Large Scale Data Assimilation with Application to the Ionosphere-Thermosphere

by

In Sung Kim

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in The University of Michigan
2008

Doctoral Committee:

Professor Dennis S. Bernstein, Co-chair
Associate Professor Aaron J. Ridley, Co-chair
Professor Pierre T. Kabamba
Professor N. Harris McClamroch

ACKNOWLEDGMENTS

I would like to thank my advisor Dennis S. Bernstein and co-advisor Aaron J. Ridley for their valuable advice and guidance in shaping and completing this dissertation. Next, I would like to thank Professor N. Harris McClamroch and Professor Pierre T. Kabamba for having been my committee members and providing helpful suggestions.

I also would like to thank the staffs at the Center for Advanced Computing in the University of Michigan for their kind helps to perform vast amount of parallel computations for this dissertation.

All the staffs at the department of Aerospace Engineering provided good support for me from the beginning to the end of my stay at the department, to which I'm grateful.

I am also grateful to the National Science Foundation for supporting my research under grant CNS-0539053.

Finally, I would like to thank all of my family members, in particular, my wife Heejin and my daughter Yeojoo who have been loving me, supporting me, and giving faith to me, which I will never forget.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Data assimilation is the process of merging measurement data with a model to estimate the states of a system that are not directly measured. By means of data assimilation, we can expand the effectiveness of limited measurements by using the model and, at the same time, increase the accuracy of model estimates using the measurements.

In this dissertation, we survey and develop data assimilation algorithms that are applicable to large-scale nonlinear systems. Very high order dynamics, nonlinearity, and input uncertainties are addressed since they characterize the problems associated with large-scale data assimilation. Specifically, we focus on developing the data assimilation algorithms for the ionosphere-thermosphere using the Global Ionosphere-Thermosphere Model (GITM).

For developing computationally tractable algorithms, we obtain finite-horizon optimal reduced-order estimators for time-varying linear systems, and, subsequently, develop linear suboptimal reduced-complexity estimators. The suboptimal estimators are based on localization and the reduced-rank square root of the error covariance.

To deal with nonlinearity, we use the unscented Kalman filter and ensemble Kalman filter. We apply suboptimal reduced-complexity algorithms developed for linear systems based on the unscented Kalman filter. Also, we develop the ensemble-on-demand Kalman filter, which can be used for the special case of a single global

disturbance, and which avoids propagating the ensemble members for all of the time steps. Furthermore, we show that the ensemble size of the ensemble Kalman filter does not have to be unnecessarily large if the statistics of the disturbance sources are identified.

Finally, we apply the ensemble-on-demand Kalman filter and ensemble Kalman filter to data assimilation based on GITM for uncertain solar EUV flux and geomagnetic storm conditions, respectively. We present data assimilation results, through extensive numerical investigations using simulated measurements. While performing simulations, we observe that poor correlations between states should be set to zero to avoid filter instability. In addition, ionosphere and thermosphere measurements can be used together with an appropriate region of data injection to guarantee overall good estimation performance. With those constraints, we show that good estimation results can be obtained using a small ensemble size for each ensemble filter.

CHAPTER I

Introduction

There is always a need to know the states of a system of interest to make an important decision, control the system, or predict reliably the states of the system in the future. A simple but sure method for knowing the states of a system is to measure them directly. However, in a general complex system, it is not feasible or even possible to directly measure all of the states of the system. For example, imagine that we need to know the air temperature of the Earth's atmosphere all over the world with 1 meter by 1 meter by 1 meter resolution and suppose that only available measuring device is a thermometer. Then, we need 5×10^{14} thermometers even for one fixed height level. Moreover, we need to install thermometers above all oceans in the world.

An alternative method for knowing the states of the system instead of measuring them all is to create and use a model for the system. If the model is perfect, we can accurately estimate the states of the system when the drivers (external disturbances) and the initial condition of the system are known exactly. However, the initial conditions are never known and there are always uncertainties in the drivers. Hence, the estimation by a model is effective only when the model error is not large, the effects of initial conditions are not significant, and the input uncertainties are small.

Data assimilation is a process of merging available measurement data with a model to estimate the states of a system other than the measurements. By means of data assimilation, we can expand the effectiveness of the limited measurements using the model and, at the same time, reduce inaccuracies of the model estimates using the measurements. Amongst various estimators for data assimilation, we consider the estimators based on the Kalman filter which is optimal in the presence of process and measurement noises [1].

In this dissertation, the system of interest is the ionosphere-thermosphere. The ionosphere is in the altitude range from 100 km to 1000 km where ionized particles are rich. The ionosphere is coupled with the thermosphere, which is in the same range of altitudes. Perturbations to the states of the ionosphere-thermosphere, or space weather, can have dramatic effects on both space- and ground-based technology. For instance, satellites in low-Earth orbit can experience significant changes in velocity when the thermospheric drag on the spacecraft changes due to expansion of the thermosphere. Furthermore, abrupt changes in the total electron content (TEC) of the ionosphere can lead to substantial errors in ground-based Global Positioning System (GPS) measurements. Most dramatically, on March 13, 1989, the blackout of the entire Quebec power grid was caused by huge ionospheric current fluctuations which induced abnormal currents in power lines, thus destroyed transformers and electrical networks [2]. The practical importance of the ionosphere-thermosphere is the fundamental motivation of this research.

The available ionosphere-thermosphere measurement data are obtained mainly from three sources: (1) ion measurements by incoherent scatter radars (ISRs), (2) air density measurements by satellites, and (3) total electron content measurements from GPS stations. ISRs typically measure electron number density N_{e}, electron

temperature T_e, ion velocity V_{ion} and ion temperature T_{ion}, which are all ion properties on the line of sight of the radar. Therefore, ISR provides ion properties at a series of points along the line of sight. In addition, available data are air density measurements along the flight path of geo-scientific satellites that operate at altitudes of the ionosphere-thermosphere. Finally, the total electron content (TEC) measurements can be obtained from GPS stations that are globally distributed. TEC is the integral of electron number density along the line connecting the GPS satellites and the ground GPS receivers.

Various studies have performed ionospheric data assimilation based on measurements of TEC [3–5] and bottom side electron density profiles [6]. More recently, with the availability of precise thermospheric density data from the Champ [7] and Grace [8] experiments, interest has arisen in assimilating thermospheric variables. Thermospheric data assimilation using simulated measurements of thermospheric composition is investigated in [9]. In all previous works, either the ionosphere or the thermosphere data assimilations have been studied separately. For example, to estimate electron density distribution, [4] uses the simple Gauss-Markov model based on the physics-based ionosphere model that uses the neutral densities and winds as some of drivers. To assimilate neutral composition data of the thermosphere, [9] uses a simple Gauss-Markov model whose states are two-dimensional thermospheric neutral compositions, specifically, the ratio of the height integrated atomic oxygen and molecular nitrogen concentrations.

In this dissertation, we incorporate the Global Ionosphere-Thermosphere Model (GITM) [10] into the data assimilation for the ionosphere and the thermosphere together, using the available ionosphere and thermosphere measurements. GITM is a physics-based fully parallel three-dimensional (3D) Computational Fluid Dy-

namics (CFD) model of the coupled ionosphere-thermosphere system in spherical coordinates. The model solves the conservation equations in altitude coordinates as opposed to pressure coordinates and does not assume hydrostatic equilibrium. Therefore, the vertical momentum equation can be solved self-consistently, and substantial vertical flows can develop [11]. GITM uses a block-based uniform domain decomposition in the horizontal direction, along with a non-uniform altitude grid, with a resolution of 1/3 scale height. The grid is entirely flexible, so the user is free to change the horizontal resolution from run to run by specifying the number of blocks to use in a given simulation. A consequence of this flexibility is that GITM can be run in 1-dimension (1D) where only a single latitude and longitude are simulated, and horizontal transport and gradients are ignored. Since 1D GITM can be run quickly on a personal computer, this feature renders long-term studies of the upper atmosphere feasible, while facilitating quick debugging of the code.

There are 14 states per each cell of GITM. The number of states in vertical one-dimensional GITM with 50 cells is as large as 700. For 3 dimensional global GITM when typical longitude-latitude resolution is 5 degree by 5 degree, the total number of states is around 2 million. That is, GITM has extremely high order dynamics. Moreover, GITM solves hydrodynamic equations with more than 6 species coupled with chemistry equations, which means that GITM is highly nonlinear. Finally, exact drivers for the ionosphere-thermosphere are not known, and thus the inputs are uncertain.

Data assimilation using a high order dynamic system model such as GITM is commonly called large scale data assimilation, which often involves high nonlinearity of the model as in the case of GITM.

As a method for data assimilation, the Kalman filter is considered to be a powerful

tool since it provides optimal estimates with error covariance information. However, the Kalman filter is not computationally tractable when the system has high order dynamics. Moreover, the Kalman filter is an optimal estimator for linear systems with linear measurement mapping. Hence, the Kalman filter cannot be directly used for nonlinear system estimation.

To overcome high order dynamics problem, the reduced-order state estimators has been developed for several decades; representative work includes [12–26]. Most of these techniques involve data injection with an estimator whose order is less than the order of the plant. The estimator dynamics are typically obtained from the full-order dynamics by a truncation or projection process, while the estimator gain is obtained from a steady-state or updated error-covariance matrix based on the full-order dynamics.

For large-scale systems, however, reduced-order filters based on a full-order error covariance may not be feasible. In particular, the effort needed to compute the steady-state error covariance or to update the time-dependent error covariance is significant, namely, $\mathrm{O}(n^3)$ for a system of order n. To relieve the $\mathrm{O}(n^3)$-computational burden of full-order-error-covariance-based estimation, a reduced-order filters based on a reduced-order error covariance is developed in [27], where balancing is used to obtain a reduced-order model that provides the basis for the error-covariance update. Although the estimator with reduced-order error covariance is suboptimal, the benefit from the reduction is greater than the loss of accuracy.

To deal with nonlinearity, extended Kalman filter [28] and SDRE filter [29–31] are most popular and used in many areas. These filters require linearization and factorization, respectively, to formulated a linear system matrix to apply the Kalman filter. However, it is not feasible to extract a linear system matrix from a nonlinear

large scale system. In contrast to extended Kalman filter and SDRE filter, ensemble Kalman filter (EnKF) [32] and unscented Kalman filter (UKF) [33] propagates ensemble of models, which eliminate the need to extract a linear system matrix. Even though EnKF and UKF can be applied to general nonlinear systems, both of them propagate an ensemble of members for every step of simulation. In particular, UKF in its original formulation, requires an ensemble of $2n + 1$ members, where n is the number of states, which is prohibitively large for a large-scale system. Therefore, EnKF and UKF are not feasible methods for large data assimilation without appropriate reductions.

Now the goal of this research is to develop and apply Kalman filter-based suitable large scale data assimilation algorithms to GITM with available ionosphere-thermosphere measurements. The algorithms must yield physically meaningful estimates, and at the same time, must be able to run at a feasible speed, eventually in real time, using available computing resources such as general public grid-computers like NYX at the University of Michigan.

This dissertation addresses the problems of developing large scale data assimilation algorithms that are directly applicable to the ionosphere-thermosphere and shows results of the data assimilation based on GITM. That is, we put high priority on the practical algorithms so that we can obtain useful results by applying developed algorithms to the ionosphere-thermosphere. However, since GITM-based data assimilation has characteristics that are common to general large scale data assimilation, the results of this dissertation provide useful contributions to the general large scale data assimilation community.

The contents of each chapter are as follows. In Chapter II, we consider optimal reduced-order estimators for time-varying linear discrete-time systems. The reduced-

order estimators are obtained using a finite-horizon minimization approach and thus do not require the solution of algebraic Lyapunov or Riccati equations. Even though the optimal-reduced order estimators reduce computational complexity in data injection, they still require full-order error-covariance propagation in order to be optimal.

Next, in Chapter III, we compare several suboptimal reduced-order Kalman filters for discrete-time LTI systems based on reduced-order error-covariance propagation. These filters use combinations of balanced model truncation and complementary steady-state covariance compensation. After describing each method, we compare their performance through numerical studies using a compartmental model example. These methods are aimed at large-scale data-assimilation problems where reducing computational complexity is critical.

In Chapter IV, we consider the unscented Kalman filter (UKF) as a large scale data assimilation method because it has deterministic UKF ensemble size and it does not require a Jacobian that is almost impossible to obtain from nonlinear large-scale system. However, UKF propagates $2n+1$ ensemble members, where n is the number of states, which is prohibitively large for a large-scale system. In this chapter, we discuss an extension of the UKF that propagates a surrogate reduced-order covariance and also uses a complementary static estimator gain based on the steady-state correlation between the error in the estimates of the state and measurements to obtain estimates of the entire state, which are introduced in Chapter III using linear system.

In Chapter V, we consider a reduced-rank square-root Kalman filter based on the Cholesky decomposition of the state-error covariance, where the reduced-rank square-root of the error covariance corresponds to the ensemble size reduction of UKF. We compare the performance of this filter with the reduced-rank square-root

filter based on the singular value decomposition. The Cholesky-based square root Kalman filter is computationally more efficient, and, in many cases, more accurate than the SVD-based.

In Chapter VI, we apply the unscented Kalman filter (UKF) to data assimilation based on vertical one-dimensional GITM with approximate disturbance covariance. To reduce the computational complexity of UKF, we introduce a localized, sampled-data update scheme with frozen-intersample error covariance, and examine its performance through numerical simulation.

