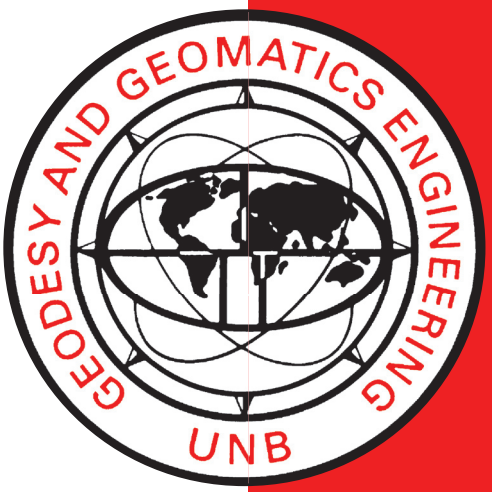
SOME ASPECTS OF KALMAN FILTERING

M. A. SALZMANN

August 1988



TECHNICAL REPORT
NO. 140

PREFACE

In order to make our extensive series of technical reports more readily available, we have scanned the old master copies and produced electronic versions in Portable Document Format. The quality of the images varies depending on the quality of the originals. The images have not been converted to searchable text.

SOME ASPECTS OF KALMAN FILTERING

Martin A. Salzmann

Department of Geodesy and Geomatics Engineering
University of New Brunswick
P.O. Box 4400
Fredericton, N.B.
Canada
E3B 5A3

August 1988
Latest Reprinting February 1996

ABSTRACT

In hydrography and surveying the use of kinematic positioning techniques is nowadays very common. An optimal estimate of position of the kinematic user is usually obtained by means of the Kalman filter algorithm. Dynamic and measurement models are established for a discrete time, time varying system. Some problems in establishing such a model are addressed. Based on this model and the derived Kalman filter several aspects of Kalman filtering that are important for kinematic positioning applications are discussed.

Computational and numerical considerations indicate that so-called covariance filters are to be used for kinematic positioning, and a specific covariance filter mechanization is described in detail. For some special applications linear smoothing techniques lead to considerably improved estimation results. Possible applications of smoothing techniques are reviewed. To guarantee optimal estimation results the analysis of the performance of Kalman filters is essential. Misspecifications in the filter model can be detected and diagnosed. The performance analysis is based on the innovation sequence.

Overall, this report presents a detailed analysis of some aspects of Kalman filtering.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

This technical report was written while the author was on leave from the Faculty of Geodesy, Delft University of Technology, Delft, The Netherlands. I would like to thank Dr. David Wells for the hospitality enjoyed during my stay at the Department of Surveying Engineering at the University of New Brunswick (UNB). Furthermore I would like to thank my fellow graduate students at UNB for the interest shown in my work.

Financial assistance for this work was provided by a Strategic Grant entitled "Marine Geodesy Applications" from the Natural Sciences and Engineering Research Council of Canada, held by David Wells. Assistance was also provided by Delft University of Technology.

1. INTRODUCTION

The past decades have shown a considerable increase in the number of applications where a real-time estimate of position is required for a user in a so-called kinematic mode. Especially in the offshore environment, the demand for precise position and velocity estimates for a kinematic user has been growing constantly. Kinematic means that the point to be positioned is actually moving. If one also takes into account the forces underlying this movement one generally speaks of dynamic positioning. Most applications of kinematic positioning are found in marine environments (e.g., hydrography, seismic surveys, navigation), but also in land surveying kinematic methods are increasingly put into use (e.g., inertial surveying, motorized levelling, real-time differential GPS). In this report we have no specific kinematic positioning application in mind. Actual applications are described in an accompanying report [Salzmann, 1988].

This report mainly deals with aspects of the estimation process most frequently used in kinematic and dynamic positioning, namely the **Kalman filter**. Kalman filters have been used successfully for years for positioning related problems, which is mainly due to their convenient recursive formulation which enables an efficient solution for time varying systems. The concepts and characteristics of Kalman filters have been discussed extensively since its original inception [Kalman, 1960]. The Kalman filter is covered in numerous textbooks (e.g., Jazwinski [1970], Gelb [1974], Anderson and Moore [1979], Maybeck [1979; 1982]). Generally the term filter is used for all estimation procedures in time varying systems. Actually filtering

encompasses the topics of prediction, where one predicts the state of a system at some future time; filtering (in the strict sense), where the state of a system is estimated using all information available at a certain time; and smoothing, where the state is estimated for some moment in the past. The so-called state of a system constitutes a vector of parameters which fully describes the system of interest (e.g., a moving vehicle).

In this report some specific aspects of Kalman filters considered relevant for kinematic positioning problems are discussed. For a general introduction and overall treatment of the estimation procedures for time varying sytems the reader is referred to the mentioned textbooks.

In Chapter 2 the discrete time linear Kalman filter and its underlying model are introduced. The Kalman filter algorithm is derived using a least-squares approach. Some comments on difficulties in establishing an actual filter model are made.

Chapter 3 is devoted to computational and numerical aspects of Kalman filtering. The concepts of covariance and inverse covariance (or information) filters are introduced. Specific implementation methods for the Kalman filter are considered. Also investigated is which specific method should be used for kinematic positioning problems.

A general overview of linear smoothing is given in Chapter 4. Smoothing algorithms are not extensively used in kinematic positioning (a smoothed estimate hampers real-time applications because of its inherent delay). If a small delay is acceptable, however, smoothing techniques lead to greatly improved estimates.

The performance analysis of Kalman filters is discussed in Chapter 5. It is very important that the filter operates at an optimum, because otherwise estimation results and all conclusions based on them are invalidated. For the performance analysis the so-called innovations approach is used.

Finally a summary of results is presented in Chapter 6.

2. THE LINEAR KALMAN FILTER

2.1 SYSTEM MODEL AND THE LINEAR KALMAN FILTER

In this chapter we introduce and briefly discuss the mathematical model and the relations of the linear discrete time Kalman filter. We are mainly interested in discrete time dynamic systems.

A discrete time dynamic system can be described by the following difference equation (called the dynamic model):

$$\underline{x}_k = \Phi(k,k-1)\underline{x}_{k-1} + B_{k-1}u_{k-1} + G_{k-1}\underline{w}_{k-1} \qquad (2.1)$$

where

k-1,k	time indices with k = 0,1,2,....
$\underline{x}_k$	n-dimensional vector of state variables; the state of a system is a vector of parameters with which the system can be fully described
$\Phi(k,k-1)$	n×n state transition matrix
u_{k-1}	p-dimensional vector of determinisitic control input
B_{k-1}	n×p control input matrix
$\underline{w}_{k-1}$	s-dimensional vector of system noise
G_{k-1}	n×s system noise input matrix.

An underscore indicates that a vector is a random variable. Note that the dimensions of the control input vector and the system noise vector are not necessarily equal to the dimension of the state vector.

Observations are available at discrete time intervals that are not necessarily equidistant. The observations are linearly related to the state by means of a design matrix and are corrupted by additive measurement noise. The so-called measurement model is given as:

$$\underline{y}_k = A_k\underline{x}_k + \underline{e}_k \tag{2.2}$$

where

$\underline{y}_k$	m-dimensional vector of observations
A_k	m×n design matrix
$\underline{e}_k$	m-dimensional vector of measurement noise.

Before we proceed, the statistical model underlying the system and measurement model is specified. The (s-dimensional) vector random system noise and the (m-dimensional) vector random measurement noise sequences are assumed to be zero mean, Gaussian, and uncorrelated. Hence:

$$E\{\underline{w}_i\} = 0$$
$$E\{\underline{w}_i,\underline{w}_j^t\} = Q_i\delta_{ij}$$
$$E\{\underline{e}_i\} = 0$$
$$E\{\underline{e}_i,\underline{e}_j^t\} = R_i\delta_{ij}$$
$$E\{\underline{e}_i,\underline{w}_j^t\} = 0$$

where $E\{.\}$ is the expectation operator, and δ_{ij} denotes the Kronecker delta (i.e., $\delta_{ij}=1$ if $i=j$, $\delta_{ij}=0$ otherwise).

The measurement and system noise sequences describe model disturbances and noise corruption that affect the system but also uncertainty about the model. Furthermore initial conditions (k=0) have to be specified. The initial state may assume a specific value, but because this value is generally not known a priori, the initial state is considered to be a random vector with a Gaussian distribution and the known statistics:

$$E\{\underline{x}_0\} = x_0$$
$$E\{(\underline{x}_0-x_0)\,(\underline{x}_0-x_0)^t\} = P_{0|0}\ .$$

Finally it is assumed that the system noise and the measurement noise random sequences are uncorrelated with the initial state.

$E\{(\underline{x}_0-x_0)\,\underline{w}_i^t\}$ for all i=0,1,2,.....
$E\{(\underline{x}_0-x_0)\,\underline{e}_i^t\}$ for all i=0,1,2,..... .

The transition matrix $\Phi(k,k-1)$, the design matrix A_k, the noise and control input matrices (G_{k-1} and B_{k-1} respectively), and the covariance matrices $P_{0|0}$, Q_k , and R_k are assumed to be known.

Depending on the application one might want to obtain an estimate of the state at a certain time. If the state is estimated for some future time, the process is called **prediction**. If the estimate is made using all measurements up to and including the current moment, one speaks of **filtering**. If an estimate is made for some time in the past using measurements until the current moment, the process is called **smoothing**.

In this chapter we limit ourselves to prediction and filtering. The Kalman filter process will now be introduced. It basically consists of two parts:

- **time update**; the prediction of the state vector and its (error) covariance using the system model and its statistics.
- **measurement update**; the improvement of the prediction (both the state and its (error) covariance) which gives the filtered state.

The time update is given as:

$$\hat{x}_{k|k-1} = \Phi(k,k-1)\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1} \tag{2.3a}$$

$$P_{k|k-1} = \Phi(k,k-1)P_{k-1|k-1}\Phi(k,k-1)^t + G_{k-1}Q_{k-1}G_{k-1}^t \tag{2.3b}$$

whilst the measurement update can be written in the following form

$$K_k = P_{k|k-1}A_k^t(A_kP_{k|k-1}A_k^t + R_k)^{-1} \tag{2.4a}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - A_k\hat{x}_{k|k-1}) \tag{2.4b}$$

$$P_{k|k} = P_{k|k-1} - K_kA_kP_{k|k-1} \tag{2.4c}$$

where K_k is the so-called Kalman gain matrix. The indices of the form i|j denote

estimates at time i based on all measurements till time j. The index k|k-1 thus indicates one step predicted values, whereas k|k denotes the estimate at time k using all measurements including y_k.

The filter model and the actual filter process are given in Fig. 2.1.

It is postulated that the Kalman filter is the best filter among the subset of all linear filters and the best filter among the set of all filters when the noise processes are Gaussian [Anderson and Moore, 1979].

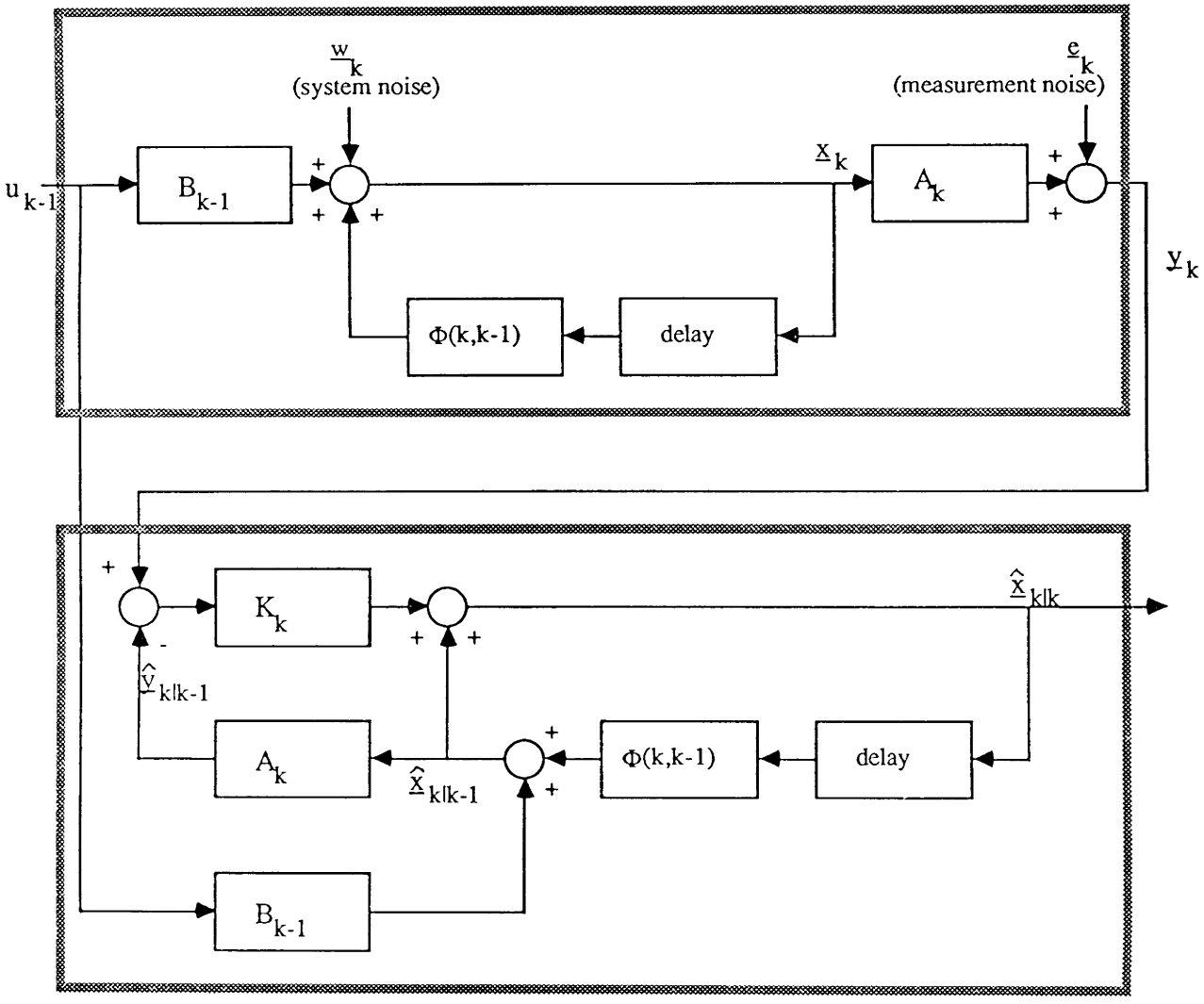


Fig. 2.1 Linear discrete time Kalman filter: filter model (top) and filter process

(bottom).

2.2 THE LINEAR KALMAN FILTER: A DERIVATION BASED ON LEAST SQUARES

There exist various derivations of the linear Kalman filter. These derivations are based on principles like least squares, minimum mean square error, maximum likelihood, and maximum a posteriori. In general the use of different principles leads to different estimators. However, in the case of linear systems where the probability density functions are assumed to be Gaussian all the above mentioned estimation methods yield the same estimator. Thus, the framework used to discuss such systems reduces to one of personal preference. Since the principle of least squares is probably the one which surveyors and hydrographers are the most familiar with, our derivation of the linear Kalman filter in this paragraph is based on this principle. The derivation is taken from Teunissen and Salzmann [1988]. Other derivatons of the filter process can be found in, e.g., Jazwinsky [1970], Gelb [1974], Anderson and Moore [1979], and Maybeck [1979].

