KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

DATA ASSIMILATION IN MAGNETOHYDRODYNAMICS SYSTEMS USING KALMAN FILTERING

Jury:
Prof. dr. ir. E. Aernoudt, voorzitter
Prof. dr. ir. B. L. R. De Moor, promotor
Prof. dr. ir. J. Vandewalle (KUL)
Prof. dr. ir. P. Van Dooren (UCL)
Prof. dr. ir. D. Roose (KUL)
Dr. ir. P. Van Overschee (IPCOS)
Prof. dr. S. Poedts (KUL)
Prof. dr. J. De Schutter (KUL)

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de toegepaste wetenschappen

door

Oscar Barrero Mendoza

U.D.C. 537.84

May 2005

©Katholieke Universiteit Leuven - Faculteit Toegepaste Wetenschappen
Arenbergkasteel, B-3001 Heverlee (Belgium)
Alle rechten voorbehouden. Niets uit deze uitgave mag vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.
All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.
D/2005/7515/48
ISBN 90-5682-615-8

Acknowledgements

I want to express my sincere gratitude to Prof. Bart De Moor for having given me the opportunity to come at the Katholieke Universiteit Leuven as a Ph.D student of the SISTA group, as well as Prof. Joos Vandewalle for his unconditional support not just for me but for all the Colombian community at the KUL. Also, I have to thank Prof. Bart de Moor for his guidance and support in the most difficult moments of this long path, and for helping me to mature as researcher.

Despite of having been just in the last stage of my doctorate, Prof. Dennis Bernstein from the University of Michigan played a very important role in my research. The two times I visited the department of Aerospace Engineering at the University of Michigan in 2004, and the continuous contact we had via e-mail, helped me to obtain the final results of my thesis. Additionally to that, I have to say that working together with Prof. Bernstein and his team has been a very enriching experience for me. Therefore, I am very thankful and I appreciate very much all the help and guidance that Prof. Bernstein gave me in these last years.

On the other hand, thanks to all people I met in ESAT during these years for letting me be part of their lives, it has been a pleasure to know you and share with you professional as well as personal experiences and thoughts. Also to Pela, Ilse, Ida, Veerle, and Bart for their support and friendship.

Apart from my studies, I have known many kind people who helped me to make the things easier during my stay in Leuven, to all of them thanks a lot. Last but not least, thanks to all my family that in one way or another always were there supporting me, and my loved Maribel and Alejandro whose company, support, understanding, care, and love were my best motivation to finish up this work.

In dedication to my loved mother, Aurora.

Abstract

This thesis explores the use of sequential data-assimilation techniques in space weather forecast whose dynamics are described by magnetohydrodynamics --MHD- systems. We focus on the problem of finding estimators for large scale systems with sparse measurement points. A broad review of different Kalman filter formulations and algorithms ranging from the linear up to non-linear ensemble-based Kalman filters is done. It turns out that the Kalman filter cannot be directly applied to large scale systems; therefore, we explore also suboptimal Kalman filter formulations. We introduce two new algorithms for large scale linear systems, namely, the singular square root Kalman filter (SSQRT-KF), and the spatially localized Kalman filter (SLKF). The first one is a very efficient algorithm for systems where the process noise can be assumed to be negligible. The second one addresses the problem of observability when there are sparse observations in large scale systems. It turns out that the best estimation you can have in large scale systems with a small number of measurements, is a local estimation. Hence, we show that the SLKF is an optimal solution for this case. Finally, we take as case study a 2D solar storm coming towards the Earth's magnetosphere. For this case study, we test and compare the performance of ensemble-based Kalman filters. The results show that sequential data-assimilation techniques are very promising in this field.

Contents

Acknowledgements i

Abstract iii

Contents v

1 General Introduction 1

1.1 Introduction and Motivation 1

1.2 Chapter by Chapter Description 8

1.3 Main Contributions 11

1.4 Published Results 11

2 Linear Discrete-Time Kalman Filtering 15

2.1 Introduction 15

2.2 The Kalman Filter 16

2.3 Numerical Formulation 18

2.4 Square Root Formulation 26

2.4.1 Introduction 26

2.4.2 Error Covariance Update using Simultaneous Processing 27

2.4.3 Error Covariance Update using Sequential Processing . 32

2.4.4 Error Covariance Update using a Matrix Transformation 35

2.5 Computational Complexity of the Algorithm 38

2.6 Conclusions 39

3 Suboptimal Kalman Filters for Linear Large Scale Systems 41

3.1 Introduction 41

3.2 Reduced Rank Square Root Kalman Filter - RRSQRT-KF . . . 43

3.3 Singular Square Root Kalman Filter SSQRT-KF 44

3.3.1 SSQRT-KF with Covariance Inflation 46

3.3.2 Heat Transfer System Example 47

3.4 Spatially Localized Kalman Filter - SLKF 61

3.4.1 Numerical Formulation 62

3.4.2 Square Root Formulation of the SLKF 66

3.4.3 Mass-Spring System Example 70

3.5 The Ensemble Kalman Filter - EnKF 77

3.5.1 Representation of Error Statistics 77

3.5.2 Prediction of Error Statistics 79

3.5.3 The Estimation Problem 80

3.6 Computational Complexity of the Algorithms 82

3.7 Conclusions 83

4 Ensemble-Based Kalman Filter for Nonlinear Large Scale Systems 85

4.1 Introduction 85

4.2 Stochastic Model Description 86

4.3 The Extended Kalman Filter –EKF 86
4.4 The Ensemble Kalman Filter –EnKF 90
4.4.1 Error Covariance Propagation 91
4.4.2 Covariance Localization 94
4.5 Square Root Ensemble Filter –EnSRF 98
4.6 Ensemble Transform Kalman Filter –ETKF 103
4.7 The Ensemble Spatially Localized Kalman Filter –EnSLKF . . 106
4.8 Computational Complexity of the Algorithms 108
4.9 Conclusions 108

5 Space Weather Forecast Case Study **111**
5.1 Introduction 111
5.2 Problem Motivation 112
5.3 Magnetohydrodynamics Model Description 118
5.3.1 Linear 2D-MHD Flow System 120
5.3.2 Description of the Nonlinear MHD Model Simulator . . 130
5.4 Data Assimilation for Space Weather Forecast 131
5.4.1 Introduction 131
5.4.2 Data Assimilation in a Linear 2D-MHD Flow System . 132
5.4.3 Data Assimilation in a Non-Linear 2D-MHD Flow System 136
5.5 Conclusions 155

6 Conclusions and Future Research **157**
6.1 Conclusions 157
6.2 Future Research 159

Bibliography **161**

Appendices **169**
A List of Abbreviations 169
B List of Symbols MHD System 169
C Random Fields 170
D Generation of Random Fields for Ensemble-Based Filters . . . 171
E MHD Data Assimilation Figures 173

Chapter 1

General Introduction

1.1 Introduction and Motivation

The region between the Sun and the planets has been termed the interplanetary medium. Although once considered a perfect vacuum, this is actually a turbulent region dominated by the solar wind, which flows at velocities of approximately 250-1000 km/s (about 9×10^5 to 36×10^5 km/h), Figure 1.1. One of the events which provoke the solar wind is the coronal mass ejection (CME). The CMEs are one of the most energetic and important solar phenomena. These events propel magnetic clouds with a mass up to 10^{17}g (g, gravitational field strength [N/kg]) to speeds up to 2600 km/s into the heliosphere, influencing near-Earth plasma conditions (space weather), causing fluctuations in the terrestrial magnetic field and in the ionospheric density, and driving auroras (magnetic solar storms). These storms can cause some dangerous effects on Earth's systems like: communication, early-warning, submarine detection, electric power, satellites, pipelines, and many others. These dangerous effects may also affect geologic explorations, climate, and even generate radiation hazards to humans. Therefore, the subtleties of the interactions between Sun and Earth, and between solar particles and delicate instruments, have become factors that affect our well being. Thus there will be continued and intensified need for space environment services to address health, safety, and commercial needs[1].

[1] http://www.sec.noaa.gov/primer/primer.html

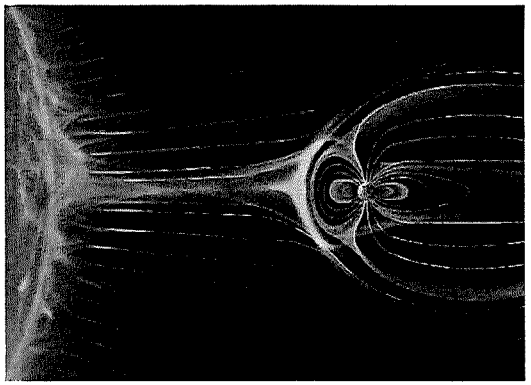


Figure 1.1: The Sun's powerful magnetic forces directly affect the Earth and the rest of the solar system. Picture taken from http://clusterlaunch.esa.int.

The aim of this thesis is to introduce the topic of data assimilation in the area of space weather forecast. Data assimilation is the technique whereby observational data are combined with outputs from a numerical model to produce an optimal estimate of the evolving state of the system. Note that this definition is similar to 'state space observer' in the systems theory literature [38], Figure 1.2. Thus, by assimilating space environment data into first-principles based global numerical models, we can improve our understanding of the physics of the geospace environment and improve the forecasting of its behavior.

There exist mainly two approaches for data assimilation, the first one, variational techniques [75], that consist of minimizing a cost function at each sampling time, quantifying the difference between model outputs and measurements, and assuming the system and measurement uncertainties as deterministic processes, equations (1.1). And the second one, sequential techniques that are based on the Kalman filter [4], that consist of minimizing a cost function recursively, quantifying the difference between the model outputs and measurements, and assuming the system and measurement uncertainties as stochastic processes, equations (1.2). Equations (1.1) and (1.2) are the state space representation of a deterministic and stochastic discrete-time dynamical system, respectively. Hence,

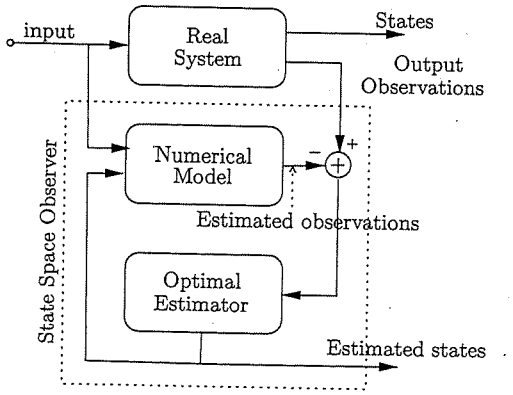


Figure 1.2: Data Assimilation System Scheme. The observations of the real system are assimilated into the numerical model to obtain a better estimation.

$$\begin{aligned} x_{k+1} &= f(x_k, \mathrm{p}_k) + B(u_k, \mathrm{p}_k), \\ y_k &= C(x_k, \mathrm{p}_k). \end{aligned} \tag{1.1}$$

$$\begin{aligned} x_{k+1} &= f(x_k) + B(u_k) + w_k, \\ y_k &= C(x_k) + v_k, \end{aligned} \tag{1.2}$$

where f represents the state equation vector function, C the output equation vector function, and B the input equation vector function. On the other hand, $x_k \in \mathbb{R}^n$ is the state space vector, $u_k \in \mathbb{R}^p$ the input vector to the system, $y_k \in \mathbb{R}^m$ the output vector of the system, $\mathrm{p} \in \mathbb{R}^e$ the uncertain parameters of the system, $w_k \in \mathbb{R}^l$ the stochastic system noise process, $v_k \in \mathbb{R}^m$ the stochastic measurement noise process, and k the time instant.

It turns out that variational data assimilation techniques are very expensive computationally, because one has to solve an optimization problem each data assimilation time step[2] to find the best p_k which minimizes the estimation error, contrary to sequential techniques that solve the optimization problem

[2]A data assimilation time step is defined as the interval of time in which the observations are taken and assimilated into the numerical model.

recursively, assuming that the system and measurement noise processes are known. Because of this, sequential data assimilation techniques using the Kalman filter have gained a lot of popularity, and have been used successfully by meteorologists and oceanographers in the last decade (some references [8, 14, 21, 28, 32, 34, 53, 58]).

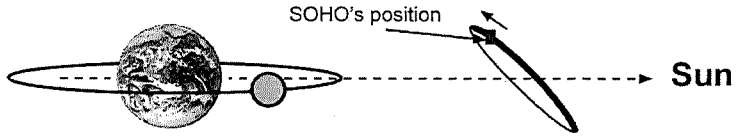


Figure 1.3: Orbit of the SOHO satellite. Picture taken from http://soho.esa.int.

On the other hand, the space physics community has been slow in implementing such techniques. In fact, it is difficult to find either references or literature about it. In [74] we can find an application for global assimilation of ionospheric measurements (GAIM), for instance. The reason is that the space systems are much more complex, larger, and there is a shortage of measurements; for example, the length scales of the Earth's weather go up to 10^4 km ($\mathcal{O}(10^6 - 10^7)$), while in the space weather can go up to 10^7 km ($\sim \mathcal{O}(10^{12})$). Moreover, there exist a massive amount of information for terrestrial weather, while for space weather exists only 5 satellites collecting information in the magnetosphere and between the Sun and Earth, see Figures 1.3, 1.4. As a result, in the last decades astrophysicists have been working to develop models and tools to understand better the space plasma dynamics [66, 43, 44]. However, there are still many things to discover about these systems and not much information available. This is why researchers in this area often use recorded data, which have been taken from special events like big CMEs to tune their models and improve the simulations [66]. Besides, as mentioned above, there are only five satellites dedicated to study the space weather, namely, the SOHO[3] mission, that has located one satellite at the L1 orbit, which is a neutral gravity point between Sun and Earth, see Figure 1.3, and the Cluster II[4] mission, that has

[3]SOHO stands for Solar and Heliosphere Observatory. It has been designed to study the internal structure of the Sun, its extensive outer atmosphere and the origin of the solar wind, the stream of highly ionized gas that blows continuously outward through the Solar System.

[4]The Cluster II mission is an in-situ investigation of the Earth's magnetosphere using four identical spacecraft simultaneously, called: salsa, samba, tango, and rumba. It will permit

located four satellites in the Earth's magnetosphere, see Figure 1.4. However, this situation is changing rapidly for the ionosphere, for instance, and it is expected that it will also start changing for the magnetosphere in some decades. Therefore, it is time to start exploring data assimilation techniques for space weather forecast.

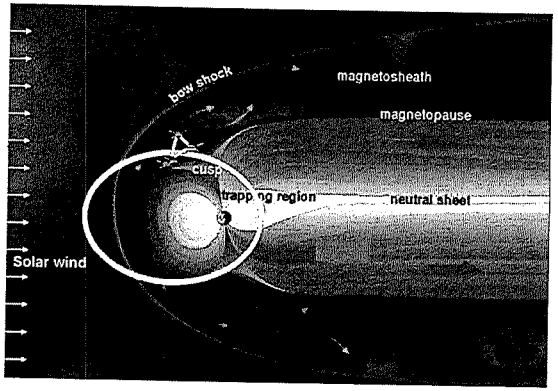


Figure 1.4: White oval, orbit of the Cluster satellites. Picture taken from http://clusterlaunch.esa.int.

One of the most common techniques used for space weather forecast nowadays, is to run a model simulation where the approximated initial conditions are given by observations taken by the satellite SOHO. It turns out that these predictions are not reliable at all, as expected, because the information used to run the model is very vague, and the models are not perfect. In other words, they make predictions based on an open loop simulation, Figure 1.5, which is not the best choice for this sort of problems. Hence, sequential data assimilation approaches offer a more robust framework to compute predictions of a dynamical system, when there exist uncertainties in the numerical model and observations. This is because it corrects the behavior of the numerical model dynamics according to current measurements of the real system. As a result, the predictions are made based on a closed loop simulation where the feedback correction is given by an optimal estimator –the Kalman filter in the linear case, Figure 1.2. We have

the accurate determination of three-dimensional and time-varying phenomena and will make it possible to distinguish between spatial and temporal variations.

to clarify at this point, that the motivation of this thesis is not to apply data assimilation to forecast the real space weather, because it is still complicated, due to computational constraints and a lack of measurement points, but to explore and study the performance of these techniques in large scale space plasma systems, whose dynamics can be described by magnetohydrodynamic systems (MHD) [44], with sparse measurement points.

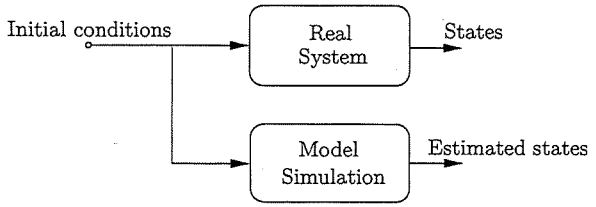


Figure 1.5: Open loop prediction scheme. After setting the initial conditions, the numerical model is run forward in time to obtain a prediction of the dynamics of the system.

The Kalman filter is certainly one of the greater discoveries in the history of statistical estimation theory [45]. Its most immediate applications have been for the control of complex dynamic systems such as continuous manufacturing processes, aircraft, ships, or spacecraft. To control a dynamic system, you must first know what it is doing. For these applications, it is not always possible or desirable to measure every variable that you want to control, and the Kalman filter provides a means for inferring the missing information from indirect (and noisy) measurements. The Kalman filter is also used for predicting the likely future courses of dynamic systems that people are not likely to control, such as the flow of rivers during flood, the trajectories of celestial bodies, or the prices of traded commodities.

The Kalman filter can be used basically either as estimator or predictor [4]. In sequential data assimilation the prediction is made by using a short-range forecast strategy, which consist of estimating the initial conditions at time k, when the observations are assimilated, and then, the prediction is done by running a model simulation forward in time until new observations are taken. see Figure 1.2. Hence, in sequential data assimilation the Kalman filter used is the estimator, §2.3. The Kalman filter estimator is computed in two steps, namely, the time update, and the measurement update, as depicted in Figure 1.6. There can be seen how the Kalman filter reduces recursively the error

estimation at time k, based on measurements taken at the same time instant k, and inputs and system noise at $k-1$. For more details about the Kalman filter estimator we refer to §2.3.

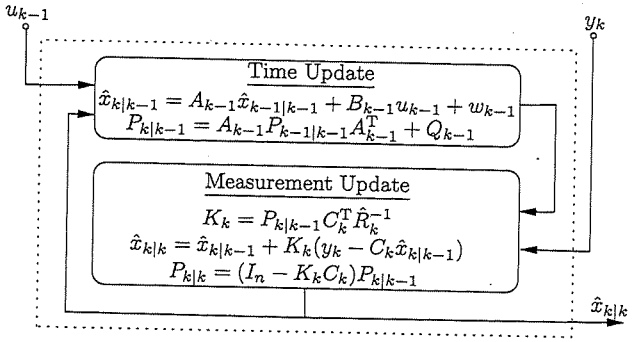


Figure 1.6: The Kalman filter seen as a feedback estimator. The Kalman filter reduces recursively the error estimation at time k, based on measurements taken at the same time instant k, and inputs and system noise at $k-1$. Details about the equations can be found in §2.3.

Up to now, everything sounds good, but if we look at the Kalman filter equations, and we think about the size of the systems we are dealing with, around $10^7 km \sim \mathcal{O}(10^{12})$, we realize that it would be difficult to use this set of equations directly to our problem. Besides, the Kalman filter is optimal for linear systems, and that is not the case of MHD systems. Therefore we have two big issues to solve: first, how to apply Kalman filter to large scale systems, and second, how we can use the Kalman filter for nonlinear systems. Hence, we focus our research in these two directions; first, we make a review and investigate Kalman filter algorithms that can be used in large scale systems, and second, we extend these results to the nonlinear case using ensemble-based Kalman filters, which turn out to be very efficient and easy to implement, as it has been shown in several publications in the weather forecast community [8, 14, 21, 28, 32, 34, 53, 58].

Now, a more detailed description of the contents of the thesis is given in the next section.

1.2 Chapter by Chapter Description

Chapter 2 - Linear Discrete-Time Kalman Filtering

Chapter 2 is dedicated to introducing the reader to the Kalman filter and its different numerical formulations and algorithms. We make a review and study of some of the more relevant algorithms found in the literature, namely, the classical Kalman filter (KF), the square root filter Kalman filter (SQRT-KF), and the transform Kalman filter (TKF). Additionally, a comparison of the numerical complexity of these algorithms in number of flops is done at the end of the chapter.

Chapter 3 - Suboptimal Kalman Filters for Linear Large Scale Systems

Although the Kalman filter has been shown to provide an optimal solution for data assimilation in linear systems with a relatively small order, for very large nonlinear systems it is not directly applicable, because of its computational complexity and the size of the error covariance matrix that has to be stored. Hence, this chapter presents an overview of different suboptimal linear Kalman filter formulations that can handle large scale systems. On the one hand, the reduced rank square root Kalman filter (RRSQRT-KF), and the ensemble Kalman filter (EnKF) which are two well known suboptimal Kalman filters. On the other hand, we propose two new algorithms, namely, the singular square root Kalman filter SSQRT-KF that is developed to be used for systems with negligible process noise, and the spatially localized Kalman filter (SLKF), which is designed to be used when the measurements are sparse.

The RRSQRT-KF, the SSQRT-KF, and the KF are tested and compared in a 2D-plate heat transfer system, where the objective is to compare the performance of the filters when the propagated error covariance matrix is approximated according to the filter used. In addition, the problem of boundary condition estimation is treated, and a solution is given by inverting the linear dynamic system.

The SLKF is tested and compared to the KF in a mass-spring system, where the objective is to study the performance of the filter for local estimation under different error weight functions.

Finally, the numerical algorithms of the filters are shown in detail, and a com-

parison of the numerical complexity is done.

Chapter 4 - Ensemble-Based Kalman Filter for Nonlinear Large Scale Systems

In this chapter, firstly a nonlinear stochastic model description is defined for environmental systems. Then an extension to the nonlinear case of the filters proposed in Chapter 3 is done, namely, the extended Kalman filter (EKF), the ensemble Kalman filter (EnKF), the square root ensemble filter (EnSRF), the ensemble transform Kalman filter (ETKF), and the ensemble spatially localized Kalman filter (EnSLKF). Finally, the discussion focuses on the advantages and disadvantages of each filter including the complexity of the algorithms.

Chapter 5 - Space Weather Forecast Case Study

This chapter is dedicated to exploring the application of sequential data assimilation techniques to space weather forecast. Notice that, in contrast to the weather forecast data assimilation problem, the number of measurement points is small compared to the order of the system. We start with a problem motivation, then a description of magnetohydrodynamics systems (MHD), which are used to simulate the space plasma dynamics, is given. Finally we apply the data assimilation techniques introduced in the previous chapters to MHD systems.

Mainly, the discussion focuses on two cases, first the linear case which is important in the space community to study the behavior of the space plasma in steady state, and second, the nonlinear case where the aim is to investigate the performance of ensemble-based Kalman filters in MHD systems for space weather prediction.

Hence, in the linear case a continuous-time 2D-linear model is obtained by taking small perturbations around steady state conditions. Then, two spatial discretization techniques are utilized in order to yield a state-space representation, namely, the Fourier collocation and the Chebyshev collocation method. Consequently, an analysis of performance of the suboptimal linear KFs introduced in Chapter 3 is carried out. For the nonlinear case a 2D-MHD system emulating the magnetosphere around the Earth is simulated using a numerical model written in Fortran (the VAC code [79]). Then, by means of a modular scheme where Matlab and VAC are used as simulator and estimator, respectively, we apply the ensemble-based filters introduced in chapter 4 to this system. Finally, the discussion focuses on the comparison of performance of the filters.

Chapter 6 - Conclusions and Future Research

In this chapter the main contributions are enumerated and some research perspectives are explored.

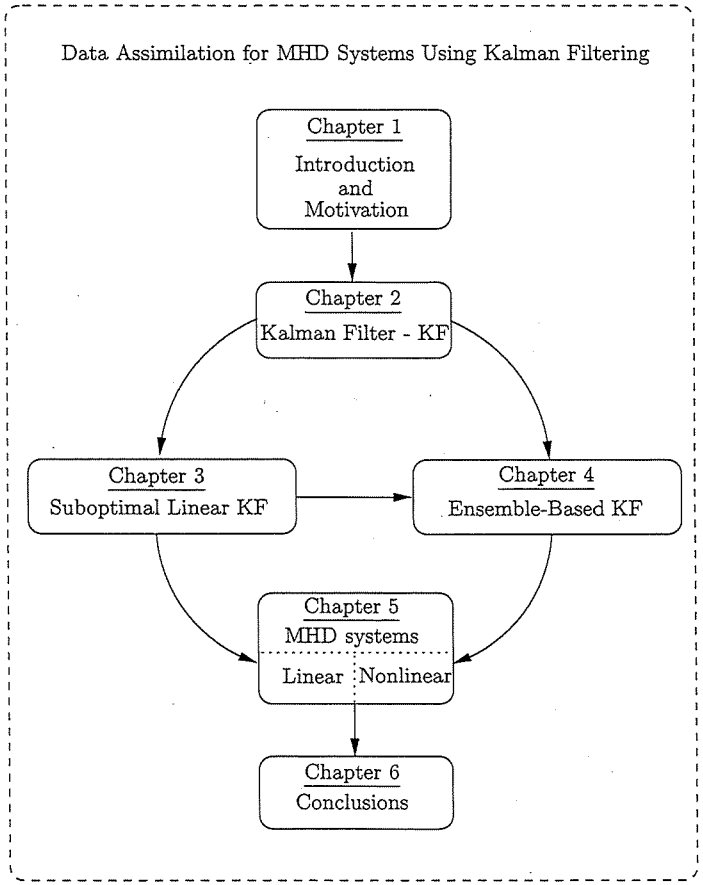


Figure 1.7: General Overview of the Thesis.

1.3 Main Contributions

The main contributions of this research work are:

- In Chapters 2, 3, and 4 a survey of the Kalman filter algorithm formulations is given, starting from the very basic linear implementations, and ending with the state of the art in nonlinear Kalman filtering for large scale systems.
- We introduce a new suboptimal linear Kalman filter algorithm –the singular square root Kalman filter (SSQRT-KF)– which has been designed for large scale systems when the process noise can be assumed to be negligible, §3.3.
- We introduce a new optimal linear Kalman filter algorithm –the spatially localized Kalman filter (SLKF)– which has been designed to make local estimations in large scale systems when the measurement points are sparse, §3.4.
- We introduce a new suboptimal nonlinear ensemble Kalman filter –the ensemble spatially localized Kalman filter (EnSLKF)– which has been designed to make local estimations in nonlinear large scale systems when the measurement points are sparse. §4.7
- In Chapter 5, we introduce the sequential data assimilation topic in space weather forecast using ensemble-based Kalman filters.
- In §5.3.1, we develop a linear discrete-time state-space representation of a 2D-MHD flow system.
- In §5.4.3, we present an implementation of a modular data-assimilation system, integrated by a numerical MHD model; the VAC code, and the ensemble-based Kalman filter algorithms; coded in Matlab.

1.4 Published Results

Barrero Mendoza O. and B. L. R. De Moor, Singular Square Root Filter Algorithm (SSQR) for Large Scale Data Assimilation, Internal report TR 03-161, Katholieke Universtiteit Leuven, ESAT/SISTA, October 20, 2003.

Barrero Mendoza O., and B. L. R. De Moor,, A Singular Square Root Filter Algorithm for Large Scale Data Assimilation, *Proceedings of the 15th IASTED International Conference in Modelling and Simulation*, Marina del Rey, L.A., March 2004, pp. 418-422.

Barrero Mendoza O., B. L. R. De Moor, and P. Van Dooren, Kalman Filter for Data Assimilation in Linear Systems with Negligible Process Noise: State and Boundary Conditions Estimation, Internal report TR 03-161b, Katholieke Universtiteit Leuven, ESAT/SISTA, September, 2004.

Barrero Mendoza O., B. L. R. De Moor, and D. S. Bernstein, Data Assimilation for Magneto-Hydrodynamics Systems, *Journal of Computational and Applied Mathematics*, September 2004. In press.

Barrero Mendoza O., D. S. Bernstein, and B. L. R. De Moor, Data Assimilation in 2D Magneto-Hydrodynamics Systems, Internal report TR 04-74, Katholieke Universtiteit Leuven, ESAT/SISTA, September 17, 2004.

Chandrasekar, J., O. Barrero Mendoza, D.S Bernstein, and B. L. R. De Moor, State Estimation for Linearized MHD Flow, *Proceedings of the 43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, December 2004, pp. 2584-2589.

Barrero Mendoza O., D. S. Bernstein, and B. L. R. De Moor, Spatially Localized Kalman Filtering for Data Assimilation, *Accepted for the American Control Conference*, June 2005.

Other publications related to the SOFT4S project I was involved at the beginning of my doctoral program.

Barrero Mendoza O. and B. L. R. De Moor, Nonparametric Regularized Time Delay Estimation, Internal report TR 03-36, Katholieke Universtiteit Leuven, ESAT/SISTA, August 22, 2003.

Barrero Mendoza O. and B. L. R. De Moor, Data Preprocessing in Soft4s, Internal report TR 02-93, Katholieke Universtiteit Leuven, ESAT/SISTA, Nov. 6, 2002.

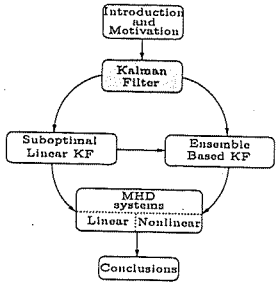
Barrero Mendoza O., L. Hoegarts, K. Pelckmans, and B. L. R. De Moor, Linear and Nonlinear Modelling in Soft4s, Internal report TR 02-188, Katholieke Universtiteit Leuven, ESAT/SISTA, Nov. 7, 2002.

Barrero Mendoza O. and B. L. R. De Moor, Nonparametric Regularized Time Delay Estimation, *Proceedings of the 2004 IEEE International Confer-*

ence on Acoustics, Speech, and Signal Processing, Montreal, Canada, May 2004, vol. 2, 541-544

Chapter 2

Linear Discrete-Time Kalman Filtering



The Kalman filter is an estimator for what is called the linear-quadratic Gaussian estimation problem, which is the problem of estimating the instantaneous stage of a linear dynamic system perturbed by a Gaussian random process -—process noise, and using measurements linearly related to the state but corrupted also by a Gaussian random process —measurement noise. The resulting estimator is statistically optimal with respect to any quadratic function of estimation error [45].

2.1 Introduction

This chapter is dedicated to introducing the reader to the Kalman filter and its different numerical formulations and algorithms. We make a review and

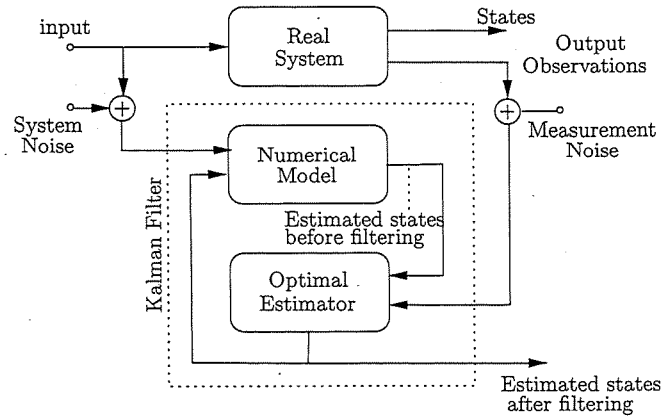


Figure 2.1: The Kalman filter.

study of some of the more relevant algorithms found in the literature, namely, the classical Kalman filter (KF), the square root filter Kalman filter (SQRT-KF), and the transform Kalman filter (TKF). Additionally, a comparison of the numerical complexity of these algorithms in number of flops is done at the end of the chapter.

2.2 The Kalman Filter

The Kalman filter is certainly one of the greater discoveries in the history of statistical estimation theory [45]. Its most immediate applications have been for the control of complex dynamic systems such as continuous manufacturing processes, aircraft, ships, or spacecraft. To control a dynamic system, you must first know what it is doing. For these applications, it is not always possible or desirable to measure every variable that you want to control, and the Kalman filter provides a means for inferring the missing information from indirect (and noisy) measurements. The Kalman filter is also used for predicting the likely future courses of dynamic systems that people are not likely to control, such as the flow of rivers during flood, the trajectories of celestial bodies, or the prices of traded commodities, Figure 2.1. In this thesis, we will focus on applications of estimating the state space variables of systems arising from discretized partial

differential equations, such as earth and space environmental systems.

It might seem strange that the term 'filter' would apply to an estimator. More commonly, a filter is a physical device for removing unwanted fractions of mixtures. Originally, a filter solved the problem of separating unwanted components of gas-liquid-solid mixtures. In the era of crystal radios and vacuum tubes, the term was applied to analog circuits that filter electronic signals. These signals are mixtures of different frequency components, and these physical devices preferentially attenuate unwanted frequencies.

This concept was extended in the 1930s and 1940s to the separation of signals from noise, both of which were characterized by their power spectral densities. Kolmogorov and Wiener used this statistical characterization of their probability distributions in forming an optimal estimate of the signal, given the sum of the signal and noise.

With Kalman filtering the term assumed a meaning that is well beyond the original idea of separation of the components of a mixture. It has also come to include the solution of an inversion problem, in which one knows how to represent the measurable variables as functions of the variables of principal interest. In essence, it inverts this functional relationship and estimates the independent variables as inverted functions of the dependent (measurable) variables. These variables of interest are also allowed to be dynamic, with dynamics that are only partially predictable.

Before making any mathematical derivation, it is important to clarify some important definitions related to the Kalman filtering that people often confuse, namely, filtering, prediction, and smoothing [4].

First, let us consider exactly what we mean by filtering. Suppose there is some quantity (possibly a vector quantity) associated with the system operation whose value we would like to know at each instant of time k. The quantity in question is denoted by $s(\cdot)$. It may be that this quantity is not directly measurable, or that it can only be measured with error. In any case, we shall suppose that noisy measurements $z(\cdot)$ are available, with $z(\cdot)$ not equal to $s(\cdot)$.

The term filtering is used in two senses. First, it is used as a generic term: filtering is the recovery from $z(\cdot)$ of $s(\cdot)$, or an approximation to $s(\cdot)$, or even some information about $s(\cdot)$. In other words, noisy measurements of a system are used to obtain information about some quantity that is essentially internal to the system. Second, it is used to distinguish a certain kind of information processing from two related kinds, smoothing and prediction. In this sense, filtering means the recovery at time k of some information about $s(k)$

using measurements up to time k. The important thing to note is the triple occurrence of the time argument k. First, we are concerned with obtaining information about $s(\cdot)$ at time k, i.e., $s(k)$. Second, the information is available at time k, not at some later time, and third, measurements right up to, but not after, time k are used.

Smoothing differs from filtering in that the information about $s(k)$ need not become available at time k, and measurements derived later than time k can be used in obtaining information about $s(k)$. This means there must be a delay in producing the information about $s(k)$, as compared with the filtering case, but the penalty of having a delay can be weighed against the ability to use more measurement data than in the filtering case in producing the information about $s(k)$. Not only does one use measurements up to time k, but one can also use measurements after time k. For this reason, one should expect the smoothing process to be more accurate in some sense than the filtering process.

Finally, prediction is the forecasting side of information processing. The aim is to obtain at time k information about $s(k+\lambda)$ for some $\lambda > 0$, i.e., to obtain information about what $s(\cdot)$ will be like subsequent to the time at which the information is produced. In obtaining the information, measurements up to time k can be used.

Now, after this short explanation a numerical derivation of the Kalman filter is given in the next section.

2.3 Numerical Formulation

A discretized linear time-dependent partial differential equation[1] (PDE) can be written as a state space representation of a linear time variant (LTV) stochastic model as follows:

$$x_{k+1} = A_k x_k + B_k u_k + G_k w_k, \tag{2.1}$$

$$y_k = C x_k + v_k. \tag{2.2}$$

[1] A PDE is an equation involving functions and their partial derivatives. In general PDEs are much more difficult to solve analytically than are ordinary differential equations. They may sometimes be solved using a Bäcklund transformation, characteristics, Green's function, integral transform, Lax pair, separation of variables, or —when all else fails (which it frequently does)— numerical methods such as finite differences. http://mathworld.wolfram.com.

where $A_k \in \mathbb{R}^{n\times n}$, $B_k \in \mathbb{R}^{n\times p}$, $G_k \in \mathbb{R}^{n\times l}$ and $C_k \in \mathbb{R}^{m\times n}$, with n the order of the system, p the number of inputs, l the number of uncertain parameters, and m the number of outputs. Additionally we have

$$\begin{aligned}\mathcal{E}[w_j w_k^{\mathrm{T}}] &\triangleq Q_k\delta_{jk},\\ \mathcal{E}[v_j v_k^{\mathrm{T}}] &\triangleq R_k\delta_{jk},\\ \mathcal{E}[v_j w_k^{\mathrm{T}}] &\triangleq 0,\end{aligned}$$

where $\mathcal{E}[\cdot]$ denotes expectation, w_k is a white Gaussian system noise process with zero mean and covariance matrix Q_k, and the white Gaussian measurement noise process v_k with zero mean and covariance matrix R_k. The process noise is introduced to take into account model uncertainties that may arise in a number of ways, for instance: unknown and unpredictable inputs except for their statistical properties, unknown and/or unpredictable boundary conditions, unknown or no modeled physics, computation errors caused by the discretization of the model, etc., and the measurement noise to take into account the measurement uncertainties.

The problem of estimating the states of (2.1) from measurements of the output (2.2) is discussed in this section. Different estimators can be derived depending on the available measurements as it was explained in the introduction of this chapter. Therefore, assuming that the data

$$\mathbf{Y}_k = \{y_i, u_i | i \leq k\}$$

is known, then, by using $\mathbf{Y}_k$ we want to estimate $x_{k+\lambda}$. There are three distinct cases:

1. Smoothing ($\lambda < 0$)
2. Estimation ($\lambda = 0$)
3. Prediction ($\lambda > 0$)

Figure 2.2 illustrates the different cases. In the next section the estimation and prediction problem are discussed. The resulting dynamic system is called a filter regardless of which of the problems is solved. We refer to [4] for the proofs of the results given in this section.

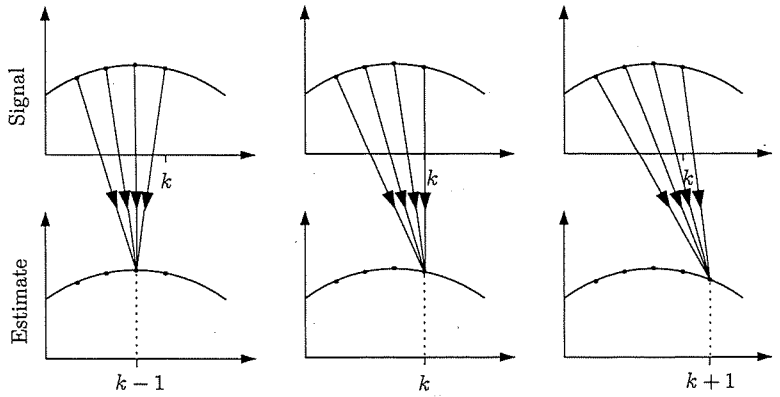


Figure 2.2: Smoothing, estimation, and prediction

Estimation Problem

An estimation of the signal x_k is concerned with the estimation of $x_{k+\lambda}$ for $\lambda = 0$ using the measurements y_i for $0 \leq i \leq k$. Consider the discrete-time dynamical system described by (2.1) and (2.2). For this system, we take a state estimator of the form

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}). \tag{2.3}$$

where $K_k \in \mathbb{R}^{n\times m}$, with output

$$\hat{y}_{k|k-1} = C_k\hat{x}_{k|k-1}.$$

In order to obtain recursively an optimal estimator K_k, the first step of the procedure is to propagate ahead $x_{k-1|k-1}$ via (2.1)

$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1}, \tag{2.4}$$

this is called the time-update step, then, define the prior state estimation error by

$$e_{k|k-1} \triangleq x_k - \hat{x}_{k|k-1},\ k > 0. \tag{2.5}$$

Substituting (2.4) and (2.1) into (2.5) we obtain

$$e_{k|k-1} = A_{k-1}e_{k-1|k-1} + G_{k-1}w_{k-1}.$$

Now, define the prior error covariance matrix by

$$P_{k|k-1} \triangleq \mathcal{E}[e_{k|k-1}e_{k|k-1}^{\mathrm{T}}],$$

hence,

$$P_{k|k-1} = A_{k-1}P_{k-1|k-1}A_{k-1}^{\mathrm{T}} + G_{k-1}Q_{k-1}G_{k-1}^{\mathrm{T}}. \tag{2.6}$$

Next, define the state estimator error

$$e_{k|k} \triangleq x_k - \hat{x}_{k|k}, \tag{2.7}$$

then, substituting (2.3) into (2.7) we have

$$e_{k|k} = (I_n - K_kC_k)e_{k|k-1} - K_kv_k. \tag{2.8}$$

Then, the Kalman gain K_k is obtained by minimizing the mean square error

$$J_k(K_k) = tr(P_{k|k}),$$

where the estimation error covariance matrix $P_{k|k} \in \mathbb{R}^{n\times n}$ is defined by

$$P_{k|k} \triangleq \mathcal{E}[(e_{k|k} - \mathcal{E}[e_{k|k}])(e_{k|k} - \mathcal{E}[e_{k|k}])^{\mathrm{T}}], \tag{2.9}$$

so, replacing (2.7) into (2.9) yields

$$P_{k|k} = (I_n - K_kC_k)P_{k|k-1}(I_n - K_kC_k)^{\mathrm{T}} + K_k\hat{R}_kK_k^{\mathrm{T}}, \tag{2.10}$$

where $\hat{R}_k \in \mathbb{R}^{m\times m}$ is defined by

$$\hat{R}_k \triangleq R_k + C_kP_{k|k-1}C_k^{\mathrm{T}}. \tag{2.11}$$

Therefore, by setting $\partial J_k(K_k)/\partial K_k = 0$ the resulting Kalman gain is

$$K_k = P_{k|k-1}C_k^{\mathrm{T}}\hat{R}_k^{-1}. \tag{2.12}$$

Next, substituting (2.12) into (2.10) the error covariance matrix update, or the measurement update step, is given by

$$P_{k|k} = (I_n - K_kC_k)P_{k|k-1}. \tag{2.13}$$

Notice that if the system is linear time invariant (LTI), observable, controllable, and $k \to \infty$, $P_{k+1|k}$ converges to a steady-state positive semidefinite matrix P_s, and K_k approaches the constant matrix L_s, given by

$$L_s = P_sC^{\mathrm{T}}\hat{R}^{-1},$$

where P_s satisfies the discrete algebraic Riccati equation (DARE)

$$P_s = (I_n - L_sC)(AP_sA^{\mathrm{T}} + GQG^{\mathrm{T}}).$$

Algorithm 1 *Kalman Filter Estimator*

Given R_k, Q_k, P_0,

For $k = 1, \ldots,$

1. *Update* $\hat{x}_{k-1|k-1}$ *to* $\hat{x}_{k|k-1}$, *and* $P_{k-1|k-1}$ *to* $P_{k|k-1}$

$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1},$$
$$P_{k|k-1} = A_{k-1}P_{k-1|k-1}A_{k-1}^{\mathrm{T}} + G_{k-1}Q_{k-1}G_{k-1}^{\mathrm{T}}.$$

2. *Compute* K_k

$$K_k = P_{k|k-1}C_k^{\mathrm{T}}\hat{R}_k^{-1}.$$

3. *Update* $\hat{x}_{k|k-1}$ *to* $\hat{x}_{k|k}$, *and* $P_{k|k-1}$ *to* $P_{k|k}$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - C_k\hat{x}_{k|k-1}),$$
$$P_{k|k} = (I_n - K_kC_k)P_{k|k-1}.$$

end

Prediction Problem

A prediction of the signal x_k is concerned with the estimation of $x_{k+\lambda}$ for $\lambda > 0$ using the measurements y_i for $0 \le i \le k$. Although a one-step or multiple-step prediction can be considered, here we analyze the one-step predictor.

For the system (2.1) and (2.2) consider the state estimator of the form

$$\hat{x}_{k+1|k} = A_k\hat{x}_{k|k-1} + B_ku_k + K_k(y_k - \hat{y}_{k|k-1}), \tag{2.14}$$

with output

$$\hat{y}_{k|k-1} = C_k \hat{x}_{k|k-1},$$

where $K_k \in \mathbb{R}^{n \times m}$ is the Kalman gain and $\hat{x}_{k|k-1}$ is the estimate of x_k based on observations up to time $k-1$. Defining the prediction error by

$$e_{k+1|k} \triangleq x_{k+1} - \hat{x}_{k+1|k}, \tag{2.15}$$

then substituting (2.1) and (2.14) into (2.15) yields

$$e_{k+1|k} = (A_k - K_k C_k)\tilde{x}_{k|k-1} - K_k v_k + G_k w_k, \tag{2.16}$$

where the Kalman gain K_k minimizes the mean square error

$$J_k(K_k) = tr(P_{k+1|k}),$$

with the prediction error covariance matrix $P_{k+1|k} \in \mathbb{R}^{n \times n}$

$$P_{k+1|k} \triangleq \mathcal{E}[(e_{k+1|k} - \mathcal{E}[e_{k+1|k}])(e_{k+1|k} - \mathcal{E}[e_{k+1|k}])^{\mathrm{T}}].$$

As a result, the Kalman gain can be obtained by

$$K_k = A_k P_{k|k-1} C_k^{\mathrm{T}} \hat{R}_k^{-1}, \tag{2.17}$$

with the error covariance matrix update given by

$$P_{k+1|k} = (A_k - K_k C_k) P_{k|k-1} A_k^{\mathrm{T}} + G_k Q_k G_k^{\mathrm{T}}, \tag{2.18}$$

which is a discrete Riccati difference equation (DRDE). Similar to the estimation case, when the system is observable, controllable, and $k \to \infty$, the steady state solution for the Kalman filter is as follows,

$$K_s = A P_s C^{\mathrm{T}} (R + C P_s C^{\mathrm{T}})^{-1},$$

where P_s satisfies the discrete algebraic Riccati equation (DARE)

$$P_s = (A - K_s C) P_s A^{\mathrm{T}} + G Q G^{\mathrm{T}}.$$

Algorithm 2 *Kalman Filter Predictor*

Given R_k, Q_k, P_0,

For $k = 1, \ldots,$

1. *Compute* K_k

$$K_k = A_k P_{k|k-1} C_k^{\mathrm{T}} \hat{R}_k^{-1}.$$

2. *Update* $\hat{x}_{k|k-1}$ *to* $\hat{x}_{k+1|k}$, *and* $P_{k|k-1}$ *to* $P_{k+1|k}$

$$P_{k+1|k} = (A_k - K_k C_k) P_{k|k-1} A_k^{\mathrm{T}} + G_k Q_k G_k^{\mathrm{T}},$$
$$\hat{x}_{k+1|k} = A_k \hat{x}_{k|k-1} + B_k u_k + K_k (y_k - C_k \hat{x}_{k|k-1}).$$

end

In this thesis the smoothing case is not analyzed because despite being a more accurate method than the previous ones, it is quite expensive to compute for large scale systems. Anyhow, there exist many literature about it, some references are [4, 45].