Next, when the localized UKF with approximate disturbance covariance is applied to 3-dimensional GITM, we are not able to obtain effective data assimilation results because the UKF turns out to be sensitive to the disturbance covariance. Hence, in Chapter VII, we examine the ensemble Kalman filter and its variants, which do not use explicit disturbance covariance. The ensemble Kalman filter for data assimilation involves the propagation of a collection of ensemble members. Under the assumption of time-sparse measurements, we avoid propagating the ensemble members for all of the time steps by creating an ensemble of models only when a new measurement is made available. We call this algorithm the ensemble-on-demand Kalman filter (EnODKF). We use guidelines for ensemble size within the context of EnODKF, and demonstrate the performance of EnODKF for representative examples, specifically, a lumped vibration problem and a heat flow problem.

In Chapter VIII, we apply EnODKF and EnKF to 3-dimensional GITM-based data assimilation. Specifically, EnODKF is used for the case of solar EUV flux disturbance, with a single global disturbance, whereas EnKF is used for geomagnetic storm conditions where the effects of geomagnetic storm drivers are propagated from near the Earth's poles to the equator. We inject data into a local region using

combinations of simulated ISR measurements of electron number density and ion temperature, satellite measurements of air density, and TEC measurements. We numerically investigate the effects of data assimilation for various regions of data injection and combinations of measurements. The results are promising since the algorithms achieve good estimation accuracy with feasible running speed.

CHAPTER II

Reduced-Order Kalman Filtering for Time-Varying Systems

2.1 Introduction

Because the classical Kalman filter provides optimal least-squares estimates of all of the states of a linear time-varying system, there is longstanding interest in obtaining simpler filters that estimate only a subset of states. This objective is of particular interest when the system order is extremely large, which occurs for systems arising from discretized partial differential equations [34].

One approach to this problem is to consider reduced-order Kalman filters, which provide state estimates that are suboptimal [35–37]. Variants of the classical Kalman filter have been developed for computationally demanding applications such as weather forecasting [38–40], where the filter gain is modified so as to reduce the computational requirements. A comparison of various techniques is given in [41]. An alternative approach to reducing complexity is to restrict the data-injection subspace to obtain a spatially localized Kalman filter [42].

In the present paper we revisit the approach of [35, 43], which considers the problem of fixed-order steady-state reduced-order estimation. For a linear time-invariant system, the optimal steady-state fixed-order filter is characterized in [35, 43] by coupled Riccati and Lyapunov equations, whose solution requires iterative techniques.

This paper extends the results of [35, 43] by adopting the finite-horizon optimization technique used in [42] to obtain reduced-order filters that are applicable to time-varying systems. This technique also avoids the periodicity constraint associated with the multirate filter derived in [44]. Related techniques are used in [45].

In addition to the reduced-order filter considered in [35, 43], we also consider a fixed-structure subspace observer constrained to estimate a specified collection of states. This problem is considered in [37, 46]. The difference between the reduced-order filter and subspace observer is apparent in the the distinct oblique projectors τ and μ, which characterize the filter and observer gains, respectively.

2.2 Optimal Finite-Horizon Reduced-Order Estimator

Consider the system

$$x_{k+1} = A_k x_k + D_{1,k} w_k, \tag{2.1}$$

$$y_k = C_k x_k + D_{2,k} w_k, \tag{2.2}$$

where $x_k \in \mathbb{R}^{n_k}$, $y_k \in \mathbb{R}^{p_k}$, and $w_k \in \mathbb{R}^{d_k}$ is a white noise process with zero mean and unit covariance. Note that $A_k \in \mathbb{R}^{n_k \times n_{k+1}}$ need not be square and may have time-varying dimension.

We consider a reduced-order estimator with dynamics

$$x_{\mathrm{e},k+1} = A_{\mathrm{e},k} x_{\mathrm{e},k} + B_{\mathrm{e},k} y_k, \tag{2.3}$$

where $x_{\mathrm{e},k} \in \mathbb{R}^{n_{\mathrm{e},k}}$. Define the augmented state covariance $\tilde{Q}_k$ by

$$\tilde{Q}_k \triangleq \mathcal{E}[\tilde{x}_k \tilde{x}_k^{\mathrm{T}}], \tag{2.4}$$

where $\tilde{x}_k \in \mathbb{R}^{\tilde{n}_k}$, $\tilde{n}_k \triangleq n_k + n_{\mathrm{e},k}$, is defined by

$$\tilde{x}_k \triangleq \left[\begin{array}{cc} x_k^{\mathrm{T}} & x_{\mathrm{e},k}^{\mathrm{T}} \end{array} \right]^{\mathrm{T}}. \tag{2.5}$$

Consider the cost function

$$J_k(A_{\mathrm{e},k}, B_{\mathrm{e},k}) \triangleq \mathcal{E}\Big[\big(L_k x_{k+1} - x_{\mathrm{e},k+1}\big)^{\mathrm{T}}\big(L_k x_{k+1} - x_{\mathrm{e},k+1}\big)\Big], \tag{2.6}$$

where $L_k \in \mathbb{R}^{n_{\mathrm{e},k} \times n_k}$ determines the subspace of the state x that is weighted. It follows from (2.4) and (2.5) that J_k is given by

$$J_k(A_{\mathrm{e},k}, B_{\mathrm{e},k}) = \mathrm{tr}\left(\tilde{Q}_{k+1}\tilde{R}_k\right), \tag{2.7}$$

where $\tilde{R}_k \in \mathbb{R}^{n+n_{\mathrm{e}}}$ is defined by

$$\tilde{R}_k \triangleq \begin{bmatrix} L_k^{\mathrm{T}} L_k & -L_k^{\mathrm{T}} \\ -L_k & I \end{bmatrix}. \tag{2.8}$$

Note that (2.1) and (2.3) imply that

$$\tilde{x}_{k+1} = \tilde{A}_k \tilde{x}_k + \tilde{D}_{1,k} w_k, \tag{2.9}$$

where

$$\tilde{A}_k \triangleq \begin{bmatrix} A_k & 0 \\ B_{\mathrm{e},k} C_k & A_{\mathrm{e},k} \end{bmatrix}, \quad \tilde{D}_{1,k} \triangleq \begin{bmatrix} D_{1,k} \\ B_{\mathrm{e},k} D_{2,k} \end{bmatrix}. \tag{2.10}$$

Therefore,

$$\tilde{Q}_{k+1} = \tilde{A}_k \tilde{Q}_k \tilde{A}_k^{\mathrm{T}} + \tilde{V}_{1,k}, \tag{2.11}$$

where

$$\tilde{V}_{1,k} \triangleq \begin{bmatrix} V_{1,k} & V_{12,k} B_{\mathrm{e}}^{\mathrm{T}} \\ B_{\mathrm{e}} V_{12,k}^{\mathrm{T}} & B_{\mathrm{e},k} V_{2,k} B_{\mathrm{e},k}^{\mathrm{T}} \end{bmatrix} \tag{2.12}$$

and

$$V_{1,k} \triangleq D_{1,k} D_{1,k}^{\mathrm{T}}, \quad V_{12,k} \triangleq D_{1,k} D_{2,k}^{\mathrm{T}}, \quad V_{2,k} \triangleq D_{2,k} D_{2,k}^{\mathrm{T}}. \tag{2.13}$$

Partitioning $\tilde{Q}_k$ as

$$\tilde{Q}_k = \begin{bmatrix} \tilde{Q}_{1,k} & \tilde{Q}_{12,k} \\ \tilde{Q}_{12,k}^{\mathrm{T}} & \tilde{Q}_{2,k} \end{bmatrix}, \tag{2.14}$$

it follows from (2.11) that

$$\tilde{Q}_{1,k+1} = A_k \tilde{Q}_{1,k} A_k^{\mathrm{T}} + V_{1,k}, \tag{2.15}$$

$$\tilde{Q}_{12,k+1} = A_k \tilde{Q}_{1,k} C_k^{\mathrm{T}} B_{\mathrm{e},k}^{\mathrm{T}} + A_k \tilde{Q}_{12,k} A_{\mathrm{e},k}^{\mathrm{T}} + V_{12,k} B_{\mathrm{e}}^{\mathrm{T}}, \tag{2.16}$$

$$\begin{aligned} \tilde{Q}_{2,k+1} = {} & B_{\mathrm{e},k} \left(C_k \tilde{Q}_{1,k} C_k^{\mathrm{T}} + V_{2,k} \right) B_{\mathrm{e},k}^{\mathrm{T}} \\ & + A_{\mathrm{e},k} \tilde{Q}_{12,k}^{\mathrm{T}} C_k^{\mathrm{T}} B_{\mathrm{e},k}^{\mathrm{T}} + B_{\mathrm{e},k} C_k \tilde{Q}_{12,k} A_{\mathrm{e},k}^{\mathrm{T}} + A_{\mathrm{e},k} \tilde{Q}_{2,k} A_{\mathrm{e},k}. \end{aligned} \tag{2.17}$$

Therefore, (2.7) and (2.8) imply that J_k can be expressed as

$$\begin{aligned} J_k \left(A_{\mathrm{e},k}, B_{\mathrm{e},k} \right) = {} & \mathrm{tr} \left[L_k \left(A_k \tilde{Q}_{1,k} A_k^{\mathrm{T}} + V_{1,k} \right) L_k^{\mathrm{T}} \right] - 2\mathrm{tr} \left[B_{\mathrm{e},k} (C_k \tilde{Q}_{1,k} A_k^{\mathrm{T}} + V_{12,k}^{\mathrm{T}}) L_k^{\mathrm{T}} \right] \\ & - 2\mathrm{tr} \left[A_{\mathrm{e},k} \tilde{Q}_{12,k}^{\mathrm{T}} A_k^{\mathrm{T}} L_k^{\mathrm{T}} \right] + \mathrm{tr} \left[B_{\mathrm{e},k} \left(C_k \tilde{Q}_{1,k} C_k^{\mathrm{T}} + V_{2,k} \right) B_{\mathrm{e},k}^{\mathrm{T}} \right] \\ & + \mathrm{tr} \left[A_{\mathrm{e},k} \tilde{Q}_{2,k} A_{\mathrm{e},k}^{\mathrm{T}} \right] + 2\mathrm{tr} \left[A_{\mathrm{e},k} \tilde{Q}_{12,k}^{\mathrm{T}} C_k^{\mathrm{T}} B_{\mathrm{e},k}^{\mathrm{T}} \right]. \end{aligned} \tag{2.18}$$

Next, assuming that $\tilde{Q}_{2,k}$ is invertible, we define $Q_k, \hat{Q}_k \in \mathbb{R}^{n_k \times n_k}$, $\tilde{V}_{2,k} \in \mathbb{R}^{p_k \times p_k}$, and $G_k \in \mathbb{R}^{n_{\mathrm{e},k} \times n_k}$ by

$$Q_k \triangleq \tilde{Q}_{1,k} - \tilde{Q}_{12,k} \tilde{Q}_{2,k}^{-1} \tilde{Q}_{12,k}^{\mathrm{T}}, \quad \hat{Q}_k \triangleq \tilde{Q}_{12,k} \tilde{Q}_{2,k}^{-1} \tilde{Q}_{12,k}^{\mathrm{T}}, \tag{2.19}$$

$$\tilde{V}_{2,k} \triangleq C_k Q_k C_k^{\mathrm{T}} + V_{2,k}, \tag{2.20}$$

$$G_k \triangleq \tilde{Q}_{2,k}^{-1} \tilde{Q}_{12,k}^{\mathrm{T}}. \tag{2.21}$$

Proposition 2.2.1. *Assume that $\tilde{Q}_{2,k}$ is positive definite and $A_{\mathrm{e},k}$ and $B_{\mathrm{e},k}$ minimize J_k. Then, $A_{\mathrm{e},k}$ and $B_{\mathrm{e},k}$ satisfy*

$$A_{\mathrm{e},k} = L_k \left(A_k - Q_{\mathrm{s},k} \tilde{V}_{2,k}^{-1} C_k \right) G_k^{\mathrm{T}}, \tag{2.22}$$

$$B_{\mathrm{e},k} = L_k Q_{\mathrm{s},k} \tilde{V}_{2,k}^{-1}, \tag{2.23}$$

where $Q_{\mathrm{s},k} \triangleq A_k Q_k C_k^{\mathrm{T}} + V_{12,k}$.

