

# Efficient Kalman Filtering Algorithms for Hydrodynamic Models

## Proefschrift

ter verkrijging van de graad van doctor  
aan de Technische Universiteit Delft,  
op gezag van de Rector Magnificus  
Prof. ir. K. Wakker,  
in het openbaar te verdedigen  
ten overstaan van een commissie,  
door het College voor Promoties aangewezen,

op maandag 27 april 1998 te 13.30 uur

door

Martinus VERLAAN,

wiskundig ingenieur,  
geboren te Gouda

Dit proefschrift is goedgekeurd door de promotor:  
Prof. dr. ir. A.W. Heemink

Samenstelling promotiecommissie:

Rector Magnificus, voorzitter,	
Prof. dr. ir. A.W. Heemink, promotor,	Technische Universiteit Delft
Prof. dr. G.J. Olsder,	Technische Universiteit Delft
Prof. dr. ir. P. Wesseling,	Technische Universiteit Delft
Prof. dr. ir. G.S. Stelling,	Technische Universiteit Delft
Dr. ir. J.J. Leendertse,	private consultant
Prof. dr. R.K. Price,	International Institute for Infrastructural Hydraulic and Environmental Engineering (IHE)
Prof. dr. ir. P.M. Van Dooren,	Université Catholique de Louvain

Dit proefschrift is mede tot stand gekomen met financiële bijdrage van het  
Rijksinstituut voor Kust en Zee, Rijkswaterstaat.

ISBN 90-369-3452-4

# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Hydrodynamic models . . . . .	7
1.2	Three hydrodynamic problems . . . . .	8
1.2.1	Storm surge forecasting . . . . .	8
1.2.2	Reconstruction of water levels for hydrographic survey . .	11
1.2.3	On-line prediction of current profiles . . . . .	12
1.3	Data assimilation . . . . .	14
1.3.1	Introduction . . . . .	14
1.3.2	Kalman filtering . . . . .	16
1.3.3	Speeding up the Kalman filter . . . . .	17
1.4	Overview . . . . .	17
<b>2</b>	<b>Data assimilation</b>	<b>19</b>
2.1	Introduction . . . . .	19
2.1.1	Basic concepts of data assimilation . . . . .	19
2.1.2	A brief history of data assimilation . . . . .	23
2.1.3	State space formulation of numerical models . . . . .	25
2.2	Two approaches to data assimilation . . . . .	33
2.2.1	Introduction . . . . .	33
2.2.2	Comparison of data assimilation algorithms . . . . .	33
2.2.3	Probabilistic methods . . . . .	35
2.2.4	Criterion based methods . . . . .	36
<b>3</b>	<b>Kalman filtering algorithms</b>	<b>39</b>
3.1	Introduction . . . . .	39
3.2	Steady state filter . . . . .	40
3.3	Filtering, forecasting and smoothing . . . . .	41
3.4	Extensions for non-linear models . . . . .	42
3.5	Numerical aspects of Kalman filter algorithms . . . . .	45
3.6	Square root filtering . . . . .	46
<b>4</b>	<b>Suboptimal schemes for large scale models</b>	<b>48</b>
4.1	Introduction . . . . .	48
4.2	A comparison of several SOS's . . . . .	50

4.2.1	The model used for comparison . . . . .	50
4.2.2	The RRSQRT filter and Eigendecomposition filter . . . . .	51
4.2.3	Steady state filter . . . . .	53
4.2.4	Banded approximation of the covariance matrix . . . . .	55
4.2.5	Coarse grid approximation of the covariance matrix . . . . .	57
4.2.6	Partial singular value Kalman filter . . . . .	59
4.2.7	Model reduction and Kalman filtering . . . . .	62
4.2.8	The ensemble Kalman filter . . . . .	64
4.2.9	A comparison of these SOS's . . . . .	67
4.3	Conclusions . . . . .	69
<b>5</b>	<b>The Reduced Rank Square Root filter</b>	<b>70</b>
5.1	Introduction . . . . .	70
5.2	The algorithm . . . . .	71
5.2.1	Time-step . . . . .	71
5.2.2	Reduction-step . . . . .	72
5.2.3	Measurement-step . . . . .	72
5.2.4	Initialization . . . . .	73
5.2.5	Remarks . . . . .	73
5.3	Nonlinear models . . . . .	73
5.3.1	The RRSQRT filter as Extended Kalman filter . . . . .	73
5.3.2	A second order accurate RRSQRT filter . . . . .	75
5.4	Some properties of the RRSQRT filter . . . . .	76
5.4.1	Reduction to exact filter for $q = n$ . . . . .	76
5.4.2	Underestimation of the error covariance . . . . .	77
5.4.3	Convergence of the RRSQRT filter . . . . .	79
5.4.4	Example: properties and convergence for the wave model	81
5.5	Scaling of state variables . . . . .	81
5.6	Implementation . . . . .	84
5.6.1	overall structure and concepts . . . . .	84
5.6.2	The stochastic model . . . . .	86
5.6.3	Interface . . . . .	87
5.6.4	Testing . . . . .	87
5.7	Reduction of computation times for the truncation step . . . . .	88
5.8	Conclusions . . . . .	91
<b>6</b>	<b>Application of Kalman filtering to two dimensional shallow water flow: storm surge forecasting</b>	<b>92</b>
6.1	Introduction to the application . . . . .	92
6.2	A deterministic model for storm surge prediction . . . . .	93
6.3	Stochastic extension of the deterministic model and measurements . . . . .	95
6.3.1	Uncertainty at the open boundary . . . . .	95
6.3.2	Uncertainty of the meteorological forcing . . . . .	95
6.3.3	State-space formulation of model and measurements . . . . .	96
6.4	Experiments with a linear model . . . . .	97

6.4.1	Experimental design . . . . .	97
6.4.2	Experiment 1: Tracking of a 'Kelvin' wave . . . . .	97
6.4.3	Experiment 2: Accuracy of the RRSQRT algorithm . . . . .	98
6.4.4	Experiment 3: incorporating uncertainty in the wind noise	102
6.4.5	Conclusions . . . . .	103
6.5	Experiments with the full CSM model . . . . .	103
6.5.1	Experiment and results . . . . .	103
6.5.2	Conclusions . . . . .	105
6.6	Estimation of system noise and measurement noise . . . . .	106
6.6.1	Introduction . . . . .	106
6.6.2	Calibration of system noise and measurement noise . . . . .	106
6.6.3	Calibration of CSM8 noise with covariance matching . . . . .	107
6.6.4	First guess of the standard deviations . . . . .	107
6.6.5	Estimation of noise standard deviations . . . . .	109
6.6.6	Estimation of spatial correlation . . . . .	113
6.6.7	Conclusions . . . . .	114
6.7	Positioning measurements . . . . .	115
6.8	Conclusions . . . . .	121
<b>7</b>	<b>Application of a Kalman smoother to two dimensional shallow water flow: Accurate water level reconstruction</b>	<b>123</b>
7.1	Introduction . . . . .	123
7.2	Smoothing and boundary extraction . . . . .	126
7.2.1	Introduction . . . . .	126
7.2.2	A fixed-lag smoother for the RRSQRT filter . . . . .	127
7.3	Experiments and results . . . . .	130
7.3.1	Experiment with a storm surge . . . . .	130
7.3.2	A comparison of filtering and smoothing . . . . .	132
7.4	Conclusions . . . . .	143
<b>8</b>	<b>Application of Kalman filtering to three dimensional shallow water flow: On-line prediction of current profiles using HF-radar</b>	<b>147</b>
8.1	Introduction to the application . . . . .	147
8.2	The model . . . . .	148
8.3	Measurements . . . . .	151
8.4	Feasibility study . . . . .	152
8.5	Kalman filter experiments . . . . .	155
8.5.1	Application of the RRSQRT algorithm with the 3-D shallow water model . . . . .	155
8.5.2	Experiment and results . . . . .	156
8.6	Conclusions . . . . .	156
<b>A</b>	<b>Proof of convergence of the RRSQRT filter</b>	<b>163</b>
<b>B</b>	<b>List of symbols</b>	<b>168</b>

<b>Samenvatting</b>	<b>181</b>
<b>Dankwoord</b>	<b>184</b>
<b>Curriculum Vitae</b>	<b>186</b>

# Chapter 1

## Introduction

### 1.1 Hydrodynamic models

In the Netherlands the Rhine, Meuse and Scheldt rivers run into the North sea. These geographical conditions together with the fact that large area's of land in this country lie below mean sea level have made the protection against flooding very important. In addition, these waters are used for transport, fishing, drinking water, recreation and many other activities. In order to predict the effects of the various activities and their sometimes complex interactions and use the rivers, estuaries and sea in a safe and sustainable way, modeling has become more and more important.

Since several decades computers have become fast enough to model tidal flow using large numerical models. In these models physical laws such as the conservation of mass and momentum are used to compute the flow on a grid incrementally, with each new solution at the next time based on the previous solution and the forcing.

It is expected that in the coming years it will become possible to build three

Figure 1.1: North sea beach during nice weather and a storm



Figure 1.2: The Netherlands; height and main rivers

dimensional numerical models with great resolution and accuracy and including many of the relevant physical processes. Still this will not automatically lead to very accurate forecasts since part of the input needed for these models is not known accurately. This will be illustrated in three examples, that will be described in detail in the next section. These examples will be used throughout this work.

## 1.2 Three hydrodynamic problems

### 1.2.1 Storm surge forecasting

On the first of February 1953, during a severe storm, the water level at the Dutch coast became so high that some of the dikes were damaged. As a result a large area in the south western part of the Netherlands was flooded. During the disaster 1835 people died and many others lost their houses. The water level at Hook of Holland became higher than ever recorded before.

To prevent disasters of this type in the future a large project called the delta project was started. In the original plans all the estuaries except the western Scheldt would be closed, which considerably shortens the length of the coast in this delta area. The dikes that remained in contact with open sea after the closures were heightened significantly. The closures implied that the estuaries would in time become fresh water lakes. In 1979 the original plan was altered. To preserve more of the original nature of the estuaries, a movable storm surge barrier would be built in the Eastern Scheldt. During calm weather the barrier can remain open allowing the tidal currents to pass, while during severe storms the barrier can be closed to protect the land behind it.

Since it takes several hours to prepare for proper protection of the dikes and to operate the movable barrier accurate forecasts of storm surges became

Figure 1.3: In 1953 large areas were flooded

Figure 1.4: Movable storm surge barrier in the Eastern Scheldt

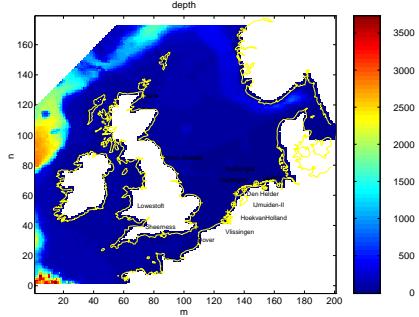


Figure 1.5: Continental shelf model

necessary. Also, it was recognized later that if there had been a proper advance warning in 1953 the death toll would have been significantly smaller.

This lead to the development of a hydrodynamic model of the European continental shelf. This model fed with meteorological forecasts can produce forecasts of the increase in water level at the Dutch coast during storms. The forecasts are now computed at regular intervals as an integrated part of the Dutch storm surge warning service.

Unfortunately, models are not perfect. Often the weather turns out quite different from the forecast given several days before. Since, the storm surge model requires input generated by a meteorological model, clearly the same limits apply to the predictability of storm surges. Water level measurements, which are now available continuously for many locations at the Scottish, English, French, Belgian and Dutch coasts, sometimes show that the model is somewhat off track. The question is, how one can improve the storm surge forecast using this additional information. However, this is far from trivial. If adaptations to the model results are not performed very carefully the model quickly becomes unstable, which makes the forecast useless. But, if the adaptations are chosen intelligently the procedure can improve the forecasts significantly. In the case of the Dutch storm surge forecast system, a procedure called the steady state Kalman filter was implemented. In fact, this procedure has more or less replaced the former system, which is now only used as a back-up.

As part of a research program to further improve the accuracy of the storm surge forecasts, the following question was posed. What is the influence of the individual measurement stations on the forecast. What happens if a station is taken out of service for financial reasons? Or, can we improve the forecasts by including more measurements? If so, where should the new measurement station be located ideally?

Figure 1.6: Measurement of water depth by echo-sounding

Figure 1.7: Example of tracks used for yearly measurement of water depth

### 1.2.2 Reconstruction of water levels for hydrographic survey

Another application in which accurate estimates of the water level are important is the hydrographic survey where measurements of the bathymetry along the Dutch coast are gathered. The depth of the coastal waters of the Netherlands is measured yearly. Using this information nautical maps can be updated, morphological changes can be studied and new input for hydrodynamic models is obtained. An important reason for monitoring the coastline is also that sand-dunes are very important for the protection of the low western part of the country against the sea. The sand on this part of the coast is not fixed. In some locations the coastline may retreat while in others sand is deposited. In order to maintain a save width of the strip of dunes everywhere, additional sand is put in some places. It is clear that a good indication of the speed of retreat of the coast is needed for determining the amount of sand needed.

The normal procedure for measuring the depth of coastal waters is by echo-sounding from a ship. For this purpose the ship will move over parallel lines perpendicular to the coast. To cover all of the Dutch coastal waters this way is a large undertaking. Using modern equipment the depth under the ship can be measured with an accuracy of only a few centimeters. But the distance between the keel of the ship and bottom is influenced by the tide. Therefore, an imaginary reference plane is used. The measured depth is translated to the depth below the reference plane. There are several different reference planes in use for this purpose: NAP (Normaal Amsterdams Peil), mean sea level and L.L.W.S. (Low Low-Water Spring) are used. At the moment there is no truly accurate way of obtaining the height of the water level to the reference plane in open sea. The accuracy of the measured depths after reduction is therefore

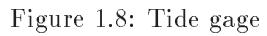


Figure 1.8: Tide gage

almost completely determined by accuracy to which the water level can be obtained, at the moment unto approximately 30 cm.

One way to compute the difference between water level and reference plane is by using measurements of the water level at a few fixed positions along the coast and to interpolate between these points. This method is relatively simple, but as stated above it is not very accurate.

An alternative would be to compute the water levels using a hydrodynamic model. A difficulty with hydrodynamic models in practice is that the model has to end somewhere because it is not feasible (nor wanted) to model all the connected waters. This results in a 'open' boundary where there is no physical boundary, but one has to be specified for the model. During very calm weather the model can be run with astronomic boundary conditions. These astronomic conditions can be determined once and used for other periods as well. The quality of the model results depends very much on the accuracy of the conditions at the open boundary. The water levels computed with such a model can have an accuracy of unto 15 cm. If the wind is stronger the meteorological influence becomes important. To include meteorological effects in the model the area of the model has to be quite large. In fact, the storm surge model mentioned in the previous application is a good choice. To use the model this way analyzed wind and pressure fields of the whole continental shelf have to be retrieved from archive. Also, the model has to cover a much larger area than the area of interest. Thus, although the reduction can in principle be computed using a hydrodynamic model, the procedure is cumbersome if meteorological effects are significant.

### 1.2.3 On-line prediction of current profiles

The Rotterdam Europort harbor is one of the busiest shipping areas in the world. The sizes of the ships in this area is very mixed and ranges from 10

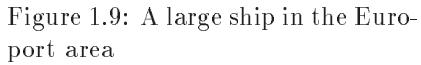


Figure 1.9: A large ship in the Europort area

meter yachts to huge container carriers. The largest ships are confined to the Euro channel and Meuse channel and need a depth of 20 meters. This means that they can not enter for some hours during low tide. The time window in which they can is influenced by negative storm surges caused by seaward wind and by the height of the waves.

Besides the depth of the water also the current is important. The large ships can only maneuver slowly and can only react to changes in the current with difficulty. Therefore, pilots are used assist and avoid dangerous situations. If necessary, tug-boats are used for additional maneuverability.

Usually pilots learn the typical patterns of currents and dangerous spots by experience. However, the flow in this region is complex and may vary strongly due to wind and river discharge. In addition new constructions, like dams and channels may alter the pattern.

To assist the pilots a current atlas is provided, in which the typical currents are shown for one tidal cycle. These charts are based on measured currents in which the effects of neap tide and spring tide are averaged and only one or two typical Rhine outflows are considered. The measurements are carried out in calm weather to reduce the meteorological effect in the resulting currents.

Some attempts have been made to produce the current atlas using one dimensional and two dimensional models. In principle the modeling approach is less costly and the meteorological effects can be fully eliminated. Mainly due to the strong three dimensional structure these attempts have not been successful. Therefore, three dimensional models were constructed and have become available recently. Another advantage of models is that also forecasts that include meteorological effects can be made. This information can be used to construct an electronic current atlas.

A parallel development is the use of advanced measurement equipment, such as Acoustic Doppler Current Profilers (or ADCP) that measure the current

Figure 1.10: Radar for measuring surface currents

Figure 1.11: Area covered by the radar

profile at a certain point almost continuously and automatically transmit the information to the shore. A disadvantage is that measurement equipment can not be positioned in the channel, since this could result in damage to ships or the equipment. Thus a representative position outside the channel has to be chosen with currents that are highly correlated to the currents in the channel.

Surface currents can also be measured with a special kind of radar, HF-radar. A great advantage is that one can measure also in the relevant areas, since the equipment is all positioned ashore. Unfortunately, the radar signal does not penetrate further than about one meter into the water, while for guiding large ships the current averaged over the upper 20 meters is needed. Moreover, the surface currents can not be extrapolated downwards using a one dimensional model (in the vertical direction) because of the complex dynamics in this area. It was observed in the 'Nieuwe Waterweg' that during part of the tidal cycle the surface current and the current near the bottom have opposite directions.

## 1.3 Data assimilation

### 1.3.1 Introduction

The reader has probably observed that the three applications just described have some common aspects. In all three cases a model exists, which can in principle produce the required information. However, the accuracy of the model is not good enough or the model has to cover a much larger area than one is interested in. On the other hand measurements exist, which are often quite accurate but can not be interpolated towards the desired locations with high accuracy and also measurements alone can not be used for forecasting. Thus there are two sources of information, model and measurements, which separately are not enough to attain the desired accuracy.

In addition these two sources of information are complementary. A large scale numerical model usually is capable of reproducing the larger patterns, but suffers from inaccurately known parameters or boundary conditions. More accurate input of the model often leads to improved performance of the numerical model. It is also possible that the accuracy of the model is deteriorated by some closure assumption or empiric formula which significantly simplifies the model. In this case it is often known which part of the model is the source of the errors. For example the mass may be preserved accurately by the model, whereas the turbulent closure equation is a large source of errors but greatly simplifies the model. Contrary to the numerical model, which covers the whole area and time interval of interest, traditional measurements are usually sparse. Typically, there are only 5 to 20 points in the area of interest where measurements are gathered. This excludes normal interpolation methods to find the values for points in between. However, the measurements are often quite accurate, thus accurate values are known at a few measurement locations. Even though there are often only relatively few measurements available the costs of obtaining these measurements are rather high, so the best possible use of these measurements has to be made. Remote sensing measurements usually are more extensive and cover the whole area of interest. Unfortunately, remote sensing equipment often does not measure directly the variable of interest, but some related quantity which is subsequently used to deduce the required information. For example, HF-radar measures the reflected radar signal, the Doppler shift of which contains information about the surface currents. To extract this information additional assumptions about the scattering process itself are needed. Furthermore for practical use of the information the surface currents have to be extrapolated downwards to recover the full current profile.

From the above it seems logical to combine the two sources of information, model and measurements, in order to take advantage of best aspects of both. In fact, it is common practice to tune a model by adapting some of the uncertain parameters to obtain a better fit with available measurements. And if we regard a simple linear relation between two variables as a model then the least squares method is also a way of combining both sources of information. However, in practice there are often too many uncertain parameters to tune by hand and the large scale numerical models are too large to use the least squares method. To overcome these problems a number of new methods were devised to integrate large models and measurements with acceptable computation times and existing theories were extended to form a common framework for comparison of the various methods and to facilitate the exchange of new idea's. Although many of the underlying theories existed for decades, or even longer, the use of large models and large numbers of measurements in a common framework has become known recently as *data assimilation*. Depending on the type of application other more or less synonymous terms as 'inverse modeling' and 'integration of data and model' are used. Data assimilation incorporates idea's from: probability theory, statistics, optimization theory, numerical linear algebra, systems theory and many other fields. The increased speed of available computers together with a wider recognition of the possibilities of a combined model and measurement

approach has lead to a large increase of research activities in the field of data assimilation in the last few years.

### 1.3.2 Kalman filtering

One way to solve many data assimilation problems, at least in principle, is by using the Kalman filter equations. This set of equations, published in 1960 [47, 48], gives a recursive solution for assimilating measurements into a linear model. The algorithm uses a two step procedure. Based on the solution at the previous time with measurements, the model is first integrated to the next instance at which measurements are available. Subsequently, this prediction is corrected using the new measurements.

It was quickly recognized by engineers that this algorithm was more useful than the Wiener filter, which was widely known at the time. And thus the Kalman filter became a popular tool in many engineering applications. Among the first applications that made use of a Kalman filter was the navigation of airplanes and spacecraft. It was for example used for navigation in the Apollo project [54].

One of the big advantages of the Kalman filter is the recursive nature of the equations, which makes it ideal for tracking an object or monitoring the present situation in a chemical plant. The new measurements can be incorporated without having to start all over. Moreover, the full problem grows as the number of measurements increase.

A disadvantage is that the Kalman filter equations provide a solution for linear models only. A simple approximate method for non-linear models is the extended Kalman filter, in which the model is linearized around the most recent estimate. For models with strong non-linearitys also higher order, more accurate methods were developed. Also a solution of the full non-linear problem does exist, but for most applications these computations are not feasible.

Another disadvantage for large models are the huge computation times needed for the Kalman filter. For example, a model involving 60000 variables, for which the numerical simulation of one day takes 20 minutes, leads to a computation of roughly 2.3 years ( $= 60.000 \times 20[\text{min}]$ ) for the Kalman filter over the same one day period. The number of 60000 variables is called the state dimension. In this example a two dimensional shallow water flow model with 20000 computational cells (or grid-points) was used, which is approximately the size of the Dutch continental shelf model that is used for the storm surge forecasts in our country. At every cell there are three variables, water level and current velocity in the two horizontal directions, hence the 60000. The 20 minutes is approximately the amount of CPU-time needed for this model on a workstation. It is clear that for storm surge forecasting these computation times are not acceptable, since the forecasts would arrive after the actual event. But also for many other applications using models of this size these computation times are too large.

### 1.3.3 Speeding up the Kalman filter

The huge computational requirements of the 'full' Kalman filter for the large scale models used in many data assimilation applications has lead to the search for faster algorithms, which still preserve the advantages of the Kalman filter. Most of these algorithms are aimed at approximating the Kalman filter equations. Since, in the linear context the Kalman filter is an optimal estimator, all the approximations will be suboptimal. This provides a basis for comparison of these suboptimal schemes (SOS's). The loss of performance can be weighed against the increase in speed. Most suboptimal schemes have some parameter that controls the computational requirements and at the same time the loss in performance.

In this thesis a new suboptimal scheme, the reduced rank square root (RRSQRT) filter, is proposed. The algorithm greatly reduces computation times, while still giving accurate results. The algorithm compares favorably to several other suboptimal schemes. The properties, among which the convergence to the optimal solution for large computation times, are studied. And the algorithm is used in three different applications. Also attention is given to non-linear aspects. Two extensions to the RRSQRT algorithm are proposed for use with nonlinear models. One is more accurate than the other at the expense of a factor two longer computation time. Both methods are relatively easy to implement.

## 1.4 Overview

The contents of this thesis can roughly be divided in two parts. In the first part consisting of chapters 2,3,4 and 5, a new method for data assimilation is proposed that is both efficient and flexible. Chapter 2 starts with a brief introduction of data assimilation in general reviewing the most important concepts. Followed by an introduction of Kalman filtering and square root filters in Chapter 3. In Chapter 4 the computational issues for large numerical models are addressed. The concept of suboptimal schemes is explained and several examples of these suboptimal schemes are given. Chapter 5 introduces a new algorithm, the 'reduced rank square root' algorithm. Several properties of this algorithm are proven and illustrated using a one dimensional model.

Contrary to the first part of the thesis which is mostly mathematical in nature, the second part describes the application of the algorithm to the three problems introduced in this chapter. In Chapter 6 the algorithm is applied to storm surge forecasting. In addition to comparison with the existing steady state filter, the possibility of providing error bounds along with the forecast and tuning of the covariances is discussed. Chapter 7 deals with the reconstruction of water levels at the open boundary for the purpose of calibrating the model or for analysis during periods where meteorological effects are not negligible. The third application in Chapter 8 is the assimilation of radar measurements of surface currents into a three dimensional shallow water model for the reconstruction

of the current profile. Finally, in Chapter 9 the most important results of this work are summarized.

# Chapter 2

## Data assimilation

### 2.1 Introduction

In this chapter a short review of data assimilation concepts will be given. The chapter is not self contained, but references to literature are included. Also the 'state space' notation used in the following chapters will be introduced here.

#### 2.1.1 Basic concepts of data assimilation

One definition of data assimilation could be:

” Data assimilation is the incorporation of measurements into a mathematical model to improve the forecasts of this model ”

Although this definition gives a first impression of the purpose of data assimilation the field of data assimilation has recently become much too broad to be captured in one sentence. A second difficulty when trying to give a definition of data assimilation is that the field is very much related to various other fields, such as meteorology, oceanography, estimation theory, Kalman filtering and optimal control. Therefore, instead of trying to refine the definition a short overview of several aspects of data assimilation will be given here.

The first aspect is what kind of mathematical models are usually used for data assimilation. A line, or more generally a linear map, is a mathematical model. Thus, using the above definition, linear regression or least squares fitting of a linear map is an example of data assimilation. Although, it may be worthwhile to study data-assimilation in such a general setting (see e.g. [68]), most of the data assimilation research is focussed on large numerical models. In this context a model is considered large if the use of a standard method, such as least squares, becomes computationally prohibitive. Thus, algorithms used for data assimilation always approximate. This may for example be due to truncation of the number of iterations in an iterative solution or simplification of the problem. The quality of a data assimilation algorithm is given by the accuracy of the result attained versus the amount of computation needed.

Most models used in data assimilation are based on a physical, first principles, model expressed in a set of partial differential equations. The numerical solution in these cases gives rise to the huge size of the models. The largest sources of uncertainty in these models are typically: closure assumptions, unmodeled processes, empirical formulas, numerical solution and uncertainty in the model input. Data assimilation methods are aimed reducing these types of uncertainty by incorporating measurements. In some data assimilation applications a statistical model is used, i.e. the model is identified from the data alone. If the size of the statistical model is large enough, incorporating measurements in these type of models is considered here to be data assimilation too.

Another restriction to the type of models used for data assimilation is that, with a few exceptions, all models are dynamical models. The situation is not stationary, but evolves in time. The temporal variable leads to three different types of applications. In some applications one wants to forecast the dynamical behavior of the system as well as possible using the measurements unto the present time and the model.

A well known example of this is the weather forecast. Using a meteorological model and various measurements the question is: "What is the best estimate that can be made of the weather for tomorrow?" The data are used to give a good estimate of the initial condition, i.e. the weather at the present time, and the model is used to compute the forecast. Since meteorological models usually show chaotic behavior the model will after some time diverge from the true weather and at the predictability limit the forecast is no longer of any value.

A second type of application tries to monitor the present situation using the model and the measurements until the present time. This type of data assimilation is often referred to as tracking, now-casting or filtering. The filtering problem is related to forecasting, but the emphasis is often on very frequent updates of the estimate, incorporating new measurements as soon as they arrive. Therefore sequential data assimilation methods that can digest new measurements without starting all over have a natural advantage in these type of applications.

In the third type of applications the dynamical behavior of the model is reconstructed afterwards using all available measurements together with the model. The results are often used for further study. Often the computational requirements for this type of application may be larger than for filtering and forecasting. Extracting the most out of sparse and expensive measurements usually has a higher priority than the speed of the method. The reconstruction using data assimilation is often called smoothing, hind-casting or re-analysis.

In addition to forecasting, filtering and smoothing there are several other types of data assimilation applications. One of these is model calibration, which is also known as parameter estimation. Many of the physical models used contain one or more empirical formula's. Also, most of these formula's have some 'constant' that must be tuned to the application for a good performance. In other cases the input needed for the model is just very difficult to measure. For example, the water level is quite hard to measure in open sea. There are possibilities, such as a pressure sensor on the seabed. The pressure differences can, after correcting for variation in air pressure, be translated to water level

variations. The equipment is usually positioned from a ship. The data is then recorded for some time, and afterwards the equipment is retrieved using the attached buoy. At the coast the water level is however measured continuously as a part of the monitoring network. It is often preferable to use these data instead of measuring in open sea. Thus, in many applications some of the parameters in the model are not known very accurately while on the other hand there are measurements available of other but related quantities or measurements of the same quantity but at a different position.

Given the parameter values, the model can be used to compute 'simulated' measurements. However, if the parameter values in the model are incorrect the 'simulated' measurements will not be equal to the true measurements. But if the error in the parameters is small the differences between 'simulated' and true measurements will in general be small too. This provides a means to estimate the parameters. Traditionally, this was done by comparing with the measurements for various values of the parameters, adapting the parameters manually before every simulation. In this manner the parameter values were improved iteratively. Many algorithms for automatic calibration operate in the same iterative way. Often additional information of the derivatives of the improvement with respect to the parameters is used to speed up the process. A very efficient method for computing these derivatives is by using an adjoint model.

Another type of data assimilation application is the design of a measurement network. Before, any of the above data assimilation procedures can be used a set of measurements has to be selected. However, some measurements contain more information than others and the total amount of measurements needed can be reduced by choosing good measurement locations. It is clear that putting two pieces of similar measurement equipment close together is often not a very good choice. But spreading the available equipment as evenly as possible over the area of interest is often not the optimal choice either. There exist some data assimilation methods that can assist in the design of a measurement network. These methods allow computation of the effects of adding or removing measurements before the actual measurements are carried out. Some methods also estimate error bounds for the estimates or the overall system performance. Ideally these should always be provided with the estimate to asses the practical value of the estimate but very often this is still computationally infeasible.

An important aspect of data assimilation problems is also the inverse nature of the problems. The measurements that are used can be computed from the model output. If the measurements were input to the model, there would not be any problem. Perhaps the measurements would have to be interpolated, but otherwise they could be used more or less directly. However, in most data assimilation applications the measurements correspond to outputs of the model and thus the model needs to be inverted. This can be illustrated using a simple example.

Figure 2.1: The inverse structure of data assimilation problems

**Example 1: Forecasts of the astronomic tide**

The tidal water elevation during calm weather can often be approximated very well with the following formula (see [72, 36] for an introduction of the astronomic tide):

$$h(t) = \sum_{i=1}^N A_i \cos(\omega_i t - \phi_i) \quad (2.1)$$

or equivalently

$$h(t) = \sum_{i=1}^N (a_i \cos(\omega_i t) + b_i \sin(\omega_i t)) \quad (2.2)$$

with

$$a_i = A_i \cos(\phi_i) \quad (2.3)$$

$$b_i = A_i \sin(\phi_i) \quad (2.4)$$

Where  $h(t)$  is the water level a time  $t$  and  $\omega_i$ ,  $A_i$ ,  $\phi_i$  are the frequency, amplitude and phase of the  $i$ 'th harmonic constituent. The frequencies ( $\omega_i$ ) are determined by the movement of the sun-earth-moon system and are considered known. Near open ocean the water level can often be approximated well using only ( $N = 10$ ) harmonic constituents, but in more shallow shelf sea's and estuaries the nonlinear nature of the tidal-wave propagation forms new higher constituents. There as many as 100 constituents may be needed.

Using equation 2.1, which is our model, one can compute forecasts of the Water level if the parameters  $A_i$  and  $\phi_i$  are known. However these parameters can not be measured directly and have to be determined from water level measurements. If we have as many measurements as parameters the parameters can, in principle, be computed by solving a set of linear equations. In practice, the water level measurements are disturbed by meteorological effects. To eliminate these effects more measurements are needed, which makes the set of linear equations over-determined. This is usually solved using a least squares method, which significantly reduces the influence of various disturbances. If the interval over which the measurements are taken is not long enough the set of equation can also be under-determined, the set of equations is then nearly singular and the solution, the parameters, become very sensitive to disturbances in the measurements. In practice the minimal length of the interval  $t_{min}$  over which measurements are needed is of order

$$t_{min} \approx \frac{1}{\min_{i \neq j}(\omega_i - \omega_j)} \quad (2.5)$$

The under-determination of the estimation process is a problem that may appear in various data assimilation applications. Often it is due to a bad design of the measurement network or the choice of parameters to estimate, but it may also be inherent to the application. Using additional information it is sometimes still possible to obtain good estimates. The regularity or smoothness of the solution is a form of additional information that is often used.

### 2.1.2 A brief history of data assimilation

The combined use of a mathematical model and measurements is probably as old as quantitative physics. When Newton deduced the value of the gravitational constant. He used both measurements, gravitational acceleration on the earth surface, trajectory of a passing comet, etc., and a model, the famous  $F = ma$  and his gravitation law for the attraction of two masses. However, the result was obtained by analytical inversion of the model. This means, the formula's were manipulated in such a way that the available measurements became input to the model and the constant the output of the model.

The next important step towards present data assimilation was provided by Gauss shortly after 1800 [22]. When studying the trajectory of planet he observed that the estimates of the approximately elliptic trajectory improved when more measurements than strictly necessary for the analytical inversion were used. In this way the influences of the other planets on the trajectory, that were present in the data but not in the model, as well as observational errors could be filtered out. In this approach both the model and the measurements are considered approximate. An estimate given by the model is improved using

the measurements, thus giving a more accurate estimate of the trajectory.

The terminology of 'data assimilation' originates from the field of meteorology, where in 50's and 60's new methods were developed to improve weather forecasts. Although, meteorological measurements date as far back as 1440, when in Korea the first rain gauge was used, a global observational network was not developed until the beginning of the 20th century. Early weather forecasts were based on 'subjective analysis', where available observations were used to draw a weather map on which the forecasts were based. Due to rapid development of computers in the 50's the applications of 'large' numerical models to weather forecasting became possible. The initial state of the model, which is the bases of the forecast, was estimated directly from the measurement. The numerical model was then used to produce the forecast. It was soon recognized that the forecasts could be improved if the initial state was not only based on the measurements but also on the forecast produced by the previous model run. The first 'data assimilation' methods just replaced a model variable by its measured value. But, this kind of forcing introduced inertial waves. Much of the research at the time was concentrated on finding a proper way to introduce the measurements without introducing oscillations. A good description of the development of data assimilation in meteorology can be found in [21].

Following the successful use of data assimilation for meteorological forecasts similar methods were also used in oceanography and later also in other fields. Compared to meteorological observations the measurements in oceanography are more sparse. On the other hand the typical time scales are longer and there is more emphasis on analysis instead of making forecasts. This allowed the use of more computationally demanding assimilation methods. In addition to the methods mentioned above there was also a strong influence of inverse methods used in solid-earth geophysics and methods used for control and filtering in engineering applications.

Control methods are normally used for automatic operation of chemical plants, movement of robots, maneuvering rockets, etc. Optimal control is based on the minimization of a formal loss function (also called criterion) [52]. This method can also be used for estimation. In this case the optimal initial and/or boundary condition is selected that has the smallest 'distance' to the measurements. Usually an adjoint model is used to efficiently determine the gradient, or direction of steepest descent. Pioneering work with this method was performed by Sasaki [64] and Chavent [14].

Another method commonly used for estimation and prediction in engineering is the Kalman filter. This method named after Kalman, who introduced the method in 1960 [47], was until recently only used for smaller models because of the huge computational burden for larger models. Some of the first applications in geophysics were those by [10, 4, 41, 42, 35]

In recent years more and more complex methods are being used. Methods of the pioneering days, such as replacement, nudging and optimal interpolation are gradually being replaced, although in real-time applications and operational forecasting optimal interpolation and its variants are still very popular because of their computational efficiency. For most other applications the methods are

either based on the minimization of a criterion, or on statistical principles, such as in the case of the Kalman filter. In both approaches the estimate is fully consistent with the dynamics of the model and the measurements, i.e. the uncertainty in both model and measurements is parameterized and the differences between model output and measurements are completely explained in terms of these uncertainties.

Another recent development is the increased use of remote sensing data. More and more data is becoming available that is not measured 'in-situ' but from a distance, hence the name remote sensing. Some examples are, radar imaging of clouds, infrared pictures from satellites, radar measurements from airplanes and many more. These methods have in common that, with some exceptions, they provide an overview or synoptic picture instead of only at sparse locations as with in-situ measurements. A disadvantage is that most remote sensing measurements have to be 'translated' to the desired form. They usually contain information about more than one quantity, but it is difficult to extract the right information from the measurements. Data assimilation is often an important step in this procedure.

### 2.1.3 State space formulation of numerical models

The analysis of data assimilation methods can be performed more easily using a systems notation rather than the original physical equations. Denoting the equations of the model in this general format stresses the similarities of data assimilation in various fields. In principle many data assimilation methods can be applied to various fields without much modification. Although the methods are generic in nature, the successful application in practice requires a thorough understanding of the physics of the model and the nature of the measurements.

Below, the notation used in the sequel for the description and analysis of the various methods is given. The analysis in this text will use discrete rather than continuous notation both in space and time. This means that the partial differential equations will be discretized before data assimilation. There are several reasons for this choice. The first is that most articles on square root filtering use a discrete time and lumped space notation and a major part of this work uses a square root filtering approach. Also some of the algorithms do not have a known continuous time equivalent. Of many other data assimilation methods both discrete and continuous forms are known. In these cases assimilation using the continuous model and discretizing afterwards often results in a similar answer than the approach used here (Compare for example [41, 19]). However, one should always be careful if the underlying model is continuous and the approach discrete. For example increasing the sampling rate of measurements with independent errors and assimilating these measurements into a continuous time model is an ill conditioned problem, which can be shown more easily in continuous time notation. For the continuous time and/or space approach the interested reader is referred to [44, 20].

The starting point in many data assimilation applications is an existing physically based mathematical model. This model is often given as a set of

partial differential equations. In the following example a simple one dimensional wave model is shown, that will be used in the sequel to illustrate the various data assimilation methods.

## Example 2: A simple wave-model

The tidal flow in a long and narrow estuary can be modeled by the De St. Venant equation [72]. For clarity we will use the following simplified equations:

$$\frac{\partial \xi}{\partial t} + D \frac{\partial u}{\partial x} = 0 \quad (2.6)$$

$$\frac{\partial u}{\partial t} + g \frac{\partial \xi}{\partial x} + c_f u = 0 \quad (2.7)$$

$$\xi(x = 0, t) = \xi_b(t) \quad (2.8)$$

$$\xi(x, t = 0) = 0 \quad (2.9)$$

$$u(x, t = 0) = 0 \quad (2.10)$$

$$u(x = L, t) = 0 \quad (2.11)$$

where  $\xi(x, t)$  is the water level above the reference plane,  $u(x, t)$  is the average current velocity over a cross section,  $t$  is the time,  $x$  is the position along the estuary,  $D$  is the water depth,  $g$  is the gravitational acceleration and  $c$  is a friction constant (see also figure 2.2). The variables  $\xi$  and  $u$  only denote the setup, i.e. the influence of meteorological effects entering through the sea-estuary boundary. The deterministic part, caused by tidal flow and the flow of the river, can be subtracted because of the linearity of the model.

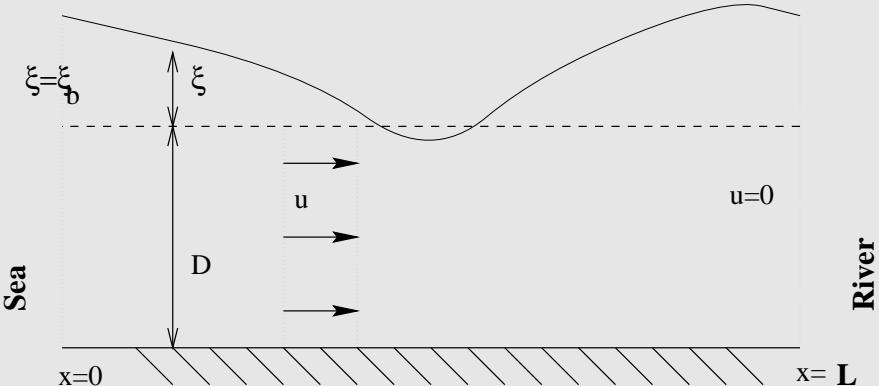


Figure 2.2: An estuary with the model variables

For this simple example an analytical solution can be found using e.g. using the method of characteristics [74], but in more complicated models this is no longer possible. Therefore the solution is often approximated numerically and implemented in a simulation package.

### Example 3: discretisation of the wave-model

For the model of equations 2.6, ..., 2.11 the numerical solution can be obtained with the finite difference approach. The discretization used here is a modification of the method of Leendertse and Stelling [67]. Their 2-D method is reduced for use with the 1-D example. The discretization is based on a staggered grid, i.e.  $\xi$  is approximated at  $x = k\Delta x$  and  $u$  at  $x = (k + \frac{1}{2})\Delta x$ . This results in the following discretized model.

$$\frac{\xi_m^{k+1} - \xi_m^k}{\Delta t} + \frac{1}{2}D \frac{u_{m+\frac{1}{2}}^k - u_{m-\frac{1}{2}}^k}{\Delta x} + \frac{1}{2}D \frac{u_{m+\frac{1}{2}}^{k+1} - u_{m-\frac{1}{2}}^{k+1}}{\Delta x} = 0 \quad (2.12)$$

$$\frac{u_{m+\frac{1}{2}}^{k+1} - u_{m+\frac{1}{2}}^k}{\Delta t} + \frac{1}{2}g \frac{\xi_{m+\frac{1}{2}}^k - \xi_m^k}{\Delta x}$$

$$+ \frac{1}{2}g \frac{\xi_{m+\frac{1}{2}}^{k+1} - \xi_m^{k+1}}{\Delta x} + \frac{1}{2}c_f u_{m+\frac{1}{2}}^k + \frac{1}{2}c_f u_{m+\frac{1}{2}}^{k+1} = 0 \quad (2.13)$$

$$\xi_0^k - \xi_b(k\Delta t) = 0 \quad (2.14)$$

$$\xi_m^0 = 0 \quad (2.15)$$

$$u_m^0 = 0 \quad (2.16)$$

$$u_{N+\frac{1}{2}}^k = 0 \quad (2.17)$$

where  $u_{m+\frac{1}{2}}^k$  and  $\xi_m^k$  denote the approximated values of  $u(x = (m + \frac{1}{2})\Delta x, t = k\Delta t)$  and  $\xi(x = m\Delta x, t = k\Delta t)$  respectively. Because of the implicit discretization every timestep results in a tridiagonal set of equations, which is solved with the double sweep method.

In the experiments the following values for the parameters are assumed  $D = 10m$ ,  $c_f = 0.0002s^{-1}$ ,  $L = 60km$ , time-step is  $\Delta t = 10min$  and  $N = 40$  grid-points for the spatial discretization, thus  $\Delta x = L/40\frac{1}{2}$ . Figure 2.3 shows the simulation of a storm surge with the model.

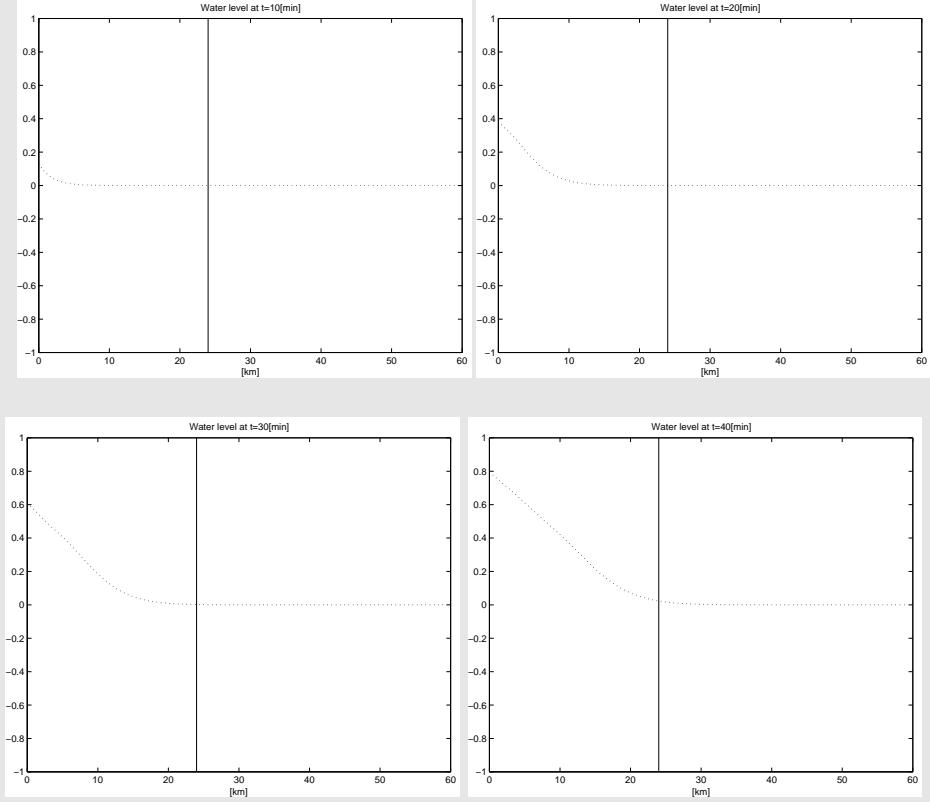


Figure 2.3: Example of a surge

For the purpose of analysis and comparison of the data assimilation methods the (deterministic) model will be denoted as

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \quad (2.18)$$

or for linear systems as

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \quad (2.19)$$

where  $\mathbf{x}(k) \in \Re^n$  is a vector containing all model variables at (discrete) time  $k$ , and  $\mathbf{u}(k) \in \Re^l$  denotes the inputs of the model. For data assimilation also a relation between the model variables and the measurements is needed. This is denoted as

$$\mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k)) \quad (2.20)$$

or for a linear relation

$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) \quad (2.21)$$

where  $\mathbf{y}(k) \in \Re^p$  is a vector containing all measurements at time  $k$ .

#### Example 4: (discrete) systems notation of wave model

In our wave example let

$$\mathbf{x}(k) = \begin{bmatrix} \xi_0^k \\ u_{\frac{1}{2}}^k \\ \xi_1^k \\ \vdots \\ u_{N-\frac{1}{2}}^k \\ \xi_N^k \\ u_{N+\frac{1}{2}}^k \end{bmatrix} \quad (2.22)$$

then the discrete model can be written as

$$\tilde{\mathbf{D}}\mathbf{x}(k+1) = \tilde{\mathbf{A}}\mathbf{x}(k) + \tilde{\mathbf{B}}\mathbf{u}(k) \quad (2.23)$$

with

$$\tilde{\mathbf{D}} := \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -\frac{g}{2\Delta x} & \frac{1}{\Delta t} & \frac{g}{2\Delta x} & 0 & \cdots & 0 & 0 & 0 \\ 0 & -\frac{D}{2\Delta x} & \frac{1}{\Delta t} & \frac{D}{2\Delta x} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -\frac{D}{2\Delta x} & \frac{1}{\Delta t} & \frac{D}{2\Delta x} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \quad (2.24)$$

$$\tilde{\mathbf{A}} := \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \frac{g}{2\Delta x} & \frac{1}{\Delta t} & -\frac{g}{2\Delta x} & 0 & \cdots & 0 & 0 & 0 \\ 0 & \frac{D}{2\Delta x} & \frac{1}{\Delta t} & -\frac{D}{2\Delta x} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{D}{2\Delta x} & \frac{1}{\Delta t} & -\frac{D}{2\Delta x} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix} \quad (2.25)$$

and

$$\tilde{\mathbf{B}} := \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.26)$$

with  $\mathbf{A} = \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{A}}$  and  $\mathbf{B} = \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{B}}$  this is in the linear form of equation 2.19. Note the time shift in  $\mathbf{u}(k) = \phi((k+1)\Delta t)$ . The number of  $N = 40$  grid-points results in a dimension of the state space of  $n = 80$ . In this example the only measurements are water level measurement at location  $x = 24[\text{km}]$ . Using

$$\mathbf{C} := [ \ 0 \ 0 \ 0 \ \cdots \ 0 \ 1 \ 0 \ \cdots \ 0 \ ] \quad (2.27)$$

with the 1 in the column 16 ( $\approx 40 \times 24/60$ ), this gives a linear measurement as in equation 2.21.

For data assimilation with a statistically based method the deterministic model must be extended to a stochastic model, i.e. the main sources of uncertainty must be given. The unknown values of these disturbances are modeled as a stochastic process (see [44] for an introduction). A well known example of a stochastic process is Gaussian white noise. Figure 2.4 shows one example of a realization of such a process. Disturbances that are correlated in time are often modeled by passing white noise through a filter. The well known auto-regressive and moving average filters ([8]) can be used for this purpose.

If the stochastic part is added to the deterministic model the resulting stochastic extension can (in most cases) be denoted as

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)) \quad (2.28)$$

$$\mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k), \mathbf{v}(k)) \quad (2.29)$$

or for linear systems as

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{F}(k)\mathbf{w}(k) \quad (2.30)$$

$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{v}(k) \quad (2.31)$$

where  $\mathbf{w}(\cdot)$  and  $\mathbf{v}(\cdot)$  are independent Gaussian white noise vector processes with dimensions  $m$  and  $p$  respectively and covariances  $\Sigma_s$  and  $\Sigma_o$ . To complete the model an additional condition for the uncertainty in  $\mathbf{x}(0)$  is needed, which is assumed to be Gaussian with mean  $E\mathbf{x}(0) = \mathbf{x}_0$  and covariance  $E(\mathbf{x}(0) - \mathbf{x}_0)(\mathbf{x}(0) - \mathbf{x}_0)' = \mathbf{P}_0$ .

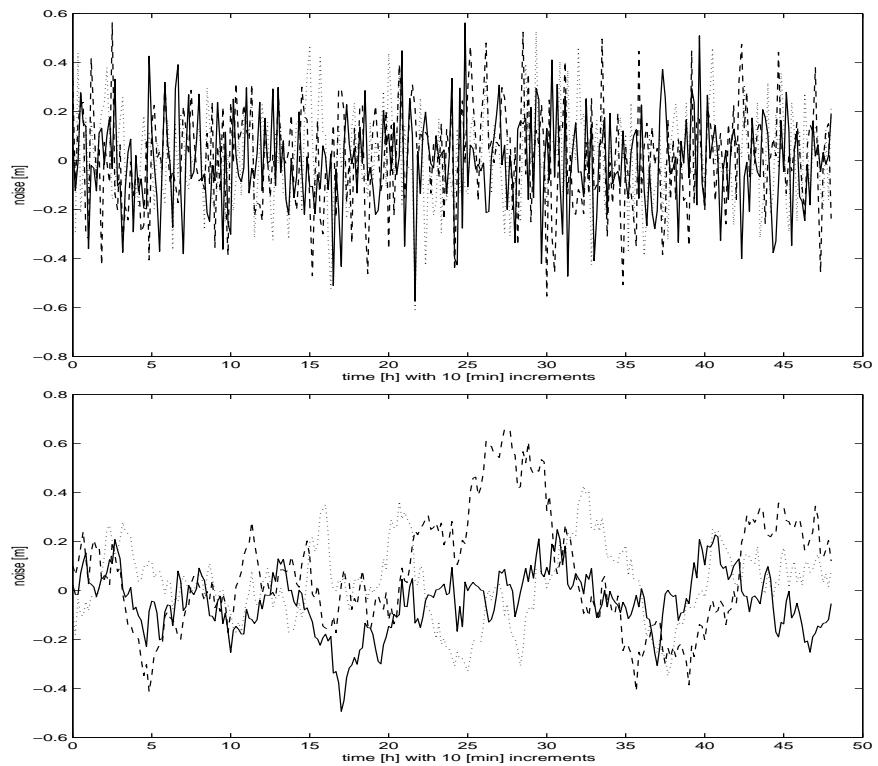


Figure 2.4: Some realizations of discrete time Gaussian white noise and correlated noise

### Example 5: stochastic extension of the wave model

The uncertainty in the wave model is caused by the unknown value of the setup at the sea-estuary boundary. It is assumed that this setup  $\xi_b$  can be modeled by an AR(1) process with  $e$ -folding time 1.0[hour] and standard deviation  $\sigma_b = 1.0[m]$ . Thus

$$\xi_b((k+1)\Delta t) = \alpha_b \xi_b(k\Delta t) + w(k) \quad (2.32)$$

with  $\alpha_b = e^{-1/6}$ ,  $E\mathbf{w}(k) = 0$  and  $E\mathbf{w}(k)\mathbf{w}(k)' = (1-\alpha^2)\sigma_b^2$ . The measurements have an accuracy of  $\sigma_m = 5.0[cm]$  standard deviation.