On the other hand and continuing with the analysis of the Kalman filter algorithms, in the next section some results obtained by De Souza *et al.* [27] and Gallivan et al. [39] are shown. These give us a deeper understanding about the convergency and numerical issues involved in the Kalman filtering computation.

Convergence of the Discrete Riccati Difference Equation –DRDE

To analyze the convergence of the DRDE (2.18) the underlying two-point boundary value problem [3, 90] is used for the case when the system, the process noise Q_k, and the measurement noise R_k are time invariant, hence, we drop the subscript k in this section. Taking $G = I_n$, it follows from lemma 1 in [39], that if $R = R^{\mathrm{T}} > 0$, (2.18) can be written as follows,

$$\begin{bmatrix} A^T & 0 \\ -Q & I_n \end{bmatrix} \begin{bmatrix} I_n \\ P_{k+1|k} \end{bmatrix} = \begin{bmatrix} I_n & C^T R^{-1} C \\ 0 & A \end{bmatrix} \begin{bmatrix} I_n \\ P_{k|k-1} \end{bmatrix} A_{F_k}$$

where

$$A_{F_k} \triangleq A - C^{\mathrm{T}} F_k,$$
$$F_k \triangleq \hat{R}_k^{-1} C P_k A^{\mathrm{T}},$$
$$\hat{R}_k \triangleq R + CPkC^{\mathrm{T}}.$$

Hence, the solution of (2.18) converge according to a subspace iteration given by

$$\begin{bmatrix} I_n \\ P_{k+1|k} \end{bmatrix} = M_1^{-1} M_2 \begin{bmatrix} I_n \\ P_{k|k-1} \end{bmatrix} A_{F_k},$$

with

$$M_1 = \begin{bmatrix} A^T & 0 \\ -Q & I_n \end{bmatrix}, \quad M_2 = \begin{bmatrix} I_n & C^T R^{-1} C \\ 0 & A \end{bmatrix},$$

and M_1 invertible.

Therefore, the following theorem introduced in [27] gives the necessary and sufficient conditions to obtain the unique stabilizing solution P_s.

Theorem 1 *A stabilizing solution P_s of the DRDE exists and is unique if and only if either of the following two conditions is satisfied.*

Subject to either $P_0 > 0$ or $P_0 \geq P_s$, then

1. *(C, A) is detectable and (A, L_Q) has no unobservable eigenvalue on the unit circle*
2. *(C, A) is detectable and the pencil $\lambda M_1 - M_2$ has no generalized eigenvalue on the unit circle.*

with $Q = L_Q L_Q^{\mathrm{T}}$.

For more details on DRDE convergence, we refer to [39, 27].

2.4 Square Root Formulation

2.4.1 Introduction

[45]

The great success of the Kalman Filter was not without its problems, not the least of which was marginal stability of the numerical solution of the associated Riccati equation. In some applications, small roundoff errors tended to accumulate and eventually degrade the performance of the filter. In the decades immediately following the introduction of the Kalman filter, there appeared several better numerical implementations of the original formulas. Many of these were adaptations of methods previously derived for the least squares problems.

The square-root methods make use of matrix decomposition methods. These include the so-called QR decomposition of a matrix as the product of an orthogonal matrix Q and a triangular matrix R. The matrix R results from the application of orthogonal transformations of the original matrix. These orthogonal transformations tend to be well conditioned numerically. The operation of applying these transformations called the triangularization of the original matrix, and triangularization methods derived by Givens [41], Householder [55], and Gentleman [40] are used to make the Kalman filtering more robust against roundoff errors.

It was discovered early on that forcing symmetry on the solution of the matrix Riccati equation improved its apparent numerical stability - a phenomenon that was later given a more theoretical basis by Verhaegen and Van Dooren [85]. It was also found that the influence of roundoff errors could be ameliorated by artificially increasing the covariance of process noise in the Riccati equation. A symmetrized form of the discrete-time Riccati equation was developed by Bucy and Joseph [20] an used by R.C.K. Lee at Honeywell in 1964. This structural reformulation of the Kalman filter equations improved robustness against roundoff error in some applications, although later methods have performed better on some problems [77].

The first of these was the square-root implementation by Potter and Stern [72], first published in 1963 and successfully implemented for space navigation on the Apollo manned lunar exploration program. Potter and Stern introduced the idea of factoring the error covariance matrix P_k into Cholesky factors, in the format

$$P_k = S_k S_k^{\mathrm{T}},$$

and expressing the observational update equations in terms of the Cholesky factor S_k, rather than P_k. The result was better numerical stability of the filter implementation at the expense of added computational complexity. A generalization of the Potter and Stern method to handle vector-valued measurements was published by one of the authors [7] in 1968, but a more efficient implementation –in terms of triangular Cholesky factor– was published by Bennet in 1967 [15].

There was a rather rapid development of faster algorithmic methods for square-root filtering in the 1970s, following the work at NASA/JPL (then called the Jet Propulsion Laboratory, at the California Institute of Technology) in the late 1960s by Dyer and McReynolds [31] on temporal update methods for Cholesky factors. Extensions of square-root covariance and information filters were introduced in Kaminski's 1971 thesis [64] at Stanford University. The first of the triangular factoring algorithms for the observational update was due to Agee and Turner [2], in a 1972 report of rather limited circulation. These algorithms have roughly the same computational complexity as the conventional Kalman Filter, but with better numerical stability. The fast-triangular algorithm of Bierman in 1974 [18] and the associated temporal update method introduced by Thornton [76]. The computational complexity of the square-root filter for time-invariant systems was greatly simplified by Morf and Kailath [69] soon after that. Specialized parallel processing architectures for fast solution of the square-root filter equations were developed by Jover and Kailath [61] and others over the next decade, and much simpler derivations of these and earlier square-root implementations were discovered by Kailath [63].

Although there exist several square-root based methods to compute the Kalman filter in the literature [4, 18, 62, 69, 81], in this chapter we will present, in our opinion and having in mind the use of them in large scale linear/nonlinear systems, some of the more representative , namely, the error covariance update using simultaneous [69] and sequential [72] processing, and the transform Kalman filter [19].

2.4.2 Error Covariance Update using Simultaneous Processing

Let us start with the derivation of the algorithm for the prediction case and later we extend these results directly to the estimation case (for more details see [69]).

First of all, substitute (2.17) in (2.18) to yield the discrete Riccati difference

equation (DRDE)

$$P_{k+1|k} = A_k P_{k|k-1} A_k^{\mathrm{T}} - A_k P_{k|k-1} C_k^{\mathrm{T}} \hat{R}_k^{-1} C_k P_{k|k-1} A_k^{\mathrm{T}} + G_k Q_k G_k^{\mathrm{T}}. \quad (2.19)$$

Note that this equation is the Schur complement of M_k with respect to $\hat{R}_k$, where M_k is positive semidefinite and it is defined as

$$M_k \triangleq \begin{bmatrix} \hat{R}_k & C_k P_{k|k-1} A_k^{\mathrm{T}} \\ A_k P_{k|k-1} C_k^{\mathrm{T}} & A_k P_{k|k-1} A_k^{\mathrm{T}} + G_k Q_k G_k^{\mathrm{T}} \end{bmatrix}.$$

From this, one easily derives a factored form of the Kalman filter algorithm as follows. It is assumed that the Cholesky factorizations of the positive semidefinite matrices R_k, Q_k and $P_{k|k-1}$ are given by

$$R_k \triangleq L_{R_k} L_{R_k}^{\mathrm{T}}, \quad Q_k \triangleq L_{Q_k} L_{Q_k}^{\mathrm{T}}, \quad P_{k|k-1} \triangleq S_{k|k-1} S_{k|k-1}^{\mathrm{T}}.$$

Next, M_k is easily factored in

$$M_k = \mathcal{G}_k \mathcal{G}_k^{\mathrm{T}}$$

where

$$\mathcal{G}_k \triangleq \begin{bmatrix} L_{R_k} & C_k S_{k|k-1} & 0 \\ 0 & A_k S_{k|k-1} & G_k L_{Q_k} \end{bmatrix}.$$

Now, from (2.19) notice that

$$P_{k+1|k} + A_k P_{k|k-1} C_k^{\mathrm{T}} \hat{R}_k^{-1} C_k P_{k|k-1} A_k^{\mathrm{T}} = A_k P_{k|k-1} A_k^{\mathrm{T}} + G_k Q_k G_k^{\mathrm{T}},$$

hence, M_k can be also written as

$$M_k = \begin{bmatrix} \hat{R}_k & C_k P_{k|k-1} A_k^{\mathrm{T}} \\ A_k P_{k|k-1} C_k^{\mathrm{T}} & P_{k+1|k} + A_k P_{k|k-1} C_k^{\mathrm{T}} \hat{R}_k^{-1} C_k P_{k|k-1} A_k^{\mathrm{T}} \end{bmatrix},$$

which can be factored by inspection as

$$M_k = \mathcal{X}_k \mathcal{X}_k^{\mathrm{T}}$$

with

$$\mathcal{X}_k \triangleq \begin{bmatrix} H_k & 0 & 0 \\ J_k & S_{k+1|k} & 0 \end{bmatrix}.$$

Where $H_k \in \mathbb{R}^{m\times m}$ satisfies

$$\hat{R}_k = H_k H_k^{\mathrm{T}},$$

and $S_{k+1|k} \in \mathbb{R}^{n\times n}$

$$P_{k+1|k} = S_{k+1|k} S_{k+1|k}^{\mathrm{T}},$$

then, $K_k \in \mathbb{R}^{n\times m}$ is computed as

$$K_k = J_k H_k^{-1}.$$

Then, it follows that as $\mathcal{G}_k$ and $\mathcal{X}_k$ are factors of M_k they must be related by an orthogonal transformation $\mathcal{U}_k$, i.e., $\mathcal{U}_k^{\mathrm{T}}\mathcal{U}_k = I$. Hence we can write the following relation,

$$\begin{bmatrix} L_{R_k} & C_k S_{k|k-1} & 0 \\ 0 & A_k S_{k|k-1} & G_k L_{Q_k} \end{bmatrix} \cdot \mathcal{U}_k = \begin{bmatrix} H_k & 0 & 0 \\ J_k & S_{k+1|k} & 0 \end{bmatrix}. \tag{2.20}$$

Therefore we can see that a technique for obtaining $\mathcal{X}_k$ is to apply any orthogonal transformation $\mathcal{U}_k$ such that we force a pattern of zeros as shown in (2.20). One technique that can be used is the QR decomposition [42], so, by computing an upper triangular QR decomposition of $\mathcal{G}_k^{\mathrm{T}}$ we have that

$$\mathcal{G}_k^{\mathrm{T}} = \mathcal{U}_k \mathcal{X}_k^{\mathrm{T}}$$

then,

$$\mathcal{G}_k \mathcal{U}_k = \mathcal{X}_k.$$

As a result, we have got an approach to update recursively the Riccati difference equation (2.19) assuring that $P_{k+1|k}$ is always positive definite. The price that has to be paid is the increase of complexity of the algorithm. The algorithm is described in Algorithm 3.

Algorithm 3 *SQRT-KF Predictor using Simultaneous Processing*

Given S_k, R_k, Q_k,

for $k = 1, \ldots,$

1. *Compute the Cholesky factorization of* R_k, *and* Q_k.
2. *Compute an upper triangular decomposition as in (2.20) to obtain* H_k, J_k, *and update* $S_{k|k-1}$ *to* $S_{k+1|k}$. *Hence,*

$$\begin{bmatrix} L_{R_k} & C_k S_{k|k-1} & 0 \\ 0 & A_k S_{k|k-1} & G_k L_{Q_k} \end{bmatrix} \cdot \mathcal{U}_k = \begin{bmatrix} H_k & 0 & 0 \\ J_k & S_{k+1|k} & 0 \end{bmatrix}.$$

3. *Compute the Kalman gain*

$$K_k = J_k H_k^{-1}.$$

4. *Update* $\hat{x}_{k|k-1}$ *to* $\hat{x}_{k+1|k}$

$$\hat{x}_{k+1|k} = A_k \hat{x}_{k|k-1} + B_k u_k + K_k(y_k - C_k \hat{x}_{k|k-1}).$$

end

These results can be easily extended to the estimation case. The main difference between the two algorithms, i.e, prediction and estimation, is that in the estimation case we have to make a time update step before updating $P_{k|k-1}$, therefore the square-root algorithm will be as follows. First the time update step is given by

$$\begin{bmatrix} A_{k-1} S_{k-1|k-1} & G_{k-1} L_{Q_{k-1}} \end{bmatrix} \cdot \mathcal{T}_k = \begin{bmatrix} S_{k/k-1} & 0 \end{bmatrix}, \tag{2.21}$$

with $\mathcal{T}_k$ an orthogonal transformation matrix as $\mathcal{U}_k$ in (2.20), then for the measurement update we defined M_k from (2.13) by

$$M_k \triangleq \begin{bmatrix} \hat{R}_k & C_k P_{k|k-1} \\ P_{k|k-1} C_k^{\mathrm{T}} & P_{k|k-1} \end{bmatrix}. \tag{2.22}$$

Next, the square root factor of M_k is given by

$$\mathcal{G}_k = \begin{bmatrix} L_{R_k} & C_k S_{k|k-1} \\ 0 & S_{k|k-1} \end{bmatrix}, \tag{2.23}$$

then, it follows from the QR decomposition of $\mathcal{G}_k^{\mathrm{T}}$ that

$$\begin{bmatrix} L_{R_k} & C_k S_{k|k-1} \\ 0 & S_{k|k-1} \end{bmatrix} \cdot \mathcal{U}_k = \begin{bmatrix} H_k & 0 \\ J_k & S_{k|k} \end{bmatrix}. \tag{2.24}$$

In order to reduce the complexity of the algorithm, one practical way to compute $\mathcal{X}_k$ is as follows, first compute $\mathcal{U}_k$ and H_k from a QR decomposition of the first row of $\mathcal{G}_k$

$$\begin{bmatrix} L_{R_k} & C_k S_{k|k-1} \end{bmatrix}^{\mathrm{T}} = \mathcal{U}_k \cdot \begin{bmatrix} H_k & 0 \end{bmatrix}^{\mathrm{T}}, \tag{2.25}$$

and then, compute J_k and $S_{k|k}$ as

$$\begin{bmatrix} 0 & S_{k|k-1} \end{bmatrix} \cdot \mathcal{U}_k = \begin{bmatrix} J_k & S_{k|k} \end{bmatrix}. \tag{2.26}$$

Another alternative is combining (2.21) and (2.24) into one step by substituting (2.6) in M_k. As a result, the square root of M_k can be rewritten as

$$\mathcal{G}_k = \begin{bmatrix} L_{R_k} & C_k A_{k-1} S_{k-1} & C_k G_{k-1} L_{Q_{k-1}} \\ 0 & A_{k-1} S_{k-1} & G_{k-1} L_{Q_{k-1}} \end{bmatrix},$$

note that the relative subscripts have changed to absolute subscripts because there is just one update step. Hence, the error covariance update can be done similarly to the original algorithm, so

$$\begin{bmatrix} L_{R_k} & C_k A_{k-1} S_{k-1} & C_k G_{k-1} L_{Q_{k-1}} \\ 0 & A_{k-1} S_{k-1} & G_{k-1} L_{Q_{k-1}} \end{bmatrix} \cdot \mathcal{U}_k = \begin{bmatrix} H_k & 0 & 0 \\ J_k & S_{k|k} & 0 \end{bmatrix}, \tag{2.27}$$

where $K_k \in \mathbb{R}^{n \times m}$ is computed as

$$K_k = J_k H_k^{-1},$$

and $S_{k|k} \in \mathbb{R}^{n \times n}$ satisfies

$$P_{k|k} = S_{k|k} S_{k|k}^{\mathrm{T}}. \tag{2.28}$$

Notice that (2.28) holds even if $P_{k+1|k}$ is not full rank. We can see the algorithm in Algorithm 4.

Algorithm 4 *SQRT-KF Estimator using Simultaneous Processing*

Given S_k, R_k, Q_k,

for $k = 1, \ldots,$

1. *Compute the Cholesky factorization of* R_k, *and* Q_k.
2. *Update* $\hat{x}_{k-1|k-1}$ *to* $\hat{x}_{k|k-1}$ *and* $S_{k-1|k-1}$ *to* $S_{k|k-1}$ *from a QR decomposition as follows*

$$\begin{bmatrix} A_{k-1} S_{k-1|k-1} & G_{k-1} L_{Q_{k-1}} \end{bmatrix} \cdot T_k = \begin{bmatrix} S_{k/k-1} & 0 \end{bmatrix},$$
$$\hat{x}_{k|k-1} = A_{k-1} \hat{x}_{k-1|k-1} + B_{k-1} u_{k-1}.$$

3. *Compute* $\mathcal{U}_k$ *and* H_k *from the QR decomposition of* $\begin{bmatrix} L_{R_k} & C_k S_{k|k-1} \end{bmatrix}^{\mathrm{T}}$ *as follows*

$$\begin{bmatrix} L_{R_k} & C_k S_{k|k-1} \end{bmatrix}^{\mathrm{T}} = \mathcal{U}_k \cdot \begin{bmatrix} H_k & 0 \end{bmatrix}^{\mathrm{T}}.$$

4. *Compute* J_k *and update* $S_{k|k-1}$ *to* $S_{k|k}$ *using*

$$\begin{bmatrix} 0 & S_{k|k-1} \end{bmatrix} \cdot \mathcal{U}_k = \begin{bmatrix} J_k & S_{k|k} \end{bmatrix}.$$

5. *Compute* K_k

$$K_k = J_k H_k^{-1}.$$

6. *Update* $\hat{x}_{k|k-1}$ *to* $\hat{x}_{k|k}$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - C_k \hat{x}_{k|k-1}).$$

end

2.4.3 Error Covariance Update using Sequential Processing

This method of propagating the error covariance matrix was proposed by Potter, [72, 4]. For the estimation case, in equations (2.6) and (2.12) we can

substitute $P_{k|k-1}$ by its square root form defined as

$$P_{k|k-1} = S_{k|k-1} S_{k|k-1}^{\mathrm{T}}.$$

Hence (2.6) can be rewritten as

$$P_{k|k} = S_{k|k-1} S_{k|k-1}^{\mathrm{T}} - S_{k|k-1} S_{k|k-1}^{\mathrm{T}} C_k^{\mathrm{T}} \hat{R}_k^{-1} C_k S_{k|k-1} S_{k|k-1}^{\mathrm{T}}, \tag{2.29}$$

and (2.12) as

$$K_k = S_{k|k-1} S_{k|k-1}^{\mathrm{T}} C_k^{\mathrm{T}} (C_k S_{k|k-1} S_{k|k-1}^{\mathrm{T}} C_k^{\mathrm{T}} + R_k)^{-1}. \tag{2.30}$$

Now assume that the observations have uncorrelated errors and are processed one at a time. As a result, for the i-th measurement point we have that $R_k(i,i) = \sigma_{i_k}^2$, where $\sigma_{i_k}^2$ is the variance of the observation error; therefore, $\hat{R}_k^{-1}$ becomes the scalar α_k defined as

$$\alpha_k \triangleq \left(\mathcal{F}_k^{\mathrm{T}} \mathcal{F}_k + \sigma^2\right)^{-1},$$

where $\mathcal{F}_k \in \mathbb{R}^{n \times m}$ is defined by

$$\mathcal{F}_k \triangleq S_{k|k-1}^{\mathrm{T}} C_k^{\mathrm{T}}. \tag{2.31}$$

Then, replace $\mathcal{F}_k$ in (2.29) and (2.30), so we obtain

$$K_k = \alpha_k S_{k|k-1} \mathcal{F}_k,$$

and,

$$P_{k|k} = S_{k|k-1} (I_n - \alpha_k \mathcal{F}_k \mathcal{F}_k^{\mathrm{T}}) S_{k|k-1}^{\mathrm{T}}.$$

Now, let us introduce a matrix $\mathcal{D}_k \in \mathbb{R}^{n \times n}$ such that

$$\mathcal{D}_k \mathcal{D}_k^{\mathrm{T}} = I_n - \alpha \mathcal{F}_k \mathcal{F}_k^{\mathrm{T}}, \tag{2.32}$$

hence

$$P_{k|k} = S_{k|k-1} \mathcal{D}_k \mathcal{D}_k^{\mathrm{T}} S_{k|k-1}^{\mathrm{T}}.$$

To find $\mathcal{D}_k$ we introduce the scalar γ_k so that

$$\left(I_n - \alpha_k \mathcal{F} \mathcal{F}^{\mathrm{T}}\right) = \left(I_n - \gamma_k \mathcal{F}_k \mathcal{F}_k^{\mathrm{T}}\right) \left(I_n - \gamma_k \mathcal{F}_k \mathcal{F}_k^{\mathrm{T}}\right)^{\mathrm{T}}. \tag{2.33}$$

Solving (2.33) for γ_k yields

$$\gamma_k = \frac{\alpha_k}{1 + \sqrt{\alpha_k \sigma^2}}.$$

Then, it follows that

$$P_{k|k} = S_{k|k} S_{k|k}^{\mathrm{T}}$$

where $S_{k|k} \in \mathbb{R}^{n \times n}$ is defined by

$$S_{k|k} \triangleq S_{k|k-1} \mathcal{D}_K. \tag{2.34}$$

If the measurements are correlated, i.e., R_k is not diagonal then these measurements can be transformed. Let $\tilde{y}_k$ be defined by

$$\tilde{y}_k \triangleq R_k^{-1/2} y_k,$$

where $R_k^{-1/2}$ is the matrix inverse of the Cholesky factor of R_k. Then

$$\tilde{y}_k = \tilde{C}_k x_k + \tilde{v}_k,$$

with,

$$\begin{aligned} \tilde{C}_k &\triangleq R_k^{-1/2} C_k, \\ \tilde{v}_k &\triangleq R_k^{-1/2} v_k. \end{aligned}$$

These transformed measurements are equivalent to the original measurements, but the covariance matrix of the errors of $\tilde{v}_k$ is the identity matrix.

Finally, the algorithm is given in Algorithm 5. It can be seen that the main advantages of this algorithm is that in step 3 (measurement update), the matrices related to the error covariance matrix update and the Kalman gain become vectors when only one measurement is processed at a time, making the computations more efficient.

Algorithm 5 *SQRT-KF Estimator using Sequential Processing*

Given S_k, R_k, Q_k,

For $k = 1, \ldots,$

1. *Compute the Cholesky factorization of* R_k, *and* Q_k.
2. *Update* $S_{k-1|k-1}$ *to* $S_{k|k-1}$ *and* $\hat{x}_{k-1|k-1}$ *to* $\hat{x}_{k|k-1}$

$$[A_k S_{k-1|k-1} \;\; G_k L_{Q_k}] U_k = [S_{k|k-1} \;\; 0],$$
$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1}.$$

3. *Update* $S_{k|k-1}$ *to* $S_{k|k}$ *and* $\hat{x}_{k|k-1}$ *to* $\hat{x}_{k|k}$ *using the sequential processing*

$S_{k|k_{(0)}} = S_{k|k-1}$,
$\hat{x}_{k|k_{(0)}} = \hat{x}_{k|k-1}$,
for $i = 1$ *to* m

$$\mathcal{F}_{k_{(i)}} = S^{\mathrm{T}}_{k|k_{(i-1)}} C^{\mathrm{T}}_{k_{(i,:)}},$$
$$\alpha_{k_{(i)}} = (\mathcal{F}^{\mathrm{T}}_{k_i}\mathcal{F}_{k_{(i)}} + R_{k_{(i,i)}})^{-1},$$
$$K_{k_{(i)}} = \alpha_{k_{(i)}} S_{k|k_{(i-1)}}\mathcal{F}_{k_{(i)}},$$
$$\hat{x}_{k|k_{(i)}} = \hat{x}_{k|k_{(i-1)}} + K_{k_{(i)}}(y_{k_{(i)}} - C_{k_{(i,:)}}\hat{x}_{k|k_{(i-1)}}),$$
$$\beta_{k_{(i)}} = \frac{1}{1+\sqrt{\alpha_{k_{(i)}} R_{k_{(i,i)}}}},$$
$$S_{k|k_{(i)}} = S_{k|k_{(i-1)}} - \beta_{k_{(i)}} K_{k_{(i)}}\mathcal{F}^{\mathrm{T}}_{k_{(i)}},$$

end.

end.

2.4.4 Error Covariance Update using a Matrix Transformation

Another way of updating the error covariance matrix is described in ([19, 78]) the ensemble transform Kalman filter (ETKF). In this section we present its linear formulation. This approach is based on the Potter algorithm §2.4.3 and

the Sherman-Morrison-Woodbury identity [42].

Notice that (2.32) for a simultaneous processing is given by

$$\mathcal{D}_k\mathcal{D}_k^{\mathrm{T}} = I_n - \mathcal{F}_k\hat{R}_k^{-1}\mathcal{F}_k^{\mathrm{T}}. \tag{2.35}$$

where $\mathcal{D}_k \in \mathbb{R}^{n\times n}$, $\mathcal{F}_k \in \mathbb{R}^{n\times m}$ is defined by (2.31), and $\hat{R}_k \in \mathbb{R}^{m\times m}$ by (2.11).

Using the Sherman-Morrison-Woodbury identity (2.36) [42],

Sherman-Morrison-Woodbury Identity

$$(A+UV^{\mathrm{T}})^{-1} = A^{-1} - A^{-1}U(I_n + V^{\mathrm{T}}A^{-1}U)^{-1}V^{\mathrm{T}}A^{-1}, \tag{2.36}$$

where $A \in \mathbb{R}^{n\times n}$, and $U,\ V \in \mathbb{R}^{n\times m}$.

we can rewrite (2.35) as follows

$$\mathcal{D}_k\mathcal{D}_k^{\mathrm{T}} = (I_n + \mathcal{F}_k R_k^{-1}\mathcal{F}_k^{\mathrm{T}})^{-1} \tag{2.37}$$

with $R_k \in \mathbb{R}^{m\times m}$ the measurement noise. Then, assuming R_k symmetric and non-singular, it can be shown that the eigenvalue decomposition of (2.37) is given by

$$(I_n + \mathcal{F}_k R_k^{-1}\mathcal{F}_k^{\mathrm{T}})^{-1} = V_k(\Lambda_{0_k} + I_n)^{-1}V_k^{\mathrm{T}},$$

with $\Lambda_{0_k} \in \mathbb{R}^{n\times n}$ is defined by

$$\Lambda_{0_k} \triangleq \begin{bmatrix} \Lambda_k & 0 \\ 0 & 0_{n-m} \end{bmatrix},$$

where $V_k \in \mathbb{R}^{n\times m}$, and $\Lambda_k \in \mathbb{R}^{m\times m}$ are the eigenvectors and eigenvalues of $\mathcal{F}_k R_k^{-1}\mathcal{F}_k^{\mathrm{T}}$. Hence,

$$\mathcal{D}_k = V_k(\Lambda_{0_k} + I_n)^{-1/2}.$$

As a result, the square root error covariance matrix update defined in (2.34) is given by

$$S_{k|k} = S_{k|k-1}V_k(\Lambda_{0_k} + I_n)^{-1/2},$$

and the Kalman gain by

$$K_k = S_{k|k-1}\mathcal{F}_k\hat{R}_k^{-1}.$$

As a result, the algorithm is shown in Algorithm 6.

Algorithm 6 *SQRT-TKF Estimator*

Given S_k, R_k, Q_k,

For $k = 1, \ldots,$

1. *Update* $S_{k-1|k-1}$ *to* $S_{k|k-1}$ *and* $\hat{x}_{k-1|k-1}$ *to* $\hat{x}_{k|k}$

$$[A_k S_{k-1|k-1} \;\; G_k L_{Q_k}]U_k = [S_{k|k-1} \;\; 0],$$
$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1}.$$

2. *Compute*

$$\mathcal{F}_k = S_{k|k-1}^{\mathrm{T}} C_k^{\mathrm{T}}.$$

3. *Compute the Kalman gain*

$$K_k = S_{k|k-1}\mathcal{F}_k\hat{R}_k^{-1}.$$

4. *Compute the eigenvalue decomposition of* $\mathcal{F}_k R_k^{-1}\mathcal{F}_k^{\mathrm{T}}$,

$$\mathcal{F}_k R_k^{-1}\mathcal{F}_k^{\mathrm{T}} = V_k\Lambda_k V_k^{\mathrm{T}}.$$

5. *Update* $S_{k|k-1}$ *to* $S_{k|k}$ *and* $\hat{x}_{k|k-1}$ *to* $\hat{x}_{k|k}$

$$S_{k|k} = S_{k|k-1}V_k(\Lambda_{0_k} + I_n)^{-1/2},$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - C_k\hat{x}_{k|k-1}).$$

end.

2.5 Computational Complexity of the Algorithm

To compare the algorithms we have computed the number of flops[2], see Table 2.1. Furthermore, In Table 2.2 the order of computation for each algorithm is shown. In Table 2.2 is observed that the order of computation for all the methods is $\mathcal{O}(n^3)$; however, the less expensive in number of flops is the SQRT-KF with error covariance sequential update. Notice that these results were obtained assuming a time invariant system, otherwise the full Kalman filter would be the less expensive in number flops. This is due to the fact that in the square-root approaches we have to add the Cholesky factorization costs of the Q_k, and R_k matrices each time step.

Full KF Estimator	$6n^3 + 2n^2(3m+1) + 2n[2m(m+1) + 2l^2 + p + 1]$
Full KF Predictor	$6n^3 + 2n^2[2m(m+1) + 2l^2 + p]$
SQRT-KF-Simul.Upd.	$16/3n^3 + 2n^2(m+l) + n(m^2 + 10/3l^2 + 1)$
SQRT-KF-Seq. Upd.	$10/3n^3 + 2n^2(m+l+7/2) + n(6m + 10/3l^2 + 2p + 2)$
SQRT-TKF	$37/3n^3 + n^2(8m+l+1) + n(6m^2 + 4m + 2/3l^2 + 2)$

Table 2.1: Number of flops, where n is the order of the system, m the number of observations, p the number of inputs, and l the order of the process-noise covariance matrix Q.

Notice that when the number of measurement points m, or the order of the process noise covariance matrix l, is equal to the order of the system n, the order of computation for the full KF predictor is $\mathcal{O}(n^4)$, while for the others stays in $\mathcal{O}(n^3)$, see Table 2.3. Now, in order to see more clear the dependence of the complexity of the algorithms on n and m, which are the most representative size dimensions, let us assume $m = l = p$, then

Full KF Estimator	$6n^3 + 2n^2(3m+1) + 2n(4m^2 + 3m + 1)$
Full KF Predictor	$6n^3 + 2n^2m(4m+3)$
SQRT-KF-Simul.Upd.	$16/3n^3 + 4n^2m + n(13/3m^2 + 1)$
SQRT-KF-Seq. Upd.	$10/3n^3 + 4n^2(m + 7/4) + n(10/3m^2 + 8m + 2)$
SQRT-TKF	$37/3n^3 + n^2(9m+1) + n(20/3m^2 + 4m + 2)$

Table 2.2: Number of flops, where n is the order of the system, and $m = l = p$.

[2] a flop is defined as a floating point operation [42]

Full KF Estimator	$\mathcal{O}(n^3)$
Full KF Predictor	$\mathcal{O}(n^3)$
SQRT-KF-Simul.Upd.	$\mathcal{O}(n^3)$
SQRT-KF-Seq. Upd.	$\mathcal{O}(n^3)$
SQRT-TKF	$\mathcal{O}(n^3)$

Table 2.3: Order of Computation

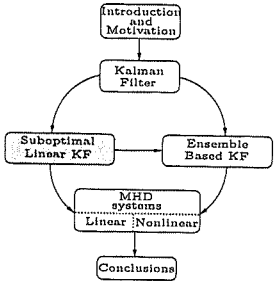
2.6 Conclusions

Basically there exist three different Kalman filter formulations, namely: 1) Predictor, 2) Estimator, and 3) Smoother. For each case there is a different problem formulation, therefore we have to use the one that better fits our problem. In fact, in most of the cases data assimilation systems use a short range forecast; which consist of making an estimation of the current states of the system, and then use a numerical model to predict the behavior of the system until new observations are taken. Hence, in this thesis we will use the estimator Kalman filter by default.

On the other hand, although the original formulation of the Kalman filter is optimal, it presents some numerical problems in the implementation, and lack of tractability when the dimensions of the system are large. In this chapter we have done a review of different approaches to compute the Kalman filter based on the square root formulation, first proposed by Potter [72]. The square root formulation has shown to improve the numerical stability as well as diminish the impact of roundoff errors. Having in mind the use of the Kalman filter further in large scale linear/nonlinear systems, basically there exist three different square root formulations, namely: 1) the error covariance update using simultaneous processing, 2) the error covariance update using sequential processing, and 3) the square root transform Kalman filter –SQRT-TKF, proposed by Bishop [19]. After calculating the number of flops needed for each algorithm, we conclude that the most efficient between them is the SQRT-KF with error covariance sequential update as can be seen in Table 2.1.

Chapter 3

Suboptimal Kalman Filters for Linear Large Scale Systems



3.1 Introduction

Although the Kalman filter has shown to provide an optimal solution for data assimilation in linear systems with a relatively small order, for very large nonlinear systems it is not directly applicable, because of its computational complexity and the size of the covariance matrix. As a result, approximations of the Kalman filter equations are needed. We will refer to these approximations as suboptimal KF. Most suboptimal KF are aimed at an approximation of either the model dynamics or the error covariance matrix, due to the fact that the

main part of the computations are used for them. The model is often simplified by removing less important terms from the equations, or by introducing other simplifying assumptions. The simplified model is then used for time propagation of the error covariance and the full model for the time propagation of the estimate. Some researchers [25], have proposed to approximate the state transition matrix by one of lower rank. The partial singular value decomposition can be used in this case to reduce the computations. On the other hand, various methods have been proposed for the approximation of the error covariance matrix. Setting correlations for large distances to zero can be exploited to speed up the algorithm considerably. However, due to the generally large condition number of the error covariance matrix negative eigenvalues may appear. A solution to this problem is to use a square root filter, as described in §2.4. Often the error covariance matrix has only a few large eigenvalues, which can be used for approximation. The resulting partial eigenvalue decomposition can be used for fast propagation of the error covariance. Todling and Cohn used this idea together with a Lanczos algorithm for the eigenvalue computations to obtain an efficient and general algorithm, the Partial Eigen-decomposition Kalman Filter or PEKF [25]. Another similar idea by Verlaan and Heemink is to approximate the error covariance matrix by one of a lower rank, the Reduced Rank Square Root algorithm or RRSQRT-KF [87]. The optimal choice for this low rank approximation results in the use of the eigenvalues and eigenvectors of the error covariance matrix.

This chapter presents an overview of different suboptimal linear Kalman filter formulations for large scale systems. On the one hand, the reduced rank square root Kalman filter (RRSQRT-KF), and the ensemble Kalman filter (EnKF), which are two well known suboptimal Kalman filters, and on the other hand we propose two new algorithms, namely, the singular square root Kalman filter SSQRT-KF [11], which is developed to be used for systems with negligible process noise, and the spatially localized Kalman filter [9], with its square root formulation SQRT-SLKF, which is designed to be used when the measurements are very sparse, so, a local estimation is the best solution we can have.

The RRSQRT-KF, the SSQRT-KF, and the KF are tested and compared in a 2D-plate heat transfer system, where the objective is to compare the performance of the filters when the propagated error covariance matrix is approximated according to the filter used. In addition, the problem of boundary condition estimation is treated, and a solution is given by inverting the linear dynamic system.

On the other hand, the SLKF is tested and compared to the KF in a mass-spring system, where the objective is to study the performance of the filter for

local estimation under different error weight functions.

Finally, the numerical algorithms of the filters are shown in detail, and a comparison of the numerical complexity is done.

3.2 Reduced Rank Square Root Kalman Filter - RRSQRT-KF

The main goal of the RRSQRT-KF is to approximate the error covariance matrix by a matrix with a smaller rank. Therefore, the square root factors are based on an eigen-decomposition. Let $P_{k|k-1} = V_k \Lambda_k V_k^T$ be the eigen-decomposition of the error covariance matrix $P_{k|k-1}$, so that $S_{k|k-1} = V_k \Lambda_k^{1/2}$ is a square root factor of $P_{k|k-1}$. The error covariance matrix is now approximated by using only q leading eigenvalues. With the ordering $|\lambda_1| \geq \ldots \geq |\lambda_n| \geq 0$, an approximation is obtained by truncating $S_{k|k-1}$ after the first q columns.

One way of computing the eigenvalue decomposition of $P_{k|k-1}$ efficiently is given in the following Theorem.

Theorem 2 *The eigenvalues different from zero and the corresponding eigenvectors, denoted as V_m, of the matrix $\mathcal{O} \in \mathbb{R}^{n \times n}$ with rank m, defined by*

$$\mathcal{O} \triangleq \mathcal{B}\mathcal{B}^{\mathrm{T}},$$

where $\mathcal{B} \in \mathbb{R}^{n \times m}$ has rank m, and $n > m$, can be computed as

$$V_m = \mathcal{B} M \Lambda_m^{-1/2},$$

where $M \in \mathbb{R}^{m \times m}$ and Λ_m are the right-eigenvectors and eigenvalues of $\mathcal{B}^{\mathrm{T}}\mathcal{B} \in \mathbb{R}^{m \times m}$ respectively, assuming $\lambda_1 > \lambda_2 > \ldots > \lambda_m$.

Proof: Given

$$\mathcal{B}\mathcal{B}^{\mathrm{T}} = V \Lambda_n V^{\mathrm{T}}$$

and,

$$\mathcal{B}^{\mathrm{T}}\mathcal{B} = M \Lambda_m M^{\mathrm{T}} \tag{3.1}$$

where Λ_m consists of the first m eigenvalues of $\mathcal{B}\mathcal{B}^{\mathrm{T}}$ and assuming $\lambda_1 > \lambda_2 > \ldots > \lambda_n$. Then, pre and post-multiplying (3.1) by $\mathcal{B}$ and $\mathcal{B}^{\mathrm{T}}$ respectively, it can

easily be shown that .

$$V_m \Lambda_m^2 V_m \mathrm{T} = \mathcal{B} M \Lambda_m^{-1/2} \Lambda_m^2 \Lambda_m^{-1/2} M^{\mathrm{T}} \mathcal{B}^{\mathrm{T}}.$$

As a result,

$$V_m = \mathcal{B} M \Lambda_m^{-1/2}.$$

The algorithm is as follows:

Algorithm 7 *RRSQRT-KF Estimator*

Given R_k, Q_k, S_0,

For $k = 1, \ldots,$

1. *Update $\hat{x}_{k-1|k-1}$ to $\hat{x}_{k|k-1}$, and $S_{k-1|k-1}$ to $S_{k|k-1}$ as follows*

$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1},$$
$$S_{k|k-1} = \left[A_{k-1}S_{k-1|k-1}\ G_{k-1}L_{Q_{k-1}}\right].$$

2. *Reduce the rank of $S_{k|k-1}$ using Theorem 2, so*

$$S^*_{k|k-1} = \left[S_{k|k-1}V_k\right]_{1:n,1:q},$$

where

$$S^T_{k|k-1}S_{k|k-1} = V_k \Lambda_k V_k^T.$$

3. *Update $\hat{x}_{k|k-1}$ and $S_{k|k-1}$ to $\hat{x}_{k|k}$ and $S_{k|k}$ respectively, using the sequential processing (see step 4 of the algorithm in §2.4.3).*

end.

3.3 Singular Square Root Kalman Filter SSQRT-KF

In the case of the state estimation for large scale systems, the computational complexity in (2.21)-(2.22) is large. However, it is possible to explore the

freedom of some parameters to derive low-rank algorithms. In this section we describe an algorithm that we call the Singular Square Root Kalman Filter (SSQRT-KF) [11, 13], which is able to compute a suboptimal Kalman filter gain very efficiently, assuming the process noise covariance $Q_{k-1} = 0$. It is inspired by a dual Linear Quadratic Regulator (LQR) algorithm described in [39]. In short, while the SSQR algorithm for the LQR focuses on obtaining a controller that just move the unstable eigenvalues of the system into the unit circle, guaranteeing the stabilization of the closed loop system, the SSQRT-KF focuses on obtaining an observer that guarantees the convergence of the estimation by making that the most dominant eigenvalues of the estimation error dynamics lie always into the unit circle, for details we refer to [13, 39].

Let us start with the time update step equation (2.10), assuming $Q_{k-1} = 0$, hence (2.21) can be written as

$$S_{k|k-1} = A_{k-1} S_{k-1|k-1}, \tag{3.2}$$

where the $S_{k-1|k-1} \in \mathbb{R}^{n \times q}$, with q chosen larger than the number of unstable eigenvalues of A_{k-1}. Therefore, we can rewrite (2.27) as follows

$$\begin{bmatrix} L_{R_k} & C_k S_{k|k-1} \\ 0 & S_{k|k-1} \end{bmatrix} \cdot \mathcal{U}_k = \begin{bmatrix} H_k & 0 \\ J_k & S_{k|k} \end{bmatrix}.$$

Then, notice that the QR decomposition can be done as in (2.25) and (2.26); therefore,

$$\begin{bmatrix} L_{R_k} & C_k S_{k|k-1} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} H_k & 0 \end{bmatrix} \cdot \mathcal{U}_k^T, \tag{3.3}$$

and then,

$$\begin{bmatrix} 0 & S_{k|k-1} \end{bmatrix} \cdot \mathcal{U}_k = \begin{bmatrix} J_k & S_{k|k} \end{bmatrix},$$

where the left-hand side of (3.3) is $m \times (m+q)$ making the QR decomposition cheap to compute. Besides, if C_k is sparse, the construction of the left factor in the left hand side of (2.22) is cheap as well.

Finally, a key characteristic of the SSQRT-KF algorithm is that the spectrum of the state space observer dynamics matrix $(I_n - P_{k|k-1} C_k^T \hat{R}_k^{-1} C_k) A_{k-1}$ of (2.8), is constructed by reflecting the eigenvalues of A_{k-1} with $|\lambda| > 1$ to their unit circle mirror images $1/|\lambda|$, and leaving the eigenvalues with $|\lambda| < 1$ unchanged. This feature is exploited to speed up the computation of the linear Kalman filter for large scale systems together with the covariance inflation factor, that is introduced in the following section.

3.3.1 SSQRT-KF with Covariance Inflation

The SSQRT-KF was designed initially to be applied in systems where the process noise can be neglected, anyhow this filter can be used when process noise is present in the system. In order to do this, a technique similar to the one introduced by Anderson [6] to avoid filter divergence in the ensemble Kalman filter, can be applied: the covariance inflation. The covariance inflation is a heuristic approach used in the SSQRT-KF to compensate the lack of the process noise term in (3.2), when the prior error covariance matrix is computed. It consists of multiplying the error covariance matrix $P_{k|k-1}$ by a factor $\kappa (\kappa > 1)$ enlarging the prior distribution artificially. Therefore, the error covariance update can be rewritten as

$$P_{k|k-1} = \tilde{A}_{k-1} P_{k-1|k-1} \tilde{A}_{k-1}$$

where $\tilde{A}_{k-1}$ is defined by

$$\tilde{A}_{k-1} \triangleq \sqrt{\kappa} A_{k-1}.$$

The choice of κ has to be done with care, because if κ is chosen too large it results in a filter in which the observations are given too much weight.

Consequently, the most interesting property of the covariance inflation factor in the case of SSQRT-KF is that when we deal with discretized PDE systems (2.1), the matrix A_k is in most of the cases stable, i.e, $|\lambda(A_k)| < 1$. Therefore we can choose κ such that the matrix $\tilde{A}_{k-1}$ gets the most dominant eigenvalues outside the unit circle. Since it was already mentioned above, one of the key characteristics of the SSQRT-KF is that the spectrum of the state space observer dynamics matrix is constructed by reflecting the eigenvalues of A_{k-1} with $|\lambda| > 1$ to their unit circle mirror images $1/|\lambda|$, leaving the eigenvalues with $|\lambda| < 1$ unchanged. As a result, the SSQRT-KF estimator will focus on stabilizing the most dominant modes of the error dynamics matrix, obtaining an efficient suboptimal Kalman filter.

Summarizing, the recursive SSQRT-KF algorithm will be as follows

Algorithm 8 *SSQRT-KF Estimator*

Given κ, R_k, Q_k, S_0,

for $k = 1, \ldots,$

1. *Update* $S_{k-1|k-1}$ *to* $S_{k|k-1}$ *and* $\hat{x}_{k-1|k-1}$ *to* $\hat{x}_{k|k-1}$

$$S_{k|k-1} = \sqrt{\kappa} A_{k-1} S_{k-1|k-1},$$
$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1}.$$

2. *Compute* H_k *and* $\mathcal{U}_k$ *from the QR decomposition of* $\left[L_{R_k} \;\; C_k S_{k|k-1}\right]^{\mathrm{T}}$ *as follows*

$$\left[L_{R_k} \;\; C_k S_{k|k-1}\right]^{\mathrm{T}} = \left[H_k \;\; 0\right] \cdot \mathcal{U}_k^T.$$

3. *Compute* J_k *and Update* $S_{k|k-1}$ *to* $S_{k|k}$

$$\left[0 \;\; S_{k|k-1}\right] \cdot \mathcal{U}_k = \left[J_k \;\; S_{k|k}\right].$$

4. *Compute* K_k

$$K_k = J_k H_k^{-1}.$$

5. *Update* $\hat{x}_{k|k-1}$ *to* $\hat{x}_{k|k}$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(y_k - C_k\hat{x}_{k|k-1}\right).$$

end.

3.3.2 Heat Transfer System Example

As an example, a heat transfer system in a finite insulated plate will be taken. The PDE that describes such a system [46] is given by

$$\frac{\partial T(t,y,x)}{\partial t} = \alpha\left(\frac{\partial^2 T(t,y,x)}{\partial x^2} + \frac{\partial^2 T(t,y,x)}{\partial y^2}\right) + u(t,y,x),$$

where $T(t,y,x)$, $u(t,y,x)$, and α are the temperature distribution, the heat input, and the heat transfer coefficient of the plate, respectively. In this exam-

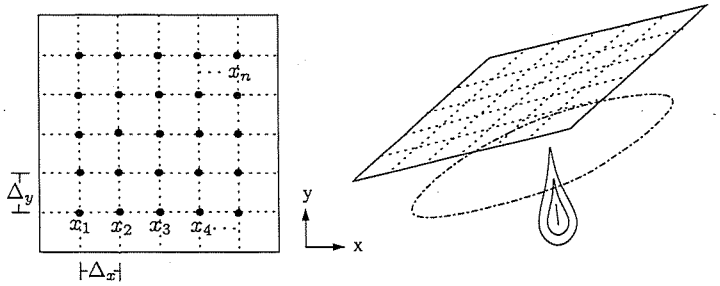


Figure 3.1: Heat transfer in a finite 2D-plate. At the left, the space discretization of the plate is shown, where each x_i point represents a state. At the right, the source of heat is represented by a flame moving under the plate.

ple the PDE is discretized using finite differences, with the plate dimensions $L_x = L_y = 1m$, $\alpha = 1 \times 10^{-5} W/^o Cm^2$, with fixed boundary conditions, initial conditions

$$T(0,y,x) = 0,$$

and the heat input

$$u(t,y,x) = 5 \times 10^{-5}\frac{\Delta t}{\Delta x} e^{\frac{-(x-x_0(t))^2}{2\sigma_x} + \frac{-(y-y_0(t))^2}{2\sigma_y}},$$

where $x_0(t)$ and $y_0(t)$ are values that change with time, and $\sigma_x = \sigma_y = 1\times 10^{-2}$. The system is discretized in 225 grid points equally spaced by $\Delta x = \Delta y = 0.0625$ m with $0 < x < 1$ and $0 < y < 1$, and a sampling time $\Delta t = 5.21$ seconds, this in order to keep the discretized system stable. Figure 3.2 shows the structure of the A matrix, this structure is very typical for discretized PDE's. We can observe that this matrix is very sparse a fact that can be used to make the computations more efficient.