The linear model of observation equations from which the linear Kalman filter can be derived is given as

$$E\left\{\begin{bmatrix} \hat{\underline{x}}_{k-1|k-1} \\ \underline{d}_k \\ \underline{y}_k \end{bmatrix}\right\} = \begin{bmatrix} I & 0 \\ -\Phi_{k,k-1} & I \\ 0 & A_k \end{bmatrix} \begin{bmatrix} x_{k-1} \\ x_k \end{bmatrix};$$

$$\begin{bmatrix} P_{k-1|k-1} & 0 & 0 \\ 0 & G_{k-1}Q_{k-1}G_{k-1}^{t} & 0 \\ 0 & 0 & R_k \end{bmatrix} \tag{2.5}$$

This model is equivalent to the general linear model of the adjustment with observation equations. Note that in this derivation a deterministic control input is not taken into account.

The Gaussian random vector $\hat{\underline{x}}_{k-1|k-1}$, with covariance matrix $P_{k-1|k-1}$, is the estimator of the state x_{k-1} at time k-1. It summarizes all the information available at time k-1 about state x_{k-1}. The Gaussian random vector $\underline{d}_k$, with covariance matrix Q_{k-1}, is the estimator of the difference between the state x_k and the propagated state $\Phi_{k|k-1}x_{k-1}$. If one would know the dynamic model perfectly, one would set both the mean $E\{\underline{d}_k\}$ and covariance matrix Q_{k-1} equal to zero. Due to all sorts of random disturbances, however, in practice one is usually not able to model the dynamics of the system completely. This is why the difference between the state and propagated state is modelled as a random vector.

The Gaussian random vector $\underline{y}_k$, with covariance matrix R_k, is an estimator of the observational variates at time k. Its mean is related to the state x_k through the design matrix A_k.

In order to estimate we need sample values. In practice we have only samples available for $\hat{\underline{x}}_{k-1|k-1}$ and $\underline{y}_k$. The sample of $\hat{\underline{x}}_{k-1|k-1}$ is given by the best estimate of x_{k-1} at time k-1, and the sample of $\underline{y}_k$ is given by the observations. There is, however, no sample available for $\underline{d}_k$. Since the difference between the state and propagated state is considered to be small, the random vector $\underline{d}_k$ is treated as a pseudo-observational variate for which the sample value can be taken equal to zero.

Prediction

The least-squares estimation of the state without the use of the observations $\underline{y}_k$ is considered first. Model (5) reduces then to

$$E\left\{\begin{bmatrix}\hat{\underline{x}}_{k-1|k-1}\\ \underline{d}_k\end{bmatrix}\right\} = \begin{bmatrix} I & 0\\ -\Phi_{k,k-1} & I\end{bmatrix}\begin{bmatrix}x_{k-1}\\ x_k\end{bmatrix}; \begin{bmatrix} P_{k-1|k-1} & 0\\ 0 & G_{k-1}Q_{k-1}G_{k-1}^t\end{bmatrix}. \tag{2.6}$$

Note that there is no redundancy since the model contains 2n equations with 2n unknowns. Thus the available estimate $\hat{x}_{k-1|k-1}$ of x_{k-1} cannot be improved upon. Due to the lack of redundancy in (2.6) the least-squares estimator of x_k, which we shall denote by $\hat{\underline{x}}_{k|k-1}$, simply follows from inverting the design matrix of (2.6). Thus

$$\hat{\underline{x}}_{k|k-1} = \Phi_{k,k-1}\hat{\underline{x}}_{k-1|k-1} + \underline{d}_k \,. \tag{2.7}$$

Application of the error propagation law gives for the covariance matrix of $\hat{\underline{x}}_{k|k-1}$:

$$P_{k|k-1} = \Phi_{k,k-1}P_{k-1|k-1}\Phi_{k,k-1}^t + G_{k-1}Q_{k-1}G_{k-1}^t \,. \tag{2.8}$$

Since the sample value of $\underline{d}_k$ is taken equal to zero it follows from (2.7) that the least-squares estimate of x_k based on model (2.6) is given by

$$\hat{x}_{k|k-1} = \Phi_{k,k-1}\hat{x}_{k-1|k-1} \,. \tag{2.9}$$

Equations (2.8) and (2.9) constitute the well known **time update equation** of the linear Kalman filter. They are equivalent to equations (2.3a) and (2.3b) in section 2.1.

Filtering

The least squares estimation of the state with the observations $\underline{y}_k$ included is now considered. In this case model (2.5) applies. Since there is redundancy (the redundancy equals the dimension of the vector of observations) the available estimate $\hat{x}_{k-1|k-1}$ of x_{k-1} can now be improved. This improvement is, however, part of smoothing (i.e., one uses the observations of time k to estimate the state at time k-1) and is not considered in the Kalman filter. The state x_{k-1} is therefore eliminated from model (2.5). This gives

$$E\left\{\begin{bmatrix}\Phi_{k,k-1}\hat{\underline{x}}_{k-1|k-1}+\underline{d}_k\\ \underline{y}_k\end{bmatrix}\right\} = \begin{bmatrix}I\\ A_k\end{bmatrix}x_k ; \begin{bmatrix}\Phi_{k,k-1}P_{k-1|k-1}\Phi^t_{k,k-1}+G_{k-1}Q_{k-1}G^t_{k-1} & 0\\ 0 & R_k\end{bmatrix}. \tag{2.10}$$

With (2.7) and (2.8) this can also be written as

$$E\left\{\begin{bmatrix}\hat{\underline{x}}_{k|k-1}\\ \underline{y}_k\end{bmatrix}\right\} = \begin{bmatrix}I\\ A_k\end{bmatrix}x_k ; \begin{bmatrix}P_{k|k-1} & 0\\ 0 & R_k\end{bmatrix}. \tag{2.11}$$

Straightforward application of the least-squares algorithm gives for the least-squares estimator of x_k, denoted by $\hat{\underline{x}}_{k|k}$:

$$\hat{\underline{x}}_{k|k} = (P^{-1}_{k|k-1} + A^t_k R^{-1}_k A_k)^{-1}(P^{-1}_{k|k-1}\hat{\underline{x}}_{k|k-1} + A^t_k R^{-1}_k \underline{y}_k) . \tag{2.12}$$

Application of the error propagation law gives for the covariance matrix of $\hat{\underline{x}}_{k|k}$:

$$P_{k|k} = (P^{-1}_{k|k-1} + A^t_k R^{-1}_k A_k)^{-1} . \tag{2.13}$$

Equations (2.12) and (2.13) constitute the so-called **measurement update equations** of the linear Kalman filter. An alternative form of the measurement update

equations can be obtained by invoking the following matrix inversion lemma:

$$(C^{-1}+B^tD^{-1}B)^{-1} = C - CB^t(D+BCB^t)^{-1}BC\ , \tag{2.14}$$

where C and D are symmetric matrices. The identity (2.14) is easily verified by multiplying the right-hand side of (2.14) with $[C^{-1}+B^tD^{-1}B]$. Application of the matrix inversion lemma (2.14) to (2.12) and (2.13) gives after some arrangements for the filtered state

$$\hat{\underline{x}}_{k|k} = \hat{\underline{x}}_{k|k-1} + P_{k|k-1}A_k^t(R_k+A_kP_{k|k-1}A_k^t)^{-1}(\underline{y}_k - A_k\hat{\underline{x}}_{k|k-1})\ , \tag{2.15}$$

and for its covariance matrix

$$P_{k|k} = P_{k|k-1} - P_{k|k-1}A_k^t(R_k+A_kP_{k|k-1}A_k^t)^{-1}A_kP_{k|k-1}\ . \tag{2.16}$$

The measurement update equations (2.15) and (2.16) are the ones which are usually presented in the literature. They are given as equations (2.4b) and (2.4c) in section 2.1.

2.3 EXTENSIONS OF THE SYSTEM MODEL

Now the Kalman filter has been established, a few remarks regarding the properties of the filter and possible extensions to the filter concept can be made.

2.3.1 Alternative System Models

The dynamic model on which the Kalman filter is based has been introduced as a discrete time linear dynamic system. Such a discrete time system can often be derived directly for the problem at hand. In many cases, however, such a system is based on a

continuous time dynamic system. Especially if one wants to describe the dynamics (e.g., forces) underlying the model, the system model can often be better represented as

$$\dot{\underline{x}}(t) = F(t)\underline{x}(t) + B(t)u(t) + G(t)\underline{w}(t) \tag{2.17}$$

which is a linear differential equation with one independent variable (in our case time t) and where

$$\dot{x}(t) = \frac{d}{dt}x(t) \ .$$

F(t) is the so-called n×n dynamics matrix. An initial condition $\underline{x}(t_0)=x_0$ is assumed. It is assumed that the measurement model is based on sampled measurements and thus the measurement model remains unchanged.

The Gaussian process $\underline{w}(t)$ is an m-dimensional Gaussian process of zero mean and strength Q(t).

$$E\{(\underline{w}(t))\} = 0$$
$$E\{\underline{w}(t)\,\underline{w}(s)^t\} = Q(t)\delta(t-s)$$

where Q(t) is a spectral density matrix [Gelb, 1974].

Given model (2.17) the time update equations can be solved in continuous time. Using numerical integration techniques a solution for the system state

$$\dot{\underline{x}}(t) = F(t)\underline{x}(t) + B(t)u(t) \tag{2.18}$$

can be obtained starting from the initial value x_0 and its covariance matrix can be computed. The update equation for the covariance is not derived (see e.g. Gelb [1974]) and is given here for easy reference:

$$\dot{P}(t) = F(t)P(t) + P(t)F(t)^t + G(t)Q(t)G(t)^t \tag{2.19}$$

(starting from the initial value P_0).

Although it might be advantageous to formulate the dynamic model in continuous time it is indicated how the discrete time equivalent of model (2.17) can be obtained. This enables us to perform the actual processing with the previously outlined discrete time algorithm.

The solution of the differential eqn. (2.17) is given by:

$$\underline{x}(t) = \Phi(t,t_0)x_0 + \int_{t_0}^{t} \Phi(t,\tau)B(\tau)u(\tau)d\tau + \int_{t_0}^{t} \Phi(t,\tau)G(\tau)\underline{w}(\tau)d\tau \quad . \tag{2.20}$$

Although the last term in eqn. (2.20) cannot be evaluated properly, this imprecise notation will be maintained (for a discussion see Maybeck [1979]). By inserting the proposed solution in eqn. (2.17) it can be seen that this is a valid solution. $\Phi(t,t_0)$ is the state transition matrix that satisfies the differential equation

$$\frac{d}{dt}(\Phi(t,t_0)) = F(t)\Phi(t,t_0)$$

and the initial condition

$$\Phi(t_0,t_0) = I \quad .$$

The transition matrix has the following property:

$$\Phi(t_2,t_0) = \Phi(t_2,t_1)\Phi(t_1,t_0)$$

and thus

$$\Phi(t,t_0)\Phi(t_0,t) = \Phi(t,t) = I$$

so that

$$\Phi(t,t_0)^{-1} = \Phi(t_0,t) \quad .$$

If the dynamics matrix F(t) is a constant matrix (i.e., F(t)=F for all t) the transition matrix $\Phi(t,t_0)$ is a function of the time difference $(t-t_0)$ only. The transition matrix can then be expressed as a matrix exponential

$$\Phi(t,t_0) = \Phi(t-t_0) = e^{F(t-t_0)}$$

or by the equivalent series expansion

$$\Phi(t-t_0) = \sum_{n=0}^{\infty} \frac{(t-t_0)^n F^n}{n!} \quad .$$

In the case of time invariant systems the transition matrix $\Phi(t-t_0)$ can also be obtained via the inverse Laplace transform

$$\Phi(t-t_0) = \pounds^{-1}(sI - F)^{-1} \quad .$$

The derivation via Laplace transforms can be very advantageous for analytical studies.

To summarize how the state transition matrix can be obtained is by numerical integration methods and in the time invariant case by a matrix series expansion or using Laplace transforms.

If one assumes that the control input u(t) is constant between two measurement updates it follows that

$$B_{k-1} = \int_{t_{k-1}}^{t_k} \Phi(t_k,\tau)B(\tau)d\tau \quad .$$

The term

$$\int_{t_{k-1}}^{t_k} \Phi(t_k,\tau)G(\tau)\underline{w}(\tau)d\tau$$

is equivalent to $G_{k-1}\underline{w}_{k-1}$ in eqn. (2.1) although it cannot be evaluated rigorously .

Now that the time update of the state has been linked to the discrete time model the state covariances are discussed. In continuous time

$$E\{(G(\tau)\underline{w}(\tau))(G(\sigma)\underline{w}(\sigma))^t\} = G(\tau)Q(\tau)G(\tau)^t\delta(\tau-\sigma) \quad .$$

The discrete time formulation can be derived as follows:

$$G_{k-1}Q_{k-1}G_{k-1}^t = E\left\{\int_{t_{k-1}}^{t_k}\int_{t_{k-1}}^{t_k}\Phi(t_k,\tau)G(\tau)\,\underline{w}(\tau)\underline{w}(\sigma)^tG(\sigma)^t\Phi(t_k,\sigma)^t d\tau d\sigma\right\}$$

which by taking the expectation operator into the integral signs and recalling the definition of the continuous time system noise can be written as:

$$G_{k-1}Q_{k-1}G_{k-1}^t = \int_{t_{k-1}}^{t_k}\Phi(t_k,\tau)G(\tau)\,Q(\tau)G(\tau)^t\Phi(t_k,\tau)^t d\tau \quad .$$

Note that Q_{k-1} is a covariance matrix. $G_{k-1}Q_{k-1}G_{k-1}{}^t$ is not necessarily positive definite, but always semi-positive definite. Furthermore it holds that

$$E\{(G_k\underline{w}_k)(G_l\underline{w}_l)^t\} = \begin{cases} G_kQ_kG_k^t & k=l \\ 0 \quad \text{otherwise} & \end{cases}$$

Starting from a continuous time dynamic system model it has been shown how the discrete time equivalent form can be obtained.

2.3.2 Model Nonlinearities

Until now we have assumed a linear dynamic model and a linear measurement model. In practice, however, often nonlinear models are encountered. Although the linear model will not provide a valid description anymore (i.e., the nonlinearities in the model are not negligible) we still want to exploit the linear estimation concepts derived earlier, and thus apply the developed linear estimation results.

In general both the dynamic model and the measurement model can be nonlinear. A continuous time nonlinear system can, e.g., be formulated as:

$$\dot{\underline{x}}(t) = f(\underline{x}(t),u(t),t) + G(t)\underline{w}(t) \qquad (2.21)$$

where f is a known n-vector of functions of three arguments. Note that the dynamic system noise ($\underline{w}(t)$) enters in a linear additive fashion. This does not generally hold true. The measurement model is still based on sampled measurements. The nonlinear measurement model is decribed by

$$\underline{y}_k = a(\underline{x}_k, t_k) + \underline{e}_k \tag{2.22}$$

with $\underline{e}_k$ as defined earlier.

Note that also in this case the noise enters in a linear additive fashion. In many applications in surveying and hydrography the system dynamic model can be modelled adequately as being linear. The measurement model, however, will usually be nonlinear. A method to deal with nonlinearities in the model is to use a first-order approximation. Then the measurement model can be expanded in a Taylor series (neglecting the higher-order terms):

$$\underline{y}_k = a(x_k^0, t_k) + A_k(\underline{x}_k - x_k^0) + \underline{e}_k$$

where

$$A_k \equiv \frac{\partial\, a(\underline{x}_k, t_k)}{\partial\, \underline{x}}\Bigg|_{\underline{x} = x_k^0} .$$

The design matrix obtained by linearization can then be used in the Kalman filter measurement update equations presented earlier.