## 2.2 Two approaches to data assimilation

### 2.2.1 Introduction

Until recently the various data assimilation methods used in different fields had no common theoretical framework and no common notation. This made discussion and comparison of the methods difficult. The two main reasons for this were that the field was still emerging from a combination of different disciplines all with their own approach, and lack of computer speed which lead to the use of many ad hoc solutions. The developments in computer hardware has reduced the importance of the last cause. Also, the need for a common theoretical framework and notation is now recognized and in the last few years there has been much progress in this area.

### 2.2.2 Comparison of data assimilation algorithms

However, there are still two main approaches for data assimilation. The first uses the minimization of a criterion and the second is based on a probabilistic description of both model and data. Although, there is a strong relation between the two, not all criterion based methods have a probabilistic interpretation. The probabilistic approach, which is also used in this work, has become dominant in the last few years, followed by the criterion based approach, but other frameworks are still in use.

To compare methods, there are various criteria in use that are independent of the approach. It is possible to compare data assimilation schemes based on their speed, complexity, limitations or flexibility and on their performance. The speed or *required computation time* is important because the computation must be feasible. Many algorithms that can easily be used for small systems ( $n < 1000$ ) are simply infeasible for large ones ( $n > 10000$ ). But also there is often a choice between increasing the resolution of the model or using a more advanced and more computationally demanding data assimilation method, which may both increase the accuracy of the forecasts to the desired level. The

*complexity* of the method includes both the effort needed to understand as well as implement the method. In this respect there are large differences between different methods, e.g. implementation may require anything from several days to several years. The *flexibility* of a method is also important, because if a small change to the model or the availability of additional data requires large changes a more complex method that requires only small changes may be preferable.

A very important aspect of every data assimilation is the *performance*. For off-line applications this is probably the most important criterion for the selection of a particular data assimilation method. Various methods are being used to measure the performance. Some are specific to the type of method used, for example in many probabilistic methods the residuals after the assimilation can be compared against the computed standard deviation. However, there are two ways of comparing data assimilation methods even if they have a completely different theoretical background.

A well established way of checking the performance is to apply the data assimilation to real data. For this the data should be split in two parts. The first part is used in the assimilation process. The second part is then used to see if the method behaves satisfactorily. By many this is regarded as the best test, because this is what the method is going to be used for. If sufficient additional data is available for checking the performance it is often possible to get a good idea of the performance in this manner. However, since there exist often data sparse area's in the model it is not possible to judge the performance there. In addition a bad performance may be due to many other things than the quality of the algorithm. If, for example the parameters of the model or the algorithm are chosen badly this may result in a bad performance, while the data assimilation method perhaps would have worked very well with other parameters. Thus, while being a very important test in the later stages of the development of a data assimilation method for a particular application, it is not always the best method of comparing different data assimilation methods.

The second approach for comparing data assimilation methods is with a 'twin experiment'. In such an experiment the data for the assimilation are generated with a model for which some parameter, the initial condition or the forcing is altered. It is then possible to compare the estimate produced by the data assimilation scheme with the 'true' values of the model variables over the whole area. Also, in this type of experiment the values of the parameters are usually known or can be estimated accurately. A disadvantage is that it is difficult to estimate what the performance of the method will be in the application using real measurements.

For relatively small linear models it is sometimes possible to compute what the performance of a data assimilation method will be. For example for the 1-D wave example we will compute the true error covariance for several methods in the sequel.

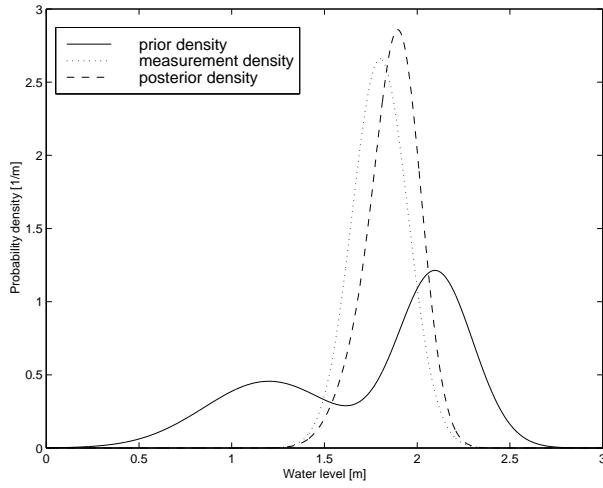


Figure 2.5: Bayes' rule applied to assimilation of measurements

### 2.2.3 Probabilistic methods

At the moment most of the more advanced data assimilation methods are based on a probabilistic approach. The main reason for this is that this approach has resulted in methods with the best performance, but also because it has proven very flexible.

Most probabilistic methods are based on the conditional density given the measurements. This conditional density, if it exists, is said to contain all the 'necessary information'. It is possible to construct from this density the conditional mean and maximum likelihood estimators as well as estimates of the errors, covariances, higher order moments etc. If the conditional density does not exist a more rigorous treatment using measure theory and probability measures is needed, but most of the main results will still hold, possibly in a slightly modified version (see [44] for an introduction).

To compute the conditional density of the random variable ( $X$ ) given the measurements ( $Y = y$ ) Bayes' rule is used. This states

$$f(X = x|Y = y) = \frac{f(Y = y|X = x)f(X = x)}{\int f(Y = y|X = x)f(X = x)dx} \quad (2.33)$$

where  $f(X = x|Y = y)$  denotes the probability density of the random variable  $X$  at  $x$  given that the random variable  $Y$  equals  $y$ . Figure 2.5 shows the effect of including the measurement for a simple 1-D example.

From equation 2.33 the conditional mean can be computed as

$$E[X|Y = y] = \int xf(X = x|Y = y)dx \quad (2.34)$$

This estimator is optimal in sense that the variance of the error is minimal. Also

this estimator is unbiased []. In the sequel the distinct notation for a random variable and its value will be dropped where no confusion can arise.

If the probability density has more than one (local) maximum the conditional mean, being the center of mass of the density, has a value between the modes. Sometimes it is preferable to choose the most like value since the conditional mean may not be possible as a realization or have a very low likelihood. For linear systems driven by Gaussian noise both methods coincide.

In principle the conditional density of the state at time  $k$  given the measurements until time  $k$ , denoted by

$$f(\mathbf{x}(k)|\tilde{\mathbf{y}}(k)) := f(\mathbf{x}(k)|\mathbf{y}(0), \dots, \mathbf{y}(k)) \quad (2.35)$$

can be calculated using the stochastic model and the measurements. However, even for very small systems these computations may become computationally infeasible. Miller estimates that a maximal state dimension of  $n = 5$  is feasible [56, 13]. This implies that for data assimilation direct computation of the conditional mean is infeasible. In the case of a linear model and Gaussian noise all (conditional)densities are Gaussian and only the mean and covariance need to be computed. These two moment completely determine the Gaussian density, which reduces the number of computation needed dramatically. In that case state dimensions of upto  $n = O(100)$  may be feasible. The Kalman filter that is discussed in the next chapter is based on this principle.

If the mode of the conditional density is chosen as an estimator computation of the full conditional density can be avoided. However if error bounds are needed the full density must still be computed. Also if the system noise is significant, i.e.  $Q > 0$  direct optimization of the conditional density has proven very difficult.

The interested reader is referred to Jazwinski [44] for a more rigorous treatment of nonlinear estimation theory.

#### 2.2.4 Criterion based methods

Many data assimilation schemes can be written as an optimization problem. There are large differences in the criteria used, but usually there is one term that is a function of the differences between measurements and the corresponding model results. Often additional terms are needed for regularization of the problem, which can for example be accomplished by adding to the criterion for changes in the control variables. These control variables are the inputs of the model that are adapted in the optimization process. Some examples of control variables are the initial condition, model parameters and model forcing.

In the last few years it was recognized that data assimilation methods based on a probabilistic approach have significant advantages. However it is often possible to express probabilistic methods as the optimization of a criterion. A good example of this is maximization of a likelihood function. By assuming Gaussian errors a quadratic criterion can be obtained, which is very practical when trying to compute derivatives.

### Example 6: estimation of the friction parameter

In our wave example we can try to estimate the friction parameter  $c_f$  of equation 2.7 by minimizing a likelihood function. If we assume that the uncertainty in this parameter dominates the model accuracy, which is of course not the case for a storm surge, but if we consider only the astronomic tide, i.e. the forcing  $\xi_b(\cdot)$  (eqn. 2.8) is taken as the astronomic tide at the open boundary and the measurements are taken during calm weather then the friction at the bottom is one of the dominant unknowns. The remaining difference between model predictions and the measurements, for the correct friction, are assumed to be measurement errors. Under these assumptions minus the logarithm of the likelihood function becomes

$$J = -\log(f(y(1), \dots, y(N)|c_f)) \quad (2.36)$$

$$= \sum_{k=1}^N (y(k) - Cx(k))' \Sigma_o^{-1} (y(k) - Cx(k)) + \chi \quad (2.37)$$

where  $\chi$  is a term that does not depend on  $c_f$  and is therefore not important in the optimization. The maximization of the likelihood function changes into a minimization of the criterion  $J$  through the minus in front of the logarithm. The best value for the friction parameter  $c_f$  can now be found by the minimizing  $J$  under the constraints imposed by the model.

There are two main methods for minimization of the criterion. The first is to use general numerical optimization procedures like steepest descent and the BFGS algorithm [31]. Many efficient numerical optimization algorithms use information about the derivatives of the criterion with respect to the control variables. An efficient way of computing these derivatives is with the use of an adjoint model. This method was first introduced for use with models described by partial differential equations [14].

The second class of optimization methods minimizes the criterion by equating the analytical derivatives of the criterion with respect to the uncertain parameters to zero. The resulting set of equations is often called the 'Euler-Lagrange' set of equations. For linear systems and a quadratic criterion this results in a set of linear equations. The next step is to solve this set of equations. Special solution methods for these equations will be needed because for practical applications the number of equations will be huge. For some special cases the number of equations can be reduced. For example, if there are relatively few measurements the number of equations can be reduced drastically using the

representer method [3].

Recently Evensen and Van Leeuwen proposed a direct ensemble smoother [71]. This algorithm illustrates that the gap between probabilistic methods and criterion based methods is closing. Their criterion is based on a likelihood function and they use Monte Carlo simulation for the optimization.

# Chapter 3

## Kalman filtering algorithms

### 3.1 Introduction

When a stochastic description of model and measurements of the form

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{F}(k)\mathbf{w}(k) \quad (3.1)$$

$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{v}(k) \quad (3.2)$$

is available it is possible to incorporate the measurements in to the model to obtain an optimal estimate of the state. For this linear model with Gaussian noise Kalman and Bucy showed that a recursive update of the estimate can be found [47, 48]. Under these assumptions the estimate is optimal for several criteria, such as minimum variance and maximum likelihood. For the estimate of  $\mathbf{x}(k)$  given  $\tilde{\mathbf{y}}(k) = \{\mathbf{y}(l), l = 0, \dots, k\}$  denoted by  $\hat{\mathbf{x}}(k|k)$  the Kalman filter equations are given by:

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}(k)\hat{\mathbf{x}}(k|k) + \mathbf{B}(k)\mathbf{u}(k) \quad (3.3)$$

$$\mathbf{P}(k+1|k) = \mathbf{A}(k)\mathbf{P}(k|k)\mathbf{A}(k)' + \mathbf{F}(k)\Sigma_s(k)\mathbf{F}(k)' \quad (3.4)$$

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) \\ &\quad + \mathbf{K}(k+1)(\mathbf{y}(k+1) - \mathbf{C}(k+1)\hat{\mathbf{x}}(k+1|k)) \end{aligned} \quad (3.5)$$

$$\begin{aligned} \mathbf{K}(k+1) &= \mathbf{P}(k+1|k)\mathbf{C}(k+1)' \\ &\quad (\mathbf{C}(k+1)\mathbf{P}(k+1|k)\mathbf{C}(k+1)' + \Sigma_o(k+1))^{-1} \end{aligned} \quad (3.6)$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{K}(k+1)\mathbf{C}(k+1)\mathbf{P}(k+1|k) \quad (3.7)$$

$$\hat{\mathbf{x}}(0|-1) = \mathbf{x}_0 \quad (3.8)$$

$$\mathbf{P}(0|-1) = \mathbf{P}_0 \quad (3.9)$$

Although these Kalman filter equations can in principle be used to solve many data assimilation problems the actual implementation for real life applic-

ations is far from trivial. It is now well known that the Kalman filter equations can be very sensitive to numerical errors. Many algebraically equivalent but numerically more stable equations were derived to attack this problem [54]. Among these are the algorithms based on square root factorizations of the error covariance  $\mathbf{P}(\cdot)$  [54, 5].

A second, perhaps more serious, difficulty is that the number of computations increases rapidly with the state dimension  $n$ . With a typical state dimension of  $n = 60000$  for WAQUA models the computation would take far too much time.

A third difficulty is the limitation to linear models. Many models are non-linear. Although the Kalman filter can be extended for non-linear models the computation times needed for these algorithms are not feasible except for very small problems [44, 56]. Therefore many approximate non-linear filters were developed.

In this chapter some of the basic Kalman filter theory as well as some of the computational issues are addressed. The huge computational burden for large models is not discussed here, but in the next chapter.

## 3.2 Steady state filter

It can be seen from the structure of equations 3.3–3.9, that  $\mathbf{K}(k)$  does not depend on the measurements  $\mathbf{y}$  and therefore can be computed in advance and stored. During the actual filtering the stored values can then be used. If the model is time invariant and stable, it can be shown that the Kalman gain  $\mathbf{K}(k)$  converges to a limit value  $\mathbf{K}$ . When this steady state Kalman gain is used for all measurement times the estimate converges to the optimal estimate for large times  $k$  (see [52] for the details).

Convergence to a steady state can also be shown if the model is linear, time invariant,  $(\mathbf{A}, \mathbf{F}\Sigma_s^{\frac{1}{2}})$  is controllable, i.e.

$$\text{rank}(\mathcal{C}) = \text{rank}\left(\sum_{k=0}^n \mathbf{A}^k \mathbf{F} \Sigma_s \mathbf{F}' (\mathbf{A}^k)'\right) = n \quad (3.10)$$

and  $(\mathbf{C}, \mathbf{A})$  is observable

$$\text{rank}(\mathcal{O}) = \text{rank}\left(\sum_{k=0}^n (\mathbf{A}^k)' \mathbf{C} \mathbf{C}' (\mathbf{A}^k)\right) = n \quad (3.11)$$

Unfortunately, a system that originates from the discretization of a set of partial differential equations in more than one dimension is seldom observable with a small number of measurements. In practice it is thus often difficult or impossible to prove convergence to a steady state in advance. Since the conditions mentioned above are only sufficient this does not imply that the Kalman filter will not converge. Often the convergence is examined by implementing the Kalman filter, possibly for a simplified model, and verify convergence by inspection of the results. Observability and controllability can also be studied in continuous space [20], but when discretizing the model observability can easily be lost [17].

To compute the steady state Kalman gain also the more efficient Chandrasekhar type algorithm [59] or doubling algorithm [1] can be used instead of the equations 3.3–3.9. Both algorithms can reduce computation times considerably. The Chandrasekhar type filter is based on a recursion for  $\Delta \mathbf{P}(k) := \mathbf{P}(k|k) - \mathbf{P}(k-1|k-1)$ . The advantage is that for a time invariant model with  $\mathbf{P}_0 = 0$  the rank of these matrices is  $m$ . The doubling algorithm is based on a different principle: it performs steps from time  $k$  to  $2k$  instead of from  $k$  to  $k+1$ . Several other algorithms have been proposed for the same purpose [46, 59, 58, 61].

The steady state approach has been used successfully for large, two dimensional, models (e.g. [40, 7, 33]). Compared to a more traditional prediction by a deterministic model only, the number of additional computations is small while the reduction in errors can be large. A disadvantage is that the steady state approach can not be used for many applications because in many applications the model is not nearly linear or the measurements are irregular in time or space.

### 3.3 Filtering, forecasting and smoothing

So far, only the filtering problem has been discussed, in which the estimate is based on the measurements until that time, but depending on the application it may be necessary to make forecasts or include measurements that were taken past the time of the estimate. We distinguish three different cases. If the estimate (of  $\mathbf{x}$ ) at time  $k$  is needed based on measurements until time  $l$ , i.e.  $\{\mathbf{y}(0), \mathbf{y}(1), \dots, \mathbf{y}(l)\}$ , then there the possibilities are:

- $k > l$  forecasting or prediction
- $k = l$  filtering or now-casting
- $k < l$  smoothing or hind-casting

Starting from the filter estimate the forecasting problem is easiest to solve. Similar to equations 3.3 and 3.4 one can prove for  $k \geq l$  (see e.g. [54, 44])

$$\hat{\mathbf{x}}(k+1|l) = \mathbf{A}(k)\hat{\mathbf{x}}(k|l) + \mathbf{B}(k)\mathbf{u}(k) \quad (3.12)$$

$$\mathbf{P}(k+1|l) = \mathbf{A}(k)\mathbf{P}(k|l)\mathbf{A}(k)' + \mathbf{F}(k)\Sigma_s(k)\mathbf{F}(k)' \quad (3.13)$$

$$(3.14)$$

The equation for the estimate and the covariance are decoupled. If only the forecast is needed it is not necessary to compute the error covariance.

The smoothing problem is more difficult. Many algorithms have been proposed for this purpose (see e.g. [55, 5, 58]). These algorithms can be divided in three different types:

- fixed point smoothing, i.e. compute  $\hat{\mathbf{x}}(k|l)$  for fixed  $l$  and  $l = k+1, k+2, \dots$
- fixed lag smoothing, i.e. compute  $\hat{\mathbf{x}}(k+l, k)$  for fixed  $l$  and  $k = 0, 1, 2, \dots$
- fixed interval smoothing, i.e. compute  $\hat{\mathbf{x}}(k|l)$  for fixed  $l$  and  $k = 0, 1, 2, \dots, l$

Smoothing algorithms also differ in the storage requirements, and the use of the inverse model ( $\mathbf{A}(k)^{-1}$ ) or the adjoint ( $\mathbf{A}(k)'$ ).

One conceptually not very difficult way to study the smoothing problem is by *augmenting the state*. For example a fixed lag smoother can be derived from the Kalman filter by use of the following augmented state

$$\mathbf{x}_a(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}(k-1) \\ \vdots \\ \mathbf{x}(k-l) \end{bmatrix} \quad (3.15)$$

It is possible to define an augmented system for this state. Using the Kalman filter is then possible to compute  $\hat{\mathbf{x}}_a(k|k)$ , by definition this is

$$\hat{\mathbf{x}}_a(k|k) = \begin{bmatrix} \hat{\mathbf{x}}(k|k) \\ \hat{\mathbf{x}}(k-1|k) \\ \vdots \\ \hat{\mathbf{x}}(k-l|k) \end{bmatrix} \quad (3.16)$$

which solves the fixed lag smoothing problem. The approach of augmenting the state to solve smoothing problems has several advantages. Firstly the approach is very flexible, i.e. a large variety of smoothing problems can be solved in this manner. The approach can be extended to non-linear models without difficulty, provided a nonlinear 'Kalman' filter is available of course. More important is that the RRSQRT algorithm proposed in the sequel can be used in combination with this approach in a straightforward manner. In addition no adjoint model or (pseudo) inverse model is needed contrary to many other smoothing algorithms.

### 3.4 Extensions for non-linear models

The original Kalman filter was derived for a linear model. Unfortunately, models are often not linear in practice and some extension of the Kalman filter is needed, which severely complicates the state estimation.

The main advantage of using a linear model driven by Gaussian noise is that the state and output will also be Gaussian. Furthermore a Gaussian distribution can be characterized by its mean and covariance only. The Kalman filter therefore has only first moment (mean) and second moment (error covariance) equations.

If the model is nonlinear the optimal minimum variance estimate, the conditional mean, can no longer be computed from the first and second moment at the previous time-step, but an infinite number of higher order moments enters the equations. Although the nonlinear state estimation problem can formally be solved using the forward Kolmogorov equation (see e.g. [44], Kushner [50, 51, 49]) practical numerical solution of these equations is only possible

for very small systems ( $n < 4$ ). At the moment these 'exact' methods are considered only as a reference solution for the study of data assimilation with highly nonlinear models [56, 13].

Several approximate solutions were developed which truncate moments of higher order. Examples of such truncated nonlinear filters are the very popular extended Kalman filter and the truncated second order filter. Although such methods perform a rather crude truncation many successful applications have been reported. Some examples of successful application of the extended Kalman filter to data assimilation are Heemink [42], Budgell [9], Boggs [6].

The extended Kalman filter is the simplest version of the truncated moment filters. Of the infinite number of moments that describe the conditional probability only the mean and error covariance are computed. The model is linearized around the most recent estimate. For the system given by equations 2.28 and 2.29 the extended Kalman filter is given by

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{f}(\hat{\mathbf{x}}(k|k), \mathbf{u}(k), \mathbf{w}(k) = 0) \quad (3.17)$$

$$\mathbf{P}_c(k+1|k) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(k) \mathbf{P}_c(k|k) \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(k)' + \frac{\partial \mathbf{f}}{\partial \mathbf{w}}(k) \Sigma_s(k) \frac{\partial \mathbf{f}}{\partial \mathbf{w}}(k)' \quad (3.18)$$

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) + \mathbf{K}_c(k+1) \\ &\quad (\mathbf{y}(k+1) - \mathbf{g}(\hat{\mathbf{x}}(k+1|k), \mathbf{w}(k) = 0)) \end{aligned} \quad (3.19)$$

$$\begin{aligned} \mathbf{K}_c(k+1) &= \mathbf{P}_c(k+1|k) \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(k+1)' \\ &\quad \left( \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(k+1) \mathbf{P}_c(k+1|k) \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(k+1)' + \right. \\ &\quad \left. \frac{\partial \mathbf{g}}{\partial \mathbf{v}}(k+1) \Sigma_o(k+1) \frac{\partial \mathbf{g}}{\partial \mathbf{v}}(k+1)' \right)^{-1} \end{aligned} \quad (3.20)$$

$$\mathbf{P}_c(k+1|k+1) = \mathbf{P}_c(k+1|k) - \mathbf{K}_c(k+1) \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(k+1) \mathbf{P}_c(k+1|k) \quad (3.21)$$

$$\hat{\mathbf{x}}(0|-1) = \mathbf{x}_0 \quad (3.22)$$

$$\mathbf{P}_c(0|-1) = \mathbf{P}_0 \quad (3.23)$$

where the first order derivatives  $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(k)$ ,  $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}(k)$  and  $\frac{\partial \mathbf{f}}{\partial \mathbf{w}}(k)$  are evaluated for  $\mathbf{x}(k) = \hat{\mathbf{x}}(k|k)$ ,  $\mathbf{u}(k)$  as specified and  $\mathbf{w}(k) = 0$ . The derivatives  $\frac{\partial \mathbf{g}}{\partial \mathbf{x}}(k+1)$  and  $\frac{\partial \mathbf{g}}{\partial \mathbf{v}}(k+1)$  are evaluated for  $\mathbf{x}(k) = \hat{\mathbf{x}}(k+1|k)$  and  $\mathbf{v}(k+1) = 0$ . Note that these equation are very similar to the 'standard' Kalman filter with the model linearized around the most recent estimate. Various modifications to this filter have been proposed (see e.g. Jazwinsky 1970 [44]). For many applications the linearization of the extended Kalman filter is sufficient, even for very non-linear models [56, 35]. However, when the time between measurements increases or the system noise is very large relative to distance from the estimate over which the non-linearities become significant the method may fail. In these cases it may be worthwhile to use an higher order truncation.

The second-order truncated filter includes the bias in the time-propagation caused by the second-order curvature of the system dynamics. Note that the modified truncated second order filter in for example Maybeck [55] and Jazwinsky [44] is in fact the only correct form of the truncated second-order filter (see Henriksen 1980 [43]). The second-order truncated filter is given by

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{f}(\hat{\mathbf{x}}(k|k), \mathbf{u}(k), \mathbf{w}(k) = 0) + \frac{1}{2} \mathbf{P}_c(k|k) \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2} \quad (3.24)$$

$$\begin{aligned} \mathbf{P}_c(k+1|k) &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(k) \mathbf{P}_c(k|k) \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(k)' + \frac{\partial \mathbf{f}}{\partial \mathbf{w}}(k) \Sigma_s(k) \frac{\partial \mathbf{f}}{\partial \mathbf{w}}(k)' \\ &+ (\mathbf{P}_c(k|k) \Sigma_s(k) \left( \frac{\partial^2 \mathbf{f}}{\partial \mathbf{w} \partial \mathbf{x}} \right)^2) \\ &+ \frac{1}{2} (\mathbf{P}_c(k|k) \Sigma_s \frac{\partial^3 \mathbf{f}}{\partial \mathbf{x}^2 \partial \mathbf{w} \partial \mathbf{w}} \frac{\partial \mathbf{f}}{\partial \mathbf{w}}) \\ &+ \frac{1}{2} (\mathbf{P}_c(k|k) \Sigma_s \frac{\partial^3 \mathbf{f}}{\partial \mathbf{x}^2 \partial \mathbf{w} \partial \mathbf{w}} \frac{\partial \mathbf{f}}{\partial \mathbf{w}})' \end{aligned} \quad (3.25)$$

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) + \mathbf{K}_c(k+1) \\ &(\mathbf{y}(k+1) - \mathbf{g}(\hat{\mathbf{x}}(k+1|k), \mathbf{w}(k) = 0) \\ &- \frac{1}{2} (\mathbf{P}_c(k+1|k) \frac{\partial^2 \mathbf{g}}{\partial \mathbf{x}^2}) \end{aligned} \quad (3.26)$$

$$\begin{aligned} \mathbf{K}_c(k+1) &= \mathbf{P}_c(k+1|k) \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(k+1)' \\ &\left( \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(k+1) \mathbf{P}_c(k+1|k) \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(k+1)' + \right. \\ &\left. \frac{\partial \mathbf{g}}{\partial \mathbf{v}}(k+1) \Sigma_o(k+1) \frac{\partial \mathbf{g}}{\partial \mathbf{v}}(k+1) \right)^{-1} \end{aligned} \quad (3.27)$$

$$\mathbf{P}_c(k+1|k+1) = \mathbf{P}_c(k+1|k) - \mathbf{K}_c(k+1) \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(k+1) \mathbf{P}_c(k+1|k) \quad (3.28)$$

$$\hat{\mathbf{x}}(0|-1) = \mathbf{x}_0 \quad (3.29)$$

$$\mathbf{P}_c(0|-1) = \mathbf{P}_0 \quad (3.30)$$

where  $\mathbf{P}_c(k|k) \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2}$  denotes a double summation over the elements of the state vector

$$[\mathbf{P}_c(k|k) \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2}]_i := \sum_{j,l} [\mathbf{P}_c(k|k)]_{j,l} \frac{\partial^2 \mathbf{f}_i}{\partial \mathbf{x}_j \partial \mathbf{x}_l} \quad (3.31)$$

For simplicity it was assumed that  $\frac{\partial^2 \mathbf{f}}{\partial \mathbf{w}^2} \equiv 0$  and  $\frac{\partial^2 \mathbf{g}}{\partial \mathbf{v}^2} \equiv 0$ , thus also  $\frac{\partial^3 \mathbf{f}}{\partial \mathbf{w}^2 \partial \mathbf{x}} = 0$ , etc.

The second-order filters require higher order derivatives of  $\mathbf{f}$  and  $\mathbf{g}$ . If the state dimension  $n$  is large the computation of these terms quickly becomes infeasible. A rather efficient method to avoid computation of higher order derivatives while still attaining second-order accuracy was proposed by Julier [45]. In this approach is based on the fact that when a finite sample has exactly the

same sample mean as the mean and the same sample covariance as the covariance then the sample mean and sample covariance of the propagated sample is second-order accurate. The finite sample in this case can easily be found using the square root of the covariance matrix.

Truncated moment Kalman filters with order higher than two are more sparse. Kushner [50] derived a fourth order filter. In some cases this gives much better results than the extended Kalman filter [56]. A disadvantage is that the forth-order central moments used in this filter grow in number with the forth-power  $n^4$  of the state dimension. This is clearly infeasible for large systems.

Another method of propagating information about the conditional probability density uses Monte Carlo simulation. Several algorithms have been proposed that are based on this principle [30, 11, 56]. The Monte Carlo approach at the moment is the most promising option for extremely non-linear systems with a large state dimension.

### 3.5 Numerical aspects of Kalman filter algorithms

Although the Kalman filter provides the optimal estimate of the state given the measurements the equations 3.3–3.9 are not a good choice from a numerical point of view. For example the error covariance matrix is by definition symmetric this is not necessarily so for its numerical counterpart. The main cause for this is the asymmetric form of equation 3.6. Another difficulty is that when accurate measurements are used the condition number of the error covariance matrix often becomes very large. This often leads to the occurrence of negative eigenvalues of the error covariance matrix, which in turn may cause instability of the filter algorithm. To avoid these and similar problems many algebraically equivalent but numerically more robust formulations have been proposed.

Van Dooren and Verhagen [29, 75] showed that by making the error covariance symmetric after every time-step the algorithm becomes much more stable while the number of additional computations is small.

Most methods to handle the large condition number of the error covariance matrix and avoid negative eigenvalues often use a factorization of the error covariance matrix. Oshman and Bar-Itzhack [60] proposed an algorithm based on a singular value decomposition of the error covariance matrix. Both the time propagation and the measurement step are performed with a singular value decomposition. An advantage of this approach is that the condition number can easily be monitored and robust algorithms are known to perform the singular value decomposition.

Several algorithms use an eigendecomposition. Van Dooren [28] proposes a method for solution of the algebraic Riccati equation (e.g. the steady state Kalman filter) based on an eigendecomposition. For the continuous time Kalman filter Oshman and Bar-Itzhack [?] show an algorithm that uses the eigendecomposition of the error covariance matrix. A remarkable property of this algorithm is that in principle the eigendecomposition is tracked in time directly, i.e. no

actual eigendecomposition is computed except at the initial time.

In addition to eigendecomposition of the error covariance matrix several other decompositions of the error covariance have been proposed, most of the form  $\mathbf{P} = \mathbf{L}\mathbf{L}'$  or  $\mathbf{P} = \mathbf{L}\mathbf{D}\mathbf{L}'$ . An advantage of decompositions of this type is that the error covariance matrix remains implicitly symmetric and positive semi definite which enhances the stability of these algorithms. Moreover, the condition number of the factor  $\mathbf{L}$  in the decomposition  $\mathbf{P} = \mathbf{L}\mathbf{L}'$  is only the square root of the condition number of  $\mathbf{P}$ .

Several well known algorithms exist where  $\mathbf{L}$  is either lower-triangular or upper-triangular. These *square root filtering algorithms* are known to be more robust than the original Kalman filtering algorithm. The interested reader is referred to Maybeck [54] for an introduction and overview of basic square root algorithms. If part of the state can not be observed accurately it may be worthwhile to use an information type square root filter where  $\mathbf{P}^{-1}$  is decomposed rather than  $\mathbf{P}$  [5].

It is also possible to combine a square root decomposition with the Chandrasekhar approach used for steady state filters [58, 61]. In that case the factorization is applied to  $\Delta\mathbf{P}(k) := \mathbf{P}(k+1|k) - \mathbf{P}(k|k-1)$ .

### 3.6 Square root filtering

As mentioned above square root filtering algorithms provide an algebraically equivalent but numerically more robust alternative for the traditional Kalman filter equations (3.3–3.9). These algorithms, which were first introduced by Potter [?] avoid numerical problems by using the square root of the error covariance matrix,  $\mathbf{P}(k|l) = \mathbf{L}(k|l)\mathbf{L}(k|l)'$ , with  $\mathbf{L}(k|l)$  a lower triangular matrix. All computations in this algorithm work with the factor  $\mathbf{L}(k|l)$  directly and the error covariance matrix  $\mathbf{P}(k|l)$  is never computed explicitly. The factors  $\mathbf{L}(k|l)$  have a much smaller range of the eigenvalues these algorithms and are therefore numerically better conditioned than the original Kalman filter algorithm. Because of the decomposition the error covariance matrix remains implicitly symmetric and positive semi definite. For computational convenience the non-unique square root factor  $L(k|l)$  is kept in a lower triangular form. A introduction can be found in [54, 5].

Most square root algorithms have the same predictor-corrector structure as the original Kalman filter. The following example of a square root algorithm was taken from Maybeck 1979 [54] p. 374 and 379. The time update of the  $\mathbf{L}$  matrix is given by

$$\mathbf{L}(k+1|k) = [\mathbf{A}(k)\mathbf{L}(k|k), \mathbf{F}(k)\Sigma_s^{1/2}]\mathbf{U}(k) \quad (3.32)$$

where  $\mathbf{U}(k)$  is a unitary matrix ( $\mathbf{U}(k)\mathbf{U}(k)' = \mathbf{U}(k)'\mathbf{U}(k) = \mathbf{I}$ ) such that the last  $m$  rows of the first factor on the right hand side become zero. Usually Householder reflections or the Modified Gramm Schmidt procedure is used for this. It is easily shown that the multiplication with a unitary  $\mathbf{U}(k)$  does not change  $\mathbf{P}(= \mathbf{L}\mathbf{L}')$ .  $\Sigma_s^{1/2}$  is a square root factor of  $\Sigma_s$ , i.e.  $\Sigma_s^{1/2}(\Sigma_s^{1/2})' = \Sigma_s$ .

The notation  $[,]$  means that a larger block matrix is built from the two sub-matrices  $\mathbf{A}(k)\mathbf{L}(k|k)$  and  $\mathbf{F}(k)\Sigma_s^{1/2}$  by putting the columns next to each other.

The measurement update for scalar measurements  $p = 1$  is given by

$$\mathbf{H}(k+1) = \mathbf{L}(k+1|k)' \mathbf{C}(k+1)' \quad (3.33)$$

$$\beta(k+1) = (\mathbf{H}(k+1)'\mathbf{H}(k+1) + \Sigma_o(k+1))^{-1} \quad (3.34)$$

$$\mathbf{K}(k+1) = \mathbf{L}(k+1|k)\mathbf{H}(k+1)\beta(k+1) \quad (3.35)$$

$$\begin{aligned} \mathbf{L}(k+1|k+1) &= \mathbf{L}(k+1|k) - \mathbf{K}(k+1)\mathbf{H}(k+1)' \\ &\quad (1 + (\beta(k+1)\Sigma_o(k+1))^{1/2})^{-1} \end{aligned} \quad (3.36)$$

If there is more than one uncorrelated measurement at one time then the measurements are processed on at a time using equations 3.33– 3.36. If the measurements are not uncorrelated they are transformed using  $\Sigma_o^{\frac{1}{2}}$  (will be described more elaborately later). The resulting independent measurements can be processed one at a time.

Although square root algorithms are more robust than the standard Kalman filter they are in general not more efficient than the standard Kalman filter algorithm and therefore the square root filter equations described above can not be used directly for large scale models.

## Chapter 4

# Suboptimal schemes for large scale models

### 4.1 Introduction

Although in principle the Kalman filter provides a solution to many data assimilation problems, the computational burden quickly becomes very large for large models. In fact, for most data assimilation problems with a model that is derived from a two dimensional or three dimensional partial differential equation, direct computation of the full Kalman filter equations 3.3, 3.7 is infeasible. In addition to the computation time also the size of the covariance matrix is problematic for large models.

In most of the applications mentioned above the time update of the error covariance matrix (eqn. 3.4) is the most problematic part. When rewriting equation 3.4 as

$$\mathbf{P}(k+1|k) = \mathbf{A}(k)\mathbf{P}(k|k)\mathbf{A}(k)' + \mathbf{F}(k)\Sigma_s(k)\mathbf{F}(k)' \quad (4.1)$$

$$= \mathbf{A}(k)(\mathbf{A}(k)\mathbf{P}(k|k))' + \mathbf{F}(k)\Sigma_s(k)\mathbf{F}(k)' \quad (4.2)$$

one can see that this equation requires essentially  $2n$  model evaluations since every multiplication of  $\mathbf{A}(k)$  with a vector is equivalent to one model evaluation (using eqn. 2.30 with  $\mathbf{w}(k) = 0$ ,  $\mathbf{u}(k) = 0$  and with the vector  $\mathbf{x}(k)$ ). First the columns of  $\mathbf{P}(k|k)$  are multiplied with  $\mathbf{A}(k)$  and subsequently the rows of  $\mathbf{A}(k)\mathbf{P}(k|k)$  are multiplied with  $\mathbf{A}(k)$ . Most numerical schemes are designed such that one model evaluation requires  $O(n)$  operations, where  $n$  is the state dimension. Thus the number of operations needed for a time update of the error covariance matrix is of order  $O(n^2)$ . It is important to note that this is one order smaller than if the matrix  $\mathbf{A}(k)$  were full ( $O(n^3)$ ).

For the two dimensional shallow water flow model that will be used in one of the applications (see chapter 6), the evaluation of one time-step takes approximately 4 seconds on a workstation. The model has about 20,000 grid-points with

3 variables at each grid-point. This results in a state dimension of  $n = 60,000$ . Thus one evaluation of equation 3.4 would take 480,000 seconds, which is approximately 5.5 days. For the simulation of a storm surge at least several days must be simulated. The time steps in this model are 10 minutes, resulting in 144 steps per day of simulation, which amounts to several years of computation time for one storm surge. Besides, the error covariance matrix  $\mathbf{P}(k|k)$  in this example contains  $n^2 = 3.6 \cdot 10^9$  elements. When this matrix is stored in the memory of a computer using 4 bytes per element approximately 14.4 Gigabytes of memory is needed. Clearly direct use of the equations 3.3-3.7 is not feasible for this application.

It is obvious that more efficient approximate schemes are needed. This observation has lead to the search for efficient approximate methods. After Todling and Cohn [16] we will refer to these approximate algorithms as Suboptimal Schemes or SOS's. By definition the estimate computed by these SOS's will (in average) be worse than the estimates of the 'full' Kalman filter. In principle the performance of the SOS's can be evaluated by comparing their estimates and the number of computations needed with those of the optimal 'full' Kalman filter.

In the 'full' Kalman filter the error covariance computed by the algorithm is, if the specification of noise in the model is correct, equal to the true error covariance of the estimate. When approximations are introduced in a SOS this is no longer true. There are now three different covariance matrices:  $\mathbf{P}_c(\cdot|\cdot)$  the error covariance computed by the algorithm,  $\mathbf{P}(\cdot|\cdot)$  the optimal error covariance matrix and  $\mathbf{P}_t(\cdot|\cdot)$  the true error covariance matrix. The true error covariance behaves according to (Jazwinski [44] p. 248):

$$\mathbf{P}_t(k+1|k) = \mathbf{A}(k)\mathbf{P}_t(k|k)\mathbf{A}(k)' + \mathbf{F}(k)\Sigma_s(k)\mathbf{F}(k)' \quad (4.3)$$

$$\begin{aligned} \mathbf{P}_t(k+1|k+1) &= [\mathbf{I} - \mathbf{K}_c(k+1)\mathbf{C}(k+1)]\mathbf{P}_t(k+1|k) \\ &\quad [\mathbf{I} - \mathbf{K}_c(k+1)\mathbf{C}(k+1)]' \\ &\quad + \mathbf{K}_c(k+1)\Sigma_o(k)\mathbf{K}_c(k+1)' \end{aligned} \quad (4.4)$$

where  $\mathbf{K}_c$  is the Kalman gain computed from the approximate error covariance  $\mathbf{P}_c$ .

The aim of a SOS in this context is to minimize  $\mathbf{P}_t(\cdot|\cdot)$  while at the same time keeping the cost of computing  $\mathbf{K}_c$  as low as possible. For linear models the Kalman gain  $\mathbf{K}$  (eqn. 3.6) is optimal in the sense that

$$\mathbf{P}(k|k) \leq \mathbf{P}_t(k|k) \quad (4.5)$$

for all  $\mathbf{K}_c$ . The inequality  $\mathbf{P}_a \geq \mathbf{P}_b$  denotes that  $\forall_{\mathbf{x} \neq 0} \mathbf{x}'(\mathbf{P}_a - \mathbf{P}_b)\mathbf{x} \geq 0$ . Thus an approximation in the computation of the gain can never improve the resulting estimate. One can only try to keep the loss  $\mathbf{P}_t(\cdot|\cdot) - \mathbf{P}(\cdot|\cdot)$  as small as possible. Since this value is a matrix the more specific loss function  $\text{trace}(\mathbf{P}_t(\cdot|\cdot) - \mathbf{P}(\cdot|\cdot))$  will be used in the sequel. But instead of giving equal weight to all variables in the state vector it may be preferable to use a different function of  $\mathbf{P}_t(\cdot|\cdot) - \mathbf{P}(\cdot|\cdot)$ . If the approximation of the Kalman filter is good enough the difference

$(\mathbf{P}_t(\cdot|\cdot) - \mathbf{P}(\cdot|\cdot))$  will often converge. However for inaccurate approximations the difference may grow without bound, a phenomenon that is known as filter divergence.

An additional difficulty is that the error covariance matrix  $\mathbf{P}(\cdot|\cdot)$  often has a very large condition number for the type of applications studied here. On the other hand without any truncation errors the error covariance matrix in the Kalman filter always remains positive semidefinite. If due to approximation the computed error covariance contains negative eigenvalues, this can easily lead to filter divergence.

For linear models the Kalman filter computes the optimal estimate and thus all approximations are necessarily suboptimal. For nonlinear models this is no longer true. Experience so far however indicates that the extended Kalman filter often gives very good results the type of nonlinear models studied here. It is assumed that methods that approximate the Kalman filter efficiently and accurately will provide good results when used with nonlinear models as well. Using this approach SOS's will first be developed and studied in a linear context and after that applied to nonlinear models. It is important to note that for some SOS's the performance may be different than expected from comparisons in the linear context. For example the Ensemble Kalman filter [30, 11] may perform better than the extended Kalman filter if there are nonlinearities present.

In this chapter a new SOS, the Reduced Rank Square Root algorithm, is proposed and compared with several other SOS's. Treatment of the algorithmic details and theoretical results for this algorithm is deferred to the next chapter. Here only the main principles are discussed shortly in a framework that facilitates comparison between the SOS's. The example of the one dimensional wave model will be used to illustrate the performance of the SOS's. Since this is a 'small' ( $n = 81$ ) linear model the true performance  $(\mathbf{P}_t(\cdot|\cdot))$  can still be evaluated, which is not possible for applications of a realistic size. It is however expected that the various SOS's show a similar behavior of the convergence for larger models. For these large models and for non-linear models the analysis of the performance is limited to comparison of various SOS's, both in twin experiments and on real data [22].

## 4.2 A comparison of several SOS's

### 4.2.1 The model used for comparison

The comparison of the SOS's in this chapter again makes use of the model used in the examples. The main features of the model are summarized below.

### Example 1: 1-D wave model summarized

The set-up or surge in an estuary can be modeled with the simplified De St. Venant equations:

$$\frac{\partial \xi}{\partial t} + D \frac{\partial u}{\partial x} = 0 \quad (4.6)$$

$$\frac{\partial u}{\partial t} + g \frac{\partial \xi}{\partial x} + c_f u = 0 \quad (4.7)$$

$$\xi(x=0, t) = \xi_b(t) \quad (4.8)$$

$$\xi(x, t=0) = 0 \quad (4.9)$$

$$u(x, t=0) = 0 \quad (4.10)$$

$$u(x=L, t) = 0 \quad (4.11)$$

where  $\xi(x, t)$  is the part of water level associated with the storm surge,  $u(x, t)$  is the corresponding average current velocity,  $t$  is the time,  $x$  is the position along the estuary,  $D$  is the water depth,  $g$  is the gravitational acceleration and  $c_f$  is a friction constant (see also figure 4.1). In this model storm surges enter the model through the sea-ward boundary.

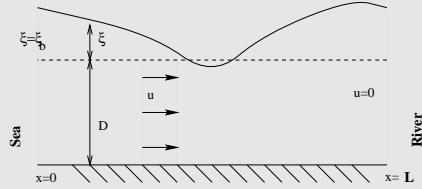


Figure 4.1: An estuary with the model variables

#### 4.2.2 The RRSQRT filter and Eigendecomposition filter

The main principle of the RRSQRT filter is to approximate the error covariance matrix by a matrix with a smaller rank. The approximated error covariance matrix is found by a truncated eigendecomposition. It is well known that the optimal rank  $q$  approximation, both in the spectral norm and the Frobenius norm, of a positive semi-definite symmetric matrix is given by a projection onto the  $q$  leading eigenvectors (see e.g. [37]). Therefore this is locally the optimal projection. The smaller rank of the approximate error covariance can be exploited to reduce both the computational burden of the Kalman filter and the memory requirements.

During the Second International Symposium on Assimilation of Observations in Meteorology and Oceanography in Tokyo 1995 two algorithms were presented independently that are based on an eigendecomposition of the error covariance

matrix. Todling and Cohn used the idea together with a Lanczos algorithm for the eigenvalue computations to obtain an efficient and general algorithm, the Partial Eigendecomposition Kalman Filter or PEKF([16]). They applied this SOS to stable and unstable meteorological models and tried several methods of compensating for the truncation. Verlaan and Heemink [76] also proposed an SOS based on the partial eigendecomposition of the error covariance. This Reduced Rank Square Root (RRSQRT) algorithm is however based on the more robust square root formulation of the Kalman filter. In both algorithms the efficiency is increased by storing the error covariance matrix relative to the space spanned by the leading eigenvectors. This way the computational burden is reduced to a fraction  $q/n$  of the full Kalman filter. Experience so far indicates that in many Kalman filtering applications there is only a relatively small number of large eigenvalues so that  $q$  can be taken much smaller than  $n$ .

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}(k)\hat{\mathbf{x}}(k|k) + \mathbf{B}(k)\mathbf{u}(k) \quad (4.12)$$

$$\begin{aligned} \mathbf{P}_c(k+1|k) &= \Pi(k+1)(\mathbf{A}(k)\mathbf{P}_c(k|k)\mathbf{A}(k)' + \mathbf{F}(k)\Sigma_s(k)\mathbf{F}(k)') \\ &\quad \Pi(k+1)' \end{aligned} \quad (4.13)$$

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) \\ &\quad + \mathbf{K}_c(k+1)(\mathbf{y}(k+1) - \mathbf{C}(k+1)\hat{\mathbf{x}}(k+1|k)) \end{aligned} \quad (4.14)$$

$$\begin{aligned} \mathbf{K}_c(k+1) &= \mathbf{P}_c(k+1|k)\mathbf{C}(k+1)' \\ &\quad (\mathbf{C}(k+1)\mathbf{P}_c(k+1|k)\mathbf{C}(k+1)' + \Sigma_o(k+1))^{-1} \end{aligned} \quad (4.15)$$

$$\mathbf{P}_c(k+1|k+1) = \mathbf{P}_c(k+1|k) - \mathbf{K}_c(k+1)\mathbf{C}(k+1)\mathbf{P}_c(k+1|k) \quad (4.16)$$

where  $\Pi(k+1)$  is a projection onto the  $q$  leading eigenvectors of  $\mathbf{A}(k)\mathbf{P}_c(k|k)\mathbf{A}(k)' + \mathbf{F}(k)\Sigma_s(k)\mathbf{F}(k)'$ . Let

$$\mathbf{A}(k)\mathbf{P}(k|k)\mathbf{A}(k)' + \mathbf{F}(k)\Sigma_s(k)\mathbf{F}(k)' = \mathbf{U}(k+1)\mathbf{D}(k+1)\mathbf{U}(k+1)' \quad (4.17)$$

be an eigendecomposition of this matrix, then

$$\Pi(K+1) = [\mathbf{U}(k+1)]_{:,1:q}'[\mathbf{U}(k+1)]_{:,1:q} \quad (4.18)$$

is a projection operator on the  $q$  leading eigenvectors. It is important to note that an efficient implementation of these equations does not use the full  $n$  by  $n$  error covariance matrix. Efficient implementation of the RRSQRT algorithm will be discussed in the sequel.

## Example 2: The RRSQRT filter for the 1-D wave model

The algorithm of equations 4.12– 4.16 can also be formulated in a numerically robust square root form. This RRSQRT algorithm will be discussed at length in the sequel. To assess the performance it was applied to the 1-D wave example. Figure 4.2 shows the computed and true RMS of the errors in the water level for 4 modes ( $q = 4$ ). For a larger number of modes  $q$  the filter quickly converges to the full Kalman filter. For the present parameter settings it was unstable for  $q \leq 3$ . For 4 modes the computational burden is approximately a fraction  $q/n = 4/81$  of the computation time for the full Kalman filter.

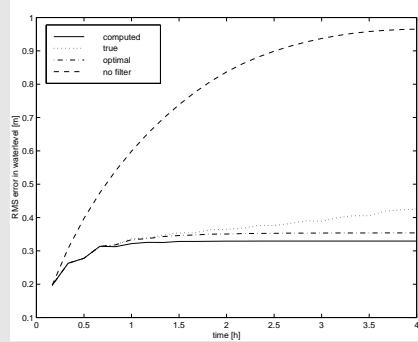


Figure 4.2: Root Mean Square error of the estimate of the water level; 4 modes

### 4.2.3 Steady state filter

One of the fastest SOS's known is the steady state Kalman filter. This approximation was proposed and analyzed already in the original papers of Kalman [47] and Kalman and Bucy [48]. The computation time of the steady state filter is only slightly larger than that for simulation alone. However the, so called, gain matrix has to be computed in advance. These off-line computations are in principle the same as those for the 'full' Kalman filter, but using the Chandrasekhar filter algorithm [59, 42, 7] this can be reduced significantly if the number of noise variables ( $m$ ) is small. The computation of the gain matrix has to be performed only once and only until the convergence is reached. It is also possible to speed up the gain computation for the steady state filter using a doubling algorithm [1] which was for instance applied by Fukumori [33].

The steady state filter is based on the fact that for linear time-invariant models with regular measurements, i.e. a time independent measurement equation

(eqn. 2.31), the Kalman gain may converge to a fixed value.

$$\lim_{k \rightarrow \infty} \mathbf{K}(k) = \mathbf{K} \quad (4.19)$$

Several sufficient conditions for convergence to a steady state have been discussed in chapter 3.

Once the steady state gain  $\mathbf{K}$  has been computed the steady state Kalman filter equation are given by

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}(k)\hat{\mathbf{x}}(k|k) + \mathbf{B}(k)\mathbf{u}(k) \quad (4.20)$$

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) \\ &\quad + \mathbf{K}(\mathbf{y}(k+1) - \mathbf{C}(k+1)\hat{\mathbf{x}}(k+1|k)) \end{aligned} \quad (4.21)$$

$$\hat{\mathbf{x}}(0|-1) = \mathbf{x}_0 \quad (4.22)$$

$$(4.23)$$

For models that are nearly linear and time invariant the steady state approximation can be quite accurate after the initial transient. There are however some severe limitations. It is not possible to include measurements that do not have a very regular pattern in space and time. For example, tide gauge measurements are performed at regular intervals and always at the same position, therefore they can be included, whereas altimeter measurements have a complicated pattern of tracks and are thus difficult to include using a steady state filter.