Proof. Setting $\frac{\partial J_k}{\partial A_{\mathrm{e},k}} = 0$, $\frac{\partial J_k}{\partial B_{\mathrm{e},k}} = 0$ and using (2.19)-(2.21) yields the result. □

Proposition 2.2.2. *Assume that $A_{\mathrm{e},k}$ and $B_{\mathrm{e},k}$ satisfy Proposition 2.2.1. Then,*

$$L_k \tilde{Q}_{12,k+1} = \tilde{Q}_{2,k+1}, \tag{2.24}$$

$$\tilde{Q}_{12,k+1} = \hat{Q}_{k+1} L_k^{\mathrm{T}}, \tag{2.25}$$

$$\tilde{Q}_{2,k+1} = L_k \hat{Q}_{k+1} L_k^{\mathrm{T}}. \tag{2.26}$$

Proof. Substituting (2.22) and (2.23) into (2.16) and (2.17) yields

$$\tilde{Q}_{12,k+1} = \left[A_k \hat{Q}_k A_k^{\mathrm{T}} + Q_{\mathrm{s},k} \tilde{V}_{2,k}^{-1} Q_{\mathrm{s},k}^{\mathrm{T}} \right] L_k^{\mathrm{T}}, \tag{2.27}$$

$$\tilde{Q}_{2,k+1} = L_k \left[A_k \hat{Q}_k A_k^{\mathrm{T}} + Q_{\mathrm{s},k} \tilde{V}_{2,k}^{-1} Q_{\mathrm{s},k}^{\mathrm{T}} \right] L_k^{\mathrm{T}}. \tag{2.28}$$

Pre-multiplying (2.27) by L_k yields $L_k \tilde{Q}_{12,k+1} = \tilde{Q}_{2,k+1}$. Using (2.19) and $L_k \tilde{Q}_{12,k+1} = \tilde{Q}_{2,k+1}$ yields $\tilde{Q}_{12,k+1} = \hat{Q}_{k+1} L_k^{\mathrm{T}}$ and $\tilde{Q}_{2,k+1} = L_k \hat{Q}_{k+1} L_k^{\mathrm{T}}$. □

Next, define $M_k \in \mathbb{R}^{n_k \times n_k}$ by

$$M_k \triangleq A_k \hat{Q}_k A_k^{\mathrm{T}} + Q_{\mathrm{s},k} \tilde{V}_{2,k}^{-1} Q_{\mathrm{s},k}^{\mathrm{T}}, \tag{2.29}$$

and define $\tau_k, \tau_{k\perp} \in \mathbb{R}^{n_k \times n_k}$ by

$$\tau_k \triangleq G_k^{\mathrm{T}} L_{k-1}, \quad \tau_{k\perp} \triangleq I - \tau_k. \tag{2.30}$$

Proposition 2.2.3. *Assume that $A_{\mathrm{e},k}$ and $B_{\mathrm{e},k}$ satisfy Proposition 2.2.1. Then, $\tau_{k+1}^2 = \tau_{k+1}$, that is, τ_{k+1} is an oblique projector.*

Proof. It follows from (2.29) that (2.27) and (2.28) can be expressed as

$$\tilde{Q}_{12,k+1} = M_k L_k^{\mathrm{T}}, \ \tilde{Q}_{2,k+1} = L_k M_k L_k^{\mathrm{T}}. \tag{2.31}$$

Hence, (2.30) implies that

$$\tau_{k+1} = M_k L_k^{\mathrm{T}} \left(L_k M_k L_k^{\mathrm{T}} \right)^{-1} L_k. \tag{2.32}$$

Therefore, $\tau_{k+1}^2 = \tau_{k+1}$. □

Proposition 2.2.4. *Assume that $A_{\mathrm{e},k}$ and $B_{\mathrm{e},k}$ satisfy Proposition 2.2.1. Then,*

$$\tau_{k+1}\hat{Q}_{k+1} = \hat{Q}_{k+1}. \tag{2.33}$$

Proof. It follows from (2.19) that

$$\hat{Q}_{k+1} = \tilde{Q}_{12,k+1}\tilde{Q}_{2,k+1}^{-1}\tilde{Q}_{12,k+1}^{\mathrm{T}}. \tag{2.34}$$

Substituting (2.31) into (2.34) yields

$$\hat{Q}_{k+1} = M_k L_k^{\mathrm{T}} \left(L_k M_k L_k^{\mathrm{T}}\right)^{-1} L_k M_k. \tag{2.35}$$

Hence, pre-multiplying (2.35) by τ_{k+1} and substituting (2.32) into the resulting expression yields (2.33).

Proposition 2.2.5. *Assume that $A_{\mathrm{e},k}$ and $B_{\mathrm{e},k}$ satisfy Proposition 2.2.1. Then,*

$$Q_{k+1} = A_k Q_k A_k^{\mathrm{T}} - Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}Q_{\mathrm{s},k}^{\mathrm{T}} + V_{1,k} + \tau_{k+1\perp}\left[A_k\hat{Q}_k A_k^{\mathrm{T}} + Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}Q_{\mathrm{s},k}^{\mathrm{T}}\right]\tau_{k+1\perp}^{\mathrm{T}}, \tag{2.36}$$

$$\hat{Q}_{k+1} = \tau_{k+1}\left[A_k\hat{Q}_k A_k^{\mathrm{T}} + Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}Q_{\mathrm{s},k}^{\mathrm{T}}\right]\tau_{k+1}^{\mathrm{T}}, \tag{2.37}$$

$$\tau_{k+1} = M_k L_k^{\mathrm{T}}\left(L_k M_k L_k^{\mathrm{T}}\right)^{-1} L_k. \tag{2.38}$$

Proof. It follows from (2.24) and (2.28) that

$$L_k\hat{Q}_{k+1}L_k^{\mathrm{T}} = L_k\left[A_k\hat{Q}_k A_k^{\mathrm{T}} + Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}Q_{\mathrm{s},k}^{\mathrm{T}}\right]L_k^{\mathrm{T}}. \tag{2.39}$$

Pre-multiplying and post-multiplying (2.39) by G_{k+1}^{T} and G_{k+1}, respectively, yields

$$\tau_{k+1}\hat{Q}_{k+1}\tau_{k+1}^{\mathrm{T}} = \tau_{k+1}\left[A_k\hat{Q}_k A_k^{\mathrm{T}} + Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}Q_{\mathrm{s},k}^{\mathrm{T}}\right]\tau_{k+1}^{\mathrm{T}}. \tag{2.40}$$

Hence, (2.37) follows from Proposition 2.2.4.

Since $\tilde{Q}_{12,k+1} = \hat{Q}_{k+1}L_k$, (2.27) and (2.30) imply that

$$\tau_{k+1}\hat{Q}_{k+1} = \tau_{k+1}\left[A_k\hat{Q}_kA_k^{\mathrm{T}} + Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}Q_{\mathrm{s},k}^{\mathrm{T}}\right]. \tag{2.41}$$

Therefore, (2.37) imply that

$$\tau_{k+1}\left[A_k\hat{Q}_kA_k^{\mathrm{T}} + Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}Q_{\mathrm{s},k}^{\mathrm{T}}\right] = \tau_{k+1}\left[A_k\hat{Q}_kA_k^{\mathrm{T}} + Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}Q_{\mathrm{s},k}^{\mathrm{T}}\right]\tau_{k+1}^{\mathrm{T}}. \tag{2.42}$$

Hence, $\hat{Q}_{k+1}$ can be expressed as

$$\hat{Q}_{k+1} = A_k\hat{Q}_kA_k^{\mathrm{T}} + Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}Q_{\mathrm{s},k}^{\mathrm{T}} - \tau_{k+1\perp}\left[A_k\hat{Q}_kA_k^{\mathrm{T}} + Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}Q_{\mathrm{s},k}^{\mathrm{T}}\right]\tau_{k+1\perp}^{\mathrm{T}}. \tag{2.43}$$

Furthermore, it follows from (2.15) and (2.19) that

$$Q_{k+1} = A_kQ_kA_k^{\mathrm{T}} + V_{1,k} + A_k\hat{Q}_kA_k^{\mathrm{T}} - \hat{Q}_{k+1}. \tag{2.44}$$

Therefore, substituting (2.43) into (2.44) yields (2.36). □

Note that although $A_{\mathrm{e},k}$ and $B_{\mathrm{e},k}$ depend on $\tilde{Q}_{12,k}$ and $\tilde{Q}_{2,k}$, it follows from Proposition 2.2.2 that $\tilde{Q}_{2,k}$ and $\tilde{Q}_{12,k}$ can be obtained from Q_k and $\hat{Q}_k$. Hence, it suffices to propagate Q_k and $\hat{Q}_k$ using (2.36) and (2.37), respectively.

Finally, we summarize the one-step reduced-order Kalman filter.

State update:

$$G_k = (L_k\hat{Q}_kL_k)^{-1}L_k\hat{Q}_k, \tag{2.45}$$

$$x_{\mathrm{e},k+1} = L_k\left(A_k - Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}C_k\right)G_k^{\mathrm{T}}x_{\mathrm{e},k} + L_kQ_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}y_k. \tag{2.46}$$

Covariance update:

$$Q_{k+1} = A_kQ_kA_k^{\mathrm{T}} - Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}Q_{\mathrm{s}}^{\mathrm{T}} + V_{1,k} + \tau_{k+1\perp}\left[A_k\hat{Q}_kA_k^{\mathrm{T}} + Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}Q_{\mathrm{s},k}^{\mathrm{T}}\right]\tau_{k+1\perp}^{\mathrm{T}}, \tag{2.47}$$

$$\hat{Q}_{k+1} = \tau_{k+1}\left[A_k\hat{Q}_kA_k^{\mathrm{T}} + Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}Q_{\mathrm{s},k}\right]\tau_{k+1}^{\mathrm{T}}, \tag{2.48}$$

$$M_k = A_k\hat{Q}_kA_k^{\mathrm{T}} + Q_{\mathrm{s},k}\tilde{V}_{2,k}^{-1}Q_{\mathrm{s},k}^{\mathrm{T}}, \tag{2.49}$$

$$\tau_{k+1} = M_kL_k^{\mathrm{T}}\left(L_kM_kL_k\right)^{-1}L_k. \tag{2.50}$$

2.3 Two-Step Reduced-Order Estimator

We now consider a two-step estimator. The *data assimilation step* is given by

$$x_{\mathrm{e},k}^{\mathrm{da}} = C_{\mathrm{e},k}^{\mathrm{f}} x_{\mathrm{e},k}^{\mathrm{f}} + D_{\mathrm{e},k}^{\mathrm{f}} y_k, \tag{2.51}$$

where $x_{\mathrm{e},k}^{\mathrm{da}} \in \mathbb{R}^{n_{\mathrm{e},k}}$ is the *reduced-order data assimilation estimate* of Lx_k, and $x_{\mathrm{e},k}^{\mathrm{f}} \in \mathbb{R}^{n_{\mathrm{e},k}}$ is the *reduced-order forecast estimate* of x_k. The *forecast step* or physics update of the estimator is given by

$$x_{\mathrm{e},k+1}^{\mathrm{f}} = A_{\mathrm{e},k}^{\mathrm{da}} x_{\mathrm{e},k}^{\mathrm{da}}. \tag{2.52}$$

Now, define the combined state and forecast estimate covariance $\tilde{Q}_k^{\mathrm{f}} \in \mathbb{R}^{\tilde{n}_k \times \tilde{n}_k}$ and the combined state and data assimilation estimate covariance $\tilde{Q}_k^{\mathrm{da}} \in \mathbb{R}^{\tilde{n}_k \times \tilde{n}_k}$ by

$$\tilde{Q}_k^{\mathrm{f}} \triangleq \mathcal{E}\left[\tilde{x}_k^{\mathrm{f}} (\tilde{x}_k^{\mathrm{f}})^{\mathrm{T}}\right], \quad \tilde{Q}_k^{\mathrm{da}} \triangleq \mathcal{E}\left[\tilde{x}_k^{\mathrm{da}} (\tilde{x}_k^{\mathrm{da}})^{\mathrm{T}}\right], \tag{2.53}$$

where $\tilde{x}_k^{\mathrm{f}}, \tilde{x}_k^{\mathrm{da}} \in \mathbb{R}^{n+n_{\mathrm{e}}}$ are defined by

$$\tilde{x}_k^{\mathrm{f}} \triangleq \begin{bmatrix} x_k \\ x_{\mathrm{e},k}^{\mathrm{f}} \end{bmatrix}, \quad \tilde{x}_k^{\mathrm{da}} \triangleq \begin{bmatrix} x_k \\ x_{\mathrm{e},k}^{\mathrm{da}} \end{bmatrix}. \tag{2.54}$$

Defining the *data assimilation cost*

$$J_k^{\mathrm{da}}\left(C_{\mathrm{e},k}^{\mathrm{f}}, D_{\mathrm{e},k}^{\mathrm{f}}\right) \triangleq \mathcal{E}\left[\left(L_k x_k - x_{\mathrm{e},k}^{\mathrm{da}}\right)^{\mathrm{T}} \left(L_k x_k - x_{\mathrm{e},k}^{\mathrm{da}}\right)\right], \tag{2.55}$$

(2.53) implies that

$$J_k^{\mathrm{da}}\left(C_{\mathrm{e},k}^{\mathrm{f}}, D_{\mathrm{e},k}^{\mathrm{f}}\right) = \mathrm{tr}(\tilde{Q}_k^{\mathrm{da}} \tilde{R}_k), \tag{2.56}$$

where $\tilde{R}_k$ is defined by (2.8).