One can linearize the measurement function about different points. If some nominal trajectory is available and linearization is done about points of this trajectory, the filter process is called a linearized Kalman filter (LKF). If the linearization is done about the predicted state estimate (i.e., $x_k^0 = \hat{x}_{k|k-1}$) the process is called an extended Kalman filter (EKF). If the solution is iterated (until a certain stopping criterion is met) using

the most recent state estimate (starting with the predicted state), the process is called an iterated extended Kalman filter (IEKF). See Fig. 2.2

In the case of an IEKF the measurement update equation for the system state can be written as:

$$\hat{x}_{k|k,(i+1)} = \hat{x}_{k|k-1} + K_{k,(i)}(y_k - a_k(\hat{x}_{k|k,(i)}) - A_{k,(i)}(\hat{x}_{k|k-1} - \hat{x}_{k|k,(i)}))$$

where i denotes the number of iterations.

Apart from this the IEKF equations are equivalent to those of the linear Kalman filter. Due to the iterative nature, the computational burden of the IEKF may be much larger than that of the linear Kalman filter. The IEKF is probably the most frequently used method in hydrography and surveying to deal with nonlinearities in the filter model.

2.3.3 Filter Design Considerations

If the Kalman filter model is linear, the Kalman filter process equations show that the propagation and update equations of the (error) covariance matrix of the system state and the Kalman gain matrix are independent of the actual observations. Thereby the opportunity is given to perform an a priori accuracy analysis of the Kalman filter performance

In case the filter model is time invariant the filter will operate in steady state conditions after some time. This facilitates the assessment of the filter accuracy for the whole period of interest by only analyzing a single steady state sample. Even in case the filter model is nonlinear, however, an a priori accuracy analysis is not impossible.

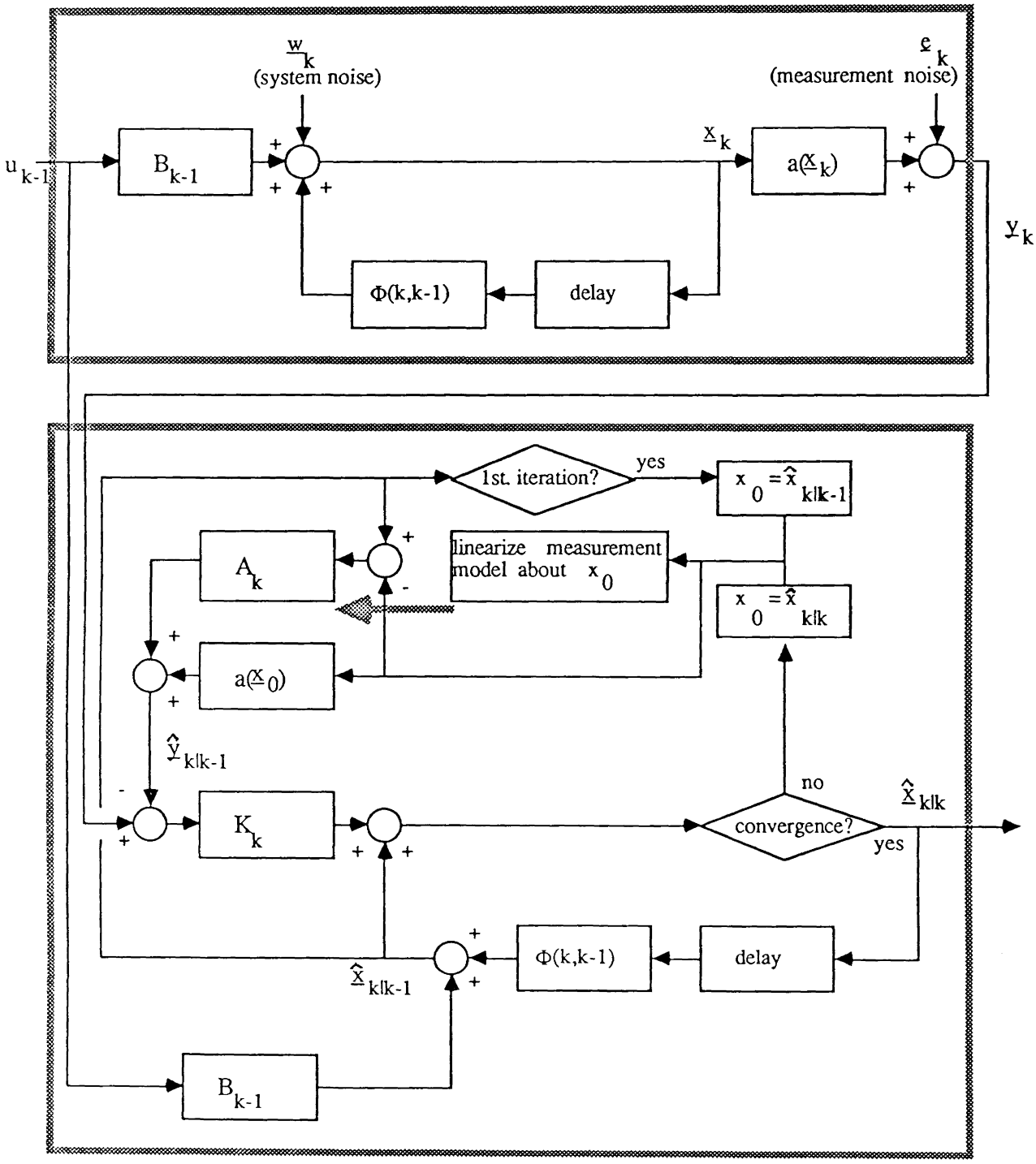


Fig. 2.2 Iterated Extended Kalman Filter for a dicrete time filter model with nonlinear measurement model.

Assuming a nominal trajectory the update and propagation equations for the (error) covariance can be performed without the actual measurements being available. The obtained results will, however, only be approximate.

2.3.4 Alternative Noise Models

The Kalman filter described in this chapter is based on the fact that the system noise and measurement noise sequences are white and mutually uncorrelated. These results can be extended by allowing correlation between both noise sequences. If one assumes a correlation between the measurement noise at time k and the dynamic system noise of the ensuing sample period the statistical model is expanded as:

$$E\{\underline{w}_k \underline{e}_j{}^t\} = S_k \delta_{kj} \ .$$

Allowing this correlation leaves the Kalman filter measurement update unchanged. The equations of the time update change however (see Maybeck [1979, p.246], Anderson and Moore [1979, p.108]).

If on the other hand one assumes correlation between the system noise over a sample period and the noise corrupting the measurement at the end of the sample interval, which can be stated as:

$$E\{\underline{w}_{k-1} \underline{e}_j{}^t\} = S_k \delta_{kj} \ ,$$

then the time update equations remain unchanged, but the equations of the measurement update are slightly different (see Maybeck [1979, p.247]).

Exploiting the correlation between system noise and measurement noise improves the estimation precision. Probabilistic information on the realization of the system noise is obtained by the observation of the particular realizations of the measurement

noise [Maybeck, 1979]. In practice it might be difficult, however, to specify a realistic S-matrix.

Another possible extension to the statistical model is the assumption that the noise sequences are not white. The noise models can then be developed further in state space models and thereby lead to state augmentation. State augmentation is the most generally applied technique in this case. State augmentation can increase the computational load of the Kalman filter process considerably.

Sometimes models for the measurement noise can be developed (e.g., the measurement noise sequence is a Markov process, that is, the realization of the noise process at certain time depends only on the realization of that process at the previous moment and a random input). This setup is equivalent to a measurement free problem and is not discussed here any further.

2.4 FINAL MODEL CONSIDERATIONS

In this chapter some aspects of the discrete time linear Kalman filter and its underlying model have been discussed. In the subsequent chapters some specific aspects of the Kalman filter will be investigated, which are basically unrelated to the exact structure of the filter model. Therefore a somewhat simplified model will be used for further investigations.

The simplified dynamic model and measurement model are of the following form:

$$\underline{x}_k = \Phi(k,k-1)\underline{x}_{k-1} + G_{k-1}\underline{w}_{k-1} \tag{2.23}$$

$$\underline{y}_k = A_k\underline{x}_k + \underline{e}_k \tag{2.24}$$

where all the terms are equivalent to those specified for the original model (eqn. (2.1) to (2.2)). Note that a deterministic control input is no longer part of the model. The

derived Kalman filter time and measurement update equations remain unchanged provided that the deterministic control input is deleted.

SOME ASPECTS OF KALMAN FILTERING

3. COMPUTATIONAL CONSIDERATIONS

3.1. INTRODUCTION

In the previous chapter the linear Kalman filter was introduced. Time update and measurement update equations were derived for the linear Kalman filter. There exist, however, a range of different algorithms, generally called mechanizations. In the sixties and early seventies limited computer core memory size and low computing speeds led to the development of various mechanizations. Theoretically all these mechanizations lead to identical results. It was found, however, that if single precision arithmetic was used for the implementation of certain filter mechanizations, numerical problems could arise during the filter computations. Therefore numerous investigations concerning these numerical problems were set up. These investigations were mainly related to aerospace problems. In this context numerical properties and the computational efficiency of almost all mechanizations have been investigated.

It is of interest up to what extent numerical problems will arise in kinematic positioning problems (and the closely related field of navigation) and which mechanization should be preferred. In this chapter we will review the two basic forms of Kalman filter mechanizations and we will investigate which mechanizations are best suited for our purposes.

This chapter only addresses the computational aspects directly related to the different filter mechanizations of the linear Kalman filter. If, as often occurs in practice, the design, transition, and covariance matrices are not known (e.g., in case of nonlinearities of the model) the computational burden of computing these matrices can

account for up to 90% of the total computational burden of the Kalman filter. As is it likely that these computations have to be performed for every mechanization, compuational aspects not directly related to a specific mechanization are not considered here.

In section 3.2. the concepts of the covariance and inverse covariance (or information) filters are introduced. Section 3.3. is devoted to square root filtering, where the square root of the covariance matrix of the state (rather than the covariance matrix itself) is propagated in time. In section 3.4 a closer look is taken at the U-D covariance factorization method which is closely related to the square root concept. Finally some implementation considerations are discussed in section 3.5.

3.2 BASIC FILTER MECHANIZATIONS

Filter mechanizations can be devided into two major classes. The **covariance filters** and the **inverse covariance filters**. Covariance filters propagate the covariance matrix of the state in time. Inverse covariance filters propagate the inverse of the covariance matrix. Inverse covariance filters are usually called **information filters**.

The covariance-type (standard) Kalman filter equations have been derived in Chapter 2. The time and measurement update equations of the linear Kalman (covariance) filter are repeated here for easy reference. The time update equations for the estimate of the state and its covariance are given as:

$$\hat{x}_{k|k-1} = \Phi(k,k-1)\hat{x}_{k-1|k-1} \tag{3.1a}$$

$$P_{k|k-1} = \Phi(k,k-1)P_{k-1|k-1}\Phi(k,k-1)^t + G_{k-1}Q_{k-1}G_{k-1}^t \quad . \tag{3.1b}$$

The measurement update equations are of the following form:

$$K_k = P_{k|k-1}A_k^t(A_kP_{k|k-1}A_k^t + R_k)^{-1} \tag{3.2a}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - A_k\hat{x}_{k|k-1}) \tag{3.2b}$$

$$P_{k|k} = P_{k|k-1} - K_kA_kP_{k|k-1} \quad . \tag{3.2c}$$

An alternative version of the linear Kalman (covariance) filter is derived in Krakiwsky [1975] and is called the **Bayes filter**. In the Bayes filter eqns. (3.2a) and (3.2c) are replaced by

$$K_k = P_{k|k}A_k^tR_k^{-1} \tag{3.3a}$$

and

$$P_{k|k} = (P_{k|k-1}^{-1} + A_k^tR_k^{-1}A_k)^{-1} \tag{3.3b}$$

respectively. It can be seen that use of the Bayes filter can be advantageous with respect to the standard Kalman filter if the number of observations is larger than than the dimension of the state vector. The Bayes filter will not be pursued here any further.

The information filter is algebraically equivalent to the linear Kalman (covariance) filter. It has, however, some unique characteristics. Because the inverse covariance is propagated in time, the information filter can be started with no information available on the initial state. The information filter can be found in, e.g., Maybeck [1979]. In order to generate a valid startup procedure for the case in which the initial inverse covariance matrix is singular, somewhat modified state variables are defined.

$$\hat{z}_{k|k-1} = P_{k|k-1}^{-1}\hat{x}_{k|k-1} \tag{3.4a}$$

$$\hat{z}_{k|k} = P_{k|k}^{-1}\hat{x}_{k|k} \quad . \tag{3.4b}$$

With

$$M_k = \Phi(k-1,k)^t P^{-1}_{k-1|k-1}\Phi(k-1,k) \tag{3.5}$$

the time update equations of the information filter are given as:

$$\hat{z}_{k|k-1} = (I - M_k G_{k-1}(G^t_{k-1}M_k G_{k-1} + Q^{-1}_{k-1})^{-1} G^t_{k-1})\Phi(k-1,k)^t \hat{z}_{k-1|k-1} \tag{3.6a}$$

$$P^{-1}_{k|k-1} = M_k - M_k G_{k-1}(G^t_{k-1}M_k G_{k-1} + Q^{-1}_{k-1})^{-1} G^t_{k-1} M_k \quad . \tag{3.6b}$$

The measurement update equations of the information filter are:

$$\hat{z}_{k|k} = \hat{z}_{k|k-1} + A^t_k R^{-1}_k y_k \tag{3.7a}$$

$$P^{-1}_{k|k} = P^{-1}_{k|k-1} + A^t_k R^{-1}_k A_k \quad . \tag{3.7b}$$

It can be seen from eqns. (3.7a) and (3.7b) that the information filter is more efficient in computing the measurement update than the covariance filter. On the other hand the time update equations of the information filter are more complex than those of the covariance filter. The inverses that have to be computed for the covariance filter recursions basically depend on the dimension of the observation process, whereas for the information filter they depend primarily on the dimension of the state vector.

The advantages of covariance filters can be summarized as follows:

- Continuous estimates of state variables and their covariances are available at no extra computational cost.
- Covariance type filters appear to be more flexible and are easier to modify to perform sensitivity and error analysis [Bierman, 1973a].

The advantages of inverse covariance or information filters are:

- Large batches of data (i.e. m>>n) are processed very efficiently from a computational point of view.

- No information concerning the initial state is required to start the information filter process (i.e., the inverse covariance matrix at the start of the information filter process may be zero).

In general covariance type filters will computationally be more efficient than information type filters if frequent estimates are required.

In cycling through the filter equations the covariance matrix (or its inverse) of the state can result in a matrix which fails to be nonnegative positive. The measurement update of the covariance filter can be rather troublesome numerically. Equation (3.2c) can involve small differences of large numbers, particularly if at least some of the measurements are very accurate. It has been shown [Bierman, 1977] that on finite wordlength computers this can cause numerical precision problems. Therefore an equivalent form of (3.2c), called the Joseph-form, is often used:

$$P_{k|k} = (I - A_k K_k) P_{k|k-1} (I - A_k K_k)^t + K_k R_k K_k^t \tag{3.8}$$

Apart from better assuring the symmetry and positive definiteness of $P_{k|k}$, the Joseph-form is also insensitive, to first order, to small errors in the computed filter gain [Maybeck, 1979]. However, the Joseph-form requires a considerably greater number of computations than (3.2c). In the literature the Joseph-form is generally called the stabilized Kalman filter.