### Example 3: Steady state Kalman filter for 1-D wave model

The 1-D wave model used for comparison here is both linear, stable and time invariant. Because of this, a steady state approximation will be accurate after the initial transient. Figure 4.3 shows the Root Mean Square values of the water level for the computed error covariance, the true error covariance, the error covariance of the full Kalman filter and the true error covariance of a simulation without assimilation of measurements.

The initial covariance of the full Kalman filter and the true error covariance was zero, which is quite far from the steady state. Therefore there is a significant transient during which the steady state filter is clearly suboptimal. After some time the approximation becomes very accurate. The computation time of the steady state filter was very close to that of a simulation without Kalman filter.

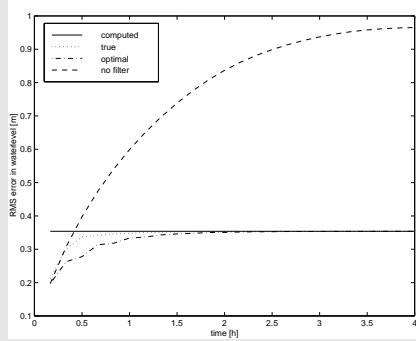


Figure 4.3: Root Mean Square error of the estimate of the water level

#### 4.2.4 Banded approximation of the covariance matrix

Since most of the computation time for the full Kalman filter is needed for the time propagation (eqn. 3.4) of the error covariance many SOS's are aimed at approximating the error covariance in such a way that this step can be performed more efficiently. An important additional advantage of these approximations is that the storage requirements are reduced at the same time.

Since in practice correlations are often limited to a certain spatial range it seems natural to discard correlations of variables with a spatial distance longer than some threshold. If this threshold is much smaller than the diameter of the model many zero entries are created in the error covariance matrix. Moreover, the non-zero elements are all within a few grid-cells distance from the central  $(i, i)$  element:  $[\mathbf{P}(\cdot, \cdot)]_{i,i}$ . This can be exploited to speed up the evaluation of

$\Psi(k+1) = \mathbf{A}(k)\mathbf{P}(k|k)$  and subsequently of  $\mathbf{A}(k)\Psi(k+1)'$ . For every column of  $\mathbf{P}(k|k)$  (or  $\Psi(k+1)$ ) only a part of the (tangent linear) model has to be evaluated. Cohn et al. used this approach for a 2-D meteorological model [62].

A major difficulty is that direct approximation of the error covariance matrix can lead to the occurrence of negative eigenvalues of the error covariance matrix, which often leads to instability of the filter algorithm. This can be avoided in a square root filter, since approximations of the square root factor can never cause eigenvalues of the error covariance matrix to become negative. Boggs et al [6] proposed an algorithm based on this approach. The algorithm is especially efficient for one dimensional models. For more than one dimension part of the sparseness is lost. This algorithm can, for  $p = 1$ , be denoted as

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}(k)\hat{\mathbf{x}}(k|k) + \mathbf{B}(k)\mathbf{u}(k) \quad (4.24)$$

$$\mathbf{L}_c(k+1|k) = \mathbf{T}_b([\mathbf{A}(k)\mathbf{L}_c(k|k), \mathbf{F}(k)\Sigma_s^{1/2}]\mathbf{U}(k)) \quad (4.25)$$

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) \\ &\quad + \mathbf{K}_c(k+1)(\mathbf{y}(k+1) - \mathbf{C}(k+1)\hat{\mathbf{x}}(k+1|k)) \end{aligned} \quad (4.26)$$

$$\mathbf{H}(k+1) = \mathbf{L}_c(k+1|k)'\mathbf{C}(k+1)' \quad (4.27)$$

$$\beta(k+1) = (\mathbf{H}(k+1)'\mathbf{H}(k+1) + \Sigma_o(k+1))^{-1} \quad (4.28)$$

$$\mathbf{K}_c(k+1) = \mathbf{L}_c(k+1|k)\mathbf{H}(k+1)\beta(k+1) \quad (4.29)$$

$$\begin{aligned} \mathbf{L}_c(k+1|k+1) &= \mathbf{L}_c(k+1|k) - \mathbf{K}_c(k+1)\mathbf{H}(k+1)' \\ &\quad (1 + (\beta(k+1)\Sigma_o(k+1))^{1/2})^{-1} \end{aligned} \quad (4.30)$$

where  $\mathbf{T}_b$  is a truncation operator that limits correlations to a distance  $b$ . For a band diagonal matrix  $\mathbf{L}_c$  the combination of the Householder reflections or Modified Gram-Schmidt ( $\mathbf{U}(k)$ ) and the truncation  $\mathbf{T}_b$  can be combined in an efficient manner [6].

#### Example 4: Square root band limited Kalman filter for 1-D wave model

The algorithm above was applied to the 1-D wave model. Correlations were truncated, in the square root of the error covariance, if the spatial distance was longer than a 'band width' parameter. The results showed a remarkable change in the performance when the band width was increased from 14 to 15 grid-cells. Figures 4.4 and 4.5 show the computed and true RMS error of the water level for both cases. For a band width of less than 14 a behavior similar to that for 14 is observed and for more than 15 the performance gradually goes to the optimal value. The performance is exact, equal to that of the full Kalman filter, for band width  $b = 39$ . The large increase in performance from  $b = 14$  to  $b = 15$  is due to the position of the measurement station (grid cell 15). For band width  $b = 15$  an efficient implementation would require approximately a fraction  $b/N = 15/40$  of the computation time of the full Kalman filter. The method is expected to perform better in applications with a smaller correlation length of the error covariance.

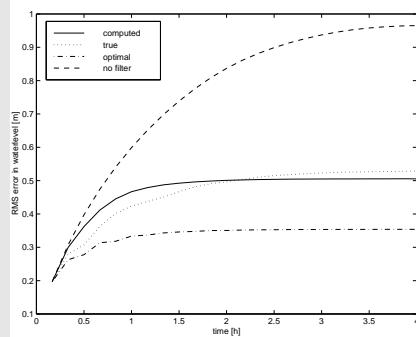


Figure 4.4: Root Mean Square error of the estimate of the water level; band width is 14

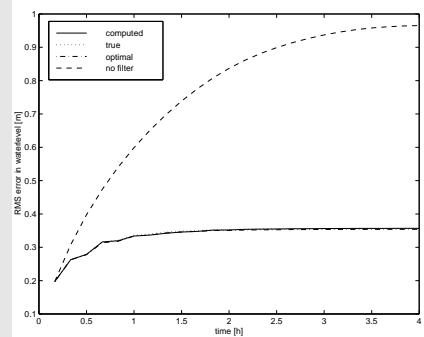


Figure 4.5: Root Mean Square error of the estimate of the water level; band width is 15

#### 4.2.5 Coarse grid approximation of the covariance matrix

Another approach to approximate the error covariance matrix is to use a coarser grid for this matrix. This reduces both the computation time of the time propagation of the error covariance as well as the storage requirements for the error covariance. The estimate itself is usually propagated on the original grid. Several successful applications of this approach have been reported (e.g. [33, 16]). The SOS tends to give satisfactory performance if the dominant phys-

ics is resolved by the coarse grid. A nice theoretical framework for this approach can be obtained from Curi et al [19], who shows from a continuous space point of view that one can use different spatial discretizations for the different equations.

Let  $\hat{\mathbf{x}}_c(\cdot)$  be the estimate represented on the coarse grid and  $\mathbf{S}$  an interpolation operator such that  $\hat{x}(\cdot) = \mathbf{S}\hat{\mathbf{x}}_c(\cdot)$  and a projection  $\mathbf{S}^*$  onto the coarse grid such that  $\mathbf{S}^*\mathbf{S} = \mathbf{I}_c$  where  $\mathbf{I}_c$  is the identity operator of the coarse grid. Using this notation the coarse grid filter can be denoted as

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}(k)\hat{\mathbf{x}}(k|k) + \mathbf{B}(k)\mathbf{u}(k) \quad (4.31)$$

$$\begin{aligned} \mathbf{P}_c(k+1|k) &= \mathbf{S}^*\mathbf{A}(k)\mathbf{S}\mathbf{P}_c(k|k)\mathbf{S}'\mathbf{A}(k)'(\mathbf{S}^*)' \\ &\quad + \mathbf{S}^*\mathbf{F}(k)\Sigma_s(k)\mathbf{F}(k)'(\mathbf{S}^*)' \end{aligned} \quad (4.32)$$

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) \\ &\quad + \mathbf{S}\mathbf{K}_c(k+1)(\mathbf{y}(k+1) - \mathbf{C}(k+1)\hat{\mathbf{x}}(k+1|k)) \end{aligned} \quad (4.33)$$

$$\begin{aligned} \mathbf{K}_c(k+1) &= \mathbf{P}_c(k+1|k)\mathbf{C}(k+1)'\mathbf{S}' \\ &\quad (\mathbf{C}(k+1)\mathbf{S}\mathbf{P}_c(k+1|k)\mathbf{S}'\mathbf{C}(k+1)' + \Sigma_o(k+1))^{-1} \end{aligned} \quad (4.34)$$

$$\mathbf{P}_c(k+1|k+1) = \mathbf{P}_c(k+1|k) - \mathbf{K}_c(k+1)\mathbf{C}(k+1)\mathbf{S}\mathbf{P}_c(k+1|k) \quad (4.35)$$

Note the error covariance  $\mathbf{P}_c(\cdot)$  is now given on the coarse grid and thus is much smaller. The expression  $\mathbf{S}^*\mathbf{A}(k)\mathbf{S}$  can also be approximated by a model evaluation directly of the coarse grid, which further decreases the computational burden of the algorithm. An important advantage of the coarse grid approximation is that the error covariance remains positive semi definite.

In more than one dimension the interpolation to the fine grid can become very difficult for more complex geometries.

### Example 5: Square root coarse grid Kalman filter for 1-D wave model

The algorithm shown above can also be formulated in a square root form. This square root coarse grid filter was applied to the 1-D wave example. For this SOS the performance increases gradually and is exact at increment 1, i.e. if the fine grid equals the coarse grid the performance is the same as for the full Kalman filter. Figure 4.6 shows the computed and true RMS of the errors for step size 8. For this implementation the computational cost was a fraction 1/8 of the cost for the full Kalman filter, because fewer model evaluations are needed. If the model is evaluated on the coarse grid the computational burden can be reduced further at the cost of a larger truncation error of the numerical scheme.

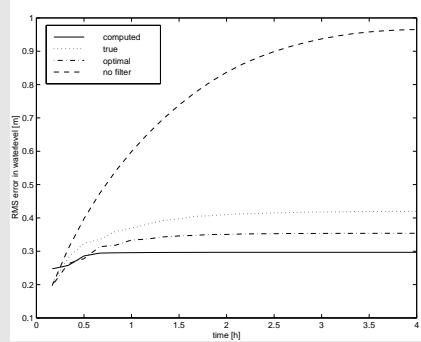


Figure 4.6: Root Mean Square error of the estimate of the water level; coarse grid is every 8'th

#### 4.2.6 Partial singular value Kalman filter

Instead of approximating the error covariance matrix it is also possible to approximate the matrix  $\mathbf{A}(k)$ . Cohn et al [16] proposed a filter based on the leading singular values of  $\mathbf{A}(k)$ . A Lanczos method (see e.g. [37]) was used to compute these leading eigenvalues efficiently. Both the tangent linear model and the adjoint model are needed for this. In their atmospheric application the uncertainty was caused mainly by the instabilities. This caused a good separation of the singular values of the matrix  $\mathbf{A}(k)$ . Therefore this matrix could be approximated accurately using only a few of the leading singular values. This partial singular value Kalman filter (PSKF) can be denoted as

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}(k)\hat{\mathbf{x}}(k|k) + \mathbf{B}(k)\mathbf{u}(k) \quad (4.36)$$

$$\begin{aligned}\mathbf{P}_c(k+1|k) &= \Pi(k+1) \left( \tilde{\mathbf{A}}(k) \mathbf{P}_c(k|k) \tilde{\mathbf{A}}(k)' + \mathbf{F}(k) \boldsymbol{\Sigma}_s(k) \mathbf{F}(k)' \right) \\ &\quad \Pi(k+1)'\end{aligned}\tag{4.37}$$

$$\begin{aligned}\hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) \\ &\quad + \mathbf{K}_c(k+1)(\mathbf{y}(k+1) - \mathbf{C}(k+1)\hat{\mathbf{x}}(k+1|k))\end{aligned}\tag{4.38}$$

$$\begin{aligned}\mathbf{K}_c(k+1) &= \mathbf{P}_c(k+1|k) \mathbf{C}(k+1)' \\ &\quad (\mathbf{C}(k+1) \mathbf{P}_c(k+1|k) \mathbf{C}(k+1)' + \boldsymbol{\Sigma}_o(k+1))^{-1}\end{aligned}\tag{4.39}$$

$$\mathbf{P}_c(k+1|k+1) = \mathbf{P}_c(k+1|k) - \mathbf{K}_c(k+1) \mathbf{C}(k+1) \mathbf{P}_c(k+1|k)\tag{4.40}$$

where  $\tilde{\mathbf{A}}(k)$  denotes the simplified model dynamics. Let

$$\mathbf{A}(k) = \mathbf{U}(k) \mathbf{D}(k) \mathbf{V}(k)'\tag{4.41}$$

be the singular value decomposition of  $\mathbf{A}(k)$ , with the singular values ordered, i.e.  $[\mathbf{D}(k)]_{1,1} = \sigma_1^2 \geq [\mathbf{D}(k)]_{2,2} = \sigma_2^2 \geq \dots \geq [\mathbf{D}(k)]_{n,n} = \sigma_n^2$  then

$$\tilde{\mathbf{A}}(k) = [\mathbf{U}(k)]_{:,1:q} [\mathbf{D}(k)]_{1:q,1:q} ([\mathbf{V}(k)]_{:,1:q})'\tag{4.42}$$

where  $q$  is the number of singular values included.  $\Pi(k+1)$  is a projection onto the  $q$  leading left singular vectors of  $A$ , i.e.  $\Pi(k+1) = [\mathbf{U}(k)]_{:,1:q} [\mathbf{U}(k)]'_{:,1:q}$ . For a computationally efficient algorithm the fact that  $\tilde{\mathbf{A}}(k) \mathbf{P}_c(k|k) \tilde{\mathbf{A}}(k)'$  has rank  $q$  can be exploited, e.g. by representing the error covariance relative to  $[\mathbf{V}(k)]_{:,1:q}$  or by using a square root formulation. If  $\mathbf{A}(k) = \mathbf{A}$  is time invariant the leading singular values need only be computed once. For time varying models the singular value decomposition must be updated regularly.

### Example 6: Partial singular-value Kalman filter for 1-D wave model]

The method was applied to the 1-D wave model. This model is significantly different from the model used by Cohn et al [16] in that the 1-D wave model is stable. Moreover, if there was no friction and a periodic boundary condition the matrix  $\mathbf{A}(k)$  would be orthogonal [67], i.e.  $\sigma_1 = \sigma_2 = \dots, \sigma_n = 1$ . The singular values for the model used in this example, i.e. with friction and different boundary conditions, are shown in figure 4.7. It can be seen that the singular values are not very well separated. It is therefore expected that the PSKF will not work very well for this model.

Figure 4.8 shows the RMS of true and computed error covariances of the water level. The number of singular values used was  $q = 10$ . If  $q$  is increased the performance slowly increases. For  $q = n$  the results are equal to those of the full Kalman filter.

Computation of the  $q$  leading singular values and singular vectors with the Lanczos iteration method requires a computation time which is a multiple of the time needed for  $q$  model evaluations. The exact number depends on the convergence rate of the algorithm. In this example the matrix  $[\mathbf{V}(k)]_{:,1:q}$  was stored, but the model was evaluated for all  $q$  columns in the square root of the covariance  $\mathbf{P}_c^{\frac{1}{2}}$ , which seems a sensible approach for slowly time varying models. In this case the computation time is approximately a fraction  $q/n = 10/81$  of the computation time needed for the full Kalman filter. If  $\mathbf{A}[\mathbf{V}]_{:,1:q}$  is stored this can be further reduced.

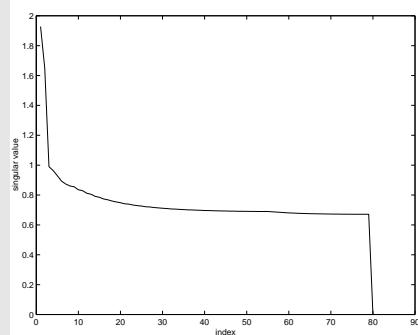


Figure 4.7: Singular values of  $\mathbf{A}$  in the 1-D wave model

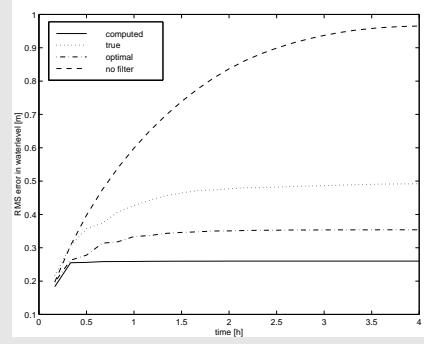


Figure 4.8: Root Mean Square error of the estimate of the water level; 10 singular values

#### 4.2.7 Model reduction and Kalman filtering

An approach related to the approximation of the transition matrix  $\mathbf{A}(k)$  is model reduction. Model reduction is a whole field of research related to systems theory where the objective is to approximate systems by smaller ones. There are several types of model reduction problems depending on whether the model is used for straightforward forecasting, filtering or control applications.

To illustrate the model reduction approach consider the problem of finding a reduced order model with similar input output behavior a the linear model of equations 2.30 and 2.31. For the moment assume that the deterministic input ( $\mathbf{u}(\cdot)$ ) is zero, and that the model is time invariant, i.e.

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{F}\mathbf{w}(k) \quad (4.43)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (4.44)$$

If a reduced order model with state dimension  $n_r < n$  is denoted as

$$\mathbf{x}_r(k+1) = \mathbf{A}_r\mathbf{x}_r(k) + \mathbf{F}_r w(k) \quad (4.45)$$

$$\mathbf{y}_r(k) = \mathbf{C}_r\mathbf{x}_r(k) \quad (4.46)$$

then the difference in input/output behavior of the two sytems can be measured with  $J$

$$J = \lim_{k \rightarrow \infty} E[(\mathbf{y}_r(k) - \mathbf{y}(k))' \mathbf{R}^{-1} (\mathbf{y}_r(k) - \mathbf{y}(k))] \quad (4.47)$$

where  $\mathbf{R}^{-1}$  is a weight matrix that expresses the relative importance attached to the outputs. In this context it is possible to define the optimal reduced order model of dimension  $n_r$  by minimizing  $J$  over the reduced order system matrices  $(\mathbf{A}_r, \mathbf{F}_r, \mathbf{C}_r)$ . The solution of this problem is given by [?]

$$\mathbf{A}_r = \boldsymbol{\Pi}_r \mathbf{A} \boldsymbol{\Pi}_l \quad (4.48)$$

$$\mathbf{F}_r = \boldsymbol{\Pi}_r \mathbf{F} \quad (4.49)$$

$$\mathbf{C}_r = \mathbf{C} \boldsymbol{\Pi}_l \quad (4.50)$$

$$\hat{\mathbf{P}} = \mathbf{A} \boldsymbol{\Pi} \hat{\mathbf{P}} \boldsymbol{\Pi}' \mathbf{A}' + \mathbf{F} \boldsymbol{\Sigma}_s \mathbf{F}' \quad (4.51)$$

$$\hat{\mathbf{Q}} = \mathbf{A}' \boldsymbol{\Pi}' \hat{\mathbf{Q}} \boldsymbol{\Pi} \mathbf{A} + \mathbf{C}' \mathbf{R}^{-1} \mathbf{C} \quad (4.52)$$

$$\boldsymbol{\Pi} = \boldsymbol{\Pi}_l' \boldsymbol{\Pi}_r \quad (4.53)$$

$$(4.54)$$

where

$$\boldsymbol{\Pi}_r \boldsymbol{\Pi}_l' = \mathbf{I}_{n_r} \quad (4.55)$$

$$\boldsymbol{\Pi} = \sum_{i=1}^{n_r} \boldsymbol{\Psi} \mathbf{E}_i \boldsymbol{\Psi}^{-1} \quad (4.56)$$

and

$$\boldsymbol{\Psi}^{-1} \hat{\mathbf{P}} (\boldsymbol{\Psi}^{-1})' = \boldsymbol{\Psi}' \hat{\mathbf{Q}} \boldsymbol{\Psi} = \hat{\mathbf{D}} \quad (4.57)$$

with  $\mathbf{E}_i$  an  $n$  by  $n$  matrix with zeros except at  $[\mathbf{E}_i]_{i,i} = 1$ . The matrix  $\hat{\mathbf{D}}$  is diagonal with diagonal elements  $[\hat{\mathbf{D}}]_{1,1} \geq [\hat{\mathbf{D}}]_{2,2} \geq \dots \geq [\hat{\mathbf{D}}]_{n,n} \geq 0$ .

The matrix  $\Psi$  denotes a state transformation  $\mathbf{x}_{tr}(k) = \Psi^{-1}\mathbf{x}(k)$  that *balances* the system, i.e. the transformed  $\hat{\mathbf{P}}$  and  $\hat{\mathbf{Q}}$  which are given by  $\Psi^{-1}\hat{\mathbf{P}}(\Psi^{-1})'$  and  $\Psi'\hat{\mathbf{Q}}\Psi$  respectively satisfy equation 4.57. Here  $\hat{\mathbf{P}}$  can be viewed as the influence of the inputs on the state variables for the reduced system and  $\hat{\mathbf{Q}}$  as the influence of the state variables for the reduced system on the outputs. The reduced order system is found from this transformation by projection on the leading  $n_r$  state variables of the transformed state ( $x_{tr}$ ). In equation 4.56 the projection on the first  $n_r$  elements of  $x_{tr}$  is computed. This projection is then factorized (eqn. 4.53, 4.55). The equations 4.48, 4.49 and 4.50 compute the projection of  $\mathbf{A}$ ,  $\mathbf{F}$  and  $\mathbf{C}$  using  $\Pi$ , but represented in the non-transformed space.

An important special case for our application is if we assume  $\hat{\mathbf{Q}} = I_n$ , i.e. all state variables are of equal importance to the outputs. In that case the equation 4.57 can be solved using the eigendecomposition of  $\hat{\mathbf{P}}$ . The matrix  $\hat{\mathbf{P}}$  is projected on the first  $n_r$  eigenvectors every time-step. But is equivalent to the RRSQRT algorithm without a measurement step (compare e.g. with equation 4.37).

A difference between the model reduction approach and the partial singular value filter is that in the model reduction approach the whole system is taken into account and not only the state transition matrix  $\mathbf{A}$ .

The model reduction approach can be extended to state estimation [?], and also to time varying models [?, ?]. For a detailed treatment of the possible extension the reader is referred to the references in these articles. An interesting development in the field of model reduction is the use of approximate (but faster) model reduction techniques [38].

An important disadvantage of the model reduction approach is that for computation of the balancing transformation two or more Lyapunov equations (such as eqns. 4.51 and 4.52) with state dimension  $n$  have to be solved. For time invariant models these computations can be performed off-line, which results in a very fast algorithm. The computation time is even smaller than one model evaluation. However, for the time varying Kalman filter problem studied here these computations can not be performed in advance since they are influenced by the value of the state estimate. Direct application of model for time varying model often leads to computation times that are even larger than for the full Kalman filter. Only if the projection is computed off-line the computations are feasible.

### Example 7: Model reduction approach for 1-D wave model

One way to use the projection of the model reduction for a SOS is to replace the projection in equation 4.37 by the projection of equation 4.56. For this the model should be linear and time invariant, although some extensions are possible. To assess the performance of this approach the SOS was applied to the 1-D wave model. Figure 4.9 shows the computed and true RMS of the error covariance. The approach seems to work quite well for a projection with rank 4. The computation time needed for this algorithm was a fraction  $n_r/n = q/n = 4/81$  of that for the full Kalman filter.

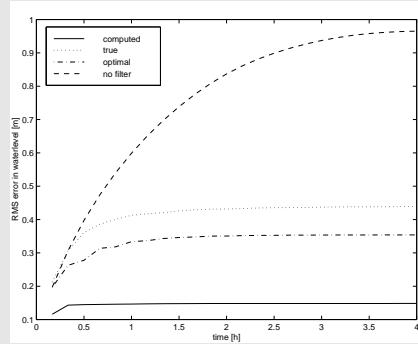


Figure 4.9: Root Mean Square error of the estimate of the water level; 4 dimensional reduced model

#### 4.2.8 The ensemble Kalman filter

The ensemble Kalman filter [30, 11] is a data assimilation method that approximates the conditional density with a Monte Carlo method. The data are assimilated using the data-step of the Kalman filter, where the error covariance matrix is replaced by the sample error covariance.

If a linear model is used the ensemble Kalman filter (EnKF) can be considered as a SOS, but for nonlinear model this method has the advantage that for large samples all the conditional moments are propagated correctly.

For the linear model of equations 2.30 and 2.31 the algorithm can be denoted as

$$\xi_i(k+1|k) = \mathbf{A}(k)\xi_i(k|k) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{F}(k)w_i(k) \quad (4.58)$$

$$\hat{\mathbf{x}}(k+1|k) = (1/q) \sum_{i=1}^q \xi_i(k+1|k) \quad (4.59)$$

$$\begin{aligned}\mathbf{P}_c(k+1|k) &= (1/q) \sum_{i=1}^q (\xi_i(k+1|k) - \hat{\mathbf{x}}(k+1|k)) \\ &\quad (\xi_i(k+1|k) - \hat{\mathbf{x}}(k+1|k))' \end{aligned}\tag{4.60}$$

$$\begin{aligned}\mathbf{K}_c(k+1) &= \mathbf{P}_c(k+1|k) \mathbf{C}(k+1)' \\ &\quad (\mathbf{C}(k+1) \mathbf{P}_c(k+1|k) \mathbf{C}(k+1)' + \boldsymbol{\Sigma}_o(k+1))^{-1} \end{aligned}\tag{4.61}$$

$$\begin{aligned}\xi_i(k+1|k+1) &= \xi_i(k+1|k) \\ &\quad + \mathbf{K}_c(\mathbf{y}(k+1) - \mathbf{C}(k)\xi_i(k+1|k) - \mathbf{v}_i(k+1)) \end{aligned}\tag{4.62}$$

where  $\xi_i(k|l)$  is an ensemble of state vectors generated with the realizations  $\mathbf{w}_i$  and  $\mathbf{v}_i$  of the processes  $\mathbf{w}$  and  $\mathbf{v}$  respectively. These realization are made using a pseudo random generator [?]. Note that  $\hat{\mathbf{x}}$  for finite samples depends on the actual realization used, i.e. different random number generators will give (slightly) different results. A big advantage of the method is that it is relatively simple to implement, a disadvantage is that the sample size needed may be quite large, unto several hundred.

### Example 8: Ensemble Kalman filter for 1-D wave model

In the following experiment the ensemble Kalman filter was applied to the 1-D wave model. Because this model is linear the true error covariance of the ensemble filter is never lower than that of the optimal 'full' Kalman filter.

Figure 4.10 shows the true and estimated RMS for an ensemble size of 10. The ensemble filter is stable and the true standard deviation of the errors is not that much larger than for the full Kalman filter. The fluctuations in the standard deviation are caused by the pseudo random numbers used. If the experiment would be carried out a second time the general behavior would be the same but the values would be different.

In the next two experiments the sample size was increased to 30 and 100. Figures 4.11 and 4.12 show the results of these experiments. It can be seen that by increasing the sample size the method becomes more accurate, but the convergence is not very fast. Without the assimilation of measurements it can be shown that the estimate converges with  $1/\sqrt{q}$ . The computational burden of the ensemble filter is approximately equivalent to one simulation for every element of the sample. In the experiments shown here this amount to fractions  $q/n = 10/81, 30/81, 100/81$  of the computation needed for the full Kalman filter.

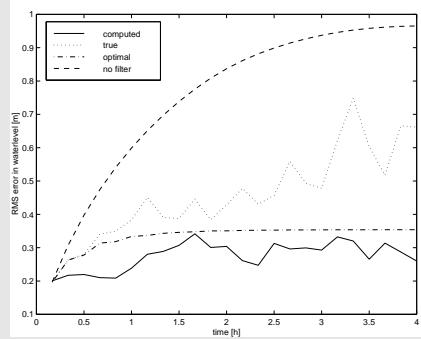


Figure 4.10: Root Mean Square error of the estimate of the water level; sample size 10

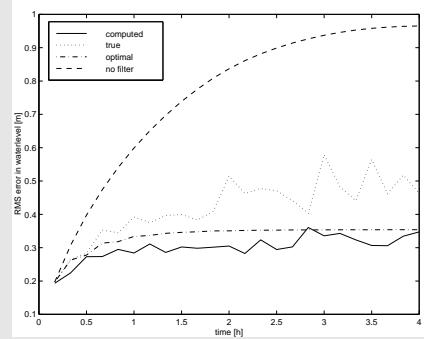


Figure 4.11: Root Mean Square error of the estimate of the water level; sample size 30

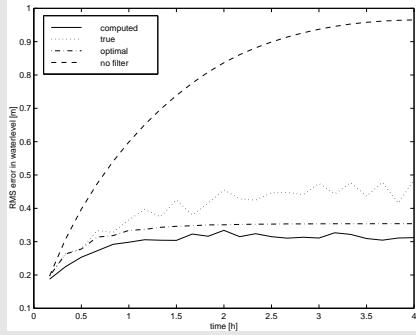


Figure 4.12: Root Mean Square error of the estimate of the water level; sample size 100

#### 4.2.9 A comparison of these SOS's

Following the overview of SOS's the question arises which SOS is most suitable for the applications described in Chapter 1. The characteristics of these applications were:

1. all three based on a large 2-dimensional or 3-D shallow water model
2. for two applications it is not clear how well they are approximated by a linear model
3. the stratified 3-D model of the third application is clearly non stationary
4. the correlation length of the model errors is more than a few grid cells
5. the models have a complicated land-sea boundary that makes interpolation difficult near this boundary

These characteristics should be compared with those of the various SOS's.

- **Steady state Kalman filter:** This algorithm is very fast and works well if the model is close to linear and time invariant. If this is not the case it is not possible to increase the accuracy.
- **Banded Kalman filter:** Straightforward truncation of correlations over distances longer than a threshold can easily lead to instability. Square root versions of this approach have much better properties in this respect, but are difficult to implement in more than one dimension. The threshold

must be longer than the correlation length of the error to obtain accurate results. It is not yet known how to implement the method efficiently. Implementation for time varying models and in an extended Kalman filter setting is straight forward.

- **Coarse grid Kalman filter:** This is a quite efficient method. If only the projection is used it is also accurate. It is possible to reduce the computation time further by using model evaluations on the coarse grid for time propagation of the error covariance. The coarse grid must still capture the essential model dynamics for the method to work well. Interpolation between the coarse grid-points must be defined, which is sometimes very difficult. An extended Kalman filter version is easily derived for this SOS.
- **PEKF and RRSQRT:** Efficient method that is easy to implement. More efficient if the spatial correlation of the errors is large. Also for these SOS's an extended Kalman filter is easily derived.
- **Partial singular value Kalman filter:** Reasonably efficient method, that is more accurate if the singular values of the system ( $\mathbf{A}$ ) are widely separated. More expensive if the dynamics is not close to time invariant because frequent computation of the partial singular value decomposition is needed in that case. Extension to an extended Kalman filter is conceptually no problem, but the matrix  $\mathbf{A}$  will become time varying in that case. Moreover, for the type of applications studied here the singular values of the tangent linear model ( $\mathbf{A}$ ) are not well separated, which makes the method not very efficient.
- **Model reduction based approach** The SOS based on model reduction is very efficient for time invariant models, and very inefficient for time varying models. For the time varying case the number of computations is even larger than for the 'full' Kalman filter. It is also possible to use only the projection of the reduction, but it is not clear how to extend this to non-linear and time varying models. However, the approach is a good starting point for the construction of new SOS's.
- **Ensemble KF** This SOS gives quite good results with little computation time, but the convergence is slow. Because of this it is not efficient if accurate results are needed. The algorithm is easy to implement and has nice properties for non-linear models.

When selecting on the basis of efficiency only the steady state filter, the coarse grid filter and the RRSQRT filter (or the PEKF filter) are preferable. Depending on the application and the desired accuracy the ensemble filter may also be a good option especially if the model is significantly nonlinear. The other SOS's shown in the comparison are slower. The steady state filter is fastest, but it is not clear if the models are nearly linear and time invariant. The coarse grid approach causes difficulties at the land-sea boundary where the interpolation is almost impossible, while on the other hand water level measurements are often

along the coast. It will therefore no be easy to implement this approach for the applications studied here.

Based on these arguments the RRSQRT filter and the Ensemble Kalman filter seem the best choice for the kind of applications studied here. If the model is nearly time invariant and linear the steady state filter is faster and may even be more accurate. Note that it is also possible to compute a steady state filter with e.g. the RRSQRT filter. If this computation is performed off-line the number of modes can be made larger for additional accuracy. For nonlinear models the Ensemble filter has distinct advantages. If the nonlinearities are less pronounced the RRSQRT algorithm is probably more efficient.

### 4.3 Conclusions

In this chapter the problems that occur when applying the Kalman filter to large models is discussed. With increasing size of the model, the computational burden becomes huge. Also storage of the error covariance matrix can become a problem for very large models. And in addition the filtering problems for large models are often ill conditioned, which necessitates the use of robust algorithms. These problems have lead to the search for efficient approximate algorithms or suboptimal schemes (SOS's). A number of examples of SOS's was given, all illustrated with the same 1-D wave model.

For the type of applications studied in this work the RRSQRT algorithm compares very well with the other SOS's. For nearly time invariant and linear models the steady state filter is preferable. For strongly nonlinear models the ensemble Kalman filter is probably the best option. The RRSQRT (or PEKF) algorithm is a good option for application with properties between these two extremes.

## Chapter 5

# The Reduced Rank Square Root filter

### 5.1 Introduction

The RRSQRT algorithm approximates the optimal error covariance  $\mathbf{P}(\cdot)$ , for forecast and analysis, by a matrix  $\mathbf{P}_c(\cdot)$  of a lower rank. Thus  $\mathbf{P}_c(\cdot)$  is an approximate error covariance 'computed' by the RRSQRT algorithm. The optimal error covariance  $\mathbf{P}(\cdot)$  is a symmetric positive semidefinite matrix. Failure to preserve this property for the approximation  $\mathbf{P}_c(\cdot)$  usually leads to instability of the filter algorithm. Using a square root factorization, where  $\mathbf{P}_c(\cdot)$  is written as  $\mathbf{P}_c = \mathbf{L}_c \mathbf{L}'_c$ , no negative eigenvalues can occur except when actually performing the computation  $\mathbf{L}_c \mathbf{L}'_c$  with a finite precision. Therefore, in the RRSQRT algorithm all computations are performed on  $\mathbf{L}_c(\cdot)$  directly and  $\mathbf{P}_c(\cdot)$  is never computed.

Square root factorizations are not unique. In the square root filters as introduced by Potter (see ([54, 5]) for an introduction) the Cholesky factorization was chosen because of its computational efficiency. For the RRSQRT algorithm the square root factorization is based on the eigendecomposition. To illustrate this factorization consider the eigendecomposition of  $\mathbf{P}(\cdot)$ , which is given by  $\mathbf{P} = \mathbf{V}\mathbf{D}\mathbf{V}'$  where the columns of  $\mathbf{V}$  contain the eigenvectors and  $\mathbf{D}$  is a diagonal matrix with the eigenvalues as its diagonal elements. The matrix

$$\mathbf{L} := \mathbf{V}\mathbf{D}^{\frac{1}{2}} \quad (5.1)$$

is now a square root factor of  $\mathbf{P}$  since  $\mathbf{P} = \mathbf{L}\mathbf{L}'$ . Eigendecompositions have been used in the past for square root filtering [60], but for handling ill-conditioned matrices not with the intent of computational efficiency.

Contrary to the Cholesky decomposition the eigendecomposition can easily be used for approximation by matrices of a lower rank, since the truncation of the eigendecomposition using only the largest  $q$  eigenvalues is the optimal rank  $q$  approximation of a symmetric positive semidefinite matrix in both spectral

and Frobenius norm ([37] p. 72). The error introduced by this approximation is the value of the first neglected eigenvalue in case of the spectral norm. For many applications, where the model was obtained from the discretization of a set of Partial Differential Equations, this approximation is quite accurate using only a few eigenvalues. Typical values of  $q$  are 10 to 100, which is much smaller than the state dimension  $n$ , typically  $10^3$  to  $10^6$ . In the square root factor (eqn. 5.1) this truncation can be performed by deleting all columns right from column  $q$  if the eigenvalues are in decreasing order. The reduction of the number of columns can easily be exploited to reduce the number of computations. The use of a low-rank approximation for systems resulting from discretisation of PDE's is quite common for the solution of certain types of PDE's [39].

Once approximated, the optimal square-root eigenfactor  $\mathbf{L}$  is no longer available. The RRSQRT algorithm now continues with the approximate factor  $\mathbf{L}_c$ . After the next time-step the result is then approximated again by maintaining again only  $q$  columns with the eigenvectors of the largest eigenvalues. In analogy to the discretization of differential equations the truncation errors per time-step can be made arbitrarily small by maintaining a sufficient number of columns. This does however not imply that the errors can not accumulate. For convergence stability of the algorithm is needed in addition to small truncation errors.

The steps of the RRSQRT algorithm resemble those of the square root filtering algorithm (see e.g. [54] Ch. 7). The three main steps are the time-step the reduction-step and the measurement-step. The main differences are that the reduction-step used here introduces an approximation contrary to 'regular' square root filters and the fact that  $\mathbf{L}_c$  has a much smaller number of columns than  $n$ , i.e.  $q \ll n$ . This reduced number of columns directly reduces the number of computations and storage requirements even though the equations in matrix notation look the same.

## 5.2 The algorithm

### 5.2.1 Time-step

The time-step performs the time propagation of the estimate and error covariance and is equivalent to equations 3.3,3.4 of the Kalman filter equations. The equations are

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}(k)\hat{\mathbf{x}}(k|k) + \mathbf{B}(k)\mathbf{u}(k) \quad (5.2)$$

$$\mathbf{L}_c(k+1|k) = [\mathbf{A}(k)\mathbf{L}_c(k|k), \mathbf{F}(k)\Sigma_s(k)^{1/2}] \quad (5.3)$$

where  $\mathbf{L}_c(k|k)$  is the  $n$  by  $q$  estimate square root of the error covariance  $\mathbf{P}(k|k)$  and  $[,]$  denotes that the large matrix is built from the two block matrices (thus  $\mathbf{L}_c(k+1|k)$  has  $q+m$  columns). The multiplication  $\mathbf{A}(k)\mathbf{L}_c(k|k)$  in equation 5.3 is much faster than for the full Kalman filter, because in the RRSQRT algorithm the matrix  $\mathbf{L}_c(k|k)$  contains  $q$  columns instead of  $n$  and  $q \ll n$ .

### 5.2.2 Reduction-step

The addition of columns,  $\mathbf{F}(k)\boldsymbol{\Sigma}_s(k)^{1/2}$  in equation 5.3, for the system noise every time-step would quickly increase computation times because with the matrix  $\mathbf{L}_c$  also the number of computations in the various steps of the algorithm would grow. Therefore the number of columns is reduced to  $q$  after every time-step. The concept of this approximation is to use only the first  $q$  leading eigenvalues and eigenvectors of the approximate error covariance matrix  $\mathbf{P}_c = \mathbf{L}_c(k+1|k)\mathbf{L}_c(k+1|k)'$ . Note however that this is the concept  $\mathbf{P}_c$  is never actually computed. The computations are based on the eigendecomposition of the matrix  $\mathbf{L}_c(k+1|k)'\mathbf{L}_c(k+1|k)$ . It is well known that this matrix has the same nonzero eigenvalues as  $\mathbf{L}_c(k+1|k)\mathbf{L}_c(k+1|k)'$ . Let

$$\mathbf{L}_c(k+1|k)'\mathbf{L}_c(k+1|k) = \mathbf{U}_c(k+1)\mathbf{D}_c(k+1)\mathbf{U}_c(k+1)' \quad (5.4)$$

be an eigenvalue decomposition, where it is assumed that the eigenvalues are in decreasing order and  $\mathbf{U}_c(k+1)'\mathbf{U}_c(k+1)$ . It can easily be shown [37](p. 427) that the eigenvectors of  $\mathbf{P}_c$  are given by the columns of  $(\mathbf{L}_c(k+1|k)\mathbf{U}_c(k+1)\mathbf{D}_c^{-1/2}(k+1))$ . Thus the  $i$ 'th column of the matrix  $\mathbf{L}_c(k+1|k)\mathbf{U}_c(k+1)$  contains the eigenvectors scaled by  $\sqrt{[\mathbf{D}_c]_{i,i}}$ . Also, it is well known that multiplying from the right with a unitary matrix does not affect the covariance ( $P = LUU'L' = LL'$ ). Truncation is performed by simply deleting some columns.

The actual computation is given by the eigenvalue decomposition of eqn. 5.4 followed by

$$\mathbf{L}_c^*(k+1|k) = [\mathbf{L}_c(k+1|k)\mathbf{U}_c(k+1)]_{1:n,1:q} \quad (5.5)$$

Here  $[\cdot]_{1:n,1:q}$  denotes the truncation of the number of columns to  $q$ . The above procedure is much faster than eigenvalue computations on the matrix  $\mathbf{P}_c = \mathbf{L}_c(k+1|k)\mathbf{L}_c(k+1|k)'$  directly or singular value computations on  $\mathbf{L}_c(k+1|k)$ , which could both also accomplish the task of reduction. This is caused by the fact that the matrix  $\mathbf{L}_c(k+1|k)'\mathbf{L}_c(k+1|k)$  is a  $q+m$  by  $q+m$  matrix and it is assumed here that  $q \ll n, m \ll n$ . Some care should be taken with the eigenvalue decomposition of  $\mathbf{L}_c(k+1|k)'\mathbf{L}_c(k+1|k)$  since small negative eigenvalues may occur due to round off in the matrix produkt; it is advisable to set these values to 0.

The position in the recursion where the reduction is carried out, before or after the time step is not very important. At first sight it seems more accurate to carry out the reduction after the measurement, since the noise introduced in the time-step is kept longer than when the truncation is carried out directly after the time-step. This observation holds only for one cycle of the algorithm. It is not yet clear whether this implies that the estimates are more accurate after more than one time step.

### 5.2.3 Measurement-step

Several of the measurement-update equations known in literature do not depend on the specific type of square-root. Here the scalar update of Potter is used ([54]

p. 374):

$$\mathbf{H}(k+1) = \mathbf{L}_c^*(k+1|k)' \mathbf{C}(k+1)' \quad (5.6)$$

$$\beta(k+1) = (\mathbf{H}(k+1)' \mathbf{H}(k+1) + \Sigma_o(k+1))^{-1} \quad (5.7)$$

$$\mathbf{K}_c(k+1) = \mathbf{L}_c^*(k+1|k) \mathbf{H}(k+1) \beta(k+1) \quad (5.8)$$

$$\begin{aligned} \mathbf{L}_c(k+1|k+1) &= \mathbf{L}_c^*(k+1|k) - \mathbf{K}_c(k+1) \mathbf{H}(k+1)' \\ &\quad (1 + (\beta(k+1) \Sigma_o(k+1))^{1/2})^{-1} \end{aligned} \quad (5.9)$$

Independent measurements can be processed one at a time. If the measurements are correlated, then these measurements can be transformed. The main difference is again the size of the matrix  $\mathbf{L}_c$  which dramatically reduces the number of computations (roughly by a factor  $n/q$ ).

#### 5.2.4 Initialization

For many applications the initial transient of the estimate is not important and  $\mathbf{P}(0|0)$  can be set to 0. In this case  $\mathbf{L}_c(0|0)$  also becomes 0. If this is not the case then  $\mathbf{P}(0|0)$  or  $\mathbf{P}(0|-1)$  can be approximated using the  $q$  leading eigenvectors and eigenvalues, for which a Lanczos type algorithm ([37] Ch. 9) can be used.

#### 5.2.5 Remarks

The columns of  $\mathbf{L}_c$  can be interpreted as error vectors in the state space. In some respects these columns are a generalization of the 'modes' of a system and will therefore also be called modes.

The number of computations required in the time propagation of the error in the covariance, which is a major fraction of the total number, is reduced by a factor  $\frac{n}{q}$  with respect to the original Kalman filter algorithm. The parameter  $q$  controls the accuracy of the approximation. The price for greater accuracy is as always a larger computational burden.

### 5.3 Nonlinear models

#### 5.3.1 The RRSQRT filter as Extended Kalman filter

So far, this work dealt with estimation for linear models. For non-linear models the RRSQRT algorithm can adapted to approximate the extended Kalman filter. The changes needed are conceptually not very difficult. The time propagation of the estimate is performed using the non-linear model. The time propagation of the error covariance estimate uses the tangent linear model  $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ , which is linearization of the model around the current estimate. In this approach the state  $\mathbf{x}$  is written as a mean estimate  $\hat{\mathbf{x}}$  plus some 'small' deviation  $\delta \mathbf{x}$  ( $\mathbf{x} = \hat{\mathbf{x}} + \delta \mathbf{x}$ ). The time propagation for  $\delta \mathbf{x}$  is (up to first order) given by

$$\delta \mathbf{x}(k+1) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\delta \mathbf{x}(k)) + \frac{\partial \mathbf{f}}{\partial \mathbf{w}} \mathbf{w}(k) \quad (5.10)$$

The main difficulty with this method in practice is that the derivation and implementation of the tangent linear model is a lot of work. To avoid the use of a tangent linear model a method based on finite differences is proposed here.

For the extended Kalman filter the matrix  $\mathbf{A}(k)$  of equation 5.3 is replaced by  $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$  evaluated at the most recent estimate  $\hat{\mathbf{x}}(k|k)$ . Let the i'th column of  $\mathbf{L}(k|k)$  be denoted by  $\mathbf{l}_i(k|k)$  then

$$\begin{aligned}\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{L}(k|k) &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}} [\mathbf{l}_1(k|k), \dots, \mathbf{l}_q(k|k)] \\ &= [\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{l}_1(k|k), \dots, \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{l}_q(k|k)]\end{aligned}\quad (5.11)$$

The column vector  $\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{l}_i(k|k)$  can be approximated by

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{l}_i(k|k) \approx \frac{\mathbf{f}(\hat{\mathbf{x}}(k|k) + \varepsilon \mathbf{l}_i(k|k)) - \mathbf{f}(\hat{\mathbf{x}}(k|k))}{\varepsilon} \quad (5.12)$$

where  $\varepsilon$  is small. And thus

$$\begin{aligned}\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{L}(k|k) &\approx \left[ \frac{\mathbf{f}(\hat{\mathbf{x}}(k|k) + \varepsilon \mathbf{l}_1(k|k)) - \mathbf{f}(\hat{\mathbf{x}}(k|k))}{\varepsilon}, \dots, \right. \\ &\quad \left. \frac{\mathbf{f}(\hat{\mathbf{x}}(k|k) + \varepsilon \mathbf{l}_q(k|k)) - \mathbf{f}(\hat{\mathbf{x}}(k|k))}{\varepsilon} \right]\end{aligned}\quad (5.13)$$

For the computation of  $\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{L}(k|k)$   $q+1$  evaluations of  $\mathbf{f}$  are needed, but  $\mathbf{f}(\hat{\mathbf{x}}(k|k))$  is also needed for equation 5.2. Usually the number of computations needed for an evaluation like  $\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{l}_i(k|k)$  is close to the number of computations needed for an evaluation of  $\mathbf{f}(\cdot)$ . In this case the proposed method requires approximately the same number of computations as the extended RRSQRT Kalman filter using a tangent linear model; the effort needed for implementation is however considerably less.

For 'small' non-linearities the extended Kalman filter, either using a tangent linear model or finite differences is expected to yield good results. This is mainly due to the exponential stability of the Kalman filter, so that errors vanish in time ([52]). For 'strong' non-linearities or discontinuities however the extended Kalman filter may fail. Many successful applications of the extended Kalman filter indicate that this approach to non-linear filtering is indeed a good alternative for practical applications (see e.g. [54]). An important discontinuity in tidal flow models is when area's with a height close to the mean sea level are flooded. It is expected that the method for approximate time propagation proposed here is somewhat robust to these discontinuities since no derivatives are needed. The experiments in the sequel show that the method also gives reasonable results in the case of a few points that become dry or flooded. Other experiments are needed to see if it is possible to use the method for models of estuaries.

### 5.3.2 A second order accurate RRSQRT filter

The main idea of the second order accurate algorithm by Julier [45] is that a finite sample can represent the mean and covariance of a random variable and that the mean and covariance of a function of this random variable can be approximated by the sample mean and sample covariance of the sample obtained by applying the function to all elements in the sample. This approximation was then shown to be second order accurate.

It is easy to show that the sample

$$S = \{\mathbf{x} + \sqrt{q}\mathbf{l}_1, \mathbf{x} - \sqrt{q}\mathbf{l}_1, \dots, \mathbf{x} + \sqrt{q}\mathbf{l}_q, \mathbf{x} - \sqrt{q}\mathbf{l}_q\} \quad (5.14)$$

where  $\mathbf{L} = [\mathbf{l}_1, \dots, \mathbf{l}_q]$ , has sample mean  $\mathbf{x}$  and sample covariance  $\mathbf{LL}'$ . Using the theorem by Julier [45] the sample

$$S = \{\mathbf{f}(\mathbf{x} + \sqrt{q}\mathbf{l}_1, \mathbf{u}), \mathbf{f}(\mathbf{x} - \sqrt{q}\mathbf{l}_1, \mathbf{u}), \dots, \mathbf{f}(\mathbf{x} + \sqrt{q}\mathbf{l}_q, \mathbf{u}), \mathbf{f}(\mathbf{x} - \sqrt{q}\mathbf{l}_q, \mathbf{u})\} \quad (5.15)$$

has approximately sample mean  $E[\mathbf{f}(\mathbf{x}, \mathbf{u})]$  and sample covariance  $E[(\mathbf{f}(\mathbf{x}, \mathbf{u}) - E[\mathbf{f}(\mathbf{x}, \mathbf{u})])(\mathbf{f}(\mathbf{x}, \mathbf{u}) - E[\mathbf{f}(\mathbf{x}, \mathbf{u})])']$ . The errors in this approximation are all fourth order. Therefore this procedure can be used for the time step in a truncated second order filter.

If the columns  $\mathbf{l}_i$  are orthogonal the factor  $\sqrt{q}$  compensates for the fact that there are only two terms that contribute in that direction, contrary to a random sample where all columns contribute approximately the same amount. Because of this factor the points in the sample  $S$  are further away from the mean, if  $q > 1$ . This will amplify the effect of higher order terms which is not desirable. Therefore we will scale the elements in the sample. This scaling has to be compensated for when computing sample mean and sample covariance.

This approach results in the following second order accurate timestep:

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k) &= \mathbf{f}(\hat{\mathbf{x}}(k|k), \mathbf{u}(k)) \\ &+ \sum_{i=1}^q \left( \frac{\mathbf{f}(\hat{\mathbf{x}}(k|k) + \varepsilon \mathbf{l}_i, \mathbf{u}(k)) - \mathbf{f}(\hat{\mathbf{x}}(k|k), \mathbf{u}(k))}{2\varepsilon^2} \right. \\ &\quad \left. + \frac{\mathbf{f}(\hat{\mathbf{x}}(k|k) - \varepsilon \mathbf{l}_i, \mathbf{u}(k)) - \mathbf{f}(\hat{\mathbf{x}}(k|k), \mathbf{u}(k))}{2\varepsilon^2} \right) \end{aligned} \quad (5.16)$$

$$\begin{aligned} \mathbf{L}(k+1|k) &= \left[ \frac{\mathbf{f}(\hat{\mathbf{x}}(k|k) + \varepsilon \mathbf{l}_1, \mathbf{u}(k)) - \mathbf{f}(\hat{\mathbf{x}}(k|k), \mathbf{u}(k))}{\sqrt{2}\varepsilon}, \dots, \right. \\ &\quad \frac{\mathbf{f}(\hat{\mathbf{x}}(k|k) + \varepsilon \mathbf{l}_q, \mathbf{u}(k)) - \mathbf{f}(\hat{\mathbf{x}}(k|k), \mathbf{u}(k))}{\sqrt{2}\varepsilon}, \\ &\quad \frac{\mathbf{f}(\hat{\mathbf{x}}(k|k) - \varepsilon \mathbf{l}_1, \mathbf{u}(k)) - \mathbf{f}(\hat{\mathbf{x}}(k|k), \mathbf{u}(k))}{\sqrt{2}\varepsilon}, \dots, \\ &\quad \left. \frac{\mathbf{f}(\hat{\mathbf{x}}(k|k) - \varepsilon \mathbf{l}_q, \mathbf{u}(k)) - \mathbf{f}(\hat{\mathbf{x}}(k|k), \mathbf{u}(k))}{\sqrt{2}\varepsilon}, \mathbf{F}\Sigma_s^{1/2} \right] \end{aligned} \quad (5.17)$$

For small  $\varepsilon$  one can also prove directly that the equations 5.16 and 5.17 are equivalent to the truncated second order timestep.

