filter can be implemented as either *infinite impulse response* (IIR)—where a recursive implementation gives the filter an "infinite" memory to all past inputs—or as *finite impulse response* (FIR) where the filter operates on a sliding window of data, and data outside the window have no effect on the output. Figure 1.5 shows the IIR implementation. Unfortunately the spectral factorization approach assumes an infinite data span, so the solution is not realizable. Wiener avoided this problem by assuming a finite delay in the filter, where the delay was chosen sufficiently long to make approximation errors small. Other extensions of Wiener's work were made by Bode and Shannon (1950), and Zadeh and Ragazzini (1950). They both assumed that the dynamic system could be modeled as a shaping filter excited by white noise, which was a powerful modeling concept.

While Kolmogorov's and Wiener's works were a significant extension of estimation technology, there were few practical applications of the methods. The assumptions of stationary random processes and steady-state solutions limited the usefulness of the technique. Various people attempted to extend the theory to nonstationary random processes using time-domain methods. Interest in state-space descriptions, rather than covariance, changed the focus of research, and led to recursive least-squares designs that were closely related to the present Kalman filter. Motivated by satellite orbit determination problems, Peter Swerling (1959) developed a discrete-time filter that was essentially a Kalman filter for the special case of noise-free dynamics; that is, it still addressed deterministic rather than stochastic problems. Events having the greatest impact on technology occurred in 1960 and 1961 when Rudolf Kalman published one paper on discrete filtering (1960), another on continuous filtering (1961), and a joint paper (Kalman and Bucy 1961) on continuous filtering. The papers used the state-space approach, computed the solution as minimum mean-squared error, discussed the duality between control and estimation, discussed observability and controllability issues, and presented the material in a form that was easy to understand and implement. The design allowed for nonstationary stochastic signals and resulted in a solution with time-varying gains. The Kalman filter was quickly recognized as a very important tool. Stanley Schmidt realized that Kalman's approach could be extended to nonlinear problems. This led to its use for midcourse navigation on the National Aeronautics and Space Administration (NASA) Apollo moon program in 1961 (Schmidt 1981; McGee and Schmidt 1985).

Figure 1.6 shows the generic model structure used as the basis of the Kalman filter design: $\mathbf{q}(t)$ and $\mathbf{r}(t)$ are white random noise and $\mathbf{\Phi}(\Delta t)$ is the state transition matrix for the time interval Δt. Notice that the dynamic and measurement models

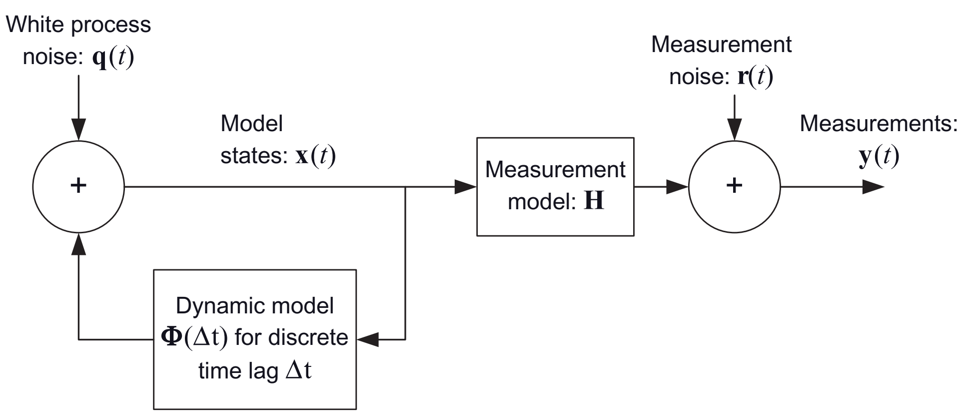


FIGURE 1.6: Forward model structure assumed by discrete Kalman filter.

are similar to those used in least-squares estimation, but they allow for the presence of random process noise, and the state vector **x** is defined using a moving epoch (t) rather than a fixed epoch (t_0).

More information on Gauss's work and subsequent estimation history can be found in Gauss (1809), Bühler (1981), Sorenson (1970), Meditch (1973), Anderson and Moore (1979), Åström (1980), Tapley et al. (2004), Simon (2006), Grewal and Andrews (2001), Kailath (1968), Bennett (1993), and Schmidt (1981).

1.3 FILTERING, SMOOTHING, AND PREDICTION

The estimation problem is further classified as either filtering, smoothing, or prediction. Figure 1.7 graphically shows the differences. In the filtering problem the goal is to continuously provide the "best" estimate of the system state at the time of the last measurement, shown as t_2. When a new measurement becomes available, the filter processes the measurement and provides an improved estimate of the state $\hat{\mathbf{x}}(t)$ at the new measurement time. In many systems—such as target tracking for collision avoidance or fire control—the goal is to provide an estimate of the state at some future time t_3. This is the prediction problem, and provided that the linear filter state estimate at t_2 is optimal, the predicted state is obtained by simply integrating the state vector at t_2:

$$\hat{\mathbf{x}}(t_3) = \hat{\mathbf{x}}(t_2) + \int_{t_2}^{t_3} \dot{\hat{\mathbf{x}}}(\lambda) d\lambda,$$

that is, the optimal linear predictor is the integral of the optimal linear filter. If the dynamic model is deterministic, the same approach can be applied to obtain the optimal state at any time prior to t_2. That is, integrate $\hat{\mathbf{x}}(t_2)$ backward in time from t_2 to t_1. This assumption is implicit when using batch least-squares estimation: the *epoch time* at which the state is defined is arbitrary as the state at any other time may be obtained by analytic or numerical integration. However, when the dynamic model is subject to random perturbations—the stochastic case handled by Wiener or Kalman filtering—the optimal smoothed estimate at times in the past cannot be obtained by simple integration. As you may guess, the optimal smoothed estimate is obtained by weighting measurements near the desired smoothed time t_1 more than those far from t_1. Smoothed estimates must be computed using a time history of filter estimates, or information that is equivalent. Optimal smoothing is discussed in Chapter 9.

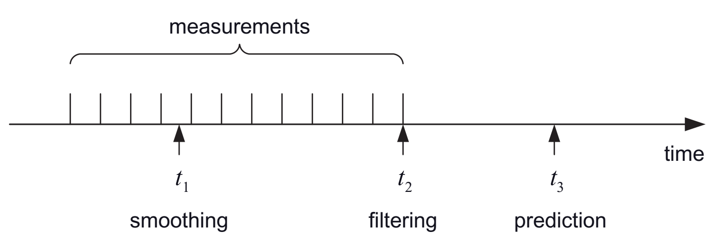


FIGURE 1.7: Filtering, smoothing, and prediction.