For the information filter a stabilized version of the covariance time update equation (which is anagolous in form to the covariance measurement update of the covariance filter) exists. This analogue of the Joseph-form for the information filter is given in [Maybeck, 1979]:

$$P_{k|k-1}^{-1} = (I - X_k G_{k-1}^t) M_k (I - X_k G_{k-1}^t)^t + X_{k-1} Q_{k-1}^{-1} X_{k-1}^t \tag{3.9}$$

where

$$X_{k-1} = M_k G_{k-1}(G_{k-1}^t M_k G_{k-1} + Q_{k-1})^{-1} \quad .$$

3.3 SQUARE ROOT FILTERING

Because the stabilized filter mechanizations required to much storage and computations for early Kalman filter applications, an alternative strategy to cope with the numerical problems encountered in computing the (error) covariance matrix was developed. The limited computer capabilities forced the practitioners to use single precision arithmetic for their computations, while at the same time numerical accuracy had to be warranted. It was soon realized that nonnegative definiteness of the covariance matrix could also be retained by propagating this matrix in a so-called square root form. If M is a nonnegative definte matrix, N is called a square root of M if $M = NN^t$. The matrix N is normally square, not necessarily nonnegative definite, and not unique. The matrix N can be recognized as a Cholesky factor, but the common name "square root" will be maintained.

Let $S_{k|k}$ and $S_{k|k-1}$ be square roots of $P_{k|k}$ and $P_{k|k-1}$ respectively. The product $P=SS^t$ is always non-negative definite and thus the square root technique avoids negative definite error covariance matrices.

An overview of different square root filters is given in Chin [1983]. We will briefly discuss the square root forms of the covariance and information filters. The presentation of the square root filters is patterned after Anderson and Moore [1979].

3.3.1 Square Root Covariance Filter

The time update equations of the square root covariance filter can be summarized as:

$$\hat{x}_{k|k-1} = \Phi(k,k-1)\hat{x}_{k-1|k-1} \tag{3.10a}$$

$$\begin{bmatrix} S^t_{k|k-1} \\ 0 \end{bmatrix} = T_1 \begin{bmatrix} S^t_{k-1|k-1}\Phi(k,k-1)^t \\ (Q^{1/2}_{k-1})^t G^t_{k-1} \end{bmatrix} . \tag{3.10b}$$

In general the matrix T_1 can be any orthogonal matrix (i.e. $T_1T_1^t = T_1^tT_1 = I$) making $S_{k|k-1}{}^t$ upper triangular. In square root implementations the square root S is chosen to be the Cholesky factor of P. The measurement update equations of the square root covariance filter can be represented as:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \bar{K}_k(A_kP_{k|k-1}A^t_k + R_k)^{-1/2}(y_k - A_k\hat{x}_{k|k-1}) \tag{3.11a}$$

$$\begin{bmatrix} ((A_kP_{k|k-1}A^t_k + R_k)^{1/2})^t & \bar{K}^t_k \\ 0 & S^t_{k|k} \end{bmatrix} = T_2 \begin{bmatrix} (R^{1/2}_k)^t & 0 \\ S^t_{k|k-1}A^t_k & S^t_{k|k-1} \end{bmatrix} \tag{3.11b}$$

with T_2 orthogonal.

We will not dwell on the problems concerning the construction of the orthogonal matrices T_1 and T_2. Methods suggested in the literature ([Kaminski et al., 1971; Bierman, 1977; Thornton and Bierman, 1980]) are closely related to well known stable orthogonalization methods as the Householder and Givens transformations or the modified Gram-Schmidt orthogonalization scheme. It is due to these numerically stable orthogonalization methods that square root filters show improved numerical

stability. Square root filters show better numerical behaviour in computing covariance matrices than the standard Kalman filter. As far as error analysis is concerned this cannot be claimed for the gain matrix (K) or the estimate itself ($\hat{x}$) (see LeMay [1984] and Verhaegen and van Dooren [1986]). For a more extensive treatment of square root covariance filters the reader is referred to Anderson and Moore [1979] and Maybeck [1979].

3.3.2 Square Root Information Filter

The square root information filter (SRIF) is presented in an analogous fashion to the square root covariance filter. For the square root information filter again a somewhat modified state vector is defined:

$$\hat{z}'_{k|k-1} = S^{-1}_{k|k-1}\hat{x}_{k|k-1} \tag{3.12a}$$

$$\hat{z}'_{k|k} = S^{-1}_{k|k}\hat{x}_{k|k} \quad . \tag{3.12b}$$

The measurement update equations of the SRIF are given as:

$$\begin{bmatrix} S^{-1}_{k|k} \\ 0 \end{bmatrix} = T_3 \begin{bmatrix} S^{-1}_{k|k-1} \\ R_k^{-1/2}A_k \end{bmatrix} \tag{3.13a}$$

$$\begin{bmatrix} \hat{z}'_{k|k} \\ * \end{bmatrix} = T_3 \begin{bmatrix} \hat{z}'_{k|k-1} \\ R_k^{-1/2}y_k \end{bmatrix} \tag{3.13b}$$

where the lower left part of the left hand side ($*$) is of no interest.

One has to find an orthogonal matrix T_3 such that the right hand side of (3.13a) is upper triangular.

The time update equations of the SRIF can be derived from:

$$\begin{bmatrix} ((Q_{k-1}^{-1} + G_{k-1}^{t} M_k G_{k-1})^{1/2})^{t} & \bar{B}_{k-1}^{t} \\ 0 & S_{k|k-1}^{-1} \end{bmatrix} =$$

$$T_4 \begin{bmatrix} (Q_{k-1}^{1/2})^{-1} & 0 \\ S_{k-1|k-1}^{-1} \Phi(k-1,k) G_{k-1} & S_{k-1|k-1}^{-1} \Phi(k-1,k) \end{bmatrix} \qquad (3.14a)$$

with M_k as defined in (3.5).

Once again the general idea is to find an orthogonal matrix (T_4) such that the right-hand side of (3.14a) is upper triangular and with this T_4 one finds:

$$\begin{bmatrix} * \\ \hat{z}'_{k|k-1} \end{bmatrix} = T_4 \begin{bmatrix} 0 \\ \hat{z}'_{k-1|k-1} \end{bmatrix} . \qquad (3.14b)$$

The square root information filter (SRIF) is covered extensively in Bierman [1977]. A large class of square root mechanizations for both covariance and information filters has been developed. These are not included here. For an overview the reader is referred to Chin [1983].

3.4 U-D COVARIANCE FACTORIZATION FILTER

A different approach to square root covariance filters is the so-called **U-D covariance factorization filter** developed by Bierman [1977]. The covariance matrix is not decomposed into square root factors, but in the form UDU^t, where U is a unitary upper traingular (i.e., with ones along the diagonal) and D is a diagonal matrix. The U-D covariance factorization filter (or U-D filter) is basically an alternative

approach to the classical square root filters (see, e.g., Kaminski et al. [1971]). The factors U and D are propagated in time. For the U-D factorization the covariance matrices of the predicted and filtered state are factored as:

$$P_{k|k-1} = U_{k|k-1} P_{k|k-1} U_{k|k-1}^t \tag{3.15a}$$

$$P_{k|k} = U_{k|k} P_{k|k} U_{k|k}^t \quad . \tag{3.15b}$$

The close relationship of the U-D filter with square root filters is apparent, because $UD^{1/2}$ corresponds directly to a covariance square root. The main advantage of the U-D filter algorithm over the conventional square root filters is that no explicit square root computations are required. At the same time the U-D factorization shares the favourable numerical characteristics of the square root methods. If a matrix is positive (semi) definite a U-D factor can always be generated. The algorithm of the U-D factorization is closely related to the backward running Cholesky decomposition algorithm and is given in Appendix I.

3.4.1 U-D Filter Measurement Update

The U-D filter measurement updates are performed component wise and hence scalar measurement updates are used. If more than one observation per update is available, the measurements are processed sequentially. If the covariance matrix of the observations (R_k) is originally not diagonal, the measurement variables have to be transformed first in order to be able to apply this algorithm. If the covariance matrix of the observations is non-diagonal the Cholesky decomposition of R_k into a lower triangular matrix (i.e., $R_k = L_k L_k^t$) is computed first. Then the measurement model

$$\underline{y}_k = A_k \underline{x}_k + \underline{e}_k$$

is converted to

$$\underline{y}_k^* = A_k^* \underline{x}_k + \underline{e}_k^*$$

where

$$L_k \underline{y}_k^* = \underline{y}_k$$

$$L_k A_k^* = A_k$$

$$L_k \underline{e}_k^* = \underline{e}_k \quad .$$

It then follows that

$$E\{\underline{e}_k^* \underline{e}_k^{*t}\} = I \quad .$$

After this transformation of variables the U-D filter measurement update can be used. Starting from the linear Kalman (covariance) filter measurement update equations of the covariance we find for a scalar measurement update:

$$P_{k|k} = P_{k|k-1} - (P_{k|k-1} A_k^t)(1/\alpha)(A_k P_{k|k-1}) \tag{3.16}$$

where

$$\alpha = A_k P_{k|k-1} A_k^t + R_k \quad .$$

This form can be factored as (given the U-D factor of $P_{k|k-1}$):

$$U_{k|k} D_{k|k} U_{k|k}^t = U_{k|k-1} D_{k|k-1} U_{k|k-1}^t - (1/\alpha)(U_{k|k-1} D_{k|k-1} U_{k|k-1}^t A_k^t)(A_k U_{k|k-1} D_{k|k-1} U_{k|k-1}^t)$$

$$U_{k|k} D_{k|k} U_{k|k}^t = U_{k|k-1}(D_{k|k-1} - (1/\alpha)(D_{k|k-1} U_{k|k-1}^t A_k^t)(D_{k|k-1} U_{k|k-1}^t A_k^t)^t) U_{k|k-1}^t \quad . \tag{3.17}$$

Defining the vectors f and v (both of dimension n) as

$$f = U_{k|k-1}^t A_k^t$$

$$v = D_{k|k-1} f$$

and substituting these in (3.17) yields:

$$U_{k|k}D_{k|k}U_{k|k}^t = U_{k|k-1}(D_{k|k-1} - (1/\alpha)\, vv^t)U_{k|k-1}^t \quad . \tag{3.18}$$

The part in parentheses in (3.18) is positive (semi) definite and can therefore be factored as UDU^t. Furthermore the product of two unitary upper triangular matrices is again unitary upper triangular so that (3.18) can be written as:

$$U_{k|k}D_{k|k}U_{k|k}^t = (U_{k|k-1}\overline{U})\overline{D}(U_{k|k-1}\overline{U})^t \tag{3.19}$$

where

$$U_{k|k} = U_{k|k-1}\overline{U}$$

$$D_{k|k} = \overline{D} \quad .$$

It can be seen that the construction of the updated U-D factors depends on the simple factorization

$$\overline{UDU}^t = D_{k|k-1} - (1/\alpha)vv^t \quad . \tag{3.20}$$

The U-D factors can be generated recursively [Bierman, 1977]. In practical implementations of the measurement update of the U-D filter the Kalman gain matrix is not computed explicitly, but if desired it can be computed at very little extra computational cost. An algorithm for computing the U-D filter measurement update is given in Appendix I.

3.4.2 U-D Filter Time Update

For the time update of the U-D filters two methods are in use. The most trivial one is to "square up" the U-D factor to obtain the time propagated error covariance:

$$P_{k|k-1} = (\Phi(k,k-1)U_{k-1|k-1})D_{k-1|k-1}(\Phi(k,k-1)U_{k-1|k-1})^t + G_{k-1}Q_{k-1}G_{k-1}^t \quad . \tag{3.21}$$

The matrix $P_{k|k-1}$ can then be factored into $U_{k|k-1}$ and $D_{k|k-1}$ using the U-D factorization algorithm. This procedure is thought to be a stable process. Numerical difficulties can arise, however, if $\Phi(k,k-1)$ is large or $P_{k|k-1}$ is ill conditioned [Thornton and Bierman, 1980]. An advantage of the above method is that the covariance matrix of the (predicted) state is readily available.

The second approach, of which the development was motivated by numerical considerations, and for which the U-D factor is updated directly is based on a generalized Gram-Schmidt orthogonalization method. For square root filters it was proven that the square root of the covariance could be updated directly using an orthogonal transformation. Thornton was the first to apply this method to the U-D factor time update [Bierman, 1977]. This orthogonalization approach that yields $U_{k|k-1}$ and $D_{k|k-1}$ directly is briefly discussed. The following matrices are defined:

$$W = \begin{bmatrix} \Phi(k,k-1)U_{k-1|k-1} & G_{k-1} \end{bmatrix} \tag{3.22a}$$

$$\bar{D} = \begin{bmatrix} D_{k-1|k-1} & 0 \\ 0 & Q_{k-1} \end{bmatrix} . \tag{3.22b}$$

W is a (nx(n+s)) matrix and $\bar{D}$ a diagonal matrix of dimension (n+s) (recall that in Chapter 2, G_{k-1} was defined as a n×s system noise input matrix). It can be seen that the form WDW^t satisfies relation (3.21). If Q_{k-1} is originally not diagonal it must be factored first as $U_{q-1}D_{q-1}U_{q-1}^t$. G_{k-1} and Q_{k-1} are then replaced by $G_{k-1}U_{q-1}$ and D_{q-1} respectively so that $G_{k-1}Q_{k-1}G_{k-1}^t = (G_{k-1}U_{q-1})D_{q-1}(G_{k-1}U_{q-1})^t$.

The procedure to transform $W\bar{D}W^t$ in the form UDU^t is derived. We will show that use of the Gram-Schmidt orthogonalization method yields the desired result. With w_i the *i* th row (with dimension n+s) of W, this matrix can be written as:

$$W = \begin{bmatrix} w_1^t \\ . \\ . \\ w_n^t \end{bmatrix} . \tag{3.23}$$

An orthogonal basis of vectors $\{v_1,....,v_n\}$ is constructed applying the Weighted Gram-Schmidt (WGS) orthogonalization method to the rows of the matrix W

$$\begin{aligned} v_n &= w_n \\ v_j &= w_j - \sum_{k=j+1}^{n} \frac{(w_j^t \bar{D} v_k)}{(v_k^t \bar{D} v_k)} v_k \ , \ j=n-1,...,1 \ . \end{aligned} \tag{3.24}$$

The algorithm defined here is given in a backward recursive form, because the result is needed to construct an upper triangular matrix factorization. We can now define an orthogonal matrix T:

$$T = (v_1,....,v_n,v_{n+1},....,v_{n+s}) \ . \tag{3.25}$$

The vectors $v_1,....,v_n$ are computed using the WGS procedure. The remaining s columns of T are additional orthogonal basis vectors (of dimension n+s) which, however, do not have to be computed explicitly. We can write the matrix product of the matrices W and T as:

$$WT = \begin{bmatrix} w_1^t v_1 & . & . & w_1^t v_{n+s} \\ w_2^t v_1 & . & . & . \\ . & . & . & . \\ w_n^t v_1 & . & . & w_n^t v_{n+s} \end{bmatrix} . \tag{3.26}$$

Matrix W has rank n. Because the basis $\{v_1,...,v_n\}$ spans its range, and the basis vectors $v_{n+1},....,v_{n+s}$ are orthogonal to this spanning set, it follows that the last s columns of (3.26) are zero. The orthogonal basis vectors $v_1,....,v_n$ are computed in a backward recursive way and thus

$$w_j^t v_k = 0 \ , \ j > k \ .$$

Therefore (3.26) can be written as

$$WT = \begin{bmatrix} w_1^t v_1 & w_1^t v_2 & . & w_1^t v_n & 0 & . & 0 \\ 0 & w_2^t v_2 & . & . & . & . & . \\ 0 & 0 & . & . & . & . & . \\ 0 & 0 & 0 & w_n^t v_n & 0 & . & 0 \end{bmatrix} . \qquad (3.27)$$

The upper left (n×n) partition of (3.27) is the upper triangular form $U_{k|k-1}$ we have been looking for. We now have to find the D-factor. To satisfy relation (3.21) while using (3.27) we can write $W\bar{D}W^t$ as $WTT^t\bar{D}T(WT)^t$. From this it follows that the updated D-factor is:

$$D_{k|k-1} = T^t\bar{D}T \ . \qquad (3.28)$$

Summarizing the time update equationsof the U-D factors are given as:

$$D_{k|k-1}(j,j) = v_j^t\bar{D}v_j \qquad (3.29a)$$

$$U_{k|k-1}(j,k) = \frac{w_j^t\bar{D}v_k}{v_k^t\bar{D}v_k} \ . \qquad (3.29b)$$

The classical (weighted) Gram-Schmidt orthogonalization method as given in (3.24) is known to be numerically unstable. The drawback of the classical algorithm is that the resulting vectors generally are not orthogonal and thus iterations are

necesssary. Actual implementations are based on the so-called Modified Weighted Gram-Schmidt (M-WGS) orthogonalization method [Kaminski et al., 1971]. The M-WGS is basically an algebraic re-arrangement of the classical algorithm. The modified procedure has favourable numerical characteristics. An algorithm for the U-D filter time update is given in Appendix I.