Next, it follows from (2.1), (2.51), and (2.54) that

$$\tilde{x}_k^{\mathrm{da}} = \tilde{A}_k^{\mathrm{f}} \tilde{x}_k^{\mathrm{f}} + \tilde{D}_{1,k}^{\mathrm{f}} w_k, \tag{2.57}$$

where $\tilde{A}_k^{\mathrm{f}} \in \mathbb{R}^{\tilde{n}_k \times \tilde{n}_k}$ and $\tilde{D}_{1,k}^{\mathrm{f}} \in \mathbb{R}^{\tilde{n}_k \times d}$ are defined by

$$\tilde{A}_k^{\mathrm{f}} \triangleq \left[\begin{array}{cc} I & 0 \\ D_{\mathrm{e},k}^{\mathrm{f}} C_k & C_{\mathrm{e},k}^{\mathrm{f}} \end{array}\right], \quad \tilde{D}_{1,k}^{\mathrm{f}} \triangleq \left[\begin{array}{c} 0 \\ D_{\mathrm{e},k}^{\mathrm{f}} D_{2,k} \end{array}\right]. \tag{2.58}$$

Therefore,

$$\tilde{Q}_k^{\mathrm{da}} = \tilde{A}_k^{\mathrm{f}} \tilde{Q}_k^{\mathrm{f}} (\tilde{A}_k^{\mathrm{f}})^{\mathrm{T}} + \tilde{D}_{1,k}^{\mathrm{f}} (\tilde{D}_{1,k}^{\mathrm{f}})^{\mathrm{T}}. \tag{2.59}$$

Hence, J_k^{da} can be expressed as

$$J_k^{\mathrm{da}} \left(C_{\mathrm{e},k}^{\mathrm{f}}, D_{\mathrm{e},k}^{\mathrm{f}} \right) = \mathrm{tr} \left[\left(\tilde{A}_k^{\mathrm{f}} \tilde{Q}_k^{\mathrm{f}} (\tilde{A}_k^{\mathrm{f}})^{\mathrm{T}} + \tilde{D}_{1,k}^{\mathrm{f}} (\tilde{D}_{1,k}^{\mathrm{f}})^{\mathrm{T}} \right) \tilde{R}_k \right]. \tag{2.60}$$

Finally, partition $\tilde{Q}_k^{\mathrm{f}}$ as

$$\tilde{Q}_k^{\mathrm{f}} = \left[\begin{array}{cc} \tilde{Q}_{1,k}^{\mathrm{f}} & \tilde{Q}_{12,k}^{\mathrm{f}} \\ (\tilde{Q}_{12,k}^{\mathrm{f}})^{\mathrm{T}} & \tilde{Q}_{2,k}^{\mathrm{f}} \end{array}\right]. \tag{2.61}$$

so that substituting (2.58) into (2.60) yields

$$\begin{aligned} J_k^{\mathrm{da}} = {} & \mathrm{tr} \left[L_k \tilde{Q}_{1,k}^{\mathrm{f}} L_k^{\mathrm{T}} \right] - 2\mathrm{tr} \left[D_{\mathrm{e},k}^{\mathrm{f}} C_k \tilde{Q}_{1,k}^{\mathrm{f}} L_k^{\mathrm{T}} \right] - 2\mathrm{tr} \left[L_k \tilde{Q}_{12,k}^{\mathrm{f}} (C_{\mathrm{e},k}^{\mathrm{f}})^{\mathrm{T}} \right] + \mathrm{tr} \left[C_{\mathrm{e},k}^{\mathrm{f}} \tilde{Q}_{2,k}^{\mathrm{f}} (C_{\mathrm{e},k}^{\mathrm{f}})^{\mathrm{T}} \right] \\ & + 2\mathrm{tr} \left[D_{\mathrm{e},k}^{\mathrm{f}} C_k \tilde{Q}_{12,k}^{\mathrm{f}} (C_{\mathrm{e},k}^{\mathrm{f}})^{\mathrm{T}} \right] + \mathrm{tr} \left[D_{\mathrm{e},k}^{\mathrm{f}} \left(C_k \tilde{Q}_{1,k}^{\mathrm{f}} C_k^{\mathrm{T}} + V_{2,k} \right) (D_{\mathrm{e},k}^{\mathrm{f}})^{\mathrm{T}} \right]. \end{aligned} \tag{2.62}$$

Assuming that $\tilde{Q}_{2,k}^{\mathrm{f}}$ is invertible, define $Q_k^{\mathrm{f}}, \hat{Q}_k^{\mathrm{f}} \in \mathbb{R}^{n_k \times n_k}$ by

$$\begin{aligned} Q_k^{\mathrm{f}} & \triangleq \tilde{Q}_{1,k}^{\mathrm{f}} - \tilde{Q}_{12,k}^{\mathrm{f}} (\tilde{Q}_{2,k}^{\mathrm{f}})^{-1} (\tilde{Q}_{12,k}^{\mathrm{f}})^{\mathrm{T}}, \\ \hat{Q}_k^{\mathrm{f}} & \triangleq \tilde{Q}_{12,k}^{\mathrm{f}} (\tilde{Q}_{2,k}^{\mathrm{f}})^{-1} (\tilde{Q}_{12,k}^{\mathrm{f}})^{\mathrm{T}}. \end{aligned} \tag{2.63}$$

Finally, define $V_{2,k}^{\mathrm{f}} \in \mathbb{R}^{p_k \times p_k}$ by

$$V_{2,k}^{\mathrm{f}} \triangleq C_k Q_k^{\mathrm{f}} C_k^{\mathrm{T}} + V_{2,k} \tag{2.64}$$

and $G_k^{\mathrm{f}} \in \mathbb{R}^{n_{\mathrm{e},k} \times n_k}$ by

$$G_k^{\mathrm{f}} \triangleq (\tilde{Q}_{2,k}^{\mathrm{f}})^{-1} (\tilde{Q}_{12,k}^{\mathrm{f}})^{\mathrm{T}}. \tag{2.65}$$

The following result characterizes $C_{\mathrm{e},k}^{\mathrm{f}}$ and $D_{\mathrm{e},k}^{\mathrm{f}}$ that minimize J_k^{da}.

Proposition 2.3.1. *Assume that* $C^{\mathrm{f}}_{\mathrm{e},k}$ *and* $D^{\mathrm{f}}_{\mathrm{e},k}$ *minimize* J^{da}_k*, and assume that* $\tilde{Q}^{\mathrm{f}}_{2,k}$ *is positive definite. Then,*

$$C^{\mathrm{f}}_{\mathrm{e},k} = L_k \left(I - Q^{\mathrm{f}}_k C^{\mathrm{T}}_k (V^{\mathrm{f}}_{2,k})^{-1} C_k\right) (G^{\mathrm{f}}_k)^{\mathrm{T}}, \tag{2.66}$$

$$D^{\mathrm{f}}_{\mathrm{e},k} = L_k Q^{\mathrm{f}}_k C^{\mathrm{T}}_k (V^{\mathrm{f}}_{2,k})^{-1}. \tag{2.67}$$

Proof. Setting $\frac{\partial J^{\mathrm{da}}_k}{\partial C^{\mathrm{f}}_{\mathrm{e},k}} = 0$, $\frac{\partial J^{\mathrm{da}}_k}{\partial D^{\mathrm{f}}_{\mathrm{e},k}} = 0$ and using (2.63)-(2.65) yields the result. □

Next, partition $\tilde{Q}^{\mathrm{da}}_k$ as

$$\tilde{Q}^{\mathrm{da}}_k = \begin{bmatrix} \tilde{Q}^{\mathrm{da}}_{1,k} & \tilde{Q}^{\mathrm{da}}_{12,k} \\ (\tilde{Q}^{\mathrm{da}}_{12,k})^{\mathrm{T}} & \tilde{Q}^{\mathrm{da}}_{2,k} \end{bmatrix}. \tag{2.68}$$

Proposition 2.3.2. *Assume that* $x^{\mathrm{da}}_{\mathrm{e},k}$ *is given by* (2.51)*, and let* $C^{\mathrm{f}}_{\mathrm{e},k}$ *and* $D^{\mathrm{f}}_{\mathrm{e},k}$ *satisfy* (2.66)*,* (2.67)*. Then,*

$$\tilde{Q}^{\mathrm{da}}_{1,k} = \tilde{Q}^{\mathrm{f}}_{1,k}, \tag{2.69}$$

$$\tilde{Q}^{\mathrm{da}}_{12,k} = \left(\hat{Q}^{\mathrm{f}}_k + Q^{\mathrm{f}}_k C^{\mathrm{T}}_k (V^{\mathrm{f}}_{2,k})^{-1} C_k Q^{\mathrm{f}}_k\right) L^{\mathrm{T}}_k, \tag{2.70}$$

$$\tilde{Q}^{\mathrm{da}}_{2,k} = L_k \left(\hat{Q}^{\mathrm{f}}_k + Q^{\mathrm{f}}_k C^{\mathrm{T}}_k (V^{\mathrm{f}}_{2,k})^{-1} C_k Q^{\mathrm{f}}_k\right) L^{\mathrm{T}}_k. \tag{2.71}$$

Proof. It follows from (2.59) that $\tilde{Q}^{\mathrm{da}}_{1,k} = \tilde{Q}^{\mathrm{f}}_{1,k}$ and

$$\tilde{Q}^{\mathrm{da}}_{12,k} = \tilde{Q}^{\mathrm{f}}_{12,k} (C^{\mathrm{f}}_{\mathrm{e},k})^{\mathrm{T}} + \tilde{Q}^{\mathrm{f}}_{1,k} C^{\mathrm{T}}_k (D^{\mathrm{f}}_{\mathrm{e},k})^{\mathrm{T}}. \tag{2.72}$$

Substituting (2.66) and (2.67) into (2.72) yields (2.70). Similarly, it follows from (2.59) and (2.68) that

$$\begin{aligned} \tilde{Q}^{\mathrm{da}}_{2,k} = {} & C^{\mathrm{f}}_{\mathrm{e},k} \tilde{Q}^{\mathrm{f}}_{1,k} (C^{\mathrm{f}}_{\mathrm{e},k})^{\mathrm{T}} + C^{\mathrm{f}}_{\mathrm{e},k} (\tilde{Q}^{\mathrm{f}}_{12,k})^{\mathrm{T}} C^{\mathrm{T}}_k (D^{\mathrm{f}}_{\mathrm{e},k})^{\mathrm{T}} \\ & + D^{\mathrm{f}}_{\mathrm{e},k} C_k \tilde{Q}^{\mathrm{f}}_{12,k} (C^{\mathrm{f}}_{\mathrm{e},k})^{\mathrm{T}} + D^{\mathrm{f}}_{\mathrm{e},k} \left(C_k \tilde{Q}^{\mathrm{f}}_{1,k} C^{\mathrm{T}}_k + V_{2,k}\right) (D^{\mathrm{f}}_{\mathrm{e},k})^{\mathrm{T}}. \end{aligned} \tag{2.73}$$

Finally, substituting (2.66) and (2.67) into (2.73) yields (2.71). □

Next, define $Q_k^{\mathrm{da}} \in \mathbb{R}^{n_k \times n_k}$ and $\hat{Q}_k^{\mathrm{da}} \in \mathbb{R}^{n_k \times n_k}$ by

$$\begin{aligned} Q_k^{\mathrm{da}} &\triangleq \tilde{Q}_{1,k}^{\mathrm{da}} - \tilde{Q}_{12,k}^{\mathrm{da}} (\tilde{Q}_{2,k}^{\mathrm{da}})^{-1} (\tilde{Q}_{12,k}^{\mathrm{da}})^{\mathrm{T}}, \\ \hat{Q}_k^{\mathrm{da}} &\triangleq \tilde{Q}_{12,k}^{\mathrm{da}} (\tilde{Q}_{2,k}^{\mathrm{da}})^{-1} (\tilde{Q}_{12,k}^{\mathrm{da}})^{\mathrm{T}}. \end{aligned} \tag{2.74}$$

Corollary 2.3.1. *Assume that $C_{\mathrm{e},k}^{\mathrm{f}}$ and $D_{\mathrm{e},k}^{\mathrm{f}}$ satisfy Proposition 2.3.1. Then,*

$$L_k \tilde{Q}_{12,k}^{\mathrm{da}} = \tilde{Q}_{2,k}^{\mathrm{da}}, \;\; \tilde{Q}_{12,k}^{\mathrm{da}} = \hat{Q}_k^{\mathrm{da}} L_k^{\mathrm{T}}, \;\; \tilde{Q}_{2,k}^{\mathrm{da}} = L_k \hat{Q}_k^{\mathrm{da}} L_k^{\mathrm{T}}. \tag{2.75}$$

Next, define $G_k^{\mathrm{da}} \in \mathbb{R}^{n_{\mathrm{e},k} \times n_k}$ by

$$G_k^{\mathrm{da}} \triangleq (\tilde{Q}_{2,k}^{\mathrm{da}})^{-1} (\tilde{Q}_{12,k}^{\mathrm{da}})^{\mathrm{T}}. \tag{2.76}$$

Also, define $M_k^{\mathrm{da}} \in \mathbb{R}^{n_k \times n_k}$ by

$$M_k^{\mathrm{da}} \triangleq \hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} \tag{2.77}$$

and τ_k^{da} and $\tau_{k\perp}^{\mathrm{da}}$ by

$$\tau_k^{\mathrm{da}} \triangleq (G_k^{\mathrm{da}})^{\mathrm{T}} L_k, \;\; \tau_{k\perp}^{\mathrm{da}} \triangleq I - \tau_k^{\mathrm{da}}. \tag{2.78}$$

Proposition 2.3.3. *Assume that $C_{\mathrm{e},k}^{\mathrm{f}}$ and $D_{\mathrm{e},k}^{\mathrm{f}}$ satisfy Proposition 2.3.1. Then, τ_k^{da} is an oblique projector.*

Proof. The proof is similar to the proof of Proposition 2.4. □

Proposition 2.3.4. *Assume that $C_{\mathrm{e},k}^{\mathrm{f}}$ and $D_{\mathrm{e},k}^{\mathrm{f}}$ satisfy Proposition 2.3.1. Then,*

$$\tau_k^{\mathrm{da}} \hat{Q}_k^{\mathrm{da}} = \hat{Q}_k^{\mathrm{da}}. \tag{2.79}$$

Proof. The proof is similar to the proof of Proposition 2.5. □

Proposition 2.3.5. *Assume that $x_{\mathrm{e},k}^{\mathrm{da}}$ is given by* (2.51)*, and $C_{\mathrm{e},k}^{\mathrm{f}}$ and $D_{\mathrm{e},k}^{\mathrm{f}}$ satisfy Proposition 2.3.1. Then,*

$$\hat{Q}_k^{\mathrm{da}} = \tau_k^{\mathrm{da}} \left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} \right) (\tau_k^{\mathrm{da}})^{\mathrm{T}}, \tag{2.80}$$

$$Q_k^{\mathrm{da}} = Q_k^{\mathrm{f}} - Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} + \tau_{k\perp}^{\mathrm{da}} \left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} \right) (\tau_{k\perp}^{\mathrm{da}})^{\mathrm{T}}. \tag{2.81}$$

Proof. It follows from (2.71) and (2.75) that

$$L_k \hat{Q}_k^{\mathrm{da}} L_k^{\mathrm{T}} = L_k \left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} \right) L_k^{\mathrm{T}}. \tag{2.82}$$

Pre-multiplying and post-multiplying (2.82) by $(G_k^{\mathrm{da}})^{\mathrm{T}}$ and G_k^{da}, respectively, yields (2.80).