For simulation purposes the short range prediction scheme is used. This scheme consists of estimating the states at time k, where the measurements y_k are available. Then, the prediction is done by projecting ahead the states using the physical first principle based model (2.1) until new measurements are taken. Three methods will be compared, namely, the Kalman filter, the RRSQRT-KF, and the SSQRT-KF.

In a first simulation, 30 measurement points were taken randomly, and the process noise was set to zero. Figure 3.3 shows a comparison of the root-mean

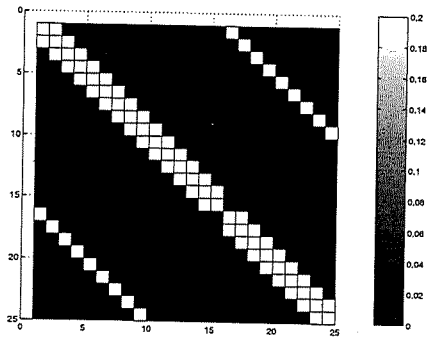


Figure 3.2: Sparse structure of the A matrix. The black areas show the zero values and the white ones the values bigger than zero.

square error (RMSE) between the three filters for this case. Although the three algorithms converge to the same solution, RRSQRT-KF converges slower while SSQRT-KF has a similar convergence behaviour to the classical KF.

In a second simulation, process noise was introduced by using just the inputs at the same points where the outputs are measured, using the same 30 measurement points of the previous simulation. The filters were computed assuming no process noise. Figure 3.4 shows the results of the RMSE, as expected none of the filters converge to zero.

The third simulation was done taking into account the process noise caused by the unknown inputs. The performance of the classical KF and RRSQRT-KF are much better now, while SSQRT-KF keeps the same performance of the previous case due to the fact that it does not take into account the process noise in its formulation, Figure 3.5 shows the results.

Figures 3.6, and 3.8 show a comparison of the RMSE, and the evolution of the temperature estimation between the three filters when covariance inflation is applied to them. Even though all the filters reduce the RMSE, SSQRT-KF reduces it drastically compared to the others. the performance of the three filters is similar,

Figure 3.7 depicts how the magnitude of the eigenvalues of the filter dynamics are affected compared to the eigenvalues of the system dynamics. At the bottom

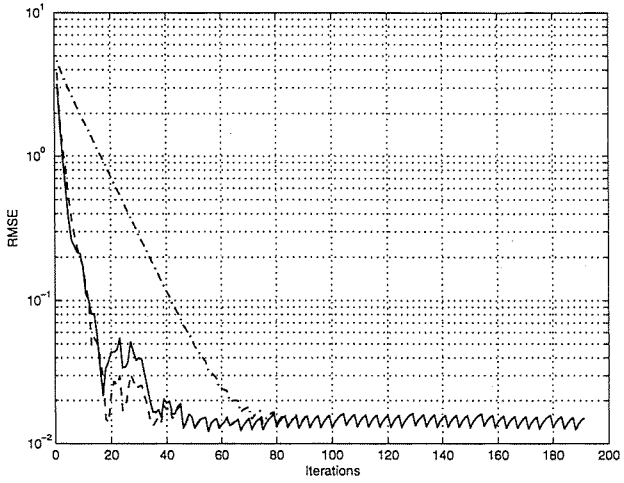


Figure 3.3: The RMSE of $\hat{x}_{k|k}$ for the KF, RRSQRT-KF, and SSQRT-KF. The KF Solid line, the RRSQRT-KF dashed-dotted line, and the SSQRT-KF dashed line. By assuming all the inputs known, the process noise is set to zero. 30 measurement points were taken randomly. As a result the estimation of the filters converge close to the real value.

can be seen for the SSQRT-KF case how the magnitude of the eigenvalues of $\tilde{A}_{k-1}$, for which $|\lambda(\tilde{A}_{k-1})| > 1$ are moved into the unit circle leaving the rest in their original locations.

Another important feature of the SSQRT-KF algorithm is its low computation cost as can be seen in Table (3.2). There can be seen that for $n \gg m$, l, q, p RRSQRT-KF and SSQRT-KF are much cheaper to compute than Kalman filter.

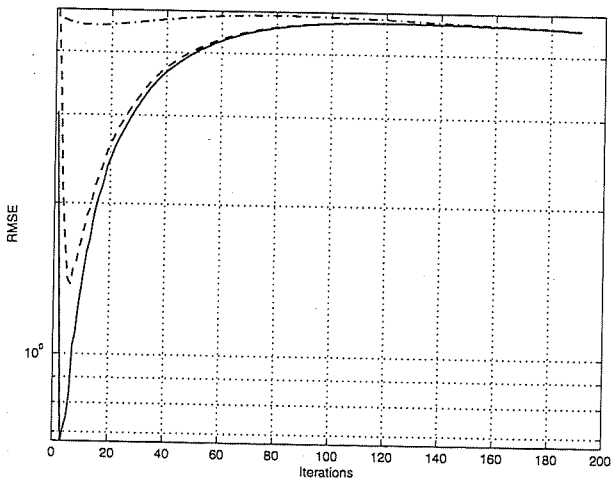


Figure 3.4: The RMSE of $\hat{x}_{k|k}$ for the KF , RRSQRT-KF, and SSQRT-KF. The KF Solid line, the RRSQRT-KF dashed-dotted line, and the SSQRT-KF dashed line. By taking only the inputs in the measurement points we generate the process noise. However, in this experiment we assume that there is no process noise for the filters. 30 measurement points were taken randomly. As a result the estimation of the filters does not converge close to the real value.

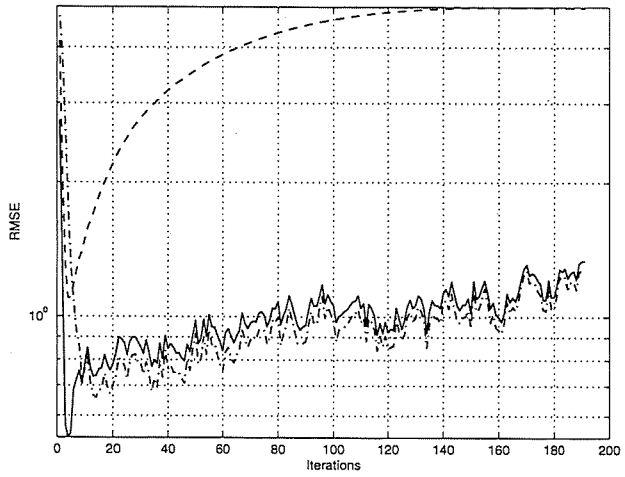


Figure 3.5: The RMSE of $\hat{x}_{k|k}$ for the KF , RRSQRT-KF, and SSQRT-KF. The KF Solid line, the RRSQRT-KF dashed-dotted line, and the SSQRT-KF dashed line. By taking only the inputs in the measurement points we generate the process noise. In this experiment the process noise is taking into account for the filters. As a result the KF and the RRSQRT-KF converge close to the real value, while SSQRT-KF does not converge as expected, because this filter always assume $Q = 0$, so, it is no affected by the fact of the presence or absence of the process noise.

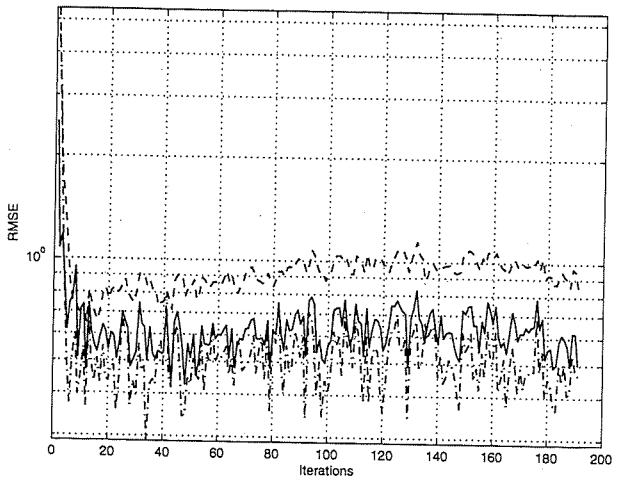


Figure 3.6: The RMSE of $\hat{x}_{k|k}$ for the KF , RRSQRT-KF, and SSQRT-KF. The KF Solid line, RRSQRT-KF dashed-dotted line, and SSQRT-KF dashed line. By taking only the inputs in the measurement points we generate the process noise. In this experiment the process noise is taking into account for the KF and RRSQRT-KF filters, and using the covariance inflation factor $\kappa = 1.3$, 3.3.1. As a result the performance of the SSQRT-KF is much better than in the previous case.

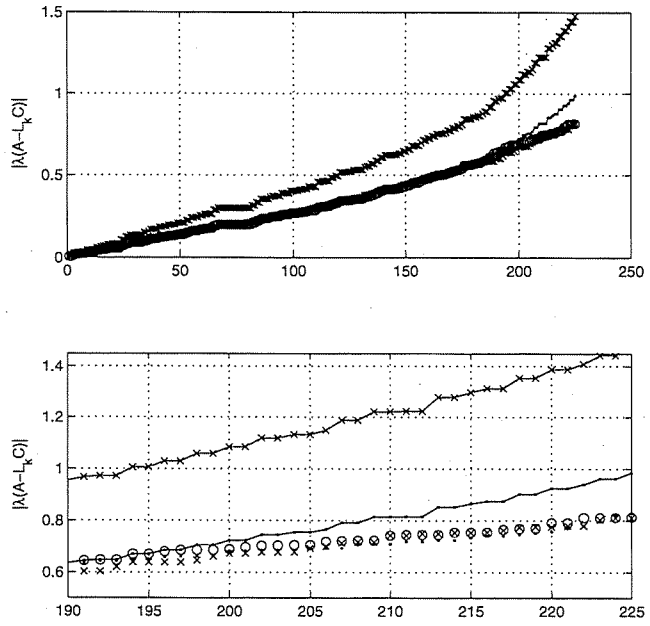


Figure 3.7: At the top can be seen the eigenvalues of the $(A-L_{kf}C)$, dot-mark, $(A-L_{ks}C)$, circle-mark, $(A-L_{kr}C)$, x-mark, A, dotted line, and $\tilde{A}$, x-mark line, where L_{kf}, L_{ks}, and L_{kr} are the KF, SSQRT-KF, and RRSQRT-KF gains, respectively. The plot at the bottom zooms into the plot at the top, and it depicts how L_{ks} relocates $|\lambda(\tilde{A})| > 1$.

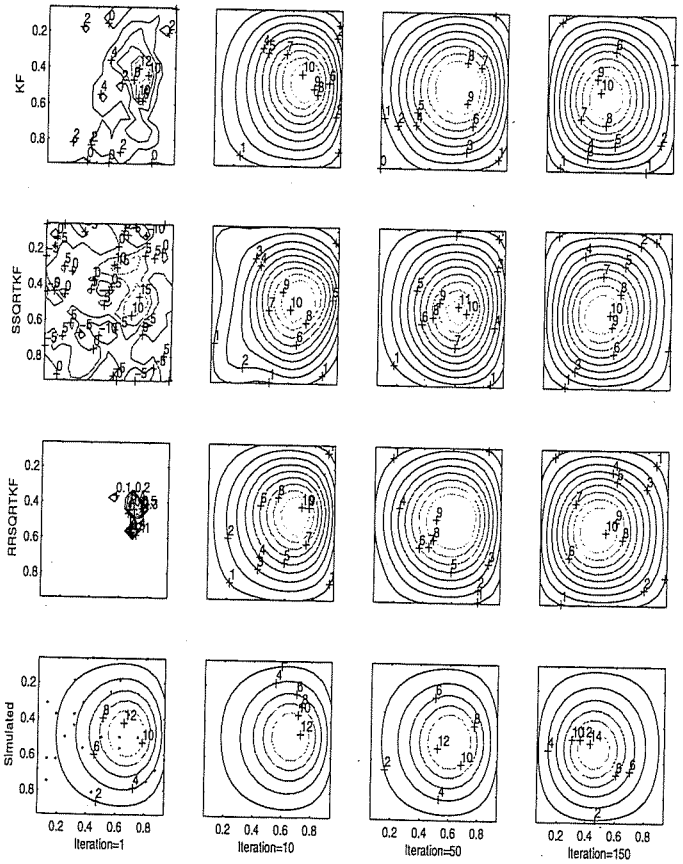


Figure 3.8: Contour plots of the temperature estimation using the three filters compared to the original one (simulated). Taking 30 measurement points (dots in the fourth row), where the inputs and outputs are measured. Covariance inflation is applied to SSQRT-KF, $\kappa = 1.3$. Random initial conditions and zero fixed boundary conditions. It can be seen that the performance of the SSQRT-KF with covariance localization is similar to the KF and RRSQRT-KF filter.

Boundary Conditions Estimation

Until now boundary conditions were assumed to be zero, but in real applications we often find that this is not true. Therefore, in this section the boundary conditions are added to the system described by (2.1) and (2.2). As a result, a discretized PDE with boundary conditions can be set as a LTV stochastic model state space representation as follows:

$$x_{k+1} = A_k x_k + B_k u_k + F_k z_k + w_k \tag{3.4}$$
$$y_k = C_k x_k + v_k. \tag{3.5}$$

where $A_k \in \mathbb{R}^{n\times n}$, $B_k \in \mathbb{R}^{n\times p}$, $C_k \in \mathbb{R}^{m\times n}$, and $F_k \in \mathbb{R}^{n\times b}$. With $z_k \in \mathbb{R}^b$ a vector of boundary conditions at time k. Now, let us define

$$\begin{aligned}\mathcal{E}[w_j w_k^T] &\triangleq Q_k \delta_{jk},\\ \mathcal{E}[v_j v_k^T] &\triangleq R_k \delta_{jk},\\ \mathcal{E}[v_j w_k^T] &\triangleq 0,\end{aligned}$$

where w_k is a white Gaussian system noise process with zero mean and covariance matrix Q_k which is introduced to take into account the model uncertainties, as well as the white Gaussian measurement noise process v_k with zero mean and covariance matrix R_k to take into account the measurements uncertainties. The initial state x_0 is assumed to be Gaussian with mean $\bar{x}_0$ and covariance matrix P_0.

In (3.4) is seen that the boundary condition vector z_k can be taken as external input to the system. Hence (3.4) and (3.5) can be rewritten as follows

$$\begin{aligned}x_{k+1} &= A_k x_k + (B_k \;\; F_k)\begin{pmatrix} u_k \\ z_k \end{pmatrix} + w_k\\ y_k &= C_k x_k + v_k.\end{aligned}$$

Now, in order to estimate the boundary conditions vector, we are going to invert the system using the Kalman filter such that the vector z_k will become part of the state space vector x_k, as explained in the following section.

Inverting Linear Dynamical Systems [86]

As an application of the Kalman filter, in this section we present how Kalman filtering can be used to invert a linear dynamical system. We start with a rather crude problem formulation. Let the signal generating system be described by the LTI formulation of the state space system (3.4), (3.5), and let

a desired output sequence y_{dk} be given for $k = 1, 2, \ldots, N$. Then the problem is to determine the input sequence z_k for $k = 1, \ldots, N$ such that y_k closely approximates y_{dk}.

A more precise problem formulation requires the definition of a model that describes how the input z_k is generated. For example, let us consider the following model representation for the class of inputs

$$z_{k+1} = \alpha z_k + w_{zk}. \tag{3.6}$$

This is an autoregressive model of order one –AR(1), with w_{zk} a white noise sequence that is independent of w_k and v_k in (3.4) and (3.5), and covariance

$$\mathcal{E}[w_{zj} w_{zk}^T] = Q_z \delta_{jk}.$$

For the input within this class of signal generating systems the problem is to determine the output covariance matrix $Q_z \in \mathbb{R}^{b \times b}$ and a realization of the input sequence z_k for $k = 1, 2, \ldots, N$, such that the output y_k is a minimum variance approximation of y_{dk}.

The combination of the time-invariant signal generating model and the model representing the class of input signals (3.6) results in the augmented state space model:

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} &= \begin{bmatrix} A & F \\ 0 & \alpha I_b \end{bmatrix} \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \\ &\quad \begin{bmatrix} I_n & 0 \\ 0 & I_b \end{bmatrix} \begin{bmatrix} w_k \\ w_{zk} \end{bmatrix}, \\ y_k &= [C \ 0] \begin{bmatrix} x_k \\ z_k \end{bmatrix} + v_k, \end{aligned} \tag{3.7}$$

with the process and measurement noise having the following covariance matrix:

$$\mathcal{E} \begin{bmatrix} w_k \\ w_{zk} \\ v_k \end{bmatrix} \begin{bmatrix} w_k^T & w_{zk}^T & v_k^T \end{bmatrix} = \begin{bmatrix} Q & 0 & 0 \\ 0 & Q_z & 0 \\ 0 & 0 & R \end{bmatrix}.$$

This augmented state space model has no measurable boundary conditions sequence. A bounded solution of the state error covariance matrix of the Kalman filter of the augmented state space model requires the pair

$$\left\{ \begin{bmatrix} A & F \\ 0 & \alpha I_b \end{bmatrix}, \begin{bmatrix} C & 0 \end{bmatrix} \right\}, \tag{3.8}$$

to be observable. The conditions under which the observability of the original pair (A, C) are preserved are given in the following lemma. Notice that here we extend the results showed in [86] to the case of $\alpha \neq 1$.

Lemma 3 *[86] Let the pair (A, C) be observable and let for all $\xi \in \mathbb{R}^m$: $C(A - \alpha I_n)^{-1} F \xi = 0$ if and only if $\xi = 0$, then the pair (3.8) is observable.*

Proof: By the Popov-Belevitch-Hautus [38] test for checking observability, we have to proof that for all eigenvectors of the system matrix

$$\begin{bmatrix} A & F \\ 0 & \alpha I_b \end{bmatrix},$$

denoted as

$$\begin{bmatrix} A & F \\ 0 & \alpha I_b \end{bmatrix} \begin{bmatrix} \eta \\ \xi \end{bmatrix} = \lambda \begin{bmatrix} \eta \\ \xi \end{bmatrix}, \tag{3.9}$$

the condition

$$\begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \xi \end{bmatrix} = 0$$

only holds provided $\begin{bmatrix} \eta \\ \xi \end{bmatrix} \equiv 0$.

From the lower part of (3.9) it follows that: $\xi = 0$ for any λ and α, or $\lambda = \alpha$ for any ξ. With $\xi = 0$ it follows from (3.9) that $A\eta = \lambda\eta$ and therefore $C\eta$ can only be zero provided that η is zero, since the pair (A, C) is observable. With $\lambda = \alpha$, the top row of (3.9) reads:

$$(A - \alpha I_n)\eta = -F\xi,$$

hence $C\eta = 0$ implies

$$C(A - \alpha I_n)^{-1} F \xi = 0,$$

but this can only hold provided that $\xi \equiv 0$.

The condition in Lemma 3 on ξ for single-input, single-output LTI systems is equivalent with the fact that the system (A, F, C) does not have zeros in the point $z = \alpha$ of the complex plane. For multivariable systems the condition corresponds to the original system having no so-called transmission zeros in the point $z = \alpha$, see details in [38].

The case where the original system has one or more zeros at α or in a circle of radius ε around $z = \alpha$, requires special attention. Different solutions are available to solve this problem. One approach is based on the factorization of rational transfer functions given in [80].

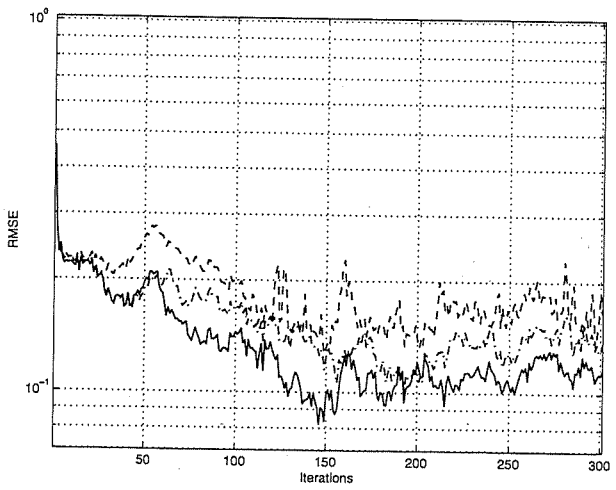


Figure 3.9: RMSE of $\hat{x}_{k|k}$ for the KF, RRSQRT-KF, and SSQRT-KF. The KF Solid line, RRSQRT-KF dashed-dotted line, and SSQRT-KF dashed line. The boundary conditions are fixed to zero except the left one, where the disturbance $z_{k+1} = z_k + w_{zk}$ is injected. Randomly 30 measurement points were taken, where the outputs are measured.

Numerical Results

In order to investigate how this technique works in a data assimilation problem, the example of section §3.3.2 will be taken. The boundaries are fixed to zero except for the left boundary where the disturbance input $z_{k+1} = z_k + w_k$ with $\alpha = 1$ is injected. The new augmented system has 15 new unknown states, corresponding to the boundary conditions of the left side.

Figure 3.9 shows the RMSE of the three filters when the disturbance input $z_{k+1} = z_k + w_{zk}$ is injected. In this simulation is assumed all the inputs are known, and 30 output measurement points. Covariance inflation is used for SSQRT-KF with $\kappa = 1.05$

Figure 3.10 shows the contour plot of the estimated temperature using the three

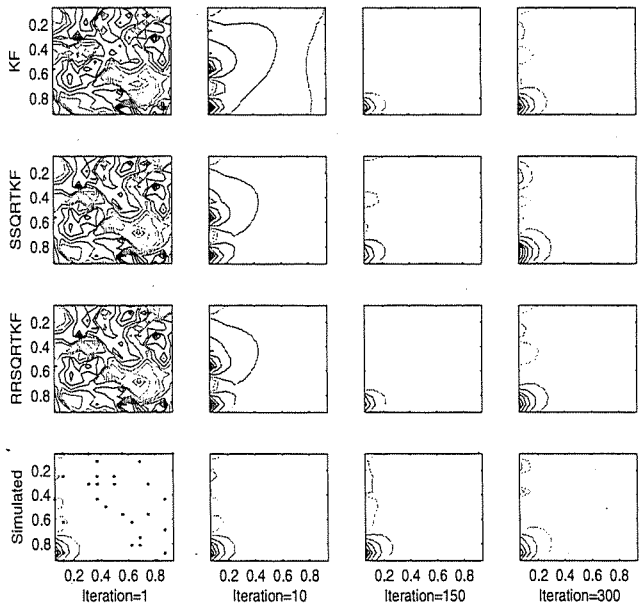


Figure 3.10: Contour plots of the estimated temperature using the three filters compared to the original one (simulated). Boundary conditions fixed to zero except the left one, where the disturbance $z_{k+1} = z_k + w_{zk}$ is injected. Randomly 30 measurement points (dots in the fourth row) were taken, where the outputs are measured. Covariance inflation is applied to SSQRT-KF, $\kappa = 1.05$.

filters. Even though the classical KF is the best, RRSQRT-KF and SSQRT-KF do a good job in estimating the unknown input disturbances.

3.4 Spatially Localized Kalman Filter - SLKF

In this section we introduce a new filter we called, the spatially localized Kalman filter (SLKF). The classical Kalman filter provides optimal least-squares estimates of all of the states of a linear time-varying system under process and measurement noise. In many applications, however, optimal estimates are desired for a specified subset of the system states, rather than all of the system states. For example, for systems arising from discretized partial differential equations, the chosen subset of states can represent the desire to estimate state variables associated with a subregion of the spatial domain. However, it is well known that the optimal state estimator for a subset of system states coincides with the classical Kalman filter.

For applications involving high-order systems, it is often difficult to implement the classical Kalman filter, and thus it is of interest to consider computationally simpler filters that yield suboptimal estimates of a specified subset of states. One approach to this problem is to consider reduced-order Kalman filters. Such reduced-complexity observers provide estimates of the desired states that are suboptimal relative to the classical Kalman filter [17, 47, 54, 48, 49, 59]. Alternative variants of the classical Kalman filter have been developed for computationally demanding data assimilation applications such as weather forecasting [36, 53, 8, 37], where the classical Kalman filter gain and covariance are modified so as to reduce the computational requirements.

The present approach is motivated by computationally demanding applications such as those discussed in [36, 53, 8, 37]. For such applications, a high-order simulation model is assumed to be available, and the derivation of a reduced-order filter in the sense of [17, 47, 54, 48, 49, 59] is not feasible due to the lack of a tractable analytic model. Instead, we consider the use of a full-order state estimator based directly on the simulation model. However, rather than implementing the classical output injection Kalman filter, we derive a suboptimal *spatially localized Kalman filter* [9, 10] in which the filter gain is constrained a priori to reflect the desire to estimate a specified subset of states, something similar to the covariance localization approach §4.4.2, but in an optimal way.

The use of a spatially localized Kalman filter in place of the classical Kalman filter is motivated by computational architecture constraints arising from a multi-processor implementation of the Kalman filter [65] in which the Kalman filter operations can be confined to the subset of processors associated with the states whose estimates are desired. An additional motivation is the use

of the extended Kalman filter for nonlinear systems. For systems with sparse measurements, observability may not hold for the entire system. In this case, the spatially localized Kalman filter can be used to obtain state estimates for the observable portion of the system.

Despite the objective of estimating a subset of the system states, the technique we develop in this section requires the propagation of a full-order covariance matrix.

3.4.1 Numerical Formulation

We begin by considering the discrete-time dynamical system

$$x_k = A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + G_{k-1}w_{k-1}, \quad k \geq 0, \tag{3.10}$$

with output

$$y_k = C_k x_k + v_k, \tag{3.11}$$

where $x_{k-1} \in \mathbb{R}^n$, $u_{k-1} \in \mathbb{R}^p$, $y_k \in \mathbb{R}^m$, and $A_{k-1}, B_{k-1}, G_{k-1}, C_k$ are known real matrices of appropriate size. The input u_{k-1} and output y_k are assumed to be measured, and $w_{k-1} \in \mathbb{R}^l$ and $v_k \in \mathbb{R}^m$ are zero-mean noise processes with known variances and correlations given by Q_{k-1}, R_k, and F_k, respectively. We assume that Q_{k-1}, R_k, and S_k are positive definite.

The problem of estimating a subset of states of (3.10) from measurements of the output (3.11) is discussed in this section.

Estimation Problem

Consider the discrete-time dynamical system described by (3.10) and (3.11). For this system, we take a state estimator of the form

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \Gamma_k K_k(y_k - \hat{y}_{k|k-1}), \quad k \geq 0, \tag{3.12}$$

with output

$$\hat{y}_{k|k-1} = C_k \hat{x}_{k|k-1}.$$

where $\hat{x}_{k|k} \in \mathbb{R}^n$ is the estimation of x_k using the measurements y_i for $0 \leq i \leq k$, $\hat{y}_{k|k-1} \in \mathbb{R}^m$, $\Gamma_k \in \mathbb{R}^{n \times n_p}$, and $K_k \in \mathbb{R}^{n_p \times m}$. The nontraditional feature of (3.12) is the presence of the term Γ_k, which, in the classical case of output injection, is the identity matrix. Here, Γ_k constrains the state estimator so

that only states in the range of Γ_k are directly affected by the gain K_k. For example, Γ_k can have the form

$$\Gamma_k = \begin{bmatrix} 0 \\ I_{n_p} \\ 0 \end{bmatrix},$$

where I_{n_p} denotes the $n_p \times n_p$ identity matrix. We assume that Γ_k has full column rank for all $k \geq 0$.

In order to find the optimal gain K_k, the first step is to project $x_{k-1|k-1}$ ahead via (3.10) using

$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1}. \tag{3.13}$$

Then, define the prior state estimation error by

$$e_{k|k-1} \triangleq x_k - \hat{x}_{k|k-1}. \tag{3.14}$$

Substituting (3.13) and (3.10) into (3.14) we obtain

$$e_{k|k-1} = A_{k-1}e_{k-1|k-1} + G_{k-1}w_{k-1}.$$

Now, define the prior error covariance matrix by

$$P_{k|k-1} \triangleq \mathcal{E}[e_{k|k-1}e_{k|k-1}^{\mathrm{T}}],$$

where $\mathcal{E}$ denotes expected value. Hence,

$$P_{k|k-1} = A_{k-1}P_{k-1|k-1}A_{k-1}^{\mathrm{T}} + G_{k-1}Q_{k-1}G_{k-1}^{\mathrm{T}}.$$

Next, define the state estimation error

$$e_{k|k} \triangleq x_k - \hat{x}_{k|k}, \tag{3.15}$$

and the weighted estimation error covariance matrix

$$J_k(K_k) \triangleq \mathcal{E}[(L_k e_{k|k})^T (L_k e_{k|k})],$$

where $L_k \in \mathbb{R}^{q \times n}$ determines the weighted error components. Then, the weighted estimation error can be obtained as

$$J_k(K_k) = \mathrm{tr}\ (P_{k|k}M_k), \tag{3.16}$$

where the error covariance matrix $P_{k|k} \in \mathbb{R}^{n \times n}$ is defined by

$$P_{k|k} \triangleq \mathcal{E}[e_{k|k}e_{k|k}^{\mathrm{T}}], \tag{3.17}$$

and $M_k \in \mathbb{R}^{n \times n}$ by

$$M_k \triangleq L_k^{\mathrm{T}}L_k.$$

Now, substituting (3.12) into (3.15) yields

$$e_{k|k} = x_k - \hat{x}_{k|k-1} - \Gamma_k K_k(y_k - C_k\hat{x}_{k|k-1}), \tag{3.18}$$

and using (3.18) with (3.17) implies

$$P_{k|k} = \hat{A}_k P_{k|k-1}\hat{A}_k^{\mathrm{T}} + \hat{Q}_k, \tag{3.19}$$

where

$$\begin{aligned}
P_{k|k-1} &\triangleq \mathcal{E}[e_{k|k-1}e_{k|k-1}^{\mathrm{T}}], \\
e_{k|k-1} &\triangleq x_k - \hat{x}_{k|k-1}, \\
\hat{A}_k &\triangleq I_{n_k} - \Gamma_k K_k C_k, \\
\hat{Q}_k &\triangleq \Gamma_k K_k \tilde{R}_k K_k^{\mathrm{T}}\Gamma_k^{\mathrm{T}} - F_k K_k^{\mathrm{T}}\Gamma_k^{\mathrm{T}} - \Gamma_k K_k F_k^{\mathrm{T}}, \\
\tilde{R}_k &\triangleq C_k S_k + S_k^{\mathrm{T}}C_k^{\mathrm{T}} + R_k, \\
F_k &\triangleq \mathcal{E}[w_{k-1}v_k^{\mathrm{T}}].
\end{aligned}$$

Hence (3.16) becomes

$$J_k(K_k) = \mathrm{tr}\ [(\hat{A}_k P_{k|k-1}\hat{A}_k^{\mathrm{T}} + \tilde{Q}_k)M_k].$$

To obtain the optimal gain K_k we set $\partial J_k(K_k)/\partial K_k = 0$, which gives

$$K_k = (\Gamma_k^{\mathrm{T}}M_k\Gamma_k)^{-1}\Gamma_k^{\mathrm{T}}M_k\hat{F}_k\hat{R}_k^{-1},$$

with

$$\begin{aligned}
\hat{F}_k &\triangleq F_k + P_{k|k-1}C_k^{\mathrm{T}}, \in \mathbb{R}^{n \times m}, \\
\hat{R}_k &\triangleq \tilde{R}_k + C_k P_{k|k-1}C_k^{\mathrm{T}}, \in \mathbb{R}^{m \times m}.
\end{aligned}$$

To update the error covariance matrix, we first note that

$$\Gamma_k K_k = \pi_k \hat{F}_k \hat{R}_k^{-1}, \tag{3.20}$$

where $\pi_k \in \mathbb{R}^{n\times n}$ is defined by

$$\pi_k \triangleq \Gamma_k(\Gamma_k^{\mathrm{T}} M_k \Gamma_k)^{-1}\Gamma_k^{\mathrm{T}} M_k.$$

Note that π_k is an oblique projector, that is, $\pi_k^2 = \pi_k$. Now using (3.20) with (3.19) yields the error covariance matrix update equation

$$P_{k|k} = P_{k|k-1} + \pi_{k\perp}\hat{F}_k\hat{R}_k^{-1}\hat{F}_k^{\mathrm{T}}\pi_{k\perp}^{\mathrm{T}} - \hat{F}_k\hat{R}_k^{-1}\hat{F}_k^{\mathrm{T}}, \tag{3.21}$$

where the complementary projector $\pi_{k\perp}$ is defined by

$$\pi_{k\perp} \triangleq I_{n_k} - \pi_k.$$

If either $M_k = I_n$ or $L_k = \Gamma_k^{\mathrm{T}}$, then π_k is the orthogonal projector

$$\pi_k = \Gamma_k(\Gamma_k^{\mathrm{T}}\Gamma_k)^{-1}\Gamma_k^{\mathrm{T}}.$$

On the other hand, if $p = q$, so that $L_k \in \mathbb{R}^{p\times p}$, then, it can be shown that

$$\pi_k = \Gamma_k(L_k\Gamma_k)^{-1}L_k.$$

Specializing to the case $F_k = 0$, $\Gamma_k = I_n$, and $L_k = I_n$, so that $\pi_{k\perp} = 0$, yields the familiar Riccati update equation

$$P_{k|k} = P_{k|k-1} - P_{k|k-1}C_k^{\mathrm{T}}\hat{R}_k^{-1}C_kP_{k|k-1}.$$

Hence the algorithm is as follows,

Algorithm 9 *SLKF Estimator*

Given R_k, Q_k, $P_0 \neq 0$, L_k, Γ_k, π_k, *and* $\pi_{k\perp}$.

For $k = 1, \ldots$

1. *Project ahead the error covariance matrix and the estimated states:*

$$P_{k|k-1} = A_{k-1}P_{k-1|k-1}A_{k-1}^{\mathrm{T}} + G_{k-1}Q_{k-1}G_{k-1}^{\mathrm{T}},$$
$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1}.$$

2. *Compute the SLFK gain*

$$\Gamma_k K_k = \pi_k\hat{F}_k\hat{R}_k^{-1}$$

3. *Update* $\hat{x}_{k|k-1}$ *to* $\hat{x}_{k|k}$ *as follows*

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \Gamma_k K_k(y_k - C_k\hat{x}_{k|k-1}).$$

4. *Update the error covariance matrix*

$$P_{k|k} = P_{k|k-1} + \pi_{k\perp}\hat{F}_k\hat{R}_k^{-1}\hat{F}_k^{\mathrm{T}}\pi_{k\perp}^{\mathrm{T}} - \hat{F}_k\hat{R}_k^{-1}\hat{F}_k^{\mathrm{T}}.$$

end

3.4.2 Square Root Formulation of the SLKF

To avoid numerical problems when computing the SLKF, a square root formulation, for the case $S_k = 0$ is presented is this section. We can rewrite (3.21) as

$$P_{k|k} = P_{1_{k|k}} + P_{2_{k|k}}, \tag{3.22}$$

where $P_{1_{k|k}}$ and $P_{2_{k|k}} \in \mathbb{R}^{n\times n}$ are defined by

$$P_{1_{k|k}} \triangleq P_{k|k-1} - \hat{F}_k\hat{R}_k^{-1}\hat{F}_k^{\mathrm{T}},$$
$$P_{2_{k|k}} \triangleq \pi_{k\perp}\hat{F}_k\hat{R}_k^{-1}\hat{F}_k^{\mathrm{T}}\pi_{k\perp}^{\mathrm{T}}. \tag{3.23}$$

Hence, the square root form of (3.22) can be written as

$$P_{k|k} = S_{k|k}S_{k|k}^{\mathrm{T}}$$

with $S_{k|k} \in \mathbb{R}^{n\times(n+m)}$ defined as

$$S_{k|k} \triangleq \left[S_{1_{k|k}}\ S_{2_{k|k}}\right],$$

where $S_{1_{k|k}} \in \mathbb{R}^{n\times n}$ and $S_{2_{k|k}} \in \mathbb{R}^{n\times m}$ satisfy

$$P_{1_{k|k}} = S_{1_{k|k}} S_{1_{k|k}}^{\mathrm{T}},$$
$$P_{2_{k|k}} = S_{2_{k|k}} S_{2_{k|k}}^{\mathrm{T}}.$$

To compute $S_{1_{k|k}}$, first notice that the Schur complement of $\hat{R}_k$ in M_k is $P_{1_{k|k}}$,

$$M_k = \begin{bmatrix} \hat{R}_k & \hat{F}_k^{\mathrm{T}} \\ \hat{F}_k & P_{1_{k|k-1}} \end{bmatrix}$$

Now specializing to the case $S_k = 0$ we have

$$M_k = \begin{bmatrix} R_k + C_k P_{k|k-1} C_k^{\mathrm{T}} & C_k P_{k|k-1} \\ P_{k|k-1} C_k^{\mathrm{T}} & P_{k|k-1} \end{bmatrix}. \tag{3.24}$$

Hence, the square root form of (3.24) is defined by

$$M_k \triangleq \beta_k \beta_k^{\mathrm{T}}$$

with $\beta_k \in \mathbb{R}^{(m+n)\times(m+N_q)}$ given by

$$\beta_k = \begin{bmatrix} L_{R_k} & C_k S_{k|k-1} \\ 0 & S_{k|k-1} \end{bmatrix} \tag{3.25}$$

with N_q the order of the rank approximation of $S_{k|k-1}$ defined below in (3.29), where $R_k \triangleq L_{R_k} L_{R_k}^{\mathrm{T}}$ and $P_{k|k-1} \triangleq S_{k|k-1} S_{k|k-1}^{\mathrm{T}}$. Next, a lower triangular QR decomposition of β_k^{T} yields

$$\begin{bmatrix} L_{R_k} & C_k S_{k|k-1} \\ 0 & S_{k|k-1} \end{bmatrix} = \begin{bmatrix} H_k & 0 \\ J_k & S_{1_{k|k}} \end{bmatrix} \cdot U_k^{\mathrm{T}},$$

where $U_k \in \mathbb{R}^{(m+N_q)\times(m+N_q)}$ is orthogonal. As a consequence, a square root factorization of M_k is given by

$$M_k = \begin{bmatrix} H_k & 0 \\ J_k & S_{1_{k|k}} \end{bmatrix} \begin{bmatrix} H_k^{\mathrm{T}} & J_k^{\mathrm{T}} \\ 0 & S_{1_{k|k}}^{\mathrm{T}} \end{bmatrix},$$

from which, assuming H_k is nonsingular, it follows that

$$K_k = \pi_k J_k H_k^{-1},$$
$$\hat{R}_k = H_k H_k^{\mathrm{T}}, \tag{3.26}$$
$$P_{k|k-1} C_k^{\mathrm{T}} = J_k H_k^{\mathrm{T}}. \tag{3.27}$$

Then, to find $S_{2_{k|k}}$ we substitute (3.26) and (3.27) into (3.23)

$$P_{2_{k|k}} = \pi_{k\perp} J_k H_k^{\mathrm{T}} (H_k H_k^{\mathrm{T}})^{-1} H_k J_k^{\mathrm{T}} \pi_{k\perp}^{T}$$

hence,

$$P_{2_{k|k}} = \pi_{k\perp} J_k J_k^{\mathrm{T}} \pi_{k\perp}^{\mathrm{T}},$$

from where

$$S_{2_{k|k}} = \pi_{k\perp} J_k.$$

As a result, the recursive SLKF algorithm is shown in Algorithm 10.

Algorithm 10 *SQRT-SLKF Estimator*

Given $S_0 \neq 0$, π_k, Γ_k, L_{Q_k}, *and* L_{R_k},

for $k = 1, \ldots,$

1. *Update* $\hat{x}_{k-1|k-1}$ *to* $\hat{x}_{k|k-1}$, *and* $S_{k-1|k-1}$ *to* $S_{k|k-1}$

$$\begin{aligned} \hat{x}_{k|k-1} &= A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1}, \\ S_{k|k-1} &= \left[A_{k-1}S_{k-1|k-1}\ G_{k-1}L_{Q_{k-1}}\right], \end{aligned} \tag{3.28}$$

where $Q_{k-1} \triangleq L_{Q_{k-1}} L_{Q_{k-1}}^{\mathrm{T}}$.

2. *Compute a reduced rank approximation of* $S_{k|k-1}$ *applying Theorem 2 as follows. First, compute the eigenvalue decomposition of* $P_{k|k-1}^{\mathrm{T}}$

$$S_{k|k-1}^{\mathrm{T}} S_{k|k-1} = V_k D_k V_k^{\mathrm{T}},$$

then, the reduced rank approximation of (3.28) can be yielded by

$$S_{k|k-1}^{*} = S_{k|k-1} V_{k_{(1:n,1:N_q)}}, \tag{3.29}$$

where $N_q \leq n$ *is the order chosen to approximate (3.28), hence* $S_{k|k-1}^{*} \in \mathbb{R}^{n \times N_q}$.

3. *Compute* H_k *and* U_k *making a QR decomposition of* $\left[L_{R_k}\ \ C_k S_{k|k-1}\right]$ *as follows*

$$\left[L_{R_k}\ \ C_k S_{k|k-1}^{*}\right] = \left[H_k\ \ 0\right] \cdot U_k^{\mathrm{T}}.$$

4. *Compute* J_k *and* $S_{1_{k|k}}$ *as follows*

$$\left[0\ \ S_{k|k-1}^{*}\right] \cdot U_k = \left[J_k\ \ S_{1_{k|k}}\right]$$

5. *Compute the SLKF gain* $\Gamma_k K_k$ *and* $S_{2_{k|k}}$

$$\begin{aligned} \Gamma_k K_k &= \pi_k J_k H_k^{-1}, \\ S_{2_{k|k}} &= \pi_{k\perp} J_k. \end{aligned}$$

6. *Update* $\hat{x}_{k|k-1}$ *to* $\hat{x}_{k|k}$, *and* $S_{k|k-1}^{*}$ *to* $S_{k|k}$

$$\begin{aligned} \hat{x}_{k|k} &= \hat{x}_{k|k-1} + \Gamma_k K_k \left(y_k - C_k \hat{x}_{k|k-1}\right) \\ S_{k|k} &= \left[S_{1_{k|k}}\ S_{2_{k|k}}\right] \end{aligned}$$

end

3.4.3 Mass-Spring System Example

To illustrate the performance of the SLKF a simple LTI mass-spring system is used (see Figure 3.11).

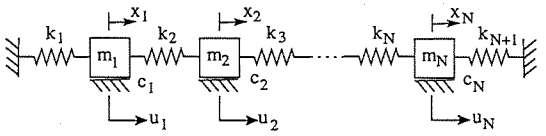


Figure 3.11: Mass-Spring system.

The state space representation in continuous time of this system is given by

$$\begin{bmatrix} \dot{z} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} A_z & A_x \\ I_N & 0_N \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} + \begin{bmatrix} I_N \\ 0_N \end{bmatrix} u,$$

with z_i and x_i the velocity and the position of the i-th mass, respectively, and N the number of masses. In this example the nodes at the extremes are assumed to be fixed, so, the number of analyzed nodes is equal to the number of masses. $A_x \in \mathbb{R}^{N \times N}$ is a tridiagonal matrix defined by

$$A_x \triangleq \begin{bmatrix} -(k_1+k_2)/m_1 & k_2/m1 & 0 & \cdots \\ k_2/m_2 & -(k_2+k_3)/m_2 & k_3/m_2 & \vdots \\ 0 & \ddots & \ddots & \ddots \\ \vdots & \cdots & k_N/m_N & -(k_N+k_{N+1})/m_N \end{bmatrix},$$

with k_i and m_i the spring constant and mass of the i-th node, respectively. $A_z \in \mathbb{R}^{N \times N}$ a diagonal matrix defined by

$$A_z \triangleq \begin{bmatrix} -c_1/m_1 & 0 & \cdots \\ 0 & \ddots & \vdots \\ \vdots & \cdots & -c_N/m_N \end{bmatrix},$$

where c_i the friction coefficient of the i-th mass.

For simplicity of the analysis we take the parameters in each node to be equal to the others; hence, $m_i = 1kg$, $k_i = 5kg/s^2$, and $c_i = 5kg/s$, with $i = 1, \ldots, N$. We selected $N = 50$, $m = 1$ i.e., one input applied in the node 25 and defined by

$$u_{25}(t) = 30\sin(t/2 + \pi/3).$$

Next, the system is discretized using the zero order hold method taking a sampling time $T_s = 0.1s$. Now, the discrete system can be represented by

$$x_{k+1} = A_d x_k + B_d u_k + w_k,$$

where $A_d \in \mathbb{R}^{n\times n}$, with $n = 2N$, $B_d \in \mathbb{R}^n$, and $w_k \in \mathbb{R}^n$ is the process noise caused by the discretization. And the output by

$$y_k = C_d x_k + v_k,$$

where $v_k \in \mathbb{R}^l$ is the measurement noise. The process and measurement noise are assumed to be uncorrelated white noises.

In order to apply the SLKF to this problem, first we define the region where we want the state estimation to be focused on. Hence, we take the node 25, this node is taken as the measurement point for the input as well as the outputs, velocity and position. Therefore we specify Γ_k such that the SLKF concentrates around this region. Assuming $M_k = I_n$, the weighting matrix Γ_k is constructed such that the entries of each column are taken from a Gaussian function with mean the position of the analyzed state. Note that for each analyzed state one column is needed, this makes the state estimation around the region of interest to be smooth, avoiding numerical problems in the model integration. To show why we have chosen a smooth function like the Gaussian to build up Γ_k, let us take two different Γ_k: one is chosen to be Gaussian as it was already mentioned and another one with column entries set to be one around the location of interest and zero otherwise. For this example $p_k = l_k = 2$, so π_k in the two cases looks as shown in Figures 3.12, and 3.13. Let us call them π_{k_1} and π_{k_2} respectively.