3.5 IMPLEMENTATION CONSIDERATIONS

Having introduced the covariance and information filters as well as their respective square root formulations and the U-D covariance factorization filter a choice between the different mechanizations for the actual implementation has to be made. To justify the choice of any mechanization its computational efficiency, numerical aspects and conditions imposed by the specific application have to be taken into account.

3.5.1 Computational Efficiency

A popular way to assess the computational efficiency of different filter mechanizations is to compare the number of operations (additions, multiplications, divisions, and square roots) necessary to compute a full filter cycle (one time update and one measurement update). These comparisons, usually called operation counts, give a measure of the relative speed of the algorithms. Operation counts for various mechanizations can be found in Kaminski et.al. [1971], Bierman [1973a,1977], Maybeck [1979], and Chin [1983].

In the literature contradictory computational efficiencies are reported (see, e.g., LeMay [1984]). This may be due to the fact that some authors apply special storage strategies or only use scalar measurement updates. Usually the operation counts only consider the filter algorithm itself. Operations not directly related to the filter process (e.g., input/output and bookkeeping logic) are not taken into account. Furthermore the computation of the transition matrix (Φ), the design matrix (A), and the process covariance matrix (Q) are not considered either. In cases where the filter model is non-linear the computations of these matrices and the necessary iterations may account for 90% of the total cycle time. In many applications a special problem structure can be exploited, which can reduce the computational burden as well, but this is not considered in the operation counts.

As we are restricting ourselves to linear models some useful remarks can be made regarding the computational efficiency of the different mechanizations. Basically the standard Kalman (covariance) filter algorithm is the simplest to implement and usually the fastest. The information filters are of computational interest if the dimension of the measurement vector is larger than the dimension of the state vector. Covariance filters are more efficient if frequent estimates of the state are required. The stabilized Kalman filter is computationally always less efficient than the standard Kalman filter by about 10%-50%. Square root covariance filters and U-D filters may be up to 50% slower than the standard Kalman filter. For certain applications, however, it has been shown [Thornton and Bierman, 1980] that non-standard mechanizations may be computationally just as efficient as the standard Kalman filter. In problems where the dimension of the state vector is small the differences between the various mechanizations are generally not of great importance, because then for all mechanizations low operation counts (and thus cycle times) can be obtained.

3.5.2 Numerical Aspects

Up to now we only dealt with different Kalman filter mechanizations from a computational point of view. The usefulness of different mechanizations depends, besides on the computational efficiency, mainly on their numerical stability. It has been indicated that the equivalence of the square root and U-D factorization algorithms with the numerically stable Householder and Givens (orthogonal) transformations guarantees the numerical stability of these mechanizations. It has also been mentioned that this is only true for the (error) covariance update. Therefore the comparison of different mechanizations from a numerical point of view is very interesting. Such comparisons can be found in Thornton and Bierman [1980] and Verhaegen and van Dooren [1986].

The basic motive for the development of square root related filters was to enable filter computations in single precision arithmetic (to ease storage requirements and to speed up the computations). Single precision arithmetic computations are indeed possible due to the enhanced numerical stability of the square root filters.

It has been shown in Thornton and Bierman [1980] that if computations are performed in double precision arithmetic, no numerical difficulties are to be encountered for any mechanization. In these cases the mechanization with the highest computational efficiency (which is not necessarily the standard Kalman filter) can be selected for the computations.

When computing with single precision arithmetic one will always encounter some numerical degradation compared to the double precision computations [Thornton and Bierman, 1980]. With the use of the square root or U-D filters only limited

degradation will occur. Thornton and Bierman also encountered numerical instability using the stabilized Kalman Filter with single precision arithmetic. On the other hand Verhaegen and van Dooren [1986] show, by means of a theoretical error analysis, that square root formulations and the stabilized Kalman filter should have the same numerical accuracy.

3.5.3 Practical Considerations

For practical applications special characteristics of the problem can often be exploited, such as the sparseness of the transition or covariance matrices (which are often reduced to diagonal form). In applications the state transition matrix has often a (block) triangular form. Exploiting these characteristics can lead to considerable computational savings.

In the investigation of Kalman filter mechanizations we dealt primarily with linear time varying systems. In some applications the Kalman filter may reach (after an initial transient period) a (almost) steady state condition. The computation of the gain matrix is the largest computational burden of the Kalman filter algorithm. The use of a steady state gain can lead to considerable computational savings. Even in time varying systems this not always leads to a serious performance degradation of the filter (although the filter will be suboptimal). In such cases an extensive a priori sensitivity analysis (see Gelb [1974]) is mandatory. An interesting application of the use of a temporary steady state gain matrix in a navigation environment is discussed by Upadhyay and Damoulakis [1980] and Gylys [1983].

Exploiting special characteristics of the problem at hand will always reduce the computational burden, irrespective of the mechanization used. Not considering the

problem characteristics has as advantage that readily available filter routines can be used in implementing a Kalman filter.

3.5.4 Filter Mechanizations for Kinematic Positioning

In kinematic positioning and navigation frequent updates of the state vector and its covariance matrix are needed. In a navigation environment the use of covariance filters seems to be the most appropriate.

The use of the standard Kalman filter in (integrated) navigation systems at sea is very common. Mostly the standard Kalman filter mechanization is implemented. As long as the computations are performed in double precision arithmetic numerical problems are not likely to be encountered. An investigation of navigation filters which deals with numerical considerations is, e.g., Ayers [1985]. Ayers concludes that for his simple positioning problem (position a ship with two lines of position (ranges)) special numerical techniques are not necessary and the standard Kalman filter performs well.

The most popular square root related covariance filter at the moment is the U-D filter. The U-D filter is very well documented [Bierman, 1977; and Thornton and Bierman, 1980]. This mechanization is computationally very efficient. It is actually the computationally most efficient square root related covariance filter mechanization. When using single precision arithmetic the results are only slightly worse than the standard Kalman filter results computed with double precision arithmetic. If computer burden is to be minimized or the computations can only be performed in single precision arithmetic (e.g., when programming in PASCAL) the U-D factorization

mechanization seems to be the most efficient alternative algorithm for navigation problems.

4. LINEAR SMOOTHING

4.1 INTRODUCTION

In Chapter 2 we dealt with the concepts of prediction and filtering. We now focus our attention on the case where one wants to obtain an optimal estimate of the system state in the past, using measurements both before and after the time of interest. This is the so-called **smoothing** problem. Although smoothing has been defined earlier the definition found in Gelb [1974] is repeated here: "Smoothing is a non-real-time data processing scheme that uses all measurements between t_0 and t_N to estimate the state of a system at a certain time t_k, where $t_0 \leq t_k \leq t_N$ ".

Because smoothing algorithms use data after the time for which the state is estimated a time delay in the estimation process is inevitable. For real-time state estimation filtering and prediction are therefore the only feasible techniques. For certain real-time applications in surveying and hydrography, however, a small time delay in obtaining a state estimate may be acceptable. In these cases a smoothed state estimate will be preferred as more information is taken into account in computing the estimate. Especially if for some application the data can be processed in an off-line, postmisson mode smoothing techniques should be considered. Generally a tradeoff has to be made between the extra computational burden and time delay related to the smoothing algorithms and the improved accuracy of a smoothed estimate.

In section 4.2 the principles of smoothing are discussed on the basis of the forward-backward filter approach In section 4.3 three classes of smoothing problems are discussed. The concept of smoothability is introduced in section 4.4 and some

comments on the possible use of smoothing techniques in hydrography and surveying are made.

4.2. PRINCIPLES OF SMOOTHING

4.2.1 Forward-Backward Filter Approach

The principles of a linear smoother can best be demonstrated as a suitable combination of two filters. The first filter, called the "forward filter", operates as a standard Kalman filter on all data up to and including time t_k, starting at t_0. At the same time a second filter, called the "backward filter", operates on all data after time t_k, starting at t_N. These two filters use all available information in the specified interval and provide two uncorrelated estimates of the state and its covariance at time t_k. The optimal combination of both estimates yields the optimal smoothed estimate (see Fig. 4.1).

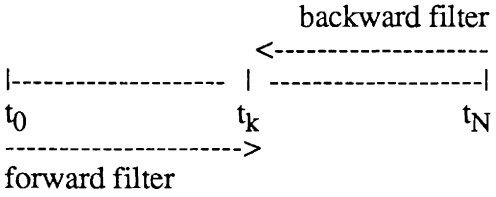
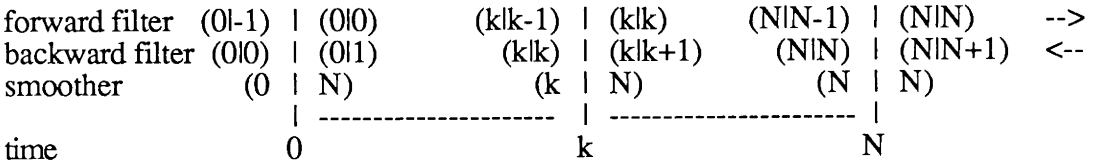


Fig. 4.1 Forward-backward filter approach.

Before the forward-backward formulation is described in more detail some notational conventions are introduced. It is assumed that measurements are available from time t=0 to t=N. The superscript b denotes backward filter estimate; the superscript s denotes smoothed estimate. The following notation for the time indices is used:



A smoothed estimate of the state and its covariance at time t_k for data given in the interval [0,N], where t_k is an element of the time interval [0,N], are given as the optimal combination of two optimal filters [Gelb, 1974]:

$$\hat{x}^s_{k|N} = P^s_{k|N}(P^{-1}_{k|k}\hat{x}_{k|k} + P^{b\,-1}_{k|k+1}\hat{x}^b_{k|k+1}) \tag{4.1a}$$

$$P^{s\,-1}_{k|N} = P^{-1}_{k|k} + P^{b\,-1}_{k|k+1} \quad . \tag{4.1b}$$

Given that the forward and backward estimates are uncorrelated these equations can readily be verified using standard adjustment calculus.The equations show that the forward filter uses all data up to and including time t_k , while the backward filter uses all data after time t_k and in the last step is predicted "backward" to time t_k. Equation (4.1b) shows that the covariance of the smoothed estimate is always smaller than or equal to the covariance of the forward filter. This is one of the motives to perform smoothing. This result could have been expected since one uses not only data up to and including time t_k , but all available data in a certain interval.

The linear (forward) Kalman filter was derived in Chapter 2. The formulation of the backward filter is less straightforward. This is mainly due to the starting values of the backward filter. Equation (4.1b) shows that the error covariance of the smoothed estimate is always smaller than the error covariance of the forward filter except for the terminal time t_N. It can be seen from eqn. (4.1b) that at time t_N the covariance of the smoothed state is equal to the forward filter covariance as both are conditioned on

exactly the same data. Hence it follows that the a priori covariance of the backward filter is infinite, i.e., no a priori statistical information is available to start up the backward filter. Thus

$$P^{b\,-1}_{N|N+1} = 0$$

and assuming $\hat{x}^b_{N|N+1}$ is finite

$$P^{b\,-1}_{N|N+1}\hat{x}^b_{N|N+1} = 0 \quad .$$

This means that the backward filter has to be implemented in the inverse covariance formulation (also called information filter), because the standard Kalman filter cannot cope with an infinite a priori covariance. The backward inverse covariance filter formulation is given in Maybeck [1982, pp. 9-10], where also a somewhat modified algorithm to compute the smoothed estimate from the the forward filter and the backward filter is given. Brown [1983] circumvents this problem by suggesting that as long as the a priori backward filter covariance is chosen 10 times as large as the a priori covariance of the forward filter a time reversed standard Kalman filter can be used to implement the backward filter. This may be a valid solution for the example Brown uses, but one can easily devise some smoothing application where this will lead to serious suboptimality of the backward filter. In general the inverse covariance formulation, at least for the first backward steps, is to be preferred. Thereafter one can use the covariance formulation for the backward filter.

The measurement update equations of the covariance form of the backward filter are given as:

$$K^b_k = P^b_{k|k+1}A^t_k(A_kP^b_{k|k+1}A^t_k + R_k)^{-1} \qquad (4.2a)$$

$$\hat{x}^b_{k|k} = \hat{x}^b_{k|k+1} + K^b_k(y_k - A_k\hat{x}^b_{k|k+1}) \tag{4.2b}$$

$$P^b_{k|k} = P^b_{k|k+1} - K^b_k A_k P^b_{k|k+1} \quad . \tag{4.2c}$$

The time update equations of the backward filter are defined as (note the reversed time order in the transition matrices):

$$\hat{x}^b_{k|k+1} = \Phi(k,k+1)\hat{x}^b_{k+1|k+1} \tag{4.3a}$$

$$P^b_{k|k+1} = \Phi(k,k+1)(P^b_{k|k+1} + G_{k+1}Q_{k+1}G^t_{k+1})\Phi(k,k+1)^t \quad . \tag{4.3b}$$

Conceptually the forward-backward formulation of the optimal smoother is the easiest way to demonstrate the properties of an optimal smoother. For practical applications other mechanizations are utilized, which will be treated in the next section. It is apparent that smoothing involves more computational effort than filtering. A tradeoff between the extra computational cost and the availability of better estimates has to be made.

The linear smoother has been introduced using a forward-backward filter formulation. In the literature numerous other derivations can be found. Kailath and Frost [1968] use the innovations approach. Koch [1982] uses best linear unbiased estimators in the regression model. Anderson and Moore [1979] employ the technique of state augmentation. Houtenbos [1982] formulates the smoothing problem as a least-squares adjustment problem. Rauch et al. [1965] use maximum likelihood estimates. Starting from single-stage and double-stage optimal smoothers Meditch [1969] derives the smoothing algorithms via algebraic manipulations and induction. An overview of the development of smoothing theory is given by Meditch [1973].

4.3 THREE CLASSES OF SMOOTHING PROBLEMS

In the literature three classes of smoothing problems are distinguished. The classification depends on how the time k (for which an estimate is needed) and the length of the data interval N are related. The three classes are **fixed-interval**, **fixed-point**, and **fixed-lag** smoothing.

- Fixed-interval smoothing (k variable, N fixed)

Given measurements in a fixed interval from initial time t_0 to final time t_N a smoothed state estimate at time t_k, where t_k is an element of the time interval $[t_0,t_N]$, is required, based on all measurements in the interval. This approach is usually followed in off-line processing.

- Fixed-point smoothing (k fixed, N increasing)

Estimate the state at a single fixed point in time t_k as more and more measurements become available after time t_k.

- Fixed-lag smoothing (k increasing, N-k fixed)

A smoothed estimate at a fixed time interval back in the past is computed. The computation of the state estimate at time t_k is delayed for a fixed time t_{lag} to take advantage of the additional information in the interval of duration t_{lag} (=N-k) of the most recent measurements.