Next, it follows from (2.70), (2.75), and (2.78) that

$$\tau_k^{\mathrm{da}} \hat{Q}_k^{\mathrm{da}} = \tau_k^{\mathrm{da}} \left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} \right). \tag{2.83}$$

Therefore, Proposition 2.3.3 and (2.80) imply that

$$\tau_k^{\mathrm{da}} \left[\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} \right] = \tau_k^{\mathrm{da}} \left[\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} \right] (\tau_k^{\mathrm{da}})^{\mathrm{T}}. \tag{2.84}$$

Hence, $\hat{Q}_k^{\mathrm{da}}$ can be expressed as

$$\hat{Q}_k^{\mathrm{da}} = \hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} - \tau_{k\perp}^{\mathrm{da}} \left[\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} \right] (\tau_{k\perp}^{\mathrm{da}})^{\mathrm{T}}. \tag{2.85}$$

Finally, note that (2.69) implies that $Q_k^{\mathrm{da}} = \tilde{Q}_{1,k}^{\mathrm{f}} - \hat{Q}_k^{\mathrm{da}}$. Hence, (2.85) yields (2.81). □

Next, we define the *forecast cost* J_k^{f} by

$$J_k^{\mathrm{f}} \left(A_{\mathrm{e},k}^{\mathrm{da}} \right) \triangleq \mathcal{E} \left[\left(L_k x_{k+1} - x_{\mathrm{e},k+1}^{\mathrm{f}} \right) \left(L_k x_{k+1} - x_{\mathrm{e},k+1}^{\mathrm{f}} \right)^{\mathrm{T}} \right]. \tag{2.86}$$

Hence, it follows from (2.53) that

$$J_k^{\mathrm{f}} \left(A_{\mathrm{e},k}^{\mathrm{da}} \right) = \mathrm{tr} \left(\tilde{Q}_{k+1}^{\mathrm{f}} \tilde{R}_k \right). \tag{2.87}$$

It follows from (2.1) and (2.52) that

$$\tilde{x}_{k+1}^{\mathrm{f}} = \tilde{A}_k^{\mathrm{da}} \tilde{x}_k^{\mathrm{da}} + \tilde{D}_{1,k}^{\mathrm{da}} w_k, \tag{2.88}$$

where $\tilde{A}_k^{\mathrm{da}} \in \mathbb{R}^{\tilde{n}_k \times \tilde{n}_k}$ and $\tilde{D}_{1,k}^{\mathrm{da}} \in \mathbb{R}^{\tilde{n}_k \times d}$ are defined by

$$\tilde{A}_k^{\mathrm{da}} \triangleq \begin{bmatrix} A_k & 0 \\ 0 & A_{\mathrm{e},k}^{\mathrm{da}} \end{bmatrix}, \quad \tilde{D}_{1,k}^{\mathrm{da}} \triangleq \begin{bmatrix} D_{1,k} \\ 0 \end{bmatrix}. \tag{2.89}$$

Therefore,

$$\tilde{Q}^{\mathrm{f}}_{k+1} = \tilde{A}^{\mathrm{da}}_k \tilde{Q}^{\mathrm{da}}_k (\tilde{A}^{\mathrm{da}}_k)^{\mathrm{T}} + \tilde{D}^{\mathrm{da}}_{1,k} (\tilde{D}^{\mathrm{da}}_{1,k})^{\mathrm{T}}. \tag{2.90}$$

Proposition 2.3.6. *Assume that $A^{\mathrm{da}}_{\mathrm{e},k}$ minimizes J^{f}_k, and assume that $\tilde{Q}^{\mathrm{da}}_{2,k}$ is positive definite. Then*

$$A^{\mathrm{da}}_{\mathrm{e},k} = L_k A_k (G^{\mathrm{da}}_k)^{\mathrm{T}}. \tag{2.91}$$

Proof. Setting $\frac{\partial J^{\mathrm{f}}_k}{\partial A^{\mathrm{da}}_{\mathrm{e},k}} = 0$ yields the result. □

Proposition 2.3.7. *Assume that $A^{\mathrm{da}}_{\mathrm{e},k}$ satisfies* (2.91)*. Then,*

$$L_k \tilde{Q}^{\mathrm{f}}_{12,k+1} = \tilde{Q}^{\mathrm{f}}_{2,k+1}, \tag{2.92}$$

$$\tilde{Q}^{\mathrm{f}}_{12,k+1} = \hat{Q}^{\mathrm{f}}_{k+1} L^{\mathrm{T}}_k, \tag{2.93}$$

$$\tilde{Q}^{\mathrm{f}}_{2,k+1} = L_k \hat{Q}^{\mathrm{f}}_{k+1} L^{\mathrm{T}}_k. \tag{2.94}$$

Proof. The proof is similar to the proof of Proposition 2.3. □

Next, define M^{f}_k by

$$M^{\mathrm{f}}_k \triangleq A_k \hat{Q}^{\mathrm{da}}_k A^{\mathrm{T}}_k \tag{2.95}$$

and define τ^{f}_k and $\tau^{\mathrm{f}}_{k\perp}$ by

$$\tau^{\mathrm{f}}_k \triangleq (G^{\mathrm{f}}_k)^{\mathrm{T}} L_{k-1},\ \tau^{\mathrm{f}}_{k\perp} \triangleq I - \tau^{\mathrm{f}}_k. \tag{2.96}$$

Proposition 2.3.8. *Assume that $A^{\mathrm{da}}_{\mathrm{e},k}$ satisfies* (2.91)*. Then, τ^{f}_{k+1} is an oblique projector.*

Proof. The proof is similar to the proof of Proposition 2.4. □

Proposition 2.3.9. *Assume that $A^{\mathrm{da}}_{\mathrm{e},k}$ satisfies* (2.91)*. Then,*

$$\tau^{\mathrm{f}}_{k+1} \hat{Q}^{\mathrm{f}}_{k+1} = \hat{Q}^{\mathrm{f}}_{k+1}. \tag{2.97}$$

Proof. The proof is similar to the proof of Proposition 2.5. □

Proposition 2.3.10. *Assume that* $A_{\mathrm{e},k}^{\mathrm{da}}$ *satisfies* (2.91)*. Then,*

$$\hat{Q}_{k+1}^{\mathrm{f}} = \tau_{k+1}^{\mathrm{f}} A_k \hat{Q}_k^{\mathrm{da}} A_k^{\mathrm{T}} (\tau_{k+1}^{\mathrm{f}})^{\mathrm{T}}, \tag{2.98}$$

$$Q_{k+1}^{\mathrm{f}} = A_k Q_k^{\mathrm{da}} A_k^{\mathrm{T}} + V_{1,k} + \tau_{k+1\perp}^{\mathrm{f}} \left(A_k \hat{Q}_k^{\mathrm{da}} A_k^{\mathrm{T}} \right) (\tau_{k+1\perp}^{\mathrm{f}})^{\mathrm{T}}. \tag{2.99}$$

Proof. The proof is similar to the proof of Proposition 2.6. □

The two-step reduced order filter can be summarized as follows.

Data assimilation step:

$$x_{\mathrm{e},k}^{\mathrm{da}} = L_k \left(I - Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k \right) (G_k^{\mathrm{f}})^{\mathrm{T}} x_{\mathrm{e},k}^{\mathrm{f}} + L_k Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} y_k, \tag{2.100}$$

$$Q_k^{\mathrm{da}} = Q_k^{\mathrm{f}} - Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} + \tau_{k\perp}^{\mathrm{da}} \left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} \right) (\tau_{k\perp}^{\mathrm{da}})^{\mathrm{T}}, \tag{2.101}$$

$$\hat{Q}_k^{\mathrm{da}} = \tau_k^{\mathrm{da}} \left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} \right) (\tau_k^{\mathrm{da}})^{\mathrm{T}}, \tag{2.102}$$

$$\tau_k^{\mathrm{da}} = M_k^{\mathrm{da}} L_k^{\mathrm{T}} \left(L_k M_k^{\mathrm{da}} L_k^{\mathrm{T}} \right)^{-1} L_k, \tag{2.103}$$

$$M_k^{\mathrm{da}} = \hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}}. \tag{2.104}$$

Forecast step:

$$x_{\mathrm{e},k+1}^{\mathrm{f}} = L_k A_k (G_k^{\mathrm{da}})^{\mathrm{T}} x_{\mathrm{e},k}^{\mathrm{da}}, \tag{2.105}$$

$$\hat{Q}_{k+1}^{\mathrm{f}} = \tau_{k+1}^{\mathrm{f}} A_k \hat{Q}_k^{\mathrm{da}} A_k^{\mathrm{T}} (\tau_{k+1}^{\mathrm{f}})^{\mathrm{T}}, \tag{2.106}$$

$$Q_{k+1}^{\mathrm{f}} = A_k Q_k^{\mathrm{da}} A_k^{\mathrm{T}} + V_{1,k} + \tau_{k+1\perp}^{\mathrm{f}} \left(A_k \hat{Q}_k^{\mathrm{da}} A_k^{\mathrm{T}} \right) (\tau_{k+1\perp}^{\mathrm{f}})^{\mathrm{T}}, \tag{2.107}$$

$$\tau_{k+1}^{\mathrm{f}} = M_k^{\mathrm{f}} L_k^{\mathrm{T}} \left(L_k M_k^{\mathrm{f}} L_k^{\mathrm{T}} \right)^{-1} L_k, \tag{2.108}$$

$$M_k^{\mathrm{f}} = A_k \hat{Q}_k^{\mathrm{da}} A_k^{\mathrm{T}}. \tag{2.109}$$

2.4 Optimal Infinite-Horizon Reduced-Order Estimator Revisited

Consider the LTI system

$$x_{k+1} = A x_k + D_1 w_k, \tag{2.110}$$

$$y_k = C x_k + D_2 w_k, \tag{2.111}$$

where $x_k \in \mathbb{R}^n$, $y_k \in \mathbb{R}^p$, and $w_k \in \mathbb{R}^d$ is a white noise process with zero mean and unit covariance. We consider a reduced-order estimator

$$x_{\mathrm{e},k+1} \;=\; A_{\mathrm{e}}x_{\mathrm{e},k} + B_{\mathrm{e}}y_k, \tag{2.112}$$

where $x_{\mathrm{e},k} \in \mathbb{R}^{n_{\mathrm{e}}}$, and the cost

$$J(A_{\mathrm{e}}, B_{\mathrm{e}}) \triangleq \lim_{k\to\infty} \mathcal{E}\Big[\big(Lx_k - x_{\mathrm{e},k}\big)^{\mathrm{T}}\big(Lx_k - x_{\mathrm{e},k}\big)\Big]. \tag{2.113}$$

If $\tilde{A}$ is asymptotically stable, then

$$\tilde{Q} \triangleq \lim_{k\to\infty} \mathcal{E}[\tilde{x}_k\tilde{x}_k^{\mathrm{T}}]. \tag{2.114}$$

exists. Moreover, $\tilde{Q}$ and its nonnegative-definite dual $\tilde{P}$ are the unique solutions of the Lyapunov equations

$$\tilde{Q} = \tilde{A}\tilde{Q}\tilde{A}^{\mathrm{T}} + \tilde{V}, \tag{2.115}$$

$$\tilde{P} = \tilde{A}^{\mathrm{T}}\tilde{P}\tilde{A} + \tilde{R}, \tag{2.116}$$

where

$$\tilde{V} \triangleq \begin{bmatrix} V_1 & V_{12}B_{\mathrm{e}}^{\mathrm{T}} \\ B_{\mathrm{e}}V_{12}^{\mathrm{T}} & B_{\mathrm{e}}V_2B_{\mathrm{e}}^{\mathrm{T}} \end{bmatrix}, \; \tilde{R} \triangleq \begin{bmatrix} L^{\mathrm{T}}L & -L^{\mathrm{T}} \\ -L & I_{n_{\mathrm{e}}} \end{bmatrix}, \tag{2.117}$$

and

$$V_1 \triangleq D_1D_1^{\mathrm{T}}, \; V_{12} \triangleq D_1D_2^{\mathrm{T}}, \; V_2 \triangleq D_2D_2^{\mathrm{T}}. \tag{2.118}$$

Proposition 2.4.1. *Assume that $\tilde{Q}_2$ is positive definite and A_{e} and B_{e} minimize $J(A_{\mathrm{e}}, B_{\mathrm{e}})$ with constraint (2.115). Then, there exist nonnegative-definite matrices $Q, \hat{Q}, \hat{P} \in \mathbb{R}^{n\times n}$ such that A_{e} and B_{e} are given by*

$$A_{\mathrm{e}} = \Gamma\left[A - Q_{\mathrm{s}}\tilde{V}_2^{-1}C\right]G^{\mathrm{T}}, \tag{2.119}$$

$$B_{\mathrm{e}} = \Gamma Q_{\mathrm{s}}\tilde{V}_2^{-1}, \tag{2.120}$$

and $Q, \hat{Q}, \hat{P}$ *satisfy*

$$Q = AQA^{\mathrm{T}} - Q_{\mathrm{s}}\tilde{V}_2^{-1}Q_{\mathrm{s}}^{\mathrm{T}} + V_1 + \tau_\perp \left(A\hat{Q}A^{\mathrm{T}} + Q_{\mathrm{s}}\tilde{V}_2^{-1}Q_{\mathrm{s}}^{\mathrm{T}}\right)\tau_\perp^{\mathrm{T}}, \tag{2.121}$$

$$\hat{Q} = \tau \left(A\hat{Q}A^{\mathrm{T}} + Q_{\mathrm{s}}\tilde{V}_2^{-1}Q_{\mathrm{s}}^{\mathrm{T}}\right)\tau^{\mathrm{T}}, \tag{2.122}$$

$$\hat{P} = \tau^{\mathrm{T}}\left[\left(A - Q_{\mathrm{s}}\tilde{V}_2^{-1}C\right)^{\mathrm{T}}\hat{P}\left(A - Q_{\mathrm{s}}\tilde{V}_2^{-1}C\right) + L^{\mathrm{T}}L\right]\tau, \tag{2.123}$$

$$\tau \triangleq G^{\mathrm{T}}\Gamma = (\hat{Q}\hat{P})(\hat{Q}\hat{P})^{\#}, \tag{2.124}$$

$$\Gamma G^{\mathrm{T}} = I_{n_{\mathrm{e}}}, \tag{2.125}$$

$$\mathrm{rank}(\hat{Q}) = \mathrm{rank}(\hat{P}) = \mathrm{rank}(\hat{Q}\hat{P}) = n_{\mathrm{e}}, \tag{2.126}$$

where $Q_{\mathrm{s}} \triangleq AQC^{\mathrm{T}} + V_{12}$, $\tilde{V}_2 \triangleq CQC^{\mathrm{T}} + V_2$, *and* $\tau_\perp \triangleq I - \tau$.