Analysis of Results

First of all, one experiment was carried out to compare the performance of the SLKF filter using π_{k_1} and π_{k_2}. In Figures 3.14 and 3.15 we observe that the estimations in the positions 2, 10, and 25 are smooth in all the cases; however, in the position 15 we see that the estimation of the SLKF using π_{k_2} is very

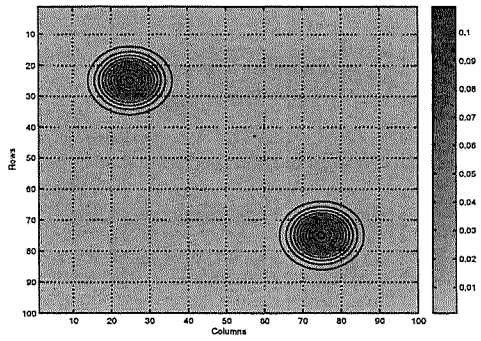


Figure 3.12: Orthogonal projector matrix π_{k_1}, Gaussian function

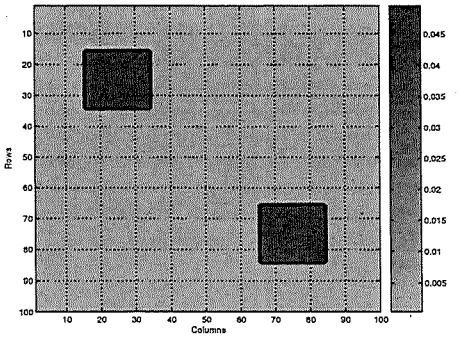


Figure 3.13: Orthogonal projector matrix π_{k_2}, no Gaussian function

noisy. As a matter of fact, we expected this result because of the function we chose, i.e., the position 15 is exactly located where π_{k_2} goes from 0.05 to 0, as a result of this abrupt change in the weighting function π_k the estimations of the SLKF filter become noisy. This effect is more clear in the velocity than in the position because there is a differentiation step involved in its calculation.

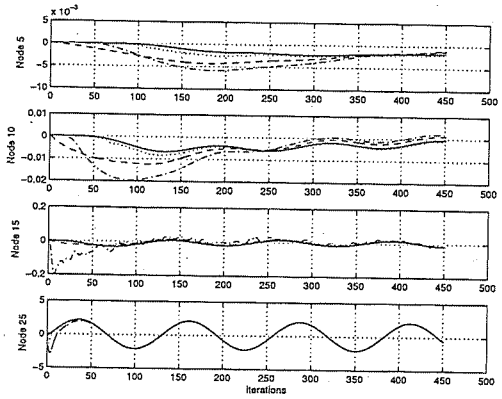


Figure 3.14: Estimation of velocity with a SNR=6dB. Solid line, original state, dotted line classical KF estimation, and dashed line SLKF estimation using π_{k_1}, and dash-dotted line SLKF estimation using π_{k_2}. At the node 15 is clearly seen that the choice of a non-smooth weighting function may generate instabilities in the numerical model.

Next, two experiments with different Signal-to-Noise ratio (SNR) were carried out, the first one with 6dB and the second with 1dB.

Figures 3.16 and 3.19 show a comparison of the performance of the SLKF to the classical KF at different locations, specifically, 2, 10, 15, and 25. Generally speaking, it can be seen that under high SNR conditions SLKF performs similar to KF for any node, whereas for low SNR conditions the performance of the SLKF is worst in the nodes far from the measurement point and similar in the nodes around it. This is due to the spatially localized strategy used in the SLKF.

In more detail, Figures 3.16 and 3.17 compare the estimation of the velocity and position of the SLKF to the classical KF for the case SNR=6dB. In both figures it is clearly seen that the estimation of the SLKF and KF at the node

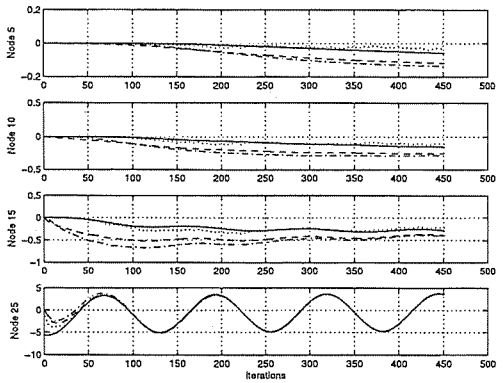


Figure 3.15: Estimation of position with a SNR=6dB. Solid line, original state, dotted line classical KF estimation, and dashed line SLKF estimation using π_{k_1}, and dash-dotted line SLKF estimation using π_{k_2}.

25 and 15 is good as expected, because they are located inside or very close to the region defined by π_k. On the other hand, while we go far from this region, at nodes 10 and 2, even though the estimation starts to be poor, it is still good.

Figures 3.18 and 3.19 compare the estimation of the velocity and position of the SLKF to the classical KF for the case SNR=1dB. Again, both filters work well at the node 25, but SLKF reduces its performance compared to the previous case in the other places, while the KF keeps its performance. As a matter of fact, the classical KF is expected to do better than SLKF in regions apart from the measurement point because it is not restricted to a certain region, whereas the SLKF satisfies the localization constraint.

Summarizing, we can say that the estimation of the SLKF and the classical KF are the same under high SNR conditions for all the states, but under low SNR conditions, the estimation of the SLKF is reliable just into the interested region.

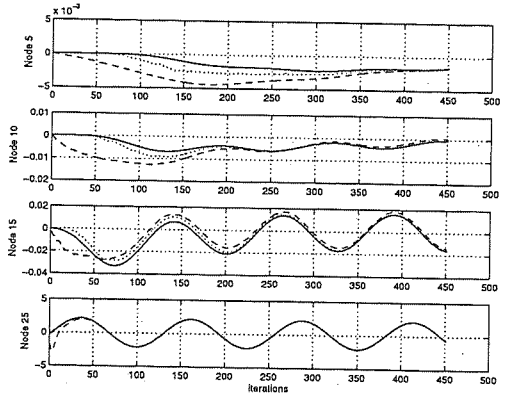


Figure 3.16: Estimation of velocity with a SNR=6dB. Solid line original state, dotted line the classical KF estimation, and dashed line the SLKF estimation. As expected, the SLKF performs similar to the KF at the node where the measurement point is located, and different at the other nodes.

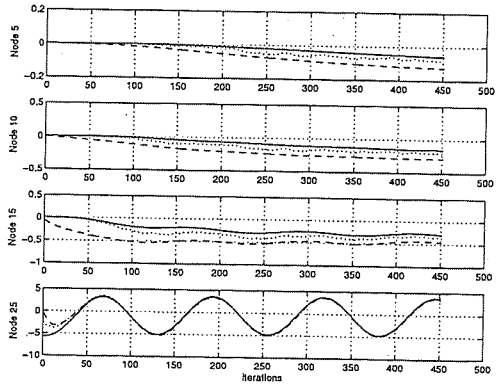


Figure 3.17: Estimation of position with a SNR=6dB. Solid line original state, dotted line the classical KF estimation, and dashed line the SLKF estimation. As expected, the SLKF performs similar to the KF at the node where the measurement point is located, and different at the other nodes.

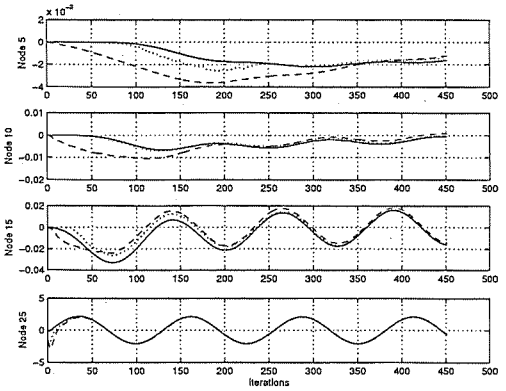


Figure 3.18: Estimation of velocity with a SNR=1dB. Solid line, original state, dotted line the classical KF estimation, and dashed line the SLKF estimation. As expected, the SLKF performs similar to the KF at the node where the measurement point is located, and different at the other nodes.

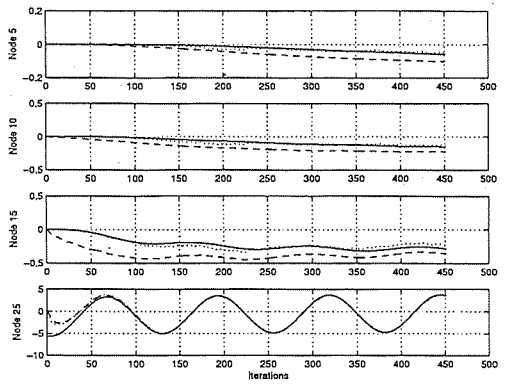


Figure 3.19: Estimation of position with a SNR=1dB. Solid line, original state, dotted line the classical KF estimation, and dashed line the SLKF estimation. As expected, the SLKF performs similar to the KF at the node where the measurement point is located, and different at the other nodes.

3.5 The Ensemble Kalman Filter - EnKF

The EnKF was first introduced by Evensen in 1994 [33], however, a more clear presentation and review can be found in [34]. Since then it has gained popularity because of its simple conceptual formulation and relative ease of implementation, e.g., in the nonlinear case it requires no derivation of a tangent linear operator or adjoint equations and no integrations backward in time, like in the extended Kalman filter §4.3. Further, computational requirements are affordable and comparable with other popular sophisticated assimilation methods as the one by Bennett, 1992 [14], and the 4DVAR method which has been much studied in the meteorological community (see [75, 26]). In this section we introduce the EnKF for the linear case and in the next chapter it will be extended to the nonlinear case.

The ensemble Kalman filter is a sequential data assimilation method where the error statistics are predicted by solving the Fokker-Planck equation (3.38), which describes the time evolution of a probability density function of a model state, using a Monte Carlo or ensemble integration. By integrating an ensemble of model states forward in time it is possible to calculate statistical moments like mean and error covariances whenever such information is required. Thus, all the statistical information about the predicted model state that is required at analysis times is contained in the ensemble.

The method is presented in three stages:

- Representation of error statistics
- Prediction of error statistics
- The estimation problem

3.5.1 Representation of Error Statistics

The error covariance matrices for the prior and the current estimate, $P_{k|k-1}$ and $P_{k|k}$, are defined in the Kalman filter in terms of the true state as

$$P_{k|k-1} = \mathcal{E}\left[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^{\mathrm{T}}\right] \tag{3.30}$$

$$P_{k|k} = \mathcal{E}\left[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^{\mathrm{T}}\right] \tag{3.31}$$

Now, for the EnKF assume that we have an ensemble of forecasted model states that randomly sample the model errors at time k. Let us denote this ensemble

as $X_{k|k-1} \in \mathbb{R}^{n\times N}$. It is defined by

$$X_{k|k-1} \triangleq (\hat{x}^1_{k|k-1}, \ldots, \hat{x}^N_{k|k-1}), \tag{3.32}$$

where the superscript denotes the ensemble member, with N the number of ensemble members. Then, the ensemble mean $\bar{\hat{x}}_{k|k-1} \in \mathbb{R}^n$ is defined by

$$\bar{\hat{x}}_{k|k-1} \triangleq \frac{1}{N}\sum_{i=1}^{N} \hat{x}^j_{k|k-1}.$$

Since the true state x_k is not known, and in order to write (3.30) and (3.31) in terms of (3.32), we therefore define the ensemble covariance matrices around the ensemble mean as follows; define the ensemble of prior estimation errors $E_{k|k-1} \in \mathbb{R}^{n\times N}$ by

$$E_{k|k-1} \triangleq (\hat{x}^1_{k|k-1} - \bar{\hat{x}}_{k|k-1}, \ldots, \hat{x}^N_{k|k-1} - \bar{\hat{x}}_{k|k-1}), \tag{3.33}$$

and the ensemble of estimation errors $E_{k|k} \in \mathbb{R}^{n\times N}$ by

$$E_{k|k} \triangleq (\hat{x}^1_{k|k} - \bar{\hat{x}}_{k|k}, \ldots, \hat{x}^N_{k|k} - \bar{\hat{x}}_{k|k}). \tag{3.34}$$

Hence,

$$P_{k|k-1} \approx \hat{P}_{k|k-1} \triangleq \frac{1}{N-1} E_{k|k-1} E^{\mathrm{T}}_{k|k-1}, \tag{3.35}$$

$$P_{k|k} \approx \hat{P}_{k|k} \triangleq \frac{1}{N-1} E_{k|k} E^{\mathrm{T}}_{k|k}, \tag{3.36}$$

which are averages over the ensembles. Thus, we can use an interpretation where the ensemble mean is the best estimate and the spreading of the ensemble around the mean is a natural definition of the error in the ensemble mean. Since the error covariances defined in (3.35) and (3.36) are defined as ensemble averages, there will clearly exist infinitively many ensembles with an error covariance equal to $\hat{P}_{k|k-1}$ and $\hat{P}_{k|k}$. Thus, instead of storing a full covariance matrix, we can represent the same error statistics using an appropriate ensemble of model states. Given an error covariance matrix, an ensemble of finite size will always provide an approximation to the error covariance matrix. However, when the size of the ensemble N increases the errors in the Monte Carlo sampling will decrease proportional to $1/\sqrt{N}$.

Suppose now that we have N model states in the ensemble, each of dimension n. Each of these model states can be represented as a single point in an N-dimensional state space. All the ensemble members together will constitute a

cloud of points in the state space. Such a cloud of points in the state space can, in the limit when N goes to infinity, be described using a probability density function

$$p(x) = \frac{dN}{N}, \tag{3.37}$$

where dN is the number of points in a small unit volume and N is the total number of points. With knowledge about either $p(x)$ or the ensemble representing $p(x)$, we can calculate whichever statistical moments (such as mean, covariances etc.) we want whenever they are needed.

The conclusion so far is that the information contained by a full probability density function can be exactly represented by an infinite ensemble of model states.

3.5.2 Prediction of Error Statistics

Conceptually, the evolution of (3.37) can be modelled with the Fokker-Planck equation

$$\frac{\partial p(x)}{\partial t} = -\sum_i \frac{\partial (A_i p(x))}{\partial x_i} + \frac{1}{2}\sum_{i,j} \frac{\partial^2 \left[p(x)(GQG^{\mathrm{T}})_{ij}\right]}{\partial x_i \partial x_j}, \tag{3.38}$$

where A_i is the component number i of the model operator A, and Q is the covariance matrix for the model errors. This equation does not apply any important approximations and can be considered as the fundamental equation for the time evolution of error statistics. A detailed derivation is given in [60]. The equation describes the change of the probability density in a local volume which is dependent on the divergence term describing a probability flux into the local volume (impact of the dynamical equation) and the diffusion term which tends to flatten the probability density due to the effect of stochastic model errors. If (3.38) could be solved for the probability density function, it would be possible to calculate statistical moments like the mean state and the error covariance for the model forecast to be used in the analysis scheme. The EnKF applies a so called Markov Chain Monte Carlo (MCMC) method to solve (3.38). The probability density can be represented using a large ensemble of model states. By integrating these model states forward in time according to the model dynamics described by (2.1), this ensemble prediction is equivalent to solving the Fokker-Planck equation using a MCMC method. This procedure forms the backbone for the EnKF.

3.5.3 The Estimation Problem

In the standard KF estimation problem the definition of $P_{k|k-1}$ and $P_{k|k}$ are used. We now give a derivation of the estimation problem for the EnKF using (3.35) and (3.36).

For the estimated-error covariance update $P_{k|k}$, the EnKF performs an ensemble of parallel data assimilation cycles, using (2.3) as follows,

for $i = 1, \ldots, N$

$$\hat{x}^i_{k|k} = \hat{x}^i_{k|k-1} + K_{e_k}\left(y^i_k - C_k \hat{x}^i_{k|k-1}\right), \tag{3.39}$$

where C_k is the observation operator, and the observations $y^i_k = y_k + \epsilon^i$ are perturbed observations defined such that $\epsilon^i \sim \mathcal{N}(0, R_{e_k})$. In the limit of an infinite ensemble the matrix R_{e_k} will converge toward the prescribed error covariance matrix R_k used in the standard KF. As a result

$$\hat{P}_{k|k} = \frac{1}{N-1} E_{k|k} E^{\mathrm{T}}_{k|k}.$$

Then it follows, similar to the standard KF (3.39), the ensemble Kalman gain $K_{e_k} \in \mathbb{R}^{n \times m}$ is computed as

$$K_{e_k} = \hat{P}_{k|k-1} C^{\mathrm{T}}_k \tilde{R}^{-1}_{e_k}, \tag{3.40}$$

where $\tilde{R}^{-1}_{e_k} \in \mathbb{R}^{m \times m}$ is defined by

$$\tilde{R}_{e_k} \triangleq C_k \hat{P}_{k|k-1} C^{\mathrm{T}}_k + R_{e_k}. \tag{3.41}$$

After the estimation is done, a short-range forecast is computed by running the physics first principle based model (2.1) until new measurements are taken, then the data assimilation is repeated.

The Initial Ensemble [34]

The initial ensemble should ideally be chosen to properly represent the error statistics of the initial guess for the model state. However, a modest misspecification of the initial ensemble normally does not influence the results very much over time. The rule of thumb seems to be that one needs to create an ensemble of model states by adding some kind of perturbations to a best-guess estimate, and then integrate the ensemble over a time interval covering

a few characteristic time scales of the dynamical system. This will ensure that the system is in dynamical balance and that proper multivariate correlations have developed.

The perturbations can be created in different ways. The simplest is to sample random numbers (for a scalar model), random curves (for a 1-D model) or random fields (for a model with 2 or higher dimensions, see appendix C), from a specified distribution.

The algorithm will be as follows,

Algorithm 11 *Linear EnKF Estimator*

First, generate an initial ensemble $X_{k-1|k-1}$ which properly represent the error statistics of the initial guess for the model state, see appendix D for details.

for $k = 1, \ldots,$

1. *Update the ensemble members of $X_{k-1|k-1}$ using (2.1) for $i = 1, \ldots, N$ according to*

$$x^i_{k|k-1} = A_k x^i_{k-1|k-1} + B_{k-1} u_{k-1}$$

2. *Compute $\hat{P}_{k|k-1}$ using (3.35).*
3. *Compute the EnKF gain using (3.40)*
4. *Update the ensemble members using (3.39)*
5. *Compute the state estimation taking the mean of the updated ensemble members*

$$\hat{x}_{k|k} \triangleq \frac{1}{N} \sum_{i=1}^{N} \hat{x}^i_{k|k}.$$

3.6 Computational Complexity of the Algorithms

All the suboptimal Kalman filters presented in this chapter are $\mathcal{O}(n^2)$ except for the SLKF which is $\mathcal{O}(n^3)$. In Table 3.2 we observe that the RRSQRT-KF and the SSQRT-KF are the most efficient ones in terms of number of flops, nevertheless the RRSQRT-KF does not have any restriction as the SSQRT-KF. On the other hand, the SQRT-SLKF has the same computational complexity as the RRSQRT-KF when $m \ll q$, which is normally the case for which the SLKF has been designed. This computational complexity can be reduced even more if we exploit the fact that SLKF makes the error covariance matrix very sparse.

RRSQRT-KF	$2n^2(q+1) + 2n(q^2 + 7/2q + l^2 + 2m + p/2 + 1/2)$
SSQRT-KF	$2n^2(q+3/2) + 2n(q^2 + 2qm + m^2 + 2m + p/2 + 1/2)$
SLKF	$2n^3 + 2n^2(11m + 3/2) + 4n(2m^2 + m + l^2 + p/2 + 1/2)$
SQRT-SLKF	$2n^2(3m + q + 1) + 2n(2q^2 + m^2 + l^2 + 5q + ql + p + 1/2)$
Linear EnKF	$4n^2(N + m + 1/4) + 2n(Np + 2Nm + 5N/2 + 2m^2 + 1/2)$

Table 3.1: Number of flops

RRSQRT-KF	$\mathcal{O}(n^2 q)$
SSQRT-KF	$\mathcal{O}(n^2 q)$
SLKF	$\mathcal{O}(n^3)$
SQRT-SLKF	$\mathcal{O}(n^2(m+q))$
Linear EnKF	$\mathcal{O}(n^2(m+N))$

Table 3.2: Order of computation

Where, n is the order of the system, m the number of observations, p the number of inputs, l the order of the process-noise covariance matrix Q, q the rank decomposition of $S_{k|k-1}$ for the RRSQRT-KF, and the number of columns of $S_{k|k-1}$ for the SSQRT-KF, and N the number of ensemble members in the EnKF. Notice that N is equivalent in computational cost to q. Furthermore, the RRSQRT-KF is not sensible to the number of observations m as the other filters whose computationally complexity can be raised to $\mathcal{O}(n^3)$ if $m = n$. On the other hand, the SSQRT-KF is not sensitive to the order of Q as the other filters whose computationally complexity can be raised to $\mathcal{O}(n^3)$ if $l = n$.

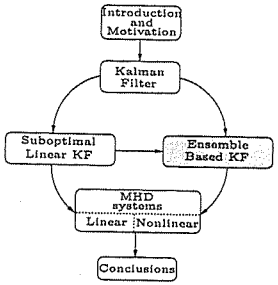
3.7 Conclusions

Suboptimal Kalman filters have shown to be a useful tool for large scale data assimilation, resulting in the flourish of several suboptimal linear and nonlinear Kalman filter formulations like the extended Kalman filter [60, 4], the ensemble Kalman filter [33, 34, 35, 82], the reduced rank square root Kalman filter [87, 53], the partial eigen-decomposition Kalman filter [25], etc. Hence, in this chapter we have presented two of the most relevant suboptimal linear Kalman filters we find in the literature, namely, the RRSQRT-KF, and the EnKF. Also, we have proposed two new suboptimal Kalman filters for two different scenarios: the first one was the SSQRT-KF that is suitable for the case when the process noise can be assumed either to be zero or negligible. As a result, by making this assumption we have shown that the complexity in the computation of the Kalman filter can be reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$. Even though the SSQRT-KF was initially obtained for systems with negligible process noise, it can be used to systems with process noise by applying the covariance inflation approach described in §3.3.1 and [13, 6]. Though the results showed a good performance compared to RRSQRT-KF and the full Kalman filter, we have to stress that the solution obtained using the covariance inflation factor is not optimal.

The second approach was the SLKF [9], which can be used when there are very few observations compared to the order of the system, hence, just local estimations can be made. Moreover, it is also required that this local estimations have to be done using the full prior-error covariance matrix $P_{k|k-1}$. As a result, the SLKF we introduced in this chapter is able to make local estimations using the full $P_{k|k-1}$ without affecting numerically regions far apart from the observation points. This could be done by choosing an appropriate local weighting matrix Γ_k and error weighting matrix L_k. On the other hand, despite the fact that in the SLKF the error-covariance matrices obtained are very sparse, its computation complexity might be of order $\mathcal{O}(n^3)$ which could make it not attractive for real applications; therefore, we presented the square root version of the SLKF that we called the SQRT-SLKF, which can be computed with a complexity of order $\mathcal{O}(n^2)$, and still exploit the sparsity of the error-covariance matrices.

Chapter 4

Ensemble-Based Kalman Filter for Nonlinear Large Scale Systems



4.1 Introduction

In the previous chapters we have expended some time introducing the Kalman filter and some suboptimal formulations of it. This was motivated by the fact that the Kalman filter is prohibitive expensive to use for estimation of large scale systems which is the main subject in this thesis. Another important issue in this thesis is that the systems we want to focus on are environmental; therefore, we expect these systems to be nonlinear in most of the cases. Because of this, in this chapter we investigate and extend the results of the previous

chapters to their nonlinear formulations based on the ensemble Kalman filter.

4.2 Stochastic Model Description

Complex environmental processes can be simulated effectively by systems of discretized partial differential equations describing the conservation laws. The discretized system of equations provides a reasonably accurate forecast of the future dynamical behaviour, provided that the initial state of the system and the boundary conditions are known. However, for real applications the complete information defining the state of the system at a specific time is rarely available. Moreover, both the models and the measured data contain inaccuracies and random noise. Hence, a nonlinear model with uncertainties can written as a discrete-time stochastic differential equation as follows,

$$x_{k+1} = \mathcal{A}(x_k, k) + G_k \eta_k \tag{4.1}$$

where $\eta_k \in \mathbb{R}^l$ is a Gaussian system noise process, and $G_k \in \mathbb{R}^{n \times l}$. Hence, the covariance matrix $Q_k \in \mathbb{R}^{n \times n}$ is defined as

$$Q_k \triangleq \mathcal{E}[\eta_k \eta_k^{\mathrm{T}}].$$

Then it follows that the output can be defined as

$$y_k = \mathcal{C}(x_k, k) + \nu_k, \tag{4.2}$$

where ν_k the Gaussian measurement noise process with zero mean and known covariance matrix R_k. The process noise and the measurement noise are assumed to be uncorrelated.

Equation (4.1) implies that even if the initial state is known precisely, future model states cannot since unknown random model errors are continually added.

Based on this model description we introduce in the next sections some classical nonlinear Kalman filtering approaches and the nonlinear extension of the ones we introduced in Chapter 3.

4.3 The Extended Kalman Filter –EKF

The extended Kalman filter approach [60, 45] consist of applying the standard Kalman filter (for linear systems) to nonlinear systems with additive white

noise by continually updating a linearization around the previous state estimate, starting with an initial guess. In other words, we only consider a linear Taylor approximation of the system function at the previous state estimate and that of the observation function at the corresponding predicted position. This approach gives a simple and efficient algorithm to handle a nonlinear model. However, convergence to a reasonable estimate may not be obtained if the initial guess is poor or if the disturbances are so large that the linearization is inadequate to describe the system.

Here it is important to highlight that when we deal with large scale systems the EKF becomes very complex to compute due to the linearization that has to be done each time iteration. We use the EKF in this thesis just for comparison purposes to ensemble-based Kalman filter approaches that we will discuss later in this chapter.

Numerical Formulation

Consider the nonlinear system described by (4.1) and (4.2). Accordingly to the EKF, it is assumed that the mean state evolves according to

$$\hat{x}_{k|k-1} = \mathcal{A}(\hat{x}_{k-1|k-1}, k-1).$$

Next, if we assume the difference between $\hat{x}_{k|k-1}$ and x_k small, then the evolution of this difference should be approximately linear, so

$$\mathcal{A}(\hat{x}_{k-1|k-1}) - \mathcal{A}(x_{k-1}) \simeq A_{k-1}(\hat{x}_{k-1|k-1} - x_{k-1}),$$

where $A_{k-1} \in \mathbb{R}^{n\times n}$ is the Jacobian matrix of $\mathcal{A}$ at time $k-1$ defined by

$$A_{k-1} \triangleq \left.\frac{\partial \mathcal{A}}{\partial x}\right|_{x=\hat{x}_{k-1|k-1}}.$$

Furthermore, if we presume again the difference between $\hat{x}_{k|k-1}$ and x_k small enough then the innovation vector can be approximated as

$$\begin{aligned} y_k - \hat{y}_{k|k-1} &= \mathcal{C}(x_k) - \mathcal{C}(\hat{x}_{k|k-1}) - \nu_k \\ &\simeq C_k(x_k - \hat{x}_{k|k-1}) - \nu_k, \end{aligned} \tag{4.3}$$

where $C_k \in \mathbb{R}^{m\times n}$ is the Jacobian of $\mathcal{C}$ at time k defined by

$$C_k \triangleq \left.\frac{\partial \mathcal{C}}{\partial x}\right|_{x=\hat{x}_{k|k-1}}.$$

In addition, note that the update equation is as follows

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\big(y_k - \mathcal{C}(\hat{x}_{k|k-1})\big),$$

where

$$\begin{aligned} K_k &= P_{k|k-1}C_k^{\mathrm{T}}(C_k P_{k|k-1} C_k^{\mathrm{T}} + R_k)^{-1}, \\ P_{k|k-1} &= A_{k-1}P_{k-1|k-1}A_{k-1}^{\mathrm{T}} + Q_{k-1}. \end{aligned}$$

Given these assumptions, an alternate form of the Kalman filter can be derived as shown on the next page.

Algorithm 12 *EKF Estimator*

Given R_k, Q_k, $\hat{x}_0$, *and* P_0,

for $k = 1, \ldots,$

1. *Update* $\hat{x}_{k-1|k-1}$ *to* $\hat{x}_{k|k-1}$

$$\hat{x}_{k|k-1} = \mathcal{A}(\hat{x}_{k|k-1}, k-1)$$

2. *Compute* $\hat{y}_{k|k-1}$

$$\hat{y}_{k|k-1} = \mathcal{C}(\hat{x}_{k|k-1}, k-1)$$

3. *Compute the Jacobian matrices* A_{k-1} *and* C_k

$$A_{k-1} = \left.\frac{\partial \mathcal{A}}{\partial x}\right|_{x=\hat{x}_{k-1|k-1}}$$

$$C_k = \left.\frac{\partial \mathcal{C}}{\partial x}\right|_{x=\hat{x}_{k|k-1}}.$$

4. *Update* $P_{k-1|k-1}$ *to* $P_{k|k-1}$

$$P_{k|k-1} = A_{k-1} P_{k-1|k-1} A_{k-1}^{\mathrm{T}} + Q_{k-1}.$$

5. *Compute the extended Kalman gain*

$$K_k = P_{k|k-1} C_k^{\mathrm{T}} (C_k P_{k|k-1} C_k^{\mathrm{T}} + R_k)^{-1}$$

6. *Update* $x_{k|k-1}$ *to* $x_{k|k}$, *and* $P_{k|k-1}$ *to* $P_{k|k}$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\big(y_k - \mathcal{C}(\hat{x}_{k|k-1})\big)$$
$$P_{k|k} = (I_n - K_k C_k) P_{k|k-1}$$

end

4.4 The Ensemble Kalman Filter –EnKF

In this section the EnKF introduced in §3.5 is extended to the nonlinear case. The EnKF was designed to resolve two major problems related to the use of the extended Kalman filter (EKF) with nonlinear dynamics in large state spaces. The first problem relates to the use of an approximate closure scheme in the EKF, and the other to the huge computational requirements associated with the storage and forward integration of the error covariance matrix. The EKF applies a closure scheme where third- and higher order moments in the error covariance equation are discarded. This linearization has been shown to be invalid in a number of applications. In fact, the equation is not longer the fundamental equation for the error evolution when the dynamical model is nonlinear.

Consider the nonlinear system described by (4.1) and (4.2), Similar to the linear case the nonlinear EnKF performs an ensemble of parallel data assimilation cycles using (2.3) to update the estimated-error covariance matrix $\hat{P}_{k|k}$ as follows,

for $i = 1, \ldots, N$

$$\hat{x}^i_{k|k} = \hat{x}^i_{k|k-1} + K_{e_k}\big(y^i_k - \mathcal{C}(\hat{x}^i_{k|k-1})\big), \tag{4.4}$$

where $\mathcal{C}$ is the observation operator, which is permitted to be a nonlinear operator, and the observations $y^i_k = y_k + \epsilon^i$ are perturbed observations defined such that $\epsilon^i \sim \mathcal{N}(0, R_e)$. In the limit of an infinite ensemble the matrix R_e will converge toward the prescribed error covariance matrix R used in the standard KF. Moreover, $K_{e_k} \in \mathbb{R}^{n \times m}$ is computed as in the classical Kalman filter as

$$K_{e_k} = \hat{P}_{k|k-1} \mathcal{C}^{\mathrm{T}} \tilde{R}^{-1}_{e_k}, \tag{4.5}$$

where $\tilde{R}^{-1}_{e_k} \in \mathbb{R}^{m \times m}$ is defined by

$$\tilde{R}_{e_k} \triangleq \mathcal{C} \hat{P}_{k|k-1} \mathcal{C}^{\mathrm{T}} + R_{e_k},$$

with the difference that $\mathcal{C}$ can be nonlinear which is a powerful advantage compared to other non-linear KF that are based on linearized models like EKF. Envision a situation where errors grow rapidly but saturate at low amplitude; the linear assumption of error growth in the EKF will result in an overestimate of the prior error variance, but the differences among ensemble members will not grow without bound and thus should provide a more accurate model of the actual prior error statistics.

On the other hand, for a complex model with a high-dimensional state vector, explicitly forming $\hat{P}_{k|k-1}$ as in (3.35) would be computationally prohibitive. However, in the EnKF, K_{e_k} can be formed without ever explicitly computing the full $\hat{P}_{k|k-1}$. Instead, the components of $\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}}$ and $\mathcal{C}\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}}$ of K_{e_k} are computed separately as follows, define $\overline{\mathcal{C}(\hat{x}_{k|k-1})} \in \mathbb{R}^{m\times n}$ by

$$\overline{\mathcal{C}(\hat{x}_{k|k-1})} \triangleq \frac{1}{N}\sum_{i=1}^{N}\mathcal{C}(\hat{x}^i_{k|k-1}), \tag{4.6}$$

which represents the mean of the estimate of the observations interpolated from the background forecasts. Then $\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}} \in \mathbb{R}^{n\times m}$ by

$$\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}} \triangleq \frac{1}{N-1}\sum_{i=1}^{N}\left(\hat{x}^i_{k|k-1} - \bar{\hat{x}}_{k|k-1}\right)\left(\mathcal{C}(\hat{x}^i_{k|k-1}) - \overline{\mathcal{C}(\hat{x}_{k|k-1})}\right)^{\mathrm{T}},$$

and $\mathcal{C}\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}} \in \mathbb{R}^{m\times m}$ by

$$\mathcal{C}\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}} \triangleq \frac{1}{N-1}\sum_{i=1}^{N}\left(\mathcal{C}(\hat{x}^i_{k|k-1}) - \overline{\mathcal{C}(\hat{x}_{k|k-1})}\right)\left(\mathcal{C}(\hat{x}^i_{k|k-1}) - \overline{\mathcal{C}(\hat{x}_{k|k-1})}\right)^{\mathrm{T}}.$$

4.4.1 Error Covariance Propagation

Ensemble-based assimilation methods leverage a potential increase in accuracy that may result from estimating covariances from an ensemble propagated with the fully nonlinear forecast model. If forecast-error dynamics are in fact quite nonlinear and saturate quickly, then the assumption of linearity in the Kalman filter and extended Kalman filter was inappropriate, and some accuracy may be gained relative to the Kalman filters by estimating covariances from a sample of fully nonlinear model forecasts.

However, in real-world applications, prior-error covariances cannot simply be estimated at the next assimilation cycle by conducting an ensemble of deterministic forecasts forward from the current cycle's estimations. Because of model deficiencies, even if the true state of the system is perfectly known, the resulting forecast will be imperfect, see equation (4.1). The assumption of no bias is probably not justified, and if the bias can be determined, the forecasts ought to be corrected for this bias [29], or even more, the forecast model ought to be improved. In any case, consider the error covariance at the next assimilation time. Assume again that prior error due to initial conditions uncertainty and model error are uncorrelated

$$\mathcal{E}\left[\left(\mathcal{A}(x_{k-1}) - \mathcal{A}(\bar{\hat{x}}_{k-1|k-1})\right)\eta^{\mathrm{T}}_{k-1}\right] = 0, \tag{4.7}$$

and assume linearity of the error growth

$$\mathcal{A}(x_{k-1}) - \mathcal{A}(\bar{\hat{x}}_{k-1|k-1}) \simeq A_{k-1}(x_{k-1} - \bar{\hat{x}}_{k-1|k-1}).$$

Then the true prior-error covariance at the next assimilation time is

$$\begin{aligned}
P_{k|k-1} &= \mathcal{E}\left[(\bar{\hat{x}}_{k|k-1} - x_k)(\bar{\hat{x}}_{k|k-1} - x_k)^{\mathrm{T}}\right] \\
&= \mathcal{E}\left[\left(\mathcal{A}(\bar{\hat{x}}_{k-1|k-1}) - \mathcal{A}(x_{k-1}) - \eta_{k-1}\right)\left(\mathcal{A}(\bar{\hat{x}}_{k-1|k-1}) - \mathcal{A}(x_{k-1}) - \eta_{k-1}\right)^{\mathrm{T}}\right] \\
&\simeq \mathcal{E}\left[A_{k-1}(\bar{\hat{x}}_{k-1|k-1} - x_{k-1})(\bar{\hat{x}}_{k-1|k-1} - x_{k-1})^{\mathrm{T}}A^{\mathrm{T}}_{k-1}\right] + \mathcal{E}\left[\eta_{k-1}\eta^{\mathrm{T}}_{k-1}\right] \\
&= A_{k-1}P_{k-1|k-1}A^{\mathrm{T}}_{k-1} + Q_{k-1},
\end{aligned} \tag{4.8}$$

where A_{k-1} is the Jacobian of the nonlinear operator $\mathcal{A}$ at time $k-1$. Consider what happens when the covariances are estimated directly from an ensemble of forecasts propagated forward from an ensemble of $i = 1, \ldots, N$ estimations using the fully nonlinear forecast model

$$\hat{x}^i_{k|k-1} = \mathcal{A}(\hat{x}^i_{k-1|k-1}). \tag{4.9}$$

Calculating the expected covariance, we get

$$\begin{aligned}
\hat{P}_{k|k-1} &= \mathcal{E}\left[\left(\hat{x}^i_{k|k-1} - \bar{\hat{x}}_{k|k-1}\right)\left(\hat{x}^i_{k|k-1} - \bar{\hat{x}}_{k|k-1}\right)^{\mathrm{T}}\right] \\
&= \mathcal{E}\left[\left(\mathcal{A}(\hat{x}^i_{k-1|k-1}) - \mathcal{A}(\bar{\hat{x}}_{k-1|k-1})\right)\left(\mathcal{A}(\hat{x}^i_{k-1|k-1}) - \mathcal{A}(\bar{\hat{x}}_{k-1|k-1})\right)^{\mathrm{T}}\right] \\
&\simeq \mathcal{E}\left[A_{k-1}(\hat{x}^i_{k-1|k-1} - \bar{\hat{x}}_{k-1|k-1})(\hat{x}^i_{k-1|k-1} - \bar{\hat{x}}_{k-1|k-1})^{\mathrm{T}}A^{\mathrm{T}}_{k-1}\right] \\
&= A_{k-1}\hat{P}_{k-1|k-1}A^{\mathrm{T}}_{k-1}.
\end{aligned} \tag{4.10}$$

Comparing (4.8) and (4.10), it is apparent that an ensemble of estimations that are simply propagated forward with the nonlinear forecast model will have too small an expected amount of spread, missing the extra model-error covariance Q_{k-1}. Let us define some hypothetical set of prior forecasts at instant $k-1$ that do have the correct covariance, i.e, define $\breve{x}^i_{k|k-1}$ such that

$$\mathcal{E}\left[(\breve{x}^i_{k|k-1} - \bar{\breve{x}}_{k|k-1})(\breve{x}^i_{k|k-1} - \bar{\breve{x}}_{k|k-1})^{\mathrm{T}}\right] = A_{k-1}\hat{P}_{k-1|k-1}A^{\mathrm{T}}_{k-1} + Q_{k-1}.$$

Such an ensemble is possible if we add noise to our existing ensemble as follows

$$\breve{x}^i_{k|k-1} \triangleq \hat{x}^i_{k|k-1} + \varepsilon^i,$$

where $\mathcal{E}[\varepsilon^i\varepsilon^{i\mathrm{T}}] = Q_{k-1}$, $\mathcal{E}[\varepsilon^i] = 0$ and $\mathcal{E}[\hat{x}^i_{k-1|k-1}\varepsilon^i] = 0$.

Several methods have been proposed for incorporating noise into the ensemble of forecasts so that they account for model errors. First, the forecast model could be stochastic-dynamic instead of deterministic, with terms to the prognostic equations to represent interactions with unresolved scales and/or misparameterized effects; in essence, $\mathcal{A}$ is changed so that the ensemble of forecasts integrates random noise in addition to the deterministic forecast dynamics. Over an assimilation cycle, this added to the ensemble as a result of integrating noise should be designed to increase the covariance by the missing Q_{k-1}. Another possibility is that one may choose to run a forecast model without integrating noise but to add noise to each member at the data assimilation time so as to increase the ensemble variance appropriate to the missing Q_{k-1}. Third, it may be possible to use a multi-model ensemble to estimate covariances.

Little work has yet been done on the first of these three approaches. Buizza et al. [21] demonstrated a simple technique for integrating noise to account for deterministic sub-gridscale parameterizations. Under their methodology, the parameterized terms in the prognostic equations were multiplied by a random number. The method was quite heuristic but appears to have increased the spread in the ensemble forecasts somewhat and increased their skill. Penland [71] outlines a more general approach for integrating system noise in numerical models. To date, however, a comprehensive noise integration scheme has not yet been demonstrated in an operational large scale prediction model.

The second general approach is to augment the ensemble-estimated model of covariances during the update step with noise representing the missing model error covariances. In [67] such an approach is described.

The third approach, use of multiple forecast models for generating the ensemble of background forecasts (e.g., [56, 52, 32, 91, 73, 58]), is appealing for its simplicity. A wider range of forecasts is typically generated when different weather forecast models are used to forecast the evolution of different ensemble members. Unfortunately, initial experimentation has shown that the multi-model ensembles tend to produce unrealistic estimates of error covariances. Forecast errors ought to be mostly balanced, but when estimated from multi-model ensembles, preliminary results suggest that the errors are excessively out of balance, with detrimental effects on the subsequent assimilation. See also [51] for a discussion of the use of multi-model ensembles in data assimilation in a simple model.

A last approach, discussed in [6] and demonstrated in [89] and [88], is to enlarge background error estimates by inflating forecast ensemble member's deviation about their mean by an amount κ (slightly greater than 1.0) before the first

observation is assimilated

$$\hat{x}^i_{k|k-1} \longleftarrow \kappa(\hat{x}^i_{k|k-1} - \bar{\hat{x}}_{k|k-1}) + \bar{\hat{x}}_{k|k-1}$$

Here, the operation $\longleftarrow$ denotes a replacement of the previous value of $\hat{x}^i_{k|k-1}$. This technique is called covariance inflation. Note that inflation effectively increases the spread of the ensemble, but it does not change the subspace spanned by the ensemble. Hence, if model error projects into a substantially different subspace, this parametrization may not be effective.

In section §3.3 we have introduced a similar approach for the linear case, the SSQRT-KF. There, it turned out that when applying the covariance inflation factor, the filter guarantees the convergence of the estimation by focusing on the most dominant modes of the error dynamics matrix.

4.4.2 Covariance Localization

In ensemble assimilation methods, the accuracy of error statistics is especially important. Unlike 3D-Var [1], the effects of a mis-specification of error statistics can affect the estimated error covariance, which is then propagated forward in time. Hence, if the estimated errors are underestimated in one cycle, the forecast errors may be underestimated in the following cycle, underweighting the new observations. The process can feed back on itself, the ensemble assimilation method progressively ignoring observational data more and more in successive cycles, leading eventually to a useless ensemble. This is known as filter divergence (e.g., [57, 82, 50]). For the ensemble-based methods, filter divergence can be induced by many causes. One of the most crucial is to model prior error covariance not realistically. As discussed in the previous section, an adequate parameterization of model error is likely to be very important to avoid divergence. However, filter divergence can occur even in simulations where the forecast model is perfect, for prior-error covariances are typically estimated imperfectly from small ensembles. While more ensemble members would be desirable to reduce the sampling error in estimating prior-error covariances, more members means more computational cost.

One common algorithmic modification to improve error covariance estimates from ensembles is covariance localization. The covariance estimate from the ensemble is multiplied point by point with a correlation function that is 1.0 at the observation location and decreases monotonically with increasing distance.

[1] 3D-Var is a variational data assimilation technique that performs the estimation using observations distributed in three dimensions without taking into account the time scale [70].

Mathematically, the Kalman gain equation $K_k = \hat{P}_{k|k-1}C\tilde{R}_{e_k}^{-1}$ is replaced by a modified gain

$$K_k = \rho_s \, o \, \hat{P}_{k|k-1}C\left[C(\rho_s \, o \, \hat{P}_{k|k-1})C^{\mathrm{T}} + R_{e_k}\right]^{-1}, \tag{4.11}$$

where the operation ρ_s in (4.11) denotes a Schur product (an element-by-element multiplication) of a correlation matrix Γ with the covariance model generated by the ensemble. The Schur product of matrices A and B is a matrix C of the same dimension, where $C_{ij} = A_{ij}B_{ij}$. As discussed in [50], covariance localization has the effect of making the prior-error covariance matrix higher in rank. When directly estimated from an N-member ensemble, the rank of $\hat{P}_{k|k-1}$ is $N-1$. With covariance localization, it can be much higher. When is used covariance localization with smaller ensembles, it can actually result in more accurate estimates than would be obtained from larger ensembles without localization (e.g., [58]). See [50] for more details on implementation of covariance localization, and see [68] for a discussion of balance issues related to the length scale of the covariance localization.

Even though this technique has shown to work quite well in systems with sparse measurement locations, the computation of the matrix ρ_s is very heuristic, so, we have to say that there is neither a proof of optimality nor any mathematical development showing how to compute the optimal ρ_s. On the contrary, in this thesis we have introduced a filter which is optimal in a quadratic sense and can be used as estimator in systems with sparse measurements, namely, the SLKF and EnSLKF, see sections §3.4 and §4.7 for details.

On the other hand, another drawback of the EnKF is that for a finite-sized ensemble, there is a sampling error in the estimation of background-error covariances (see [89]). In other words, $P_\infty \neq P_N$, where P_N is the error covariance matrix obtained from an N-member ensemble and P_∞ is the exact error covariance, defined to be that which would be obtained from an infinite ensemble. When observations are perturbed in the EnKF, there is also sampling error in the estimation of the measurement error covariance. In [89] is shown that if the same gain and the same observations are used to update each ensemble member in an ensemble data assimilation system, estimated error covariances will be systematically underestimated, and filter divergence may occur. To overcome this, an ensemble of perturbed observations can be assimilated, whose statistics reflect the known observation errors. In the limit of infinite ensemble size, in a system where all sources of error (both observation and model) are correctly sampled, this approach yields the correct estimated-error covariances. However, when ensemble sizes are finite, the noise added to the observations produces spurious error covariances associated with sampling errors in the estimation of the measurement error covariances. Hence, in the next section the Ensemble Square Root Filter approach (EnSRF) [89] is introduced. This filter does not require perturbing the observations, because the ensemble members are updated independently of the ensemble mean using the equation of the estimated-error covariance update of the square root Kalman filter. The algorithm is shown in Algorithm 13.