4.3.1 Fixed-Interval Smoothing

The fixed-interval smoothing algorithm described in this paragraph is due to Rauch et al. (1965) and is known as the Rauch-Tung-Striebel (RTS) algorithm. For the complete time interval (say [0,N]) a forward filter solution is computed and the predicted and updated state estimates and their corresponding error covariances are stored. Starting from the boundary conditions

$$\hat{x}^s_{N|N} = \hat{x}_{N|N}$$
$$P^s_{N|N} = P_{N|N}$$

the smoothed estimate and its covariance are computed backwards (for k = N-1, N-2,...,0) as:

$$\hat{x}^s_{k|N} = \hat{x}_{k|k} + B_k(\hat{x}^s_{k+1|N} - \hat{x}_{k+1|k}) \tag{4.4a}$$

$$P^s_{k|N} = P_{k|k} + B_k(P^s_{k+1|N} - P_{k+1|k})B^t_k \tag{4.4b}$$

respectively, where

$$B_k = P_{k|k}\Phi(k+1,k)^t P^{-1}_{k+1|k} \tag{4.4c}$$

is the smoothing gain matrix.

Drawbacks of the above algorithm are that the predicted covariance matrix ($P_{k+1|k}$) has to be inverted for every recursion and that all results of the forward filter have to be stored.

If some state variables become very well determined, the inversion of $P_{k+1|k}$ can lead to instability of the RTS-algorithm. To avoid this (numerical) instability Bierman

[1973b] developed an alternative algorithm (Modified Bryson-Frazier smoother). Gonthier [1984] found that if the smoother algorithm is implemented in double precision arithmetic both algorithms lead to identical results and no numerical problems are encountered.

In the forward-backward formulation of the smoother, as described in section 4.2, the storage of intermediate results was not necessary. This, however, is only true if merely one smoothed state estimate at time t_k in the interval [0,N] is required. Besides, the computational efficiency of the forward-backward formulation suffers from the combined use of the covariance and inverse covariance filter mechanizations.

Because fixed-interval smoothers are mainly used in off-line processing storage requirements are not critical. Off-line the implementation of the algorithm can be done in double precision arithmetic, so that the RTS algorithm can be used.

4.3.2 Fixed-Point Smoothing

The fixed-point smoother as given in this paragraph is taken from Meditch [1969]. The fixed-point smoother uses the output of a standard forward running Kalman filter. The initial values of the state and its error covariance are:

$$\hat{x}^s_{k|k} = \hat{x}_{k|k}$$
$$P^s_{k|k} = P_{k|k}$$

Assume that the fixed-point estimate at a certain time t_k is required. The optimal fixed-point smoothed state estimate and its covariance are generated by the following system of equations (for k fixed and j = k+1,k+2,...):

$$\hat{x}^s_{k|j} = \hat{x}_{k|j-1} + C_j(\hat{x}_{j|j} - \hat{x}_{j|j-1}) \qquad (4.5a)$$

$$P^s_{k|j} = P^s_{k|j-1} + C_j(P_{j|j} - P_{j|j-1})C^t_j \tag{4.5b}$$

or

$$P^s_{k|j} = P^s_{k|j-1} - C_jK_jA_jP_{j|j-1}C^t_j \tag{4.5c}$$

where

$$C_j = \prod_{i=k}^{j-1} B_i \tag{4.5d}$$

and

$$B_i = P_{i|i}\Phi(i+1,i)^tP^{-1}_{i+1|i} \quad . \tag{4.5e}$$

Maybeck [1982, p. 16] gives an alternative algorithm which avoids the inversion of $P_{i+1|i}$ as required for each recursion.

4.3.3 Fixed-Lag Smoothing

The fixed-lag smoother is the most complicated one of the three smoothing categories discussed. One reason for this is the startup problem. The fixed-lag smoother is derived by Meditch [1969] and in an alternative way by Anderson and Moore [1979]. In this paper the fixed-lag smoother due to Meditch is given. The optimal fixed-lag smoothed state estimate and its covariance for a N-step time lag are generated by the following recursions (for k = 0,1,2, . . .):

$$\hat{x}^s_{k+1|k+N+1} = \Phi(k+1,k)\hat{x}^s_{k|k+N} + C_{k+N+1}K_{k+N+1}(y_{k+N+1} - A_{k+N+1}\hat{x}_{k+N+1|k+N}) + U_{k+1}(\hat{x}^s_{k|k+N} - \hat{x}_{k|k}) \tag{4.6a}$$

$$P^s_{k+1|k+N+1} = P_{k+1|k} - C_{k+N+1}K_{k+N+1}A_{k+N+1}P_{k+N+1|k+N}C^t_{k+N+1} - B_k^{-1}(P_{k|k} - P^s_{k|k+N})B_k^{-t} \tag{4.6b}$$

where

$$C_{k+N+1} = \prod_{i=k+1}^{k+N} B_i = B_k^{-1}C_{k+N}B_{k+N} \quad , \tag{4.6c}$$

B_k is defined as in (4c), and U_{k+1} is given by:

$$U_{k+1} = Q_k\Phi(k,k+1)^tP_{k|k}^{-1} \quad . \tag{4.6d}$$

Equation (4.6d) is a simplified version of the general formula and is only valid if the dynamic system model is defined as:

$$\underline{x}_{k+1} = \Phi(k+1,k)\underline{x}_k + \underline{w}_k \quad .$$

The fixed-lag smoother uses the output of a simultaneously running Kalman filter. The smoothed estimate and its covariance are computed with (4.6a) and (4.6b) respectively, starting from the initial conditions $\hat{x}^s_{0|N}$ and $P^s_{0|N}$. These initial smoothed estimates must be generated by a fixed-point smoother which is iterated N times, starting at time t_k=0. The start up problem increases the computational complexity of the fixed-lag smoother.

From eqns. (4.6a) to (4.6d) it can be seen that the computational and storage burden of this algorithm is considerably larger than that of an linear Kalman filter for the same problem. The added computational burden and the availability of the state estimate after a delay of N-k steps have to be counterbalanced with the performance benefit of the fixed-lag smoother.

Anderson and Moore [1979, ch. 7] state that the mechanization of the filter as given in (4.6a) to (4.6d) is unstable. Their alternative derivation of the fixed-lag

smoother is quite long and will be not be repeated here. Furthermore Anderson and Moore appear to be the only authors who seem to have noticed the inherent instability of the presented algorithm. Even the more recent textbooks (e.g. Maybeck [1982], Brown [1983]) give the fixed-lag smoother as derived by Meditch [1969].

To avoid the computational complexity of the fixed-lag smoother Brown [1983] suggests the use of the RTS-algorithm if the lag is not too large. Using this approach one filters forward to the current measurement and then sweeps back a fixed number of steps with the RTS algorithm outlined in section 4.3.1. The start up problem can be avoided by starting the backward sweep at time $t = t_{lag}$. No smoothed estimates for the first t_{lag} instants are then available. Brown [1983, pp. 283-285] gives a solution for the computation of these first estimates.

4.4. SMOOTHABILITY, GENERAL REMARKS, AND APPLICATIONS

4.4.1 Smoothability

The definition of smoothability is taken from Gelb [1974, p.163]: "A state is said to be smoothable if an optimal smoother provides a state estimate superior to that obtained when the final optimal filter estimate is extrapolated backwards in time". By definition only components of the state which are controllable with respect to the dynamic system noise are smoothable. The concept of controllability is not discussed here; the reader is referred to Maybeck [1979, Ch. 2].

In case of a linear system with no system noise (i.e., $Q_k = 0$ for every k) smoothing should not be considered. Constant states are not smoothable. The optimal estimate of a constant state can be found by propagating the final forward filter state estimate backward in time. This is illustrated by means of the RTS-algorithm. The smoothing gain (4.4c) is given as:

$$B_k = P_{k|k}\Phi(k+1,k)^t P_{k+1|k}^{-1} \quad .$$

In case $Q_k = 0$,

$$P_{k+1|k}^{-1} = (\Phi(k+1,k)P_{k|k}\Phi(k+1,k)^t)^{-1} \quad . \tag{4.7}$$

Inserting (4.7) in (4.4c) one finds:

$$B_k = (\Phi(k+1,k))^{-1} = \Phi(k,k+1) \quad . \tag{4.8}$$

The computation of the smoothed state estimate (4.4a) thus reduces to:

$$\hat{x}_{k|N}^s = \hat{x}_{k|k} + \Phi(k,k+1)(\hat{x}_{k+1|N}^s - \Phi(k+1,k)\hat{x}_{k|k}) = \Phi(k,k+1)\hat{x}_{k+1|N}^s \tag{4.9}$$

and finally

$$\hat{x}_{k|N}^s = \Phi(k,N)\hat{x}_{N|N} \tag{4.10a}$$

which is nothing but the final forward filter state estimate propagated backwards in time. Similarly one finds for the covariance:

$$P_{k|N}^s = \Phi(k,N)P_{N|N}\Phi(k,N)^t \quad . \tag{4.10b}$$

4.4.2 General Remarks

In this chapter the algorithms for three classes of smoothing problems have been presented. Although the algorithms may look very different, they also have a lot in common.

- A (standard) forward running filter is at the basis of all smoothing algorithms. A smoothed estimate cannot be obtained without a simultaneously forward running filter.
- Unlike the filter algorithm (see Chapter 2) the computation of the smoothed state estimate ($\hat{x}^s_{k|N}$) does not require the computation of the corresponding smoothed covariance ($P^s_{k|N}$).
- For actual smoother implementations most programming effort is related to data management and storage problems.

4.4.3 Applications

For surveying applications mainly fixed-interval and fixed-lag smoothing are of interest. Fixed-interval smoothing is usually performed off-line, because the (fixed) interval of data can be quite long. Fixed-lag smoothing is the real-time oriented class of smoothing problems, as the estimate of the state only lags for a certain lag t_{lag}. For on-line processing only fixed-lag smoothing is of interest. A fixed-lag smoother can be implemented using the RTS-algorithm, as was suggested at the end of section 4.3.3.

Applications of smoothing techniques in surveying and hydrography are error control for inertial systems [Gonthier, 1984], the postmission analysis of three-dimensional seismic campaigns at sea, and hydrographic surveys [Guenther and

Thomas, 1987]. It seems, however, that for most kinematic or dynamic problems in surveying filtering is deemed sufficient.

Smoothing may be a useful extension to filtering in kinematic positioning if now and then no valid data are available due to, e.g., data outages or (rejected) outliers in the data. As smoothing always implies forward filtering the optimal filtered estimate is available, while smoothing may help bridge short data gaps.

5. PERFORMANCE ANALYSIS OF KALMAN FILTERS — THE INNOVATIONS APPROACH —

5.1 INTRODUCTION

In kinematic positioning and navigation it is common to process data from different sensors simultaneously in a so-called integrated navigation system to obtain a best estimate of position. The algorithm implemented in these integrated navigation systems is usually the Kalman filter. To obtain useful (positioning) results using an integrated navigation system it is absolutely crucial that the performance of the underlying filter is at an optimum. Therefore performance analysis of Kalman filters is of considerable importance. Misspecifications in the dynamic and/or measurement model invalidate the results of estimation. It is therefore essential to have ways to verify the validity of the assumed mathematical model and to detect any misspecifications in the mathematical model. An important role in the process of performance analysis is played by the so-called **innovation sequence**.

Methods for the detection of departures from optimality are all based on the innovation sequence. Performance analysis of Kalman filters based on the innovation sequence was introduced by Mehra and Peschon [1971]. The innovation sequence of an optimal filter has precisely defined characteristics which can be compared with the output of an actually implemented Kalman filter. The innovation process contains all information to assess the optimality of filter operations. Furthermore the innovation

process is the primary source for the detection of misspecifications of the model (e.g., outliers in the data) and adaptive filtering.

In section 5.2 the innovation sequence is introduced and its characteristics are outlined. The monitoring of the innovation sequence is discussed extensively in section 5.3. An introduction to error detection based on the innovation sequence is presented in section 5.4. Finally some implementation considerations are discussed in section 5.5.

5.2 THE INNOVATION SEQUENCE

The innovation sequence is defined as the difference between the actual system output and the predicted output based on the predicted state. The innovation sequence is given by:

$$\underline{v}_k = \underline{y}_k - A_k \hat{\underline{x}}_{k|k-1} \ , \ \text{for } k=1,2,.... \ . \tag{5.1}$$

The sequence is called innovation sequence because it represents the new information brought in by the latest observation vector. As can be seen from the Kalman filter measurement update equations presented in Chapter 2 the filtered state is a linear combination of the predicted state and the innovation. Hence the innovations are an important quantity in the Kalman filter process. This was already pointed out by Kalman in his original derivation of the filter equations [Kalman, 1960]. In fact the filter algorithm can be derived completely based on the innovation sequence only [Kailath, 1968].

For the objectives of this chapter the innovation sequence is considered to be the output of the filter. For linear systems the output is Gaussian if the input is Gaussian and thus the innovation sequence is considered to be Gaussian. Hence only the mean and covariance of the innovation sequence have to be specified to describe the statistical properties of the innovation sequence completely. Kailath [1968] has shown that if the filter is optimal, the innovation sequence is a white noise sequence with zero mean and known covariance (a sequence {x(n)} is called white (or purely random) if it consists of a sequence of uncorrelated random variables). Thus:

$$E\{\underline{v}_k\} = 0$$

$$E\{\underline{v}_k \underline{v}_l^t\} = Q_{v_k} \delta_{kl}$$

where

$$Q_{v_k} = A_k P_{k|k-1} A_k^t + R_k \tag{5.2}$$

is the covariance matrix of the innovation $\underline{v}_k$.

The covariance of the innovation has been encountered in the derivation of the Kalman filter measurement update as well and thus it can be seen that the innovation sequence is an integral part of the Kalman filter process.

Now the statistical properties of the innovation sequence have been described we indicate how the innovations can be used for the performance analysis of the Kalman filter. Since the properties of the innovation sequence are strictly defined if the filter is optimal, the innovation sequence resulting from an actually implemented filter can be monitored and compared to the faultless model description. Deviations from the theoretical characteristics may be caused by mismodelling of the dynamic and/or measurement model, failure of sensors, and outliers in the data. Mismodelling of any

kind will make the innovation sequence depart from its theoretically defined nominal values.

If the system is modelled perfectly, the innovations will be "small" and correspond to random fluctuations in the output since all systematic trends are eliminated by the model. If the model is misspecified, the innovations will be "large" and will display a trend because the model no longer represents the physical system adequately. The general performance of the filter can be monitored by analyzing the zero mean, Gaussianness, given covariance, and whiteness of the innovation sequence. Furthermore the innovation sequence is the sole source for outlier detection. Outlier detection is also related to the wider field of failure detection, in which one tries to detect abrupt changes, called failures, in dynamical systems. The general theory of failure detection falls outside the scope of this paper. A survey of this topic is given by Willsky [1976]. Finally the innovation sequence offers a possible approach to adaptive filtering. Adaptive filtering techniques based on the innovation sequence are reviewed by Chin [1979]. The innovations approach thus facilitates an array of very useful Kalman filter related techniques.

5.3 MONITORING THE INNOVATION SEQUENCE

In this section we will focus our attention on the performance analysis of Kalman filters based on the approach of Mehra and Peschon [1971]. In their approach the innovation sequence is analyzed in the time domain and is used to assess the general performance of the filter. To facilitate performance analysis in both on-line and off-line environments the approach in the time domain is pursued throughout.

In monitoring the innovation sequence the assumption is that the filter operates at an optimum, i.e., the innovation sequence is zero mean, Gaussian, white, and of known covariance. This constitutes our so-called null hypothesis. The alternative hypothesis is that the null hypothesis is false. In the approach described in this section all characteristics of the innovation sequence are tested separately. The null hypothesis will not be tested in a large, all encompassing, test. In this section only tests for departures from the null hypothesis are presented. No methods for system diagnosis (i.e., determining the cause of of a detected departure) are presented.