Since $\Gamma G^{\mathrm{T}} = I_{n_{\mathrm{e}}}$, it follows that τ is an oblique projector. The notation $(\ )^{\#}$ indicates the group generalized inverse [47].

2.5 Optimal Finite-Horizon Subspace Estimator

We now consider reduced-order estimator that focuses on a specific subspace of the state. Without loss of generality, we partition the system (2.1), (2.2) as

$$\begin{bmatrix} x_{\mathrm{r},k+1} \\ x_{\mathrm{s},k+1} \end{bmatrix} = \begin{bmatrix} A_{\mathrm{r},k} & A_{\mathrm{us},k} \\ 0 & A_{\mathrm{s},k} \end{bmatrix} \begin{bmatrix} x_{\mathrm{r},k} \\ x_{\mathrm{s},k} \end{bmatrix} + \begin{bmatrix} D_{1\mathrm{r},k} \\ D_{1\mathrm{s},k} \end{bmatrix} w_k, \tag{2.127}$$

$$y_k = \begin{bmatrix} C_{\mathrm{r},k} & C_{\mathrm{s},k} \end{bmatrix} \begin{bmatrix} x_{\mathrm{r},k} \\ x_{\mathrm{s},k} \end{bmatrix} + D_{2,k}w_k, \tag{2.128}$$

and we seek a reduced-order subspace estimator

$$x_{\mathrm{e},k+1} = A_{\mathrm{e},k}x_{\mathrm{e},k} + B_{\mathrm{e},k}y_k, \tag{2.129}$$

$$y_{\mathrm{e},k} = C_{\mathrm{e},k}x_{\mathrm{e},k}, \tag{2.130}$$

that minimizes

$$J_k(A_{\mathrm{e},k}, B_{\mathrm{e},k}, C_{\mathrm{e},k+1}) \triangleq \mathcal{E}\left(\left[L_{k+1}x_{k+1} - y_{\mathrm{e},k+1}\right]^{\mathrm{T}} R_k \left[L_{k+1}x_{k+1} - y_{\mathrm{e},k+1}\right]\right). \tag{2.131}$$

In this formulation the plant state x_k is partitioned into subsystems for $x_{\mathrm{r},k}$ and $x_{\mathrm{s},k}$ of dimension $n_{\mathrm{r},k}$ and $n_{\mathrm{s},k}$, respectively. The state $x_{\mathrm{r},k}$ may contain the components of x_k of interest. Furthermore, the state weighting matrix L_k is partitioned as $L_k \triangleq [L_{\mathrm{r},k}\ L_{\mathrm{s},k}]$, where $L_{\mathrm{r},k}$ and $L_{\mathrm{s},k}$ have dimensions $q_k \times n_{\mathrm{r},k}$ and $q_k \times n_{\mathrm{s},k}$, respectively. The order $n_{\mathrm{e},k}$ of the estimator state $x_{\mathrm{e},k}$ is chosen to be $n_{\mathrm{r},k}$. Thus, the goal of the optimal reduced-order subspace estimator problem is to design an estimator of order $n_{\mathrm{r},k}$ that yields least-squares estimates of specified linear combinations of the states of the system.

We define the error state $z_k \triangleq x_{\mathrm{r},k} - x_{\mathrm{e},k}$, which satisfies

$$z_{k+1} = (A_{\mathrm{r},k} - B_{\mathrm{e},k}C_{\mathrm{r},k})x_{\mathrm{r},k} - A_{\mathrm{e},k}x_{\mathrm{e},k} + (A_{\mathrm{us},k} - B_{\mathrm{e},k}C_{\mathrm{s},k})x_{\mathrm{s},k} + (D_{1\mathrm{u},k} - B_{\mathrm{e},k}D_{2,k})w_k. \tag{2.132}$$

By constraining $A_{\mathrm{e},k} = A_{\mathrm{r},k} - B_{\mathrm{e},k}C_{\mathrm{r},k}$, (2.132) becomes

$$z_{k+1} = (A_{\mathrm{r},k} - B_{\mathrm{e},k}C_{\mathrm{r},k})z_k + (A_{\mathrm{us},k} - B_{\mathrm{e},k}C_{\mathrm{s},k})x_{\mathrm{s},k} + (D_{1\mathrm{u},k} - B_{\mathrm{e},k}D_{2,k})w_k.$$

Furthermore, the explicit dependence of the estimation error in (2.131) on the $x_{\mathrm{r},k}$ subsystem can be eliminated by constraining $C_{\mathrm{e},k} = L_{\mathrm{r},k}$. Now, from (2.127)-(2.130) it follows that

$$\tilde{x}_{k+1} = \tilde{A}_k\tilde{x}_k + \tilde{D}_k w_k, \tag{2.133}$$

where

$$\tilde{x}_k \triangleq \begin{bmatrix} z_k \\ x_{\mathrm{s},k} \end{bmatrix},\ \tilde{A}_k \triangleq \begin{bmatrix} A_{\mathrm{r},k} - B_{\mathrm{e},k}C_{\mathrm{r},k} & A_{\mathrm{us},k} - B_{\mathrm{e},k}C_{\mathrm{s},k} \\ 0 & A_{\mathrm{s},k} \end{bmatrix},\ \tilde{D}_k \triangleq \begin{bmatrix} D_{1\mathrm{r},k} - B_{\mathrm{e},k}D_{2,k} \\ D_{1\mathrm{s},k} \end{bmatrix}. \tag{2.134}$$

Then, the problem can be restated as finding $B_{\mathrm{e},k}$ that minimizes

$$J_k(B_{\mathrm{e},k}) = \mathrm{tr}\left(Q_{k+1}\tilde{R}_{k+1}\right), \tag{2.135}$$

where $\tilde{R}_{k+1} \triangleq L_{k+1}^{\mathrm{T}} R_k L_{k+1}$ and Q_k is the $n_k \times n_k$ state-error covariance $Q_k \triangleq \mathcal{E}\left[\tilde{x}_k \tilde{x}_k^{\mathrm{T}}\right]$.

Following the procedure in Section 2, we obtain the optimal finite-horizon reduced-order subspace estimator given by

$$x_{\mathrm{e},k+1} = \Phi_k A_k (I - Q_k C_k^{\mathrm{T}} \hat{V}_k^{-1} C_k) F_k^{\mathrm{T}} x_{\mathrm{e},k} + \Phi_k A_k Q_k C_k^{\mathrm{T}} \hat{V}_k^{-1} y_k, \tag{2.136}$$

$$Q_{k+1} = A_k Q_k A^{\mathrm{T}} - A_k Q_k C^{\mathrm{T}} \hat{V}_k^{-1} C_k Q_k A_k^{\mathrm{T}} + V_{1,k} + \mu_{k+1\perp} A_k Q_k C^{\mathrm{T}} \hat{V}_k^{-1} C_k Q_k A_k^{\mathrm{T}} \mu_{k+1\perp}^{\mathrm{T}}, \tag{2.137}$$

where $\mu_{k\perp} \triangleq I - \mu_k$, $\hat{V}_k \triangleq C_k Q_k C_k^{\mathrm{T}} + V_{2,k}$, $F_k \triangleq \left[I_{n_{\mathrm{r},k}} \; 0_{n_{\mathrm{r},k} \times n_{\mathrm{s},k}} \right]$,

$$\Phi_k \triangleq [I_{n_{\mathrm{r}}} \; (L_{\mathrm{r},k}^{\mathrm{T}} R_k L_{\mathrm{r},k})^{-1} (L_{\mathrm{r},k}^{\mathrm{T}} R_k L_{\mathrm{s},k})], \tag{2.138}$$

$$\mu_k \triangleq F_k^{\mathrm{T}} \Phi_k = \begin{bmatrix} I_{n_{\mathrm{r}}} & (L_{\mathrm{r},k}^{\mathrm{T}} R_k L_{\mathrm{r},k})^{-1} (L_{\mathrm{r},k}^{\mathrm{T}} R_k L_{\mathrm{s},k}) \\ 0 & 0 \end{bmatrix}. \tag{2.139}$$

Next, we consider two-step estimator. The *data assimilation step* is given by

$$x_{\mathrm{e},k}^{\mathrm{da}} = A_{\mathrm{e},k}^{\mathrm{f}} x_{\mathrm{e},k}^{\mathrm{f}} + B_{\mathrm{e},k} y_k, \tag{2.140}$$

$$y_{\mathrm{e},k}^{\mathrm{da}} = C_{\mathrm{e},k} x_{\mathrm{e},k}^{\mathrm{da}} \tag{2.141}$$

where $x_{\mathrm{e},k}^{\mathrm{da}} \in \mathbb{R}^{n_{\mathrm{e}}}$ is *the reduced-order data assimilation estimate of the subspace* $x_{\mathrm{u},k}$, and $x_{\mathrm{e},k}^{\mathrm{f}} \in \mathbb{R}^{n_{\mathrm{e}}}$ is *the reduced-order forecast estimate of subspace* $x_{\mathrm{u},k}$, while the *forecast step* is given by

$$x_{\mathrm{e},k+1}^{\mathrm{f}} = A_{\mathrm{e},k}^{\mathrm{da}} x_{\mathrm{e},k}^{\mathrm{da}}, \tag{2.142}$$

$$y_{\mathrm{e},k+1}^{\mathrm{f}} = C_{\mathrm{e},k+1} x_{\mathrm{e},k+1}^{\mathrm{f}}. \tag{2.143}$$

Defining the data-assimilation cost J_k^{da} and the forecast cost J_k^{f} as

$$J_k^{\mathrm{da}}(H_{\mathrm{e},k}, B_{\mathrm{e},k}, C_{\mathrm{e},k}) \triangleq \mathcal{E}\left(\left[L_k x_k - y_{\mathrm{e},k}^{\mathrm{da}} \right]^{\mathrm{T}} R_k \left[L_k x_k - y_{\mathrm{e},k}^{\mathrm{da}} \right] \right), \tag{2.144}$$

$$J_{k+1}^{\mathrm{f}}(A_{\mathrm{e},k}, C_{\mathrm{e},k+1}) \triangleq \mathcal{E}\left(\left[L_{k+1} x_{k+1} - y_{\mathrm{e},k+1}^{\mathrm{f}} \right]^{\mathrm{T}} R_k \left[L_{k+1} x_{k+1} - y_{\mathrm{e},k+1}^{\mathrm{f}} \right] \right), \tag{2.145}$$

we obtain the following two-step optimal finite-horizon subspace estimator:

Data assimilation step:

$$x_{\mathrm{e},k}^{\mathrm{da}} = \Phi_k(I - Q_k^{\mathrm{f}} C_k^{\mathrm{T}} \hat{V}_{2,k}^{-1} C_k) F_k^{\mathrm{T}} x_k^{\mathrm{da}} + \Phi_k Q_k^{\mathrm{f}} C_k^{\mathrm{T}} \hat{V}_{2,k}^{-1} y_k, \tag{2.146}$$

$$Q_k^{\mathrm{da}} = Q_k^{\mathrm{f}} - Q_k^{\mathrm{f}} C^{\mathrm{T}} \hat{V}_{2,k}^{-1} C_k Q_k^{\mathrm{f}} + \mu_{k\perp} Q_k^{\mathrm{f}} C^{\mathrm{T}} \hat{V}_{2,k}^{-1} C_k Q_k^{\mathrm{f}} \mu_{k\perp}^{\mathrm{T}}, \tag{2.147}$$

$$\hat{V}_{2,k} = C_k Q_k^{\mathrm{f}} C_k^{\mathrm{T}} + V_{2,k},\ \mu_k = \Phi_k F_k^{\mathrm{T}}. \tag{2.148}$$

Forecast step:

$$x_{\mathrm{e},k+1}^{\mathrm{f}} = \Phi_k A_k F_k^{\mathrm{T}} x_{\mathrm{e},k}^{\mathrm{da}}, \tag{2.149}$$

$$Q_{k+1}^{\mathrm{f}} = A_k Q_k^{\mathrm{da}} A_k^{\mathrm{T}} + V_{1,k}. \tag{2.150}$$

2.6 Optimal Infinite-Horizon Subspace Estimator

For the LTI system, the optimal infinite-horizon subspace estimator can be obtained by reformulating the cost

$$J(B_{\mathrm{e}}) \triangleq \lim_{k\to\infty} \mathcal{E}\left([Lx_k - y_{\mathrm{e},k}]^{\mathrm{T}} R [Lx_k - y_{\mathrm{e},k}] \right), \tag{2.151}$$

where we constrain

$$A_{\mathrm{e}} \triangleq A_{\mathrm{r}} - B_{\mathrm{e}} C_{\mathrm{r}}, \tag{2.152}$$

$$C_{\mathrm{e}} \triangleq L_{\mathrm{r}}. \tag{2.153}$$

If $\tilde{A}$ is asymptotically stable, then $Q \triangleq \lim_{k\to\infty} \mathcal{E}[\tilde{x}_k \tilde{x}_k^{\mathrm{T}}]$ exists.