Algorithm 13 *EnKF Estimator*

First, generate an initial ensemble $X_{k-1|k-1}$ which properly represent the error statistics of the initial guess for the model state.

for $k = 1, \ldots$

1. *Update the ensemble $X_{k-1|k-1}$ to $X_{k|k-1}$ using (4.1) for $i = 1, \ldots, N$ according to*

$$x^i_{k|k-1} = \mathcal{A}(x^i_{k-1|k-1}, k-1) + \varepsilon^i_{k-1}.$$

2. *Compute $\overline{\mathcal{C}(\hat{x}_{k|k-1})}$*

$$\overline{\mathcal{C}(\hat{x}_{k|k-1})} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{C}(\hat{x}^i_{k|k-1}, k-1).$$

3. *Compute the $\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}}$*

$$\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}} = \frac{1}{N-1}\sum_{i=1}^{N} \left(\hat{x}^i_{k|k-1} - \bar{\hat{x}}_{k|k-1}\right)\left(\mathcal{C}(\hat{x}^i_{k|k-1}) - \overline{\mathcal{C}(\hat{x}_{k|k-1})}\right)^{\mathrm{T}}.$$

4. *Compute $\mathcal{C}\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}}$*

$$\mathcal{C}\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}} = \frac{1}{N-1}\sum_{i=1}^{N} \left(\mathcal{C}(\hat{x}^i_{k|k-1}) - \overline{\mathcal{C}(\hat{x}_{k|k-1})}\right)\left(\mathcal{C}(\hat{x}^i_{k|k-1}) - \overline{\mathcal{C}(\hat{x}_{k|k-1})}\right)^{\mathrm{T}}.$$

5. *Compute K_{e_k}*

$$K_{e_k} = \hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}}\left(\mathcal{C}\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}} + R_{e_k}\right).$$

6. *Update $X_{k|k-1}$ to $X_{k|k}$, for $i = 1, \ldots, N$*

$$x^i_{k|k} = x^i_{k|k-1} + K_{e_k}\left(y^i_k - \mathcal{C}(x^i_{k|k-1})\right).$$

7. *Compute the state estimation taking the mean of the updated ensemble members*

$$\hat{x}_{k|k} = \frac{1}{N}\sum_{i=1}^{N} \hat{x}^i_{k|k}.$$

end

4.5 Square Root Ensemble Filter –EnSRF

Several authors have pointed out that the perturbation of measurements used in the standard EnKF analysis equation, may be an additional source of sampling errors that reduces the estimated error covariance accuracy and increases the probability of underestimating the estimated error covariance [5, 89, 19, 78]. Hence, some methods based on the square root form of the Kalman filter for computing the estimated states without introducing measurement noise have recently been presented. In this section we will describe the EnSRF [89].

Let us start taking the prior and estimated error covariances $P_{k|k-1}$ and $P_{k|k}$ of a N-member ensemble given by (3.35) and (3.36) respectively. Therefore, a matrix square-root $\tilde{S}_{k|k-1} \in \mathbb{R}^{n\times N}$ of $P_{k|k-1}$ and $\tilde{S}_{k|k} \in \mathbb{R}^{n\times N}$ of $P_{k|k}$ can be defined by

$$\begin{aligned} \tilde{S}_{k|k-1} &\triangleq \frac{1}{(N-1)^{1/2}} E_{k|k-1}, \\ \tilde{S}_{k|k} &\triangleq \frac{1}{(N-1)^{1/2}} E_{k|k}, \end{aligned} \tag{4.12}$$

where $E_{k|k-1}$ and $E_{k|k}$ are defined by (3.33) and (3.34).

Then, rewriting (2.13) we have

$$\begin{aligned} P_{k|k} \simeq \hat{P}_{k|k} &= \tilde{S}_{k|k}\tilde{S}^{\mathrm{T}}_{k|k} \\ &= \left(I_n - \hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}}\tilde{R}^{-1}_{e_k}\mathcal{C}\right)\hat{P}_{k|k-1} \\ &= \tilde{S}_{k|k-1}\left(I_N - S^{\mathrm{T}}_{k|k-1}\mathcal{C}^{\mathrm{T}}\tilde{R}^{-1}_{e_k}\mathcal{C}\tilde{S}_{k|k-1}\right)\tilde{S}^{\mathrm{T}}_{k|k-1} \\ &= \tilde{S}_{k|k-1}\left(I_N - \tilde{\mathcal{F}}_k\tilde{R}^{-1}_{e_k}\tilde{\mathcal{F}}^{\mathrm{T}}_k\right)\tilde{S}^{\mathrm{T}}_{k|k-1}, \end{aligned}$$

where $\tilde{\mathcal{F}}_k \in \mathbb{R}^{N\times m}$ is defined as

$$\tilde{\mathcal{F}}_k \triangleq \left(\mathcal{C}\tilde{S}_{k|k-1}\right)^{\mathrm{T}}, \tag{4.13}$$

$$= \frac{1}{(N-1)^{1/2}}\left[\mathcal{C}(\hat{x}^1_{k|k-1}) - \overline{\mathcal{C}(\hat{x}_{k|k-1})}, \ldots, \mathcal{C}(\hat{x}^N_{k|k-1}) - \overline{\mathcal{C}(\hat{x}_{k|k-1})}\right]^{\mathrm{T}}, \tag{4.14}$$

with $\overline{\mathcal{C}(\hat{x}_{k|k-1})}$ defined by (4.6), and the innovation matrix $\tilde{R}_{e_k}$ as in (3.41).

Next, the estimated square root matrix $\tilde{S}_{k|k}$ is computed from

$$\tilde{S}_{k|k} = \tilde{S}_{k|k-1}\tilde{\mathcal{D}}_k \tag{4.15}$$

where $\tilde{\mathcal{D}}_k \in \mathbb{R}^{N\times N}$ satisfies

$$\tilde{\mathcal{D}}_k\tilde{\mathcal{D}}_k^{\mathrm{T}} = (I_N - \tilde{\mathcal{F}}_k\tilde{R}_{e_k}^{-1}\tilde{\mathcal{F}}_k^{\mathrm{T}}).$$

As formulated, the updated ensemble $\tilde{S}_{k|k}$ is a linear combination of the columns of $\tilde{S}_{k|k-1}$ and is obtained by inverting $\tilde{R}_{e_k}$ and computing a matrix square-root $\tilde{\mathcal{D}}_k$ of the $N \times N$ matrix $\left(I_N - \tilde{\mathcal{F}}_k\tilde{R}_{e_k}^{-1}\tilde{\mathcal{F}}_k^{\mathrm{T}}\right)$.

In many typical Earth and Space Science data assimilation applications the state space dimension n is large, while the number of observations m are large for the Earth and small for the Space data assimilation case respectively. Therefore, the method for computing the matrix square-root $(I_N - \tilde{\mathcal{F}}_k\tilde{R}_{e_k}^{-1}\tilde{\mathcal{F}}_k^{\mathrm{T}})^{1/2}$ and the updated ensemble $\tilde{S}_{k|k}$ must be chosen accordingly.

A direct approach is to solve first the linear system

$$\tilde{R}_{e_k}\mathcal{Y}_k = \tilde{\mathcal{F}}_k^{\mathrm{T}}, \tag{4.16}$$

where $\mathcal{Y}_k \in \mathbb{R}^{m\times N}$ is defined by

$$\mathcal{Y}_k \triangleq \tilde{R}_{e_k}^{-1}\tilde{\mathcal{F}}_k^{\mathrm{T}},$$

then it follows that

$$I_N - \tilde{\mathcal{F}}_k\tilde{R}_{e_k}^{-1}\tilde{\mathcal{F}}_k^{\mathrm{T}} = I_N - \tilde{\mathcal{F}}_k\mathcal{Y}_k,$$

from where the square-root matrix $\tilde{\mathcal{D}}_k$ is computed. The solution of (4.16), even when m is large, is practical when the prior error covariance has a low-rank representation and the inverse of the measurement error covariance is available.

When observation errors are uncorrelated, observations can be assimilated one at a time or serially ([72, 58, 19]). For a single observation, $m = 1$, $\tilde{\mathcal{F}}_k$ is a column vector, and the matrix $\tilde{R}_{e_k}$ becomes the scalar σ_k^2, which is the variance of the observation. In this case $\tilde{\mathcal{D}}_k$ can be computed in closed form as shown in §2.4.3, so

$$\tilde{\mathcal{D}}_k = \left(I_N - \tilde{\gamma}_k\tilde{\mathcal{F}}_k\tilde{\mathcal{F}}_k^{\mathrm{T}}\right), \tag{4.17}$$

where $\tilde{\gamma}_k$ is a scalar defined by

$$\tilde{\gamma}_k \triangleq \frac{\tilde{\alpha}_k}{1+\sqrt{\tilde{\alpha}_k\sigma_k^2}},$$

with $\tilde{\alpha}_k$ scalar defined by

$$\tilde{\alpha}_k \triangleq \left(\tilde{\mathcal{F}}^{\mathrm{T}}\tilde{\mathcal{F}} + \sigma_k^2\right)^{-1}.$$

Next, replacing (4.17) into (4.15) we have

$$\begin{aligned}\tilde{S}_{k|k} &= (I_N - \tilde{K}_k\mathcal{C})\tilde{S}_{k|k-1}\\ &= \tilde{S}_{k|k-1} - \tilde{K}\tilde{\mathcal{F}}_k^T,\end{aligned}$$

where $\tilde{K}_k \in \mathbb{R}^{n\times m}$ is defined by

$$\tilde{K}_k \triangleq \tilde{\gamma}_k\tilde{S}_{k|k-1}\tilde{\mathcal{F}}_k.$$

As a result, it is convenient for the EnSRF to update the equations for the ensemble mean and the deviation of the i-th member from the mean separately, hence

$$\bar{\tilde{x}}_{k|k} = \bar{\tilde{x}}_{k|k-1} + K_k\big(y_k - \mathcal{C}(\bar{\tilde{x}}_{k|k-1})\big), \tag{4.18}$$

$$\hat{\tilde{x}}^i_{k|k} = \hat{\tilde{x}}^i_{k|k-1} - \tilde{K}_k\mathcal{C}(\hat{\tilde{x}}^i_{k|k-1}), \tag{4.19}$$

with $K_k \in \mathbb{R}^{n\times m}$ the traditional Kalman filter gain defined as

$$K_k \triangleq \tilde{\alpha}_k\tilde{S}_{k|k-1}\tilde{\mathcal{F}}_k. \tag{4.20}$$

Notice from (4.20) that

$$\tilde{K}_k = \beta_k K_k,$$

with β_k a scalar defined by

$$\beta_k = \frac{1}{1+\sqrt{\tilde{\alpha}_k\sigma_k^2}},$$

thus the quantity multiplying K_k is a scalar between 0 and 1. This means that, in order to obtain the correct predicted error covariance with unperturbed observations, a modified Kalman gain which is reduced in magnitude relative to the traditional Kalman gain is used to update deviations from the ensemble mean. Consequently, deviations from the mean are reduced less in the analysis using $\tilde{K}_k$ than they would be using K_k. In the EnKF, the excess variance reduction caused by using K_k to update deviations from the mean is compensated for by the introduction of noise to the observations.

In the EnSRF, the mean and departures from the mean are updated independently according to (4.18) and (4.19). If observations are processed one at a time, the EnSRF requires about the same computation as the standard EnKF with perturbed observations, but for moderately sized ensembles and processes that are generally linear and Gaussian, the EnSRF produces predictions with significantly less error [89]. The algorithm is described in Algorithm 14.

Algorithm 14 *EnSRF Estimator*

First, generate an initial ensemble $X_{k-1|k-1}$ which properly represent the error statistics of the initial guess for the model state.

For $k = 1, \ldots$

1. *Update the ensemble members and the mean value of $X_{k-1|k-1}$ using (4.1) for $i = 1, \ldots, N$ according to*

$$\hat{x}^i_{k|k-1} = \mathcal{A}(\hat{x}^i_{k-1|k-1}) + \epsilon^i_{k-1},$$

$$\bar{\hat{x}}_{k|k-1} = \frac{1}{N}\sum_{i=1}^{N} \hat{x}^i_{k|k-1}.$$

2. *Compute $\tilde{S}_{k|k-1}$ using (4.12)*
3. *Compute $\overline{\mathcal{C}(\hat{x}_{k|k-1})}$ using (4.6)*
4. *Update $\tilde{S}_{k|k-1}$ to $\tilde{S}_{k|k}$, and $\bar{\hat{x}}_{k|k-1}$ to $\bar{\hat{x}}_{k|k}$ using sequential processing as follows,*

   $\tilde{S}_{k|k_{(0)}} = \tilde{S}_{k|k-1}$

   for $j = 1$ to m,

$$\tilde{\mathcal{F}}_{k_{(j)}} = \frac{1}{(N-1)^{1/2}}\left[\mathcal{C}_{(j)}(\hat{x}^1_{k|k-1}) - \overline{\mathcal{C}_{(j)}(\hat{x}_{k|k-1})}, \right.$$
$$\left. \ldots, \mathcal{C}_{(j)}(\hat{x}^N_{k|k-1}) - \overline{\mathcal{C}_{(j)}(\hat{x}_{k|k-1})}\right]^{\mathrm{T}}$$
$$\tilde{\alpha}_{k_{(j)}} = \left(\tilde{\mathcal{F}}^{\mathrm{T}}\tilde{\mathcal{F}} + \sigma^2_k\right)^{-1}$$
$$K_{k_{(j)}} = \alpha_{k_{(j)}}\tilde{S}_{k|k_{(j-1)}}\tilde{\mathcal{F}}_{k_{(j)}}$$
$$\bar{\hat{x}}_{k|k} = \bar{\hat{x}}_{k|k-1} + K_{k_{(j)}}(y_k - \mathcal{C}_{k_{(j,:)}}(\bar{\hat{x}}_{k|k-1}))$$
$$\beta_{k_{(j)}} = \frac{1}{1 + \sqrt{\tilde{\alpha}_{k_{(j)}} R_{k_{(j,j)}}}}$$
$$\tilde{K}_{k_{(j)}} = \beta_{k_{(j)}} K_{k_{(j)}}$$
$$\tilde{S}_{k|k_{(j)}} = \tilde{S}_{k|k_{(j-1)}} - \tilde{K}_{k_{(j)}}\tilde{\mathcal{F}}^{\mathrm{T}}_{k_{(j)}}$$

   end

5. *Compute $X_{k|k} = (N-1)^{-1/2}\tilde{S}_{k|k} + \bar{\hat{x}}_{k|k} \cdot \mathbf{1}_N$, where $\mathbf{1}_N \in \mathbb{R}^{1\times N}$ is a vector whose elements are 1s.*

end

4.6 Ensemble Transform Kalman Filter –ETKF

The ETKF approach is based on the TKF in §2.4.4. This filter has been introduced by Bishop [19]. It differs from other ensemble Kalman filters in that it uses ensemble transformation and a normalization to rapidly obtain the prediction error covariance matrix associated with a particular deployment of observational resources. The basic idea behind the ETKF is to define a transformation $\mathcal{D}_k$ so that the transformed $\tilde{S}_{k|k-1}$ can describe the forecast error covariance $\hat{P}_{k|k}$.

Numerical Formulation

Let us start with a general formulation. First observe that if a normalized observation operator $\tilde{C}_k \in \mathbb{R}^{m\times n}$ is defined by

$$\tilde{\mathcal{C}} \triangleq L_{R_k}^{-1}\mathcal{C}$$

where $L_{R_k} \in \mathbb{R}^{m\times m}$ satisfies the Cholesky factorization of measurement error covariance matrix R_k, so

$$R_k = L_{R_k}L_{R_k}^{\mathrm{T}},$$

then $\hat{P}_{k|k}$ can be factored as

$$\hat{P}_{k|k} = \tilde{S}_{k|k-1}\left[I_N - \tilde{S}_{k|k-1}^{\mathrm{T}}\tilde{\mathcal{C}}^{\mathrm{T}}(\tilde{\mathcal{C}}S_{k|k-1}\tilde{S}_{k|k-1}^{\mathrm{T}}\tilde{\mathcal{C}}^{\mathrm{T}} + I_m)^{-1}\tilde{\mathcal{C}}\tilde{S}_{k|k-1}\right]\tilde{S}_{k|k-1}^{\mathrm{T}} \tag{4.21}$$

$$\hat{P}_{k|k} = \tilde{S}_{k|k-1}(I_N - \hat{\mathcal{F}}\mathcal{T}^{-1}\hat{\mathcal{F}}^{\mathrm{T}})\tilde{S}_{k|k-1}^{\mathrm{T}},$$

where $\tilde{S}_{k|k-1}$ is defined by (4.12), and $\hat{\mathcal{F}}_k \in \mathbb{R}^{N\times m}$ and $\mathcal{T}_k \in \mathbb{R}^{m\times m}$ by

$$\hat{\mathcal{F}}_k \triangleq \left(\tilde{\mathcal{C}}\tilde{S}_{k|k-1}\right)^{\mathrm{T}} = \tilde{\mathcal{F}}_k L_{R_k}^{-T},$$
$$\mathcal{T}_k \triangleq \hat{\mathcal{F}}_k^{\mathrm{T}}\hat{\mathcal{F}}_k + I_m,$$

with $\tilde{\mathcal{F}}_k$ defined by (4.13).

Next, using the identity (2.36) $\hat{P}_{k|k}$ can be rewritten as

$$\hat{P}_{k|k} = \tilde{S}_{k|k-1}\left(\hat{\mathcal{F}}_k\hat{\mathcal{F}}_k^{\mathrm{T}} + I_N\right)^{-1}\tilde{S}_{k|k-1}^{\mathrm{T}}. \tag{4.22}$$

Then, it follows from (4.21) that the transformed Kalman gain $\hat{K}_k \in \mathbb{R}^{n\times m}$ can be obtained as

$$\hat{K}_k = \tilde{S}_{k|k-1}\hat{\mathcal{F}}_k\mathcal{T}_k^{-1}.$$

Notice that the standard Kalman filter gain K_k is related to $\hat{K}_k$ by

$$K_k = \hat{K}_k L_{R_k}^{-1}.$$

To obtain the square root representation of $\hat{P}_{k|k}$ we can rewrite (4.22) as follows,

$$\hat{P}_{k|k} = \tilde{S}_{k|k-1}\mathcal{D}_k\mathcal{D}_k^{\mathrm{T}}S_{k|k-1}^{\mathrm{T}},$$

where $\mathcal{D}_k \in \mathbb{R}^{N\times N}$ is the square root of $(\hat{\mathcal{F}}_k\hat{\mathcal{F}}_k^{\mathrm{T}} + I_N)^{-1}$ and can be computed as

$$\mathcal{D}_k = V_k(\Lambda_k + I_N)^{-1/2},$$

where $V_k \in \mathbb{R}^{N\times N}$, and $\Lambda_k \in \mathbb{R}^{N\times N}$ are the left eigenvectors and eigenvalues of $(\hat{\mathcal{F}}_k\hat{\mathcal{F}}_k^{\mathrm{T}})^{-1}$, respectively. Thus, the update of the square root covariance matrix can be yielded by

$$\tilde{S}_{k|k} = \tilde{S}_{k|k-1}V_k(\Lambda_k + I_N)^{-1/2}.$$

The algorithm is shown in Algorithm 15.

Algorithm 15 *ETKF Estimator*

First, generate an initial ensemble $X_{k-1|k-1}$ which properly represent the error statistics of the initial guess for the model state.

For $k = 1, \ldots,$

1. *Update the ensemble members and the mean value of $X_{k-1|k-1}$ using (2.1) for $i = 1, \ldots, q$ according to*

$$\hat{x}^i_{k|k-1} = \mathcal{A}(\hat{x}^i_{k-1|k-1}, k-1) + \varepsilon^i_{k-1},$$
$$\bar{\hat{x}}_{k|k-1} = \frac{1}{N}\sum_{i=1}^{N} \hat{x}^i_{k|k-1}.$$

2. *Compute $\tilde{S}_{k|k-1}$ from (4.12)*

3. *Compute $\hat{\mathcal{F}}_k$ and $\mathcal{T}_k$*

$$\hat{\mathcal{F}}_k = \frac{1}{(N-1)^{1/2}}\left[\mathcal{C}(\hat{x}^1_{k|k-1}) - \overline{\mathcal{C}(\hat{x}_{k|k-1})}, \ldots, \mathcal{C}(\hat{x}^N_{k|k-1}) - \overline{\mathcal{C}(\hat{x}_{k|k-1})}\right]^{\mathrm{T}} L_{R_k}^{-\mathrm{T}},$$
$$\mathcal{T}_k \triangleq \hat{\mathcal{F}}_k^{\mathrm{T}}\hat{\mathcal{F}}_k + I_m.$$

4. *Compute the ETKF gain and update $\bar{\hat{x}}_{k|k-1}$ to $\bar{\hat{x}}_{k|k}$*

$$\hat{K}_k = \tilde{S}_{k|k-1}\hat{\mathcal{F}}_k\mathcal{T}_k^{-1},$$
$$\bar{\hat{x}}_{k|k} = \bar{\hat{x}}_{k|k-1} + \hat{K}_k\left[L_{R_k}^{-1}y_k - \bar{\mathcal{C}}(\bar{\hat{x}}_{k|k-1})\right].$$

5. *Compute the eigenvalue decomposition of $\hat{\mathcal{F}}_k\hat{\mathcal{F}}_k^{\mathrm{T}}$*

$$\hat{\mathcal{F}}_k\hat{\mathcal{F}}_k^{\mathrm{T}} = V_k\Lambda_k V_k^{\mathrm{T}},$$

6. *Compute $\tilde{S}_{k|k}$*

$$\tilde{S}_{k|k} = \tilde{S}_{k|k-1}V_k(\Lambda_k + I_N)^{-1/2}$$

7. *Compute $X_{k|k}$*

$$X_{k|k} = (N-1)^{1/2}\tilde{S}_{k|k} + \bar{\hat{x}}_{k|k}\cdot \mathbf{1}_N,$$

where $\mathbf{1}_N \in \mathbb{R}^{1\times N}$ is a vector whose elements are 1s.

end

4.7 The Ensemble Spatially Localized Kalman Filter –EnSLKF

Comparing the algorithms of the SLKF §3.4 to the EnKF §4.4, we observe that the error covariance update from $P_{k|k-1}$ to $P_{k|k}$ in the EnKF is simply done by updating each ensemble member using (4.4) instead of using (2.13). Therefore, by combining the SLKF and the EnKF we can easily obtain a nonlinear version of the SLKF.

The main difference between the two algorithms is the computation of the filter gain, while for the EnKF it is given by (4.5) for the SLKF by (3.20). Hence, the EnSLKF gain can be defined as

$$\Gamma_k\tilde{K}_k \triangleq \pi_k\hat{P}_{k|k-1}\mathcal{C}_k^{\mathrm{T}}\hat{R}_{e_k}^{-1}. \tag{4.23}$$

Then it follows that the error covariance update, equation (3.21), is done by updating each ensemble member as follows

$$x^i_{k|k} = x^i_{k|k-1} + \Gamma_k\tilde{K}_k\big(y_k - \mathcal{C}(x_{k|k-1})\big).$$

Apart from these changes, notice that the rest of the algorithm for the EnSLKF stays equal to the EnKF.

On the other hand, observe in (4.23) that when n is large, π_k becomes too big to store, so, if we can assume Γ_k and M_k to be time invariant, then $\Gamma_k K_k$ can be written such that the computation of the EnSLKF gain is done more efficiently. Hence, let us define

$$\Gamma_k\tilde{K}_k \triangleq \Upsilon\Phi_k,$$

where $\Upsilon \in R^{n\times n_p}$ is defined by

$$\Upsilon \triangleq \Gamma(\Gamma^{\mathrm{T}}M\Gamma)^{-1},$$

and $\Phi_k \in R^{n_p\times m}$ is defined by

$$\Phi_k \triangleq \Gamma^{\mathrm{T}}\hat{P}_{k|k-1}\mathcal{C}_k^{\mathrm{T}}\hat{R}_{e_k}^{-1}.$$

Consequently, the term Υ becomes constant, therefore it can be computed off-line and stored, while Φ_k is computed once each iteration. The algorithm is described in Algorithm 16.

Algorithm 16 *EnSLKF Estimator*

First, generate an initial ensemble $X_{k-1|k-1}$ which properly represent the error statistics of the initial guess for the model state. Then define the weighting matrix Γ_k and compute $\Upsilon = \Gamma(\Gamma^{\mathrm{T}} M \Gamma)^{-1}$. Then,

for $k = 1, \ldots,$

1. *Update the ensemble members of $X_{k-1|k-1}$ using (4.1) for $i = 1, \ldots, N$ according to*

$$x^i_{k|k-1} = \mathcal{A}(x^i_{k-1|k-1}, k-1) + \varepsilon^i_{k-1}$$

2. *Compute $\overline{\mathcal{C}(\hat{x}_{k|k-1})}$*

$$\overline{\mathcal{C}(\hat{x}_{k|k-1})} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{C}(\hat{x}^i_{k|k-1}),$$

3. *Compute the $\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}}$*

$$\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}} = \frac{1}{N-1}\sum_{i=1}^{N} \left(\hat{x}^i_{k|k-1} - \bar{\hat{x}}_{k|k-1}\right)\left(\mathcal{C}(\hat{x}^i_{k|k-1}) - \overline{\mathcal{C}(\hat{x}_{k|k-1})}\right)^{\mathrm{T}},$$

4. *Compute $\mathcal{C}\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}}$*

$$\mathcal{C}\hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}} = \frac{1}{N-1}\sum_{i=1}^{N} \left(\mathcal{C}(\hat{x}^i_{k|k-1}) - \overline{\mathcal{C}(\hat{x}_{k|k-1})}\right)\left(\mathcal{C}(\hat{x}^i_{k|k-1}) - \overline{\mathcal{C}(\hat{x}_{k|k-1})}\right)^{\mathrm{T}}.$$

5. *Compute Φ_k and K_{e_k}*

$$\Phi_k = \Gamma^{\mathrm{T}} \hat{P}_{k|k-1}\mathcal{C}^{\mathrm{T}} \hat{R}^{-1}_{e_k},$$
$$\Gamma_k \tilde{K}_k = \Upsilon \Phi_k.$$

6. *Update $X_{k|k-1}$ to $X_{k|k}$, for $i = 1, \ldots, N$*

$$x^i_{k|k} = x^i_{k|k-1} + \Gamma_k \tilde{K}_k \left(y^i_k - \mathcal{C}(x^i_{k|k-1})\right)$$

7. *Compute the state estimation taking the mean of the updated ensemble members*

$$\hat{x}_{k|k} = \frac{1}{N}\sum_{i=1}^{N} \hat{x}^i_{k|k}.$$

end

4.8 Computational Complexity of the Algorithms

In Table 4.2 we observe that the computational complexity of the EKF is higher than the other filters, i.e., while the EKF is $\mathcal{O}(n^3) + 1(ME)$, the others are $\backsim \mathcal{O}(n) + N(ME)$, where ME is a model evaluation. Even tough the EKF filter has 1 model evaluations compare to N of the EnKF-like filters, this is still more demanding and the implementation more complex than the others 4.3.

EKF	$6n^3 + 3n^2(2m+1) + n(4m^2+3m+1) + 1(ME)$
EnKF	$n[3N + m(2m+7) + 2] + Nm(2m+1) + m(m^2/3 + 2m + 1) + N(ME)$
EnSRF	$n[8N + m(5N+8) + 1] + Nm(m+2) + 4m + N(ME)$
ETKF	$n[N(N+m+1) + m(m+1)] + Nm(m+1) + N^3 + m^2(m+1) + N(ME)$
EnSLKF	$n[3N + m(6m+7) + 2] + Nm(2m+1) + m(m^2/3 + 2m + 1) + N(ME)$

Table 4.1: Number of flops. ME=Model Evaluations.

EKF	$\mathcal{O}(n^3) + 1(ME)$
EnKF	$\mathcal{O}(n(N + m^2 + m)) + N(ME)$
EnSRF	$\mathcal{O}(n(N + Nm + m)) + N(ME)$
ETKF	$\mathcal{O}(n(N^2 + m^2 + Nm + N + m)) + N(ME)$
EnSLKF	$\mathcal{O}(n(N + m^2 + m)) + N(ME)$

Table 4.2: Order of computation. ME=Model Evaluations.

Where, n is the order of the system, m the number of observations, p the number of inputs, l the order of the process-noise covariance matrix Q, and N the number of ensemble members in the EnKF.

All the ensemble-based Kalman filter algorithms are very sensitive to the number of ensemble members N, and the observations m. Additionally, it can be clearly seen in Table 4.2 that the ETKF is the most expensive algorithm to compute.

4.9 Conclusions

In this chapter we have made a short review about the most representative nonlinear Kalman filter approaches, namely, the extended Kalman filter, and the ensemble Kalman filter. Despite being a well known nonlinear Kalman

filter, the EKF is restricted to systems of small order with weak nonlinearities, consequently it can hardly be applied to complex environmental systems, where it one often encounters complex large scale systems with strong nonlinearities. Nevertheless, this is a good point of reference when we need to assess other nonlinear Kalman filter algorithms.

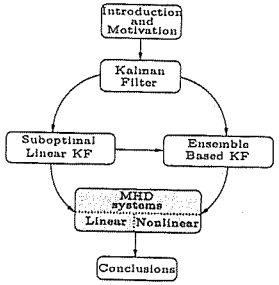
On the other hand, though there is still a lot of research to be done about it, the EnKF has shown to be a powerful algorithm for estimation in large scale and complex nonlinear systems, as reported by several authors on different applications (see [6, 12, 9, 32, 34, 35, 36, 51, 53, 57, 58, 67, 68, 71, 73, 78, 89, 88, 91]). Moreover, the EnKF is very attractive due to the fact that it is easy to deploy, that is, you do not need to compute any linearized model of your system, and it can be used directly with any nonlinear system simulator code. As a result, the EnKF offers an excellent framework to make data assimilation by combining a reliable nonlinear model with an optimal observer.

Apart from this, we have extended the results of the previous chapter to their nonlinear cases, namely, the EnKF, the EnSRF, the ETKF, and the EnSLKF. The EnSLKF is the nonlinear version of the SLKF based on the EnKF formulation. This filter has been designed to be used in large scale systems when the number of observations is small, hence, the aim of this filter is to make local estimations around the measurement points, without disturbing the numerical stability of the system. These local estimations create sparsity in the error covariance matrix which can be exploited to speed up the computation of the filter. On the other hand, the formulation of the EnSLKF is an optimal alternative to the heuristic covariance localization approach, widely used in the Earth weather forecast community.

Finally, the correct definition of the process noise covariance matrix Q is one of the most difficult tasks in order to obtain a good Kalman filter estimator. Additionally, in large scale systems the storage of the error covariance matrix P, and the process noise covariance matrix Q is another issue of a lot of concern when we want to apply the Kalman filters-like approach to real environmental systems, like oceanic or atmospheric. Hence, we described two different methods to cope with this problem, namely, the covariance localization to deal with the size of P, and the covariance inflation to avoid filter divergence caused by a mis-specification of Q.

Chapter 5

Space Weather Forecast Case Study



5.1 Introduction

This chapter is dedicated to exploring the application of sequential data assimilation techniques to space weather forecast. Notice that, contrary to the weather forecast data assimilation problem, the number of measurement points is small compared to the order of the system. We start with a problem motivation, then a description of magnetohydrodynamics (MHD) systems, which are used to simulate the space plasma dynamics, is done. Finally we apply the data assimilation techniques introduced in the previous chapters to MHD systems.

Mainly, the discussion focuses on two cases. First the linear case which is

important in the space community to study the behavior of the space plasma in steady state, and second, the nonlinear case where the aim is to investigate the performance of ensemble-based Kalman filters in MHD systems for space weather prediction.

First, in the linear case a continuous-time 2D-linear model is obtained by taking small perturbations around steady state conditions. Then, two spatial discretization techniques are utilized in order to yield a state-space representation, namely, the Fourier collocation and the Chebyshev collocation method. Consequently, an analysis of the performance of the suboptimal linear KFs introduced in Chapter 3 is carried out.

Secondly, for the nonlinear case a 2D-MHD system emulating the magnetosphere around the Earth is simulated using a numerical model written in Fortran (the VAC code [79]). Then, by means of a modular scheme where Matlab and VAC are used as estimator and simulator, respectively, we apply the ensemble-based filters introduced in Chapter 4 to this system. Finally, the discussion focuses on the comparison of performance of the filters.

5.2 Problem Motivation

The Sun is an average star, similar to millions of others in the Universe. It is a prodigious energy machine, manufacturing about 3.8×10^{23} kW (or kJ/s). In other words, if the total output of the Sun was gathered for one second it would provide the U.S. with enough energy, at its current usage rate, for the next 9 millions of years. The basic energy source for the Sun is nuclear fusion, which uses the high temperatures and densities within the core to fuse hydrogen, producing energy and creating helium as a byproduct. The core is so dense and the size of the Sun so great that energy released at the center of the Sun takes about 50 millions of years to make its way to the surface, undergoing countless absorptions and re-emissions in the process. If the Sun were to stop producing energy today, it would take 50 million of years for significant effects to be felt at Earth.

The region between the Sun and the planets has been termed the interplanetary medium. Although once considered a perfect vacuum, this is actually a turbulent region dominated by the solar wind, which flows at velocities of approximately 250-1000 km/s (about 9×10^5 to 36×10^5 km/h). Other characteristics of the solar wind (density, composition, and magnetic field strength, among others) vary with changing conditions on the Sun. The effect of the

solar wind can be seen in the tails of comets which always point away from the Sun. One of the events which provoke the solar wind is the coronal mass ejection (CME). The CMEs are one of the most energetic and important solar phenomena. These events propel magnetic clouds with a mass up to 10^{17} g to speeds up to 2600 km/s into the heliosphere, influencing near-Earth plasma conditions (space weather), causing fluctuations in the terrestrial magnetic field and in the ionospheric density, and driving auroras.

The solar wind flows around obstacles such as planets, but those planets with their own magnetic fields respond in specific ways. Earth's magnetic field is very similar to the pattern formed when iron filings align around a bar magnet. Under the influence of the solar wind, these magnetic field lines are compressed in the Sunward direction and stretched out in the downwind direction forming the magnetosheath (see Figure 5.1). Then close to Earth it creates the magnetosphere, a complex, teardrop-shaped cavity around Earth. The Van Allen radiation belts are within this cavity, as is the ionosphere, a layer of Earth's upper atmosphere where photo ionization by solar x-rays and extreme ultraviolet rays creates free electrons. Earth's magnetic field senses the solar wind's speed, density, and magnetic field. Because the solar wind varies over time scales as short as seconds, the interface that separates interplanetary space from the magnetosphere is very dynamic. Normally this interface called the magnetopause lies at a distance equivalent to about 10 Earth radii in the direction of the Sun. However, during episodes of elevated solar wind density or velocity, the magnetopause can be pushed inward to within 6.6 Earth radii (the altitude of geosynchronous satellites). As the magnetosphere extracts energy from the solar wind, internal processes produce geomagnetic storms. These storms cause some dangerous effects on Earth, let us mention some of the most affected areas:

- Communications

  Many communication systems utilize the ionosphere to reflect radio signals over long distances. Ionospheric storms can affect radio communication at all latitudes. Some radio frequencies are absorbed and others are reflected, leading to rapidly fluctuating signals and unexpected propagation paths. TV and commercial radio stations are little affected by solar activity, but ground-to-air, ship-to-shore, Voice of America, Radio Free Europe, and amateur radio are frequently disrupted. Radio operators using high frequencies rely upon solar and geomagnetic alerts to keep their communication circuits up and running.

  Some military detection or early-warning systems are also affected by solar activity. The Over-the-Horizon Radar bounces signals off the iono-

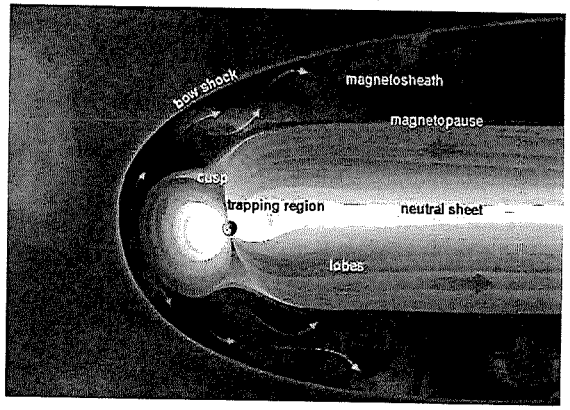


Figure 5.1: The shocked solar wind forms the magnetosheath, taken from http://sci.esa.int.

sphere in order to monitor the launch of aircraft and missiles from long distances. During geomagnetic storms, this system can be severely hampered by radio clutter. Some submarine detection systems use the magnetic signatures of submarines as one input to their locating schemes. Geomagnetic storms can mask and distort these signals.

The Federal Aviation Administration of U.S.A. routinely receives alerts of solar radio bursts so that they can recognize communication problems and forego unnecessary maintenance. When an aircraft and a ground station are aligned with the Sun, jamming of air-control radio frequencies can occur. This can also happen when an Earth station, a satellite, and the Sun are in alignment.

Systems such as LORAN and OMEGA are adversely affected when solar activity disrupts their signal propagation. The OMEGA system consists of eight transmitters located throughout the world. Airplanes and ships use the very low frequency signals from these transmitters to determine their positions. During solar events and geomagnetic storms, the system can give navigators information that is inaccurate by as much as several miles. If navigators are alerted that a proton event or geomagnetic storm is in progress, they can switch to a backup system. GPS signals are affected when solar activity causes sudden variations in the density of the ionosphere.

- Satellites

  Geomagnetic storms and increased solar ultraviolet emission heat Earth's upper atmosphere, causing it to expand. The heated air rises, and the density at the orbit of satellites up to about 1000 km increases significantly. This results in increased drag on satellites in space, causing them to slow and change orbit slightly. Unless low-Earth-orbit satellites are routinely boosted to higher orbits, they slowly fall, and eventually burn up in Earth's atmosphere.

  Skylab is an example of a spacecraft re-entering Earth's atmosphere prematurely as a result of higher-than-expected solar activity. During the great geomagnetic storm of March 1989, four of the Navy's navigational satellites had to be taken out of service for up to a week.

  As technology has allowed spacecraft components to become smaller, their miniaturized systems have become increasingly vulnerable to the more energetic solar particles. These particles can cause physical damage to microchips and can change software commands in satellite-borne computers.

  Another problem for satellite operators is differential charging. During geomagnetic storms, the number and energy of electrons and ions increase. When a satellite travels through this energized environment, the charged particles striking the spacecraft cause different portions of the spacecraft to be differentially charged. Eventually, electrical discharges can arc across spacecraft components, harming and possibly disabling them. Bulk charging (also called deep charging) occurs when energetic particles, primarily electrons, penetrate the outer covering of a satellite and deposit their charge in its internal parts. If sufficient charge accumulates in any one component, it may attempt to neutralize by discharging to other components. This discharge is potentially hazardous to the satellite's electronic systems.

- Radiation Hazards to Humans

  Intense solar flares release very-high-energy particles that can be as injurious to humans as the low-energy radiation from nuclear blasts. Earth's atmosphere and magnetosphere allow adequate protection for us on the ground, but astronauts in space are subject to potentially lethal dosages of radiation. The penetration of high-energy particles into living cells, measured as radiation dose, leads to chromosome damage and, potentially, cancer. Large doses can be fatal immediately. Solar protons with energies greater than 30 MeV are particularly hazardous. In October 1989, the Sun produced enough energetic particles that an astronaut on

  the Moon, wearing only a space suit and caught out in the brunt of the storm, would probably have died. (Astronauts who had time to gain safety in a shelter beneath moon soil would have absorbed only slight amounts of radiation.)

  Solar proton events can also produce elevated radiation aboard aircraft flying at high altitudes. Although these risks are small, monitoring of solar proton events by satellite instrumentation allows the occasional exposure to be monitored and evaluated.

- Geologic Exploration

  Earth's magnetic field is used by geologists to determine subterranean rock structures. For the most part, these geodetic surveyors are searching for oil, gas, or mineral deposits. They can accomplish this only when Earth's field is quiet, so that true magnetic signatures can be detected. Other surveyors prefer to work during geomagnetic storms, when the variations to Earth's normal subsurface electric currents help them to see subsurface oil or mineral structures. For these reasons, many surveyors use geomagnetic alerts and predictions to schedule their mapping activities.

- Electric Power

  When magnetic fields move about in the vicinity of a conductor such as a wire, an electric current is induced into the conductor. This happens on a grand scale during geomagnetic storms. Power companies transmit alternating current to their customers via long transmission lines. The nearly direct currents induced in these lines from geomagnetic storms are harmful to electrical transmission equipment. On March 13, 1989, in Montreal, Quebec, 6 million people were without commercial electric power for 9 hours as a result of a huge geomagnetic storm. Some areas in the northeastern U.S. and in Sweden also lost power. By receiving geomagnetic storm alerts and warnings, power companies can minimize damage and power outages.

- Pipelines

  Rapidly fluctuating geomagnetic fields can induce currents into pipelines. During these times, several problems can arise for pipeline engineers. Flow meters in the pipeline can transmit erroneous flow information, and the corrosion rate of the pipeline is dramatically increased. If engineers unwittingly attempt to balance the current during a geomagnetic storm, corrosion rates may increase even more. Pipeline managers routinely receive alerts and warnings to help them provide an efficient and long-lived system.

- Climate

  The Sun is the heat engine that drives the circulation of our atmosphere. Although it has long been assumed to be a constant source of energy, recent measurements of this solar constant have shown that the base output of the Sun can vary by up to two tenths of a percent over the 11-year solar cycle. Temporary decreases of up to one-half percent have been observed. Atmospheric scientists say that this variation is significant and that it can modify climate over time. Plant growth has been shown to vary over the 11-year sunspot and 22-year magnetic cycles of the Sun, as evidenced in tree-ring records.

  While the solar cycle has been nearly regular during the last 300 years, there was a period of 70 years during the 17th and 18th centuries when very few sunspots were seen (even though telescopes were widely used). This drop in sunspot number coincided with the timing of the little ice age in Europe, implying a Sun-to-climate connection. Recently, a more direct link between climate and solar variability has been speculated. Stratospheric winds near the equator blow in different directions, depending on the time in the solar cycle. Studies are under way to determine how this wind reversal affects global circulation patterns and weather.

  During proton events, many more energetic particles reach Earth's middle atmosphere. There they cause molecular ionization, creating chemicals that destroy atmospheric ozone and allow increased amounts of harmful solar ultraviolet radiation to reach Earth's surface. A solar proton event in 1982 resulted in a temporary 70 % decrease in ozone densities.

- Biology

  There is a growing body of evidence that changes in the geomagnetic field affect biological systems. Studies indicate that physically stressed human biological systems may respond to fluctuations in the geomagnetic field. Interest and concern in this subject have led the Union of Radio Science International to create a new commission entitled Electromagnetics in Biology and Medicine.

  Possibly the most closely studied of the variable Sun's biological effects has been the degradation of homing pigeons' navigational abilities during geomagnetic storms. Pigeons and other migratory animals, such as dolphins and whales, have internal biological compasses composed of the mineral magnetite wrapped in bundles of nerve cells. While this probably is not their primarily method of navigation, there have been many pigeon race smashes, a term used when only a small percentage of birds return home from a release site. Because these losses have occurred during geomagnetic storms, pigeon handlers have learned to ask for geomagnetic alerts and warnings as an aid to scheduling races.

As a conclusion, it has been realized and appreciated only in the last few decades that solar flares, CMEs, and magnetic storms affect people and their activities. The list of consequences grows in proportion to our dependence on technological systems. The subtleties of the interactions between Sun and Earth, and between solar particles and delicate instruments, have become factors that affect our well being. Thus there will be continued and intensified need for space environment services to address health, safety, and commercial needs[1].

5.3 Magnetohydrodynamics Model Description

The topic of MHD is ubiquitous in plasma physics. Examples where the theory has been used with success range from explaining the dynamo generation and subsequent evolution of magnetic fields within stellar and planetary interiors, to accounting for the gross stability of magnetically confined thermonuclear plasmas. It transpires that MHD is capable of providing a good description of such large scale disturbances, indicating that the MHD account of plasma behavior is necessarily a macroscopic one. In essence, MHD is a macroscopic, non-relativistic theory that is concerned with global phenomena in magnetic plasmas. It gives an accurate description of many of the complicated interactions of magnetic fields with the plasmas of the sun and stars. The theory is a marriage between fluid mechanics and electromagnetism. Despite its apparent simplicity, MHD describes a remarkably rich and varied mix of phenomena and the subject is one whose development continues to flourish [30, 44, 22].

The basic equations of MHD systems can be summarized as follows:

- Mass continuity: it expresses the fact that the rate of change of mass inside a finite volume equals the mass flow through the boundaries.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0. \tag{5.1}$$

[1] This problem motivation has been taken from http://www.sec.noaa.gov/primer/primer.html

- Adiabatic equation of state: it states that for any element of fluid the term p/ρ^{γ}, is constant, where γ is the ratio of specific heats.

$$\frac{d}{dt}\left(\frac{p}{\rho^{\gamma}}\right) = 0. \tag{5.2}$$

- Momentum equation: the left-hand side of (5.3) is the rate of change of momentum of a fluid element, while the right-hand side is the sum of the forces due to the fluid pressure p and the magnetic field.

$$\rho\frac{\partial v}{\partial t} = J \times B - \nabla p, \tag{5.3}$$

- Ampère's law: in magnetohydrodynamics one generally deals with low frequencies effects in which the displacement current is negligible, hence, the Ampère's law can be written as (5.4). Therefore, equation (5.4) means that a point in an electromagnetic field, the curl of the magnetic field intensity B is equal to the current density J due to flow of charges.

$$\nabla \times B = \mu_0 J. \tag{5.4}$$

- Faraday's law: the Faraday's law relates the electric and magnetic fields in a conducting fluid.

$$\nabla \times E = -\frac{\partial B}{\partial t}. \tag{5.5}$$

- Gauss' law: it tell us that the divergence of the magnetic flux density at a point is equal to zero.

$$\nabla \cdot B = 0, \tag{5.6}$$

- Resistive Ohm's law: the current is determined by Ohm's law, which, since the electric field in a frame of reference moving with the fluid is $E + v \times B$, takes the form of (5.7) with η the resistivity of the plasma.

$$E + v \times B = \eta J, \tag{5.7}$$

where each of the symbols has its customary meaning (see the list of symbols in Appendix B), with the convective derivative

$$\frac{d}{dt} \triangleq \frac{\partial}{\partial t} + v \cdot \nabla.$$

On the other hand, the right-hand side of (5.7) may be neglected to yield the ideal Ohm's law:

$$E + v \times B = 0. \tag{5.8}$$

This states that there is no electric field in the rest frame of the fluid. Equations (5.1)-(5.6) with (5.8) constitute the ideal MHD equations, which is usually contracted to MHD. The inclusion of (5.7) is described as resistive MHD.

It can be seen that the equations are essentially an amalgam of fluids mechanics and 'pre-Maxwell' electromagnetism. The fluid inertia is affected by forces due to the fluid pressure gradients and to the $J \times B$ term, which is the Lorentz force in continuum form. It will be noted however that the Ohm's law couples the fluid to the fields. If the magnetic flux is conserved, then this equation provides constraints on the allowable class of fluid displacements described by the theory, and this in turn has implications for the topology of the magnetic fields.

MHD possesses those conservation properties enjoyed by fluid mechanics and electromagnetism, namely:

- conservation of mass,
- conservation of momentum,
- conservation of energy (both mechanical and electromagnetic),
- conservation of magnetic flux.

5.3.1 Linear 2D-MHD Flow System

If you take a look at the set of the MHD equations you come to realize that we are dealing with a complicated set of non-linear partial differential equations (PDE). This set of PDEs have to be supplemented with boundary conditions if the plasma that we are looking at occupies a finite volume, which is in reality always the case. Therefore, there is no doubt that the set of MHD PDEs is very complicated from a mathematical and computational point of view because it contains a lot of physics.