Before the actual monitoring techniques are discussed a general philosophy of data analysis, that is basically adhered to in this section, is outlined. Bendat and Piersol [1986] describe a general strategy for data acquisition and data processing. Once the data — in our case the innovations — have been obtained two actions are distinguished: data qualification and data analysis. Data qualification should precede data analysis, but in practice data qualification and data analysis are usually performed at the same time.

Data qualification encompasses the investigation of the basic characteristics of the data, namely: stationarity, presence of periodicities, and normality. A sequence is called stationary if its statistical properties do not change in time. Stationarity of a sequence can often be derived from the properties of the underlying model generating the random data. In case of an optimal linear Kalman filter one expects a stationary innovation sequence. Nonstationarity in a stochastic sequence can, for instance, be revealed as a trend in the mean value of the data. Periodicity of the data is not investigated explicitely. Some inferences regarding periodicities can be made from the estimated autocorrelation function, to be discussed shortly.

Whiteness is the most important property of the innovation sequence. If whiteness of the innovation sequence is not established, data analysis cannot be performed properly. To test if the sequence of innovations is independent Bendat and Piersol [1986] introduce two non-parametric tests, namely, the run test and the reverse arrangements test. Non-parametric means that no assumptions concerning the probability distribution of the data being evaluated are made.

Although it has been stated several times that the innovation sequence is Gaussian if the observations are distributed normally, normality has to be verified. The equivalence of the probability density function of the innovations to the normal density function is tested by means of the (Chi-square) goodness-of-fit test, once more a non-parametric test [Bendat and Piersol, 1986].

Independence can also be tested with tests based on the autocorrelation function. The autocorrelation function describes the correlation between data for various time lags and renders some more information than the run test and reverse arrangements test. By means of the autocorrelation function one may, for example, be able to reveal periodicities in the data.

Although it would seem logical to start with a test of the mean of the innovation sequence, we will defer the tests of mean to a later stage because they are dependent on the estimate of the covariance, which in its turn is based on the assumption of whiteness. We will see that to perform the analysis of whiteness, the sample mean of the innovation sequence still has to be computed first, only its analysis is deferred to a later stage.

In the following time series and statistical techniques are used extensively. An overview of these topics can be found in Priestley [1981] and Morrison [1976]

respectively. To ease the derivations normalized innovations are considered in the rest of this section:

$$\underline{\nu}_k = (A_k P_{k|k-1} A_k^t + R_k)^{-1/2} \underline{\nu}_k \tag{5.3}$$

so that

$$E\{\underline{\nu}_k \underline{\nu}_l^t\} = I\delta_{kl} \ . \tag{5.4}$$

In the following N denotes sample size (the number of samples used to analyse the innovation sequence).

The autocorrelation function ρ_k of a stationary (normalized innovation) sequence for a lag k is defined as:

$$\rho_k = E\{(\underline{\nu}_i - \bar{\nu})(\underline{\nu}_{i+k} - \bar{\nu})^t\} \ , \ \text{for } k=0,\pm1,\pm2,.... \tag{5.5}$$

where $\bar{\nu}$ denotes the expected value of $\underline{\nu}_i$ (i.e., $\bar{\nu} = E\{\underline{\nu}_i\}$).

Assuming that the innovation sequence is ergodic (a sequence is called ergodic if its ensemble averages (eqn. 5.5) correspond with its time averages (eqn. 5.6)) the autocorrelation function can be estimated as:

$$\hat{R}_k = \frac{1}{N} \sum_{i=1}^{N-|k|} (\nu_i - \hat{\bar{\nu}})(\nu_{i+|k|} - \hat{\bar{\nu}})^t, \quad k = 0,\pm1,\pm2, \ldots, \pm(N-1) \tag{5.6}$$

where $\hat{\bar{\nu}}$ is the sample mean

$$\hat{\bar{\nu}} = \frac{1}{N} \sum_{i=1}^{N} \nu_i \ . \tag{5.7}$$

Under the null hypothesis the autocorrelation functions for different lags k are asymptotically independent and are normally distributed with zero mean and

covariance I/N. In general it suffices to compute and check the diagonal elements of the autocorrelation function. At the α-level of significance the null hypothesis concerning whiteness is rejected if

$$| \hat{R}_k(i) | > N_{(1/2)\alpha} / \sqrt{N}$$

where $\hat{R}_k(i)$ is the *i* th diagonal element of the autocorrelation matrix for lag k, and $N_{(1/2)\alpha}$ is the upper α probability point of the normal distribution N(0,1).

The estimate of the autocorrelation function (eqn. 5.6) for a zero lag is nothing but the estimate of the covariance matrix of the innovation sequence:

$$\hat{R}_0 = \frac{1}{N} \sum_{i=1}^{N} (\nu_i - \hat{\nu}) (\nu_i - \hat{\nu})^t \quad . \tag{5.8}$$

Under the null hypothesis the separate elements of the m-dimensional innovation sequence are uncorrelated. Each correlation coefficient (of a total of 1/2m(m-1)) can be tested for significance separately. It is prudent, however, to begin each study with a test of the hypothesis

$$H_0 : \hat{R}_0 = I$$

against the alternative hypothesis

$$H_a : \hat{R}_0 \neq I \quad .$$

An approximation of a test statistic for this test is given by Morrison [1976]:

$$X^2 = (N - 1 - (2m + 5) / 6) \sum_{i<j} \sum \hat{r}_{ij}^2 \tag{5.9}$$

where $\hat{r}_{ij}$ are the elements of matrix $\hat{R}_0$. The statistic X^2 has a Chi-squared distribution with 1/2m(m-1) degrees of freedom. The null hypothesis is rejected at the α-level of significance whenever

$$X^2 > \chi^2_{\alpha;\ 1/2m(m-1)}$$

where $\chi^2_{\alpha;1/2m(m-1)}$ is the upper α probability point of the Chi-squared distribution $\chi^2_{1/2m(m-1)}$ with 1/2m(m-1) degrees of freedom.

The correlation coefficients can also be tested separately. Subject to some conditions [Bendat and Piersol, 1986] the quantity

$$w_{ij} = \tfrac{1}{2} \ln ((1 + \hat{r}_{ij})(1 - \hat{r}_{ij})) \qquad i < j \tag{5.10}$$

is distributed approximately normal under the null hypothesis with zero mean and variance

$$\sigma^2_{w_{ij}} = 1/(N-3) \ .$$

The null hypothesis concerning a single correlation coefficient between elements of the innovation sequence is rejected at the α-level of significance if

$$|w_{ij}| > N_{(1/2)\alpha}/\sqrt{N-3} \ .$$

If it is established that the innovation sequenece is white (i.e., correlations in time are negligible) the covariance matrix (which under the null hypothesis corresponds to the correlation matrix for a zero lag) and the mean can be tested. Both tests assume that the innovation sequence is white.

Morrison [1976] shows that under the null hypothesis the covariance matrix times N (i.e. $N*\hat{R}_0$) has a Wishart distribution. Mehra and Peschon [1971] state that the

trace of $\hat{R}_0$ has a Chi-squared distribution with (n-1)m degrees of freedom. Using the trace of the sample covariance matrix it can thus be tested easily if the covariance matrix is equal to the identity matrix.

Under the null hypothesis the innovation sequence is zero mean and the mean (eqn. 5.7) is normally distributed and has a covariance of 1/N. At the α-level of significance the null hypothesis concerning the zero mean is rejected whenever

$$|\nu_i| > N_{1/2\alpha}, \quad i=1,\ldots,m \quad .$$

As is indicated in Mehra and Peschon [1971] this test suffers from the fact that the covariance of the innovation sequence is assumed to be known. They propose to use T^2-statistic instead:

$$T^2 = N \hat{\nu}^t \hat{R}_0^{-1} \hat{\nu} \quad . \tag{5.11}$$

The derivation of this statistic can be found in Morrison [1976]. Under the null hypothesis the quantity is distributed according to the F-distribution with m and N-m degrees of freedom. The null hypothesis concerning the mean is rejected at the α-level of significance if

$$T^2 > \frac{m(N-1)}{N-m} F_{\alpha\,;\,m,N-m} \quad .$$

In this section a general methodology for the analysis of (normalized) innovation sequences has been presented. Departures from zero mean, normality, whiteness, and a known covariance can be detected by the methods described herein. The described techniques, however, pertain to the innovation sequence in general. More specific alternative hypotheses could be formulated if one has some idea of the causes leading to the departures from nominal values (e.g., sensor failures, outliers in the data, etc.).

As part of the innovation sequence monitoring and analysis some extra model parameters can be estimated. One can try, for instance, to derive a possible model that accounts for departures from whiteness. The reader is referred to Priestley [1981] for details about the estimation of additional parameters.

5.4 ERROR DETECTION

Now that several techniques to monitor the general filter performance using the innovation sequence have been discussed, another important application of the innovation sequence can be introduced. We restrict ourselves to a specific application of the monitoring of the innovation sequence. The innovation process can be used to detect outliers in the observations.

In section 5.2 we defined the innovation sequence as the sequence that contains all new information brought in by the latest observation. An outlier in the observations is certainly new information as the filter model cannot anticipate possible outliers. Therefore the innovation sequence is at the base of all outlier detection algorithms. Outliers in the data affect the property of zero mean of the innovation sequence. In statistics and adjustment theory various tests have been developed which deal with such phenomena. Misspecifications of the model at a certain time can be detected by a so-called **overall model test**. A misspecification detected by such an overall model test can be diagnosed further by a so-called **slippage test** if the misspecification affects the property of zero mean of the random variable. The application of overall model and slippage tests to Kalman filter performance analysis is discussed in Teunissen and Salzmann [1988]. The use of these tests in a more general setting in

adjustments is discussed in, e.g., Kok [1984]. In this section the terminology introduced in Teunissen and Salzmann [1988] is maintained. It must be kept in mind that a single outlier in the data not only affects the tests mentioned above but also the general innovation sequence monitoring described in the previous section.

In adjustment theory most interest has been directed to so-called local tests. Local means that the tests performed at time t_k only depend on the predicted state at time t_k and the observations at time t_k. The local overall model (LOM) test detects misspecifications in the mathematical model occurring at time t_k. It is defined as:

$$\underline{T}_k = \frac{\underline{v}_k^t (A_k P_{k|k-1} A_k^t + R_k)^{-1} \underline{v}_k}{m} = \frac{\underline{v}_k^t Q_{v_k}^{-1} \underline{v}_k}{m} \quad . \tag{5.12}$$

Whenever at a certain time t_k

$$T_k > \chi^2_{\alpha;\, m}$$

a misspecification of the model is detected. If it is assumed that the detected misspecification is due to a single outlying observation (this constitutes our so-called alternative hypothesis) we can apply the one-dimensional local slippage test

$$\underline{w}_k = \frac{c_i^t Q_{v_k}^{-1} \underline{v}_k}{(c_i^t Q_{v_k}^{-1} c_i)^{1/2}} \tag{5.13}$$

where

$$c_i = (0,.....,0,\ 1,\ 0,.....,0)^t \ , \quad \text{for } i=1,....,m \quad .$$
$$\qquad\quad 1,\quad i-1,i,i+1,\quad ,m$$

The vector c_i indicates that for the alternative hypothesis we assume that an outlier in the *i* th observation is the possible cause of the misspecification of the model. The observation i for which the w-test statistic is a maximum is then the most likely

outlying observation. In general other alternative hypotheses may be specified and this will affect the form of the c-vector.

Equations (5.12) and (5.13) show that the test statistics are functions of the innovations. The above tests, frequently applied in adjustments, have the advantage that they can be executed in real time. Corrective action is thus also possible in real time.

To apply testing methods for the detection of outliers the null hypothesis and the alternative hypothesis have to be defined quite precisely. The mere introduction of system noise in the filter model indicates that in general the knowledge of the underlying model for dynamic systems is not as perfect as in problems usually considered in surveying. Furthermore it is expected that in dynamic environments the measurement sensors are more prone to failures of any kind. Apart from the fact that the modelling of dynamic systems may not be as sophisticated as the models used in classical adjustment problems in surveying, often also the redundancy for a single Kalman filter measurement update can be quite low. Therefore a more cautious approach is usually followed for error detection in dynamic systems as the local tests may not be able to detect global unmodelled trends.

In Willsky [1976] the following test statistic is defined:

$$\underline{T}^{k} = \sum_{i=k-M+1}^{k} \underline{v}_i^t (A_i P_{i|i-1} A_i^t + R_i)^{-1} \underline{v}_i \tag{5.14}$$

where M denotes the delay one is willing to accept in detecting a model misspecification. In practice a small delay M may be acceptable, though no real-time corrective action can be taken anymore. It can be seen that (5.14) actually represents nothing but a sum of the local overall model test statistics introduced earlier. The test

statistic (5.14) is actually closely related to the global overall model (GOM) test as given in Teunissen and Salzmann [1988]

$$\underline{T}^{k}_{GOM} = \frac{\sum_{i=k-M+1}^{k} m_i \underline{T}_i}{\sum_{i=k-M+1}^{k} m_i} \quad . \tag{5.15}$$

A decision a misspecification has occurred is made once

$$T^{k}_{GOM} > \chi^{2}_{\alpha;\ \sum_{i=k-M+1}^{k} m_i} \quad .$$

This test statistic is the weighted mean of the local overall model test statistics and can thus be computed very easily. Rejection of the global overall model test is due to misspecifications in the time interval [k-M+1,k]. The type of misspecification can be diagnosed with the global slippage test. The reader is referred to Teunissen and Salzmann [1988].

5.5 IMPLEMENTATION CONSIDERATIONS

The innovation sequence is an intrinsic element of the Kalman filter. The innovations as well as their (second order) statistics are generated automatically by the filter process. The performance analysis of Kalman filters was dealt with in two separate sections. The general filter performance can be monitored using the techniques described in section 5.3. Specific model misspecifications can be detected more easily with the tests introduced in section 5.4. Although it is recommended that both types of

performance analysis techniques should be implemented some remarks concerning their implementation are made.

The general filter performance analysis is extremely useful in the design and implementation stage of a Kalman filter. If the performance analysis can be performed off-line (which will usually be the case in the design stage), the full range of analysis techniques described in section 5.3 can be applied. It is felt, however, that this performance analysis can be executed in an on-line environment as well. The extent of the on-line performance analysis depends primarily on the computer power available. The number of samples used to analyse the innovation sequence (i.e., N) should neither be chosen too small or too large. A large value of N (e.g., N>200) requires considerable computing time, whilst a too small N (e.g., N<50) does not enable a proper time series analysis. Depending on the computer facilities available a happy medium has to be found for on-line applications. The analysis discussed in section 5.3 requires computations every *N*th cycle of the Kalman filter. For on-line environments it is suggested that the general performance analysis be used as an alarm system. If the null hypothesis (i.e., the Kalman filter operates at an optimum and thus the innovation process has strictly defined statistical properties) is (constantly) rejected, the user should be informed. If sufficient computer power is available, it can even be attempted to estimate extra model parameters that account for the possible departures from the nominal characteristics of the innovation sequence. If no additional parameters are estimated, the monitoring of the innovation sequence constitutes a fault detection technique. A diagnosis of what causes the filter to depart from its optimal properties is generally not performed.

To detect misspecifications in the model on a real time or nearly real-time basis specific tests were introduced in section 5.4. It was shown that especially the local

tests can be implemented very easily. As the local redundancy in Kalman filter measurement updates is sometimes quite low and the dynamic model may not be very well defined it can be advantageous to implement the global overall model test and the global slippage test if a small delay in the error detection is acceptable. For the global tests the delay (i.e., M) should not be chosen too large. Firstly, a large M will hamper taking corrective action if a misspecification has been detected. Secondly, a large M may decrease the probability of correct detection of a model misspecification because averaging over a large number of innovations may smooth out the effect of a model misspecification. As a diagnostic tool (particularly if the model misspecifications are caused by outliers in the data) the tests described in section 5.4 are very powerful and supplement the general analysis discussed in section 5.3.