Proposition 2.6.1. *Assume that B_{e} minimizes $J(B_{\mathrm{e}})$ with constraints (2.152) and (2.153). Then there exist nonnegative-definite matrices $Q, P \in \mathbb{R}^{n\times n}$ such that A_{e} and B_{e} are given by*

$$A_{\mathrm{e}} = \Phi A (I - Q C^{\mathrm{T}} \hat{V}^{-1} C) F^{\mathrm{T}}, \tag{2.154}$$

$$B_{\mathrm{e}} = \Phi A Q C^{\mathrm{T}} \hat{V}^{-1}, \tag{2.155}$$

and Q and P satisfy

$$Q = AQA^{\mathrm{T}} - AQC^{\mathrm{T}}\hat{V}^{-1}CQA^{\mathrm{T}} + \mu_{\perp}AQC^{\mathrm{T}}\hat{V}^{-1}CQA^{\mathrm{T}}\mu_{\perp}^{\mathrm{T}} + V_1, \tag{2.156}$$

$$P = A^{\mathrm{T}}PA - Q_a\mu^{\mathrm{T}}PA - A^{\mathrm{T}}P\mu Q_a^{\mathrm{T}} + Q_a\mu^{\mathrm{T}}P\mu Q_a^{\mathrm{T}} + L^{\mathrm{T}}RL, \tag{2.157}$$

where $\mu_{\perp} \triangleq I - \mu$, $\hat{V} \triangleq CQC^{\mathrm{T}} + V_2$, $Q_a \triangleq C^{\mathrm{T}}\hat{V}^{-1}Q$, $F \triangleq \begin{bmatrix} I_{n_{\mathrm{r}}} & 0_{n_{\mathrm{r}}\times n_{\mathrm{s}}} \end{bmatrix}$,

$$\Phi \triangleq [I_{n_{\mathrm{r}}} \;\; P_1^{-1}P_{12}], \tag{2.158}$$

$$\mu \triangleq F^{\mathrm{T}}\Phi = \begin{bmatrix} I_{n_{\mathrm{r}}} & P_1^{-1}P_{12} \\ 0 & 0 \end{bmatrix}. \tag{2.159}$$

2.7 Application to Periodically Time-Varying Multirate Estimation

Consider the transverse deflection $v(x,t)$ of a simply supported Euler-Bernoulli beam. The modal decomposition of $v(x,t)$ has the form

$$v(x,t) = \sum_{r=1}^{\infty} V_r(x)q_r(t), \;\; \int_0^l mV_r^2(x)\mathrm{d}x = 1, \;\; V_r(x) = \sqrt{\frac{2}{ml}}\sin\frac{r\pi x}{l}, \tag{2.160}$$

where the modal coordinates q_r satisfy

$$\ddot{q}_r(t) = 2\zeta\omega_r\dot{q}_r(t) + \omega_r^2 q_r(t) = \int_0^l f(x,t)V_r(x)\mathrm{d}x, \;\; r = 1, 2, \ldots. \tag{2.161}$$

For simplicity we assume $l = \pi$ and $m = 2/\pi$ so that $\sqrt{\frac{2}{ml}}$ =1. We assume that displacement sensors located at $x = 0.55\pi$ and $x = 0.65\pi$ are sampled at 50 Hz and 30 Hz, respectively. Also, it is assumed that a white noise disturbance of unit intensity acts on the beam at $x = 0.45\pi$. For estimator design, we weight the performance of the beam displacement at $x = 0.65\pi$. Finally, retaining the first five modes and defining the plant states as $x = [q_1, \dot{q}_1, \ldots, q_5, \dot{q}_5]^{\mathrm{T}}$, the resulting continuous-time

state-space model is

$$A = \underset{i=1,\ldots,5}{\text{block}-\text{diag}} \begin{bmatrix} 0 & 1 \\ -\omega_i^2 & -2\zeta\omega_i \end{bmatrix},\ \omega_i = i^2,\ ,i = 1,\ldots,5,\ \zeta = 0.005,$$

$$C = \begin{bmatrix} 0.9877 & 0 & -0.3090 & 0 & -0.8910 & 0 & 0.5878 & 0 & 0.7071 & 0 \\ 0.8910 & 0 & -0.8090 & 0 & -0.1564 & 0 & 0.9511 & 0 & -0.7071 & 0 \end{bmatrix},$$

$$L = \begin{bmatrix} 0.8910 & 0 & -0.8090 & 0 & -0.1564 & 0 & 0.9511 & 0 & -0.7071 & 0 \end{bmatrix},$$

$$D_1 = \begin{bmatrix} 0 & 0.9877 & 0 & 0.3090 & 0 & -0.8900 & 0 & -0.5878 & 0 & -0.7071 \end{bmatrix},$$

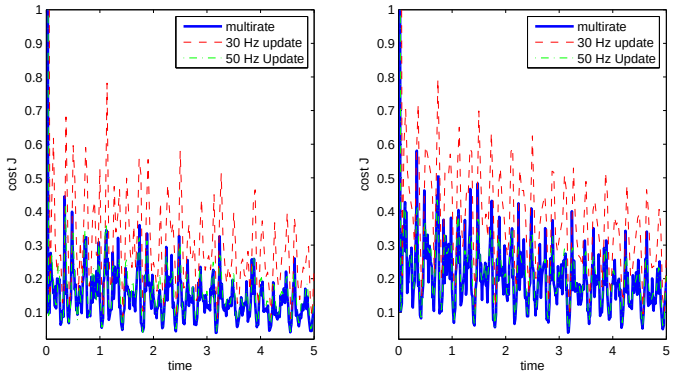
$$V_2 = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}.$$

The continuous-time model is discretized according to the given sample rates, which yields a time-varying system. Then the periodic sequence of sensor information is

$$\{s_1, s_2\}, \{s_1\}, \{s_2\}, \{s_1\}, \{s_1\}, \{s_2\}, \{s_1\}, \{s_1, s_2\}, \ldots \tag{2.162}$$

where s_1 and s_2 denote the signals from sensor 1 and sensor 2, respectively. The system is thus discretized according to the above sampling sequence as a periodically time-varying system.

Figure 2.1 shows the evolution of the costs of the one-step and two-step finite-horizon reduced-order estimators with $n = 10$, $n_e = 1$. The performance of the finite-horizon reduced-order estimators for the multirate system is compatible with the performance of the same estimator applied to single rate system where both signals are sampled at 50 Hz.



(a) (b)

Figure 2.1: Performance comparisons of reduced-order estimators when applied to the periodically time-varying multirate sampling system and fixed sample-rate systems. (a) is for the one-step reduced order estimator and (b) is for the two-step reduced-order estimator.

2.8 Asymptotically Stable Mass-Spring-Dashpot Example

We consider a zero-order hold discretized model of the mass-spring-dashpot structure consisting of 10 masses shown in Figure 2.2 so that $n = 20$. For $i = 1, \ldots, 10$, $m_i = 1.0$ kg, while, for $j = 1, \ldots, 11$, $k_j = 1.0$ N/m and $c_j = 0.05$ N-s/m. We set the initial error covariance $P_0 = 100I$, and we assume that $V_{1,k} = I$, $V_{2,k} = I$ for all $k \geqslant 0$.

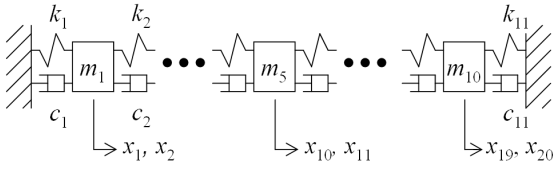


Figure 2.2: Mass-spring-dashpot system.

Let x_i denote the position of the ith mass so that

$$x \triangleq \begin{bmatrix} x_1 & \dot{x}_1 & \cdots & x_{10} & \dot{x}_{10} \end{bmatrix}. \tag{2.163}$$

We assume that measurements of position and velocities of $m_1, \ldots, m_4$ are available so that $C_k = [I_8 \ 0_{8\times 12}]$ for all $k \geqslant 0$. Next, we obtain state estimates from the

reduced-order estimator with $n_e = 8$. Meanwhile, for the subspace estimator, we consider a change of basis so that the system has the block upper triangular structure shown in (2.127). The costs for the estimators are defined in (2.6) and (2.131) with $R_k = I$. The ratio of the cost J_k to the best achievable cost when a full-order Kalman filter is used is shown in Figure 2.3. As indicated by ratios greater than 1, the performance of the reduced-order filter is never better than the full-order Kalman filter.

Next, we assume that measurements of positions and velocities of $m_1, \ldots, m_8$ are available so that $C_k = [I_{16}\ 0_{16\times 4}]$ for all $k \geqslant 0$. The performance of the reduced-order estimator with $n_e = 16$ is shown in Figure 2.3. The objective in both cases is to obtain estimates of Lx_k, where for $i = 1, \ldots, n_e$, $j = 1, \ldots, n$, the (i,j)th entry of $L \in \mathbb{R}^{n_e \times n}$ is given by

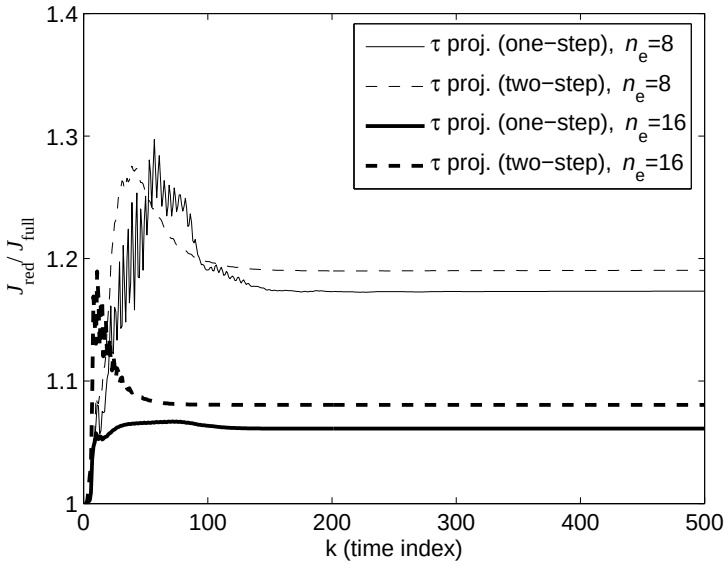
$$L_{(i,j)} = \begin{cases} 1, & \text{if } i = j, \\ 0.05, & \text{else.} \end{cases} \tag{2.164}$$

The plots also demonstrate that the one-step and two-step estimators are not equivalent.

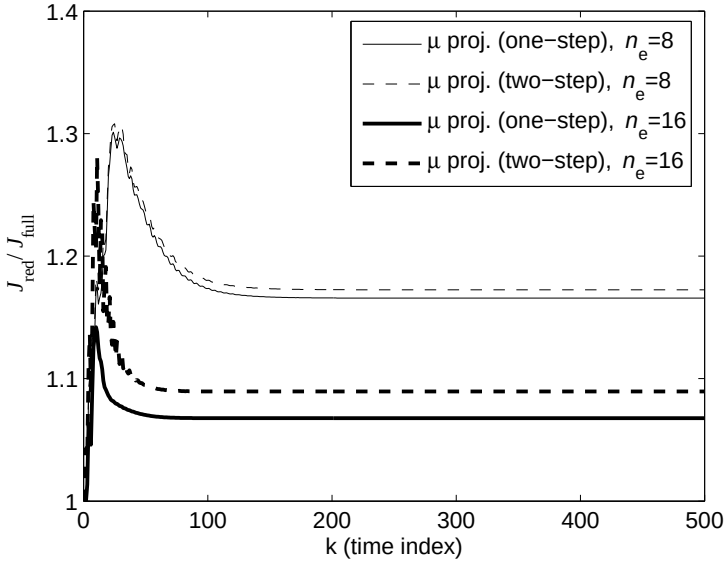
2.9 Mass-Spring-Dashpot Example with Rigid-Body Mode

Finally, we consider the case in which both ends of the mass-spring-dashpot structure are free, that is, $k_1 = k_{11} = 0.0$ and $c_1 = c_{11} = 0.0$, and thus the structure has an unstable rigid-body mode. Let q_i denote the position of i^{th} mode in modal coordinates so that

$$x \triangleq \begin{bmatrix} q_1 & \dot{q}_1 & \cdots & q_{10} & \dot{q}_{10} \end{bmatrix}. \tag{2.165}$$



(a)



(b)

Figure 2.3: Cost ratios for the (a) reduced-order and (b) subspace estimators for the asymptotically stable mass-spring-dashpot system. J_{red} is the estimation cost for the reduced-order estimator and J_{full} is for the full-order system. The plots also demonstrate that the one-step and two-step estimators are not equivalent.