Fortunately, often the amplitudes of the wave variables are small compared to their equilibrium values. This fortunate situation enables us to linearize the set of original non-linear MHD equations. The linearized versions of the MHD equations are much more tractable to mathematical analysis and numerical simulations. Linear differential equations are well documented in mathematics. Linear theory of MHD waves is a valid approximation of reality if the plasma configuration under study occurs in a steady dynamical time scale of the system, which is the Alfvén transit time. In the solar atmosphere there are

structures that exist during days, weeks and months. So, nature provides an occasion where it is obvious interest to know about equilibrium states.

MHD stability and MHD waves are two complementary facets of the dynamics of a plasma. MHD stability is concerned with the unstable motions of the plasma that terminate the equilibrium while MHD waves are the stable motions that can persist in the plasma equilibrium for long time spans until they are damped by dissipation. It is no surprise that they both can be studied with linear ideal MHD. But it is good to be aware of the limitations of linear ideal MHD. Because it is linear, linear ideal MHD cannot give any information on the non-linear evolution of a linearly unstable perturbation, the stability with respect to large amplitude motions, dissipative instabilities, etc., and the use is restricted to small amplitude MHD waves. Nevertheless the linear ideal MHD can give us a lot of information on the dynamics of magnetic plasmas [44].

Hence, in this section we derive a linear state space representation of an ideal 2D MHD flow system. This model will be used in Section 5.4.2 to study the performance of linear sequential data assimilation techniques, namely the Kalman filter, the SSQRT-KF, the RRSQRT-KF, and the linear EnKF, in MHD systems [24].

Model Linearization

Let us start determining steady-state flow and magnetic field configurations that are consistent with the ideal MHD equations. The domain in this case is the box $\Omega = \{(x, y) \in [L_{x_1}, L_{x_2}] \times [0, M]\}$. Consider a cold plasma (a plasma is cold when the pressure term is negligible in the momentum equation (5.3)), that is, $p \equiv 0$ in (5.3). Furthermore, we assume that the plasma flowing through a 2D channel as shown in Figure 5.2, is incompressible, which implies that the density is constant, temporally and spatially, that is, $\rho = \rho_0$ and hence the adiabatic equation of state (5.2) can be ignored.

Let $\vec{v}_0$ and $\vec{B}_0$ be the steady state solution of the MHD equations (5.1)-(5.6), that is,

$$\frac{\partial \vec{v}_0}{\partial t} = 0, \quad \frac{\partial \vec{B}_0}{\partial t} = 0, \tag{5.9}$$

Hence, it follows from (5.9) and (5.1)-(5.6) that $\vec{v}_0$ and $\vec{B}_0$ satisfy

$$\nabla \cdot \vec{v}_0 = 0, \tag{5.10}$$

$$\rho_0 (\vec{v}_0 \cdot \nabla) \vec{v}_0 = \frac{1}{\mu_0} (\nabla \times \vec{B}_0) \times \vec{B}_0, \tag{5.11}$$

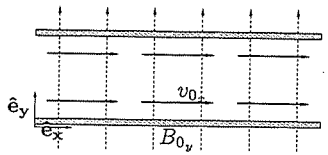


Figure 5.2: The plasma is assumed to flow along the $\hat{\mathbf{e}}_\mathbf{x}$ direction only. The magnetic field is along $\hat{\mathbf{e}}_\mathbf{y}$ and is constant. The magnetic field exerts no pressure on the flow when the velocity of the plasma is parallel to the magnetic field. The pressure exerted by the magnetic field on the plasma when it flows perpendicular to the field depends on the gradient of the magnetic field, which is zero by assumption.

$$\nabla \times (\vec{v}_0 \times \vec{B}_0) = 0, \tag{5.12}$$

$$\nabla \cdot \vec{B}_0 = 0. \tag{5.13}$$

Next, assume that the plasma flows along the $\hat{\mathbf{e}}_\mathbf{x}$ direction with a constant velocity so that

$$\vec{v}_0 = v_{0_x} \hat{\mathbf{e}}_\mathbf{x}$$

and let the constant magnetic field be prescribed by

$$\vec{B}_0 = B_{0_y} \hat{\mathbf{e}}_\mathbf{y}.$$

Note that the prescribed velocity and magnetic field satisfy (5.10) and (5.13), respectively, and hence $\vec{v}_0 = v_{0_x} \hat{\mathbf{e}}_\mathbf{x}$ and $\vec{B}_0 = B_{0_y} \hat{\mathbf{e}}_\mathbf{y}$ are steady-state solutions of the ideal MHD equations.

Next, define the perturbation variables $\vec{v}_\delta$ and $\vec{B}_\delta$ by

$$\vec{v}_\delta \triangleq \vec{v} - \vec{v}_0, \quad \vec{B}_\delta \triangleq \vec{B} - \vec{B}_0. \tag{5.14}$$

Substituting $\rho = \rho_0$ and (5.14) into (5.1) yields

$$\frac{\partial \rho_0}{\partial t} + \rho_0 \nabla \cdot \vec{v}_0 + \rho_0 \nabla \cdot \vec{v}_\delta = 0. \tag{5.15}$$

Since the density is constant, substituting (5.10) into (5.15) yields

$$\nabla \cdot \vec{v}_\delta = 0. \tag{5.16}$$

Then substituting (5.9) and (5.11) into the resulting equation, and ignoring second and higher order perturbation terms yields

$$\rho_0 \frac{\partial \vec{v}_\delta}{\partial t} + \rho_0(\vec{v}_0 \cdot \nabla)\vec{v}_\delta + \rho_0(\vec{v}_\delta \cdot \nabla)\vec{v}_0 = \frac{1}{\mu_0}\left[(\nabla \times \vec{B}_0) \times B_\delta + (\nabla \times \vec{B}_\delta) \times B_0\right]. \tag{5.17}$$

Substituting (5.14) into (5.5) yields

$$\nabla \times \left[(\vec{v}_0 + \vec{v}_\delta) \times (\vec{B}_0 + \vec{B}_\delta)\right] = \frac{\partial}{\partial t}(\vec{B}_0 + \vec{B}_\delta). \tag{5.18}$$

Substituting (5.9) and (5.12) into (5.18) and ignoring second order perturbation terms yields

$$\nabla \times (\vec{v}_0 \times \vec{B}_\delta) + \nabla \times (\vec{v}_\delta \times \vec{B}_0) = \frac{\partial \vec{B}_\delta}{\partial t}. \tag{5.19}$$

Substituting (5.14) and (5.13) into (5.6) yields

$$\nabla \cdot \vec{B}_\delta = 0. \tag{5.20}$$

Hence, (5.17) yields

$$\frac{\partial \vec{v}_\delta}{\partial t} + c_1 \frac{\partial \vec{v}_\delta}{\partial x} = c_2 \frac{\partial \vec{B}_\delta}{\partial y} \tag{5.21}$$

and it follows from (5.19) that

$$\frac{\partial \vec{B}_\delta}{\partial t} + c_1 \frac{\partial \vec{B}_\delta}{\partial x} = c_3 \frac{\partial \vec{v}_\delta}{\partial y}, \tag{5.22}$$

where $c_1 \triangleq v_{0_x}$, $c_2 \triangleq B_{0_y}/(\rho_0\mu_0)$, and $c_3 \triangleq B_{0_y}$. Therefore, (5.16), (5.20), (5.21), and (5.22) are the linearized equations that govern the dynamics of the perturbation variables $\vec{v}_\delta$ and $\vec{B}_\delta$.

Note that (5.21) and (5.22) resemble a two dimensional wave equation. Let, $\vec{v}_\delta$ and $\vec{B}_\delta$ have components $\vec{v}_\delta = \begin{bmatrix} v_{\delta_x} & v_{\delta_y} \end{bmatrix}^{\mathrm{T}}$ and $\vec{B}_\delta = \begin{bmatrix} B_{\delta_x} & B_{\delta_y} \end{bmatrix}^{\mathrm{T}}$, respectively.

Taking the partial derivative of (5.21) with respect to t and x yields,

$$\partial_{tt}\vec{v}_\delta + c_1 \partial_{tx}\vec{v}_\delta = c_2 \partial_{ty}\vec{B}_\delta \tag{5.23}$$

and

$$\partial_{xt}\vec{v}_\delta + c_1 \partial_{xx}\vec{v}_\delta = c_2 \partial_{xy}\vec{B}_\delta, \tag{5.24}$$

respectively.

Taking the partial derivative of (5.22) with respect to y yields

$$\partial_{yt}\vec{B}_\delta + c_1 \partial_{yx}\vec{B}_\delta = c_3 \partial_{yy}\vec{v}_\delta. \tag{5.25}$$

Dividing (5.23) by c_2, and multiplying (5.24) by $\frac{c_1}{c_2}$, and adding the resulting equation with (5.25) yields

$$\partial_{tt}\vec{v}_\delta = -2c_1\partial_{tx}\vec{v}_\delta - c_1^2\partial_{xx}\vec{v}_\delta + c_2c_3\partial_{yy}\vec{v}_\delta. \tag{5.26}$$

Note the symmetry in (5.21) and (5.22) with respect to $\vec{v}_\delta$ and $\vec{B}_\delta$. Hence, using a similar procedure, it can be shown that

$$\partial_{tt}\vec{B}_\delta = -2c_1\partial_{tx}\vec{B}_\delta - c_1^2\partial_{xx}\vec{B}_\delta + c_2c_3\partial_{yy}\vec{B}_\delta. \tag{5.27}$$

Now, the equations governing the perturbations in the velocity field $\vec{v}_\delta$, and the magnetic field $\vec{B}_\delta$, have been decoupled.

State-Space Modeling using Spatial Discretization Methods

The partial differentiation equation (5.26) involves the ∂_{tx} operator and hence a separation of variable technique cannot be used to obtain an equivalent ordinary differential equation representation. Hence, we use spatial discretization methods to obtain an ODE model of the linearized perturbation dynamics. Let $\vec{v}_\delta = v_{\delta_x}\hat{\mathbf{e}}_\mathbf{x} + v_{\delta_x}\hat{\mathbf{e}}_\mathbf{y}$. It follows from (5.16) that there exists a scalar potential function $\psi(x, y, t)$ such that

$$v_{\delta_x} = \frac{\partial \psi}{\partial y}(x, y, t), \quad v_{\delta_y} = -\frac{\partial \psi}{\partial x}(x, y, t). \tag{5.28}$$

Hence, it follows from (5.26) that

$$\partial_{tty}\psi + 2c_1\partial_{txy}\psi + c_1^2\partial_{xxy}\psi = c_2c_3\partial_{yyy}\psi. \tag{5.29}$$

Assume (5.29) has a solution $\psi(x, y, t)$ of the form

$$\psi(x, y, t) = U_m(x, t)W_m(y), \tag{5.30}$$

where for all $m = 0, 1, \ldots$, $U_m : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and $W_m : \mathbb{R} \to \mathbb{R}$. Substituting (5.30) into (5.29) and dividing the resulting equation by $v_{\delta_x} = U_m(x,t)W_m'(y)$ yields

$$\frac{\partial_{tt}U_m(x,t)}{U_m(x,t)} + 2c_1\frac{\partial_{tx}U_m(x,t)}{U_m(x,t)} + c_1^2\frac{\partial_{xx}U_m(x,t)}{U_m(x,t)} = c_2c_3\frac{W_m''(y)}{W_m(y)}, \tag{5.31}$$

which implies that

$$\frac{W_m''(y)}{W_m(y)} = k_m, \tag{5.32}$$

where $k_m \in \mathbb{R}$ is a constant determined by the boundary conditions. We assume that the boundary conditions along $\hat{\mathbf{e}}_y$ are fixed and given by

$$v_{\delta_x}(x,0,t) = 0, \quad v_{\delta_x}(x,M,t) = 0. \tag{5.33}$$

Note that (5.33) is a simplifying assumption and is not equivalent to the no-slip boundary conditions because viscosity has not been included in the ideal MHD equations. It follows from (5.28) that a solution to (5.32) that satisfies (5.33) is

$$W_m(y) = w_{0,m}\sin\left(\frac{m\pi y}{M}\right)$$

where $w_{0,m}$ is determined by the initial condition, and can be assumed to be equal to one without any loss of generality, and

$$k_m = -\omega_{\mathrm{n},m}^2 = -\left(\frac{m\pi}{M}\right)^2. \tag{5.34}$$

Hence, it follows from (5.31) that

$$\partial_{tt}U_m(x,t) + 2c_1\partial_{tx}U_m(x,t) + c_1^2\partial_{xx}U_m(x,t) + k_{0,m}U_m(x,t) = 0, \tag{5.35}$$

where $k_{0,m} \in \mathbb{R}$ is defined by

$$k_{0,m} \triangleq c_2c_3\omega_{\mathrm{n},m}^2 > 0.$$

Fourier Collocation Method

In this section, we assume that the boundary conditions are fixed along the $\hat{\mathbf{e}}_x$ direction, and hence, express $U_m(x,t)$ as a Fourier series in x with time-varying

coefficients. Let $L_{x_1} = 0$ and $L_{x_2} = 2\pi$, and for $i = 1, \ldots, n$, let $x_{\mathrm{f}i} \triangleq \frac{2\pi(i-1)}{n}$, be the n collocation points along the $\hat{\mathbf{e}}_x$ direction. Define $q_m^{x_{\mathrm{f}i}}$ by

$$q_m^{x_{\mathrm{f}i}} \triangleq U_m(x_{\mathrm{f}i}, t). \tag{5.36}$$

Next, for all $k = -\frac{n}{2}, \ldots, \frac{n}{2} - 1$, define the discrete Fourier coefficients $\tilde{q}_{m_k}$ by $\tilde{q}_{m_k} \triangleq \frac{1}{n}\sum_{i=1}^{n} q_m^{x_{\mathrm{f}i}} e^{-\jmath(kx_{\mathrm{f}i})}$ so that $U_m(x,t)$ can be expressed by the discrete Fourier series

$$U_m(x,t) = \sum_{k=-\frac{n}{2}}^{\frac{n}{2}-1} \tilde{q}_{m_k} e^{\jmath(kx)}.$$

Next, define $Q_m \in \mathbb{R}^n$ by

$$Q_m \triangleq \begin{bmatrix} q_m^{x_{\mathrm{f}1}} & \cdots & q_m^{x_{\mathrm{f}n}} \end{bmatrix}^{\mathrm{T}}. \tag{5.37}$$

Using the Fourier collocation differentiation matrix [23] in (5.34) yields

$$\ddot{Q}_m + 2c_1\mathcal{D}_{\mathrm{F}n}\dot{Q}_m + c_1^2\mathcal{D}_{\mathrm{F}n}^2Q_m + k_{0,m}Q_m = 0, \tag{5.38}$$

where $\mathcal{D}_{\mathrm{F}n} \in \mathbb{R}^{n\times n}$ and the (i,j)th element of $\mathcal{D}_{\mathrm{F}n}$ is given by

$$\mathcal{D}_{\mathrm{F}n(i,j)} = \begin{cases} \frac{1}{2}(-1)^{i+j}\cot[\frac{(i-j)\pi}{n}], & i \neq j. \\ 0, & i = j. \end{cases} \tag{5.39}$$

A state-space representation of (5.38) is

$$\begin{bmatrix} \dot{Q}_m \\ \ddot{Q}_m \end{bmatrix} = A_m \begin{bmatrix} Q_m \\ \dot{Q}_m \end{bmatrix}, \tag{5.40}$$

where $A_m \in \mathbb{R}^{2n\times 2n}$ is defined by

$$A_m \triangleq \begin{bmatrix} 0_n & I_n \\ -(c_1^2\mathcal{D}_{\mathrm{F}n}^2 + k_{0,m}I_n) & -2c_1\mathcal{D}_{\mathrm{F}n} \end{bmatrix}. \tag{5.41}$$

Note that A_m can be factored as

$$A_m = P\begin{bmatrix} -c_1\mathcal{D}_{\mathrm{F}n} + \jmath k_{0,m}^{1/2}I_n & 0 \\ S & -c_1\mathcal{D}_{\mathrm{F}n} - \jmath k_{0,m}^{1/2}I_n \end{bmatrix}P^{-1},$$

where $S \in \mathbb{R}^{n\times n}$ is defined by

$$S \triangleq -(c_1^2\mathcal{D}_{\mathrm{F}n}^2 + k_{0,m}I_n).$$

$T \in \mathbb{C}^{n\times n}$ is defined by

$$T \triangleq (c_1\mathcal{D}_{\mathrm{F}n} - \jmath k_{0,m}^{1/2}I_n)S^{-1},$$

and $P \in \mathbb{R}^{2n\times 2n}$ is defined by

$$P \triangleq \begin{bmatrix} I_n & T \\ 0_n & I_n \end{bmatrix},$$

which implies that

$$\operatorname{spec}(A_m) = \operatorname{spec}(-c_1\mathcal{D}_{\mathrm{F}n}+\jmath k_{0,m}^{1/2}I_n) \cup \operatorname{spec}(-c_1\mathcal{D}_{\mathrm{F}n}-\jmath k_{0,m}^{1/2}I_n). \quad (5.42)$$

It follows from (5.39) that $\mathcal{D}_{\mathrm{F}n}$ is skew symmetric and hence all its eigenvalues lie on the imaginary axis. Hence, (5.42) implies that the eigenvalues of A_m are also confined to the imaginary axis, that is, for all $\lambda \in \operatorname{spec}(A_m)$, $\operatorname{Re}(\lambda) = 0$.

Note that (5.38) is a second order system and can be expressed as

$$M\ddot{Q}_m + G\dot{Q}_m + KQ_m = 0,$$

where $M = I_n$, $G = 2c_1\mathcal{D}_{\mathrm{F}n}$, and $K = c_1^2\mathcal{D}_{\mathrm{F}n}^2 + k_0I_n$. Since $GK = KG$ and $K + \frac{1}{4}GG^{\mathrm{T}}$ is positive definite, it follows from Proposition 3 of [16] that (5.38) is Lyapunov stable.

Note that (5.38) represents the dynamics of the mth mode and it follows from the principle of superposition that the solution to (5.29) is given by $\psi(x,y,t) = \sum_{m=1}^{\infty} U_m(x,t)W_m(y)$.

Retaining r modes and defining the modal state vector $\tilde{Q} \in \mathbb{R}^{2nr}$ by

$$\tilde{Q} \triangleq \begin{bmatrix} Q_1^{\mathrm{T}} & \dot{Q}_1^{\mathrm{T}} \cdots Q_r^{\mathrm{T}} & \dot{Q}_r^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}},$$

it follows from (5.40) and (5.41) that

$$\dot{\tilde{Q}} = A\tilde{Q}, \quad (5.43)$$

where $A \in \mathbb{R}^{2nr\times 2nr}$ is the block-diagonal matrix

$$A \triangleq \begin{bmatrix} A_1 & 0 & \cdots \\ 0 & \ddots & 0 \\ \vdots & 0 & A_r \end{bmatrix}.$$

Hence, $\operatorname{spec}(A) = \operatorname{spec}(A_1)\cup\cdots\cup\operatorname{spec}(A_r)$ and, for all $\lambda \in \operatorname{spec}(A)$, $\operatorname{Re}(\lambda) = 0$. Where $\operatorname{spec}(\cdot)$ denotes the eigenvalues of the matrix $(\cdot)$.

Let $y_{\mathrm{x}_{\mathrm{out},i,j}} \triangleq v_{\delta_x}(x_{\mathrm{f}_i}, y_j, t)$ be the measured perturbation in the flow velocity u_x from u_{0_x} at $(x_{\mathrm{f}i}, y_j)$, where $x_{\mathrm{f}i}$ is one of the grid points. It follows from (5.28), (5.30), and (5.36) that

$$y_{\mathrm{x}_{\mathrm{out},i,j}} = \partial_y\psi(x,y,t)\Big|_{(x=x_{\mathrm{f}_i}, y=y_j)} = C_{\mathrm{x}_{i,j}}\tilde{Q}, \quad (5.44)$$

where $C_{\mathrm{x}_{i,j}} \in \mathbb{R}^{1\times 2nr}$ has entries $C_{\mathrm{x}_{i,j}} = \begin{bmatrix} C^1_{\mathrm{x}_{i,j}} & \cdots & C^m_{\mathrm{x}_{i,j}} \end{bmatrix}$, and, for all $m = 1,\ldots,r$, $C^m_{\mathrm{x}_{i,j}} \in \mathbb{R}^{2n}$ is defined by

$$C^m_{\mathrm{x}_{i,j}} \triangleq \begin{bmatrix} 0_{i-1} & W_m'(y_j) & 0_{2n-i} \end{bmatrix}.$$

Next, we consider the case when the measurement $\tilde{y}_{\mathrm{y}_{\mathrm{out},i,j}} \triangleq v_{\delta_y}(x_{\mathrm{f}i}, y_j, t)$ is the perturbation in the flow velocity u_y from u_{0_y} at (x_{f_i}, y_j). It follows from (5.28) that

$$\tilde{y}_{\mathrm{y}_{\mathrm{out},i,j}} = -\partial_x\psi(x,y,t)\Big|_{(x=x_{\mathrm{f}_i}, y=y_j)}. \quad (5.45)$$

Using (5.30), (5.36), and the Fourier collocation differentiation matrix D_{F_n} in (5.45) yields

$$\tilde{y}_{\mathrm{y}_{\mathrm{out},i,j}} = -C_{\mathrm{y}_{i,j}}D_{\mathrm{F}_n}\tilde{Q}, \quad (5.46)$$

where $C_{\mathrm{y}_{i,j}} \in \mathbb{R}^{1\times 2nr}$ has entries $C_{\mathrm{y}_{i,j}} = \begin{bmatrix} C^1_{\mathrm{y}_{i,j}} & \cdots & C^m_{\mathrm{y}_{i,j}} \end{bmatrix}$, and, for all $m = 1,\ldots,r$, $C^m_{\mathrm{y}_{i,j}} \in \mathbb{R}^{2n}$ is defined by

$$C^m_{\mathrm{y}_{i,j}} \triangleq \begin{bmatrix} 0_{i-1} & W_m(y_j) & 0_{2n-i} \end{bmatrix}. \quad (5.47)$$

Note that (5.26) and (5.27) are similar and hence the solution to $\vec{B}_\delta$ is similar to that of $\vec{v}_\delta$, and the constants are determined by the initial and boundary values of the magnetic field instead of the velocity field.

Chebyshev Collocation Method

Next, we express $U_m(x,t)$ as a Chebyshev series in x with time varying coefficients. Let $L_1 = -1$ and $L_2 = 1$, and, for all $i = 1,\ldots,n$, let

$$x_{ci} = -\cos\Big[\frac{(i-1)\pi}{n-1}\Big]$$

be the n Gauss-Lobatto grid points in the interval $[-1, 1]$ (see [23, 1]). Consider a solution of the form (5.30) and define $q_m^{x_{ci}} \triangleq U_m(x_{ci}, t)$. The truncated Chebyshev series expansion for the solution $U_m(x, t)$ is (see [1])

$$U_m(x,t) = \sum_{k=0}^{n-1} \tilde{q}_{m_k} \phi_k(x),$$

where $\phi_k(x) \triangleq \cos(k \cos^{-1}(x))$, and, for all $k = 0, \ldots, n-1$, $\tilde{q}_{m_k}$ is defined by

$$\tilde{q}_{m_k} \triangleq \frac{1}{\gamma_k} \sum_{i=1}^{n} q_m^{x_{ci}} \phi_k(x_i) w_i,$$

where

$$\gamma_k = \begin{cases} \pi, & k = 0 \text{ or } k = n-1, \\ \frac{\pi}{2}, & 0 < k < n-1, \end{cases} \quad w_i = \begin{cases} \frac{\pi}{2(n-1)}, & i = 1 \text{ or } i = n, \\ \frac{\pi}{n-1}, & 1 < i < n. \end{cases}$$

Next, defining $Q_m \in \mathbb{R}^n$ by (5.37) with $q_m^{x_{fi}}$ replaced by $q_m^{x_{ci}}$ and using the Chebyshev collocation differentiation matrix (see [23]) in (5.34) yields (5.38) with $\mathcal{D}_{Fn}$ replaced by $\mathcal{D}_{Cn}$, where the (i, j)th entry of $\mathcal{D}_{Cn}$ is defined by

$$\mathcal{D}_{Cn_{(i,j)}} = \begin{cases} \frac{c_i}{c_j} \frac{(-1)^{i+j}}{(x_{ci} - x_{cj})}, & i \neq j, \\ \frac{-x_{ci}}{2(1-x_{ci}^2)}, & 1 < i = j \leq n, \\ \frac{2(n-1)^2+1}{6}, & i = j = 1, \\ -\frac{2(n-1)^2+1}{6}, & i = j = n, \end{cases}$$

and c_i is defined by

$$c_i = \begin{cases} 2, & i = 1 \text{ or } i = n, \\ 1, & 1 < i < n. \end{cases}$$

The state-space model is then given by (5.43), where A_m is defined by (5.41) with $\mathcal{D}_{Fn}$ replaced by $\mathcal{D}_{Cn}$. The outputs $y_{x_{out,i,j}} \triangleq v_{\delta_x}(x_{ci}, y_j, t)$ and $y_{y_{out,i,j}} \triangleq v_{\delta_y}(x_{ci}, y_j, t)$ are given by (5.44), (5.45).

As a result, in both cases, namely, the Fourier collocation and mapped Chebyshev collocation, all the eigenvalues of A are confined to the imaginary axis. Due to the roundoff errors in the mapped Chebyshev collocation method, the absolute value of the eigenvalues is very large, which implies that the dynamics as given in (5.43) is oscillatory with very high frequency. Hence, the Fourier collocation method will be used later for simulating the MHD flow.

5.3.2 Description of the Nonlinear MHD Model Simulator

In this thesis the software package VAC (Versatile Advection Code) [79] is used to simulate the nonlinear space plasma dynamics. The VAC is a general purpose hydrodynamic and magnetohydrodynamic software package. It uses modern shock capturing schemes with explicit or implicit time stepping on one, two, or three dimensional finite volume grids. Initial and boundary conditions and different source terms can be defined by parameters and/or user written subroutines. The VAC can be compiled with a Fortran 77, Fortran 90, or High Performance Fortran compiler and it runs efficiently on platforms ranging from personal computers and work stations to vector and parallel supercomputers.

Equations

Though the physical assumptions vary a lot among the applications, in most cases a conservative system of hyperbolic partial differential equations with additional non-hyperbolic source terms

$$\partial_t \mathbf{U} + \sum_i \partial_i \mathbf{F}_i(\mathbf{U}) = \mathbf{S}(\mathbf{U}, \partial_i \mathbf{U}, \partial_i \partial_j \mathbf{U}, \mathbf{x}, t)$$

has to be solved, where $\mathbf{U}$, $\mathbf{F}_i$, and $\mathbf{S}$ are the conservative variables, the fluxes and the source terms, respectively, and i represents the 1, 2, or 3 components of the spatial coordinate $\mathbf{x}$. In hydrodynamics, for example, $\mathbf{U}$ consists of mass, momentum and total energy densities, in MHD the magnetic field components are the additional variables. The source terms may be functions of $\mathbf{U}$, $\mathbf{x}$, and time t, e.g. for external gravitational forces, radiative cooling or heating terms etc. In case of viscosity, magnetic resistivity, thermal conduction, etc., $\mathbf{S}$ may also depend on the spatial derivatives of $\mathbf{U}$.

Using the conservative form of the equations for numerical discretization has two major advantages. First, the global conservation of mass, momentum, energy and magnetic induction is ensured, second, appropriate numerical schemes can find weak solutions with discontinuities. In astrophysical problems the flows are often supersonic or super Alfvénic, thus the proper representation of shock waves and other discontinuities is necessary. At the moment five modules are implemented, these are independent of each other, but the equations they describe can be regarded as special cases of the resistive MHD equations

$$\frac{\partial_\rho \mathbf{v}}{\partial_t} + \nabla \cdot (\mathbf{v}\rho) = \mathbf{S}_\rho$$

$$\frac{\partial_\rho \mathbf{v}}{\partial_t} + \nabla \cdot (\mathbf{v}\rho\mathbf{v} - \mathbf{BB}) + \nabla p_{tot} = \mathbf{S}_{\rho v}$$
$$\frac{\partial_e}{\partial_t} + \nabla \cdot (\mathbf{v}e + \mathbf{v}p_{tot} - \mathbf{BB}\cdot\mathbf{v}) = \nabla \cdot (\mathbf{B}\times\eta\mathbf{J}) + \mathbf{S}_e$$
$$\frac{\partial \mathbf{B}}{\partial_t} + \nabla \cdot (\mathbf{vB} = \mathbf{Bv}) = -\nabla \times (\eta\mathbf{J}) + \mathbf{S}_B, \tag{5.48}$$

where the total pressure, thermal pressure, and current density are

$$p_{tot} = p + \mathbf{B}^2/2$$
$$p = (\gamma - 1)(e - \rho\mathbf{v}^2/2 - \mathbf{B}^2/2)$$
$$\mathbf{J} = \nabla \times \mathbf{B}, \tag{5.49}$$

for an ideal gas with adiabatic index γ. Hence, the state space variables are:

- Density, ρ
- Momentum density in x- and y-direction, m_x, m_y
- Energy density, e
- Magnetic field in x- and y-direction B_x, B_y

and the outputs:

- Plasma velocity in x- and y-directions, $v_x = m_x/\rho$ and $v_y = m_y/\rho$
- Pressure, p_{tot}
- Magnetic field in x- and y-direction, B_x, B_y

5.4 Data Assimilation for Space Weather Forecast

5.4.1 Introduction

Although data assimilation techniques using Kalman filter have been used successfully for meteorologists and oceanographers in the last decade (some references [8, 14, 21, 28, 32, 34, 53, 58]), the space physics community has been slow in implementing such techniques. This is due to the fact that the space

systems are much more complex, larger, and there is a lack of measurements. In the last decades, astrophysicists have done a lot of research in developing models and tools to understand better the space plasma dynamics [66, 43, 44]. Anyhow, there are still many things to discover about this kind of systems and not much information is available. This is why researchers in this area often use truth recorded data, which have been taken from special events like big CMEs, to tune their models and improve the simulations. The problem arises when we want to use these models for real time predictions. There the conditions of the model simulation have to be updated on-line, and this is only possible by assimilating recent measurements into the model.

Even though there already exist good models that can describe the space plasma dynamics, applying data assimilation techniques to this system in real life is still complicated. Mainly, because there are still big computational constraints and a shortage of measurements. However, this situation is changing rapidly for the Ionosphere, for instance, and it is expected that it will also start changing for the magnetosphere in some decades. Therefore, one of the main goals of this thesis, is to make a first study of the performance of large scale sequential data assimilation techniques applied to space plasma, described by MHD systems.

5.4.2 Data Assimilation in a Linear 2D-MHD Flow System

The model we use in this section is taken from §5.3.1 using the Fourier collocation spatial discretization method. We consider a 20×20 grid with equidistant points (the grid points along the $\hat{\mathbf{e}}_x$ direction are the Fourier collocation points), where $0 < x < 2\pi$, $0 < y < 1$, and sample time $T_s = 10^{-3}$ seconds. The number of modes retained is $m = 5$ and hence, $A_d \in \mathbb{R}^{200\times200}$, and $C_d \in \mathbb{R}^{400\times200}$.

Although the system may be fully observable with just one measurement output, the discrete-time linearized system turns out to be marginally observable, because all the poles are clustered on the unit circle, which entails numerical round-off errors, see Figure 5.3. Figure 5.4 shows that at least 10 output measurements are needed to guarantee the full observability of the system. Hence, we use 50 measurements to ensure that the system is observable.

Figure 5.5 shows the eigenvalue positions of the error dynamic matrix for the classical KF, SSQRT-KF, RRSQRT-KF, and EnKF using 50 measurement points. There, we can compare the steady state performance of the suboptimal filter with the classical KF. As can be seen in Figure 5.5 the suboptimal

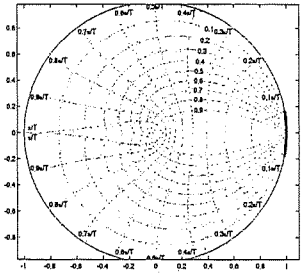


Figure 5.3: Eigenvalues of the A matrix. All the eigenvalues are located on the unit circle.

Kalman filter estimators are marginally stable, this is caused by the fact that eigenvalues of the pencil $\lambda M_1 - M_2$, with M_1 and M_2 defined in §2.3, are located very close to the unit circle, see Figure 5.6. Consequently, the performance of the suboptimal KF shows an oscillatory behavior as can be seen in Figures 5.7 and 5.8, and the convergence of the RMSE is bounded to a value bigger than

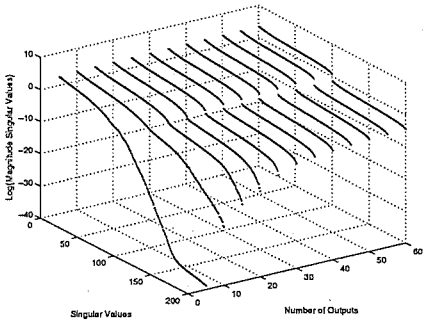


Figure 5.4: Logarithm of the singular values of the observability matrix for different number of measurements. We observe that the condition number of the observability matrix is high for a small number of measurements and the condition number is small when the number of measurements is larger than $\sim$ 10.

zero, even for the classical KF.

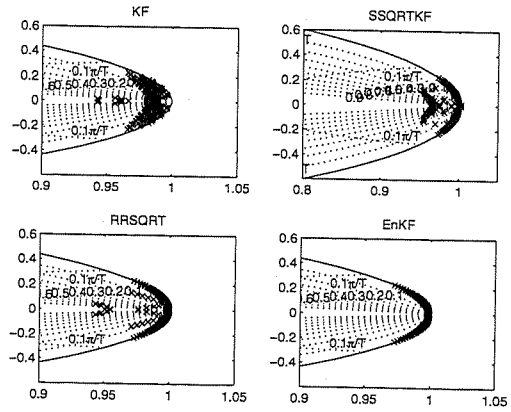


Figure 5.5: In the top-left eigenvalues of the discrete model, top-right, bottom-left, bottom-right, are the eigenvalues of matrix dynamics of the error for the classical KF, SSQRT-KF, and RRSQRT-KF, respectively. For the three methods we have chosen 50 output measurements. For RRSQRT-KF $q = 50$, and for SSQRT-KF $l = 50$.

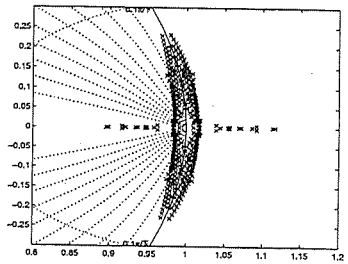


Figure 5.6: Eigenvalues of the pencil $\lambda M_1 - M_2$. It can be seen that some of the eigenvalues of the pencil are located on the unit circle. As a result the suboptimal Kalman filter estimators become marginally stable.

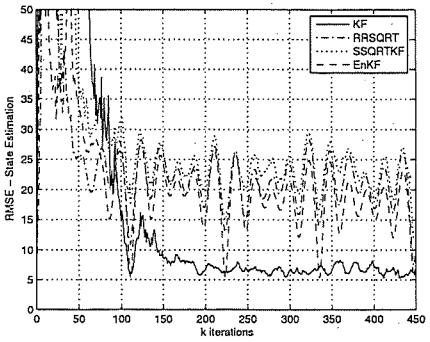


Figure 5.7: RMSE of the estimated states taking 50 measurements. Solid line KF, dash-dotted line SSQRT-KF, dotted line RRSQRT-KF, and dashed line EnKF. The SNR $= -3$dB for the measurement noise. It can be seen that even though the filters do not diverge, their estimations show an oscillatory behaviour.

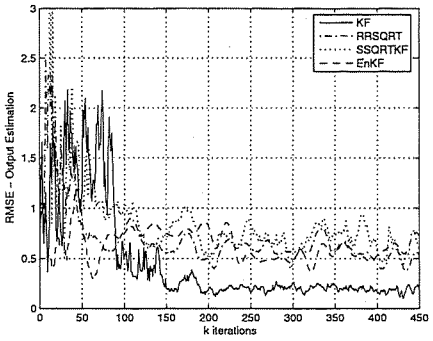


Figure 5.8: RMSE of the estimated outputs taking 50 measurements. Solid line KF, dash-dotted line SSQRT-KF, dotted line RRSQRT-KF, and dashed line EnKF. The SNR $= -3$dB for the measurement noise. It can be seen that even though the filters do not diverge, their estimations show an oscillatory behaviour.

5.4.3 Data Assimilation in a Non-Linear 2D-MHD Flow System

In order to investigate how sequential data assimilation techniques could operate in space weather forecast, we simulate a 2D-MHD system which emulates the Earth's magnetosphere. Hence, we took an ideal MHD system, equations (5.1)-(5.6), and (5.8), setting the boundary conditions like in Table 5.1, so that it simulates a bow shock around the Earth (see Figures 5.1, 5.9). This system was simulated using the VAC code [79].

Setting the VAC code

In order to simulate the bow shock in an ideal MHD system, we set the following parameters in the VAC code:

- Grid size: 34 x-direction $\times$ 54 y-direction
- Initial conditions of the state space variables:
  - Mass density, $\rho = 1.0$ Kg/m^3
  - Momentum density in x- and y-directions, $m_x = 20$, $m_y = 0$ kg/m^{2}s
  - Energy density, $e = 115.0$ kg/ms^2
  - Magnetic field in x- and y-directions, $B_x = 0$ mT, $B_y = 1.0$ mT
- ratio of specific heats, $\gamma = 5/3$
- Simulation sampling time, $T_s = 1 \times 10^{-4}$ seconds
- Data assimilation sampling time, $10T_s$ seconds
- Spatial discretization method: Total Variation Diminishing Lax-Friedrich - TVDLF, using the Powell's scheme to keep $\nabla \cdot B = 0$.

Consequently, the order of the system is $n = 11016$, 6 state space variables and 34×54 grid-points. To excite the system, a square sinusoidal wave for B_y varying from 1 to 1.5 mT, and m_x from 20 to 30 kg/m^2s, were generated at the left-hand boundary, simulating a magnetic storm. The profile of the magnetic field and momentum density excitations are shown in Figure 5.10. Note that the in first 50 iterations the amplitude of the perturbations is zero, this in order to let develop the multivariate correlations of the estimated error covariance matrix.

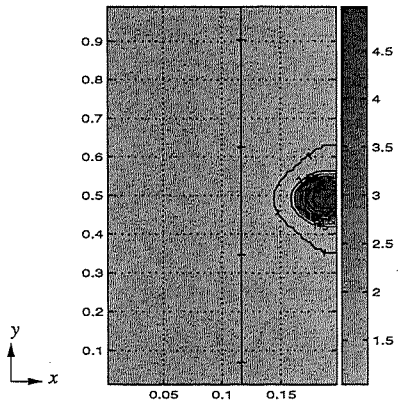


Figure 5.9: Initial conditions of the magnetic field magnitude [mT].

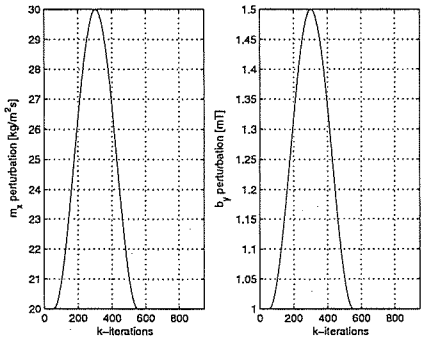


Figure 5.10: Profile of the magnetic field B_y and momentum density m_x perturbations at the left-hand boundary. 1 iteration = 1 Ts.

	Left-hand	Right-hand Out bow shock	Right-hand Inside bow shock	Top	Bottom
ρ	fixed	open	symmetric	open	open
v_x	fixed	open	asymmetric	open	open
v_y	fixed	open	symmetric	open	open
p	fixed	open	symmetric	open	open
B_x	fixed	open	asymmetric	open	open
B_y	fixed	open	symmetric	open	open

Table 5.1: Settings of the boundary conditions used in the VAC code to simulate a bow shock.

Data Assimilation System Implementation

Figure 5.11 depicts a general scheme of an ensemble filter implementation by running Matlab and the VAC code simultaneously. As we can see in the scheme, we did not need any extra code for the numerical model, however, we had to write the algorithms for each of the ensemble filters in Matlab, and the interface to read and write data between Matlab and the VAC code which is written in Fortran. As a result, we have got a modular data assimilation system where the nonlinear model integration module is executed by the VAC code, while the data assimilation module by Matlab. This is another motivation for using this sort of filters, because it is quite straightforward to deploy, and the results are reliable.

Another important problem is the localization of the measurement points. As it is known from system theory, the observability of a distributed system depends among other issues on the location of these points. The study of the dependence of the observability on the location of the measurement points for distributed MHD systems goes farther off the goals of this thesis, therefore, we do not go deeper in this topic, and we rely in very practical assumptions. In this example we have chosen the place for most of the these points in areas where most of the dynamics effects happen, that is, around the Earth, right-hand side, and the rest radially distributed taking the Earth's position as the reference point.

For the computations a Fedora Core 3 Linux machine was used with the following features:

- Model name: Pentium 4 (dual).
- CPU: 2.80 GHz.
- RAM: 1 GB.

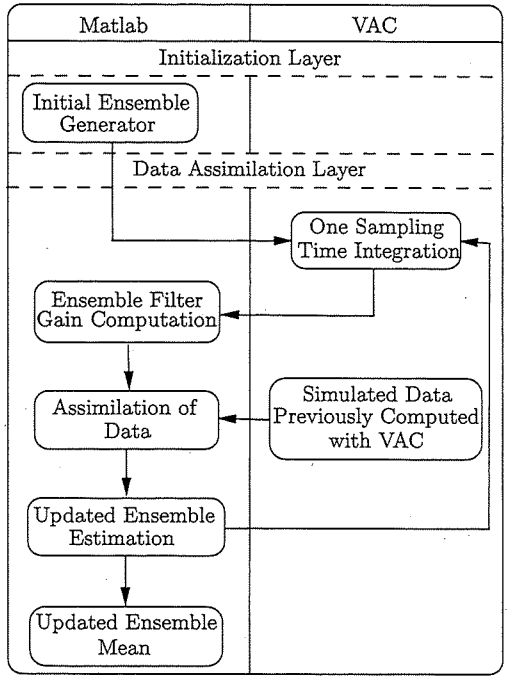


Figure 5.11: Data assimilation implementation scheme. At the left the modules computed with Matlab, and at the right, the ones computed with the VAC code. From top to bottom, first the initialization layer, where the initial ensemble is generated in Matlab. Then, in the data assimilation layer, the initial ensemble is integrated using the VAC code. This updated ensemble is send to Matlab, where it is used to computed the ensemble Kalman gain. Afterwards, the observations, previously computed with VAC code and stored as a data Matlab file, are assimilated using the ensemble Kalman filter gain. As a result, the estimated state vector is computed by taking the mean value of the updated ensemble. Finally, the updated ensemble is integrated with the VAC code and the procedure is repeated again.

The compilation of the VAC code was done using the GNU compiler g77, and the Matlab used was the version 7.0 for Linux.

	N	$m = 6$	$m = 22$
EKF	-	8×10^{12} flops+1ME	8×10^{12} flops+1ME
EnKF	50	2.9×10^{6} flops+50ME	1.4×10^{7} flops+50ME
	200	7.9×10^{6} flops+200ME	1.9×10^{7} flops+200ME
	500	1.8×10^{7} flops+500ME	2.9×10^{7} flops+500ME
EnSRF	50	2.1×10^{7} flops+50ME	6.7×10^{7} flops+50ME
	200	8.4×10^{7} flops+200ME	2.6×10^{8} flops+200ME
	500	2.1×10^{8} flops+500ME	6.5×10^{8} flops+500ME
ETKF	50	3.1×10^{7} flops+50ME	4.5×10^{7} flops+50ME
	200	4.6×10^{8} flops+200ME	5×10^{8} flops+200ME
	500	2.9×10^{9} flops+500ME	3×10^{9} flops+500ME
EnSLKF	50	4.5×10^{6} flops+50ME	3.5×10^{7} flops+50ME
	200	9.5×10^{6} flops+200ME	4×10^{7} flops+200ME
	500	1.9×10^{7} flops+500ME	5×10^{7} flops+500ME

Table 5.2: Comparison of the number of flops between the EKF, EnKF, EnSRF, ETKF, EnSLKF for the nonlinear 2D-MHD flow system example, with N the number of ensemble members, m the number of measurement points, and ME the number of model evaluations.

Analysis of Results

To investigate the performance of the ensemble-based filters, namely, the EnKF, the EnSRF, the EnSLKF, and the ETKF, in MHD systems a magnetic storm coming towards the Earth was simulated by setting the boundary conditions as described in Table 5.1. We studied three cases: 6, 13, and 22 measurement points, where we assumed that all the output variables, namely, velocity, pressure, and magnetic field (see §5.3.2) were available, and for each case we used 50, 200, and 500 ensemble members, respectively. The initial ensemble was generated as indicated in Appendix D, taking as first guess the same initial conditions of the simulated system for all the state variables.

Table 5.2 shows a comparison of the number of flops needed for the filters studied. There can be seen that the EnSRF and ETKF are the most sensitive to the number of ensemble members. On the other hand, even though the number of flops of the EKF $\mathcal{O}(10^{12})$ is higher than any of the ensemble based filters $\sim \mathcal{O}(10^{8})$, the ensemble based filters need more model evaluations, consequently, the computational complexity between the two techniques is very similar. Finally, the influence of m in the number of flops is not observed in this example, because m does not change very much. As a result, the number of ensemble members N is the critical parameter for the computational

complexity of the filters in this example.

Figures 5.12-5.14, and 5.15-5.20 show a comparison of the global performance of the filters for the magnetic field magnitude, velocity magnitude, and pressure (see appendix E for more details). In general, it can be seen that the best global performance of the filters is obtained when the number of ensemble members as well as the number of measurement points are increased, as expected. It is also clearly seen that 50 ensemble members are not enough to describe the error covariance matrix, because the performance of the filters is very poor. On the other hand, in spite of having some small differences in the global RMSE, we can say that the EnKF, the EnSRF, and the ETKF behave very similarly. In contrast, at first sight the EnSLKF seems to have a poor performance; however, when we look at the results in detail, Figures 5.12-5.14, we find some interesting features: first of all, notice that it is very stable, it never makes crash the numerical model as happens with the EnSRF and the ETKF when there are a small number of ensemble members and measurement points. Another interesting characteristic, is that it needs less ensemble members than the other filters to represent the error covariance matrix. This is because the filter carries out a local optimization around the measurement points, so, the error covariance matrix is constrained to these regions. And finally, by choosing an smooth function for the weighting matrices Γ_k and M_k, as explained in §3.4, we avoid numerical instabilities in the model simulator. We have to stress that by choosing smooth functions around the measurement points to build up the matrices Γ_k and M_k, what we are basically doing is applying covariance localization §4.4.2 in an optimal way.