As far as outliers in the data or sensor failures are not the cause for departures from the nominal characteristics of the innovation sequence, misspecifacations of the system and measurement noise are usually at the basis of these departures. This brings us into the realm of adaptive filtering which is outside the scope of this chapter. The interested reader is referred to Chin [1979].

6. SUMMARY

The use of Kalman filters in kinematic positioning is well established. In this report several aspects of Kalman filtering are reviewed that pertain to kinematic positioning problems. Actual applications of Kalman filtering techniques to specific kinematic positioning problems are not considered.

Linear Kalman Filter

A general discrete time dynamic and measurement models are introduced and the linear Kalman filter for this model is subsequently derived using a least-squares approach. It is indicated how continuous time models can be transformed to equivalent discrete time models. Some attention is paid to possible nonlinearities of the model. For applications in hydrography and surveying the iterated extended Kalman filter is probably the most frequently applied filter algorithm.

Computational Considerations

For actual implementations of Kalman filters various filter mechanizations can be used. If frequent estimates of the state and its covariance are required (as is the case in kinematic positioning) the so-called covariance filter mechanizations are to be preferred. In these mechanizations the covariance of the system state is propagated in time. It is found that if double precision arithmetic is used in the filter computations no numerical difficulties arise for any filter mechanization and thus the most efficient mechanization can be implemented. If computations have to performed in single

precision arithmetic or the reduction of the computational burden is of the utmost importance, the U-D covariance factorization filter can be considered for implementation. This mechanization is closely related to the square root covariance filters. For every application considerable computational savings can be achieved if the problem structure is taken into account.

Linear Smoothing

The number of applications of smoothing techniques is rather limited because of the delay inherent in the smoothing algorithm. In situations where off-line computations are feasible, however, smoothing techniques can greatly enhance the position estimates in a postmission approach. Conceptually the smoothing problem can be decomposed into two filtering problems. The smoothed estimate can be obtained from a combination of the two filtered estimates. The classes of fixed-interval, fixed-point, and fixed-lag smoothing are discussed. For kinematic positioning problems fixed-interval smoothing seems most appropriate for a postmission approach. If a small delay in obtaining the state estimate is acceptable, a fixed-lag smoother can be considered for "real-time" applications.

Performance Analysis

If a Kalman filter operates at an optimum, the innovation sequence generated by the filter has some very well defined statistical properties. Any misspecification in the filter model will cause departures from these optimal characteristics. Therefore the innovation sequence of an actually implemented Kalman filter is the primary source for the analysis of the performance of the filter. A general filter performance analysis methodology is discussed which basically functions as an alarm system. General and

specific misspecifications in the model, in particular outliers in the observations, can be detected and diagnosed with the overall model test and the slippage tests respectively. These test statistics are also functions of the innovations.

BIBLIOGRAPHY

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

Ayers, H.B. (1985). Least-squares radio positioning in the offshore Environment. UCSE Report #20012, Division of Surveying Engineering, University of Calgary.

Bendat, J.S. and A.G. Piersol (1986). *Random Data: Analysis and Measurement Procedures*. 2nd ed.,Wiley-Interscience, New York.

Bierman. G.J. (1973a). A comparison of discrete linear filtering algorithms. *IEEE-Trans. on Aerospace and Electronic Systems,* Vol. IEEE-AES 9(1), pp. 28-37.

Bierman, G.J. (1973b). Fixed interval smoothing with discrete measurements. *Int. Journal of Control* , Vol. 18(1), pp. 65-75.

Bierman, G.J. (1977). *Factorization Methods for Discrete Sequential Estimation.* Academic Press, New York.

Box, G.E.P and G.M. Jenkins, (1970). *Time Series Analysis Forecasting and Control.* Holden-Day, San Francisco.

Brown, R.G. (1983). *Introduction to Random Signal Analysis and Kalman Filtering.* Wiley, New York.

Chin, L. (1979). Advances in adaptive filtering. In: *Control and Dynamic Systems,* Vol. 15, Ed. C.T. Leondes, Academic Press, New York.

Chin, L. (1983). Advances in computational efficiencies of linear filtering. In: *Control and Dynamic Systems,* Vol. 19, Ed. C.T. Leondes, Academic Press, New York.

Gelb, A. (ed.) (1974). *Applied Optimal Estimation.* MIT-Press, Cambridge, Ma.

Gonthier, M. (1984). Smoothing procedures for inertial survey systems of local level type. UCSE Report #20008, Division of Surveying Engineering, University of Calgary.

Guenther, G.C. and W.L. Thomas (1987). Optimum utilization of positioning data in SDS III. *International Hydrographic Review,* Vol 64(1), pp. 35-58.

Gylys, V.B. (1983). Design of real-time estimation algortihms for implementation in microprocessor and distributed processor systems. In: *Control and Dynamic Systems,* Vol 19, Ed.C.T. Leondes, Academic, New York.

Hasenack, E. (1986). Precision and reliability of a Kalman filter (in Dutch). Thesis, Department of Geodesy, Delft University of Technology.

Houtenbos, A.P.E.M. (1982). Prediction, filtering, and smoothing of offshore navigation data. *The Hydrographic Journal,* No. 25, pp. 5-16.

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

Jenkins, G.M. and D.G. Watts (1968). *Spectral Analysis and its Applications*. Holden-Day, San Francisco.

Kailath, T. (1968). An innovations approach to least-squares estimation Part I: Linear filtering in additive white noise. *IEEE-Trans. on Automatic Control,* Vol. IEEE-AC 13(6), pp.646-655.

Kailath, T. and P. Frost (1968). An innovations approach to least-squares estimation Part II: Linear smoothing in additive white noise. *IEEE-Trans. on Automatic Control,* Vol. IEEE-AC 13(6), pp. 655-660.

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, Vol. 82, pp. 34-45.

Kaminski, P.G., A.E. Bryson, and S.F. Schmidt (1971). Discrete square root filtering: A survey of current techniques. *IEEE Trans. on Automatic Control,* Vol. IEEE-AC 16(6), pp. 727-736.

Koch, K.R. (1982). Kalman filter and optimal smoothing derived by the regression model, *Manuscripta Geodaetica,* Vol. 7(2), pp. 133-144.

Kok, J.J. (1984). *On Datasnooping and Multiple Outlier Testing*. NOAA Technical Report NOS NGS 30, U.S. Department of Commerce, Rockville, Md.

Krakiwsky, E. (1975). A synthesis of recent advances in the method of least squares. Department of Surveying Engineering, Lecture Notes #24,University of New Brunswick.

LeMay, J.L. (ed.) (1984). *Kalman Filtering* (2 volumes). Lecture notes of a five day short course, Aug. 27-31, University of California at Los Angeles.

Maybeck, P.S. (1979). *Stochastic Models, Estimation, and Control.* Vol. 1, Academic Press, New York.

Maybeck, P.S. (1982). *Stochastic Models, Estimation, and Control.* Vol. 2, Academic Press, New York.

Meditch, J.S. (1969). *Stochastic Optimal Linear Estimation and Control.* McGraw-Hill, New York.

Meditch, J.S. (1973). A survey of data smoothing for linear and nonlinear dynamic systems. *Automatica,* Vol. 9, pp. 151-162.

Mehra, R.K., and J. Peschon (1971). An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica,* Vol. 7, pp. 637-640.

Morrison, D.F. (1976). *Multivariate Statistical Methods.* 2nd ed, McGraw-Hill, New York.

Priestley, M.B. (1981). *Spectral Analysis and Time Series.* Academic Press, London.

Rauch, H.E., F. Tung, and C.T. Striebel. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA Journal,* Vol 8(3), pp. 1445-1450.

Salzmann, M.A. (1988). The impact of system integration on position estimation - two case studies. Department of Surveying Engineering, Technical Memorandum, University of New Brunswick.

Teunissen, P.J.G. and M.A. Salzmann (1988). Performance analysis of Kalman filters. Paper presented at HYDRO88, Amsterdam, 15-17 November.

Thornton, C.L. and G.J. Bierman (1980). UDU^t covariance factorization for Kalman Filtering. In: *Control and Dynamic Systems,* Vol. 16, Ed. C.T. Leondes, Academic, New York.

Upadhyay, T.N. and J.N. Damoulakis (1980). Sequential piecewise recursive filter for GPS low dynamics navigation. *IEEE-Trans. on Aerospace and Electronic Systems.* Vol. IEEE-AES 16(4), pp. 481-491.

Verhaegen, M. and P. van Dooren (1986). Numerical aspects of different Kalman filter implementations. *IEEE Trans. on Automatic Control,* Vol. IEEE-AC 31(10), pp. 907-917.

Willsky, A.S., (1976). A survey of design methods for failure detection in dynamic systems. *Automatica,* Vol 12, pp.601-611.

APPENDIX I

U-D COVARIANCE FACTORIZATION FILTER MECHANIZATION

In this appendix (pseudo) FORTRAN mechanizations of the basic computational parts of the U-D filter are given. The mechanizations are kept as simple as possible in order to provide a more detailed insight than is provided by the general algorithms. The mechanizations are derived from the algorithms as given in Maybeck [1979, pp. 392-397]. These algorithms are not necessarily the most efficient ones computationally. Special (and probably faster) mechanizations can be found in Bierman [1977] and Thornton and Bierman [1980] (in the latter publication some printing errors are present in the provided FORTRAN mechanizations).

For actual implementation the programmer can exploit the unitary upper triangular characteristic of the U-factor. The elements of the D-factor can be stored on the diagonal of U. Storage requirements can be reduced further by storing the matrices vector-wise. In these cases some bookkeeping logic has to be added.

In the mechanizations n, s, and m denote the dimension of the state vector, the dimension of the system noise vector, and the dimension of the observation process respectively.

I.1 U-D Factorization

An algorithm (closely related to the backward running Cholesky decomposition) to generate the U-D factors from a positive definite symmetric matrix is presented.

in:	P	positive definite symmetric matrix	matrix (dimension n)
out:	U	U-factor (unitary upper triangular)	matrix (dimension n)
	D	D-factor	vector (dimension n)

```
D( n , n) = P(n,n)
U(n,n) = 1
do i = n-1,....,n
    U( i, n) = P( i, n)/ D( n, n)
end do
do j = n-1,....,1
    D( j, j) =  P( j, j)
    do k = j+1,...., n
        D( j, j) =  D( j, j) - D( k, k) U( j,  k)^2
    end do
    do i  = j ,....,1
        if ( i = j)  then
            U( i , j)  =  1
        else
            U( i, j) =  P( i, j)
            do k  = j+1,...., n
                U( i, j)  =  U( i, j) - D( k, k) U( i, k) U( j, k)
            end do
            U( i, j)  =  U( i, j) / D( j, j)
        end if
    end do
end do
```

I.2 U-D Covariance Factorization Filter Measurement Update

The U-D filter processes observations sequentially. It is assumed that in general more than one observation is available for a measurement update. Therefore the complete

design matrix is introduced in the mechanization. The observations are assumed to be uncorrelated. If the original observations are correlated, a whitening procedure has to be applied first. If the Kalman gain matrix is to be computed explicitly the unweighted Kalman gain has to be extracted from the routine. In this mechanization the unweighted Kalman gain is computed solely per observation.

For observation #i one finds: $K(i,j) = b(j)/\alpha(i)$ for $j=1,....,n$, with $b(j)$ the unweighted Kalman gain and $\alpha(i)$ the variance of the innovation of observation #i.

in:	U	U-factor (unitary upper triangular) $U_{k\|k-1}$	matrix (dimension n)
	D	D-factor $D_{k\|k-1}$	vector (dimension n)
	A	design matrix	matrix (m×n)
	y	observations	vector (dimension m)
	x	predicted state $\hat{x}_{k\|k-1}$	vector (dimension n)
	R	variances of observations	vector (dimension m)
out:	U	U-factor (unitary upper triangular) $U_{k\|k}$	
	D	D-factor $D_{k\|k}$	
	α	variances of innovations	vector (dimension m)
	res	innovations $(y_k - A_k\hat{x}_{k\|k-1})$	vector (dimension m)
	x	updated state $\hat{x}_{k\|k}$	vector (dimension n)

f(*) and v(*) are auxiliary vectors (of dimension n), and γ, temp, and p are auxiliary variables. b(*) is the unweighted Kalman gain.

```
do l = 1,...., m                          @ process m
                                            observations
   res( l) = y( l)                        @ compute innovations
   do  i = 1,.... n
      res( l) = res( l) - A( l, i ) x( i)
   end do
   do  i = 1,....,  n                     @ reset f and b for every
                                            loop
      f(i) = 0
      b(i) = 0
   end do
   do  i = 1,...., n                      @ compute f and v
                                            vectors
      do  j =  1,.... i
         f( i) = f( i) + U( j, i) A( l, j)
      end do
```

```
    end do
    do i = 1,...., n
        v( i) = D( i, i) f( i)
    end do
    γ = R( l)                                          @ update U-D factor
    α( l) = γ + f(1) v( 1)
    D( 1, 1) = D( 1, 1) *( γ / α( l ))
    b(1) = v(1)
    do k = 2,...., n
        γ = α( l)
        α( l) = γ + f(k) v(k)
        D( k, k) = D( k, k) *( γ / α( l ))
        b(k) = v(k)
        p = -f(k) / γ
        do j = 1,...., k-1
            temp = U( j, k)
            U( j, k) = U( j, k) + b( j ) p
            b( j) = b( j) + temp v( k)
        end do
    end do                                             @ save unweighted
                                                       Kalman gain if necessary
    do i = 1,.... n                                    @ update state
        x( i) = x( i) + ( b( i)/ α( l)) res( l)
    end do
end do
```

I.3 U-D Covariance Factorization Filter Time Update

For this mechanization the modified Weighted Gram-Schmidt orthogonalization method is used . The following matrices are defined:

$$W = \begin{bmatrix} \Phi(k,k-1)U_{k-1|k-1} & G_{k-1} \end{bmatrix}$$

$$\overline{D} = \begin{bmatrix} D_{k-1|k-1} & 0 \\ 0 & Q_{k-1} \end{bmatrix} .$$

In actual implementations the matrices W and $\bar{D}$ have not to be used explicitely. The matrix W is destroyed completely during the update. The system noise variances (stored in Q_{k-1}) remain unchanged.

in:	W		matrix (n×(n+s))
	U	U-factor (unitary upper triangular) $U_{k-1\|k-1}$	matrix (dimension n)
	$\bar{D}$	D-factor and diagonal part system noise	vector (dimension n+s)
out:	U	updated U-factor (unitary upper triangular) $U_{k\|k-1}$	
	D	updated D-factor $D_{k\|k-1}$	

c(*) and d(*) are auxiliary vectors of dimension (n+s).

```
do k = n,....,1
    do  j = 1,...., n+s
        c( j) = D̄( j, j) W( k, j)
    end do
    D( k, k) = 0
    do  j = 1,...., n+s
        D( k, k)  =  D( k, k) + W( k, j) c( j)
    end do
    do  j = 1,...., n+s
        d( j)  =  c( j) /  D( k, k)
    end do
    do j = 1,...., k-1
        U( j, k) = 0
        do  j = 1,...., n+s
            U( j, k)  =  U( j, k) +  W( j, i) d( i)
        end do
        do i = 1,...., n+s
            W( j, i)  =  W( j, i) -  U( j, k) W( k, i)
        end do
    end do
end do
```