The equations 5.16 and 5.17 need  $2q + 1$  evaluations of the function  $\mathbf{f}$ . If  $q \ll n$  the number of computations is reduced considerably. The number of computations is about two times that needed for the first order RRSQRT algorithm, but this timestep is second order accurate.

## 5.4 Some properties of the RRSQRT filter

### 5.4.1 Reduction to exact filter for $q = n$

An important property of the RRSQRT filter is that it reduces to an 'exact' square root filter for  $q = n$  and thus it is algebraically equivalent to the Kalman filter for  $q = n$ . The proof that square root filters are algebraically equivalent to the Kalman filter can for instance be found in [54, 5]. Here we will only illustrate the main points of the proof.

The equivalence of 'full' ( $q = n$ ) square root filters is most easily shown using induction. For  $k = 0$  the equivalence is trivial. Now, consider the three steps of the algorithm separately:

- **Time-step**

The time-step of equation 5.3 is equivalent to 3.4. This can be seen from

$$\begin{aligned} & \mathbf{L}(k+1|k)\mathbf{L}(k+1|k)' \\ &= [\mathbf{A}(k)\mathbf{L}(k|k), \mathbf{F}(k)\Sigma_s(k)^{1/2}][\mathbf{A}(k)\mathbf{L}(k|k), \mathbf{F}(k)\Sigma_s(k)^{1/2}]' \\ &= \mathbf{A}(k)\mathbf{L}(k|k)\mathbf{L}(k|k)'\mathbf{A}(k)' + \mathbf{F}(k)\Sigma_s(k)\mathbf{F}(k)' \\ &= \mathbf{A}(k)\mathbf{P}(k|k)\mathbf{A}(k)' + \mathbf{F}(k)\Sigma_s(k)\mathbf{F}(k)' \\ &= \mathbf{P}(k+1|k) \end{aligned}$$

Thus  $\mathbf{P}(k+1|k) = \mathbf{L}(k+1|k)\mathbf{L}(k+1|k)'$  if  $\mathbf{P}(k|k) = \mathbf{L}(k|k)\mathbf{L}(k|k)'$ .

- **Reduction-step**

The error introduced by the reduction-step equals the first neglected eigenvalue. But,  $\mathbf{L}(\cdot, \cdot)$  has at most  $n$  positive eigenvalues because it has  $n$  rows. Therefore the truncation error for  $q = n$  must be 0.

- **Measurement-step**

To prove the equivalence of the measurement-step of equations 3.5, 3.6, 3.7 with 5.6,5.7,5.8,5.9 three steps are needed. First it is easy to show that the transformation of the measurements does not alter the filter results, and thus  $\Sigma_o$  can be assumed diagonal. The second step is to show the equivalence between processing one measurement at a time or processing all measurements at one time ( $k + 1$ ) at once:

It is well known that equations 3.6,3.7 imply that

$$\mathbf{P}(k+1|k+1)^{-1} = \mathbf{P}(k+1|k)^{-1} + \mathbf{C}(k+1)'\Sigma_o(k+1)\mathbf{C}(k+1) \quad (5.18)$$

If  $\Sigma_o$  is diagonal and the rows of  $\mathbf{C}$  are denoted as  $\mathbf{c}_1, \dots, \mathbf{c}_p$  then equation 5.18 can be written as

$$\begin{aligned}\mathbf{P}(k+1|k+1)^{-1} &= \mathbf{P}(k+1|k)^{-1} \\ &+ \mathbf{c}_1(k+1)'\Sigma_{1,1}(k+1)\mathbf{c}_1(k+1) + \dots \\ &+ \mathbf{c}_p(k+1)'\Sigma_{p,p}(k+1)\mathbf{c}_p(k+1)\end{aligned}\quad (5.19)$$

which shows that the measurements can be added one at a time. The proof for  $\hat{\mathbf{x}}$  is less illustrative and will be omitted here.

The third part of the proof shows the equivalence for one measurement. Consider the scalar measurement  $\mathbf{y}_i(k) = \mathbf{c}_i\mathbf{x}(k) + \mathbf{v}_i(k)$ . Combining equations 3.6, 3.7 we obtain

$$\begin{aligned}\mathbf{P}(k+1|k+1) &= \mathbf{P}(k+1|k) - \mathbf{P}(k+1|k)\mathbf{C}(k+1)' \\ &\quad (\mathbf{C}(k+1)\mathbf{P}(k+1|k)\mathbf{C}(k+1)' + \Sigma_o(k+1))^{-1} \\ &\quad \mathbf{C}(k+1)\mathbf{P}(k+1|k) \\ &= \mathbf{L}(k+1|k)(I - \beta\mathbf{L}(k+1|k)'\mathbf{C}(k+1)'\mathbf{C}(k+1)\mathbf{L}(k+1|k))\mathbf{L}'(k+1|1) \\ &= \mathbf{L}(k+1|k)(I - \gamma\beta\mathbf{L}(k+1|k)'\mathbf{C}(k+1)'\mathbf{C}(k+1)\mathbf{L}(k+1|k)) \\ &\quad (\mathbf{L}(k+1|k)(I - \gamma\beta\mathbf{L}(k+1|k)'\mathbf{C}(k+1)'\mathbf{C}(k+1)\mathbf{L}(k+1|k)))'\end{aligned}$$

where  $\beta = 1/(\mathbf{C}(k+1)\mathbf{P}(k+1|k)\mathbf{C}(k+1)' + \Sigma_o(k+1))$  and  $\gamma = 1/(1+\sqrt{\beta})$ . Here we recognise equations 5.6, 5.7, 5.8, 5.9.

### 5.4.2 Underestimation of the error covariance

Another important property of the RRSQRT filter is that the computed error covariance  $\mathbf{P}_c = \mathbf{L}_c\mathbf{L}_c'$  systematically underestimates the covariance. Moreover, the following lemma can be proven.

#### Lemma 1

$$\forall_k \mathbf{P}_c(k|k) \leq \mathbf{P}(k|k) \quad (5.20)$$

$$\forall_k \mathbf{P}_c(k+1|k) \leq \mathbf{P}(k+1|k) \quad (5.21)$$

#### Proof:

Define the error in the computed error covariance as

$$\mathbf{E}_1(k|k) := \mathbf{P}(k|k) - \mathbf{P}_c(k|k) \quad (5.22)$$

$$\mathbf{E}_1(k+1|k) := \mathbf{P}(k+1|k) - \mathbf{P}_c(k+1|k) \quad (5.23)$$

The positiveness of  $\mathbf{E}_1$  proceeds using induction. First  $\mathbf{P}_c(0|-1) \leq \mathbf{P}(0|-1)$  since either both are zero or  $\mathbf{P}_c(0|-1)$  is computed using a truncated eigenvalue decomposition (similar to eqn. 5.26).

Since the time-step of the RRSQRT filter (eqn. 5.3) is equivalent to the time-step of the Kalman filter (eqn. 3.4) we have

$$\mathbf{P}_c(k+1|k) = \mathbf{A}(k)\mathbf{P}_c(k|k)\mathbf{A}(k)' + \mathbf{F}(k)\Sigma(k)\mathbf{F}(k)' \quad (5.24)$$

and by using the definition of  $\mathbf{E}_1$

$$\mathbf{E}_1(k+1|k) = \mathbf{A}(k)\mathbf{E}_1(k|k)\mathbf{A}(k)' \quad (5.25)$$

If  $\mathbf{E}_1(k|k) \geq 0$  then (using eqn. 5.25) also  $\mathbf{E}_1(k+1|k) \geq 0$ .

The reduction-step can also be written as

$$\mathbf{P}_c^*(k+1|k) = \mathbf{P}_c(k+1|k) - \mathbf{T}(k+1) \quad (5.26)$$

where  $\mathbf{P}_c^*(k+1|k) := \mathbf{L}_c^*(k+1|k)(\mathbf{L}_c^*(k+1|k))'$  and (see also eqn 5.5)  $\mathbf{T}(k+1) = [\mathbf{L}_c(k+1|k)\mathbf{U}(k+1)]_{1:n,q+1:q+m}(\mathbf{L}_c(k+1|k)\mathbf{U}(k+1))_{1:n,q+1:q+m}'$ . This can easily be shown using the orthogonality of the columns of  $\mathbf{L}_c(k+1|k)\mathbf{U}(k+1)$ . The matrix  $\mathbf{T}(k+1)$  is always positive semi-definite. Because  $\mathbf{T}(k+1) \geq 0$  the reduction-step preserves the inequality; if  $\mathbf{E}_1(k+1|k) \geq 0$  then  $\mathbf{E}_1^*(k+1|k) \geq 0$ .

The measurement-step of the RRSQRT filter is equivalent to the measurement-step of the Kalman filter

$$\begin{aligned} \mathbf{K}_c(k+1) &= \mathbf{P}_c(k+1|k)\mathbf{C}(k+1)' \\ &\quad (\mathbf{C}(k+1)\mathbf{P}_c(k+1|k)\mathbf{C}(k+1)' + \Sigma_o(k+1))^{-1} \end{aligned} \quad (5.27)$$

$$\mathbf{P}_c(k+1|k+1) = \mathbf{P}_c(k+1|k) - \mathbf{K}_c(k+1)\mathbf{C}(k+1)\mathbf{P}_c(k+1|k) \quad (5.28)$$

These equations and the equations 3.6, 3.7 can be rewritten as ([54])

$$\mathbf{P}(k+1|k+1) = ((\mathbf{P}(k+1|k))^{-1} + \mathbf{C}(k+1)'\Sigma_o(k+1)\mathbf{C}(k+1))^{-1} \quad (5.29)$$

$$\mathbf{P}_c(k+1|k+1) = ((\mathbf{P}_c(k+1|k))^{-1} + \mathbf{C}(k+1)'\Sigma_o(k+1)\mathbf{C}(k+1))^{-1} \quad (5.30)$$

Using the Lemma 2 of appendix A, it is easy to show that if  $\mathbf{E}_1(k+1|k) \geq 0$  then  $\mathbf{E}_1(k+1|k+1) \geq 0$ . The equations 5.29, 5.30 only hold for strictly positive  $\mathbf{P}(k+1|k)$  and  $\mathbf{P}_c(k+1|k)$ ; if this is not the case then eqns. 3.6, 3.7 and 5.27, 5.28 are still valid. To complete the proof in this case  $\mathbf{P}(k+1|k)$  and  $\mathbf{P}_c(k+1|k)$  are approximated by strictly positive matrices, which can become arbitrarily close, and taking limits.

The proof of the lemma is now completed by an induction argument.

For the 'full' Kalman filter (eqns. 3.3-3.9)  $\mathbf{P}$  is the error covariance of the estimate. When an approximation  $\mathbf{P}_c$  of this matrix is used for computation of the gain the real error of the estimate no longer has this (optimal) error covariance but a larger value  $\mathbf{P}_t$ , called the true error covariance. This true error covariance behaves according to (Jazwinski [44] p. 248):

$$\mathbf{P}_t(k+1|k) = \mathbf{A}(k)\mathbf{P}_t(k|k)\mathbf{A}(k)' + \mathbf{F}(k)\Sigma_s(k)\mathbf{F}(k)' \quad (5.31)$$

$$\begin{aligned} \mathbf{P}_t(k+1|k+1) &= ((\mathbf{I} - \mathbf{K}_c(k+1)\mathbf{C}(k+1))\mathbf{P}_t(k+1|k) \\ &\quad (\mathbf{I} - \mathbf{K}_c(k+1)\mathbf{C}(k+1))') \\ &\quad + \mathbf{K}_c(k+1)\Sigma_o(k)\mathbf{K}_c(k+1)' \end{aligned} \quad (5.32)$$

where  $\mathbf{K}_c$  is the Kalman gain computed from the approximate error covariance  $\mathbf{P}_c$ . Combining the non optimality of the approximate filter and the lemma above we obtain:

$$\forall_k \mathbf{P}_c(k|k) \leq \mathbf{P}(k|k) \leq \mathbf{P}_t(k|k) \quad (5.33)$$

$$\forall_k \mathbf{P}_c(k+1|k) \leq \mathbf{P}(k+1|k) \leq \mathbf{P}_t(k+1|k) \quad (5.34)$$

From these inequalities we can observe that although the computed error covariance is stable if the optimal one is, this is not necessarily true for the true error covariance. If the number of modes is not sufficient the computed error covariance remains stable but the estimate itself does not. In the sequel it will be shown that if the truncation error is small enough the estimate will be stable and both the computed error covariance and the true error covariance converge to the optimal error covariance.

#### 5.4.3 Convergence of the RRSQRT filter

It was shown in the previous section that if the number of modes  $q$  is set equal to the number of state variables  $n$ , then the RRSQRT introduces no approximation. In this sense the algorithm always converges. However it still has to be shown that for  $q < n$  the error can be made small under certain conditions.

##### Theorem 1 (convergence RRSQRT)

If

- the system (eqns. 2.30, 2.31) is time-invariant, i.e.  $\forall_k \mathbf{A}(k) = \mathbf{A}$ ,  $\mathbf{B}(k) = \mathbf{B}$ ,  $\mathbf{C}(k) = \mathbf{C}$ ,  $\mathbf{F}(k) = \mathbf{F}$ ,  $\Sigma_s(k) = \Sigma_s$  and  $\Sigma_o(k) = \Sigma_o > 0$
- and the limit

$$\lim_{k \rightarrow \infty} \mathbf{P}(k+1|k) = \mathbf{P} \quad (5.35)$$

exists and

$$\Phi := (\mathbf{I} - \mathbf{K}\mathbf{C})\mathbf{A} \quad (5.36)$$

is stable, where  $\mathbf{K} = \lim_{k \rightarrow \infty} \mathbf{K}(k)$

then

for every  $\varepsilon > 0$  there exists a  $\delta > 0$  such that if  $\forall_k \|\mathbf{T}_k\| < \delta$  then

$$\forall_k \|\mathbf{P}(k|k) - \mathbf{P}_c(k|k)\| < \varepsilon \quad (5.37)$$

##### Proof:

The proof can be found in appendix A

**Remarks:**

-The type of norm used is not important since all norms are equivalent on  $R^{n \times n}$ .

- $\|\mathbf{P}(k|k) - \mathbf{P}_c(k|k)\| < \varepsilon$  can be replaced by  $\|\mathbf{P}(k+1|k) - \mathbf{P}_c(k+1|k)\| < \varepsilon$ ,

$\|\mathbf{P}_t(k|k) - \mathbf{P}_c(k|k)\| < \varepsilon$ ,  $\|\mathbf{P}_t(k+1|k) - \mathbf{P}_c(k+1|k)\| < \varepsilon$  or  $\|\mathbf{K}_c(k) - \mathbf{K}(k)\| < \varepsilon$ .

-If the system is time invariant and  $(\mathbf{C}, \mathbf{A})$  is observable

(i.e.  $\mathcal{O} = \sum_{i=0, n-1} \mathbf{C}\mathbf{A}^i(\mathbf{C}\mathbf{A}^i)'$  has rank  $n$ ) and  $(\mathbf{A}, \mathbf{F}\Sigma_s^{\frac{1}{2}})$  is controllable (i.e.

$\mathcal{C} = \sum_{i=0, n-1} (\mathbf{A}^i \mathbf{F}\Sigma_s^{\frac{1}{2}})(\mathbf{A}^i \mathbf{F}\Sigma_s^{\frac{1}{2}})'$  has rank  $n$ ) then both  $\mathbf{K}(k)$  and  $\mathbf{P}(k|k)$  converge to steady state values,  $\mathbf{K}$  and  $\mathbf{P}$  respectively and  $\Psi = (\mathbf{I} - \mathbf{K}\mathbf{C})\mathbf{A}$  is stable.

This well known result was first proved by Kalman [47]. A steady state solution also exists if  $\mathbf{A}$  is stable. In this case no observability and controllability conditions are needed [52]. The conditions above are only sufficient and can be relaxed in various ways [?]. These results can be used in combination with the above theorem to prove convergence of the RRSQRT for many applications.

- It follows from the proof that the approximation is more sensitive if the measurement errors are small. This effect can also be observed in experiments.

- Theorem 1 still does not guarantee that there exists a good approximation for  $q < n$ . It is not difficult to construct a counter example where the eigenvalues of the error covariance are all equal and there exists no good approximation for  $q < n$ . However, for discrete time systems that originate from the discretization of Partial Differential Equations the error covariance matrix usually has only a few relatively large eigenvalues. This may be due to the dissipation or diffusion that is introduced in most numerical schemes for stability reasons or was present in the model before discretization. In these type of applications the algorithm can be expected to produce accurate estimates for  $q \ll n$ .

- The uniform bound on  $T_k$  is used in the proof to show convergence in the limit ( $k \rightarrow \infty$ ). This is not strictly necessary since the amplification of truncation errors is partly a cumulative effect, thus errors can be compensated for in subsequent steps as long as  $\Phi_c := (I - \mathbf{K}_c \mathbf{C})\mathbf{A}$  remains stable. The amplification of small errors is determined by the stability of  $\Psi$ . Thus a very stable or well observable system allows larger truncation errors. It should be kept in mind however that larger truncation errors also may destabilize  $\Phi_c$ .

- The RRSQRT filter has been applied successfully to:

- Two dimensional air pollution model [79]
- One dimensional hydrodynamic model [76] and example below
- Two dimensional hydrodynamic model [77, 12, 70]
- Two dimensional meteorological model [15]. Cohn and Todling did not use the RRSQRT filter, but their PEKF filter, that is based on similar principles, but is algorithmically quite different. Since their SOS is algebraically equivalent the proof of convergence also holds for this algorithm.
- A three dimensional hydrodynamic model. This application will be shown in the sequel.

In the applications above the state dimension varies between  $n \approx 100$  and  $n \approx 60000$ . The number of modes needed was  $q \approx 8$  for the one dimensional model and  $q \approx 50$  for all the two dimensional models, which results in huge reduction of computation time and storage requirements compared to the 'full' Kalman filter for these applications.

#### 5.4.4 Example: properties and convergence for the wave model

The properties of the RRSQRT filter discussed above can be illustrated with the 1-D wave model of the examples. For this purpose a twin experiment with this model and the RRSQRT filter was used.

The estimated standard deviation of the error at the measurement point, together with the corresponding true and optimal values are shown as a function of time in figure 5.1. The estimate and true values are given for both 2 and 3 modes. The RRSQRT filter appears to be unstable for  $q = 2$  and reasonably accurate for  $q = 3$ . Note that with state dimension  $n = 81$ , the RRSQRT filter is roughly  $81/3 = 27$  times faster than the full Kalman filter.

In the proof of convergence of the RRSQRT filter it was found that the accuracy of the measurements may strongly influence the convergence of the RRSQRT filter. Therefore, an experiment was performed with more accurate measurements ( $0.01[m]$  instead of  $0.10[m]$  standard deviation of the measurement noise). The results (see figure 5.2) indeed show a worse performance of the RRSQRT filter. Although unstable for  $q = 3$ , the RRSQRT filter was stable and reasonably accurate for  $q = 5$  for this example. It is hard to distinguish this curve from the optimal standarddeviation in the figure.

To further illustrate the behaviour of the filter an experiment with generated data was performed. In a 'truth' model a 'pulse' of 2 hour duration was generated. Measurements of the water-level were generated at  $x_m = 24[km]$ . These generated measurement were subsequently used as input for the filter. In figures 5.3 and 5.4, the true and estimated water-levels are shown at  $t = 30, 100[min]$ . For this experiment  $q = 4$  and the measurement accuracy is  $1.0[cm]$ . The results clearly show that left from the measurement station the errors in the estimates are much larger. This could be expected, because the surge travels from left to right and thus everything seaward of the measurement station is not yet observed. The nonzero estimate left of the measurement station is mainly due extrapolation by the filter based on the time correlation of the AR(1) process.

### 5.5 Scaling of state variables

In the presentation of the RRSQRT algorithm above the choice of the state variables was assumed to be given. However, for any given model it is always possible to transform the state vector. Although the Kalman filter is invariant under transformation of the state vector the RRSQRT algorithm is not. This is mainly caused by the truncation.

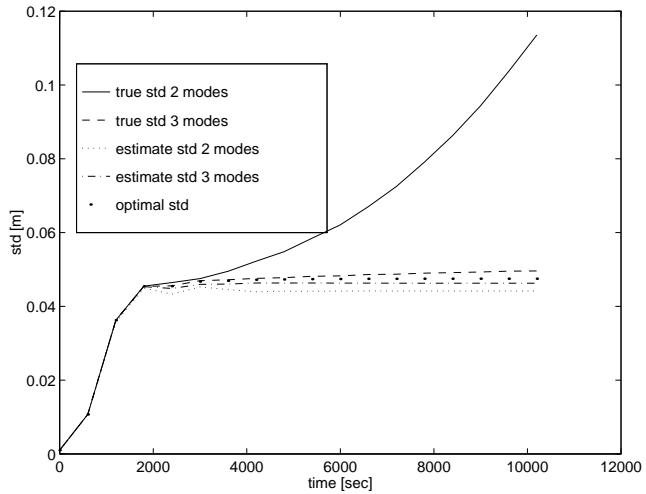


Figure 5.1: Approximate, optimal and true error covariance at the measurement point for several values of  $q$

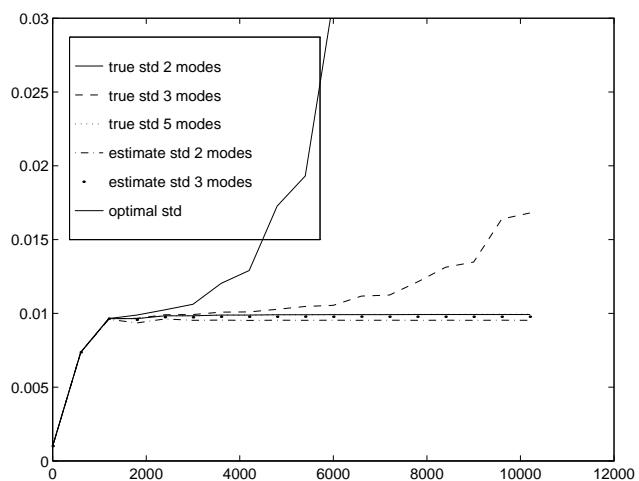


Figure 5.2: Approximate, optimal and true error covariance at the measurement point for several values of  $q$  for more accurate measurements

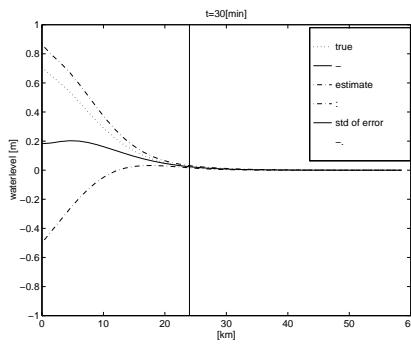


Figure 5.3: Estimated and true (generated) waterlevel at  $t = 30[\text{min}]$

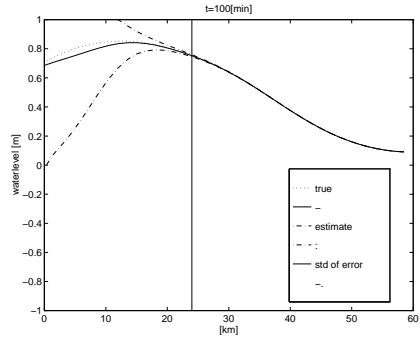


Figure 5.4: Estimated and true (generated) waterlevel at  $t = 100[\text{min}]$

The effect of a state transformation on the truncation-step can easily be shown. Let  $\tilde{\mathbf{x}} := \Psi\mathbf{x}$  be an arbitrary state transformation. Without any truncation the square root factor would be transformed as  $\tilde{\mathbf{L}} = \Psi\mathbf{L}$ . The corresponding inner product in the truncation-step is

$$\tilde{\mathbf{L}}'\tilde{\mathbf{L}} = \mathbf{L}'(\Psi'\Psi)\mathbf{L} \quad (5.38)$$

When an eigendecomposition is applied to this matrix the transformation has a large influence. Scaling of the state variables is equivalent to choosing different weights for these variables in the truncation. Transformation with a unitary matrix has no influence on the truncation.

At the moment no practical method is known to compute the state transformation that results in an optimal performance of the RRSQRT algorithm. Some insight however can be gained by studying the problem from the model reduction perspective (see also section 4.2.7). In the standard model reduction problem the projection, which is the same as our truncation, is based on the covariance weighted by the 'energy' produced on the outputs. This can also be interpreted as a scaling based on the observability.

For the shallow water equations the scaling based on energy produced at the output can be used as a heuristic to find a practical scaling method. Roughly the state vector should be scaled such that all state variables become equally observable. One approach that uses this argument is based on the energy. The potential energy of a surface elevation  $\xi$  above the reference plane for one grid-cell is

$$E_\xi = 1/2g\xi^2\rho_w\Delta x\Delta y \quad (5.39)$$

and the kinetic energy of this grid-cell is

$$E_{u,v} = 1/2(u^2 + v^2)D\rho\Delta x\Delta y \quad (5.40)$$

Assume one measures surface elevations. Through propagation of the model kinetic energy may become potential energy and because the model is dissipative

the sum of the two can only decrease or at most remain the same. This suggests that scaling the state variables according to the energy they represent, creates approximately equal observability if the dissipation is small. In this case the water levels should be scaled with  $\sqrt{g}$  and the velocities with  $\sqrt{D}$ .

The scaling of variables in the state vector that correspond to noise variables is slightly more difficult. For example, consider a noise variable added to the open boundary. Assume these variables are defined for sections of the open boundary of length  $L$ . If this boundary noise is modeled as an AR(1) process only a fraction  $\alpha_b^2$  remains the next time-step. Thus a fraction  $(1 - \alpha_b^2)$  of the energy has gone into the model. The energy that enters the system through a change of the open boundary in one time-step can easily be measured with one model evaluation, which directly gives the scaling factor.

Experiments with the scaling of the state variables as described here indeed show a better performance of the RRSQRT algorithm.

## 5.6 Implementation

### 5.6.1 overall structure and concepts

In principle the implementation of the RRSQRT algorithm is not very difficult. The matrix operations can be translated more or less directly into code. A difficulty is however that the algorithm must often be connected to an existing model, that was not designed for use in a data assimilation system. As a result the state, inputs and outputs are often not very clear in the code nor documentation.

The general structure for the implementation should show a clear separation between the Kalman filter routines and the model routines. This can be seen in figure 5.5. Since the interaction between the filter algorithm and the model is not very close the interfacing can be kept at a rather abstract level. For example the Kalman filter does not distinguish different types of state variables and thus the interface can use a state vector. The filter routines need no information about the interpretation of the elements of this vector.

For a good communication between the Kalman filter and the model an interface should be designed. The Kalman filter routines do not have to 'know' much about the specific model. Thus it is most convenient to keep all other information hidden for the Kalman filter. From the Kalman filter end the interface can almost literally be taken as in equations 5.2, 5.3 5.4, 5.5, and 5.6 – 5.9. If the model is nonlinear equation 5.11 can be used for an extended Kalman filter or equation 5.16 for a second order accurate one. In fact the only information the Kalman filter needs is:

- the current estimate of the state vector ( $\hat{\mathbf{x}}(k|l)$ )
- the deterministic inputs ( $\mathbf{u}(.)$ )
- the measurements ( $\mathbf{y}(.)$ )

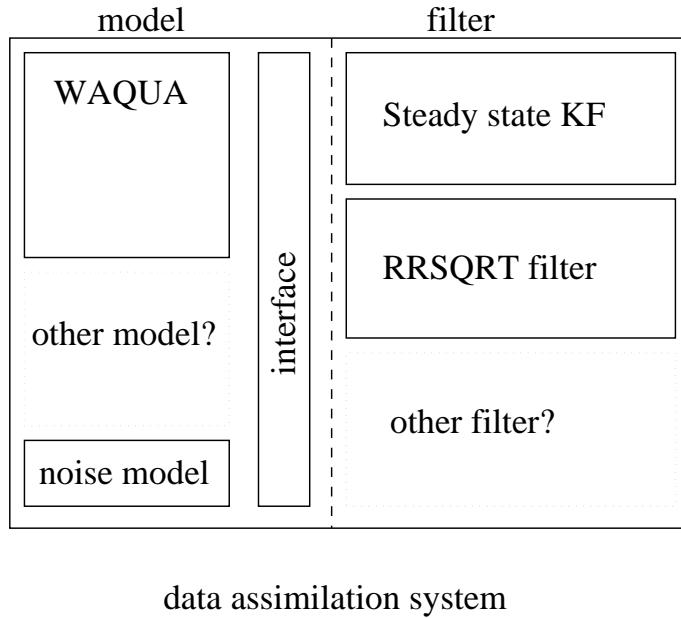


Figure 5.5: structure of the on-line data assimilation system

- the covariance of the system noise ( $\Sigma_s$ )
- the covariance of the measurement noise ( $\Sigma_o$ )
- the current time  $k$
- the time of the last measurements  $l$
- a function that computes the predicted value of a measurement from the current estimate of the state ( $\mathbf{g}()$ )
- the state transition function ( $\mathbf{f}()$ ) that is often available as a subroutine.

From the model side the of interface more detailed information is needed. For example the elements of the state vector should be linked to variables in the model (water levels, currents, etc.). Figure 5.6 shows this link between parts of the state vector and the variables for a 2-D shallow water model. The interface for the state vector consists of two copying routines and a routine to restore the dependent variables in the model. Sometimes it is also possible to use links (e.g. with pointers) instead of copying which can save some computation time.

Many models have internal variables that are there only to speed up the computations. For a shallow water flow model there may be for example arrays for the friction in  $x$  and  $y$ -direction. These arrays can be computed from the water level and a coefficient (Chézy or Manning). If the state is changed in the filter algorithm then these values are no longer consistent with the new water

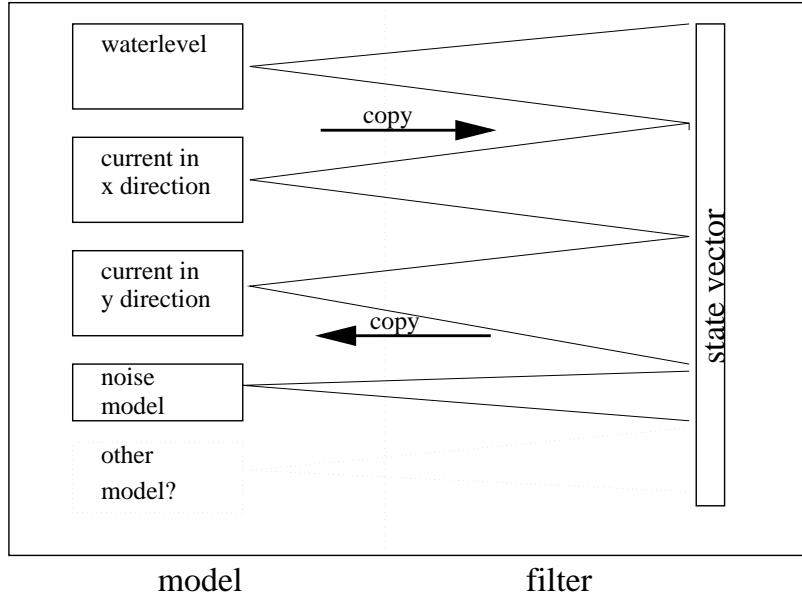


Figure 5.6: interfacing with the state vector

levels and should therefore be computed again. Similar to the state vector an interface is needed for the inputs ( $\mathbf{u}()$ ) and the outputs ( $\mathbf{y}()$ ).

An advantage of the implementation with a well defined interface is that new models or data assimilation schemes can be attached more easily (see also figure 5.5). This makes it possible to implement different assimilation algorithms. The user can then select the assimilation scheme that is most appropriate for the particular application. Also different models can be implemented without changes to the data assimilation algorithm. A possible implementation of the RRSQRT algorithm according to the principles described in this chapter is given in appendix ??.

### 5.6.2 The stochastic model

For data assimilation with a Kalman filter a stochastic model is needed. Often the stochastic model can be based on a deterministic numerical model of the physics. A separate model (noise model) can then be used to describe the uncertainty in the model. An important class of noise models are those where the errors in the forcing are considered to be the main source of errors. For this class a more or less separate noise model can be used to process the forcing before it is used in the deterministic model. This way the noise model can be added more or less separately to the deterministic model (see also figure 5.5). Since the model will often also be used without a Kalman filter it is important that the code of the filter and the model are as much separated as possible. The

filter routines should only be on the highest level in the code of the deterministic model.

### 5.6.3 Interface

If a deterministic model and a noise model for the forcing are available the interface can be designed and connected to it. The interface will contain several parts. First a mapping has to be defined between the variables that compose the state in the deterministic model and noise model on one side and the state vector for the filter on the other (see also figure 5.6). For every new model it is important to first realize what the state variables are for that model. This choice is not unique and one choice may be more convenient than another. Sometimes some variables can easily be computed from other variables. In that case it is often more efficient not to add these variables to the state vector but to recompute them in the interface routine.

For the actual exchange of data between the filter and the model several solutions are possible. One possibility is to copy the variables between the model state variables and the state vector. This model often results in the most readable code when implemented in FORTRAN. Another possibility is to associate the state variables in the vector with the variables in the model through a reference array. In FORTRAN it is not possible to hide this indirect referencing at higher levels, which makes the calls in the code less clear. An advantage of this solution is that it is faster since no values have to be copied. Several other languages are more convenient in this respect.

### 5.6.4 Testing

If implemented with a 'clean' interface the RRSQRT algorithm can be tested separately from the model. A very simple model can be used for these tests. If the number of modes is set equal to the state dimension the RRSQRT should produce exact results, which can be compared to the results of the 'full' Kalman filter for small models.

An important property of the RRSQRT filter is that the routine that advances the model one time step is called more than once every time step. During the design of a deterministic numerical model this possible use of the model is usually not considered. Therefore it can be expected that some changes have to be made before it becomes possible to call the model more than once on one time step. This step should be tested separately, i.e. the results of different iterations in one time step (with identical forcing) should be equal.

The last step of the implementation of the RRSQRT filter is to connect it all together. Since every part was tested separately the number of errors at this stage should be small. This is important, since the errors can be anywhere in the program in this stage, which makes correcting them more difficult. It is probably most convenient to use a twin experiment for this purpose. To speed up the computations a small grid should be used.

## 5.7 Reduction of computation times for the truncation step

When the RRSQRT algorithm is applied with a number of modes that is small then most of the computational effort is needed for the propagation in time of the columns of the square root factor  $\mathbf{L}_c$ , but with an increasing number of modes the effort needed for the truncation, which grows faster than linear in the number of modes, becomes important as well. Below several adaptations to the RRSQRT filter are proposed that reduce the computational burden of this part of the algorithm.

The truncation-step consists of three parts:

1. the inner product  $LTL = \mathbf{L}'_c \mathbf{L}_c$
2. computation of eigenvalues  $LTL: \mathbf{U} \mathbf{D} \mathbf{U}' = LTL$
3. orthogonalization  $\mathbf{L}_c: \tilde{\mathbf{L}}_c = \mathbf{L}_c \mathbf{U}$

If the dimensions of  $\mathbf{L}_c$  are  $n$  by  $q + m$  then the number of floating point operations (flops) needed in the three steps is:

1. the inner product  $LTL = \mathbf{L}'_c \mathbf{L}_c: 2n(q + m)^2$  flops
2. computation of eigenvalues:  $O((q + m)^2)$  flops
3. orthogonalization  $2nq(q + m)$  flops

For large models especially the first and third step require much effort. If  $m$  is much larger than  $q$  then the first step becomes dominant.

### 1. Storing several nearly time-invariant matrices

The matrix  $\mathbf{L}_c$  is composed of two parts after the time-step  $\mathbf{A}(k)\mathbf{L}_c(k+1|k)$  and  $\mathbf{F}(k)(\Sigma_s)^{\frac{1}{2}}$ . In some applications  $\mathbf{F}(k)(\Sigma_s)^{\frac{1}{2}}$  is almost time invariant. This requires that both the system noise is time invariant and the the system responds approximately linear and time-invariant to the disturbances caused by the system noise, at least in the first time-step. The validity of this assumption of course needs to be tested for every application separately, but assume that it holds. In the first place it is then no longer necessary to evaluate this term every time-step which may considerably reduce the number of computations in the time-step of the algorithm depending on the noise input dimension  $m$ .

Under the assumptions mentioned above it is also possible to reduce the number of computations needed for the inner product  $\mathbf{L}'_c \mathbf{L}_c$ . Just after the time-step the square root factor of the error covariance is

$$\mathbf{L}_c(k+1|k) = [\mathbf{A}(k)\mathbf{L}_c(k|k), \mathbf{F}(k)\Sigma_s(k)] \quad (5.41)$$

If the second part of this matrix remains constant then also part of the inner product remains constant. This time-invariant part of  $\mathbf{L}'_c \mathbf{L}_c$  can be stored, which reduces the number of computations to

- (a) inner product  $\mathbf{LTL} = \mathbf{L}'_c \mathbf{L}_c$ :  $2n(q^2 + qm)$  flops
- (b) computation of eigenvalues  $O((q + m)^2)$  flops
- (c) orthogonalisation  $2nq(q + m)$  flops

It is not necessary to make a copy of the part  $\mathbf{F}(k)\Sigma_s(k)$  if the truncation is carried out on the columns of  $\mathbf{U}$  before the multiplication in the third step instead of multiplying first. This also reduces the number of computations of the third step.

The modification proposed here has been applied to the applications of chapter 6 and 7. The differences between the results with and without the modification were small, while for the CSM model the reduction was quite large; 71% for the truncation step ( $q = 50, m = 121$ ).

## 2. Symmetry of the inner product

Because the matrix  $\mathbf{L}'_c \mathbf{L}_c$  is symmetric the number of computations in the product can roughly be reduced by 50% by computing only the elements below and on the diagonal. This reduction does not introduce any additional approximations and should therefore always be used. With this modification the number of operations in the three steps of the truncation become

- (a) inner product  $\mathbf{LTL} = \mathbf{L}'_c \mathbf{L}_c$ :  $nq(q + 1)^2$  flops
- (b) computation of eigenvalues:  $O((q + m)^2)$  flops
- (c) orthogonalization  $2nq(q + m)$  flops

It is also possible to exploit the symmetry of  $\mathbf{L}'_c \mathbf{L}_c$  when part of this matrix is stored. In that case the number of operations needed becomes

- (a) inner product  $\mathbf{LTL} = \mathbf{L}'_c \mathbf{L}_c$ :  $n(q^2 + 2qm)$  flops
- (b) computation of eigenvalues:  $O((q + m)^2)$  flops
- (c) orthogonalization  $2nq(q + m)$  flops

When applied to the CSM model the reduction of exploiting symmetry is almost 50%, when combined with the storage of part of  $\mathbf{L}'_c \mathbf{L}_c$  the reduction becomes 75% for the truncation-step.

## 3. A modification of the discretization of the system noise

With a minor modification of the introduction (discretization) of the system noise a large part of the matrix  $\mathbf{F}(k)\Sigma_s(k)$  can be made zero. This can be used to reduce the computation times as well as the storage requirements. In addition this matrix becomes time-invariant if the system

noise is, and thus this modification can easily be combined with the modifications mentioned above.

Assume that the system noise is added to the forcing of the model and let the deterministic model be denoted as

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k+1) \quad (5.42)$$

where the forcing of the model contains a deterministic part  $\mathbf{u}_d$  and some additional noise  $\mathbf{u}_n$ , thus

$$\mathbf{u}(k) = \mathbf{u}_d(k) + \mathbf{u}_n(k) \quad (5.43)$$

and assume that the system noise can be modelled with an AR(1) process, i.e.

$$\mathbf{u}_n(k+1) = \alpha\mathbf{u}_n(k) + \mathbf{w}(k) \quad (5.44)$$

Together the equations 5.42–5.44 form the stochastic system

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{u}_n(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(k) & \alpha\mathbf{B}(k) \\ 0 & \alpha\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}_n(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B}(k) \\ 0 \end{bmatrix} \mathbf{u}_d(k+1) + \begin{bmatrix} \mathbf{B}(k) \\ \mathbf{I} \end{bmatrix} \mathbf{w}(k) \quad (5.45)$$

For this system the square root factor  $\mathbf{L}_c(k+1|k)$  becomes

$$\mathbf{L}_c(k+1|k) = \left[ \begin{bmatrix} \mathbf{A}(k) & \alpha\mathbf{B}(k) \\ 0 & \alpha\mathbf{I} \end{bmatrix} \mathbf{L}_c(k|k) \quad \begin{bmatrix} \mathbf{B}(k) \\ \mathbf{I} \end{bmatrix} \Sigma_s^{\frac{1}{2}} \right] \quad (5.46)$$

In this example the forcing is on the implicit level of the discretization. If the forcing was used on the explicit level  $k$  then the square root factor would be

$$\mathbf{L}_c(k+1|k) = \left[ \begin{bmatrix} \mathbf{A}(k) & \mathbf{B}(k) \\ 0 & \alpha I \end{bmatrix} \mathbf{L}_c(k|k) \quad \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \Sigma_s^{\frac{1}{2}} \right] \quad (5.47)$$

The additional zeros significantly reduce the computations in the truncation-step of the RRSQRT algorithm. For systems with the forcing on the implicit level the augmented state can easily be adapted to create zeros similar to the explicit case. The only change needed is a shift in time of the noise part of the forcings.

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{u}_n(k+2) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(k) & \mathbf{B}(k) \\ 0 & \alpha\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}_n(k+1) \end{bmatrix} + \begin{bmatrix} \mathbf{B}(k) \\ 0 \end{bmatrix} \mathbf{u}_d(k+1) + \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \mathbf{w}(k) \quad (5.48)$$

Combined with the other modifications this gives the following number of computations

- (a) inner product  $LTL = \mathbf{L}_c' \mathbf{L}_c$ :  $nq(q+1) + 2qm^2$  flops
- (b) computation of eigenvalues:  $O((q+m)^2)$  flops
- (c) orthogonalisation  $2nq^2 + 2qm^2$  flops

With the combined modifications the reduction in the truncation-step becomes 95.6% for the CSM8 model. The reduction of the computational burden for the complete algorithm for this model on a HP workstation is given in the following table.

modification	relative computation time
standard RRSQRT	100%
symmetry and storage	75%
all three combined	45%

A disadvantage of the proposed modification in the discretization of system noise is that the algorithm becomes more sensitive to the choice and scaling of the state variables. The experiments so far indicate that at least the state variables corresponding to noise variables must be scaled to their relative importance.

## 5.8 Conclusions

In this chapter a new suboptimal scheme is proposed that reduces the computational burden by orders of magnitude compared to the 'full' Kalman filter. The algorithm is based on the approximation of the error covariance by a matrix of a lower rank. A square root decomposition of the error covariance is used to successfully exploit the reduced rank structure of the approximation. Moreover, the square root decomposition ensures that the error covariance remains positive at all times and the lower condition number of square root type decomposition increases the numerical stability.

In addition two extensions of the algorithm for nonlinear models are proposed. One is first order accurate and the other at twice the computational cost is second order accurate. Both methods do not require a tangent linear model but instead use the forward model iteratively which significantly simplifies the implementation of the algorithm.

Several properties of the algorithm are shown and illustrated with the 1-D wave example. It is shown that under some restrictions the results of the RRSQRT algorithm converges to those of the 'full' Kalman filter. The type of behavior found in the proof is shown to be consistent with the observations for the example. Finally several implementational issues are addressed.

## Chapter 6

# Application of Kalman filtering to two dimensional shallow water flow: storm surge forecasting

### 6.1 Introduction to the application

To protect the densely populated Netherlands from storm surges many dikes and barriers were constructed. For the large barriers in the Eastern Scheldt and 'Nieuwe Waterweg' accurate predictions for waterlevels are needed 6 hours in advance to decide whether these barriers have to be closed or not. Also for the protection of the dikes and for the ships entering the harbour at Rotterdam these predictions are needed.

At the moment the predictions for waterlevels during storm surges are computed using a two-dimensional shallow water flow model of the North Sea and a steady state Kalman filter algorithm to assimilate waterlevel measurements into the model to improve the model forecasts [40, 42]. The number of additional computations needed for data assimilation with the steady state Kalman filter is very small. Provided that the model is time invariant and approximately linear this procedure works quite well and has been used on a routine basis for some years. To speed up of the (off-line) steady state gain computations a Chandrasekhar type algorithm [59, 40, 7] was applied.

In this chapter a time varying Kalman filter is developed that can incorporate the non-linearities of the model. The filter is based on the RRSQRT algorithm and the finite difference approach described in Chapter 5. The method will be shown to work and the results are compared with the results of the steady state filter.

Although the shallow water flow model can be approximated reasonably well

by a linear model it contains some non-linearities and therefore an extended Kalman filter could in principle give more accurate results than the present steady state filter. An example of a non-linearity in the model is the friction at the sea bed. But also non-stationarity is important. For example errors in the wind forcing depend on the meteorological state and the wind friction coefficient depends on the mean wave-height which varies during a storm. However, on the scale of the continental shelf the impact of these non-linearities is not very large and therefore the difference in performance between a steady state filter and an extended Kalman filter is expected to be small. Only changes in the noise model that capture the time varying nature of the uncertainty better are expected to result in significant improvement of the performance. The main reason for the experiments in this chapter is therefore not to improve the filter performance, but to validate the performance of the RRSQRT filter.

Although, statistics of the uncertainty in the model are assumed to be the same for the steady state filter and the RRSQRT filter the parameter values of the error covariance can not be taken directly from the steady state filter. The main reason for this is that these values were obtained from calibration using the steady state model. Here, a method is proposed to calibrate these parameters for the RRSQRT filter. Using this method, covariance matching, it is also possible to gain insight in the structure of the model errors. It is shown that the true residuals behave different from the statistics computed by the Kalman filter. An adaptation to the structure of the errors is proposed to improve the noise model used by the Kalman filter.

Another way to further optimize the performance of the Kalman filter is to change the configuration of the measurements. To some extend the number and selection of tide gauges that are assimilated into the model is not fixed. Ideally including all available measurements would result in the highest accuracy. However, in practice not all data is of the same quality. Moreover, some locations are not captured very well by the model, for example due to the finite grid size. Finally the theoretical increase of accuracy of two tide gauges only separated by a small distance over one is very small and does not balance the additional cost of gathering the measurements and the cost of the increased computation time. It is expected that a careful selection of the configuration of the measurements can contribute to an improved performance of the Kalman filter for this application.

## 6.2 A deterministic model for storm surge prediction

In order to obtain estimates that are consistent with physical laws like conservation of mass and momentum the stochastic model is based on a deterministic model that reflects these laws. For storm surge prediction the two dimensional shallow water equations can be used for the deterministic model [67, 73]. The

conservation of mass and momentum are expressed in this model as

$$\frac{\partial \xi}{\partial t} + \frac{\partial Hu}{\partial x} + \frac{\partial Hv}{\partial y} = 0 \quad (6.1)$$

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + g \frac{\partial \xi}{\partial x} - fv + \frac{gu\sqrt{u^2 + v^2}}{C^2 H} \\ - C_d \frac{\rho_a}{\rho_w} \frac{V^2 \cos \psi}{H} + \frac{1}{\rho_w} \frac{\partial p_a}{\partial x} = 0 \end{aligned} \quad (6.2)$$

$$\begin{aligned} \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + g \frac{\partial \xi}{\partial y} + fu + \frac{gv\sqrt{u^2 + v^2}}{C^2 H} \\ - C_d \frac{\rho_a}{\rho_w} \frac{V^2 \sin \psi}{H} + \frac{1}{\rho_w} \frac{\partial p_a}{\partial y} = 0 \end{aligned} \quad (6.3)$$

where:

$x, y$	= cartesian coordinates in horizontal plane
$u$	= depth-averaged current in $x$ direction
$v$	= depth-averaged current in $y$ direction
$\xi$	= water level above the reference plane
$D$	= water depth below the reference plane
$H = \xi + D$	= total water depth
$g$	= gravity acceleration
$f$	= coefficient for the Coriolis force
$C$	= Chezy coefficient
$V$	= wind velocity
$\psi$	= wind angle with respect to the positive $x$ -axis
$C_d$	= wind friction coefficient
$p_a$	= air pressure at the surface
$\rho_w$	= density of sea water
$\rho_a$	= density air at the surface

A shallow water model usually has two or three types of boundaries. At land-water boundaries the normal flow is set to zero. At the open boundaries no physical boundaries exist and thus artificial ones will have to be specified. Often the surface level or the flow is prescribed at these boundaries, but also non-reflecting boundaries are used [67]. The waterlevels are often specified by their harmonic constituents.

When meteorological activity is low and there are no external surges, this deterministic model is quite accurate. In these cases the Root Mean Square (RMS) error in the waterlevels is approximately 15 cm. In case of storm surges the accuracy is less and it is believed that the main sources of error are the open boundary condition and the wind input as provided by the meteorological model.

These equations are discretized using an Alternating Directions Implicit (ADI) method and a staggered grid that is based on the method by Leenderstse and Stelling [67]. This numerical model was available by means of the WAQUA package [57].

## 6.3 Stochastic extension of the deterministic model and measurements

Before a Kalman filter can be applied a description of the errors in the model and the measurements are needed, since the covariances of the errors determine how the model predictions and the measurements will be weighted.

An important tool for the description of the errors in the model and measurements are ARMA processes [8]. An ARMA model maps a white noise process to an error process with the desired shape of the autocorrelation function. ARMA models are also easily expressed in state space form, i.e. as a Markov process.

For storm surge forecasting in the North Sea it is assumed that errors in the forecast are mainly caused by the uncertainty at the open boundary and in the meteorological forcing, i.e. wind stress and pressure gradients. The covariances of these errors will be modeled using ARMA models. The noise model is very similar to the one used by Heemink [40], which is part of the dutch operational storm surge warning system.

### 6.3.1 Uncertainty at the open boundary

The model area will usually contain several boundaries that have no physical interpretation but that are there since they bound the area of interest. Often it is just not possible or necessary to model the whole body of water connected to the area of interest. To avoid these 'open' boundaries it would for instance be necessary to model the atlantic ocean if the area of interest is the North Sea. A difficulty is that the waterlevel and current as a function of time are not very well known at the open boundary. However, the numerical model requires a boundary condition at these boundaries. During calm weather the waterlevel can be approximated by a set of harmonic constituents (see e.g. [36] and example 1 ). The amplitudes and phases of the components only have to be given once. It is common practice to estimate these harmonic constituents from tide gauge measurements at the model interior. The change of the waterlevel due to meteorological effects remains unknown.

Here the meteorological set-up at the open boundary is modelled as an AR(1) process in time [8]. Thus for any point at the open boundary the waterlevel is given by:

$$\Delta\xi_b(k) = \alpha_b\Delta\xi_b(k-1) + \varepsilon_b(k) \quad (6.4)$$

where  $\varepsilon_b(\cdot)$  is a zero mean white noise process. In space the set-up is specified only at several points. The set-up at the other points of the open boundary are computed using linear interpolation. This is similar to the treatment of open boundaries in WAQUA.