As a result, under circumstances where the number of measurement locations is small compared to the size of the system, the EnSLKF is very robust numerically, and the computational cost is less than the cost of classical ensemble-based Kalman filters. This is more clearly seen when we compare the performance of the filter for the cases of 200 and 500 ensemble members in the magnetic field magnitude, Figure 5.12. Also in Figures 5.15, 5.18 can be seen how the EnSLKF focuses the estimation close to the measurement points without generating any instabilities in locations far away. On the contrary, the other filters try to make estimations in the whole grid, with the consequence that in places distant to the measurement points the estimated values yielded are physically wrong, this is called filter divergence. Hence, the numerical model becomes unstable and it may crash.

As it was mentioned in §4.4.2 one way to avoid the filter divergence is to use the covariance localization approach, often cited in the weather forecast community [50]. Therefore, the following case we are going to study focuses

on the comparison of performance between the ensemble-based Kalman filters using the same covariance localization matrix and the EnSLKF. Figures 5.21-5.14 show a comparison of the global RMSE for all the filters. As we observe, the covariance localization has to be chosen with care otherwise it can cause filter divergence or degrade drastically the performance of the estimation, as we can see in the EnKF and EnSRF cases, respectively. The ETKF shows a much better performance similar to the EnSLKF. This might happen because this filter propagates the error covariance matrix using an eigenvalue decomposition of a matrix that contains the prior error covariance matrix, so it becomes robust against instabilities that can be generated for the covariance localization step. The point we have to stress here, is that in spite of the covariance localization being a logical approach to deal with sparse measurements in large scale systems, the solution as proposed by [50] is not the best one, and it is very heuristic; therefore the results are not reliable at all. In contrast, we have shown that the EnSLKF gives the optimal solution in the quadratic sense, and the results are very reliable, as depicted in the analyses.

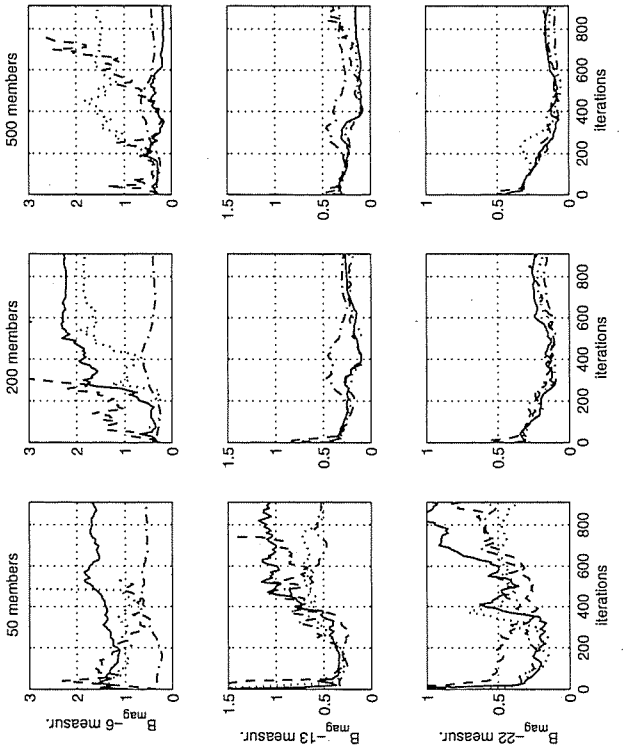


Figure 5.12: RMSE comparison of the magnetic field magnitude. Solid line the EnKF, dashed line the EnSRF, dash-dotted line the EnSLKF, and dotted line ETKF. In general, it can be seen that the best global performance of the filters is obtained when the number of ensemble members as well as the number of measurement points are increased. Also, observe that 50 ensemble members are not enough to describe the error covariance matrix. Furthermore, it is important to highlight the good performance of the EnSLKF compared to the other filters with six measurement points. Finally, note that the EnSRF for the case of 6 measurement points make crash the simulator, this is also observed with the ETKF for 50 ensemble members with 6 and 13 measurement points. This shows how relevant turns out to be the local estimation when we have sparse measurement points. Otherwise, the estimator introduces instabilities to the numerical model degrading its performance, and increasing the risk of aborting the simulation, something no desirable in a data assimilation system. 1 iteration = 1 Ts.

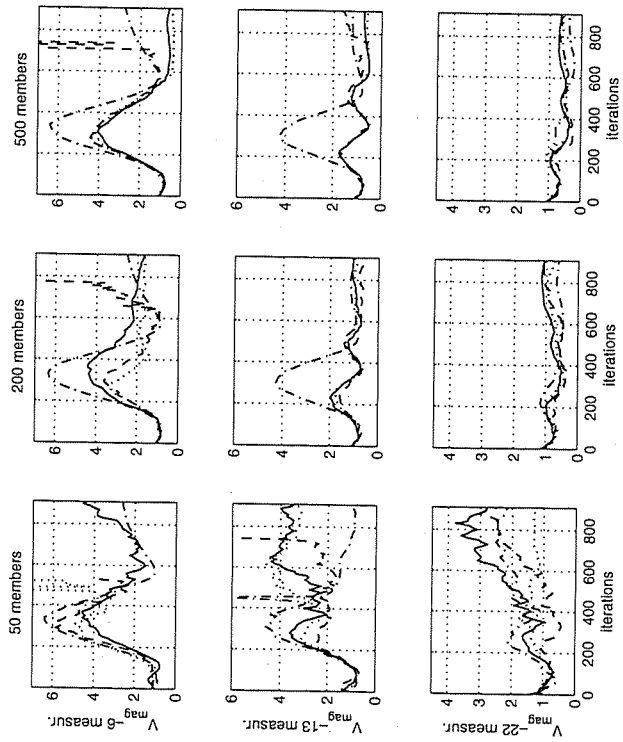


Figure 5.13: RMSE comparison of the velocity magnitude. Solid line the EnKF, dashed line the EnSRF, dash-dotted line the EnSLKF, and dotted line ETKF. In general, it can be seen that the best global performance of the filters is obtained when the number of ensemble members as well as the number of measurement points are increased. Also, observe that 50 ensemble members are not enough to describe the error covariance matrix. In this case, the performance of the EnSLKF is very similar to the other filters. The big peak show at 300 iterations for 6 and 13 measurement points is caused by the fact that the perturbation has not arrived to the measurement points yet. Finally, note that the EnSRF for the case of 6 measurement points make crash the simulator, this is also observed with the ETKF for 50 ensemble members with 6 and 13 measurement points. This shows how relevant turns out to be the local estimation when we have sparse measurement points. Otherwise, the estimator introduces instabilities to the numerical model degrading its performance, and increasing the risk of aborting the simulation, something no desirable in a data assimilation system. 1 iteration = 1 Ts.

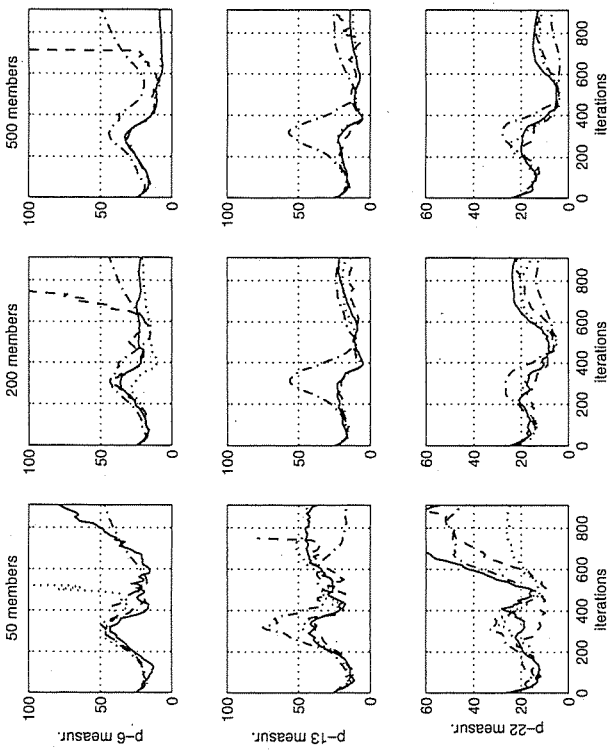


Figure 5.14: RMSE comparison of the pressure. Solid line the EnKF, dashed line the EnSRF, dash-dotted line the EnSLKF, and dotted line ETKF. In general, it can be seen that the best global performance of the filters is obtained when the number of ensemble members as well as the number of measurement points are increased. Also, observe that 50 ensemble members are not enough to describe the error covariance matrix. In this case, the performance of the EnSLKF is very similar to the other filters. The big peak show at 300 iterations for 6 and 13 measurement points is caused by the fact that the perturbation has not arrived to the measurement points yet. Finally, note that the EnSRF for the case of 6 measurement points make crash the simulator, this is also observed with the ETKF for 50 ensemble members with 6 and 13 measurement points. This shows how relevant turns out to be the local estimation when we have sparse measurement points. Otherwise, the estimator introduces instabilities to the numerical model degrading its performance, and increasing the risk of aborting the simulation, something no desirable in a data assimilation system. 1 iteration = 1 Ts.

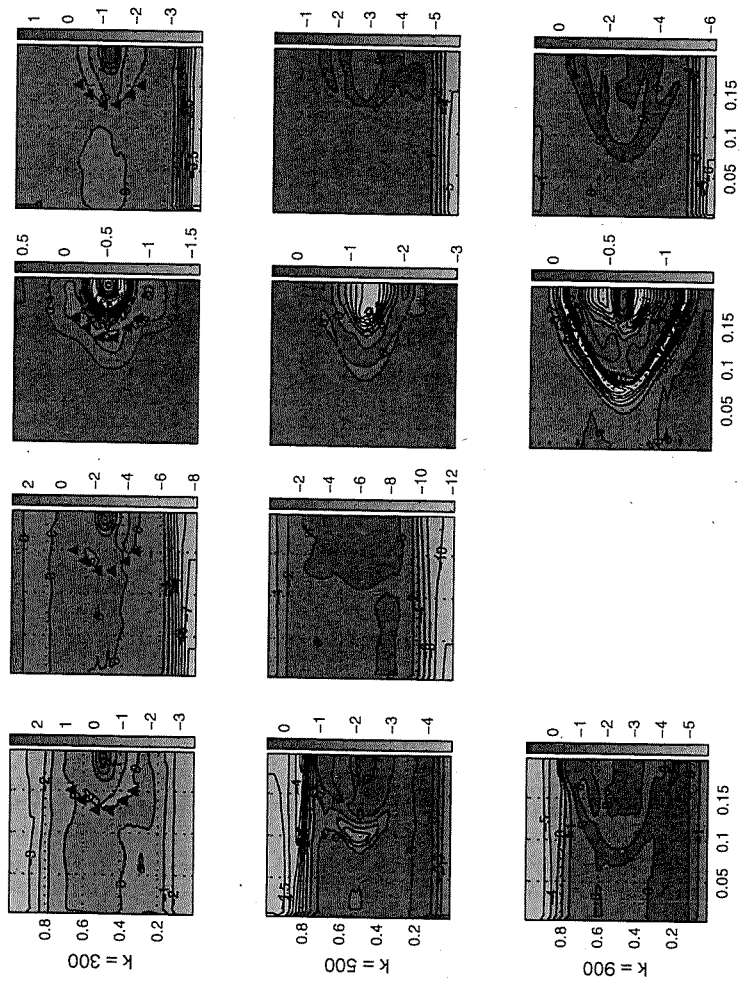


Figure 5.15: Magnetic field magnitude residuals [mT] with 6 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4} s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points. The empty figure space means that the algorithm has crashed, something that is no desirable in a data assimilation system, as explained in the analysis of the results.

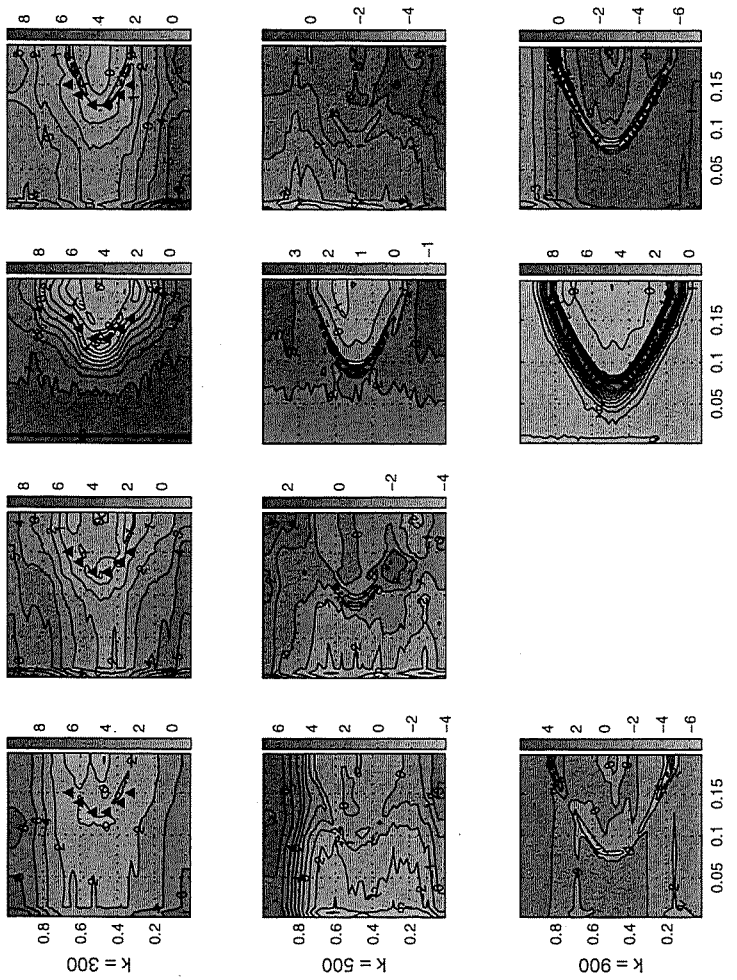


Figure 5.16: Velocity magnitude residuals [m/s] with 6 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points. The empty figure space means that the algorithm has crashed, something that is no desirable in a data assimilation system, as explained in the analysis of the results.

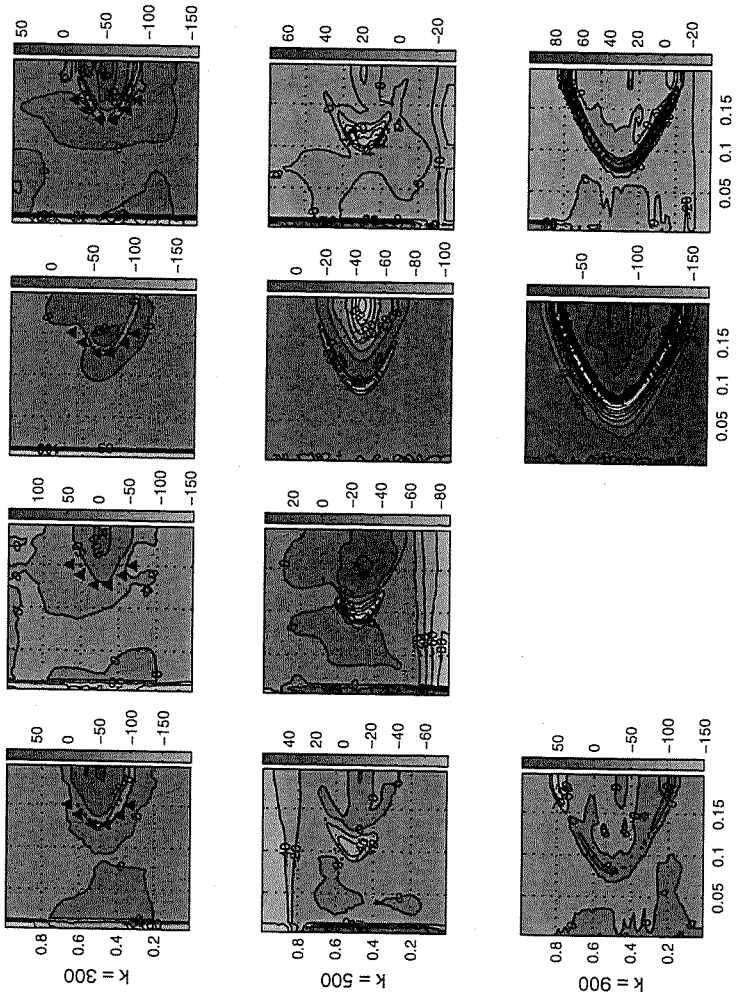


Figure 5.17: Pressure residuals [kg/ms^2] with 6 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points. The empty figure space means that the algorithm has crashed, something that is no desirable in a data assimilation system, as explained in the analysis of the results.

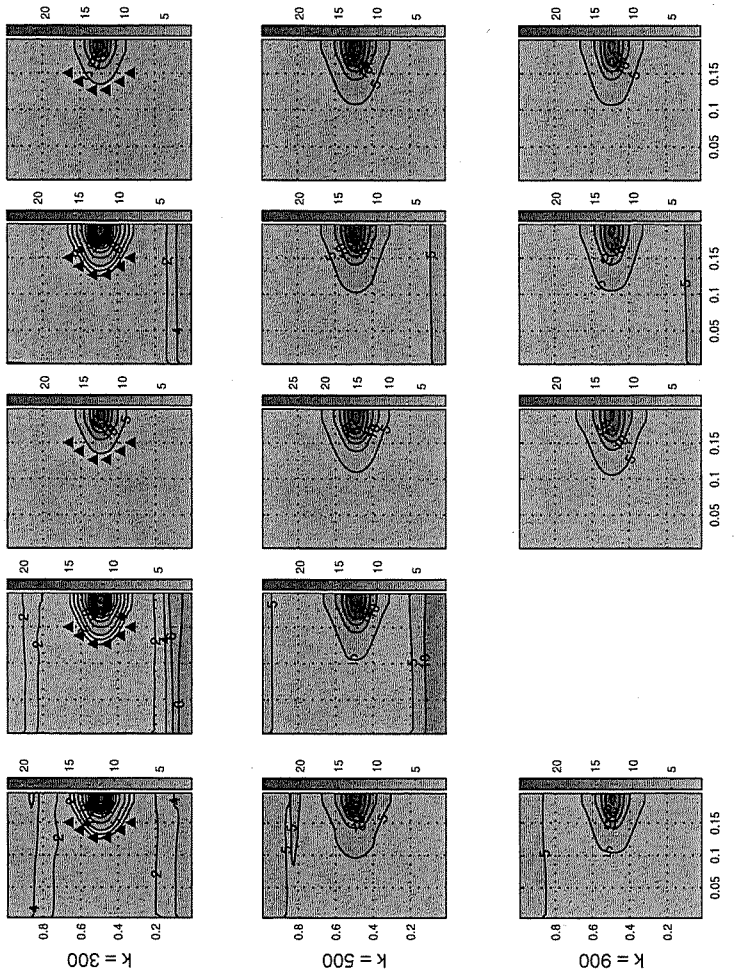


Figure 5.18: Magnetic field magnitude [mT] with 6 measurements points and 200 ensemble members. From top to bottom, the simulate data, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points. The empty figure space means that the algorithm has crashed. something that is no desirable in a data assimilation system, as explained in the analysis of the results.

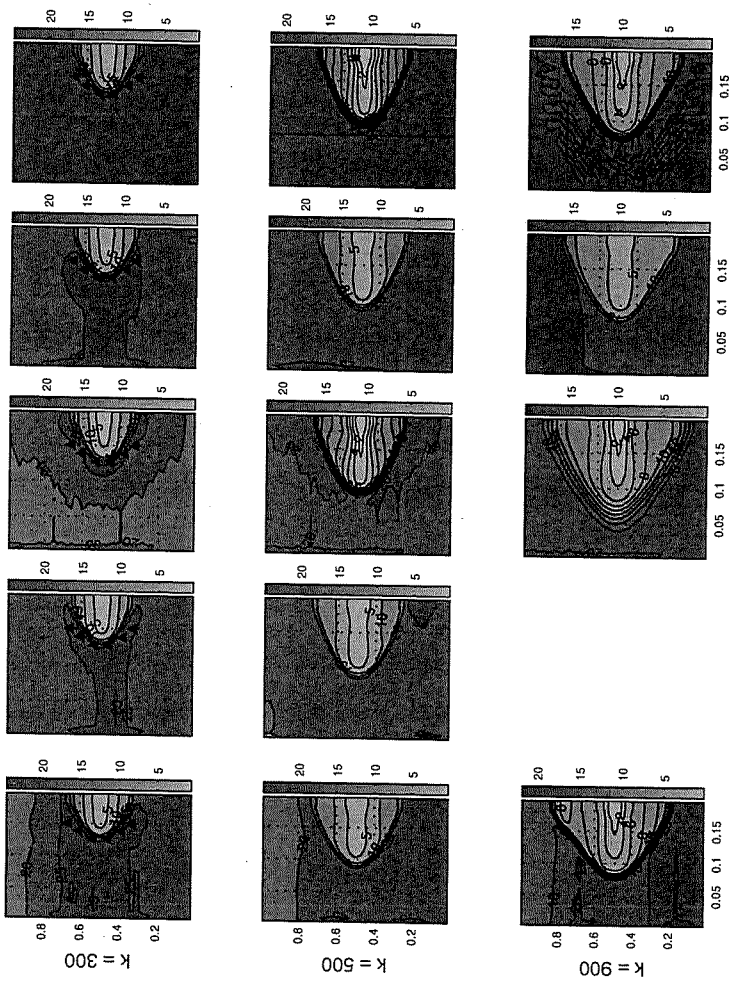


Figure 5.19: Velocity magnitude [m/s] with 6 measurements points and 200 ensemble members. From top to bottom, the simulate data, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points. The empty figure space means that the algorithm has crashed. something that is no desirable in a data assimilation system, as explained in the analysis of the results.

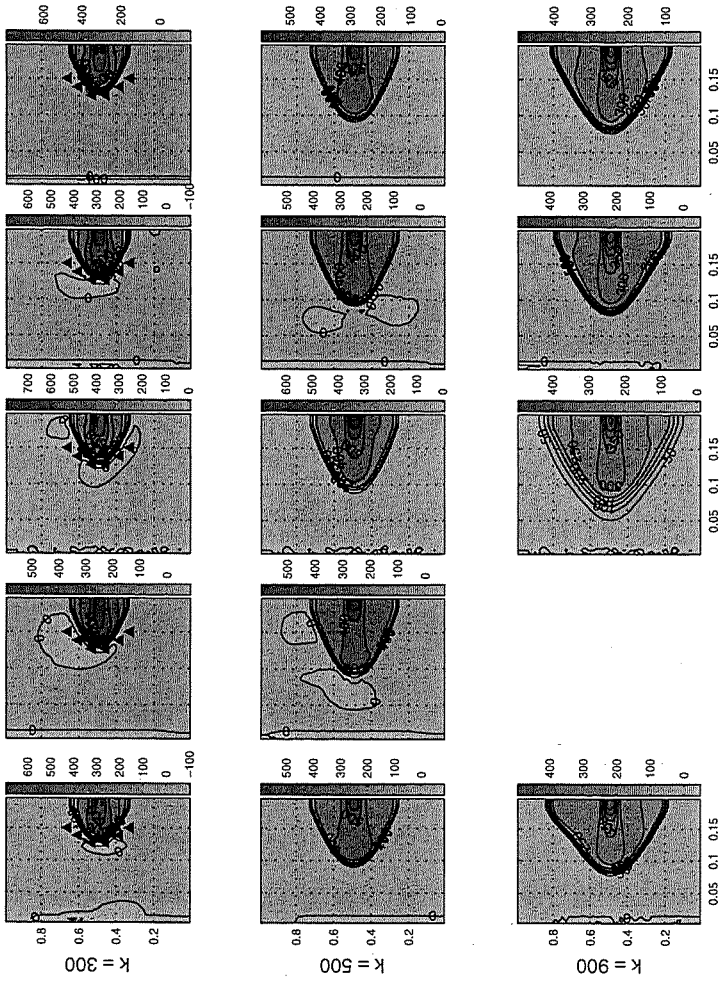


Figure 5.20: Pressure [kg/ms^2] with 6 measurements points and 200 ensemble members. From top to bottom, the simulate data, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4} s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points. The empty figure space means that the algorithm has crashed. something that is no desirable in a data assimilation system, as explained in the analysis of the results.

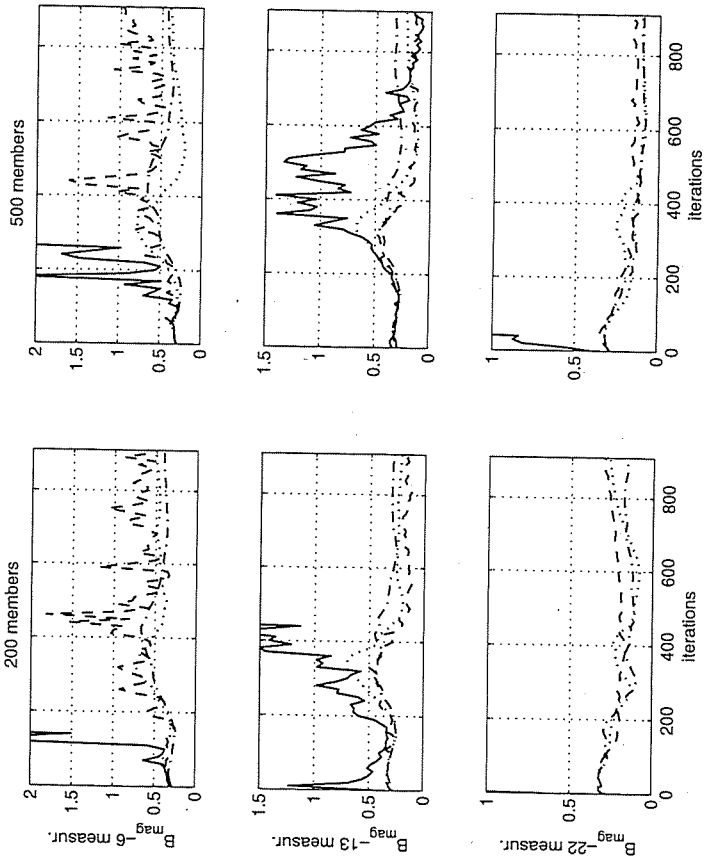


Figure 5.21: RMSE comparison of the magnetic field magnitude. Solid line the EnKF, dashed line the EnKF with covariance localization, dotted line the EnSRF with covariance localization, and dash-dotted line the EnSLKF. Note how the covariance localization approach can cause filter divergence and degrades the performance of a filter if it is not well defined, see the EnKF, something that may be difficult to achieve in a heuristic method. On the other hand, the performance of the EnSLKF is similar to the other filters with covariance localization; therefore, we can say that the EnSLKF is an optimal method to do covariance localization. 1 iteration = 1 Ts.

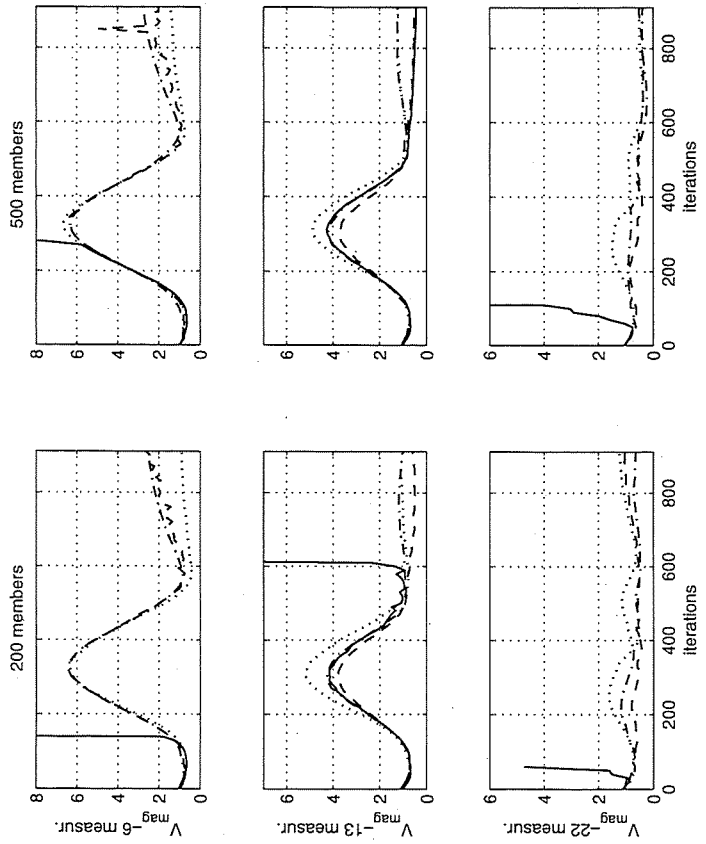


Figure 5.22: RMSE comparison of the velocity magnitude. Solid line the EnKF, dashed line the EnKF with covariance localization, dotted line the EnSRF with covariance localization, and dash-dotted line the EnSLKF. Note how the covariance localization approach can cause filter divergence and degrades the performance of a filter if it is not well defined, see the EnKF, something that may be difficult to achieve in a heuristic method. On the other hand, the performance of the EnSLKF is similar to the other filters with covariance localization; therefore, we can say that the EnSLKF is an optimal method to do covariance localization. 1 iteration = 1 Ts.

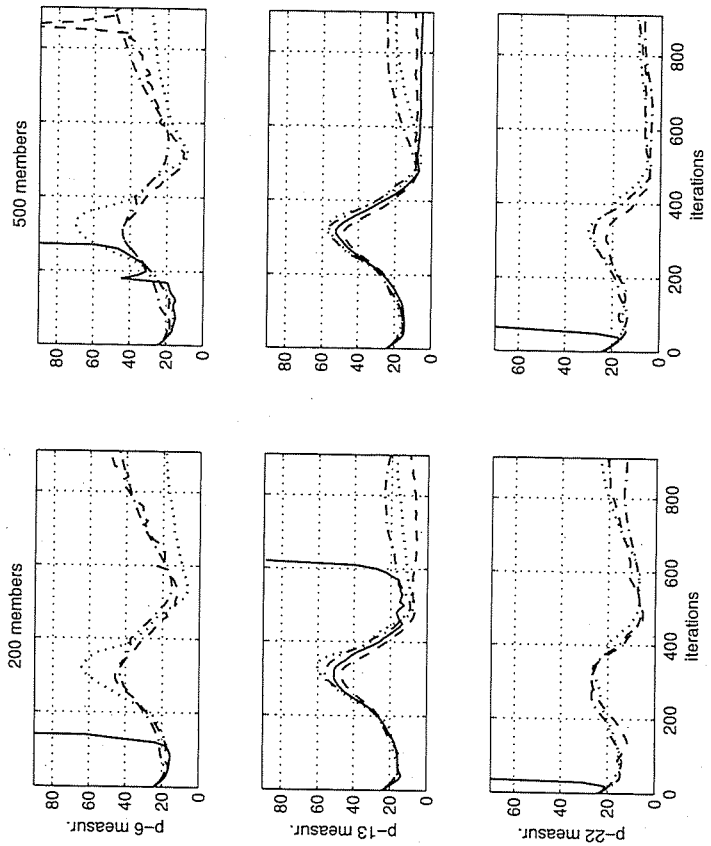


Figure 5.23: RMSE comparison of the pressure. Solid line the EnKF, dashed line the EnKF with covariance localization, dash-dotted line the EnSRF with covariance localization, and dash-dotted line the EnSLKF. Note how the covariance localization approach can cause filter divergence and degrades the performance of a filter if it is not well defined, see the EnKF, something that may be difficult to achieve in a heuristic method. On the other hand, the performance of the EnSLKF is similar to the other filters with covariance localization; therefore, we can say that the EnSLKF is an optimal method to do covariance localization. 1 iteration = 1 Ts.

5.5 Conclusions

In this chapter we have studied the performance of data-assimilation techniques in MHD systems.

In the first part, a discrete-time linearized model for a flow of plasma in a two-dimensional channel, without taking into account any dissipation, was obtained. Hence, this model was marginally stable, with all the poles clustered on the unit circle. This entailed numerical round-off errors making the system difficult to observe.

Four different state space observers were studied, namely, the Kalman filter, the RRSQRT-KF filter, the SSQRT-KF, and the EnKF filter. Although the Kalman filter performed better than the suboptimal Kalman filters, as expected, it converged very slowly. This is caused by the fact that the eigenvalues of the pencil $\lambda M_1 - M_2$ are located very close to the unit circle. This fact also had a big impact on the performance of the suboptimal KF studied in this section, so that the filters became marginally stable, and the estimation error was bounded. Hence, the estimation error showed an oscillatory behavior.

As a result, we could not exploit the advantages of the proposed suboptimal Kalman filters for the linearized MHD system. Since the numerical round-off error problem is mainly related to the spatial discretization techniques, another alternatives have to be investigated in order to avoid the numerical and observability problems. This issue is beyond the scope of this thesis, so, we leave it open for future research.

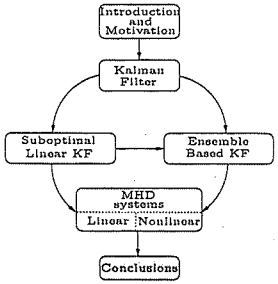
In the second part, a modular data-assimilation scheme was presented. Using this scheme we investigated the performance of ensemble-based Kalman filters in MHD systems, taking as case study the space weather forecast. Three well known approaches, namely, the EnKF, the EnSRF, and the ETKF, were compared to the one we proposed, the EnSLKF. In general, the results showed that when there is not a lack of measurements, and we take enough ensemble members, in this case study 200 was enough, the filters converged. But this performance degrades drastically when we reduced the number of measurement points, 6 out of 1836 in the example. Under these circumstances, which are very realistic, we found that the EnSLKF performed more robust numerically, because it just optimizes the estimation around the measurement points, contrary to the other filters which try to reduce the error in the whole grid, generating instabilities to the numerical model. Additionally, it needs less ensemble members to describe the error covariance matrix. This is because it constraints this matrix to regions close to the measurement points. Hence, the

computational cost can be reduced dramatically; first, by using a reduced number of ensemble members, and second, by making use of the sparsity generated by the local optimization.

Finally, the EnSLKF was compared with the covariance localization approach proposed by [50] to deal with sparse measurements in large scale system, and we found that the EnSLKF performs much better and reliable than covariance localization. This conclusion is supported by the results obtained in the case study, and also by the fact that the EnSLKF has been obtained from an optimization problem, contrary to the covariance localization approach which is a heuristic method.

Chapter 6

Conclusions and Future Research



6.1 Conclusions

This thesis describes the application of data-assimilation techniques to large scale environmental systems described by PDEs (partial differential equations), where the ultimate goal is to introduce these techniques in the field of space weather forecast. It turns out that space plasma system dynamics, used for space weather forecast, can be described by magnetohydrodynamic (MHD) systems. Additionally, they are very large scale, and suffer of the lack of measurements. The research focused on two directions; first, to investigate the performance of sequential data-assimilation techniques to estimate the dynamics of the space plasma in steady state, which can be done using a linearized

MHD system. And second, the most important, to investigate the performance of sequential data-assimilation techniques for space weather forecast, which can be carried out using a nonlinear MHD system.

For the first case, in Chapters 2 and 3, we made a review of optimal and suboptimal linear Kalman filter formulations and algorithms that could be applied to this case. There, we proposed a new algorithm, the singular square root algorithm (SSQRT-KF), which originally was designed for estimation in systems where the process noise is negligible, but we extended it to other cases by applying a heuristic technique called covariance inflation. This algorithm was compared to other suboptimal Kalman filters for state and boundary conditions estimation in a 2D heat transfer system with good results. Then in Chapter 5, a linearized 2D-MHD flow system in state space was yielded by using two spatial discretization techniques, specifically, the Chebyshev and Fourier Collocation. This linearized MHD system resulted to be marginally observable, causing the suboptimal Kalman filters to be marginally stable. As a result, we observe a low rate of convergence, and in most of the cases an oscillatory behavior. Therefore due to these numerical issues, we found that data assimilation is hard to apply. Nevertheless, this problem could be solved by exploring other discretization techniques to avoid the observability problems.

For the second case, we focused on three aspects: 1) numerical efficiency and stability, 2) sparsity of the measurements, and 3) facility and reliability in the deployment of the data assimilation system. Taking into account these three issues, we found out that ensemble-based data assimilation techniques are the best option. This is supported by the fact that they have been shown to work quite well in weather forecast systems, which has some similarities to space weather systems. Moreover, they are easy to implement and robust because they can be treated as modular systems, where each module is specialized in its task, it means, the system integration is done by a nonlinear numerical model, and you do not have to care about any modeling or linearization, and the filtering can be coded apart for the user. Then, the modules are run in parallel to assimilate the data.

Therefore, taking this as a basis, and having in mind that the best estimation we can get in a large scale system with sparse measurements is a local estimation, we focused on developing a filter that could handle this problem. Hence, in Chapter 3, we introduced a new optimal Kalman filter, the spatially localized Kalman filter (SLKF) which can deal with this problem without generating instabilities to the numerical model, in other words, it avoids filter divergence. We tested it in a 1D mass-spring system, the dynamical structure of which is similar to the one we get from a discretized environmental system. Chapters

3 and 4 are also devoted to introducing and studying some of the more relevant ensemble-based data assimilation techniques and algorithms, namely, the EnKF, the EnSRF, and the ETKF. Also in Chapter 4, we extended the SLKF to the nonlinear case based on the EnKF, which we called the EnSLKF.

Then in Chapter 5, we investigated and compared the performance of the ensemble-based Kalman filters. For this study we used the VAC code as numerical model, where we emulate a solar storm coming towards the Earth. We set a very simple and small 2D-MHD flow system with $34 \times 54 = 1836$ grid points, and six state space variables, so that the order is equal to 11016. Generally speaking, it has been shown that when the ensemble-based filters have got a lot of information from the real system (measurements), and a large ensemble is taken, the estimation error converges to zero in the ideal case. However, in conditions of a reduced number of measurements or ensemble members, the filters may diverge or have a poor performance. In contrast, the EnSLKF showed to be more robust than the global filters in this situation, because it optimizes the estimation around the measurement points. This has some advantages. First, the number of ensemble members needed to represent the error covariance matrix is smaller because it constraints this matrix to the regions close to the measurement points. And second, the fact that the error covariance matrix is constrained to certain regions, generates sparsity in the matrix. Hence, this sparsity can be utilized to reduce the computational load of the filter.

On the other hand, the EnSLKF was compared with the covariance localization approach proposed by [50] to deal with sparse measurements in large scale systems, and we found that the EnSLKF performs much better and reliably than covariance localization. This conclusion is supported by the results obtained in the case study, and also by the fact that the EnSLKF has been obtained from an optimization problem, contrary to the covariance localization approach, which is a heuristic method.

Finally, we have introduced and tested the sequential data-assimilation topic in the area of space weather forecast with very promising results. However, there is still a long way to go before deploying the first real data-assimilation system in this field.

6.2 Future Research

As I mentioned in the conclusions, there are a lot problems and open issues that have to be solved before a data-assimilation system for space weather forecast

becomes functional.

One of the open issues in the Kalman filter is how to define the process noise matrix of a system. One possible solution could be using subspace identification, where the idea is to compute a model where the outputs can be defined as the error between the numerical model and real measurements. Then, the measurement noise matrix obtained by the identification will give us a good hint of the process noise of the system.

Another important aspect of the Kalman filter, is that we always assume that the forecast variables are Gaussian, although it is known that the variance is no longer optimal in the non-Gaussian case. Therefore, the use of alternative Monte Carlo methods for the update step can be investigated. For instance, importance sampling could be used to sample from the posterior distribution given by Bayes' formula. The algorithm could be made sequential and an additional resampling step added to prevent algorithm degeneracy. These Sequential Importance Resampling (SIR) methods are more general than Kalman filtering since they do not need to assume Gaussianity [83].

In the implementation of a data-assimilation system, we have to take into account that the size of the real system is really large, hence, we have to investigate how to parallelize the computation of the ensemble-based Kalman filters together with the numerical models in order to speed up the efficiency of the system.

Furthermore, as one of the major problems in applying data assimilation to space weather forecast is the shortage of measurements, it would be also interesting to investigate how to take direct and indirect measurements of the space plasma. Although this issue is changing in the Ionosphere, when we go to higher layers this becomes very difficult and expensive.

Finally, In the space plasma modeling, we can explore the use of data-assimilation techniques off-line for parameter estimation in order to improve the tuning of the existing model. One possibility would be the ensemble smoother Kalman filter.

On the other hand, the ensemble-based Kalman filter theory developed in the weather forecast community in the last decade, could be used in systems and control community to solve nonlinear control and observer problems. To mention some examples where this theory could be applied, we can think about biochemical, petrochemical, and aerospace systems.

Bibliography

[1] O. M. Aamo and M. Kristic. *Flow Control by Feedback.* Springer, 2003.

[2] W. S. Agee and R. H. Turner. Triangular decomposition of a positive definite matrix plus a symmetric dyad, with applications to Kalman filtering. *White Sands Missile Range Tech. Rep. No. 38*, Oct 1972.

[3] B. D. O. Anderson. Second order convergent algortihms for the steady-state riccati equation. *Int. J. Control*, 28:295–306, 1978.

[4] B. D. O. Anderson and J. B. Moore. *Optimal Filtering.* Prentice-Hall, 1979.

[5] J. L. Anderson. An ensemble adjustment Kalman filter for data assimilation. *Monthly Weather Review*, 129:2884–2903, 2001.

[6] J. L. Anderson and S. L. Anderson. A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127:2741–2758, 1999.

[7] A. Andrews. A square root formulation of the Kalman covariance equations. *Journal of AIAA*, 6:1165–1166, 1968.

[8] J. Ballabrera-Poy, A. J. Busalacchi, and R. Murtugudde. Application of a reduced-order Kalman filter to initialize a coupled atmosphere-ocean model: Impact on the prediction of el niño. *Journal of Climate*, 14:1720–1737, 2001.

[9] O. Barrero Mendoza, D. S. Bernstein, and B. L. R. De Moor. Spatially localized Kalman filtering for data assimilation. Internal Report TR 04-156, Katholieke Universtiteit Leuven, ESAT/SCD-SISTA, September 2004.

[10] O. Barrero Mendoza, D. S. Bernstein, and B. L. R. De Moor. Spatially localized Kalman filtering for data assimilation. In *Proceedings of the 24th American Control Conference - ACC 2005*, Portland, Oregon, June 2005. IFAC. Accepted.

[11] O. Barrero Mendoza and B. L. R. De Moor. A singular square root filter algorithm for large scale data assimilation. In *Proceedings of the 15th IASTED Conference in Modelling and Simulation*, Marina del Rey, CA., USA, March 2004.

[12] O. Barrero Mendoza, B. L. R. De Moor, and D. S. Bernstein. Data assimilation for magneto-hydrodynamics systems. *Journal of Computational and Applied Mathematics, special edition, proceedings ICCAM 2004*, 2004. In press.

[13] O. Barrero Mendoza, B. L. R. De Moor, and P. Van Dooren. Kalman filter for data assimilation in linear systems with negligible process noise: State and boundary conditions estimation. Internal Report TR 03-161b, Katholieke Universtiteit Leuven, ESAT/SCD-SISTA, August 2004.

[14] A. F. Bennet. *Inverse Methods in Physical Oceanography.* Cambridge University Press, 1992.

[15] J. M. Bennet. Triangular factors of modified matrices. *Numerische Mathematik*, 7:217–221, 1963.

[16] D. S. Bernstein and S. P. Bhat. Semistability and asymptotic stability of matrix second order systems. *Trans. of ASME. 50th Anniversary Design Issue*, pages 145–153, 1995.

[17] D. S. Bernstein and D. C. Hyland. The optimal projection equations for reduced-order state estimation. *IEEE Transactions on Automatic Control*, AC(30):583–585, 1985.

[18] J. Bierman. *Factorization Methods for Discrete Sequential Estimation.* New York: Academic, 1977.

[19] C. H. Bishop, B. J. Etherton, and S. J. Majumdar. Adaptive sampling with the ensemble transform Kalman filter. part i: Theoretical aspects. *Monthly Weather Review*, 129, 2001.

[20] R. S. Bucy and P. D. Joseph. *Filtering for Stochastic Processes, with Applications to Guidance.* Wiley, 1968.

[21] R. Buizza, M. Miller, and T. N. Palmer. Stochastic representation of model uncertainties in the ECMWF ensemble prediction system. *Quarterly Journal of the Royal Meteorological Society*, 125:2887–2908, 1999.

[22] R. A. Cairns. *Plasma Physics.* Blackie & Son Ltd., 1985.

[23] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods in Fluid Dynamics.* Springer-Verlag, 1988.

[24] J. Chandrasekar, O. Barrero Mendoza, D.S Bernstein, and B. L. R. De Moor. State estimation for linearized MHD flow. In *Proceedings of the 43rd IEEE of the Conference on Decision and Control*, pages 2584–2589, Atlantis, Paradise Island, Bahamas, December 2004.

[25] S. Cohn and R. Todling. Approximate data assimilation schemes for stable and unstable dynamics. *Meteorological Society of Japan*, 74:63–75, 1996.

[26] P. Courtier. Dual formulation of variational assimilation. *Quarterly Journal of the Royal Meteorological Society*, 123:2449–2461, 1997.

[27] C. E. De Souza, M. R. Gevers, and G. C. Goodwin. Riccati equations in optimal filtering of non-stabilizable systems having singular transitions matrices. *IEEE Transactions on Automatic Control*, AC(31):831–838, 1986.

[28] D. P. Dee. On-line estimation of error covariance parameters for atmospheric data assimilation. *Monthly Weather Review*, 123:1128–1145, 1995.

[29] D. P. Dee and R. Todling. Data assimilation in the presence of forecast bias: the GEOS moisture analysis. *Monthly Weather Review*, 128:3268–3282, 2000.

[30] R. Dendy. *Plasma Physics: an Introductory Course.* Cambridge University Press, 1st edition, 1996.

[31] P. Dyer and S. McReynolds. Extension of square-root filtering to include process noise. *Journal of Optimization Theory and Application*, 3:444–458, 1969.

[32] R. E. Evans, M. S. J. Harrison, and R. J. Graham. Joint medium-range ensembles from the meteorological office and ECMWF systems. *Monthly Weather Review*, 128:3104–3127, 2000.

[33] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99(C5):10,143–10,162, 1994.

[34] G. Evensen. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003.

[35] G. Evensen. Sampling strategies and square root analysis schemes for the EnKF. *Internal Report, Hydro Research Center, Bergen, Norway*, October 2004.

[36] B. F. Farrell and P. J. Ioannou. State estimation using a reduced-order Kalman filter. *Journal of Atmospheric Sciences*, 58:3666–3680, 2001.

[37] P. Fieguth, D. Menemenlis, and I. Fukumori. Mapping and pseudo-inverse algorithms for data assimilation. *Proceedings of the International Geoscience Remote Sensing Symposium*, pages 3221–3223, 2002.

[38] G. F. Franklin, J. D. Powell, and A. Emami-Naeni. *Feedback Control of Dynamic Systems.* Prentice Hall, 4th edition, 2002.

[39] K. Gallivan, X. Rao, and P. Van Dooren. Singular Riccati equations stabilizing large-scale systems. *Linear Algebra and its Applications*, 2003.