### 6.3.2 Uncertainty of the meteorological forcing

Although meteorological forecasts of the wind are becoming more and more accurate the wind forcing is still a major source of error for storm surge forecasts

using shallow water models. But even if wind forecasts were very accurate the model that translates them into stresses on the water surface is not. In addition the coefficient  $C_d$  in the model is not known accurately, nor is it believed to be really a constant. Together, these effects cause an error in the surface stress that is used in the model.

To model the uncertainty a noise process is added to the depth averaged momentum equations. This process is constructed using an AR(1) process for the correlation in time.

$$\Delta u(x, y, k) = \alpha_w \Delta u(x, y, k - 1) + \varepsilon_u(x, y, k) \quad (6.5)$$

$$\Delta v(x, y, k) = \alpha_w \Delta v(x, y, k - 1) + \varepsilon_v(x, y, k) \quad (6.6)$$

The spatial correlation is given by

$$\frac{E[\Delta u(x_1, y_1, k)\Delta u(x_2, y_2, k)]}{\sqrt{E[\Delta u(x_1, y_1, k)^2]}\sqrt{E[\Delta u(x_2, y_2, k)^2]}} = e^{-\sqrt{(x_1-y_1)^2+(y_1-y_2)^2}/d_w} \quad (6.7)$$

$$\frac{E[\Delta v(x_1, y_1, k)\Delta v(x_2, y_2, k)]}{\sqrt{E[\Delta v(x_1, y_1, k)^2]}\sqrt{E[\Delta v(x_2, y_2, k)^2]}} = e^{-\sqrt{(x_1-y_1)^2+(y_1-y_2)^2}/d_w} \quad (6.8)$$

$$\frac{E[\Delta u(x_1, y_1, k)\Delta v(x_2, y_2, k)]}{\sqrt{E[\Delta v(x_1, y_1, k)^2]}\sqrt{E[\Delta v(x_2, y_2, k)^2]}} = 0 \quad (6.9)$$

where

$$\sqrt{E[\Delta u(x_1, y_1, k)^2]} = \sqrt{E[\Delta v(x_1, y_1, k)^2]} = \sigma_w \quad (6.10)$$

and  $d_w$  is a characteristic distance for the correlation. To reduce the number of noise inputs the noise is first introduced on a coarser grid and subsequently interpolated. The interpolation is performed using bilinear interpolation.

### 6.3.3 State-space formulation of model and measurements

For the assimilation of measurements by means of a Kalman filter into the model just described a state-space form of the stochastic model and the measurement relation is needed. The RRSQRT filter was implemented with a clear interface between the algorithm and the model according to the principles of section 5.6.

The state vector was defined as the water levels and current velocities at the computational cells, i.e. grid cells that are permanently dry are not included. Several variables were added to the state vector for the support points of the noise at the open boundary and the wind noise. the deterministic part of the input  $\mathbf{u}$  consist of the tidal forcing at the open boundary and the meteorological forcing, wind velocity and atmospheric pressure. The measurement relation for the tide gauges was taken simply as the nearest water level in the model.

## 6.4 Experiments with a linear model

### 6.4.1 Experimental design

To evaluate the performance of the RRSQRT algorithm some experiments with a simplified linear model were performed. The measurements were generated using the same linear model as is used for the Kalman filter, i.e. a twin experiment. Contrary to using field data this allows for comparison between the true state and the estimate of the state. Although the RRSQRT algorithm is especially suited for non-linear and time-varying models, a linear time-invariant model was used, so that results can be compared with the optimal estimate, which in this case can be obtained using a Chandrasekhar filter algorithm.

The simplified linear model was obtained by linearization of the 2-D shallow water water model of section 6.2 around the equilibrium  $(\xi, u, v = 0)$ .

$$\frac{\partial \xi}{\partial t} + D \frac{\partial u}{\partial x} + D \frac{\partial v}{\partial y} = 0 \quad (6.11)$$

$$\frac{\partial u}{\partial t} + g \frac{\partial \xi}{\partial x} - fv + \frac{c_f u}{D} - \frac{\tau_x}{D} = 0 \quad (6.12)$$

$$\frac{\partial v}{\partial t} + g \frac{\partial \xi}{\partial y} + fu + \frac{c_f v}{D} - \frac{\tau_y}{D} = 0 \quad (6.13)$$

where  $c_f$  is the coefficient for the linear friction. The friction term does not arise automatically in the linearization but it is necessary to include such a linear friction since it is the only dissipative part of the model.

For simplicity the effects of wind and errors in the wind are neglected in all but the last experiment. The remaining uncertainty on the open boundary is modelled using an AR(1) model. The system noise is inserted only at a few points on the boundary. For points in between the values are interpolated.

The area of the southern North Sea covered by the coarse linear model is shown in figure 6.1. The grid size used is  $\Delta x = 6400[m]$  and  $\Delta y = 6400[m]$ . This results in a 95 by 60 grid with 2669 computational gridpoints. The time-step is  $\Delta T = 1800[s]$ . The friction coefficient is  $c_f = 2.4 \cdot 10^{-3}$ . The system noise at the northern boundary is generated by an AR(1) process with correlation  $\alpha_b = 0.9$  over one time-step and white noise with standard deviation  $\sigma_b = \sqrt{0.2}[m]$ . This results in a standard deviation of  $0.46[m]$  of the waterlevel at the boundary. The noise is generated at four points ( $m = 2, 30, 60, 93, n = 60$ ) with linear interpolation in between. The measurements are generated at A ( $m, n = (8, 55)$ ), B ( $m, n = (10, 40)$ ) and C ( $m, n = (40, 18)$ ). The measurement errors are assumed to be independent with standard deviation  $\sigma_m = 0.1[m]$ . The initial estimate of the state is 0. The initial error covariance is also 0. The number of modes is  $q = 20$ .

### 6.4.2 Experiment 1: Tracking of a 'Kelvin' wave

As a first test a positive Kelvin wave is generated at the western part of the open boundary. Snap-shots of the true and estimate waterlevels are shown in

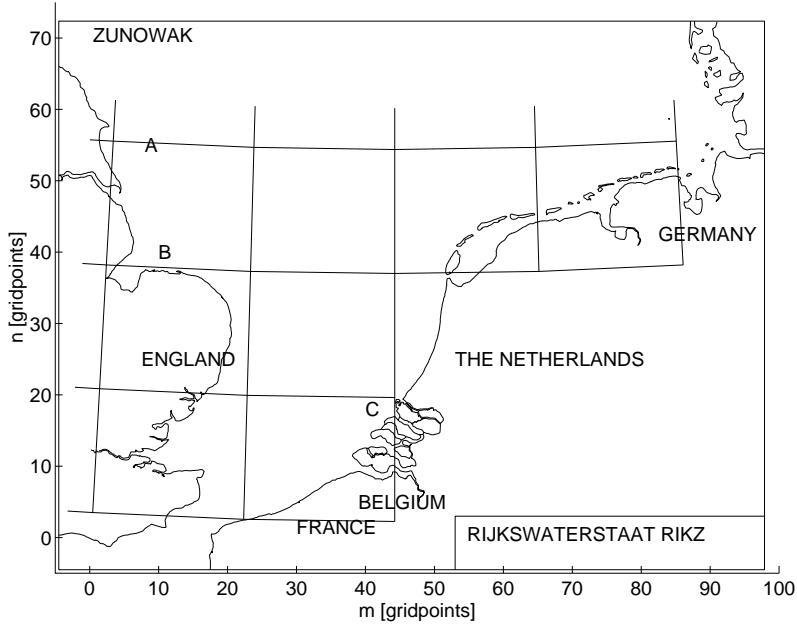


Figure 6.1: Area covered by the ZUNOWAK model

figure 6.2. Only measurement station A was used for the assimilation. The figure shows that without noise the the filter is able to track the state with only one measurement station. The accuacy is surprisingly high for only one measurement station. Without measurements the model would not be able to capture the incoming wave at all. The main reason for this high observability is the fact that the wave is forced to the right along the coast by the Corriolis force. The Kalman filtering approach as well as the RRSQRT algorithm seem to be working well.

#### 6.4.3 Experiment 2: Accuracy of the RRSQRT algorithm

For application of the RRSQRT algorithm it is very important to know the number of modes needed to obtain accurate results. In this second experiment the approximation errors, or loss of accuracy, caused by the RRSQRT algorithm is investigated. For this experiment system noise and measurement noise were generated using a random number generator, with variance according to the values above. Where possible the same realization of the noise is used for all experiments. Again only measurements from station A were used. Figures 6.3, 6.4 shows the RMS error for various settings of the number of modes  $q$ , as well as the optimal Kalman filter estimate, which in this special case of a time invariant linear model with zero initial noise can be computed efficiently using the Chandrasekhar algorithm. The 'true' RMS values are computed using, i.e.

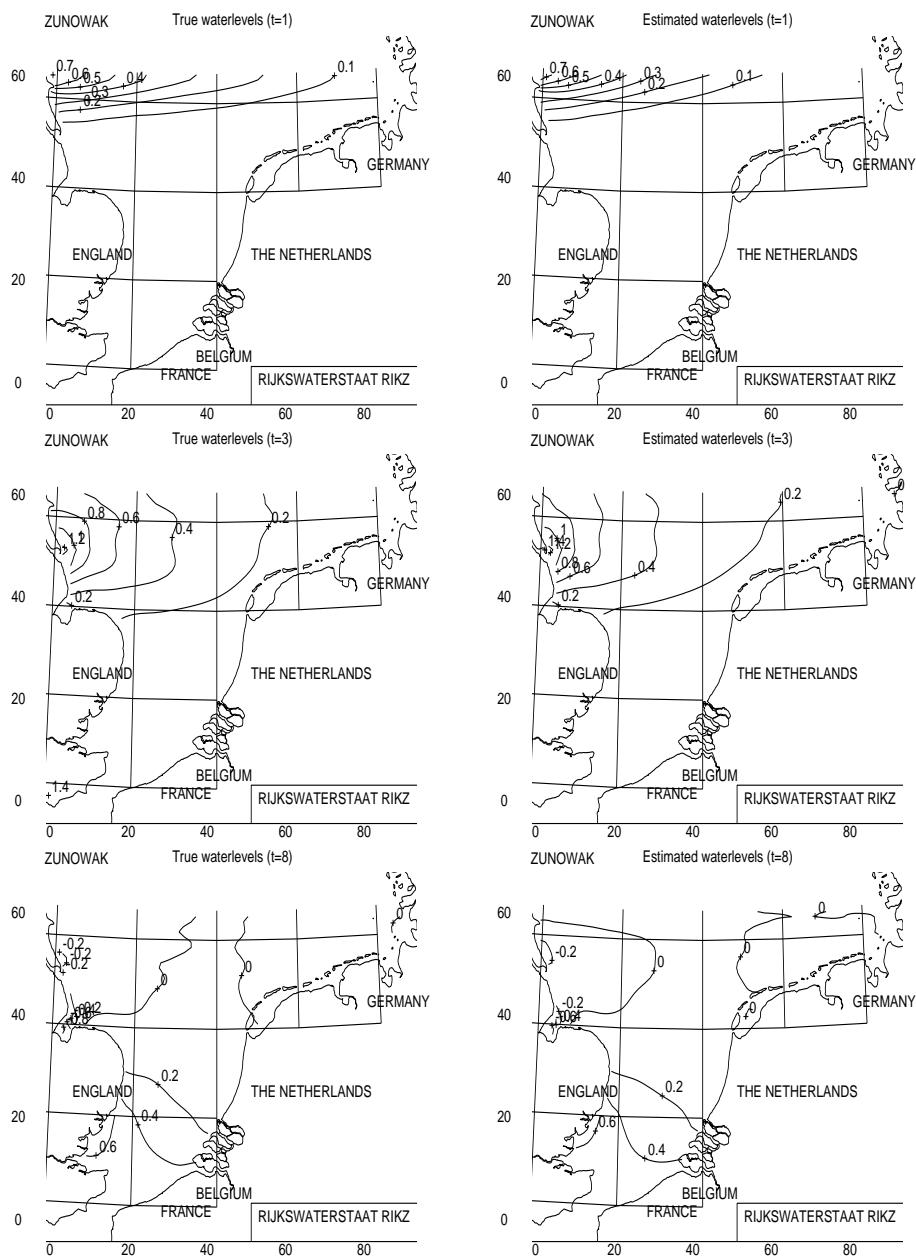


Figure 6.2: True and estimated waterlevels

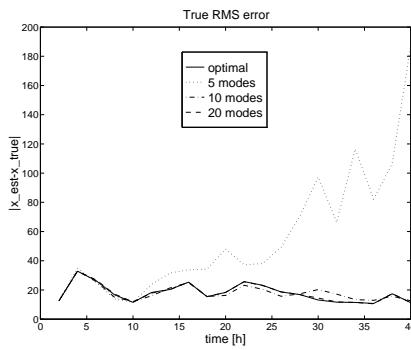


Figure 6.3: True RMS

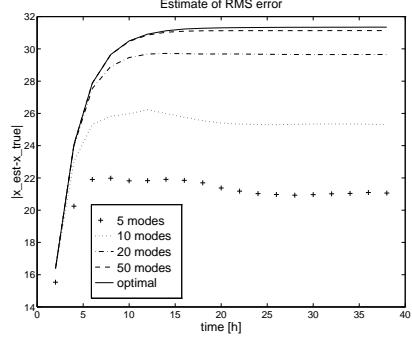


Figure 6.4: Estimated RMS

the Euclidean norm of the error in the estimate.

$$\epsilon(k) = \sqrt{(\hat{\mathbf{x}}(k|k) - \mathbf{x}(k))'(\hat{\mathbf{x}}(k|k) - \mathbf{x}(k))} \quad (6.14)$$

This shows how well the filter is doing. How well the filter 'thinks' it is doing can be seen from the computed error covariance  $\mathbf{P}_e(k|k)$ . The error  $e$  can be compared with the estimate  $e'$  given by

$$e'(k) = \sqrt{\text{trace} \mathbf{P}_e(k|k)} \quad (6.15)$$

which can be computed from the sum of the eigenvalues in the RRSQRT algorithm. For the 'full' Kalman filter the expectation of  $e(k)$  is  $e'(k)$ . The actual value of  $e(k)$  depends on the value of the pruedo random number used in the experiment. In figures 6.3,6.4 the true and estimated RMS values are shown for several number of modes.

The results show that for 5 modes the system with filter is probably unstable, for 10 modes the filter works well and for 20 modes the results are almost the same as for the full Kalman filter. The RRSQRT algorithm systematically underestimates the errors it is making, but with more modes the estimated error variance grows to the value of the full Kalman filter.

In figure 6.5 the computed eigenvalues are shown for various numbers of modes. These plots show that the RRSQRT algorithm mainly underestimates the errors in the smaller (faster) modes. The large range in eigenvalues is the reason accurate low rank approximations of the error covariance can be made.

Since the optimal gain can be computed in this special case it is possible to determine the relative error in the gain. The Frobenius norm was used for these computations. The results are shown in figure 6.6.

An important aspect of the RRSQRT algorithm is the number of modes needed to get a good approximation because the computation time is approximately proportional to this number. The number of modes can be used as a trade off between the amount of computation needed and the approximation error of the algorithm. The number of modes needed will also depend on the

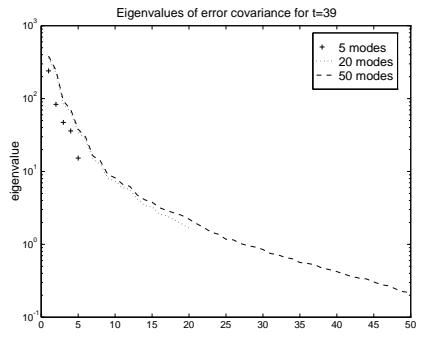


Figure 6.5: Eigenvalues of the approximate error covariance matrix

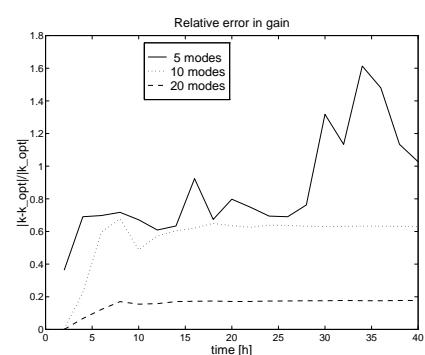


Figure 6.6: Relative error in the gain matrix

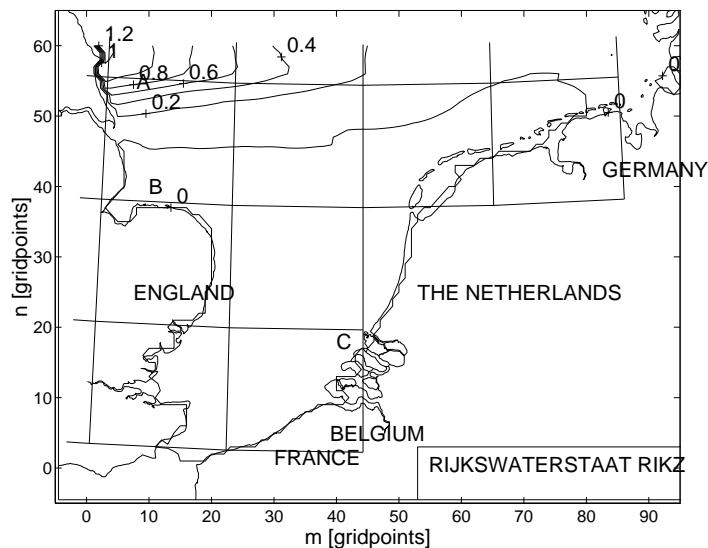


Figure 6.7: Part of the gain associated with the waterlevels

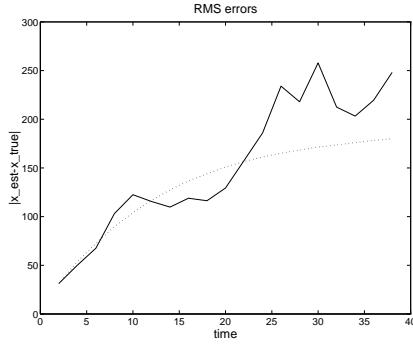


Figure 6.8: True and estimated RMS for filter with wind associated uncertainty

model used, the values of the parameters, the position and number of measurements and the system noise and measurement noise. This number is not known advance, but has to be determined from experiments such as the experiments shown here. In the experiments shown above the number of modes can be chosen as  $q = 20$ , unless accurate pictures of the gain or standard deviation are needed. In that case  $q = 50$  modes will be enough.

#### 6.4.4 Experiment 3: incorporating uncertainty in the wind noise

So far all differences between model and measurements were attributed to uncertainty in the forcing at the boundary and uncertainty in the measurements. However for storm surge forecasting the uncertainty in the meteorological forecast (wind) can not be neglected and even if the wind was very accurate the stresses computed from these winds are not. In the next experiment additional uncertainty in the wind-stress is introduced. An AR(1) model was used to model correlation in time ( $\alpha = 0.9$ ) and an exponential correlation function was used for correlation in space with decorrelation length 19.5 gridpoints and driven by white noise with standard deviation 10.001 (eqn. 6.7). To reduce the number of noise input variables the uncertainty was introduced on a subgrid ( $m = 1, 24, 47, 60, 83, n = 1, 20, 39, 58$ ). Measurements from stations A,B and C were used for assimilation.

Figure 6.8 shows the true RMS and estimated RMS values for 50 modes. For fewer than 50 modes ( $q = 20, 30, 40$ ) the filter is unstable. For 100 modes the RMS values are very close to the optimal values. The additional uncertainty introduced by the wind requires the use of more modes, but with these additional modes the algorithm works well.

#### 6.4.5 Conclusions

The first conclusion that can be drawn from the experiments in that the RRSQRT algorithm works very well for this problem. The number of modes and thus the computation time is such that the computational burden is only a fraction of that of the 'full' Kalman filter. The amount of computation needed for the 'full' Kalman filter is comparable with  $q = n \approx 6000$  modes for the RRSQRT filter.

### 6.5 Experiments with the full CSM model

The operational storm surge forecasting system of the Netherlands is based on a steady state filter [41, 42]. This provides the opportunity to compare the RRSQRT filter results for this model with the results of the steady state filter. The two algorithms are based on a very similar model for the uncertainty in wind and the boundary condition. Thus in principle the results should be comparable. However the steady state filter is based on a different approximation of the Kalman filter than the RRSQRT algorithm. An important assumption for the steady state Kalman filter is that the model is nearly linear. The RSQRT algorithm introduces truncation errors by truncating to a small number of modes. Still if both approximations are accurate then the filter results should be similar.

In the following the experiments a comparison between the steady state filter and the RRSQRT filter is used to validate the performance of the RRSQRT filter.

#### 6.5.1 Experiment and results

The comparison will use the dutch continental shelf model. This model, that is successor of the one presently used for the operational forecasts, covers the continental shelf of the North Sea (see figure 6.9). The grid is based on spherical coordinates with an average size of the cells of 8 by 8 km. This CSM8 model is driven with 10 harmonic constituents at the open boundary. Since the open boundary is everywhere located on deep water the wind effects on the water level will be relatively small, which is the most important reason for drawing the open boundary where it is. The pressure however has a significant effect on the boundary. This effect is sometimes called the inverse barometer effect, because the water rises when the pressure is low. Because of this effect the forcing at the open boundary is corrected for the pressure differences. The wind input needed for this model was provided by the dutch meteorological institute KNMI.

Measurements of the waterlevel were available for 12 different stations. The measurement stations are marked as + in figure 6.9. The measurements of 8 stations were used by the Kalman filter and the 4 other stations were used for validation and not included in the filtering process. These validation stations are marked o in figure 6.9.

The filter parameters were set to the following values. Standard deviation of the waterlevel at the open boundary is 6.8 cm, with the time correlation

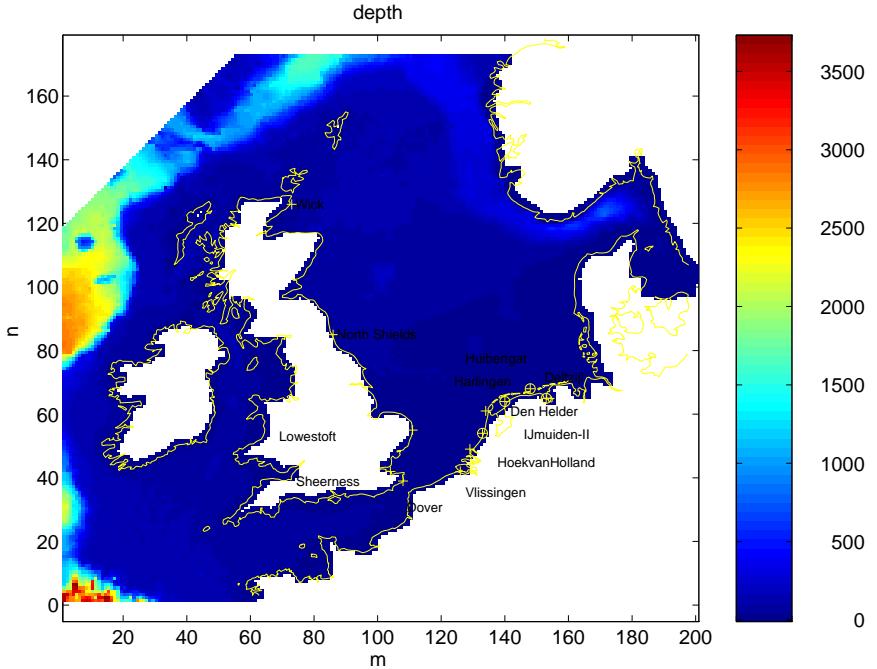


Figure 6.9: Area and depth of the CSM8 model

of  $\alpha_b = 0.9$  per timestep. Five independent AR(1) processes were located at the corners of the open boundary for this purpose. Between these points the waterlevels were interpolated in a linear fashion. The uncertainty in the wind, contrary to equations 6.7–6.9 was not added to the wind stress, but directly to the velocities of the model. This noise was introduced on a coarse grid with increments of 28 in both directions. A standard deviation of 0.0068 m/s per timestep was used together with a time correlation of  $\alpha_w = 0.9$  and a spatial correlation length of 26.38 grid-cells. For the RRSQRT filter  $q = 50$  modes were used. The experiments were carried out with field data, but with analysed wind. Thus the experiment resembles the operational setting, except for the fact that in that case forecasts of the winds are used. For the simulated period the storm of februari 1993 was selected.

For the RRSQRT filter only one model specific adaptation was needed other than implementation of the interface. The shallow water model allows for drying and flooding which is a very nonlinear effect that is in principle difficult to handle with a Kalman filter. For the CSM8 model a simple truncation of negative water depths was sufficient.

Figure 6.10 show some waterlevels for both the steady state filter and the RRSQRT filter for two filter stations (Hoek van Holland, Lowestoft) and 2 validation stations (IJmuiden, Harlingen). It can be seen that the results of

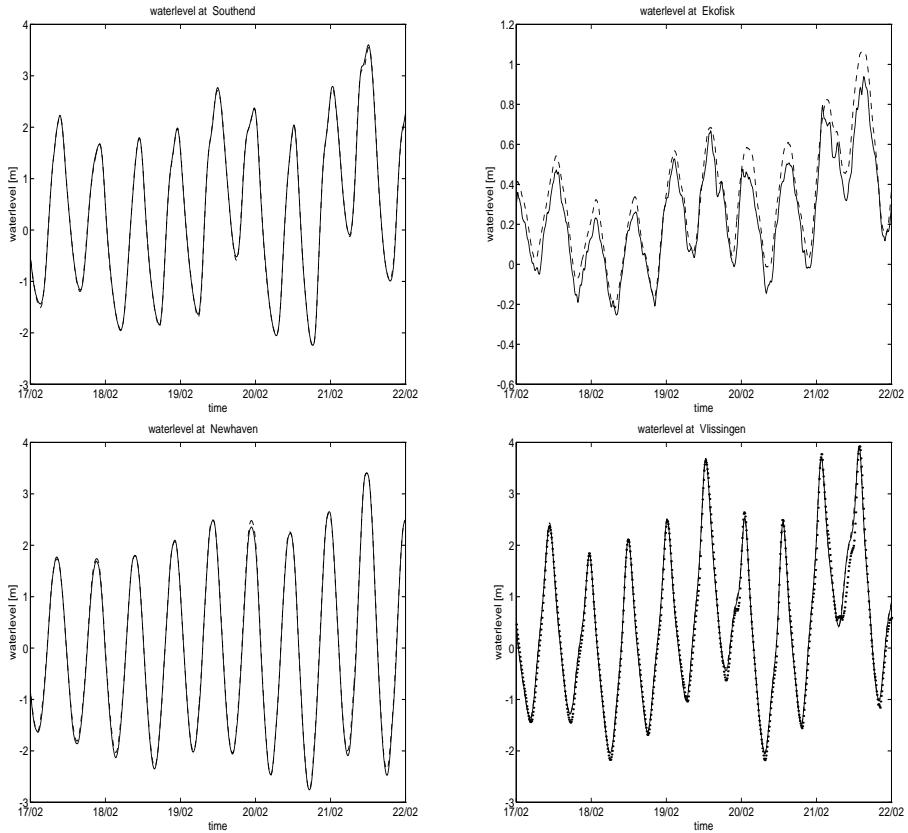


Figure 6.10: Waterlevels for steady state filter (dashed) and the RRSQRT filter (drawn line) versus measurements (dotted)

both filters are quite similar. This indicates that the RRSQRT filter is working well. Also, this suggests that both the truncation errors of the RRSQRT filter and the linearization errors of the steady state filter are quite small. From this figure we can also see that the RRSQRT does not perform significantly better (or worse) than the steady state filter. Thus although some of the non-linearities are resolved more accurately this this does not improve the overall performance.

### 6.5.2 Conclusions

The steady state filter and the RRSQRT algorithm give very similar results. It can therefore be concluded that the RRSQRT is working well for this application. It is very likely that the nonlinearities of the model in this application are not significant. Therefore the estimate computed by the steady state filter is nearly optimal and further improvement of the accuracy with the present model

is not likely.

## 6.6 Estimation of system noise and measurement noise

### 6.6.1 Introduction

Three ingredients are needed to build a Kalman filter application: a numerical model, measurements and a stochastic extension of the deterministic model. This stochastic embedding of the deterministic model is an important part. The system noise, that is added to the deterministic part, determines to a large extend the result of the Kalman filter. In the experiments above the formulation of the system noise for the RRSQRT filter was deliberately chosen the same as for the steady state filter to make comparison possible. However, the RRSQRT filter allows a much broader class of system noise specifications. This additional freedom can be used to improve the system noise specification. In this section a first attempt will be made to analyse the model errors and propose a new specification. Before this is done, the system noise parameters and the measurement noise are calibrated for the RRSQRT filter, which can give different results because the original calibration of the noise parameters was carried out with steady state filter [41].

### 6.6.2 Calibration of system noise and measurement noise

Various methods have been proposed to estimate system noise and measurement noise for Kalman filter applications (see [55] for an overview). Many of the traditional methods however require too much computation for large models such as the shallow water model studied here. Similar to the SOS's for the Kalman filter efficient approximations are needed. In addition, there is often not enough information available to estimate many parameters and it is necessary to use a very simple parametrization of the system noise. In general, the structure of the system noise in large scale models is considered an open problem [27].

Although, our knowledge of the system noise is still very limited, it is possible to build a working Kalman filter application with a very simple parametrization of the system noise. The number of parameters should be kept small because all these parameters have to be calibrated and often there is not enough data available to do this properly. Since the parameters are often chosen in such a way that they represent 'physical' quantities, like standard deviation and e-folding time-scales, a reasonable first guess can often be given (see e.g. [24, 27]).

In many large scale filtering applications the next step is a manual calibration of these parameters [34]. The performance of the filter can be measured with the covariance of the forecast errors or by using additional measurements. It is important not to evaluate the performance with residuals at measurement locations because these residuals can become small even if the filter is performing

badly. The filter parameters are calibrated by evaluating the filter performance for various values of these parameters.

A second approach is to compare innovations, i.e. measurement minus forecast, or the difference of measurement and model with the covariance computed by the filter. This covariance contains two components one caused by model errors and one by measurement errors. The covariance of the sum can be computed easily from the innovations. The difficulty is to separate the two. Fukumori [32] assumed that model signal, model errors and measurement errors are independent, which can be used to separate the components. Sometimes the measurement errors are known or relatively small. In that case the error covariance of the measurements can simply be subtracted. If one assumes that measurement errors are more or less uncorrelated while the model errors have a significant spatial correlation the off-diagonal elements of the error covariance matrix contain no component caused by measurement errors and can be used directly for estimation of the covariance of the model errors.

The innovations can even be used adaptively to improve the parameters, i.e. while the filter is running. Dee [26, 25] proposed an algorithm based on this approach.

In Kalman filtering literature adapting the parameters so that the computed covariance of the innovations resembles the true covariance of the innovations is known as covariance matching. It can be shown ([55] p. 109,121) this is a simplification of the maximum likelihood estimator when the sensitivity of the filter solution to the parameters is small.

### 6.6.3 Calibration of CSM8 noise with covariance matching

In the parametrization of the system noise, as given in section 6.3, there are 5 parameters:

- $\sigma_b$  standard deviation of error in waterlevel at the open boundary
- $\alpha_b$  time correlation of error in waterlevel at the open boundary
- $\sigma_w$  standard deviation of error in the wind forcing
- $\alpha_w$  time correlation of error in the wind forcing
- $d_w$  spatial correlation length (e-folding length) of error in wind
- $\sigma_m$  standard deviation of error in measurements forcing

In the sequel an attempt will be made to calibrate these parameters with covariance matching and to improve the parametrization of the system noise.

### 6.6.4 First guess of the standard deviations

As a first guess for the parameters the values from the steady state filter were used.

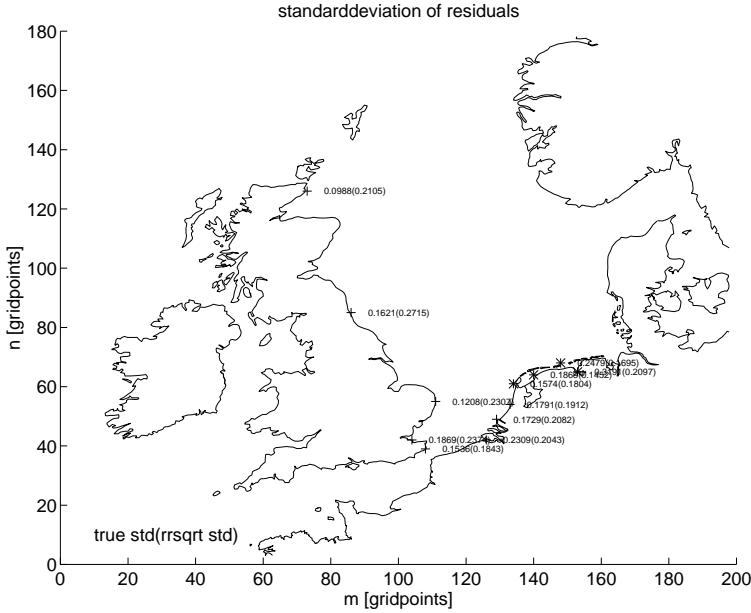


Figure 6.11: Standard deviation at measurement locations: noise for boundary and wind

parameter	value
$\sigma_b$	0.0688 [m]
$\alpha_b$	0.9 per timestep
$\sigma_w$	0.00688 [m/s/time-step]
$\alpha_w$	0.9 per timestep
$d_w$	26.3788 [gridcells]
$\sigma_m$	0.10 [m]

The timestep in the CSM8 model is 10 minutes. These values are not as accurate as the number of digits suggests, but in this way the filter is compatible with older versions. For example the standard deviation at the open boundary in older versions was given through the standard deviation of  $\varepsilon_b$ ,  $\sigma_\varepsilon = 0.03$  and  $\sigma_b = \sigma_\varepsilon \sqrt{1 - \alpha_b^2}$ .

In a first experiment the standard deviations of the residuals as estimated by the filter was compared with the true standard deviation. In this experiment the measurements were not assimilated, thus the variance of the residuals is just the sum of the variance of the model errors and the variance of the measurement errors.

Figure 6.11 shows these true and estimated standard deviations. It can be seen that the errors are overestimated in Scotland and underestimated in the Netherlands. The estimated standard deviations are quite constant over the model area, but the true standard deviations are larger in the southern North

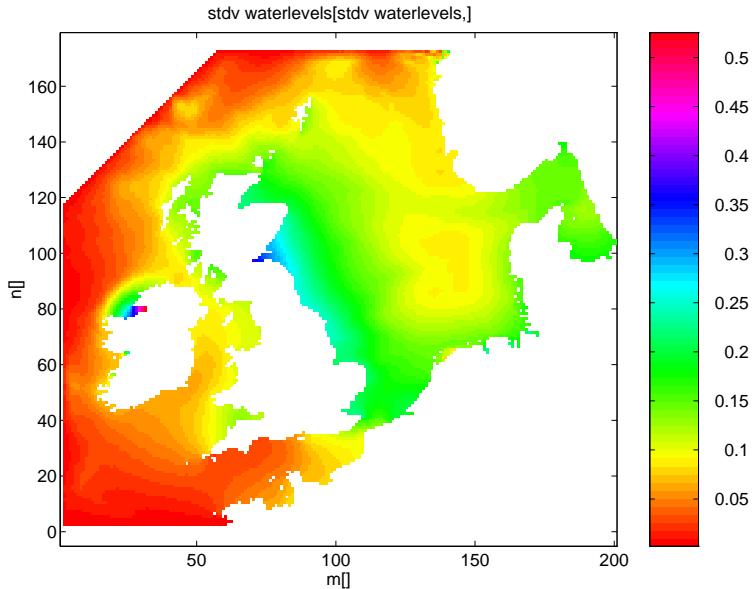


Figure 6.12: Standard deviation of model caused by wind

sea. The southward increase can be explained by the northwest wind during this storm. The effect of wind stress on the water surface accumulates in the direction of the wind, which also holds for errors in the wind. However, in the parametrization of the system noise, the errors in the forcing are independent of the wind direction. This was necessary of the steady state filter because dependence on the wind-direction implies time-dependence of the model. In the RRSQRT filter this is no longer a limitation.

Figure 6.12 and 6.13 show the estimated model standard deviations with only wind noise or only boundary noise included in the model. This indicates the relative size of the two components of the system noise. The estimated standard deviation due to the wind noise is much larger than the influence of the open boundary in the southern part of the north sea, while in Wick and the western part of the model they are approximately equal in size. Because the patterns of the standard deviation caused by wind en boundary are similar it will be difficult to separate the two based on the observed standard deviations.

### 6.6.5 Estimation of noise standard deviations

The first parameters to be estimated are the standard deviations of system noise, both at the open boundary and for the wind, and the standard deviation of measurement noise. The correlation lengths will be estimated subsequently. If necessary, the two groups of parameters will be adapted iteratively until a satisfactory estimate for both is obtained.

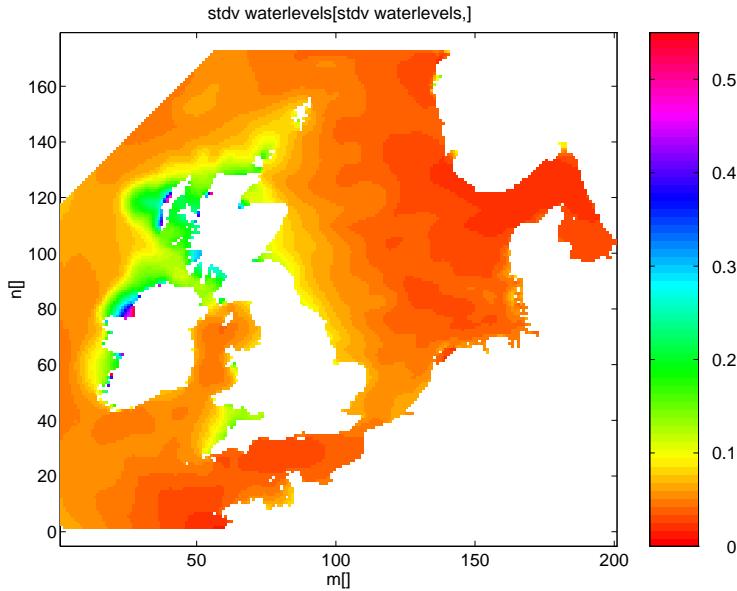


Figure 6.13: Standard deviation of model caused by boundary

If the model were linear then the error covariances due to wind, the open boundary and measurements would also behave in a linear fashion. This linearity can be used to estimate the standard deviations efficiently. Here the least squares method will be used for this purpose, i.e. the standard deviations of wind, open boundary and measurements are optimized with respect to the criterion

$$J(\sigma_w, \sigma_b, \sigma_m) = \sum_{i=1, j=1}^p ([P_o - P_c(\sigma_w, \sigma_b, \sigma_m)]_{i,j})^2 \quad (6.16)$$

where  $P_c$  is the computed covariance of the residuals and  $\mathbf{P}_o$  is the covariance of the observed residuals (for the februari 1993 storm). No measurements were assimilated. If the computed covariance is linearized the following linear least squares problem results

$$J(\sigma'_w, \sigma'_b, \sigma'_m) = \sum_{i=1, j=1}^p ([P_o - P_c - \sigma'_w \frac{\partial P_c}{\partial \sigma_w} - \sigma'_b \frac{\partial P_c}{\partial \sigma_b} - \sigma'_m \frac{\partial P_c}{\partial \sigma_m}]_{i,j})^2 \quad (6.17)$$

where  $\sigma'_w, \sigma'_b, \sigma'_m$  are the changes to  $\sigma_w, \sigma_b, \sigma_m$ . In this approach it is important that the model is nearly linear.

To check if the model is close to linear the covariance  $P_c$  with the present settings, but  $\sigma_m = 0$ , was compared with the sum of the two contributions wind and boundary computed separately. Figure 6.14 show the two results. The horizontal axis shows the index of the element of the 12 by 12 covariance matrix columnwise. Although there are significant differences the main pattern

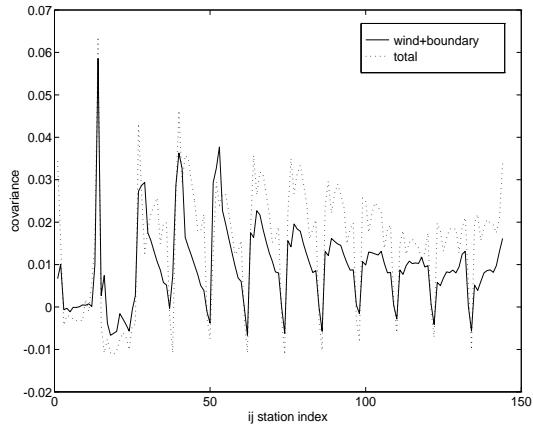


Figure 6.14: Controle lineariteit van het model

of both covariance matrices is the same. When using a linearization around the present estimate the linear least square estimate may be slightly different from the 'optimal' nonlinear estimate, but a good improvement per step can be expected.

Subsequently, the linear least squares method was applied using the criterion of equation 6.17 with a linearization around the present estimate  $\sigma_b = 0.0688[m]$ ,  $\sigma_w = 0.00688[m/s/time-step]$ ,  $\sigma_m = 0.10[m]$  and another linearization around zero, i.e.  $\sigma_b = 0[m]$ ,  $\sigma_w = 0[m/s/time-step]$ ,  $\sigma_m = 0.0[m]$ . The following table gives the estimates:

adaptation	linearized around 0	lin. present estimate
$\sigma_w$	-8%	0%
$\sigma_b$	0%	0%
$\sigma_m$	+19%	+55%

In both cases the measurement noise is increased significantly and the noise at the open boundary remains unchanged, but the estimate for the standard deviation of the wind noise is different for the two linearizations. Because the differences between the two linearizations are significant, the linearization around the present estimate is probably more accurate and a linearization around zero is not advised.

The least squares result is surprising when compared with figure 6.11. This figure shows that in average the standard deviation computed by the model is approximately the same as the observed value. There are large differences in the standard deviation of the observed residuals, but the average has the right size.

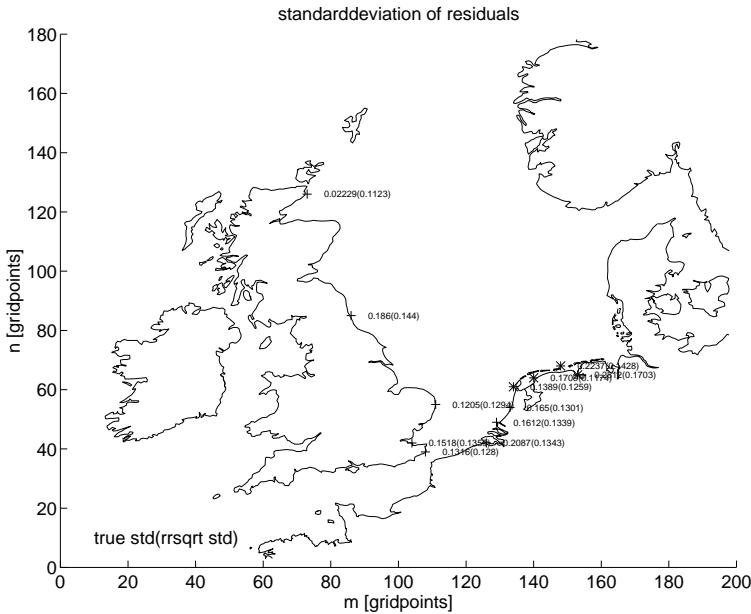


Figure 6.15: Standard deviations at measurement locations : only Wick assimilated

To investigate this discrepancy further an additional experiment was carried out in which Wick was assimilated.

In the preliminary experiment it was shown that an external surge can be eliminated effectively using only one measurement station. Thus, assimilating the measurements at Wick should eliminate the external surge. The remaining residuals are then caused by errors in the wind forcing and measurement errors. A comparison of observed and computed standard deviations of the residuals provides additional information about the parameter values. Figure 6.15 shows these standard deviation in the case measurements from Wick are assimilated. The computed standard deviation decreases significantly along the english coast, while the observed standard deviation does not. This indicates that the standard deviation for the open boundary is over estimated. The remaining standard deviations are all quite well predicted by the model. It is not possible to conclude that  $\sigma_b$  should be reduced based on this experiment alone because the noise at the open boundary was introduced for external surges and the storm surge of februari 21 1993 was not caused by an external surge.

Instead of assimilating only the measurements from Wick, the measurements of 8 stations were assimilated in the next experiment. Figure 6.16 shows the observed standard deviation of the residuals and the computed standard deviation of the innovations. For the validation stations both values are for the residuals. Since assimilated measurements are used to improve the estimate they can not be used to validate the performance, the validation stations however

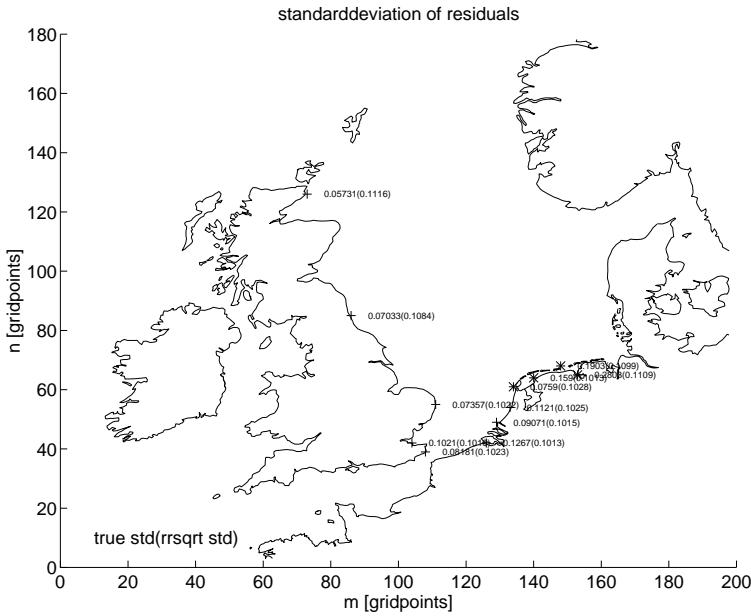


Figure 6.16: Standard deviations at measurement locations : 8 stations assimilated

can be used for this purpose. For the validation stations Harlingen, Delfzijl and Huibertgat the estimated standard deviation is too small, while at IJmuiden the estimate is quite good. Increasing the standard deviation of the measurements will increase the estimated standard deviation and thus decrease the difference between estimated and observed standard deviations for Harlingen, Delfzijl and Huibertgat.

#### 6.6.6 Estimation of spatial correlation

So far, only standard deviations have been estimated, but these are only part of the parameters. There are also parameters that specify the amount of correlation either in space or time. To estimate these parameters a slightly different approach than for the standard deviations is needed.

In general larger correlation in the systems noise, in time or space, will lead to larger spatial correlation for the waterlevels. In principle the correlation can then be estimated by comparing the estimated covariance and the observed covariance, but to do this a criterion for comparing spatial correlation is needed.

If the correlation is isotropic and homogeneous then the correlation is a function of the distance only. The correlations can then be plotted against the distance and a function (variogram) can be fitted. This procedure is similar to Kriging [18] and the approach followed by Dee [23]. These correlation functions can not be chosen at random because variances based on them should always

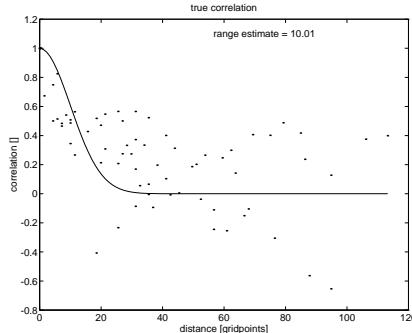


Figure 6.17: Observed correlation versus distance

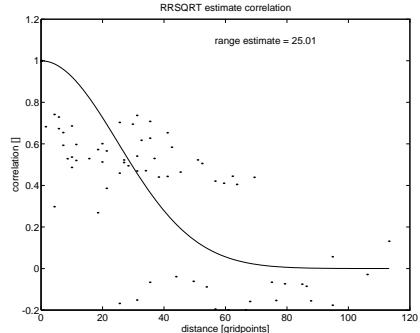


Figure 6.18: Estimated correlation versus distance

be non-negative. One example of a well known correlation function is Gaussian correlation

$$\rho(x_1 - x_2, y_1 - y_2) = \exp((x_1 - x_2)^2 + (y_1 - y_2)^2 / 2d^2) \quad (6.18)$$

The parameter for the correlation length  $d$  can for example be estimated with nonlinear least squares. In the application at hand the assumptions for a correlation function of this type are violated, because the variance depends the position and the correlation is affected by shallow and deep area's and by land-sea boundaries. But even though the method can not be applied the correlation length determined in this way is still a measure of the amount of correlation. When applied to the estimated and observed correlations it still provides a means of comparison.

Figures 6.17 and 6.18 show the observed and estimated correlations against the distance in grid-points. Also shown is the Gaussian correlation fitted to these points and the corresponding correlation length. It can be seen that the spreading of the points is quite large, which is partly caused by the short length of the series and partly by a violation of isotropy and homogeneity. The estimated correlation length is larger than the observed 'true' correlation, but there are too few points to make this significant with this large spreading. The number of points can be increased by using more storms or by adding more measurement stations, which can result in a more accurate estimate of the correlation length. With the present values there is not enough justification to adapt the correlation parameters.

### 6.6.7 Conclusions

The parameters that specify the uncertainty of model and measurements usually have to be found by calibration. Here a method is proposed that is based on covariance matching. The method is applied to the calibration of the RRSQRT Kalman filter for the dutch continental shelf model. The method confirms the

values of the parameters as they were found by previous manual calibration. With only one storm the results are not very accurate, as can be expected, more storms should therefore be used. If possible, also more stations should be included because more data results in more reliable and more accurate parameters.

When the estimated and observed standard deviations are compared it can be seen that the parametrisation of the system noise is not very accurate. There are large differences between the spatial patterns of the standard deviations that can not be altered by changing the parameters. The present system noise results in little variation of the estimated standard deviations, while the observed 'true' standard deviations increase in southward direction. One suggestion for further research is to make the uncertainty caused by the wind dependent of the direction of the wind. In the storm used here the wind was north-westerly which explains the larger uncertainty in the southern North sea.

## 6.7 Positioning measurements

So far, the number and position of the measurements has been considered given. In practice it is sometimes possible to add or move measurement equipment. This is however restricted by the available budget, which makes it necessary to position the equipment in such a way that a maximum amount of information can be extracted.

The impact of adding, removing or moving stations can be studied by comparing the estimated covariances as computed by the RRSQRT filter. For example, charts showing the standard deviation of the waterlevel in space can be used for this purpose. Figures 6.19 and 6.20 show the impact of the present 8 measurement stations. It can be seen that the impact is mainly in the region where the measurements are taken. A second observation is that the assimilation decreases the errors everywhere, which can easily be proven (for linear models).

Using these charts of the standard deviation of the error it is possible to study the effects of removing some of the observations. Figure 6.21 shows for example the standard deviation if the tide gauge at Wick would be removed. The estimated standard deviation increases in a rather large region around Wick. The region where the estimate remains (very) accurate becomes much smaller, which reduces the time interval over which accurate forecasts can be made. This is amplified by the fact that the tidal wave travels along the coast because of the Coriolis force.

Figures 6.22, 6.23 and 6.24 show the impact of removing the observations from North Shields, Lowestoft and Vlissingen respectively. The impact of these stations is clearly much smaller than that of Wick. Still the stations North Shields and Lowestoft contribute significantly to the total variance reduction. The impact of Vlissingen is negligible and it is preferable to remove at least one of the stations at the Dutch coast from the assimilation.