[40] W. M. Gentleman. Least squares computations by Givens transformations without square roots. *Journal of the Institute for Mathematical Applications*, 12:329–336, 1973.

[41] W. Givens. Computation of plane unitary rotations transforming a general matrix to triangular form. *Journal of the Society for Industrial and Applied Mathematics*, 20:26–50, 1958.

[42] G. Golub and C. Van Loan. *Matrix Computation.* John Hopkins University Press, 3rd edition, 1996.

[43] T. I. Gombosi, K. G. Powell, D. L. De Zeeuw, and et al. Solution adaptive MHD for space plasmas: Sun-to-earth simulations. *Computing in Science and Engineering*, in press, 2004.

[44] M. Goossens. *An introduction to Plasma Astrophysics and Magnetohydrodynamics.* Kluwer Academic Publishers, 2003.

[45] S. M. Grewal and A. P. Andrews. *Kalman Filtering: Theory and Practice Using Matlab.* Wiley Inter-Science, 2nd edition, 2001.

[46] R. Haberman. *Elementary Applied Partial Differential Equations.* Prentice-Hall, 3rd edition, 1998.

[47] W. M. Haddad and D. S. Bernstein. The optimal projection equations for discrete-time reduced-order state estimation for linear systems with multiplicative white noise. *Systems and Control Letters*, 8:381–388, 1987.

[48] W. M. Haddad and D. S. Bernstein. Optimal reduced-order observer-estimators. *AIAA Journal of Guidance, Control, and Dynamics*, 13:1126–1135, 1990.

[49] W. M. Haddad, D. S. Bernstein, H. H. Huang, and Y. Halevi. Fixed-order sampled-data estimation. *International Journal of Control*, 55:129–139, 1992.

[50] T. M. Hamill and C. Snyder. Distance-dependent filtering of background-error covariance estimates in an ensemble Kalman filter. *Monthly Weather Review*, 129:2776–2790, 2001.

[51] J. A. Hansen. Accounting for model error in ensemble-based state estimation and forecasting. *Quarterly Journal of the Royal Meteorological Society*, 130:2373–2391, 2002.

[52] M. S. J. Harrison, T. N. Palmer, D. S. Richardson, and R. Buizza. Analysis and model dependencies in medium-range ensembles: two transplant studies. *Quarterly Journal of the Royal Meteorological Society*, 125:2487–2515, 1999.

[53] A. W. Heemink, M. Verlaan, and A. J. Segers. Variance reduced ensemble Kalman filtering. *Monthly Weather Review*, 129:1718–1728, 2001.

[54] P. Hippe and C. Wurmthaler. Optimal reduced-order estimators in the frequency domain: The discrete-time case. *International Journal of Control*, 52:1051–1064, 1990.

[55] A. S. Householder. Unitary triangularization of a nonsymmetric matrix. *Journal of the Association for Computing Machinery*, 5:339–342, 1958.

[56] P. L. Houtekamer, J. Derome, H. Ritchie, and H. L. Mitchell. A system simulation approach to ensemble prediction. *Monthly Weather Review*, 124:1225–1242, 1996.

[57] P. L. Houtekamer and H. L. Mitchell. Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, 126:796–811, 1998.

[58] P. L. Houtekamer and H. L. Mitchell. A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129:123–137, 2001.

[59] C. S. Hsieh. The unified structure of unbiased minimum-variance reduced-order filters. *Proceedings of the Control Decision Conference, Mahui, HI*, 2003.

[60] A. H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, 1970.

[61] J. M. Jover and T. Kailath. A parallel architecture for Kalman filter measurement update and parameter update. *Automatica*, 22:783–786, 1986.

[62] T. Kailath. Some new algorithms for recursive estimation in constant linear systems. *IEEE Transactions on Information Theory*, IT-19(6):750–760, 1973.

[63] T. Kailath. *State-Space Modelling: Square Root Algorithms.* in Systems and Control Encyclopedia. Pergamon, 1984.

[64] P. G. Kaminski. *Square Root Filtering and Smoothing for Discrete Processes.* Ph.D. Thesis, Stanford University, 1971.

[65] D. I. Lawrie, P. J. Fleming, G. W. Irwin, and S. R. Jones. Kalman filtering: A survey of parallel processing alternatives. *Proceedings of the IFAC Workshop on Algorithms and Architectures for Real-Time Control*, pages 49–56, 1992.

[66] W. B. Manchester, A. J. Gombosi, A. J. Ridley, and et al. Modeling a space weather event from the sun to the earth: CME generation and interplanetary propagation. *Journal of Geophysical Research-Atmospheres*, 109(A2):A02107, 2004.

[67] H. L. Mitchell and P. L. Houtekamer. An adaptive ensemble Kalman filter. *Monthly Weather Review*, 128:416–433, 2000.

[68] H. L. Mitchell, P. L. Houtekamer, and G. Pellerin. Ensemble size, balance, and model-error representation in an ensemble Kalman filter. *Monthly Weather Review*, 130:2791–2808, 2002.

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

[70] D. Parrish and J. Derber. The national meteorological center's spectral statistical-interpolation analysis system. *Monthly Weather Review*, 120:1747–1763, 1992.

[71] C. Penland. A stochastic approach to nonlinear dynamics: a review. *Bulletin of the American Meteorological Society*, 84:921–925, 2003.

[72] J. E. Potter and R. G. Stern. Statistical filtering of space navigation measurements. *Proceedings of AIAA Guidance and Control Conference*, 1963.

[73] D. S. Richardson. Ensembles using multiple models and analyses. *Quarterly Journal of the Royal Meteorological Society*, 127:1847–1864, 2000.

[74] R. W. Schunk and et al. Global assimilation of ionospheric measurements (GAIM). In *Proceedings of the 10th International Ionospheric Effects Symposium*, pages 50–61, Alexandria, VA, 2002.

[75] O. Talagrand and P. Courtier. Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory. *Quarterly Journal of the Royal Meteorological Society*, 113:1311–1328, 1987.

[76] C. L. Thornton. *Triangular Covariance Factorizations for Kalman Filtering.* Ph.D. Thesis, University of California at Los Angeles, School of Engineering, 1976.

[77] C. L. Thornton and G. J. Bierman. A numerical comparison of discrete Kalman filtering algorithms: An orbit determination case study. *JPL Technical Memorandum*, pages 33–771, 1976.

[78] M. K. Tippett, J. L. Anderson, C. H. Bishop, T. M. Hamill, and J. S. Whitaker. Ensemble square-root filters. *Monthly Weather Review*, 131:1485–1490, 2003.

[79] G. Tóth. General code for modeling MHD flows on parallel computers: Versatile advection code. *Astrophysical Letters & Communications*, 34:25, 1996, http://www.phys.uu.nl/~ toth/.

[80] P. Van Dooren. Rational and polynomial matrix factorizations via recursive pole-zero cancellation. *Linear Algebra and its Applications*, 137-138:663–697, 1990.

[81] P. Van Dooren and M. Verhaegen. *Linear Algebra and its Role in Linear Systems Theory.* chapter 47, pp. 447-463, AMS Contemporary Mathematics Series, Provedence, RI, 1985.

[82] P. J. Van Leeuwen. Comment on "Data assimilation using an ensemble Kalman filter technique". *Monthly Weather Review*, 127:1374–1377, 1999.

[83] P. J. Van Leeuwen. A variance-minimizing filter for large-scale applications. *available on ftp://ftp.phys.uu.nl/pub/leeuwen/publ/SIR.ps.*, 2002.

[84] E. Vanmarcke. *Random Fields: Analysis and Synthesis.* The MIT Press, Cambridge, Massachusetts, 1983.

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

[86] M. Verhaegen and V. Verdult. *Lecture Notes on Filtering and System Identification: An Introduction.* TUDelft, 2002.

[87] M. Verlaan and A. W. Heemink. Reduced rank square root filters for large scale data assimilation problems. *2nd International Symposium on Assimilation of Observations in Meteorology and Oceanography, World Meteorological Organization*, pages 247–252, march 1995.

[88] J. S. Whitaker, G. P. Compo, X. Wei, and T. M. Hamill. Reanalysis without radiosondes using ensemble data assimilation. *Monthly Weather Review*, 132:1190–1200, 2003.

[89] J. S. Whitaker and T. M. Hamill. Ensemble data assimilation without perturbed observations. *Monthly Weather Review*, 130:1913–1924, 2002.

[90] A. White and C. Sage. *Optimal Systems Control.* Prentice Hall, Englewood Cliffs, 2nd edition, 1977.

[91] C. Ziehmann. Comparison of a single-model EPS with a multi-model ensemble consisting of a few operational models. *Tellus*, 52(a):280–299, 2000.

Appendix

A List of Abbreviations

AR	–	Autoregressive.
ARE	–	Algebraic Riccati equation.
DARE	–	Discrete algebraic Riccati equation.
DRDE	–	Discrete Riccati difference equation.
CME	–	Corona mass ejection.
EKF	–	Extended Kalman filter.
EnKF	–	Ensemble Kalman filter.
EnSLKF	–	Ensemble spatially localized Kalman filter.
EnSRF	–	Ensemble square root Kalman filter.
ETKF	–	Ensemble transform Kalman filter.
KF	–	Kalman filter.
LTI	–	Linear time invariant.
LTV	–	Linear time variant.
MCMC	–	Monte Carlo Markov chain.
MHD	–	Magnetohydrodynamics system.
PEKF	–	Partial eigen-decomposition Kalman filter.
PDE	–	Partial differential equation.
RMSE	–	Root mean square error.
RRSQRT-KF	–	Reduced rank square root Kalman filter.
SLKF	–	Spatially localized Kalman filter.
SNR	–	Signal to noise ratio.
SQRT-KF	–	Square root Kalman filter.
SQRT-SLKF	–	Square root spatially localized Kalman filter.
SQRT-TKF	–	Square root transform Kalman filter.
SSQRT-KF	–	Singular square root Kalman filter.
TKF	–	Transform Kalman filter.
VAC	–	Versatile Advection Code.
2D-MHD	–	Two dimensional magnetohydrodynamic systems.

B List of Symbols MHD System

- μ_0 - permeability of free space [N A^2]
- η - resistivity of plasma [Ω]

- ρ - average mass density of plasma [kg/m^3]
- p - pressure [kg/ms^2]
- e - energy density [kg/ms^2]
- γ - ratio of specific heats
- v - velocity of the fluid element [m/s]
- m - momentum density [kg/m^2s]
- J - current density [A/m^2]
- E - electric field [volt/m]
- B - magnetic field [Tesla]

C Random Fields

The general representation of an event that is space dependent as a random field [84] is:

$$w(\hat{r}) = \bar{w}(\hat{r}) + f(\hat{r}) \tag{C.1}$$

where,

$w(\hat{r})$ is the realization or sample of the field at $\hat{r}$,

$\hat{r} = (x(t), y(t), z(t))$ is the spatial location, with t the time,

$\bar{w}$ is the trend component of the field (commonly referred to as the mean value) at $\hat{r}$,

$f(\hat{r})$ is zero trend fluctuating component of field at $\hat{r}$.

A Homogeneous Random Field

A random field is said to be homogeneous when the statistical values of the point properties are constant and the statistical values of the cross-point properties depend only on the distance or separation between points. Specifically, the statistical values of a homogeneous field are given by

$$\mathcal{E}[w(\hat{r})] = \mathcal{E}[\bar{w}(\hat{r})] = \mu \tag{C.2}$$

$$\mathcal{E}[w(\hat{r}) - \bar{w}(\hat{r})][w(\hat{r}) - \bar{w}(\hat{r})]^{\mathrm{T}} = \mathcal{E}[f(\hat{r})]^2 = \sigma^2 \tag{C.3}$$

$$\mathcal{E}[w(\hat{r}_0) - \bar{w}(\hat{r}_0)][w(\hat{r}_1) - \bar{w}(\hat{r}_1)]^{\mathrm{T}} = \mathcal{E}[f(\hat{r}_0)f(\hat{r}_1)] = \sigma^2\rho(\hat{r}_0 - \hat{r}_1) \tag{C.4}$$

where μ is the mean value, σ^2 is the variance, $\rho(\cdot)$ is the correlation function. When the cross point properties are function of absolute distances between points, the field is said to be isotropic. That is

$$\mathcal{E}[w(\hat{r}_0) - \bar{w}(\hat{r}_0)][w(\hat{r}_1) - \bar{w}(\hat{r}_1)]^{\mathrm{T}} = \mathcal{E}[f(\hat{r}_0)f(\hat{r}_1)] = \sigma^2\rho(|\hat{r}_0 - \hat{r}_1|). \tag{C.5}$$

A Non-Homogeneous Random Field

A random field is said to be non-homogeneous when any of the statistical values of the point or cross point properties is dependent on the location. Specifically, the statistical values of point and cross point properties are given by

$$\mathcal{E}[w(\hat{r})] = \mathcal{E}[\bar{w}(\hat{r})] = \mu(\hat{r}), \tag{C.6}$$

$$\mathcal{E}[w(\hat{r}) - \bar{w}(\hat{r})][w(\hat{r}) - \bar{w}(\hat{r})]^{\mathrm{T}} = \mathcal{E}[f(\hat{r})]^2 = \sigma^2(\hat{r}), \tag{C.7}$$

$$\mathcal{E}[w(\hat{r}_0) - \bar{w}(\hat{r}_0)][w(\hat{r}_1) - \bar{w}(\hat{r}_1)]^{\mathrm{T}} = \mathcal{E}[f(\hat{r}_0)f(\hat{r}_1)] = \sigma^2\rho(\hat{r}_0, \hat{r}_1). \tag{C.8}$$

D Generation of Random Fields for Ensemble-Based Filters

In this section we describe a simple way to generate homogeneous random fields in cases where an eigenvalue decomposition of the initial error covariance matrix P can be computed. First, generate a desired covariance matrix $P \in \mathbb{R}^{n\times n}$, then compute its eigenvalue decomposition, hence

$$P = JDJ^{\mathrm{T}}. \tag{D.1}$$

Now, define a matrix $T \in \mathbb{R}^{n\times n}$ so that

$$T = JD^{1/2} \tag{D.2}$$

Afterwards, we compute a vector m_k using a Gaussian random generator with zero mean, and variance one. Then, the random field perturbations $f(\hat{r})_k$ can be computed as

$$f(\hat{r})_k = Tm_k. \tag{D.3}$$

Finally, the random field perturbations are added to the random field mean (initial state space guess plus white noise ϵ_k) to obtain the random fields. As a result

$$w(\hat{r})_k = \bar{w}(\hat{r}) + Tm_k + \epsilon_k \tag{D.4}$$

E MHD Data Assimilation Figures

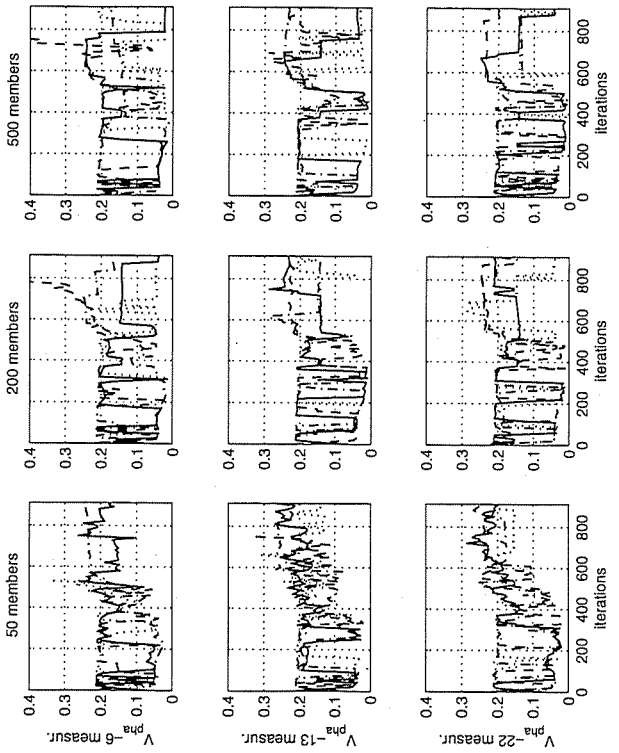


Figure E.1: RMSE comparison of the velocity phase. Solid line the EnKF, dashed line the EnSRF, dash-dotted line the EnSLKF, and dotted line ETKF. 1 iteration = 1 Ts.

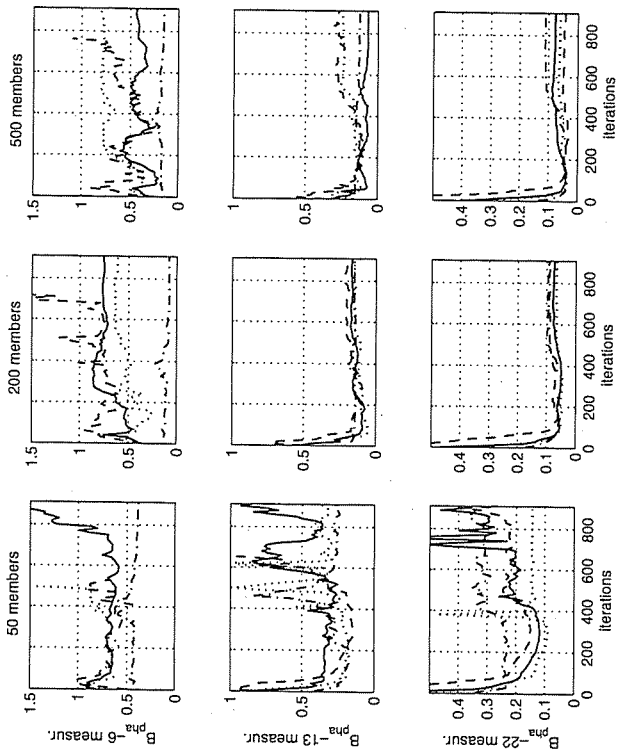


Figure E.2: RMSE comparison of the velocity phase. Solid line the EnKF, dashed line the EnSRF, dash-dotted line the EnSLKF, and dotted line ETKF. 1 iteration = 1 Ts.

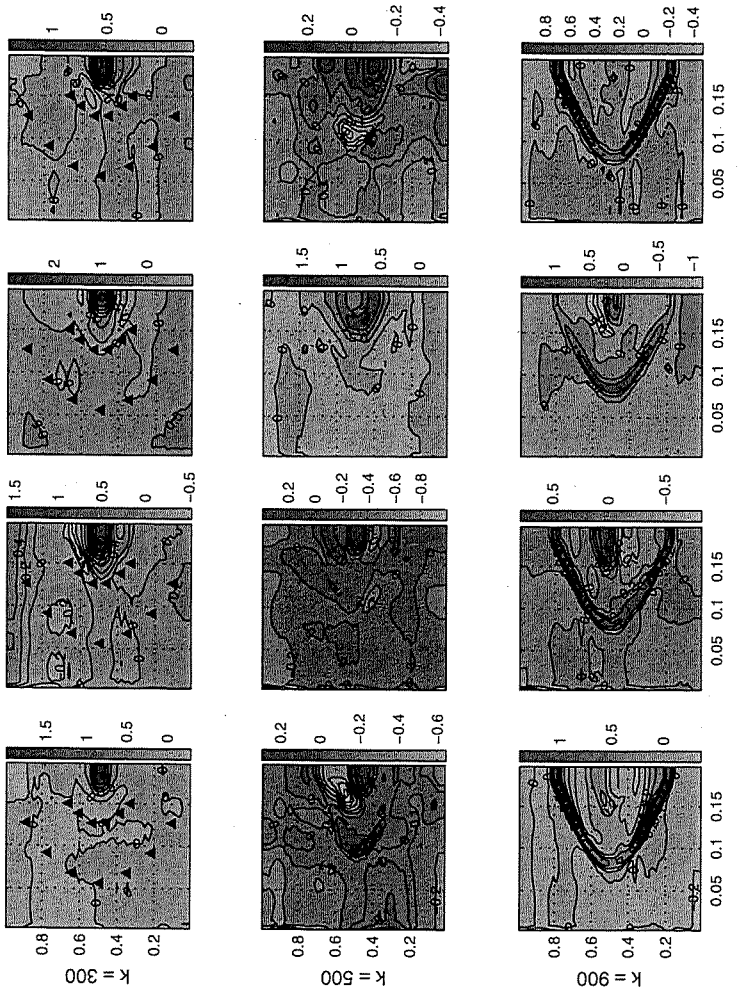


Figure E.3: Magnetic field magnitude residuals [mT] with 13 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points.

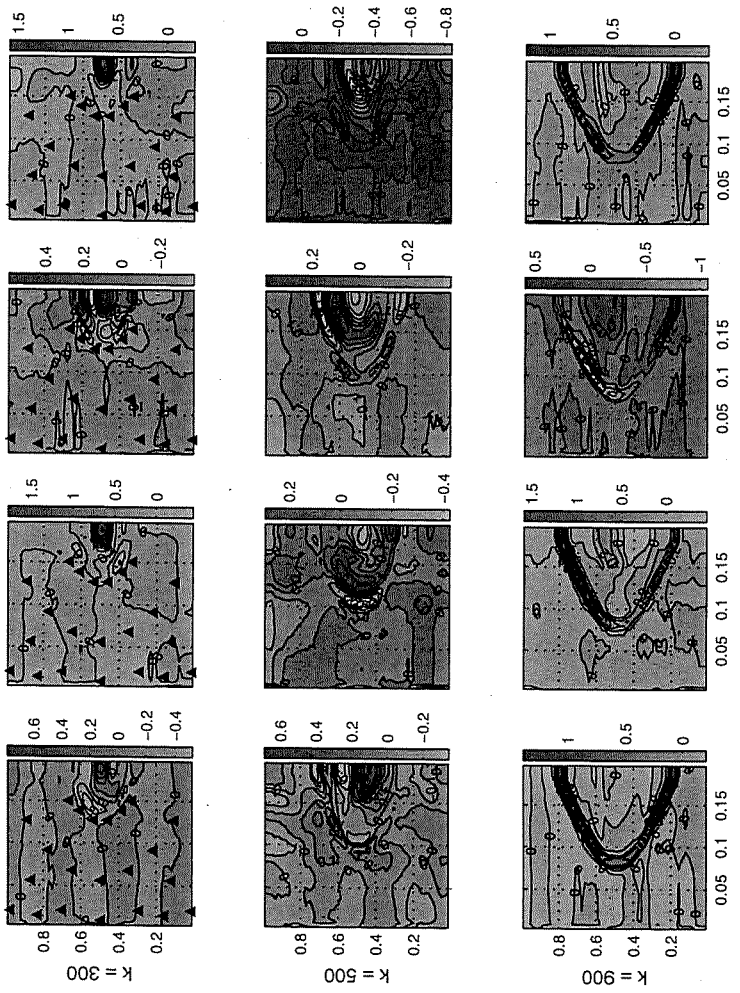


Figure E.4: Magnetic field magnitude residuals [mT] with 22 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points.

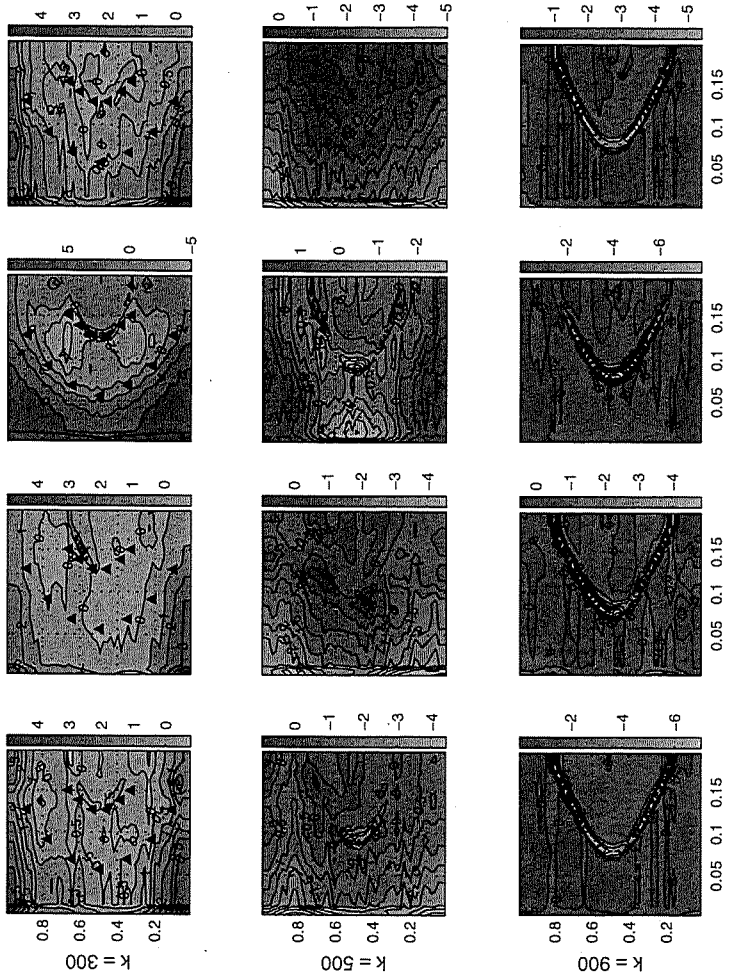


Figure E.5: Velocity magnitude residuals [m/s] with 13 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points.

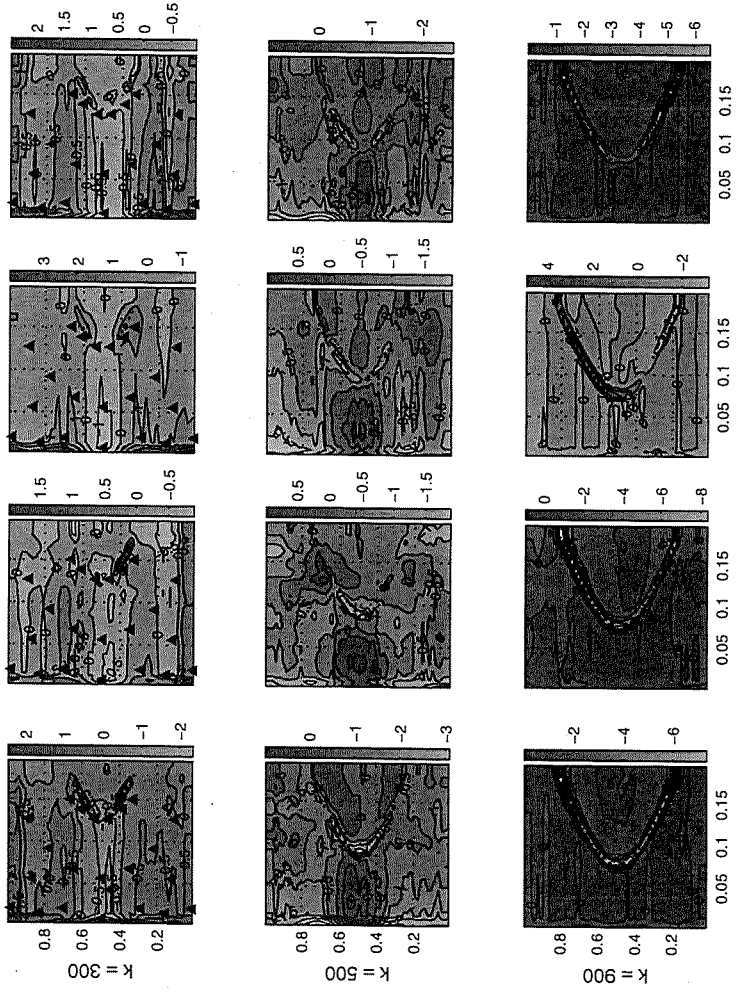


Figure E.6: Velocity magnitude residuals [m/s] with 22 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points.

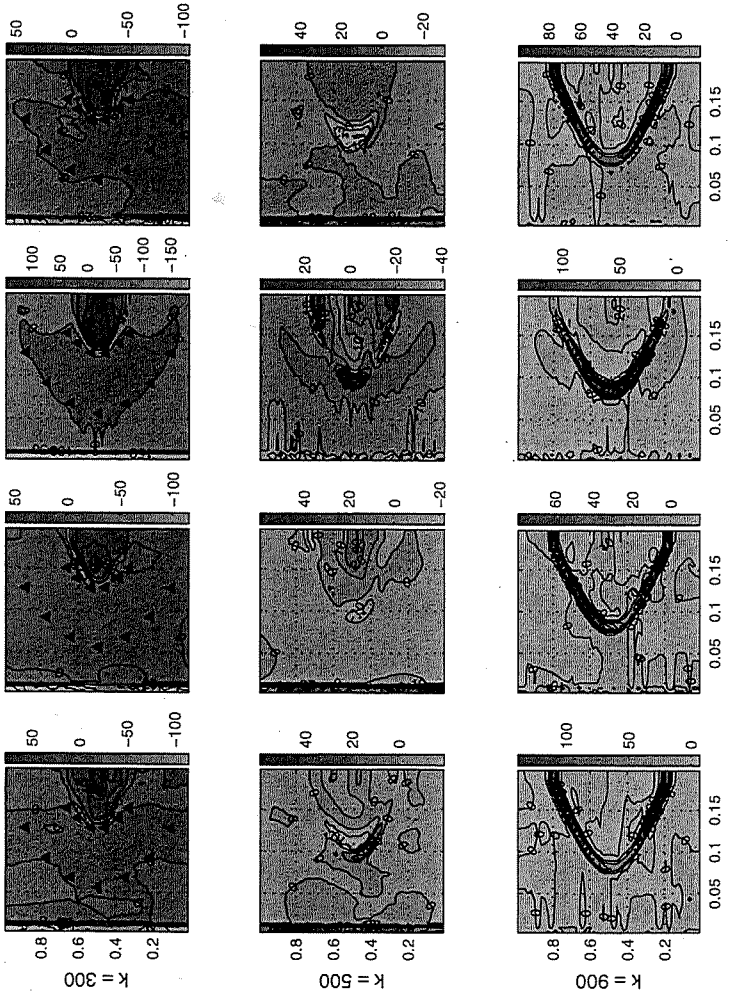


Figure E.7: Velocity magnitude residuals [kg/ms^2] with 13 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points.

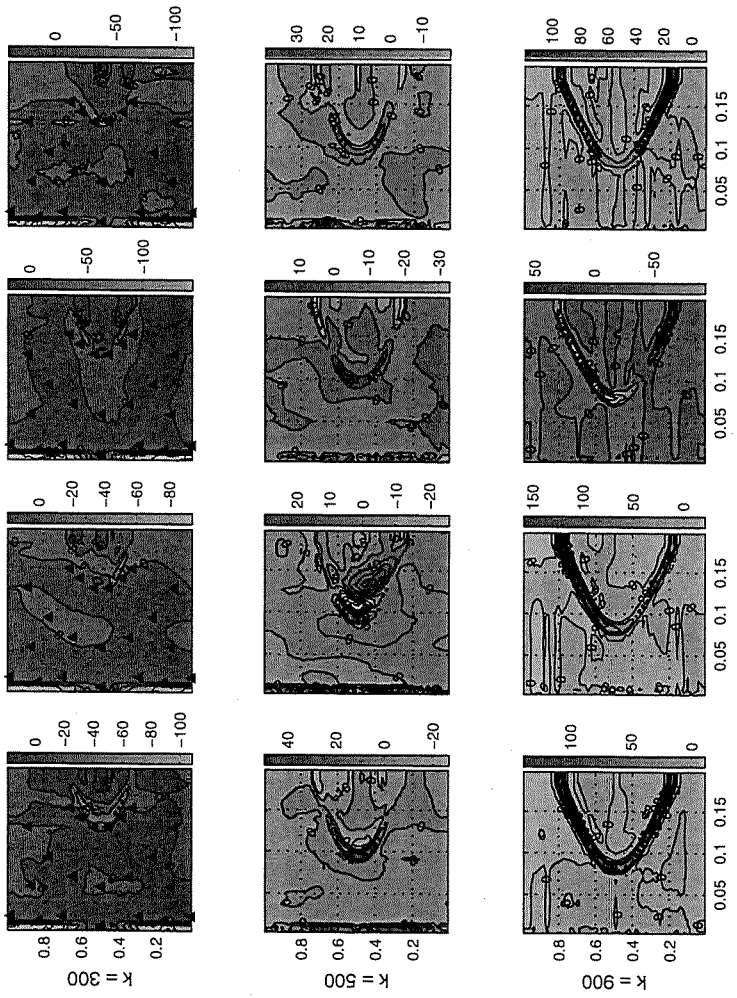


Figure E.8: Velocity magnitude residuals [kg/ms^2] with 22 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points.

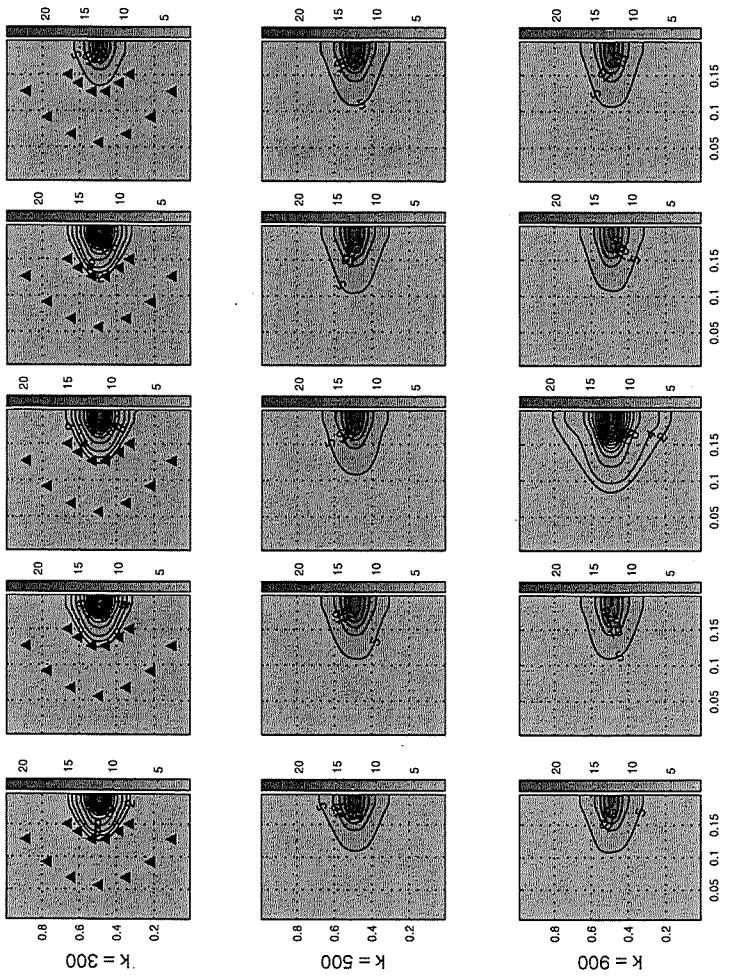


Figure E.9: Magnetic field magnitude [mT] with 13 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points.

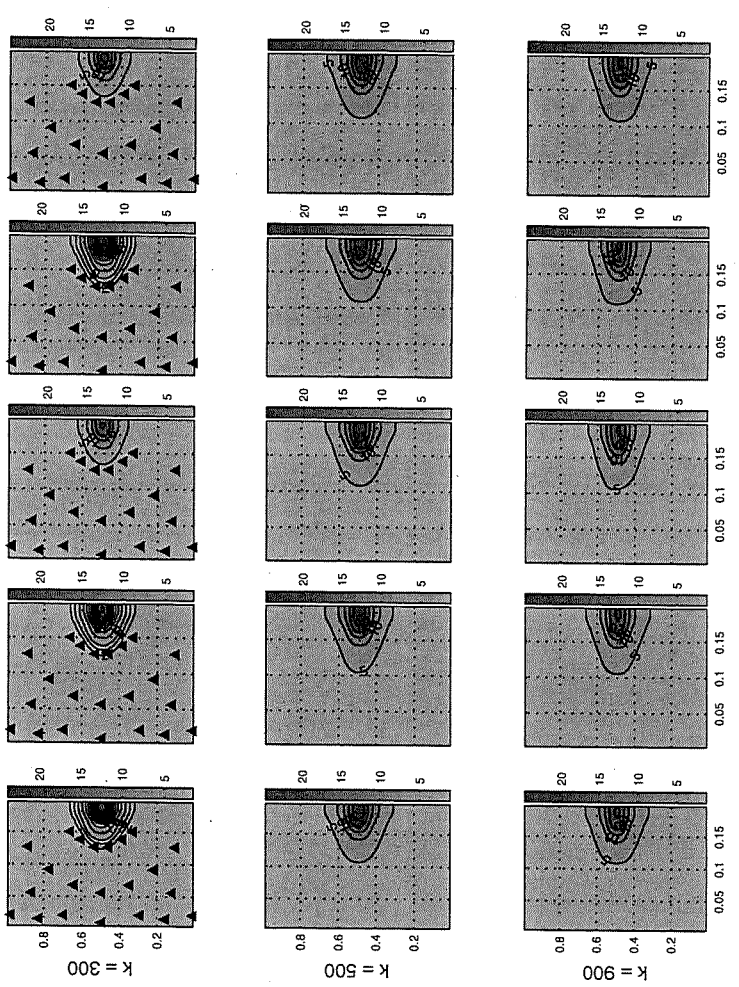


Figure E.10: Magnetic field magnitude [mT] with 22 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points.

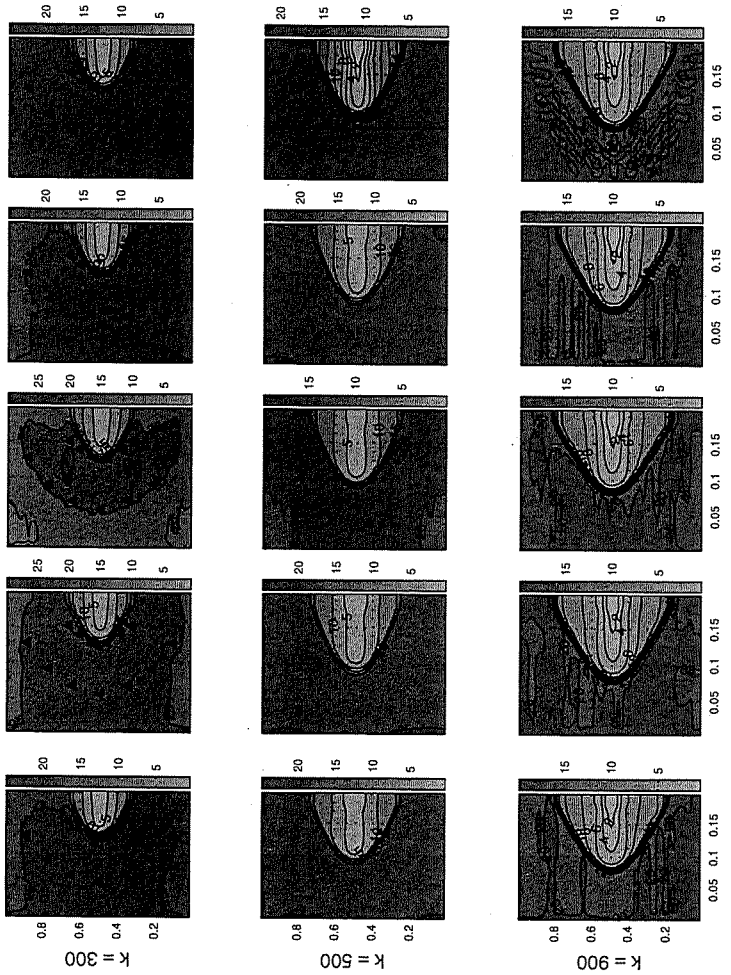


Figure E.11: Velocity magnitude [m/s] with 13 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points. The empty figure space means that the algorithm has crashed, something that is no desirable in a data assimilation system, as explained in the analysis of the results.

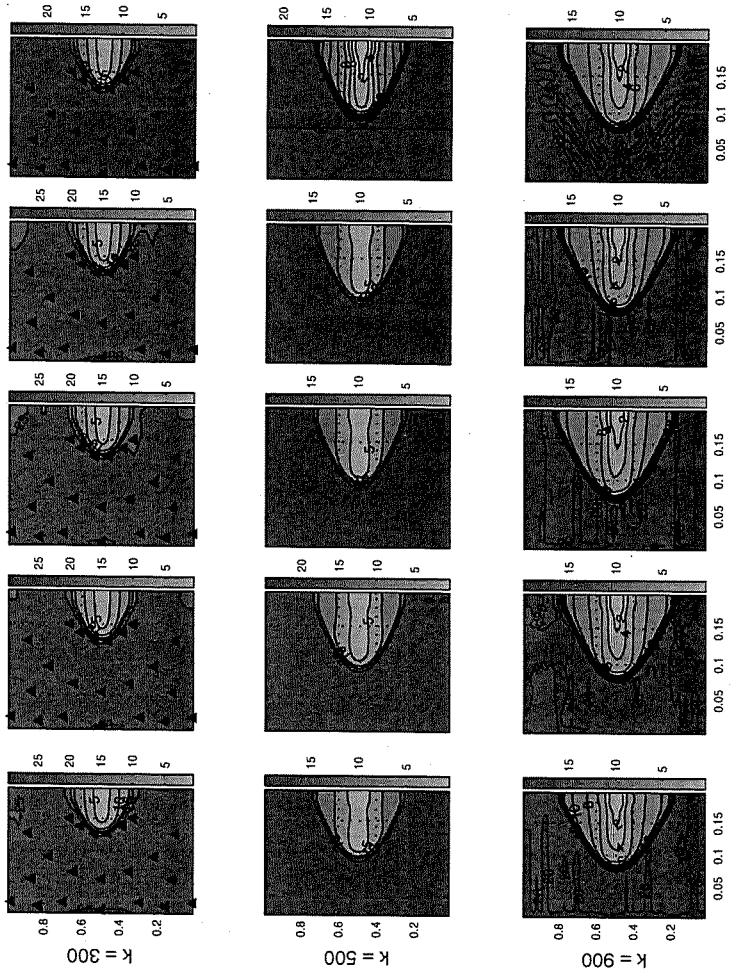


Figure E.12: Velocity magnitude residuals [m/s] with 22 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points.

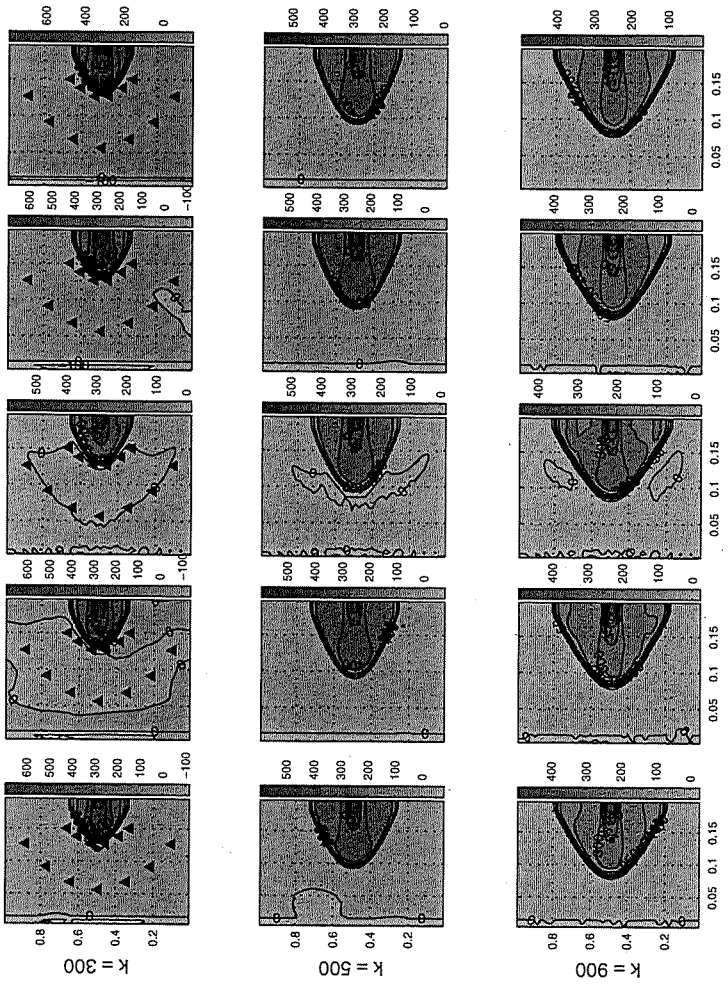


Figure E.13: Velocity magnitude residuals [kg/ms^2] with 13 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points.

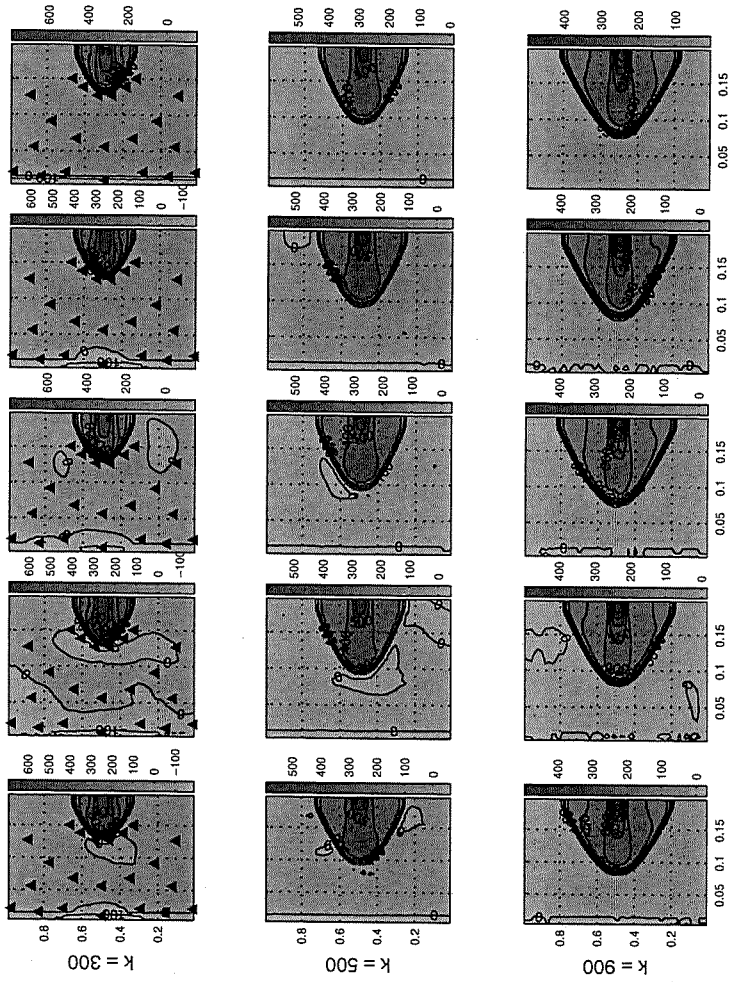


Figure E.14: Velocity magnitude residuals [kg/ms^2] with 22 measurements points and 200 ensemble members. From top to bottom, the ETKF, the EnSLKF, the EnSRF, and the EnKF. Simulation sampling time $T_s = 1 \times 10^{-4}s$. Data assimilation sampling time $k = 4T_s$. Black triangles indicate the measurement points.