This approach can easily be extended to include non-existing measurement

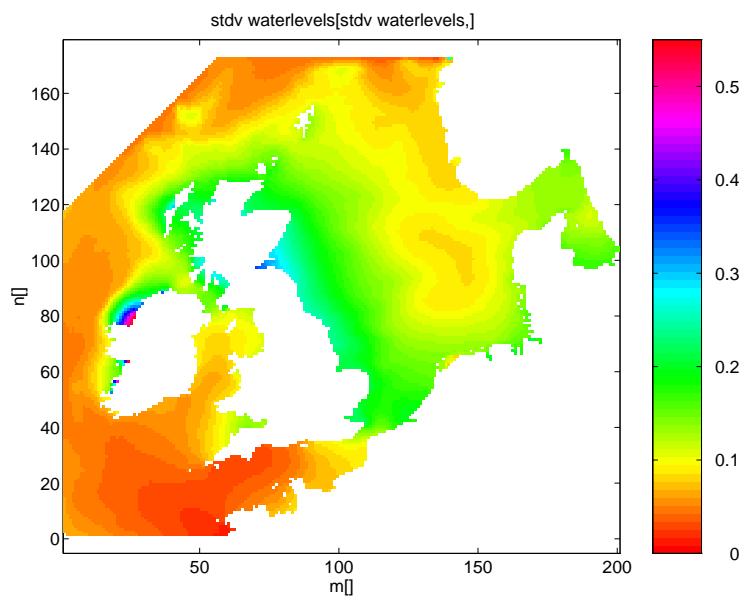


Figure 6.19: Standard deviation of model

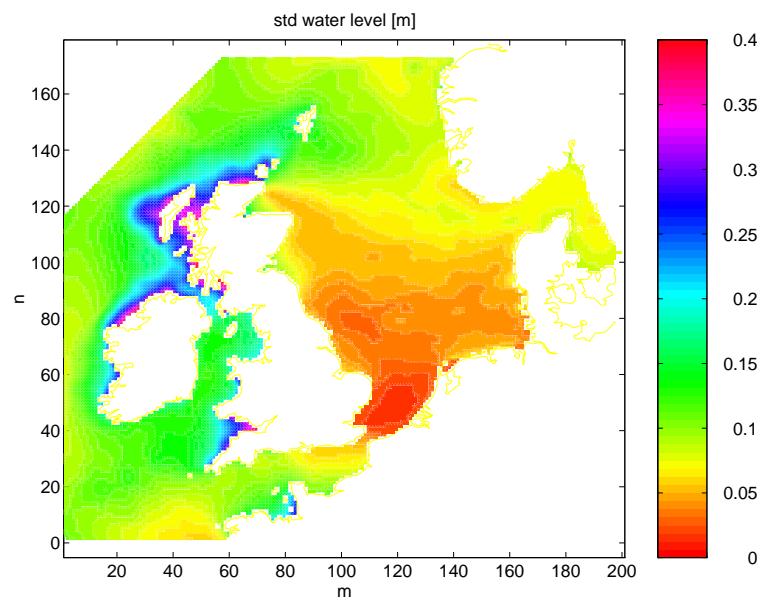


Figure 6.20: Standard deviation of filter using 8 measurement locations

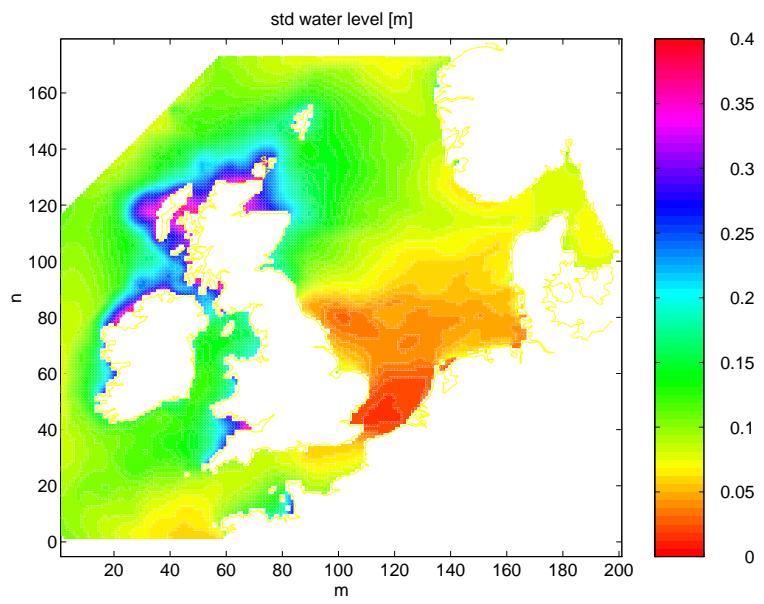


Figure 6.21: Standard deviation of filter using 7 measurement locations : no Wick

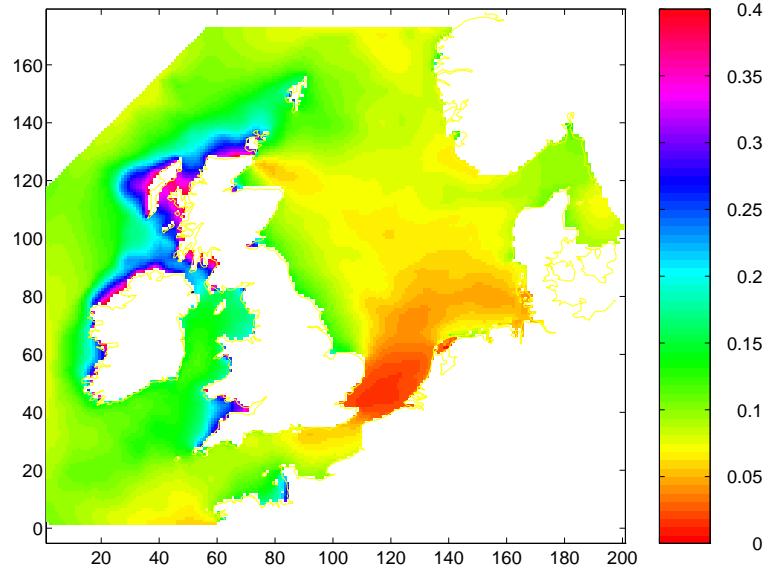


Figure 6.22: Standard deviation of filter using 8 measurement locations : no North Shields

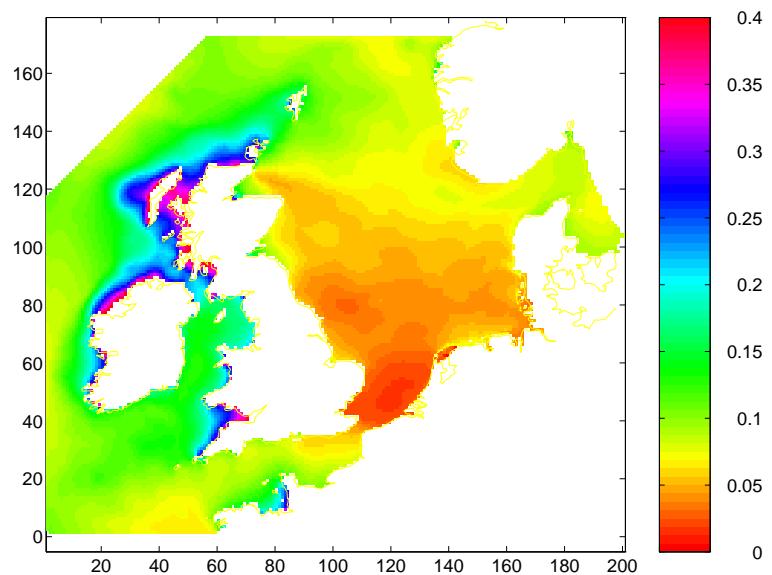


Figure 6.23: Standard deviation of filter using 8 measurement locations : no Lowestoft

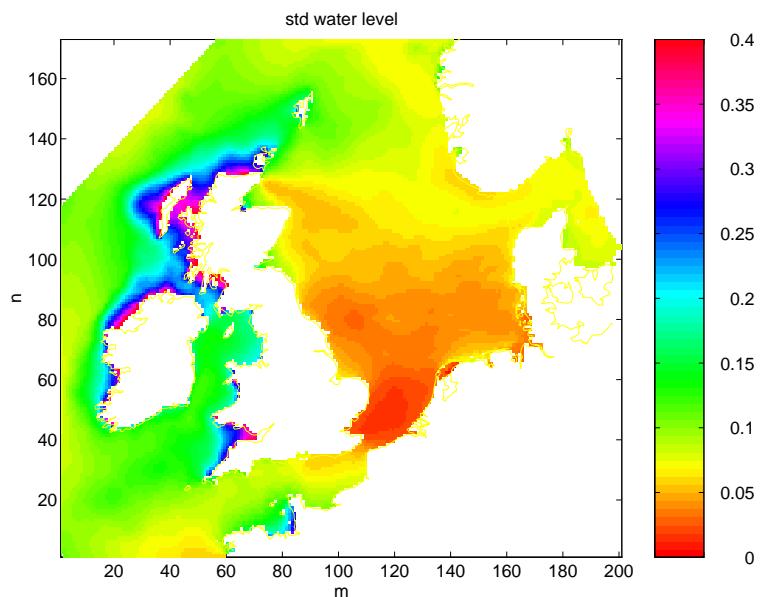


Figure 6.24: Standard deviation of filter using 8 measurement locations : no Vlissingen

locations because if the model is sufficiently linear the actual value of the measurement has little effect on the standarddeviation. It is therefore possible to use generated measurements, for example assuming the innovations zero at the imaginary location. In a similar manner it is possible to study the impact of removing an existing measurement location from the assimilation process.

In the approach proposed above it would be necessary to carry out a new Kalman filter computation for every different configuration of the measurement locations. In this way only a limited number of additional measurement locations can be studied, while ideally one wants to give a huge number of allowed locations and study the impact for all these locations. However it is often not necessary to have very detailed information for all possible locations, but as a first indication the decrease of the RMS for the waterlevel over the whole model domain would be sufficient.

The average RMS error estimated by the Kalman filter is given by

$$RMS_I = \sqrt{\sum_{i \in I} [\mathbf{P}]_{i,i}/n_I} \quad (6.19)$$

where the summation is over the indices of the state vector associated with waterlevels.  $n_I$  denotes the number of points in the summation. It is also possible to use the whole state vector, but in that case variables with different units are mixed and thus the RMS can longer be interpreted as the average uncertainty in meters. In the case of a shallow water model the variables can all be scaled such that their squared values are all energies. Finally, it is also possible to use only part of the model domain in the evaluation.

From equations 3.5 and 3.6 that the improvement of the RMS of adding one measurement is

$$\Delta(RMS_I^2) = \sum_{i \in I} [\mathbf{P} \mathbf{C}' (\mathbf{C} \mathbf{P} \mathbf{C}' + \Sigma_o)^{-1} \mathbf{C} \mathbf{P}]_{i,i}/n_I \quad (6.20)$$

where  $\mathbf{C}$  is the measurement vector for the (imaginary) measurement. In the RRSQRT filter the error covariance matrix  $\mathbf{P}$  is approximated by  $\mathbf{L}\mathbf{L}'$ , which makes it possible to compute the improvement more efficiently.

$$\Delta(RMS_I^2) \approx \sum_{i \in I} [\mathbf{H}' \mathbf{Q} \mathbf{H} (\mathbf{H}' \mathbf{H} + \Sigma_o)^{-1}]_{i,i}/n_I \quad (6.21)$$

$$\mathbf{H} = \mathbf{L}' \mathbf{C}' \quad (6.22)$$

$$[\mathbf{Q}]_{i,j} = \sum_{k \in I} \mathbf{L}(k,i) \mathbf{L}(k,j) \quad (6.23)$$

Based on this the improvement  $IMP_I$  is defined as

$$IMP_I := \sqrt{\Delta(RMS_I^2)} \quad (6.24)$$

Since the value of  $IMP_I$  can be computed efficiently it becomes feasible to try many possible locations. Figure 6.25 shows the values for the waterlevels at

all grid-points. The summation is over all waterlevel indices. The error covariance matrix was obtained from the Kalman filter experiment with all 8 stations assimilated. When compared with figure 6.20 the improvement seems roughly proportional to the standard deviation of the error. This is not necessarily so since the correlation is also important for the impact of a measurement. The values of the improvement indicate that the best choice for an additional measurement location would be in western Scotland or northwestern Ireland. Most likely, additional observations in this region would mainly reduce the standard deviation more or less locally, with no real improvement in the southern North Sea. Because the influence of this region on the short term forecasts is small, additional measurements would mainly be of interest for forecasts longer than 12 hours.

Increasing the number of measurement locations in the southern north sea is not expected to increase the accuracy of the storm surge forecasts. The decrease of the estimated standard deviation is very small here. Also the estimated accuracy in this region is already high. The largest impact for forecasts less than 12 hours ahead can be expected from additional observations along the northeastern coast of Scotland. There the impact is not negligible while the area is close enough to influence the forecasts. In addition a little redundancy for the location Wick is created.

In previous experiments it was shown that the system noise formulation does not match the true model error very well. In particular the influence of the wind direction is not very well represented. One should therefore be careful when interpreting the results. The changes to the standard deviation are probably not very accurate in magnitude, but the patterns are likely to be more or less correct. Also, the improvement  $IMP_I$  only shows the impact of single measurements while tide gauges provide observations at regular intervals. The combined impact of these measurements in most cases is larger than for one single measurement as in figure 6.25. Because information, i.e. a decrease of the error covariance due to assimilation of observations, is propagated using the model this information 'travels' in the same direction as the tidal wave. The combined effect of adding observations of a location at the east coast of Scotland will probably extend in the direction of the southern North Sea.

The estimated standard deviation is however not the only argument on which the measurement locations should be selected. One important other constraint is that it is much cheaper to measure the waterlevel along the coast than further from the coast, and except for some special locations, such as oil-rigs, positioning tide gauges at open sea is not feasible. It is also important to consider local effects around the measurement location that are not modelled. The coarse model used for storm surges is not able to capture for example the bathymetry at some locations; in the Wadden-sea the 8 km grid can not be used to follow the intricate pattern of gullies. Also locations that become dry during ebb should be avoided. The non-linearities involved in drying and flooding are quite severe and can cause instabilities in the extended Kalman filter.

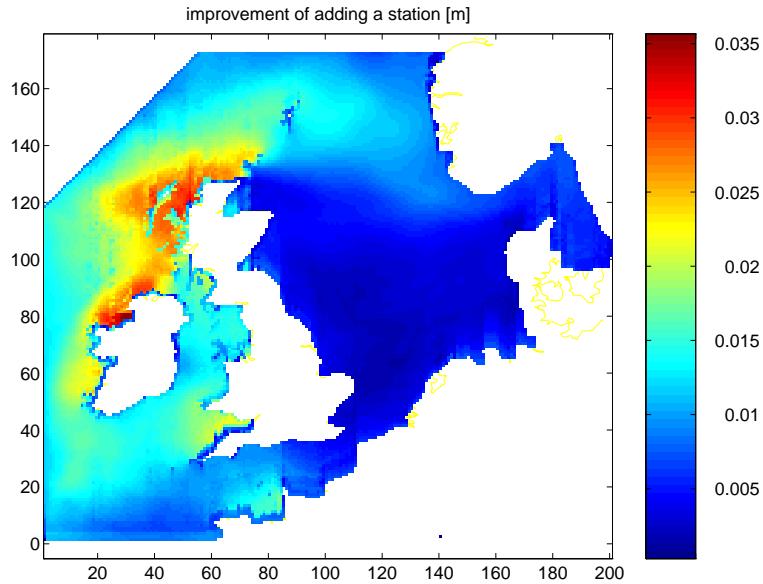


Figure 6.25: Improvement  $IMP_I$

## 6.8 Conclusions

In this chapter the reduced rank square root (RRSQRT) algorithm was applied to a two dimensional shallow water flow model with the aim of comparing the results with the existing steady state Kalman filter and if possible to improve the performance. The experiments show that the algorithm works well for this application. The required computation time is several orders of magnitude smaller than that for the 'full' Kalman filter.

A comparison with an existing steady state Kalman filter for the operational Dutch storm surge forecasting model showed that the performance of both filters is comparable. Although in theory the extended Kalman filter could perform better due to the nonlinearities of the model, it seems that these nonlinearities are only of little importance for this application.

In subsequent experiments the method of covariance matching was used to calibrate the parameters of the Kalman filter. Though the method worked well it was not possible to improve the performance by calibration. A comparison of observed and computed covariances of the residuals showed that the specification of the system noise is not very accurate. In particular the influence of the wind direction is not modeled. Adaptation of the system noise in this respect could improve the performance, but also excludes the use of a steady state Kalman filter.

In the present operational storm surge forecasting system measurements from 8 locations are assimilated. The standard deviation that can be computed from the square root of the error covariance in the RRSQRT filter was used to as-

sess the impact of adding or removing a location. The sensitivity of the Kalman filter estimate with respect to the various locations turns out to depend highly on the position of the measurements. For example, the sensitivity with respect to Wick is very large to measurements in the southern North Sea. Because a Kalman filter simulation is needed for every configuration of the measurements it is only possible to investigate a very limited number of adaptations to the present configuration. Therefore a method to estimate the change in RMS of the standard deviation of the error at a large number of (imaginary) measurement locations was proposed. For the CSM mode the proposed method shows that the largest impact of an additional measurement location can be expected in north or northwest Scotland or Northern Ireland. Most likely additional measurements in this region will mainly affect the long term forecasts. Further research is needed to find the optimal configuration of the measurements.

## Chapter 7

# Application of a Kalman smoother to two dimensional shallow water flow: Accurate water level reconstruction

### 7.1 Introduction

The accuracy of water depths measured with echo-sounding is mainly determined by the correction for the tide. Therefore accurate now-casting or reconstruction of tidal elevation is very important in this procedure.

At the moment for computing the water level. The first method is by model simulation. During very calm weather accurate water levels can be obtained by simulation of the tide with a hydrodynamic model. For this purpose a two dimensional shallow water model is used (see also eqns 7.1, 7.2 and 7.3).

$$\frac{\partial \xi}{\partial t} + \frac{\partial Hu}{\partial x} + \frac{\partial Hv}{\partial y} = 0 \quad (7.1)$$

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + g \frac{\partial \xi}{\partial x} - fv + \frac{gu\sqrt{u^2 + v^2}}{C^2 H} \\ - C_d \frac{\rho_a}{\rho_w} \frac{V^2 \cos \psi}{H} + \frac{1}{\rho_w} \frac{\partial p_a}{\partial x} = 0 \end{aligned} \quad (7.2)$$

$$\begin{aligned} \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + g \frac{\partial \xi}{\partial y} + fu + \frac{gv\sqrt{u^2 + v^2}}{C^2 H} \\ - C_d \frac{\rho_a}{\rho_w} \frac{V^2 \sin \psi}{H} + \frac{1}{\rho_w} \frac{\partial p_a}{\partial y} = 0 \end{aligned} \quad (7.3)$$

where:

$x, y$	= cartesian coordinates in horizontal plane
$u$	= depth-averaged current in $x$ direction
$v$	= depth-averaged current in $y$ direction
$\xi$	= water level above the reference plane
$D$	= water depth below the reference plane
$H = \xi + D$	= total water depth
$g$	= gravity acceleration
$f$	= coefficient for the Coriolis force
$C$	= Chezy coefficient
$V$	= wind velocity
$\psi$	= wind angle with respect to the positive $x$ -axis
$C_d$	= wind friction coefficient
$p_a$	= air pressure at the surface
$\rho_w$	= density of sea water
$\rho_a$	= density air at the surface

The coastal model uses curvilinear coordinates even though the equations shown here make use of Cartesian coordinates.

Ideally, the area covered by the model should not be much larger than the area that is to be measured. In this way the effort in the model is spent where it is needed, in a strip of approximately 30 to 40 km along the coast, in estuaries and in the Waddenze. Figure 7.1 shows a model grid for the area of interest. For smaller measuring campaigns it is possible to use only part of this model in order to reduce the computation time.

Except for very calm weather the water level is also influenced by wind and differences in air pressure. For prediction (reconstruction) of these effects a much larger model than the one shown in figure 7.1 is needed since the spatial scale of the meteorological processes involved is much larger. The spatial scale of the astronomic tide is also much larger, but the harmonic constituents at the open boundary are determined only once, by nesting in a larger model and subsequent calibration. From these constituents the astronomic tidal elevation at the open boundary can be computed for every time interval needed. The smaller model can thus be used directly for computing the astronomic tide, but for modeling of the meteorological effects the larger model is needed every time.

A second option for the reconstruction of water levels near the coast is by interpolation of tide gauge measurements. Because the water level does not vary quickly in space interpolation is possible over several kilometers, but especially further from the coast the coverage of the tide gauges is not dense enough to allow direct interpolation. Therefore, additional temporary tide gauges are used during echo sounding of the bathymetry.

Using data assimilation it is possible to combine the two sources of information, a shallow water model and tide gauge measurements. Since the model is used to interpolate the data in a manner that is consistent with the physics it may be possible to reduce the number of temporary tide gauges, or omit them completely. The approach followed here is to use the RRSQRT Kalman filter

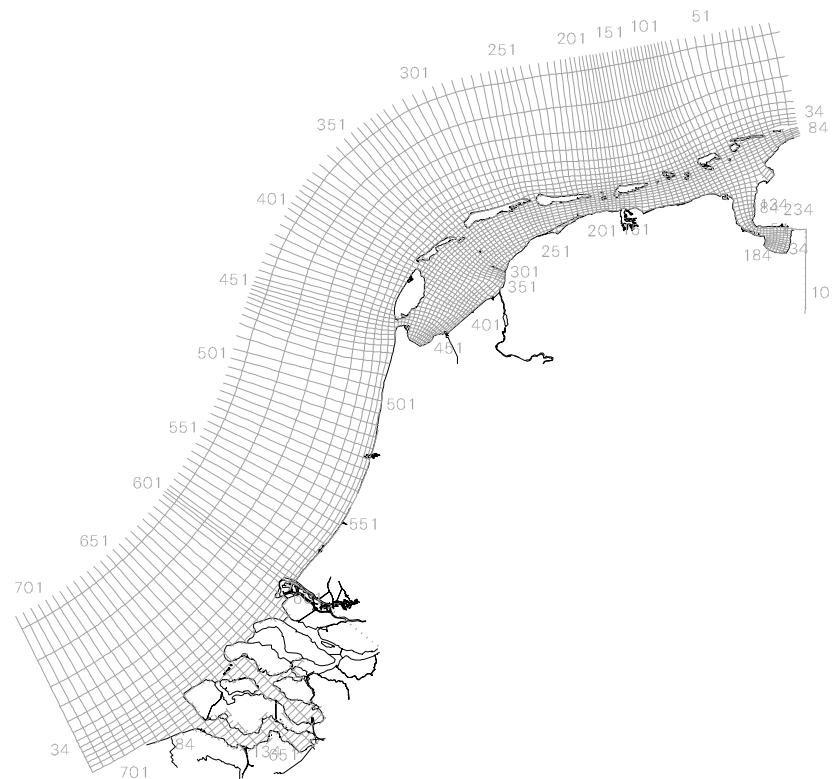


Figure 7.1: Model grid of the KUSTSTROOK model

to assimilate the measurements. Astronomic tide is used as a first guess at the open boundary of the hydrodynamic model and it is assumed that the uncertainty of the model forecasts is dominated by neglecting the set-up at the open boundary.

Although filtering is preferable, so that the correction of the depth can be carried out immediately, it is also possible to use smoothing with 'future' measurements for additional accuracy. Experiments with both filtering and smoothing will be used to study the differences in accuracy for both procedures.

So far, the accurate reconstruction of water levels has been motivated from the reduction of echo soundings, but the methods described here are also very useful for other purposes. If very accurate reconstruction of water levels is possible these water levels can serve as pseudo measurements e.g. to fill data gaps or reduce the number of tide gauges in a measurement network. It is also useful for studying water levels and currents in situations with significant meteorological forcing. As will be shown in the sequel the smoothing algorithm proceeds by first reconstructing the open boundary and subsequent use of the model to reconstruct the water level over the whole model domain. However, the boundary computed in this way can also be used to compute water levels and currents during a storm when the geometry is changed, e.g. with an additional island for the extension of Schiphol airport.

## 7.2 Smoothing and boundary extraction

### 7.2.1 Introduction

The Kalman filter computes estimates in which measurements until that time are assimilated. With smoothing algorithms it is possible to include 'future' measurements past this time for increased accuracy.

It can easily be shown that waiting and including later measurements can improve the estimate. It is well known that the solution of the shallow water equation can be decomposed into several components that travel along the characteristics. [74] Figure 7.2 shows the surge of the examples again. In this example the characteristics move left and right with velocity  $c = \sqrt{gH} = \sqrt{9.8 \cdot 10} \approx 9.9[m/s]$ . Shortly after the surge enters the model only the component moving to the right is present, the Riemann invariant moving left is absent. The wave travels in  $T = x_m/c = 24 \cdot 10^3 / 9.9[s] = 40.4[min]$  from the boundary to the measurement location. Because the damping is small, the wave is almost the same when it passes the measurement location. Thus, if one waits for 40.4[min] a good reconstruction of the boundary can be given by just shifting the observed water level over 40.4[min] backwards in time.

However, if the estimate is computed using only measurements until that time then the water level left of the measurement location is mainly the result of a forecast of the boundary starting from 40.4[min] before the present time using the time correlation of errors in the boundary. The accuracy of this forecast depends on how well the statistical model of the system noise at the

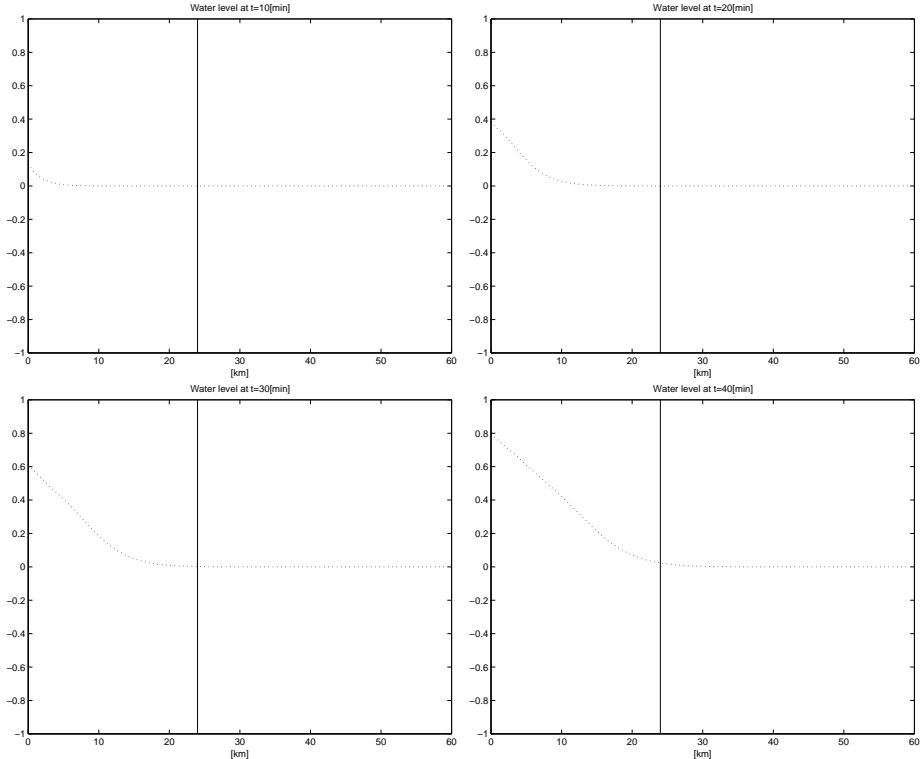


Figure 7.2: Example of a surge in a 1-D model

open boundary describes the statistics of the surge, but this forecast is likely to be less accurate than smoothing. A mathematical argument is that one can see from equation 3.6 that the measurements do not increase the uncertainty, since the second term is positive semi definite, and that if  $CP(k+1|k)C' \neq 0$ , i.e. the additional measurements give information about something that is not already certain then the error in the estimate strictly decreases. In this argument an augmented state approach is assumed.

### 7.2.2 A fixed-lag smoother for the RRSQRT filter

Although there exist many smoothing algorithms, straight forward application of these algorithms is not feasible for the same reasons as the 'full' Kalman filter is not computationally feasible. In general smoothing algorithms require even more computation than the Kalman filter. In chapter 3 it was shown that a fixed lag smoother can be constructed from the Kalman filter by augmenting the state vector. Since an efficient approximation of the Kalman filter is available through the RRSQRT algorithm, an efficient smoother can be constructed by augmenting

the state vector and applying the RRSQRT algorithm to the augmented model. The approach followed here is similar to the one described in Brummelhuis [69, 2] and Zhang [78, 79].

Of the three types of smoothers, fixed-lag, fixed interval and fixed-point, the fixed-lag smoother is most appropriate for the reconstruction of water levels in our application. The fixed-lag smoother can be derived from the Kalman filter by augmenting the state vector in the following way

$$\mathbf{x}_a(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}(k-1) \\ \vdots \\ \mathbf{x}(k-l) \end{bmatrix} \quad (7.4)$$

Using this definition  $\hat{\mathbf{x}}_a(k|k)$  becomes

$$\hat{\mathbf{x}}_a(k|k) = \begin{bmatrix} \hat{\mathbf{x}}(k|k) \\ \hat{\mathbf{x}}(k-1|k) \\ \vdots \\ \hat{\mathbf{x}}(k-l|k) \end{bmatrix} \quad (7.5)$$

The augmented model is given by

$$\mathbf{x}_a(k+1) = \begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{x}(k) \\ \vdots \\ \mathbf{x}(k-l+1) \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)) \\ \mathbf{x}(k) \\ \vdots \\ \mathbf{x}(k-l+1) \end{bmatrix} \quad (7.6)$$

This is just one model evaluation and a shift of the states. The RRSQRT algorithm can in principle be applied directly to this augmented model, but this would result in an increase of  $l$  times of the required memory compared to the RRSQRT filter.

For the present application the computational burden and storage requirements can be reduced further by exploiting the fact that the model output is governed completely by the past forcing (here at the open boundary only) and the initial state, i.e.

$$\hat{\mathbf{x}}_{k|\tilde{\mathbf{w}}(k)} = E[\mathbf{x}(k)|\mathbf{w}(k), \dots, \mathbf{w}(0), \mathbf{x}(0)] \quad (7.7)$$

$$= \mathbf{f}(\mathbf{f}(\dots \mathbf{f}(\mathbf{x}(0), \mathbf{u}(0)) \dots), \mathbf{u}(k-1), \mathbf{w}(k-1)), \mathbf{u}(k), \mathbf{w}(k)) \quad (7.8)$$

and

$$E[(\mathbf{x}(k) - \hat{\mathbf{x}}_{k|\tilde{\mathbf{w}}(k)}) (\mathbf{x}(k) - \hat{\mathbf{x}}_{k|\tilde{\mathbf{w}}(k)})'] = 0 \quad (7.9)$$

where the repeated use of  $\mathbf{f}$  just denotes using the model with the given forcing.

Thus if the state is augmented as

$$\mathbf{x}_a(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{w}(k) \\ \mathbf{w}(k-1) \\ \vdots \\ \mathbf{w}(0) \\ \mathbf{x}(0) \end{bmatrix} \quad (7.10)$$

then the smoother estimate can be computed from

$$\hat{\mathbf{x}}(k|l) = E[\mathbf{x}(k)|\mathbf{y}(0), \dots, \mathbf{y}(l)] \quad (7.11)$$

$$= E[\mathbf{x}(k)|\tilde{\mathbf{y}}(l)] \quad (7.12)$$

$$= E_{\tilde{\mathbf{w}}(l)} [E[\mathbf{x}(k)|\tilde{\mathbf{y}}(l)], E[\mathbf{w}(0)|\tilde{\mathbf{y}}(l)], \dots, E[\mathbf{w}(k)|\tilde{\mathbf{y}}(l)]] \quad (7.13)$$

$$= E[\mathbf{x}(k)|E[\mathbf{w}(0)|\tilde{\mathbf{y}}(l)], \dots, E[\mathbf{w}(k)|\tilde{\mathbf{y}}(l)]] \quad (7.14)$$

where  $E_{\tilde{\mathbf{w}}(l)}$  denotes the expectation taken over the random variables  $E[\mathbf{w}(i)|\tilde{\mathbf{y}}(l)]$   $i = 0, \dots, k$ . The right-hand side in equation 7.14 can be evaluated using equation 7.8. All the ingredients for this can be found in  $\hat{\mathbf{x}}_a(k|l)$ . Thus when the surge at the open boundary is added to the state vector, it is possible to first reconstruct the state and then use the model to find the estimate for the other variables. A practical difficulty is that the size of the augmented state defined as in equation 7.10 grows in time. It can however be shown that after some time the first few  $\mathbf{w}$ 's are not changed anymore, i.e.  $E[\mathbf{w}(k)|\tilde{\mathbf{y}}(k+j)] \approx E[\mathbf{w}(k)|\tilde{\mathbf{y}}(k+i)]$  for some  $i > 0$  and all  $j > i$ . This can be used to construct an efficient approximate algorithm.

The following efficient approximation to the fixed lag Kalman filter is proposed:

Define an augmented state vector as

$$\mathbf{x}_a(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{w}(k) \\ \mathbf{w}(k-1) \\ \vdots \\ \mathbf{w}(k-i) \end{bmatrix} \quad (7.15)$$

Use the RRSQRT filter with the augmented model. At every time-step the estimate  $E[\mathbf{w}(k-i)|\tilde{\mathbf{y}}(k)]$ , which is the last part of  $\hat{\mathbf{x}}_a(k|k)$ , is written to disk. A 'normal' model simulation is then used to obtain the smoother estimates. The procedure described above only strictly applies for linear models. For nonlinear models the extended Kalman filter version of the RRSQRT algorithm can be used to obtain approximate results.

## 7.3 Experiments and results

### 7.3.1 Experiment with a storm surge

In the first experiment the RRSQRT filter (without smoother) was applied to a part of the model shown in figure 7.1. The smaller model is shown in figure 7.3. As an extreme test case the storm period of February 21 1993 was selected. This has no practical value, other than to test the method, because it is impossible to carry out echo sounding measurements during such a storm. At the open boundary astronomic tides were used as a first guess.

The uncertainty at the open boundary was modeled as an AR(1) process as was described in section 2.4. No noise in the wind forcing was included because errors in the open boundary will dominate. The following parameter values were used

parameter	symbol	value
standard deviation of water level at the boundary	$\sigma_b$	0.3 [m]
time correlation of noise at open boundary	$\alpha_b$	0.9
standard deviation of measurement errors	$\sigma_m$	0.2

The time correlation is per time-step of 2[min]. For this experiment the number of modes was set to  $q = 50$ . The measurements at Hoek van Holland, Petten Zuid, Cadzand and Euro Platform were assimilated into the model. The other measurement stations are used for validation.

Figures 7.4 – 7.6 show time series of water levels computed with the Kalman filter compared to the ones computed with a direct model simulation with the astronomic boundary condition. As can be seen from these graphs without assimilation of the measurements the storm surge is completely absent. With assimilation, the measurements are followed closely. Not only at the locations used in the filter, but also in the locations used for validation. The Kalman filter is thus able to interpolate quite accurately along the coast. Only 4 measurement stations are needed in this case to compute a reconstruction of the water levels along the part of the coast covered by the model.

At some locations, such as Scheveningen, the time-series computed by the Kalman filter shows some fluctuations. This is caused partly by the approximations in the RRSQRT algorithm and partly this is a natural phenomenon, since every 10 minutes new observations are assimilated and the water level is adapted. Additional experiments have shown that when the number of modes is increased the fluctuations become smaller.

What happens near the boundary can be seen in figures 7.7, 7.8 and 7.9. The water levels are corrected in a very large part of the model, but near the boundary the adaptations are somewhat smaller. Since there are no measurements available near the western open boundary there is no means of validating the Kalman filter estimate near the open boundary. However, it is not likely that the surge is very different near this boundary during a storm. It can be concluded that although the RRSQRT filter gives very good results near the

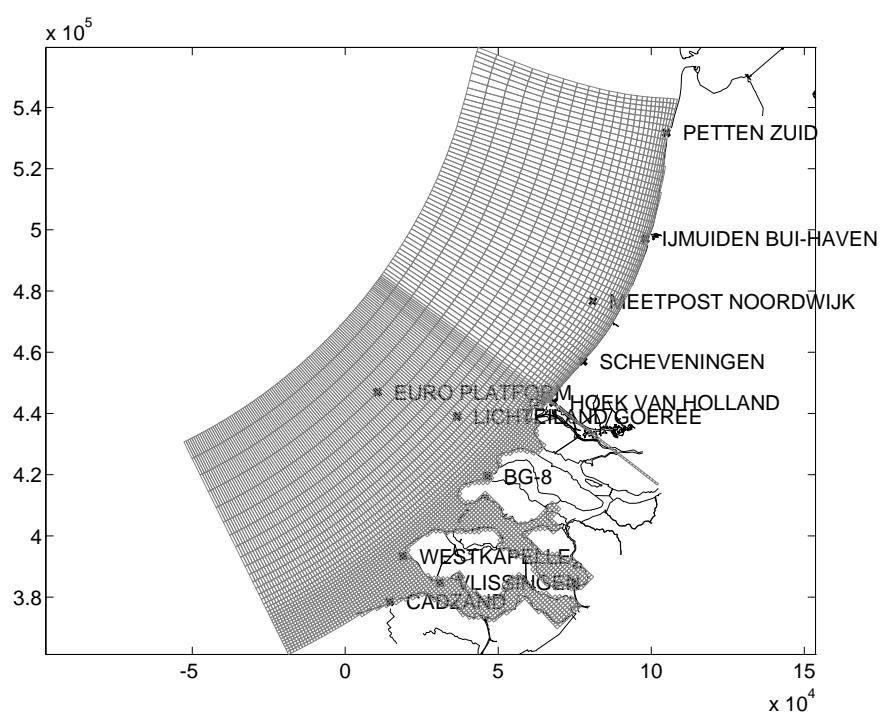


Figure 7.3: Grid and bathymetry of smaller model

shore, the water levels are probably less accurate near the western boundary. The charts also show some spatial fluctuations. These are mainly caused by the small temporal correlation in this experiment which necessitates a larger number of modes.

### 7.3.2 A comparison of filtering and smoothing

In the following experiments the performance of the smoother proposed in section 7.2.2 will be examined and compared with the RRSQRT filter. The results of the smoother will also be compared with validation measurement to see if the objective, high accuracy reconstruction of water levels, can be achieved.

Figure 7.3 again shows the bathymetry and grid of the section of the model used for the experiments. The measurement locations are also indicated. Some of the measurements, the stations 'Noordwijk' and 'Westkapelle' are used for validation, i.e. these are not assimilated. All the measurements are synthetic. They were obtained from the harmonic constants for these measurement locations. The system noise is assumed to be dominated by the errors caused by prescribing an approximate astronomic tide at the open boundary. For this purpose an AR(1) model is employed with a standard deviation of 0.30[m] of the error in the water level at the open boundary. The time correlation is 0.98 per time-step of 2[min]. The standard deviation of measurement errors is assumed to be 0.20[m] for 'Cadzand' and 'BG-8' and 0.10[m] for Petten, Scheveningen and IJmuiden.

With these values for the parameters a model run and a Kalman filter run were performed. Figure 7.10 shows a comparison of water levels at the measurement locations. It can be seen that the filter significantly reduces the differences between measurements and the estimate both for the assimilated measurements and for the validation measurements. This is consistent with the results of the previous experiments.

The fluctuations are almost absent for the Kalman filter in this experiment. The first cause for the smaller fluctuations is that the time correlation for the noise at the open boundary were chosen differently for this experiment, a correlation time scale of 99[min] instead of 47[min]. This setting makes the RRSQRT filter more stable, so that the fluctuations will be smaller for the same number of modes.

In previous experiments it was observed that the corrections at the open boundary were smaller than the corrections in the rest of the model. It was assumed that the smaller corrections meant that they were probably less accurate. One way of testing this hypothesis is to extract the estimate computed by the Kalman filter for the open boundary and to use this corrected boundary as input for the model. If the estimate is accurate than the astronomic boundary used in the model then the results of this experiment will be more accurate than those of the original model run.

From the Kalman filter experiment above the estimates of the water level at the open boundary were extracted. Time-series of some points along this boundaries are shown in figure 7.11. The estimates are well behaved and not

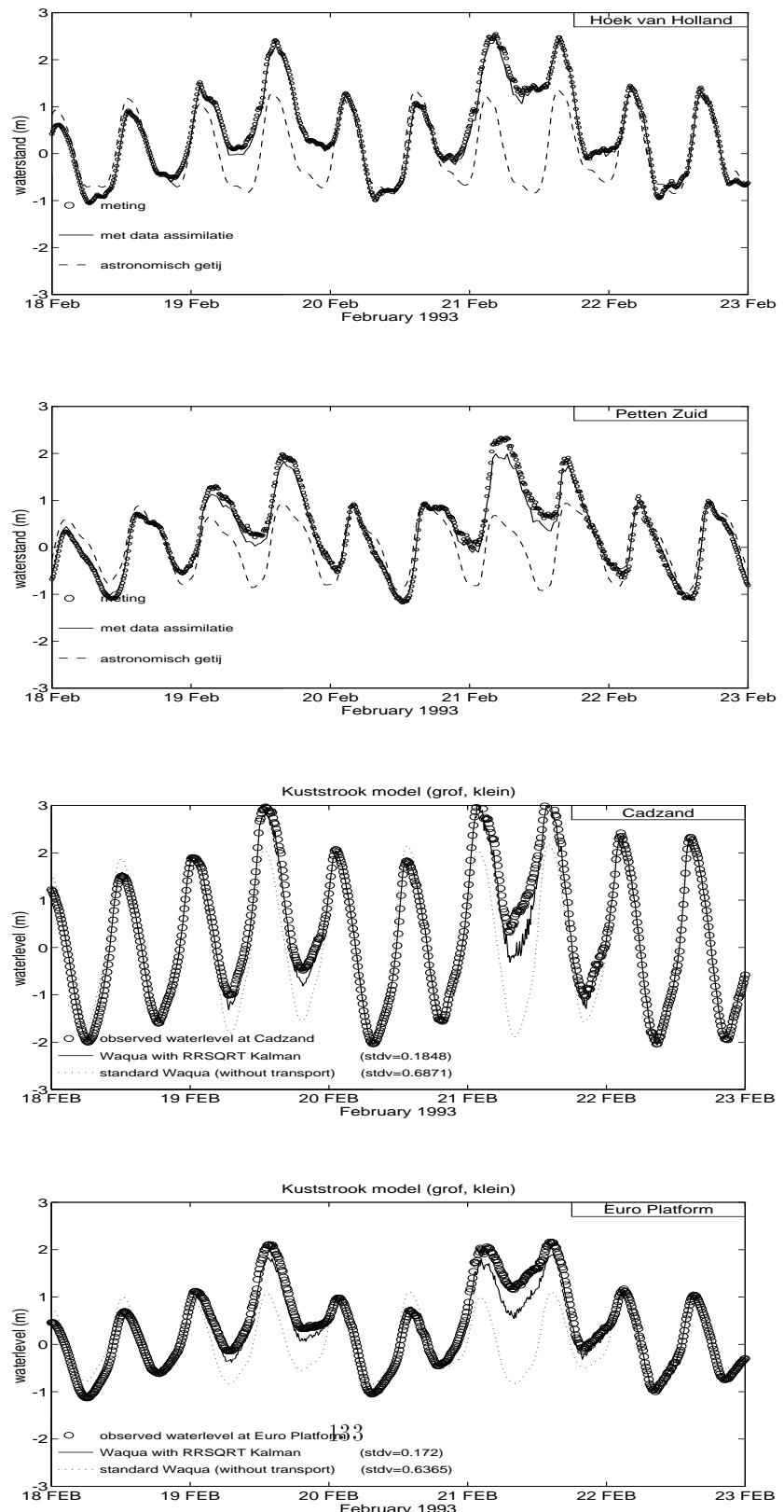


Figure 7.4: Water level series computed by model, filter and measured

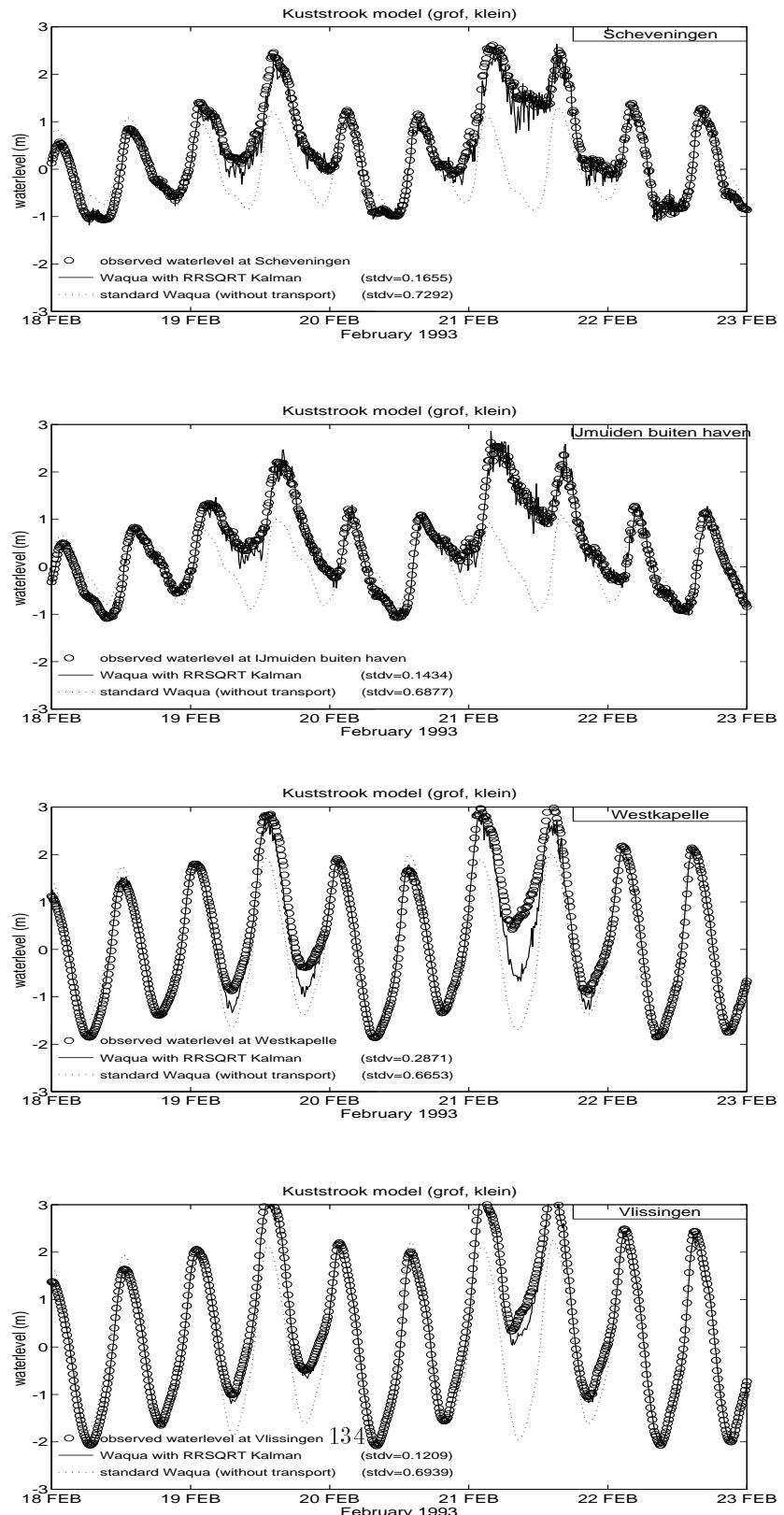


Figure 7.5: Water level series computed by model, filter and measured

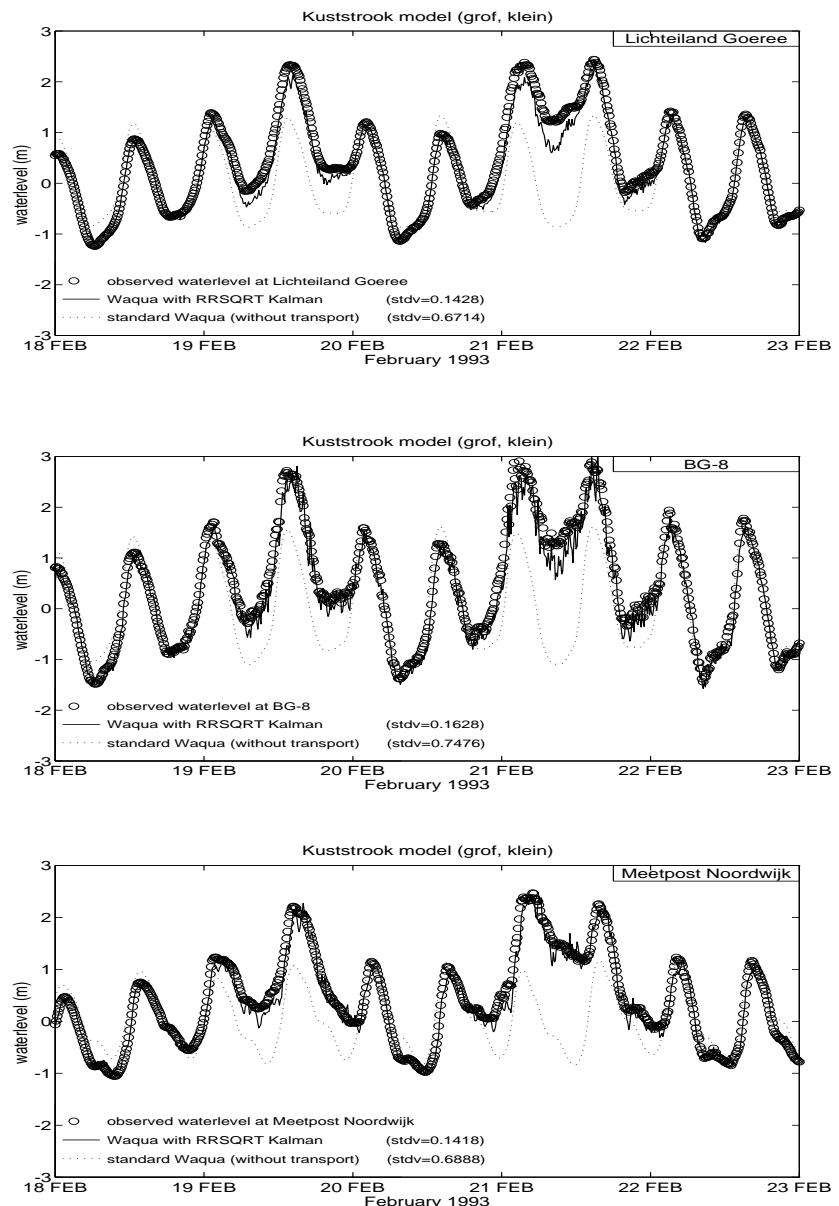


Figure 7.6: Water level series computed by model, filter and measured

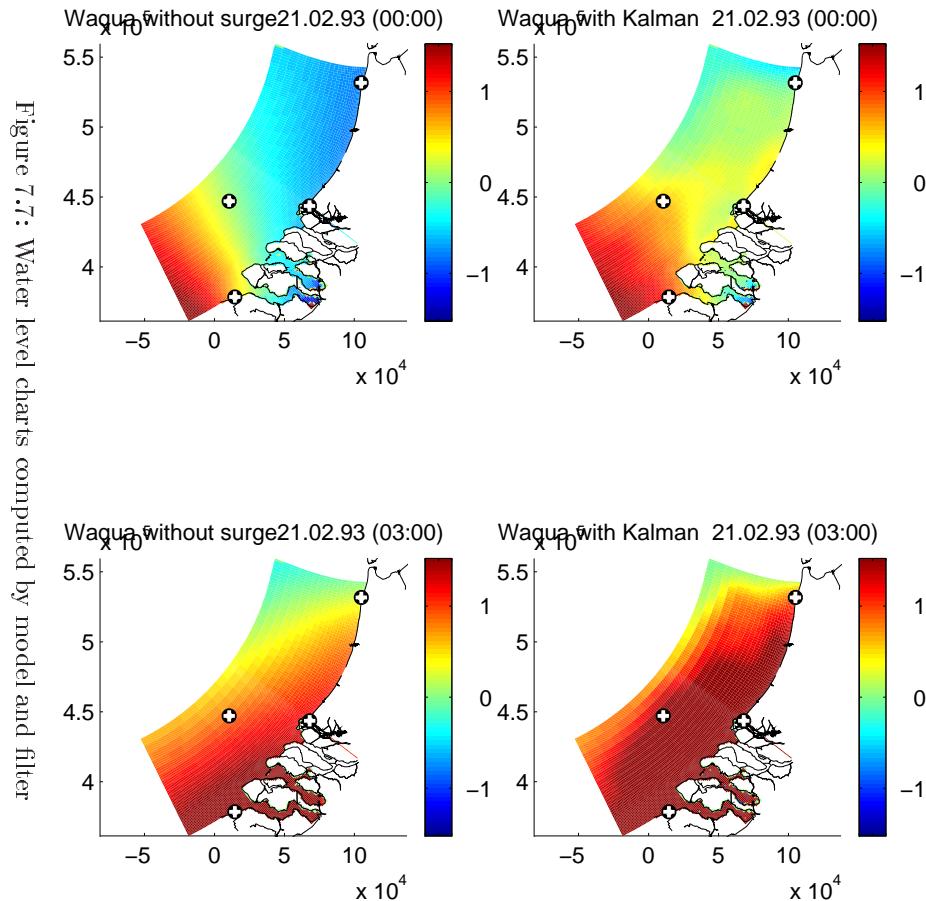


Figure 7.7: Water level charts computed by model and filter

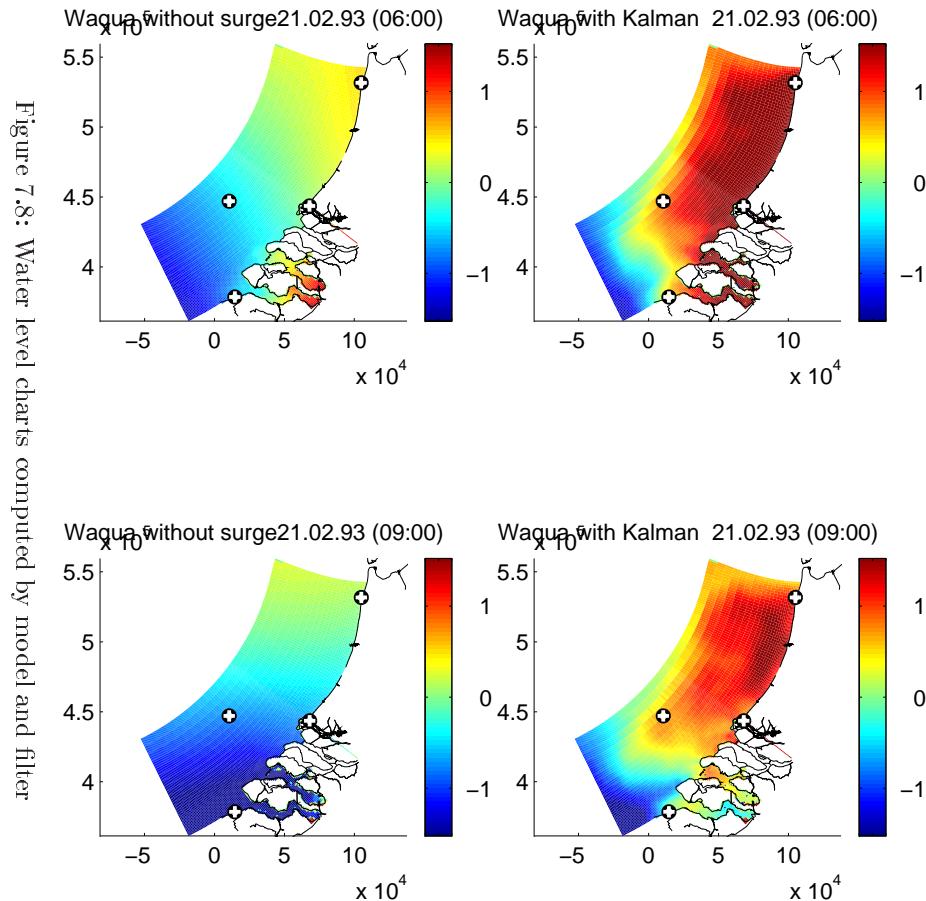


Figure 7.8: Water level charts computed by model and filter

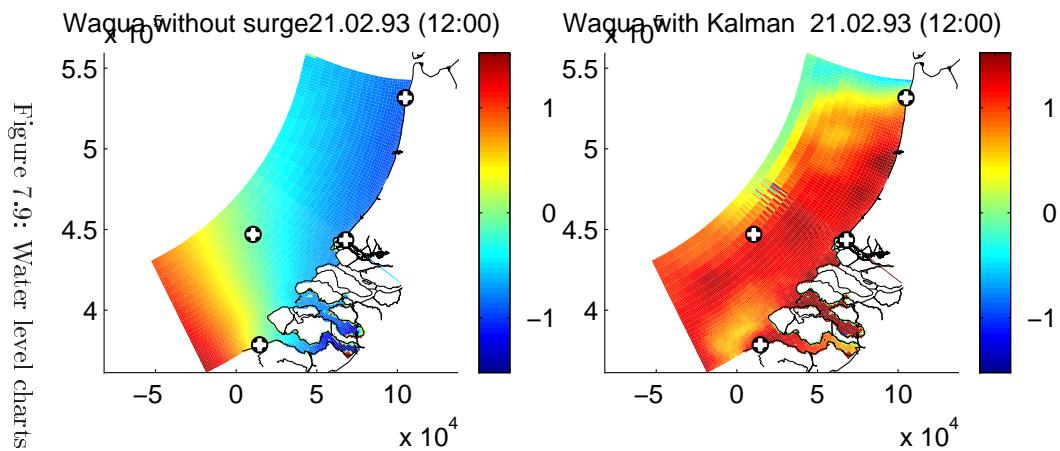


Figure 7.9: Water level charts computed by model and filter

too large. These water levels were subsequently used as forcing for a model simulation (no data assimilation was used in this part). Some water level time-series computed by the model with this 'new' and the 'old' boundary are shown in figure 7.12. As was assumed the 'new' boundary does not result in estimates with an accuracy comparable with the ones computed directly with the Kalman filter. In fact, the water levels found with the 'new' boundary are not much better than computed with the original 'old' boundary. This shows that the accuracy of the estimates of the Kalman filter is not very high at the open boundary.

In order to increase the accuracy further from the shore a Kalman smoother was applied. The RRSQRT based algorithm of section 7.2.2 was used for this purpose. The number of modes was set to  $q = 100$  instead of  $q = 50$  for the filter because a larger number of modes is used for convergence of the RRSQRT algorithm in case a smoother is used. Figure 7.13 shows some time-series of water levels for model, smoother and measurements. The water levels computed by the smoother are much improved over the direct model results. The accuracy both at the measurement locations and validation locations is comparable to that for the filter. The corrections near the open boundary however are very different from the corrections of the filter (Figure 7.14). Replacing the open boundary with the smoother estimate would contrary to the filter case not result in deteriorated estimates because the estimates are already computed with an adapted boundary.

The estimates of the Kalman filter and the Kalman smoother can also be compared directly at the open boundary. It can be seen (fig. 7.14) that the corrections at the open boundary are larger for the smoother. The difference is largest at the western boundary, with a maximum of 20[cm] in the northwest corner. This indicates that the additional measurements included in the smoother have a significant effect further away from the measurements. The distribution of the corrections between the measurement locations and the boundary can be seen in figure 7.15. These charts again show that although the differences near the coast are small the corrections are different for the Kalman filter and smoother further from the coast.

The experiments clearly show that both the Kalman filter and the smoother provide a more accurate reconstruction of the water level along the coast than a direct model simulation. The following table quantifies the accuracy of the smoother.

station	type	standard deviation	maximum error
Cadzand	assimilated	25[mm]	70[mm]
Petten Zuid	assimilated	21[mm]	70[mm]
BG-8	assimilated	28[mm]	109[mm]
Scheveningen	assimilated	19[mm]	63[mm]
IJmuiden	assimilated	20[mm]	100[mm]
MP Noordwijk	validation	29[mm]	70[mm]
Westkapelle	validation	23[mm]	71[mm]

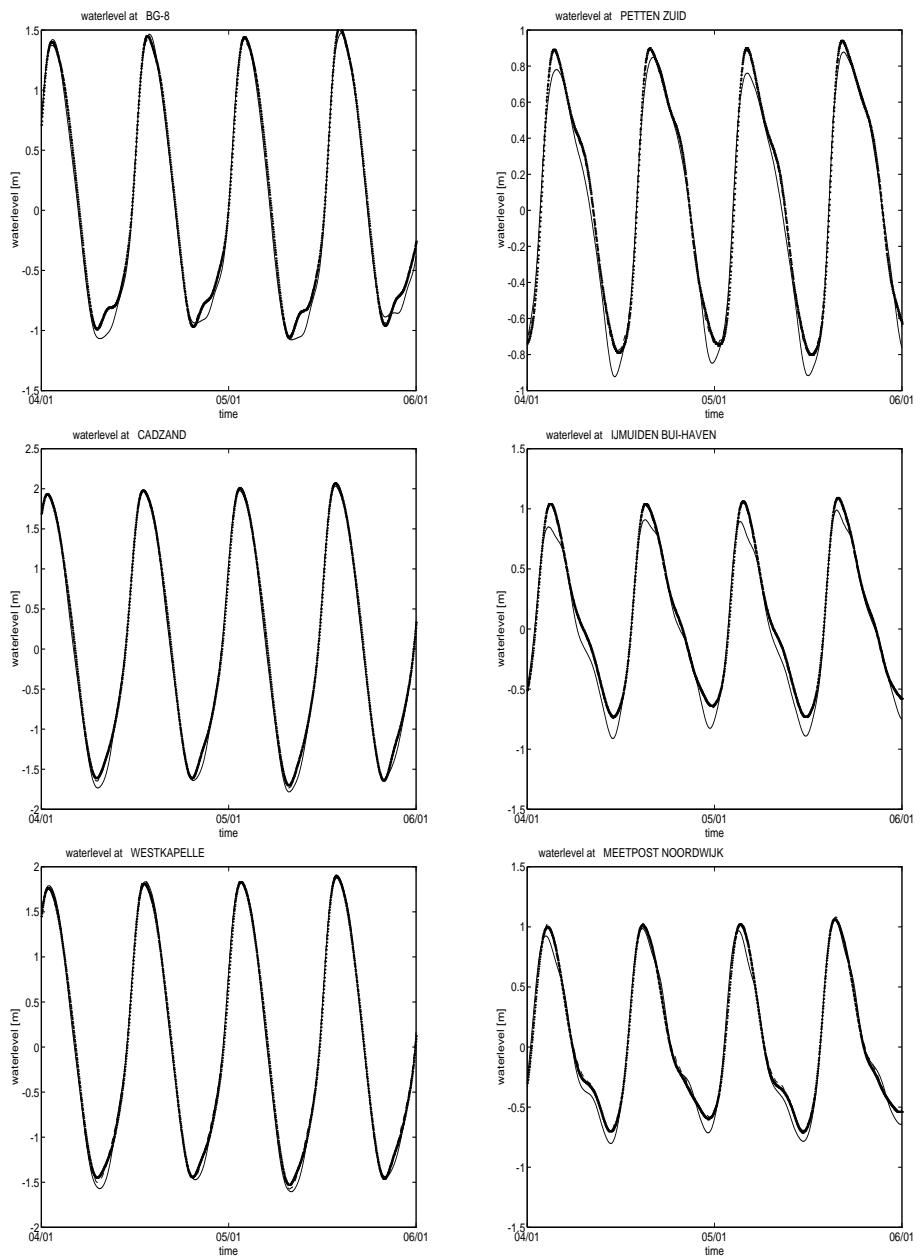


Figure 7.10: Water level series for filter (dashed), model (line) and measurements (dots)

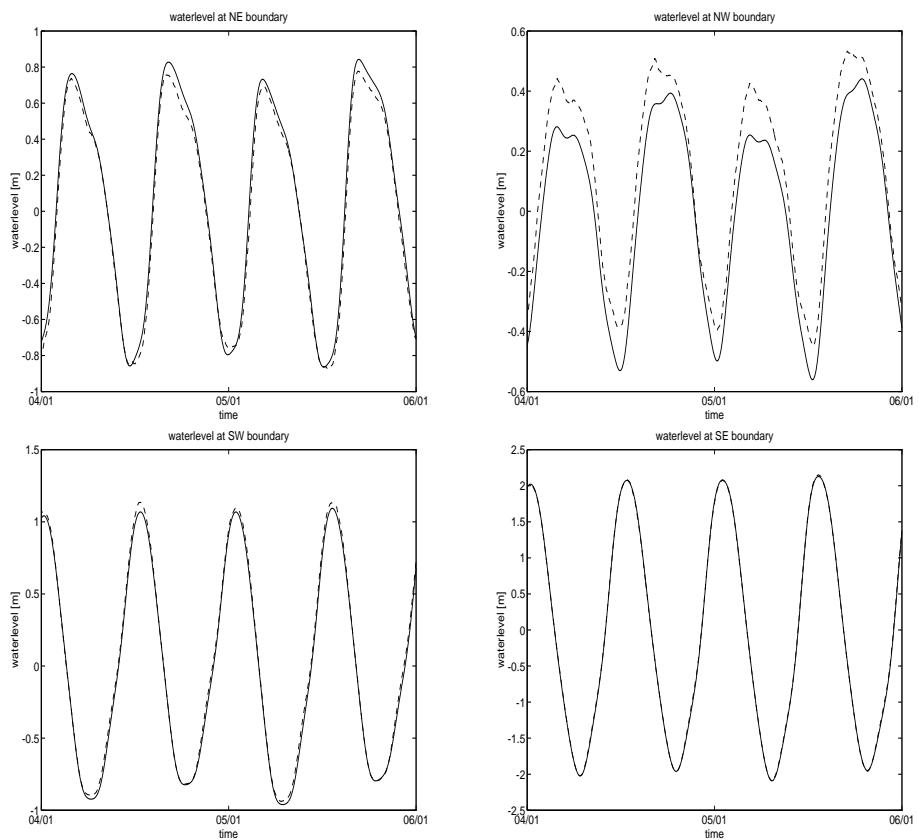


Figure 7.11: Water level series for model with new (dashed) and old (line) boundary

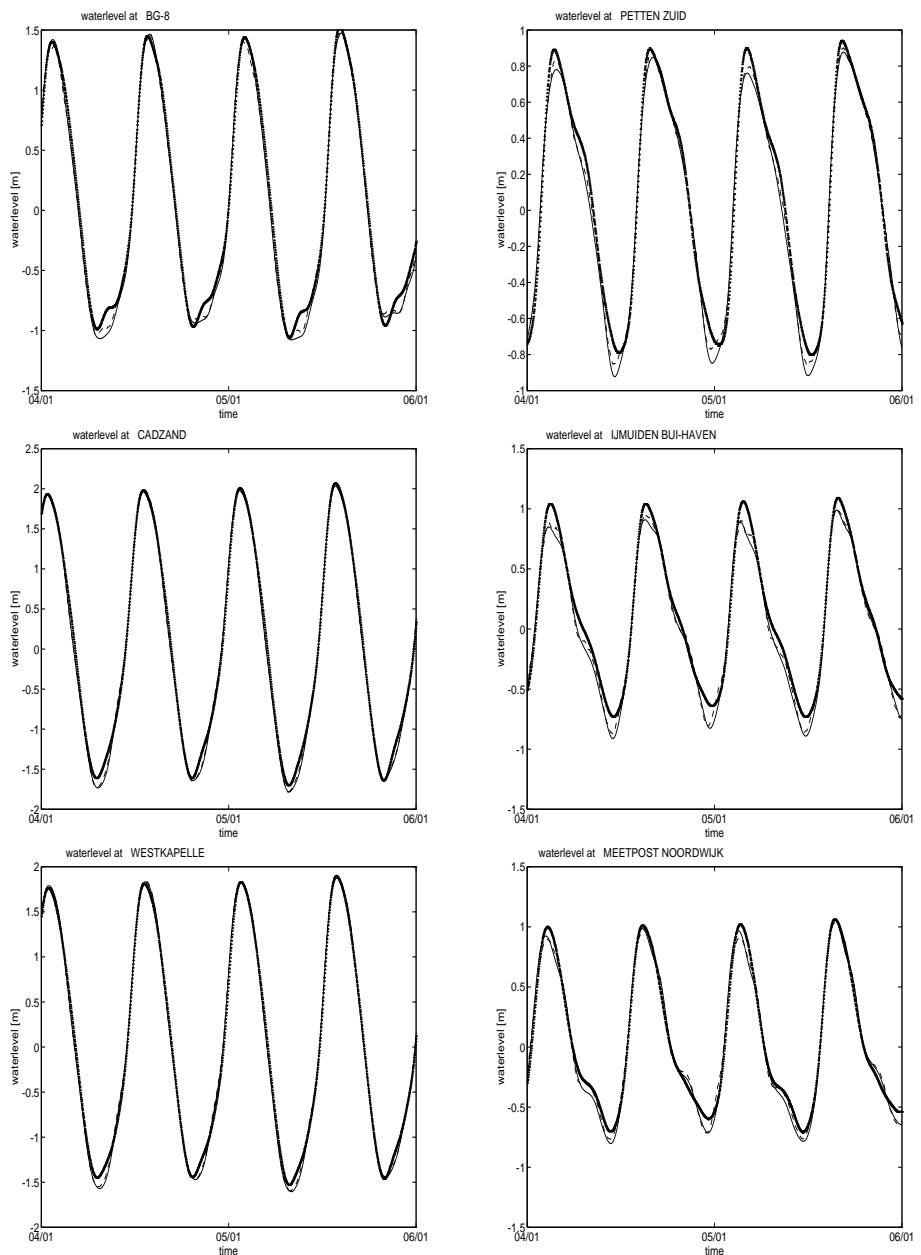


Figure 7.12: Water level series for model with new (dashed) and old (line) boundary and measurements (dotted)

The standard deviation and maximum error were computed over the same 2 day period shown in the figures. It may be possible to improve this a little by tuning the parameters of the smoother or by decreasing further the (small) fluctuations in the smoother estimate. Because the smoother estimates are computed by direct simulation of a corrected boundary and the fact that the standard deviations of the errors are quite constant along the coast, it is likely that the accuracy of the smoother is similar (approximately 3[cm]) over most of the model domain. However, additional measurements further from the coast are needed to validate this.

The computational requirements of the filter and smoother are almost the same for the same number of modes. The increase in size of the state vector is relatively small and also the additional computations for the smoothed boundary are small per mode. However, the RRSQRT smoother needed a larger number of modes to give accurate results in the experiments above. For the RRSQRT filter  $q = 50$  modes was sufficient while for the smoother  $q = 100$  modes were needed. Since this application is of-line the additional computational requirement is outweighed by the increased accuracy.

## 7.4 Conclusions

In this chapter a method is proposed for accurate reconstruction of water levels. The method is based on a modified RRSQRT version of the fixed-lag Kalman smoother. In this approach the smoother estimate is computed by first computing a smoother estimate of the forcing, here the open boundary, and a regular model simulation for the interior of the model domain. Several experiments were performed that indicate that the proposed smoother is very accurate along the coast, approximately 3[cm] standard deviation of the error. Also some indirect evidence was given that the method also computes accurate estimates further from the coast, but the true accuracy there still has to be determined. The method was also compared with direct model simulation and the RRSQRT Kalman filter. The estimates computed by the smoother are clearly more accurate than the ones produced by direct simulation of the model although this clearly depends on the calibration of the open boundary of the model. The RRSQRT filter gives results of similar accuracy as the smoother near the coast, but the experiments indicate that the smoother is more accurate further from the coast.

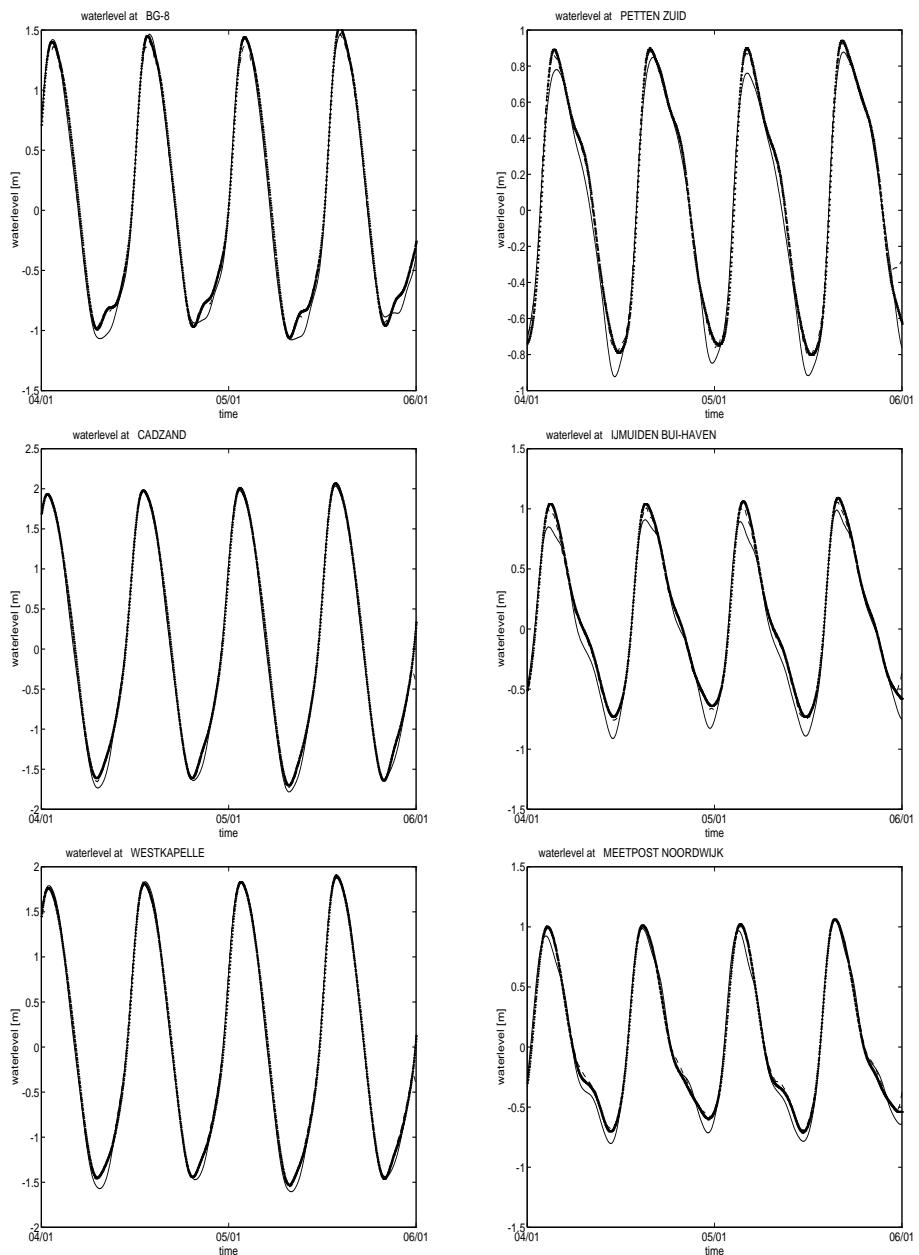


Figure 7.13: Water level series for smoother (dashed), model (line) and measurements (dots) at the boundary

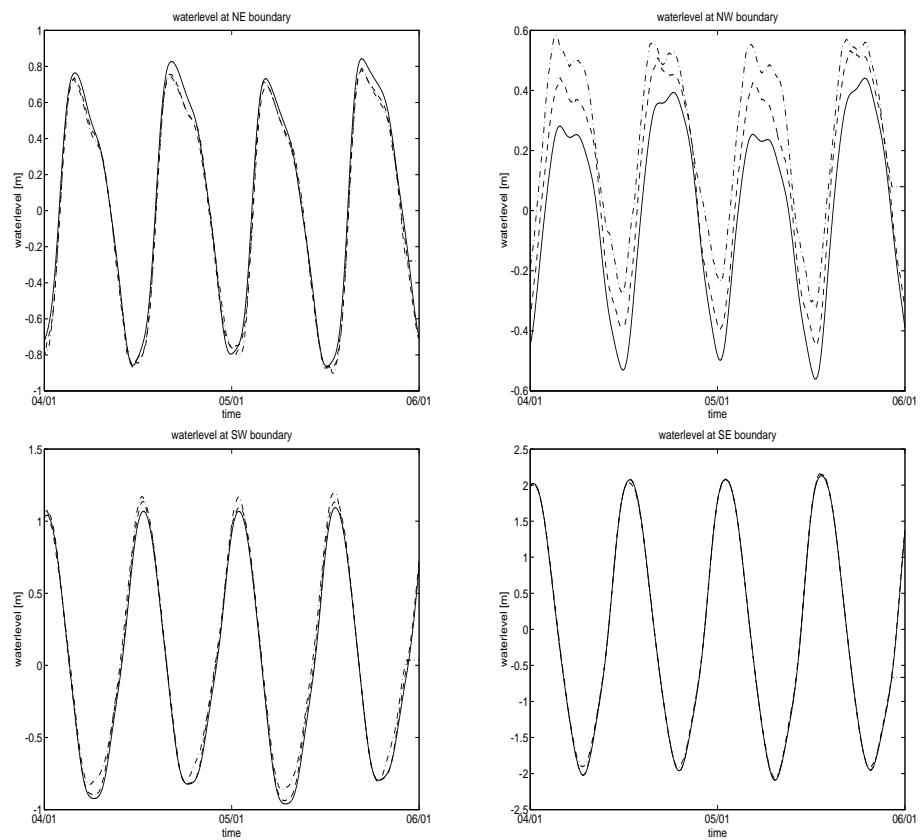


Figure 7.14: Water level series for smoother (dash-dot), filter (dashed) and model (line) at the corners of the open boundary

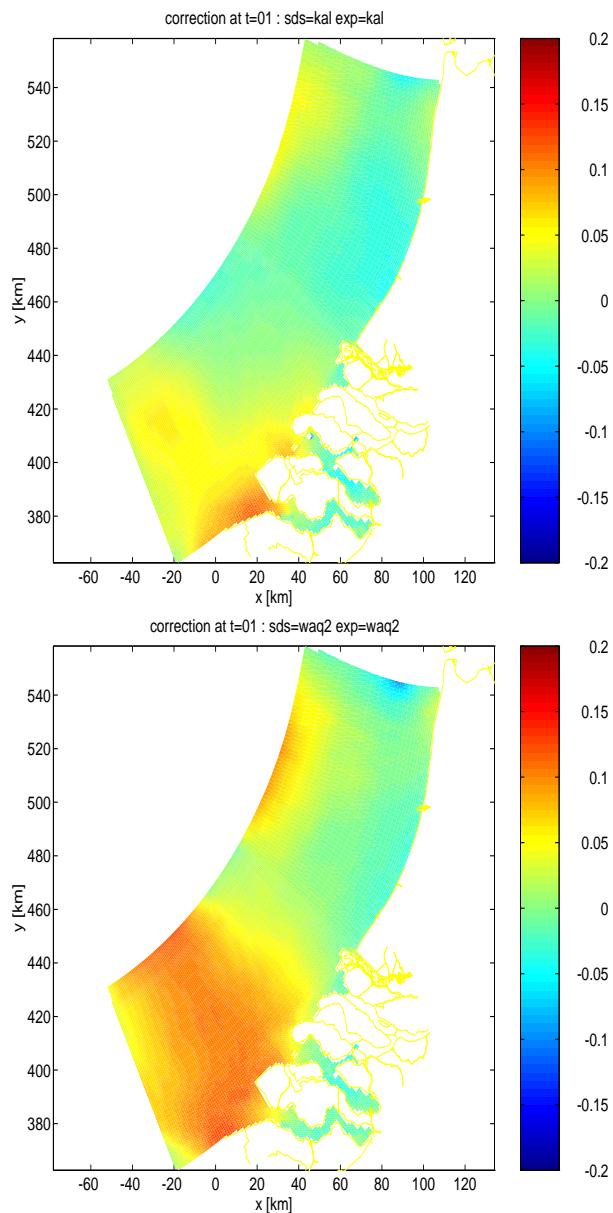


Figure 7.15: Corrections at January 05 21:00h for Kalman filter and smoother

## Chapter 8

# Application of Kalman filtering to three dimensional shallow water flow: On-line prediction of current profiles using HF-radar

### 8.1 Introduction to the application

In order to provide pilots guiding ships near the harbor of Rotterdam in the Netherlands with accurate current information a considerable effort has gone into the modeling of the region around the 'Hook of Holland'.

The currents in this region to a large extend are influenced by the flow of fresh water from the Meuse and Rhine rivers and by the wind. This necessitates the on-line forecasting of the currents, since the number of possible combinations of wind and river outflow is simply too large to tabulate. Moreover, the clear three dimensional structure of the currents makes the use of a three dimensional model necessary.

At the moment, there is no model that can accurately reproduce the complicated current patterns in area around Hook of Holland under all conditions and thus produce reliable forecasts for guiding ships. The HF-radar can accurately measure the currents in this area, but only for the upper 1 meter of the water column. The aim in this project is to integrate these two sources of information. This way the HF-radar measurements are extrapolated to greater depths and the model forecasts are improved.

## 8.2 The model

The flow in rivers, sea's and estuaries can be modeled quite accurately by the three dimensional shallow water equations. These equations can be derived from the well known Navier-Stokes equation by assuming hydrostatic pressure, using the Boussinesc approximation and Reynolds averaging in time. This results in the following equations

$$\frac{\partial \xi}{\partial t} + \frac{\partial (\int_{-D}^{\xi} u dz)}{\partial x} + \frac{\partial (\int_{-D}^{\xi} v dz)}{\partial y} = 0 \quad (8.1)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} + \frac{\frac{\partial p}{\partial x}}{\rho_w} - fv - \frac{\frac{\partial \tau_x}{\partial z}}{\rho_w} = 0 \quad (8.2)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} + \frac{\frac{\partial p}{\partial y}}{\rho_w} + fu - \frac{\frac{\partial \tau_y}{\partial z}}{\rho_w} = 0 \quad (8.3)$$

$$w + \frac{\partial (\int_{-D}^z u dz)}{\partial x} + \frac{\partial (\int_{-D}^z v dz)}{\partial y} = 0 \quad (8.4)$$

with

$$p = p_a + g \int_{z=-D}^{\xi} \rho(s) dz \quad (8.5)$$

$$\tau_x = \begin{cases} \rho \nu_v \frac{\partial u}{\partial z} & -h < z < \xi \\ c_D \rho_{air} |(u_w, v_w)| u_w & z = \xi \\ \frac{-g}{C^2} |(u, v)| u & z = -h \end{cases} \quad (8.6)$$

$$\tau_y = \begin{cases} \rho \nu_v \frac{\partial v}{\partial z} & -h < z < \xi \\ c_D \rho_{air} |(u_w, v_w)| v_w & z = \xi \\ \frac{-g}{C^2} |(u, v)| v & z = -h \end{cases} \quad (8.7)$$

where

variable	description
$t$	time
$(x, y)$	position in horizontal plane
$z$	position above reference plane
$\xi$	water level above the reference plane
$u, v, z$	current in $x, y$ and $z$ (=up) directions
$p$	pressure
$s$	salinity
$\rho_w$	density of the water
$f$	constant for Coriolis force
$d$	depth below reference level
$D = d + \xi$	total water depth
$g$	acceleration of gravity
$\tau_x, \tau_y$	stresses in $x$ and $y$ directions
$\nu_v$	eddy viscosity in vertical direction
$\nu_h$	eddy viscosity in horizontal direction
$c_D$	drag coefficient in wind stress
$u_w, v_w$	wind velocity in $x$ and $y$ directions
$T$	temperature

Equation 8.1,8.4 describe the conservation of mass, equations 8.2 and 8.3 the conservation of momentum. Equation 8.5 shows the hydrostatic pressure relation and equations 8.6 and 8.7 the computation of stresses between horizontal layers, top and bottom.

If there are significant differences in salinity, like in our application, then the density can not be assumed constant. The following empirical formula's (Eckhart) were used

$$\rho_w = \frac{p_0}{\lambda + \alpha_0 p_0} \quad (8.8)$$

$$\lambda = 1779.5 + 11.25T - 0.0745T^2 - (3.80 + 0.01T)s \quad (8.9)$$

$$\alpha_0 = 0.6980 \quad (8.10)$$

$$p_0 = 5890 + 38T - 0.37T^2 + 3s \quad (8.11)$$

To obtain a closed set of equations, an additional transport equation for the salinity is needed

$$\frac{\partial s}{\partial t} + \frac{\partial us}{\partial x} + \frac{\partial vs}{\partial y} + \frac{\partial ws}{\partial z} - \frac{\partial D_h \frac{\partial s}{\partial x}}{\partial x} - \frac{\partial D_h \frac{\partial s}{\partial y}}{\partial y} - \frac{\partial D_v \frac{\partial s}{\partial z}}{\partial z} = 0 \quad (8.12)$$

where  $D_h$  and  $D_v$  are the horizontal and vertical eddy diffusivity coefficients. Often the eddy diffusivity is related to the eddy viscosity using the Schmidt number  $\sigma$ .

$$D_h = \nu_h / \sigma_h \quad (8.13)$$

$$D_v = \nu_v / \sigma_v \quad (8.14)$$

The turbulence models used for the experiments are both based on the Prandtl mixing length principle (see [63] for an overview). The first turbulence model is algebraic. The turbulent kinetic energy at surface and bottom are derived from the stresses at the surface and the bottom. Linear interpolation is used in between. Finally the mixing length is based on the distance to surface and bottom.

$$L = \kappa(z + d) \sqrt{1 - \frac{z + D}{H}} \quad (8.15)$$

$$k = \frac{1}{\rho c_\mu} \left( \tau(z = -h)(1 - \frac{z + d}{H}) + \tau(z = \xi) \frac{z + d}{H} \right) \quad (8.16)$$

$$\nu_v = c_{\mu'} L \sqrt{k} \quad (8.17)$$

where

variable	description
$k$	turbulent kinetic energy
$L$	mixing length
$\kappa \approx 0.41$	von Karman constant
$c_{\mu'} \approx 0.58$	calibration constant
$c_\mu \approx 0.09$	calibration constant

In more complex flows the direct specification of a mixing length is difficult and it is more convenient to use the dissipation rate of kinetic energy. The  $k - \epsilon$  model is based on this idea and uses advection equations for both the turbulent kinetic energy  $k$  and the dissipation rate  $\epsilon$ .

$$\begin{aligned} \frac{\partial k}{\partial t} + \frac{\partial u k}{\partial x} + \frac{\partial v k}{\partial y} + \frac{\partial w k}{\partial z} - \frac{\partial D_h \frac{\partial k}{\partial x}}{\partial x} - \frac{\partial D_h \frac{\partial k}{\partial y}}{\partial y} - \frac{\partial D_v \frac{\partial k}{\partial z}}{\partial z} &= P_k + B_k - \epsilon \\ \frac{\partial \epsilon}{\partial t} + \frac{\partial u \epsilon}{\partial x} + \frac{\partial v \epsilon}{\partial y} + \frac{\partial w \epsilon}{\partial z} - \frac{\partial D_h \frac{\partial \epsilon}{\partial x}}{\partial x} - \frac{\partial D_h \frac{\partial \epsilon}{\partial y}}{\partial y} - \frac{\partial D_v \frac{\partial \epsilon}{\partial z}}{\partial z} &= P_\epsilon + B_\epsilon - \epsilon \\ \nu_v &= C_\mu \frac{k^2}{\epsilon} \end{aligned}$$

where  $P_k, P_\epsilon, \epsilon, \epsilon_k$  are production and dissipation terms and the terms  $B_k$  and  $B_\epsilon$  are damping terms for vertical density gradients.

To complete the model various boundary conditions are needed. Their treatment falls outside the scope of this paper and the interested reader is referred to Leendertse ([63]).

To simulate the model the equations above were discretized. For efficiency a staggered grid was used. In time an Alternating Directions Implicit method for the horizontal plane is combined with a fully implicit solution for the vertical direction. In the horizontal plane curvilinear coordinates were used; in the vertical  $\sigma$  coordinates. A detailed description can be found in [53].

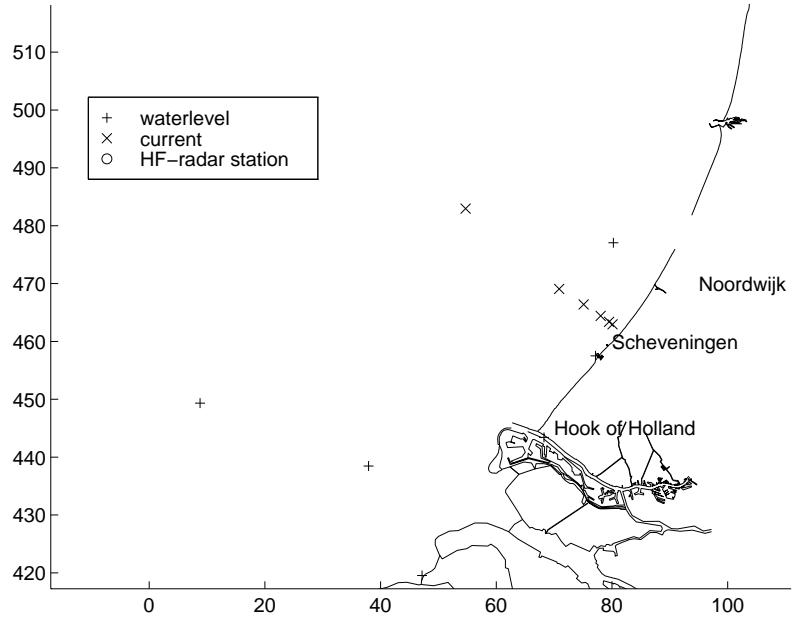


Figure 8.1: Measurement locations

### 8.3 Measurements

Four types of measurements were used for the experiments: water level measurements at some locations near the coast, current measurements at several points and depths along a transect starting near Scheveningen, measured wind velocity and direction at Noordwijk and HF-radar measurements. Figure 8.1 shows the locations of the various measuring devices.

The HF-radar provides measurements of currents near the surface. It is based on measurement of the Doppler shift of the return signal. From the Doppler shift the radial component of the current can be deduced. Therefore two stations are needed. The radar signal can only penetrate about one meter into the sea-water and thus the return signal is only influenced by currents in the upper meter. The radar signal is transmitted in a narrow beam, the angle of which is constantly adapted in order to scan the whole area. The measurements cover an area of about 25km by 25km with a resolution of 1km. Thus a synoptic picture is produced, that is updated every 30 minutes.

The measurements used in this paper were obtained as a part of the European MAST project in October 1990. The area of interest of this campaign was slightly north of the harbor area, but many of the physical processes are similar in this area.

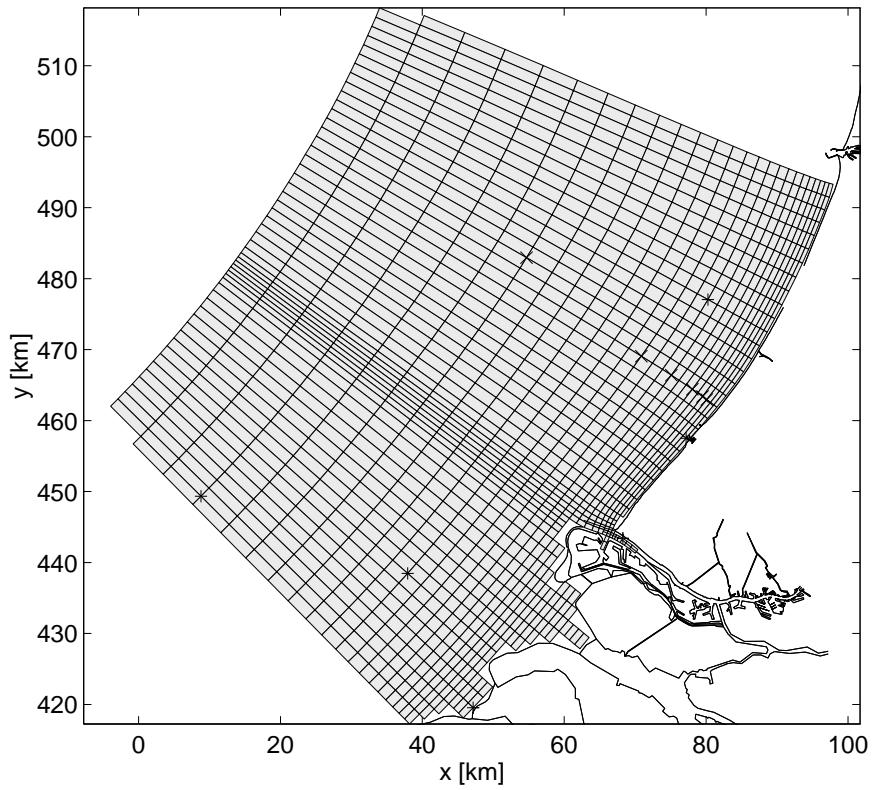


Figure 8.2: Model grid

## 8.4 Feasibility study

To assess the accuracy of the present model, the accuracy of the HF-radar measurements and the possibility of data assimilation a large number of model runs was performed and the results were compared with the various measurements.

The grid of the model used in the experiments is shown in figure 8.2. The model was deliberately taken quite small since the aim is to use data assimilation for the correction for errors in the external forcing.

For such a small scale model most of the setup, which is caused by wind and pressure gradients, is generated outside the model and should therefore be included into to boundary conditions. In practice the meteorological effect at the open boundary is not known. For this reason usually a large model with a coarser grid is used to compute these effects. In this project we use the measured set up in Hook of Holland to correct the open boundary. After some filtering with a Godin filter [36] this produced accurate results for the water levels in the model (see figure 8.3). This shows that even a very simple data assimilation

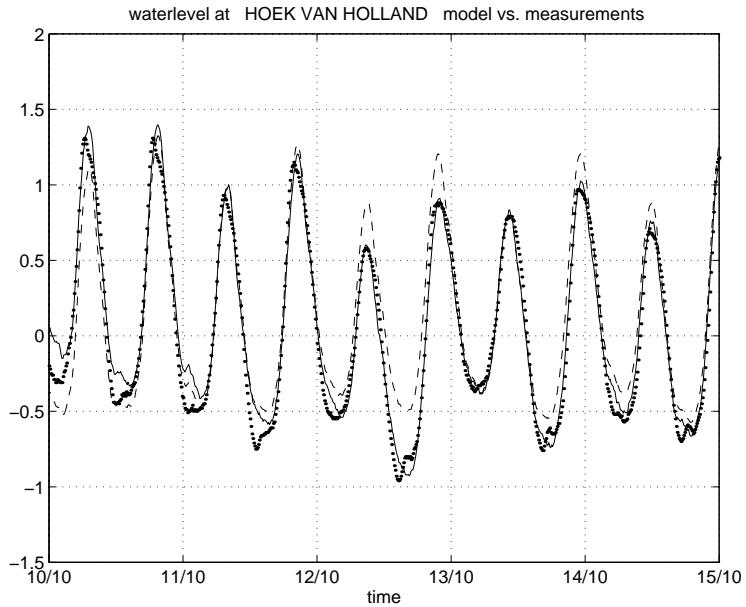


Figure 8.3: Water levels with astronomic (dashed) and corrected (line) boundary conditions vs. measurements (dotted)

scheme can eliminate the need for a larger model in case only now-casts and hind-casts are needed.

A second external forcing of the model is the stress caused by wind on the water surface. In larger models the wind-input is often produced by a meteorological model. In this case the measured winds at Noordwijk were used as a global input for the model. The errors of using global time-series for the wind forcing are expected to be small, since the correlation of the measured wind series of Noordwijk and Hook of Holland had a correlation of 0.9 over October '90.

Although the water levels can be reproduced quite accurately by the model, it is much more difficult to obtain accurate currents. In long shore direction the depth averaged current is dominated by the tide, which can be modeled quite well. The profile however is strongly influenced by the stratification, due to the inflow of fresh-water from the Rhine and Meuse rivers. The vertical mixing of the fresh-water and salt seawater is in turn influenced by the turbulence generated by the tide and wind and damped by smaller turbulent fluctuations near the salinity gradient.

A very important secondary effect are the cross-shore currents generated by the Coriolis force acting on the differences in long-shore currents at different depths (see [66, 65] for a description of this effect, called Tidal Straining). Figure 8.4 shows some cross-shore current time-series with the algebraic turbulence

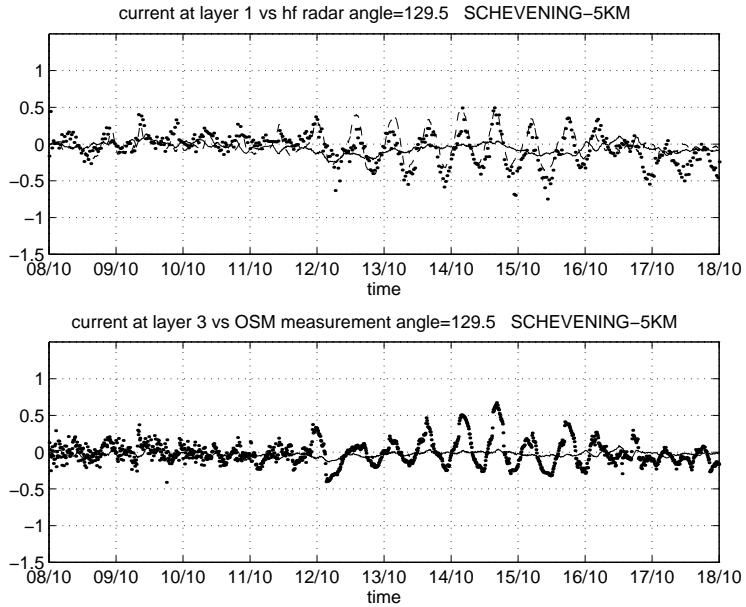


Figure 8.4: Cross-shore currents for algebraic (line) and  $k-\epsilon$  model (dashed) vs. measurements (dotted)

model and with the  $k-\epsilon$  turbulence model. It is clear that only the  $k-\epsilon$  model can capture the tidal straining effects during periods with large stratification (October 8–12). Before October 8 and after October 12 the wind was so strong that the water was well mixed. Further experiments indicate that the model results are very sensitive to vertical resolution and the various parameters that control the mixing rate. Therefore, some effort will be spent in further adaptation of the parameters in the future.

For the remaining differences between model and measurements we believe the main causes are inaccurate formulation of the wind-stress as a function of the wind velocity and errors in the vertical mixing rate. The first type of errors can be observed from the HF-radar measurements of the surface currents, while errors of the second type can be detected using measurements of the current profile. The use of ‘acoustic Doppler’ measurements can produce the accurate current profiles needed for this.

The fact that the main errors in the model results can be observed with the HF-radar and profile measurements shows that both sources of information are complementary. It thus is expected that data assimilation can have a significant impact.

## 8.5 Kalman filter experiments

In order to improve model predictions and to extrapolate the radar surface current measurements over the vertical the measurements can be assimilated into the model. Since the surface current measurements will only improve the accuracy locally and estimates are needed only locally a model not much larger than the area covered by the radar will be used, in this case the model area shown in figure 8.2). The local model will by itself not be able to forecast wind set-up. By assimilating some water level measurements the set-up at the open boundary can be corrected. Since the surface current measurements and the water level measurements contain almost no information about the differences in vertical direction of the currents it may be necessary to assimilate additional measurements of the current profile at some locations.

### 8.5.1 Application of the RRSQRT algorithm with the 3-D shallow water model

The implementation of the Kalman filter for this application is based on the RRSQRT filter. A 'full' Kalman filter is not feasible for a 3-D model because the size of the state vector is almost  $5 \cdot 10^4$  elements large even for this small model. For example storing the covariance matrix would require 10 Gigabytes.

The state vector of the model shown in section 8.2 consists of Water levels  $\xi$ , current velocities  $(u, v, w)$ , salinities  $s$  and turbulent kinetic energy and dissipation  $k, \epsilon$  in case a  $k - \epsilon$  model is used. To apply the RRSQRT filter an interface must be constructed for these variables.

The main uncertainty in the model is assumed to be caused by the set-up at the open boundary and uncertainty in the wind-forcing at the surface. The uncertainty at the open boundary is modeled using an AR(1) process [8] at a few locations with linear interpolation in between.

The uncertainty in the wind-forcing is modeled by adding the noise directly to the stress at the surface.

$$\Delta\tau_x(x, k) = \alpha_\tau \Delta\tau_x(x, k-1) + \varepsilon_{\Delta\tau_x}(x, k) \quad (8.18)$$

$$\Delta\tau_y(x, k) = \alpha_\tau \Delta\tau_y(x, k-1) + \varepsilon_{\Delta\tau_y}(x, k) \quad (8.19)$$

The spatial correlation is given by

$$\frac{E[\Delta\tau_x(x_1, k)\Delta\tau_x(x_2, k)]}{\sqrt{E[\Delta\tau_x(x_1, k)^2]}\sqrt{E[\Delta\tau_x(x_2, k)^2]}} = e^{-|x_1-x_2|/d_\tau} \quad (8.20)$$

$$\frac{E[\Delta\tau_y(x_1, k)\Delta\tau_y(x_2, k)]}{\sqrt{E[\Delta\tau_y(x_1, k)^2]}\sqrt{E[\Delta\tau_y(x_2, k)^2]}} = e^{-|x_1-x_2|/d_\tau} \quad (8.21)$$

$$E[\Delta\tau_x(x_1, k)\Delta\tau_y(x_2, k)] = 0 \quad (8.22)$$

where

$$\sqrt{E[\Delta\tau_x(x_1, k)^2]} = \sqrt{E[\Delta\tau_y(x_1, k)^2]} = \sigma_\tau \quad (8.23)$$

and  $d_\tau$  is a characteristic distance for the correlation. To reduce the number of noise inputs the noise is first introduced on a coarser grid and subsequently interpolated. The interpolation is performed using bilinear interpolation.

### 8.5.2 Experiment and results

In order to study the impact of assimilation HF-radar measurements into a 3-D shallow water model the RRSQRT filter was applied. Both radar measurements of surface currents and the measurements of two water level stations (BG-8 and Scheveningen) were assimilated. The other water level measurements and all in-situ current measurements were used for validation. The model and Kalman filter computations used salinity transport and an algebraic turbulence model.

As a first guess, the water level was computed with a model of the continental shelf of the North sea. The AR(1) model for the uncertainty used the following values for the parameters: time correlation 0.95 per time-step of 5[min] and a standard deviation  $\sigma_b = 0.2[m]$ . The parameters in equations 8.20–8.21 for the uncertainty in the wind-stress were  $d_\tau = [grid - cells]$ ,  $\alpha_\tau = 0.95$  per time-step and  $\sigma_\tau = 0.58[m^2/s]$ . The measurements were assumed to be independent in time and space with standard deviations 0.2[m/s] for the HF-radar and 0.1[m] for the water levels.

The charts in figures 8.5, 8.6 and 8.7 show a comparison of radar current measurements, model results and the filter results. The impact of the radar measurements can clearly be seen. The currents of the model are changed significantly in the direction of the measurements.

From a comparison of the filter results with the validation measurements it is possible to assess the true impact of the radar observations. In figure 8.8 an example of a comparison with in-situ current measurements is shown. Also here, the impact of the radar observation on the assimilation results is clearly positive.

Figure 8.9 shows a comparison with assimilated and validation water level measurements. The assimilation of the observation has a positive impact on the results. The water levels at the validation station Hoek van Holland have become consistent with the other observations because of the assimilation. The impact of the water level measurements is probably larger than that of the radar measurements in this case.

Summarising these results, the RRSQRT filter shows encouraging results. The accuracy of the filter now-cast can probably be improved significantly by tuning the filter and using the  $k - \varepsilon$  turbulence model. Further research in this direction is being carried out at the moment.

## 8.6 Conclusions

Due to the complicated structure of the currents at the entrance of the Rotterdam harbour it is very difficult to provide accurate model predictions even using three dimensional hydrodynamic models. A feasibility study indicates that a

Figure 8.5: Chart of surface currents for model, hf-radar and filter

Figure 8.6: Chart of surface currents for model, hf-radar and filter

Figure 8.7: Chart of surface currents for model, hf-radar and filter

Figure 8.8: Time-series of currents for model, measurements and filter

Figure 8.9: Time-series of water level for model, measurements and filter

three dimensional shallow water model with a  $k - \epsilon$  turbulence model is able to predict the currents with a reasonable accuracy and this accuracy can probably be improved by further adaptation of the parameters that control the vertical mixing. However, especially in windy conditions the results can be improved significantly by assimilating remote sensing measurements of the surface currents. Furthermore by incorporating water level measurements the need for a larger (nested) model to compute the wind set-up at the open boundary is eliminated, so that the system can run with a local model and local measurements only.

The data assimilation procedure is again based on an extended Kalman filter version of the RRSQRT filter. Both HF-radar measurements and water level measurements are assimilated into the three dimensional model. The filter prediction is improved significantly when compared to the assimilated radar measurements, which shows that the assimilation procedure works satisfactorily. Also compared to several in-situ validation measurements the assimilation of radar measurements improves the predicted currents significantly.

## Appendix A

# Proof of convergence of the RRSQRT filter

**Lemma 2** Let  $\mathbf{P}_1, \mathbf{P}_2 \in R^{n \times n}$ ,  $\mathbf{P}_1 = \mathbf{P}'_1$ ,  $\mathbf{P}_2 = \mathbf{P}'_2$  and  $\mathbf{P}_1 \geq \mathbf{P}_2 > 0$  then  $\mathbf{P}_2^{-1} \geq \mathbf{P}_1^{-1}$

### Proof:

Since  $\mathbf{P}_1 > 0$  and  $\mathbf{P}_2 > 0$   $\mathbf{P}_1^{-1}$  and  $\mathbf{P}_2^{-1}$  exist. Let  $\mathbf{P}_2 = \mathbf{L}_{\mathbf{P}_2} \mathbf{L}'_{\mathbf{P}_2}$  be a square root (eg. Cholesky) factorization of  $\mathbf{P}_2$ . If  $\mathbf{P}_1 \geq \mathbf{P}_2$

$$\begin{aligned}\mathbf{P}_1 \geq \mathbf{P}_2 &\Rightarrow \mathbf{P}_1 - \mathbf{P}_2 \geq 0 \\ &\Rightarrow \mathbf{P}_1 - \mathbf{L}_{\mathbf{P}_2} \mathbf{L}'_{\mathbf{P}_2} \geq 0 \\ &\Rightarrow (\mathbf{L}_{\mathbf{P}_2}^{-1} \mathbf{P}_1 \mathbf{L}_{\mathbf{P}_2}^{-T}) - \mathbf{I} \geq 0\end{aligned}$$

Let  $\mathbf{UDU}'$  be a eigendecomposition of  $(\mathbf{L}_{\mathbf{P}_2}^{-1} \mathbf{P}_1 \mathbf{L}_{\mathbf{P}_2}^{-T})$  then

$$\begin{aligned}\mathbf{P}_1 \geq \mathbf{P}_2 &\Rightarrow (\mathbf{L}_{\mathbf{P}_2}^{-1} \mathbf{P}_1 \mathbf{L}_{\mathbf{P}_2}^{-T}) - \mathbf{I} \geq 0 \\ &\Rightarrow \mathbf{UDU}' - \mathbf{I} \geq 0 \\ &\Rightarrow \mathbf{D} - \mathbf{I} \geq 0 \\ &\Rightarrow \mathbf{D}^{-1} - \mathbf{I} \leq 0 \\ &\Rightarrow \mathbf{UD}^{-1}\mathbf{U}' - \mathbf{I} \\ &\Rightarrow (\mathbf{L}_{\mathbf{P}_2}^{-1} \mathbf{P}_1 \mathbf{L}_{\mathbf{P}_2}^{-T})^{-1} - \mathbf{I} \leq 0 \\ &\Rightarrow (\mathbf{L}'_{\mathbf{P}_2} \mathbf{P}_1^{-1} \mathbf{L}_{\mathbf{P}_2} - \mathbf{I} \leq 0 \\ &\Rightarrow \mathbf{P}_1^{-1} - \mathbf{P}_2^{-1} \leq 0 \\ &\Rightarrow \mathbf{P}_2^{-1} - \mathbf{P}_1^{-1} \geq 0\end{aligned}$$

**Theorem 1**

If

- the system (eqns. 2.30, 2.31) is time-invariant, i.e.  $\forall_k \mathbf{A}(k) = \mathbf{A}, \mathbf{B}(k) = \mathbf{B}, \mathbf{C}(k) = \mathbf{C}, F(k) = F, \Sigma_s(k) = \Sigma_s$  and  $\Sigma_o(k) = \Sigma_o > 0$
- and the following limit exists

$$\lim_{k \rightarrow \infty} \mathbf{P}(k+1|k) = \mathbf{P} \quad (\text{A.1})$$

(from eqn. 3.6  $\lim_{k \rightarrow \infty} \mathbf{K}(k) = \mathbf{K}$  exists) and

$$\Phi := (\mathbf{I} - \mathbf{K}\mathbf{C})\mathbf{A} \quad (\text{A.2})$$

is stable

then

for every  $\varepsilon > 0$  there exists a  $\delta > 0$  such that

if  $\forall_k \|\mathbf{T}_k\| < \delta$  then

$$\forall_k \|\mathbf{P}(k|k) - \mathbf{P}_c(k|k)\| < \varepsilon \quad (\text{A.3})$$

**Proof:**

The proof is based on an induction argument moving  $M$  time-steps at a time. This way the stability of  $\Phi$  can be exploited for obtaining an upper bound of the error  $\|\mathbf{P}(k|k) - \mathbf{P}_c(k|k)\|$ . The induction is started at  $k = N$  where  $k$  is sufficiently large so that a steady state is almost reached.

Instead of working with  $\|\mathbf{P}(k|k) - \mathbf{P}_c(k|k)\|$  directly, we will try first to bound  $\|\mathbf{P}_t(k|k) - \mathbf{P}_c(k|k)\|$ . If we can obtain  $\|\mathbf{P}_t(k|k) - \mathbf{P}_c(k|k)\| < \varepsilon$  then from equation 5.33 it follows that

$$\|\mathbf{P}(k|k) - \mathbf{P}_c(k|k)\| \leq \|\mathbf{P}_t(k|k) - \mathbf{P}_c(k|k)\| \leq \varepsilon \quad (\text{A.4})$$

where  $\|\cdot\|$  in this inequality must be the spectral norm. Now define

$$\mathbf{E}_2(k|k) := \mathbf{P}_t(k|k) - \mathbf{P}_c(k|k) \quad (\text{A.5})$$

$$\mathbf{E}_2(k+1|k) := \mathbf{P}_t(k+1|k) - \mathbf{P}_c(k+1|k) \quad (\text{A.6})$$

Subtracting equations 5.25, 5.26, 5.31 and 5.27, 5.28, 4.4 one can write the evolution of  $\mathbf{E}_2$  as

$$\mathbf{E}_2(k+1|k) = \mathbf{A}\mathbf{E}_2(k|k)\mathbf{A}' + \mathbf{T}(k+1) \quad (\text{A.7})$$

$$\mathbf{E}_2(k+1|k+1) = (\mathbf{I} - \mathbf{K}_c(k+1))\mathbf{E}_2(k+1|k)(\mathbf{I} - \mathbf{K}_c(k+1))' \quad (\text{A.8})$$

We proceed by bounding the error  $\mathbf{E}_2$  for the initial  $N$  time-steps. In order to obtain  $\|\mathbf{E}_2(N|N)\| = \|\mathbf{P}_t(N|N) - \mathbf{P}_c(N|N)\| < \varepsilon$  consider the map

$$g : (\mathbf{T}(1), \dots, \mathbf{T}(N)) \mapsto \mathbf{E}_2(N|N) \quad (\text{A.9})$$

which can be obtained by a combination of equations A.7, A.8. Since  $\Sigma_o > 0$  the matrix  $\mathbf{C}\mathbf{P}_c(k+1|k)\mathbf{C}' + \Sigma_o$  is never singular and thus  $g$  is continuous. For  $\mathbf{T}(1) = 0, \dots, \mathbf{T}(N) = 0$  the algorithm is exact and thus  $g(0, \dots, 0) = 0$ . This implies

$$\exists \delta_1 > 0 \forall \|\mathbf{T}(1)\| < \delta_1, \dots, \|\mathbf{T}(N)\| < \delta_1 \quad \|\mathbf{P}_t(N|N) - \mathbf{P}_c(N|N)\| < \varepsilon \quad (\text{A.10})$$

which concludes the initial step. Note that  $N$  can still be chosen freely.

Let, for some  $k \geq N$ ,  $\|\mathbf{P}_t(k|k) - \mathbf{P}_c(k|k)\| < \varepsilon$ . Repeated application of equations A.7 and A.8 gives

$$\begin{aligned} \mathbf{E}_2(k+M|k+M) &= \Phi_c(k, k+M)\mathbf{E}_2(k|k)\Phi_c(k, k+M)' \\ &+ \sum_{i=1}^M (\Phi_c(k+i|k+M)(\mathbf{I} - \mathbf{K}_c(k+i)\mathbf{C})\mathbf{T}(k+i) \\ &\quad (\mathbf{I} - \mathbf{K}_c(k+i)\mathbf{C})'\Phi_c(k+i|k+M)') \end{aligned} \quad (\text{A.11})$$

from which we obtain the following upper bound

$$\begin{aligned} \|\mathbf{E}_2(k+M|k+M)\| &= \|\Phi_c(k, k+M)\|^2 \mathbf{E}_2(k|k) \\ &+ \sum_{i=1}^M (\|\Phi_c(k+i|k+M)(\mathbf{I} - \mathbf{K}_c(k+i)\mathbf{C})\|^2 \\ &\quad \|\mathbf{T}(k+i)\|) \end{aligned} \quad (\text{A.12})$$

If  $\|\Phi_c(k, k+M)\|$  can be made smaller than  $1/2$  and  $\|\Phi_c(k+i|k+M)(\mathbf{I} - \mathbf{K}_c(k+i)\mathbf{C})\|$  can be bounded by some positive real number  $\phi$  then equation A.12 results in

$$\begin{aligned} \|\mathbf{E}_2(k+M|k+M)\| &= (1/4)\mathbf{E}_2(k|k) \\ &+ \sum_{i=1}^M \phi^2 \|\mathbf{T}(k+i)\| \end{aligned} \quad (\text{A.13})$$

From which  $\|\mathbf{E}_2(k+M|k+M)\| < \varepsilon$  is obtained if

$$\|\mathbf{T}(k+i)\| < \delta < \frac{3\varepsilon}{4\phi^2} \quad (\text{A.14})$$

It remains to be shown that  $\|\Phi_c(k, k+M)\| < 1/2$  and that  $\|\Phi_c(k+i|k+M)(\mathbf{I} - \mathbf{K}_c(k+i)\mathbf{C})\| < \phi$  for  $\|\mathbf{E}_2(k|k)\| < \varepsilon$  and  $\|\mathbf{T}(k)\| < \delta$ .

Define

$$\Phi(k) := (\mathbf{I} - \mathbf{K}(k+1)\mathbf{C})\mathbf{A} \quad (\text{A.15})$$

$$\Phi_c(k) := (\mathbf{I} - \mathbf{K}_c(k+1)\mathbf{C})\mathbf{A} \quad (\text{A.16})$$

$$\Phi(k, k+j) := \prod_{i=k}^{k+j-1} (\mathbf{I} - \mathbf{K}(k+1)\mathbf{C})\mathbf{A} \quad (\text{A.17})$$

$$\Phi_c(k, k+j) := \prod_{i=k}^{k+j-1} (\mathbf{I} - \mathbf{K}_c(k+1)\mathbf{C})\mathbf{A} \quad (\text{A.18})$$

The first step is to find a bound for  $\|\Phi_c(k, k+M)\|$  for some  $M$ . Since  $\Phi$  is stable there exists an  $M$  such that  $\|\Phi^M\| < 1/4$ . The map

$$f : (\kappa_1, \dots, \kappa_M) \mapsto \prod_{i=1}^M (\mathbf{I} - \kappa_i \mathbf{C})\mathbf{A} \quad (\text{A.19})$$

is continuous and for  $\kappa_1 = \mathbf{K}, \dots, \kappa_m = \mathbf{K}$  it maps to  $\Phi^M$ . Thus

$$\exists \gamma_1 > 0 \forall \|\kappa_1 - \mathbf{K}\| < \gamma_1, \dots, \|\kappa_M - \mathbf{K}\| < \gamma_M \Rightarrow \|f(\kappa_1, \dots, \kappa_M)\| < 1/2 \quad (\text{A.20})$$

which implies that if  $\|\mathbf{K}_c(k+1) - \mathbf{K}\| < \gamma_1, \dots, \|\mathbf{K}_c(k+M) - \mathbf{K}\| < \gamma_1$  then  $\|\Phi_c(k, k+M)\| < 1/2$ . Here we see that in order to bound  $\Phi^M$  a bound on  $\|\mathbf{K}_c(k) - \mathbf{K}\|$  is needed.

Bounding  $\|\mathbf{K}_c(k) - \mathbf{K}\|$  proceeds in two steps. If  $\|\mathbf{K}_c(k) - \mathbf{K}(k)\| < \gamma_1/2$  and  $\|\mathbf{K}(k) - \mathbf{K}\| < \gamma_1/2$  then

$$\begin{aligned} \|\mathbf{K}_c(k+1) - \mathbf{K}\| &= \|\mathbf{K}_c(k+1) - \mathbf{K}(k) + \mathbf{K}(k) - \mathbf{K}\| \\ &\leq \|\mathbf{K}_c(k) - \mathbf{K}(k)\| + \|\mathbf{K}(k) - \mathbf{K}\| \\ &= \gamma_1 \end{aligned} \quad (\text{A.21})$$

But because  $\lim_{k \rightarrow \infty} \mathbf{K}(k) = \mathbf{K}$

$$\exists N \geq 0 \forall k \geq N \|\mathbf{K}(k) - \mathbf{K}\| < \gamma_1/2 \quad (\text{A.22})$$

This gives us the number of time-steps  $N$  for the initial step of the induction.

To show that  $\|\mathbf{K}_c(k) - \mathbf{K}(k)\| < \gamma_1/2$  under some additional constraints on  $\|\mathbf{T}(k+i)\|$  and  $\|\mathbf{E}_2(k|k)\|$  consider the maps

$$h_{k,i} : (\mathbf{E}_2(k|k), \mathbf{T}(k+1), \dots, \mathbf{T}(k+M)) \mapsto \mathbf{K}_c(k+i) \quad (\text{A.23})$$

which can be obtained by repeated application of equations A.7, A.8. Since the map is composed of continuous parts it is continuous. Also  $h_{k,i}(0, \dots, 0) = \mathbf{K}(k+i)$  thus

$$\begin{aligned} \exists \varepsilon_1, \delta_2 > 0 \forall & \|\mathbf{T}(k+1)\| < \delta_2, \dots, \|\mathbf{T}(k+M)\| < \delta_2, \|\mathbf{E}_2(k|k)\| < \varepsilon_1 \\ & \|\mathbf{K}_c(k) - \mathbf{K}(k)\| < \gamma_1/2 \end{aligned}$$

which leads to  $\|\Phi_c(k, k + M)\| < 1/2$  under these conditions.

A similar continuity argument can be used to show that  $\|\Phi_c(k + i|k + M)(\mathbf{I} - \mathbf{K}_c(k + i)\mathbf{C})\|$  can be bounded from above by some real number  $\phi > 0$ .

For  $\delta < \min(\delta_1, \delta_2)$  all inequalities above are met, and the proof is concluded.

# Appendix B

## List of symbols

### Shallow water flow model

symbol	SI units	description
$B_k$	$m^2/s^3$	Dissipation of turbulent kinetic energy by buoyancy
$B_\epsilon$	$m^2/s^4$	Loss of dissipation due to buoyancy effects
$C$	$m^{1/2}/s$	Chézy coefficient
$C_d$		Wind drag coefficient
$c_f$	$1/s$	Constant for linearized friction
$c_\mu$		constant in turbulence model
$c'_\mu$		constant in turbulence model
$D_h$	$m^2/s$	Eddy diffusivity for horizontal plane
$D_v$	$m^2/s$	Eddy diffusivity for vertical
$E_\xi$	$J$	Potential energy of one grid cell
$E_{u,v}$	$J$	Kinetic energy of one grid cell
$f$	$1/s$	Constant for Coriolis force
$g$	$m/s^2$	Acceleration of gravity
$H$	$m$	Total water depth $H = D + \xi$
$k$	$m^2/s^2$	Turbulent kinetic energy divided by $\rho_w$
$L$	$m$	Prandtl mixing length
$P_k$	$m^2/s^3$	Production rate of turbulent kinetic energy (scaled by $1/\rho_w$ )
$P_\epsilon$	$m^2/s^4$	Production rate of dissipation (scaled by $1/\rho_w$ )
$p$	$N/m^2$	Hydrostatic pressure
$p_a$	$N/m^2$	Air pressure at surface
$p_0$	$kg/m^3$	Variable in Eckhart formula
$s$		Salinity (promille)
$T$	$K$	Temperature
$u$	$m/s$	Current velocity in $x$ -direction
$u_w$	$m/s$	wind velocity in $x$ -direction

symbol	SI units	description
$V$	$m/s$	Wind velocity
$v$	$m/s$	Current velocity in $y$ -direction or
$v_w$	$m/s$	wind velocity in $y$ -direction
$w$	$m/s$	current velocity in $z$ -direction
$x$	$m$	Cartesian coordinate in horizontal plane
$y$	$m$	Cartesian coordinate in horizontal plane
$z$	$m$	Cartesian coordinate for vertical
$\alpha_0$	$m^3/kg$	Constant in Eckhart formula
$\epsilon$	$m^2/s^3$	Dissipation of turbulent kinetic energy (scaled $1/\rho_w$ )
$\varepsilon_\epsilon$	$m^2/s^4$	Decay rate of dissipation
$\kappa$		von Karman constant
$\lambda$		Variable in Eckhart formula
$\nu_v$	$m^2/s$	Eddy viscosity used for vertical
$\nu_h$	$m^2/s$	Eddy viscosity used for horizontal
$\xi$	$m$	Water level above reference plane
$\xi_b$	$m$	Water level above reference plane at open boundary
$\rho_a$	$kg/m^3$	Density of air at surface
$\sigma_h$		Schmidt number for horizontal plane
$\sigma_v$		Schmidt number for vertical
$\rho_w$	$kg/m^3$	Density of sea water
$\tau_x$	$N/m^2$	Stress in $x$ -direction
$\tau_y$	$N/m^2$	Stress in $y$ -direction
$\psi$	$rad$	Wind angle

Discretization		
symbol	SI units	description
$k$		Index for time
$l$		Time index relative to $k$
$m$		Index in $x$ -direction
$N$		Number of grid-cells
$n$		Index in $y$ -direction
$u_m^k$	$m/s$	Discrete $u$ , $u_m^k \approx u(x = m\Delta x, t = k\Delta t)$
$\Delta t$	$s$	Time-step $t = k\Delta t$
$\Delta x$	$m$	Grid size in $x$ -dirextion
$\Delta y$	$m$	Grid size in $y$ -dirextion
$\xi_m^k$	$m/s$	Discrete $\xi$ , $\xi_m^k \approx \xi(x = m\Delta x, t = k\Delta t)$

### Stochastic extension of shallow water flow model

symbol	SI units	description
$d_w$	$m$	Spatial correlation length for $\Delta u$
$d_w$	$m$	Spatial correlation length for $\Delta \tau$
$P_c$	$m^2$	Computed covariance of water level residuals
$P_o$	$m^2$	Observed covariance of water level residuals
$\alpha_b$		Time correlation at open boundary
$\alpha_w$		Time correlation for wind noise
$\alpha_\tau$		Time correlation for noise on wind stress
$\Delta u$	$m/s$	System noise for current velocity $u$
$\Delta v$	$m/s$	System noise for current velocity $v$
$\Delta \xi_b$	$m$	Set-up at open boundary
$\Delta \tau_x$	$N/m^2$	System noise for wind stress in $x$ -direction
$\Delta \tau_y$	$N/m^2$	System noise for wind stress in $y$ -direction
$\varepsilon_b$	$m$	White noise driving AR(1) process at open boundary
$\varepsilon_{\Delta \tau_x}$	$N/m^2$	White noise driving AR(1) process for system noise on wind stress
$\varepsilon_{\Delta \tau_y}$	$N/m^2$	White noise driving AR(1) process for system noise on wind stress
$\sigma_b$	$m$	Standard deviation of noise in water level at the open boundary
$\sigma_m$	$m$	Standard deviation of measurement noise for water level measurements
$\sigma_w$	$m/s$	Standard deviation of wind noise per time-step
$\sigma_\tau$	$N/m^2$	Standard deviation of noise on wind stress per time-step

## Stochastic systems and Kalman filtering

symbol	SI units	description
$\mathbf{A}$		State to state transition matrix
$\tilde{\mathbf{A}}$		State to state transition for implicit system $\mathbf{A} = \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{A}}$
$\mathbf{B}$		Deterministic input to state transition matrix
$\tilde{\mathbf{B}}$		Deterministic input to state transition for implicit system $\mathbf{B} = \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{B}}$
$\mathbf{C}$		State to output transition matrix
$\mathcal{C}$		Controllability matrix
$\tilde{\mathbf{D}}$		Matrix for implicit system
$\mathbf{F}$		Noise input to state transition matrix
$\tilde{\mathbf{F}}$		Deterministic input to state transition for implicit system $\mathbf{F} = \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{F}}$
$\mathbf{f}$		Nonlinear state transition function
$\mathbf{g}$		Nonlinear output function
$\mathbf{K}$		Kalman gain matrix
$l$		Dimension of deterministic input $\mathbf{u}(k) \in \Re^l$
$m$		Dimension of noise input $\mathbf{w}(k) \in \Re^m$
$n$		Dimension of state vector $\mathbf{x}(k) \in \Re^n$
$\mathcal{O}$		Observability matrix
$\mathbf{P}$		Error covariance matrix
$\mathbf{P}_0$		Error covariance of initial estimate $\mathbf{x}_0$
$p$		Dimension of output $\mathbf{y}(k) \in \Re^p$
$\mathbf{u}$		Deterministic input
$\mathbf{u}_d$		Deterministic part of forcing
$\mathbf{u}_n$		Noise part of forcing
$\mathbf{v}$		Measurement noise vector
$\tilde{\mathbf{w}}(k)$		Combined vector containing $\mathbf{w}(0), \dots, \mathbf{w}(k)$
$\mathbf{w}$		System noise vector
$\mathbf{x}$		State vector
$\mathbf{x}_a$		Augmented state vector used for smoothing
$\mathbf{x}_0$		Initial estimate of state vector
$\mathbf{y}$		Output vector
$\hat{\mathbf{y}}(k)$		Combined vector containing $\mathbf{y}(0), \dots, \mathbf{y}(k)$
$\Sigma_o$		Covariance matrix of observational noise
$\Sigma_s$		Covariance matrix of system noise
$\Phi$		State transition for system with Kalman filter
$\alpha$		Correlation per time-step of an AR(1) process

## Square root filtering and suboptimal schemes

symbol	SI units	description
$\hat{\mathbf{A}}$		Approximation of $\mathbf{A}$
$\mathbf{A}_r$		State transition for reduced order model
$b$		Bandwidth parameter
$\mathbf{C}_r$		State to output map for reduced order model
$\mathbf{D}$		Diagonal matrix in decomposition of $\mathbf{L}\mathbf{L}'$ or $\mathbf{L}'\mathbf{L}$ or $A$
$\mathbf{E}_1$		Error matrix, difference optimal and computed covariance
$\mathbf{E}_2$		Error matrix, difference true and computed covariance
$e$		True RMS error
$e'$		Estimate of RMS error from $\mathbf{P}_c$
$\mathbf{F}_r$		Noise input to state map for reduced order model
$\mathbf{H}$		Matrix insquare root measurement update
$IMP$		Decrease of $RMS$
$\mathbf{K}_c$		Computed Kalman gain
$\mathbf{K}_c$		Computed Kalman gain on coarse grid
$\mathbf{L}$		Square root of error covariance matrix
$\mathbf{L}_c$		Computed approximation of $\mathbf{L}$
$\mathbf{L}_c^*$		Computed approximation of $\mathbf{L}$ after truncation
$LTL$		Inner products $\mathbf{L}'\mathbf{L}$
$\mathbf{l}_i$		$i$ 'th column of $\mathbf{L}$
$n_r$		Dimension of reduced order model
$\hat{\mathbf{P}}$		Controllability matrix for model reduction
$\mathbf{P}_c$		Computed (approximate) error covariance of state vector
$\mathbf{P}_c$		Computed error covariance on coarse grid
$\mathbf{P}_t$		True error covariance
$\mathbf{Q}$		Temporary matrix for computing decrease in $RMS$
$\hat{\mathbf{Q}}$		Observability matrix for model reduction
$q$		Rank of approximation of error covariance $\mathbf{P}$
$\mathbf{R}$		Weight matrix for criterion
$\mathbf{S}$		Synthetic sample of state vectors
$\mathbf{S}$		Interpolation operator
$\mathbf{S}^*$		Projection onto coarse grid
$\mathbf{T}$		Change in error covariance $\mathbf{P}\mathbf{P}_c$ due to truncation
$\mathbf{T}_b$		Truncation operator, sets elements further than $b$ from diagonal to 0
$\mathbf{U}$		Unitary matrix in eigen decomposition of $\mathbf{L}'\mathbf{L}$ or $\mathbf{L}\mathbf{L}'$ or in singular value decomposition of $A$
$\mathbf{V}$		Unitary matrix in singular value decomposition or eigen decomposition
$\mathbf{v}_i$		$i$ 'th realization of measurement noise process

symbol	SI units	description
$\mathbf{w}_i$		I'th realization of system noise process
$\mathbf{x}_r$		State vector of reduced order model
$\mathbf{x}_{tr}$		Transformed state vector
$\mathbf{y}_m$		Output of model
$\mathbf{y}_r$		Output of reduced order model
$\beta$		Variable in square root measurement update
$\delta \mathbf{x}$		Small change to $x$
$\kappa$		Matrix substituted for $\mathbf{K}$
$\xi_i$		I'th member of ensemble
$\Pi$		Projection on subspace
$\Pi_l$		Left factor of projection $\Pi$
$\Pi_r$		Right factor of projection $\Pi$
$\sigma_i$		I'th diagonal value of $\mathbf{D}$
$\Psi$		State transformation
$\Psi$		Matrix in time-step $\Psi := \mathbf{A}\mathbf{P}$

### Probability and statistics

symbol	SI units	description
$E$		Expectation operator
$f$		Probability density function
$J$		Criterion based on log likelihood
$RMS$		Root mean square error (or an estimate)
$X$		Random variable
$x$		Value of random variable $X$
$Y$		Random variable
$y$		Value of random variable $Y$
$\rho$		Correlation between two random variables
$\chi$		Constant in log likelihood function

### General

symbol	SI units	description
$\mathbf{E}_i$		Matrix with $[\mathbf{E}]_{i,i} = 1$ and 0 elsewhere
$f$		A function
$g$		A function
$\mathbf{I}$		Identity matrix
$I$		Index set
$i$		Index
$j$		Index
$M$		Increment
$N$		Upper value or lower value for index
$O(n)$		Order symbol
$\delta$		Small number
$\varepsilon$		Small number
$\gamma$		Small number
$\phi$		A positive number $\phi \in \Re$

### Physics general

symbol	SI units	description
$A$	$m$	Total amplitude harmonic/Fourier analysis
$a$	$m$	Amplitude of cosine in harmonic/Fourier analysis
$b$	$m$	Amplitude of sine in harmonic/Fourier analysis
$c$	$m/s$	Wave velocity
$L$	$m$	Length
$t$	$s$	Time
$\phi$	$rad$	Phase
$\omega$	$rad/s$	Angular frequency

# Bibliography

- [1] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, 1979.
- [2] A. Bagchi and P. ten Brummelhuis. Parameter identification in tidal models with uncertain boundaries. *Automatica*, 30(5):745–759, 1994.
- [3] A. F. Bennett. *Inverse Methods in Physical Oceanography*. Cambridge University Press, Cambridge, 1992.
- [4] A.F. Bennett. The kalman filter for a linear quasi-geostrophic model of ocean circulation. *Dyn. Atmos. Oceans*, 13:219–267, 1989.
- [5] G. J. Bierman. *Factorization Methods for Discrete Sequential Estimation*, volume 128 of *Mathematics in Science and Engineering*. Academic Press, Academic Press, New York, 1977.
- [6] D. Boggs, M. Ghil, and C. Keppenne. A stabilized sparse-matrix U-D square-root implementation of a large-state extended kalman filter. In *Second International Symposium on Assimilation of Observations in Meteorology and Oceanography*, pages 219–224. World Meteorological Organization, WMO, March 1995.
- [7] K. Bolding. Using a kalman filter in operational storm surge prediction. In *Second International Symposium on Assimilation of Observations in Meteorology and Oceanography*, pages 379–383. World Meteorological Organization, WMO, March 1995.
- [8] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis*. Prentice Hall, Englewood Cliffs, 3 edition, 1994.
- [9] W. P. Budgell. Nonlinear data assimilation for shallow water equations in branched channels. *J. Geophys. Res.*, 10:633–644, 1986.
- [10] W.P. Budgell. Stochastic filtering of shallow water wave processes. *SIAM J. Sci. Stat. Comput.*, 8(2):152–170, March 1987.
- [11] G. Burgers, P.J. van Leeuwen, and G. Evensen. On the analysis scheme in the ensemble kalman filter. *submitted to Monthly Weather Review*, 1996.

- [12] A.W. Cañizares, R. Heemink and H.J. Vested. Sequential data assimilation in a fully non-linear hydrodynamical model. In Andreas Muller, editor, *Hydroinformatics '96*, volume 2, pages 463–470. Balkema, Rotterdam, September 1996.
- [13] E.F. Carter, R.N. Miller, and S.T. Goodlett. Data assimilation into non-linear stochastic models. *www*, 1995.
- [14] G. Chavent. Identification of distributed parameter systems: About the output least square method, its implementation and identifiability. In *Identification and System Parameter Estimation*, pages 85–97. IFAC, 1979.
- [15] S. E. Cohn and R. Todling. Approximate data assimilation schemes for stable and unstable dynamics. *Journal of the Meteorological Society of Japan*, 74(1):63–75, 1995.
- [16] S. E. Cohn and R. Todling. Approximate kalman filters for unstable dynamics. In *Second International Symposium on Assimilation of Observations in Meteorology and Oceanography*, pages 241–246. World Meteorological Organization, WMO, March 1995.
- [17] S.E. Cohn. Observability of discretized partial differential equations. Technical report, 1986.
- [18] Cressie. *Statistics for Spatial Data*. Wiley, New York, 1991.
- [19] R. C. Curi, T. E. Unny, K. W. Hipel, and K Ponnambalam. Application of the distributed parameter filter to predict simulated tidal induced shallow water flow. *Stochastic Hydrology and Hydraulics*, 9:13–32, 1995.
- [20] R.F. Curtain and A.J. Pritchard. *Infinite dimensional linear systems theory*. Springer Verlag, 1978.
- [21] R. Daley. *Atmospheric data analysis*. Cambridge University Press, 1991.
- [22] R. Daley. *Data assimilation in meteorology and oceanography: Theory and practice*, chapter III, pages 209–219. Special issue of Journal of the Met. Soc. of Japan. University Academy Press, Tokyo, 1997.
- [23] D. Dee. Testing the perfect-model assumption in variational data assimilation. 1995.
- [24] D. P. Dee. Simplification of the kalman filter for meteorological data-assimilation. *Quarterly Journal of the Royal Meteorological Society*, 117(498):365–384, 1991.
- [25] D. P. Dee. A simple scheme for tuning forecast error covariance matrices. In *Variational Assimilation with Special Emphasis on 3-D Aspects*, pages 191–205, Reading, November 1992. ECMWF.

- [26] D. P. Dee, S. E. Cohn, A. Dalcher, and M. Ghil. An efficient algorithm for estimating noise covariances in distributed systems. *IEEE Transactions on Automatic Control*, AC-30(11):1057–1065, 1985.
- [27] D.P. Dee. On-line estimation of error covariance parameters for atmospheric data assimilation. *Monthly weather review*, 1995. to appear.
- [28] P. Van Dooren. A generalized eigenvalue approach for solving riccati equations. *SIAM J. Sci. Stat. Comput.*, 2(2):121–135, June 1981.
- [29] P. Van Dooren and M. Verhaegen. On the reliability and efficiency of different kalman filter algorithms. part I: Theoretical results. *IEEE Automatic Control*, 1986.
- [30] G. Evensen and P. J. van Leeuwen. Advanced data assimilation based on ensemble statistics. In *Second International Symposium on Assimilation of Observations in Meteorology and Oceanography*, pages 153–158. WMO, WMO, March 1995.
- [31] R. Fletcher. *Practical Methods of Optimization: Unconstrained Optimization*, volume 1. Wiley, Chichester, 1980.
- [32] I. Fukumori. Assimilation of TOPEX sea level measurements with a reduced-gravity shallow water model of the tropical pacific ocean. *J. Geophys. Res.*, 100:25,027–25,039, 1995.
- [33] I. Fukumori and P. Melanotte-Rizzoli. An approximate kalman filter for ocean data assimilation; an example with an idealized gulf stream model. *Journal of Geophysical Research*, 100:6777–6793, 1995.
- [34] H. Gerritsen and N. Praagman. On the calibration of tidal flow simulation models. nota gwao-89.012, Tidal Waters DIvision (DGW) Rijkswaterstaat, Den Haag, 1989.
- [35] M. Ghil, S. E. Cohn, J. Tavantzis, K. Bube, and E. Isaacson. *Application of Estimation Theory to Numerical Weather Prediction*. Springer Verlag, unknown. Dynamic Meteorology: Data Assimilation Methods 330p.
- [36] G. Godin. *The Analysis of Tides*. Liverpool University Press, 1972.
- [37] G. Golub and C. Van Loan. *Matrix Computations*. John Hopkins University Press, 2nd ed. edition, 1989.
- [38] E.J. Grimme, D.C. Sorensen, and P. van Dooren. Model reduction of state space systems via an implicitly restarted lanczos method. May 1994.
- [39] P.C. Hansen. Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank. *SIAM J. Sci. Stat. Comput.*, 11(3):503–518, may 1990.

- [40] A. W. Heemink. *Storm Surge Prediction Using Kalman Filtering*. PhD thesis, Twente University of Technology, 1986.
- [41] A. W. Heemink. Two-dimensional shallow water flow identification. *Appl. Math. Modelling*, 12:109–118, April 1988.
- [42] A. W. Heemink and H. Kloosterhuis. Data assimilation for non-linear tidal models. *International Journal for Numerical Methods in Fluids*, 11:1097–1112, 1990.
- [43] R. Henriksen. A correction of a common error in truncated second order non-linear filters. *Modeling, Identification and Control*, 1(3):187–193, 1980.
- [44] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, New York, 1970.
- [45] S. Julier and J. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. *unknown*, 1995. submitted, <http://phoebe.robots.ox.ac.uk/default.html>.
- [46] T. Kailath. Some new algorithms for recursive estimation in constant linear systems. *IEEE transactions on information theory*, IT-19(6):750–760, November 1973.
- [47] R. E. Kalman. A new approach to linear filter and prediction theory. *J. Basic. Engr.*, 82D:35–45, 1960.
- [48] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *J. Basic. Engr.*, 83D:95–108, 1961.
- [49] H.J. Kushner. Approximations to optimal nonlinear filters. *IEEE Trans. Automatic Control*, AC-12(5):546–556, oct 1967.
- [50] H.J. Kushner. Dynamical equations for optimum nonlinear filtering. *J. Differential equations*, 3:179–190, april 1967.
- [51] H.J. Kushner. Nonlinear filtering: the exact dynamical equations satisfied by the conditional mode. *IEEE Trans. Automatic Control*, AC-12(3):362–367, june 1967.
- [52] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. Wiley-Interscience, New York, 1972.
- [53] .W.M. Lander, P.A. Blokland, and J.M. de Kok. The three-dimensional shallow water model triwaq with a flexible vertical grid definition. SIMONA report SIMONA 96-01, RIKZ OS-96.104, RIKZ, January 1994.
- [54] P. S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 141-1 of *Mathematics in Science and Engineering*. Academic Press, New York, 1979.

- [55] P. S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 141-2 of *Mathematics in Science and Engineering*. Academic Press, New York, 1979.
- [56] R.N. Miller, M. Ghill, and F. Ghautiez. Advanced data assimilation in strongly nonlinear dynamical systems. *J. of the Atmos. Sc.*, 51(8):1037–1055, apr 1994.
- [57] Ministry of Transport and Public Works, The Netherlands. *User’s guide WAQUA*, version 8.21 edition, jan 1996.
- [58] M. Morf, B. Lévy, and T. Kailath. Square-root algorithms for the continuou-time linear least-square estimation problem. *IEEE Transactions on Automatic Control*, AC-23(5):907–911, October 1978.
- [59] M. Morf, G. S. Sidhu, and T. Kailath. Some new algorithms for recursive estimation in constant, linear, discrete-time systems. *IEEE Transactions on Automatic Control*, AC-19(4):315–323, August 1974.
- [60] Y. Oshman and I. Y. Bar-Itzhack. Square root filtering via covariance and information eigenfactors. *Automatica*, 22(5):599–604, 1986.
- [61] P. G. Park and T. Kailath. New square-root smoothing algorithms. *IEEE Transactions on Automatic Control*, 1994.
- [62] D. F. Parrish and S. E. Cohn. A kalman filter for a two-dimensional shallow-water model: Formulation and preliminary experiments. office note 304, New York University, February 1985.
- [63] W. Rodi. *Turbulence Models and their Application in Hydraulics*. IAHR, at Delft Hydraulics, 1984.
- [64] Y. Sasaki. Some basic formalisms in numerical variational analysis. *Monthly Weather Review*, 98(12):875–883, dec 1970.
- [65] A.J. Souza and I.D. James. A two-dimensional (x-z) model of tidal straining in the rhine rofi. *Continental Shelf Research*, 16(7):949–966, January 1996.
- [66] A.J. Souza and J.H. Simpson. The modification of tideellipses by stratification in the rhine rofi. *Continental Shelf Research*, 16(8):997–1007, January 1996.
- [67] G. S. Stelling. *On the Construction of Computational Methods for Shallow Water Flow Problems*. PhD thesis, Delft University of Technology, 1984. Rijkswaterstaat Communications no.35.
- [68] A. Tarantola. *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*. Elsevier, Amsterdam, 1987.

- [69] P. G. J. ten Brummelhuis and A. W. Heemink. Parameter identification in tidal models with uncertain boundary conditions. *Stochastic Hydrology and Hydraulics*, 4:193–208, 1990.
- [70] P.G.J. ten Brummelhuis. Kalman filtering in real time control systems. In A. Mueller, editor, *Hydroinformatics'96*, pages 461–462. Balkema, Rotterdam, September 1996.
- [71] P. J. van Leeuwen and G. Evensen. Data assimilation and inverse metod in terms of a probabilistic formulation. *submitted Monthly Weather Review*, 1995.
- [72] L.C. van Rijn. *Principles of Fluid Flow and Surface Waves in Rivers, Estuaries, Seas and Oceans*. Aqua Publications, Amsterdam, 1994.
- [73] G.K. Verboom, J.G. de Ronde, and R.P. van Dijk. A fine grid flow and storm surge model of the north sea. *Continental Shelf Research*, 12(23):213–233, 1992.
- [74] G.K. Verboom, G.S. Stelling, and M.J. Officier. Boundary conditions for the shallow water equations. tech-report, Delft Hydraulics, june 1980. also published in Computational Hydraulics.
- [75] M. Verhaegen and P. Van Dooren. On the reliability and efficiency of different kalman filter algorithms. part II: flight-path reconstruction, a case study. *IEEE Automatic Control*, 1986.
- [76] M. Verlaan and A. W. Heemink. Reduced rank square root filters for large scale data assimilation problems. In *Second International Symposium on Assimilation of Observations in Meteorology and Oceanography*, pages 247–252. World Meteorological Organization, WMO, March 1995.
- [77] M. Verlaan and A. W. Heemink. Tidal flow forecasting using reduced rank square root filters. Technical Report 95-75, Delft University of Technology, ftp: //ftp.twi.tudelft.nl /TWI/publications/tech-reports/1995, 1995. accepted Stochastic Hydrology and Hydraulics.
- [78] X.F. Zhang. *Data Assimilation in Air Pollution Modelling*. PhD thesis, Delft University of Technology, nov 1996.
- [79] X.F. Zhang and A.W. Heemink. The application of kalman filter in the evaluation of the  $ch_4$  budget in europe. In *Air Pollution IV, monitoring, simulation and control*, pages 155–164. Wessex Institute of Technology, August 1999.

# Samenvatting

Als het Kalman filter rechtstreeks zou worden toegepast voor grootschalige data-model integratie vraagstukken dan zouden er een aantal typen problemen ontstaan. Ten eerste neemt voor grote modellen de rekentijden van het Kalman filter explosief toe. Tegelijk met deze rekentijd wordt ook de benodigde geheugencapaciteit enorm. Ten derde is de conditionering van het standaard Kalman filter voor dit type problemen vaak erg slecht hetgeen het gebruik van een numeriek robuuster algoritme noodzakelijk maakt. Om bovenstaande problematiek het hoofd te kunnen bieden zijn er recentelijk een aantal efficiënte benaderende algoritmes ontwikkeld, die bekend staan onder de naam 'suboptimal schemes' (afgekort S.O.S.).

Het hoofdonderwerp van dit werk is een nieuw 'suboptimal scheme'. Dit 'reduced rank square root filter' (RRSQRT) is vele malen sneller dan het oorspronkelijke Kalman filter wanneer het wordt toegepast voor grootschalige numerieke modellen. De werking van dit algoritme berust op een benadering van de covariantiematrix van de fouten door een matrix met lagere rang. Door gebruik te maken van een 'square root' factorisatie kan deze eigenschap worden benut om de rekentijd te bekorten en het geheugengebruik terug te dringen. Bovendien garandeert deze factorisatie dat de covariantiematrices ten allen tijde positief semi-definiet blijven en is tegelijkertijd de conditionering van deze factoren veel gunstiger dan die van de covarianties wat de numerieke stabiliteit ten goede komt. Een aantal eigenschappen van het RRSQRT wordt wiskundig bewezen en geïllustreerd aan de hand van een eenvoudig model.

Vervolgens worden er twee uitbreidingen van het RRSQRT algoritme voorgesteld voor toepassingen met niet lineaire modellen. Het betreft een eerste en een tweede orde nauwkeurige benadering. Een voordeel in vergelijk met andere algoritmes die gebruik maken van formele afgeleiden, zoals een lokaal gelineariseerd model, is dat er geen code met deze afgeleiden van de modelvergelijkingen hoeft te worden bepaald. In plaats daarvan wordt het model meerdere malen aangeroepen. Dit vereenvoudigt de implementatie van de methode aanzienlijk.

Als het RRSQRT algoritme voor een n-dimensionaal lineair model wordt vergeleken met een aantal van de bekendste SOS's dan blijkt het redelijk efficiënt te zijn. Voor lineaire tijdinvariante modellen met metingen op vaste locaties en regelmatige intervallen is het 'steady state filter' nog veel efficiënter en voor

sterk niet-lineaire modellen kan het 'ensemble Kalman filter' nauwkeuriger zijn. Voor een groot aantal toepassingen dat zich tussen deze uitersten bevindt is het RRSQRT algoritme echter een aantrekkelijk alternatief.

In het tweede deel van dit werk wordt het RRSQRT algoritme toegepast op een drietal data-assimilatieproblemen. De eerste toepassing betreft een tweedimensionaal waterbewegingsmodel voor de voorspelling van stormvloeden. Om een zo nauwkeurig mogelijke voorspelling te maken wordt een aantal waterstandsmetingen gebruikt om onzekerheden een in de begintoestand te verkleinen. Een vergelijking van het RRSQRT algoritme met het steady-state filter dat op dit moment operationeel wordt gebruikt laat zien dat het voorgestelde algoritme voor deze toepassing goed werkt. Hoewel het model enkele niet-lineaire termen bevat lijken deze van ondergeschikt belang, zodat de mogelijke winst van een tijdafhankelijk filter niet opweegt tegen de hogere rekenkosten. Wel zou in de toekomst de extra vrijheid van een tijdafhankelijk filter kunnen worden benut om een nauwkeuriger model van de onzekerheden in de windschuifspanning te construeren.

In het huidige operationele systeem voor de voorspelling van stormvloeden worden de waterstanden van 8 locaties geassimileerd. Met behulp van een aantal experimenten is bestudeerd of de nauwkeurigheid verder zou kunnen worden opgevoerd door het toevoegen van extra meetstations en tevens wat het effect zou zijn van het verwijderen van een van de bestaande stations. De invloed van deze ingrepen is bestudeerd uitgaande van de nauwkeurigheid (covariantie van de fouten) die door het Kalman filter wordt berekend. Met behulp van deze methode kunnen fictieve situaties worden doorgerekend waardoor het mogelijk is een geschikte locatie te selecteren voordat de meetapparatuur wordt geplaatst. De experimenten laten zien dat het toevoegen van meetlocaties in de zuidelijke Noordzee voor deze toepassing waarschijnlijk een geringe meerwaarde heeft. Dit in tegenstelling tot locaties ten westen van Ierland en Schotland waar extra metingen naar alle waarschijnlijkheid een veel grotere invloed op de (langere termijn) voorspellingen zullen hebben. Verder onderzoek is nodig om een optimale configuratie te kunnen bepalen.

In de tweede toepassing wordt een methode voorgesteld om waterstanden te reconstrueren. De methode is gebaseerd op een RRSQRT versie van het 'fixed-lag smoother'. In deze aanpak worden eerst alleen de waterstanden op de open rand berekend en wordt vervolgens een gewone modelsimulatie gebruikt om de waterstanden op de gewenste posities te berekenen. De uitgevoerde experimenten laten zien dat de methode langs de kust erg nauwkeurig is. Omdat er geen meetgegevens verder van de kust beschikbaar waren voor de periode van het experiment kan de nauwkeurigheid elders niet direct worden bepaald. Wel is op een indirect manier getoond dat de nauwkeurigheid in het gehele modeldomein waarschijnlijk van dezelfde orde van grootte is. Als de 'smoother' wordt vergeleken met het direct gebruik van het RRSQRT filter dan blijkt de nauwkeurigheid langs de kust ongeveer even groot. Verder van de kust echter is de nauwkeurigheid van de 'smoother' groter.

De derde toepassing betreft het assimileren van radarmetingen van oppervlaktesnelheden in een driedimensionaal waterbewegingsmodel. Door de in-

gewikkeld structuur van de stromingen rond de monding van de Nieuwe Waterweg is het zelfs met driedimensionale modellen moeilijk deze patronen te voorspellen. Een voorstudie laat zien een driedimensionaal ondiepwatermodel de stromingen in een gestratificeerd gebied onder gemiddelde omstandigheden waarschijnlijk redelijk kan reproduceren. In extreme situaties met veel wind en/of extreem hoge danwel lage afvoer van de rivieren kunnen de voorspellingen waarschijnlijk aanzienlijk worden verbeterd door het assimileren van gemeten oppervlaktestromingen. Bovendien verwijnt met het gebruik data assimilatie de noodzaak voor het gebruik van een omvattend model voor het berekenen van de opzet ter plaatse van de open rand. De gebruikte data assimilatiemethode is wederom een extended Kalman filter versie van het RRSQRT filter. In de uitgevoerde experimenten zijn zowel radarmetingen van de oppervlaktesnelheden als waterstanden geassimileerd. Dit leidt tot duidelijk nauwkeuriger profielen, zoals blijkt uit een vergelijking met in-situ metingen. Door aanpassing van de systeemruis kunnen deze resultaten mogelijk nog verder worden verbeterd.

# Dankwoord

De afgelopen jaren, de periode waarin dit proefschrift tot stand is gekomen, zijn voor mij een hele plezierige en leerzame tijd geweest. De prettige omgeving (lees collega's) op zowel de Technische Universiteit Delft als bij het RIKZ (Rijksinstituut voor Kust en Zee) hebben hier zeker een belangrijke bijdrage aan geleverd. Hoewel het aantal mensen dat een inhoudelijke en/of morele bijdrage heeft geleverd aan dit werk te groot is om ze hier allemaal te vermelden wil ik er toch een paar noemen.

Dit proefschrift was nooit tot stand gekomen zonder mijn promotor Arnold Heemink. De discussies met hem waren een constante bron van inspiratie voor het onderzoek en bron van motivatie voor mij.

Een groot deel van het onderzoek is tot stand gekomen in nauwe samenwerking met het RIKZ. Naast een financiële bijdrage en het beschikbaar stellen van een aanzienlijke ondersteuning en een flinke hoeveelheid rekencapaciteit hebben ook de discussies met een groot aantal collega's dit werk gevormd tot wat het nu is. In het bijzonder wil ik Krystyna Robaczewska bedanken voor het inzicht dat onderzoek geen problemen kent, maar slechts kansen en uitdagingen.

In November '96 heb ik het geluk gehad een aantal weken met Jan Leenderste te mogen samenwerken. Zijn enthousiasme en oog voor nieuwe mogelijkheden hebben mij een flinke duw in de rug gegeven voor de 3-D toepassing.

In het bijzonder wil ik ook Regien Brouwer bedanken voor de hulp met alle zaken rond alle implementatie en computer gebruik. Vaak waren haar vragen de aanleiding voor verder onderzoek. Ook wees ze altijd op de mogelijkheid van een implementatiefout als ik al een halve theorie had bedacht voor een opmerkelijk resultaat van een experiment. Hiervoor bedankt.

In de afgelopen is bij mij het respect voor modelleurs en fysici gestaag gegroeid. Begonnen met het idee dat een zo wiskundig mogelijke aanpak de beste was is steeds meer het besef ontstaan dat een uitgebreide ervaring en kennis van de fysische toepassing een onmisbare basis vormt voor elke toepassing.

Ten slotte wil ik ook mijn vriendin, ouders, familie en vrienden bedanken voor de steun. Ongetwijfeld viel het niet mee het gezwam over Kalman filters en partiële differentiaalvergelijkingen aan te horen. Toch is jullie bijdrage misschien wel de belangrijkste, want gewoonweg volhouden is misschien wel het moeilijkste onderdeel van een promotieonderzoek.

Ongetwijfeld ben ik nu een heleboel mensen vergeten.

# Curriculum Vitae

Martin Verlaan werd geboren op 22 december 1969 te Gouda. In 1988 slaagde hij voor het atheneum examen aan het Marcuscollege te Grootebroek. Aansluitend studeerde hij Toegepaste Wiskunde aan de Universiteit Twente. In 1993 studeerde hij met lof af in de richting Systeem- en Regeltechniek onder leiding van Prof. Kwakernaak. Het afstudeerwerk werd uitgevoerd bij de Technische Universiteit van Luleå in noord Zweden. Onderwerp van het afstudeerwerk was een algoritme voor de classificatie van bacteriën.

Van 3 september 1993 tot 31 juni 1997 was hij als Assistent In Opleiding verbonden aan de faculteit Technische Wiskunde en Informatica van de Technische Universiteit Delft. Tijdens deze periode is dit proefschrift tot stand gekomen. Bij de subgroep Wiskundige Analyse van Grootschalige Systemen werd hij begleid door Prof. Heemink. Het onderzoek is uitgevoerd in nauwe samenwerking met het Rijksinstituut voor Kust en Zee van Rijkswaterstaat.

Sinds 1 juli 1997 is werkt hij bij het RIKZ als projectleider Hydrodynamica en Data-assimilatie en is hij in deeltijd verbonden aan de faculteit Informatietechnologie en Systemen van de Technische Universiteit Delft, waar hij onderzoek verricht naar diverse aspecten van data-assimilatie.