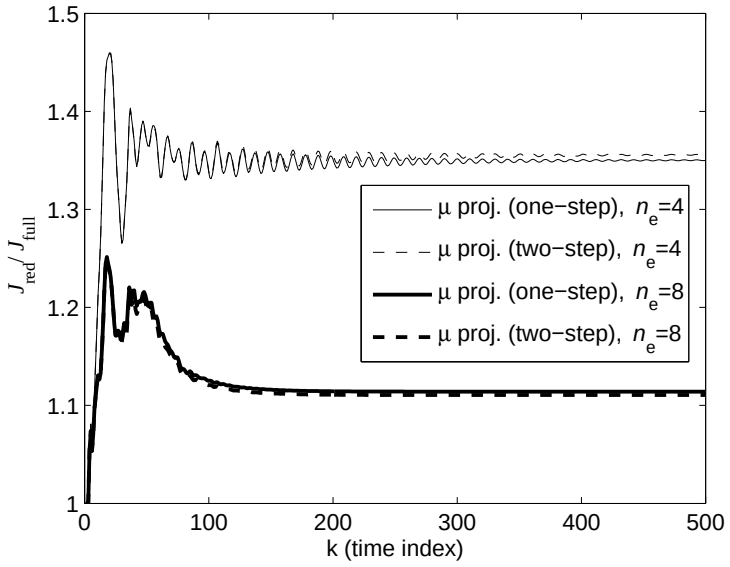


Figure 2.4: Cost ratios of J for the subspace estimator applied to the unstable mass-spring-dashpot system. The subspace estimator is able to handle the unstable modes in its filter structure.

We consider only the subspace estimator with $x_{\mathrm{r}} = \begin{bmatrix} q_1 & \dot{q}_1 \end{bmatrix}$. We assume that measurements of the position and velocity of m_1 are available and L is given by (2.164) in modal coordinates with $n_{\mathrm{e}} = 4, 8$. The performance of the subspace estimator with $n_{\mathrm{e}} = 4, 8$ is shown in Figure 2.4. The plots show that the subspace estimator is able to capture the unstable modes in the system.

2.10 Conclusion

Using finite-horizon optimization, an optimal reduced-order estimator and optimal fixed-structure subspace estimator were obtained in the form of recursive update equations for time-varying systems. These estimators are characterized by the oblique projectors τ and μ. Moreover, we derived one-step and two-step update equations for each estimator. When the order of each estimator is equal to the order of the system, the oblique projectors become the identity and the estimators are

equivalent to the classical optimal recursive full-order filter. We demonstrated the performance of the reduced-order and the subspace estimators for lumped structures.

CHAPTER III

State Estimation for Large-Scale Systems Based on Reduced-Order Error-Covariance Propagation

3.1 Introduction

The development of reduced-order state estimators has been of interest for several decades; representative work includes [12–26]. Most of these techniques involve data injection with an estimator whose order is less than the order of the plant. The estimator dynamics are typically obtained from the full-order dynamics by a truncation or projection process, while the estimator gain is obtained from a steady-state or updated error-covariance matrix based on the full-order dynamics.

For large-scale systems, however, reduced-order filters based on a full-order error covariance may not be feasible. In particular, the effort needed to compute the steady-state error covariance or to update the time-dependent error covariance is significant, namely, $\mathrm{O}(n^3)$ for a system of order n. Relevant applications include systems modeled by discretized partial differential equations such as weather forecasting [27, 48–52], where state estimation is generally referred to as data assimilation [53].

To overcome the $\mathrm{O}(n^3)$-computational burden of full-order-error-covariance-based estimation, we are interested in reduced-order filters based on a reduced-order error covariance. One such technique is developed in [27], where balancing is used to obtain a reduced-order model that provides the basis for the error-covariance update. By

using the reverse transformation to convert the reduced-order error covariance to a full-order error covariance in the original basis, data injection is performed on the full-order model so that estimates of all states are obtained in the original, physically meaningful basis. Although performance bounds are not available for this technique, the approach is consistent with the use of balancing in model reduction [54] while reducing the computational burden of the error-covariance update.

In the present chapter we compare the performance of the algorithm developed in [27] with several alternative algorithms. These alternative algorithms use balancing or truncation in various combinations to achieve a reduced-order-error-covariance for data injection with either the full-order model or a reduced-order model. Some of these algorithms use an initial balancing transformation, while others use an initial model truncation along with a steady-state error covariance. Algorithms that avoid the need for a balancing step are desirable when the system order is sufficiently high that balancing and transformation are prohibitive. For example, in weather applications, a state dimension greater than 10^6 is commonplace [27, 48–52].

As in [27], our study is primarily numerical, although we provide analytical performance bounds for the complementary steady-state error-covariance filters. Our goals in the present paper are thus to 1) clarify the nature of the reduced-order-error-covariance estimation problem, 2) present a collection of reduced-order-error-covariance estimators that are potentially useful in practice, and 3) numerically compare the performance of these filters on representative examples. This study is a precursor to the development of estimators for large-scale systems with nonlinear dynamics; preliminary results are described in [55].

In the classical Kalman filter, the full-order error covariance is propagated to obtain the estimator gain by which measurements are injected into the full state to

obtain optimal state estimates under uncertain disturbances and measurement noises. However, for large scale systems, propagation of the full-order error covariance is computationally infeasible. Hence, we consider reduced-order error-covariance filters. In the following subsections, we describe these filters. To fix notation, we begin with a brief review of the full-order Kalman filter.

3.2 Full-Order Kalman Filter (FOKF)

Consider the discrete-time LTI system

$$x_{k+1} = Ax_k + Gw_k, \tag{3.1}$$

$$y_k = Cx_k + v_k, \tag{3.2}$$

where $x_k \in \mathbb{R}^n$, $w_k \in \mathbb{R}^d$, $y_k, v_k \in \mathbb{R}^p$ and A, G, C are known real matrices of appropriate size. The plant disturbance Gw_k has the covariance $Q_k \triangleq \mathcal{E}[Gw_k w_k^{\mathrm{T}} G^{\mathrm{T}}]$, while the sensor noise v_k has the covariance $R_k \triangleq \mathcal{E}[v_k v_k^{\mathrm{T}}]$. The objective is to obtain an estimate x_k^{f} of the state x_k using measurements y_k to minimize $\mathrm{tr}P_k^{\mathrm{f}}$, where $e_k^{\mathrm{f}} \in \mathbb{R}^n$ is defined by

$$e_k^{\mathrm{f}} \triangleq x_k - x_k^{\mathrm{f}} \tag{3.3}$$

and the state-error covariance $P_k^{\mathrm{f}} \in \mathbb{R}^{n \times n}$ is defined by

$$P_k^{\mathrm{f}} \triangleq \mathcal{E}[e_k^{\mathrm{f}} e_k^{\mathrm{fT}}]. \tag{3.4}$$

The full-order Kalman filter is expressed in two steps, namely, the *forecast step*, which uses the model, and the *data assimilation step*, where the measurement is used to update the states. These steps can be summarized as follows:

Forecast Step

$$x_{k+1}^{\mathrm{f}} = Ax_k^{\mathrm{da}}, \tag{3.5}$$

$$P_{k+1}^{\mathrm{f}} = AP_k^{\mathrm{da}}A^{\mathrm{T}} + Q_k^{\mathrm{f}}. \tag{3.6}$$

Data Assimilation Step

$$K_k = P_k^{\mathrm{f}}C^{\mathrm{T}}(CP_k^{\mathrm{f}}C^{\mathrm{T}} + R_k)^{-1}, \tag{3.7}$$

$$P_k^{\mathrm{da}} = (I - K_kC)P_k^{\mathrm{f}}, \tag{3.8}$$

$$x_k^{\mathrm{da}} = x_k^{\mathrm{f}} + K_k(y_k - Cx_k^{\mathrm{f}}). \tag{3.9}$$

3.3 Kalman Filter with Error-Propagation in Balanced-Reduction Model (KFEBRM)

Applying the similarity transformation $x_k = T\hat{x}_k$, the system (3.1), (3.2) becomes

$$\hat{x}_{k+1} = \hat{A}\hat{x}_k + \hat{G}w_k, \tag{3.10}$$

$$y_k = \hat{C}\hat{x}_k + v_k, \tag{3.11}$$

where $\hat{A} \triangleq T^{-1}AT,\ \hat{G} \triangleq T^{-1}G,$ and $\hat{C} \triangleq CT.$

We choose the transformation T such that the controllability and observability gramians of the transformed system (3.10), (3.11) are diagonal and equal, that is, the system (3.10), (3.11) is a balanced realization of the system (3.1), (3.2). Then, we reduce the transformed system by retaining the dominant subspace as determined by the Hankel singular values $\sigma_1, \ldots, \sigma_n$, which describe the relative importance of each transformed state. The Hankel singular values $\sigma_1, \ldots, \sigma_n$ are the diagonal entries of the diagonal matrix Σ given by

$$\Sigma \triangleq \hat{W}_{\mathrm{c}} = \hat{W}_{\mathrm{o}}, \tag{3.12}$$

where $\hat{W}_{\mathrm{c}}$,$\hat{W}_{\mathrm{o}} \in \mathbb{R}^{n\times n}$ are the controllability and observability gramians of the transformed system, respectively. The reduced model of order n_r is given by

$$\hat{x}_{\mathrm{r},k+1} = \hat{A}_{\mathrm{r}}\hat{x}_{\mathrm{r},k} + \hat{G}_{\mathrm{r}}w_k, \tag{3.13}$$

$$y_{\mathrm{r},k} = \hat{C}_{\mathrm{r}}\hat{x}_{\mathrm{r},k} + v_k, \tag{3.14}$$

where $\hat{A}_{\mathrm{r}} \triangleq (T^{-1})_{\mathrm{r}}AT_{\mathrm{r}}$, $\hat{G}_{\mathrm{r}} \triangleq (T^{-1})_{\mathrm{r}}G$, $\hat{C}_{\mathrm{r}} \triangleq CT_{\mathrm{r}}$, $T_{\mathrm{r}} \triangleq T\left[I_{n_{\mathrm{r}}}\ 0_{n_{\mathrm{r}}\times(n-n_{\mathrm{r}})}\right]^{\mathrm{T}}$, and $(T^{-1})_{\mathrm{r}} \triangleq \left[I_{n_{\mathrm{r}}}\ 0_{n_{\mathrm{r}}\times(n-n_{\mathrm{r}})}\right]T^{-1}$.

The method used in [27] propagates the error covariance for a model of order $n_{\mathrm{r}} < n$ truncated according to the Hankel singular values. Furthermore, at each time step k, the full-order error covariance is approximated using the reduced-order model-error covariance by means of

$$\hat{P}^{\mathrm{f}}_k = T_{\mathrm{r}}P^{\mathrm{f}}_{\mathrm{r},k}T^{\mathrm{T}}_{\mathrm{r}}, \tag{3.15}$$

where $P^{\mathrm{f}}_{\mathrm{r},k}$ is the $n_{\mathrm{r}} \times n_{\mathrm{r}}$ reduced-order error-covariance matrix propagated for the reduced-order model (3.13), (3.14), and $\hat{P}^{\mathrm{f}}_k$ is the $n \times n$ approximate full-order error covariance matrix. The resulting forecast and data assimilation steps are given as follows:

Forecast Step

$$x^{\mathrm{f}}_{k+1} = Ax^{\mathrm{da}}_k, \tag{3.16}$$

$$P^{\mathrm{f}}_{\mathrm{r},k+1} = \hat{A}_{\mathrm{r}}P^{\mathrm{da}}_{\mathrm{r},k}\hat{A}^{\mathrm{T}}_{\mathrm{r}} + \hat{Q}^{\mathrm{f}}_{\mathrm{r},k}. \tag{3.17}$$

Data Assimilation Step

$$K_{\mathrm{r},k} \triangleq P^{\mathrm{f}}_{\mathrm{r},k}\hat{C}^{\mathrm{T}}_{\mathrm{r}}(\hat{C}_{\mathrm{r}}P^{\mathrm{f}}_{\mathrm{r},k}\hat{C}^{\mathrm{T}}_{\mathrm{r}} + R_k)^{-1}, \tag{3.18}$$

$$K_k = T_{\mathrm{r}}K_{\mathrm{r},k} = \hat{P}^{\mathrm{f}}_k C^{\mathrm{T}}(C\hat{P}^{\mathrm{f}}_k C^{\mathrm{T}} + R_k)^{-1}, \tag{3.19}$$

$$P^{\mathrm{da}}_{\mathrm{r},k} = (I - K_{\mathrm{r},k}\hat{C}_{\mathrm{r}})P^{\mathrm{f}}_{\mathrm{r},k}, \tag{3.20}$$

$$x^{\mathrm{da}}_k = x^{\mathrm{f}}_k + K_k(y_k - Cx^{\mathrm{f}}_k). \tag{3.21}$$

3.4 Localized Kalman Filter (LKF)

We now assume that the system (3.1), (3.2) can be partitioned as

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + \begin{bmatrix} G_1 w_k \\ G_2 w_k \end{bmatrix}, \tag{3.22}$$

$$y_k = \begin{bmatrix} C_1 & 0 \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + v_k, \tag{3.23}$$

where $x_{1,k} \in \mathbb{R}^{n_1}$ and $x_{2,k} \in \mathbb{R}^{n_2}$. Note that y_k depends only on $x_{1,k}$, which means physically that y_k is a local measurement. Truncating (3.22), (3.23) yields

$$x_{1,k+1} = A_{11} x_{1,k} + G_1 w_k, \tag{3.24}$$

$$y_k = C_1 x_{1,k} + v_k, \tag{3.25}$$

which is used for error-covariance propagation and data injection as follows:

Forecast Step

$$x^{\mathrm{f}}_{k+1} = A x^{\mathrm{da}}_k, \tag{3.26}$$

$$P^{\mathrm{f}}_{1,k+1} = A_{11} P^{\mathrm{da}}_{1,k} + Q^{\mathrm{f}}_{1,k}. \tag{3.27}$$

Data Assimilation Step

$$K_{1,k} = P^{\mathrm{f}}_{1,k} C_1^{\mathrm{T}} (C_1 P^{\mathrm{f}}_{1,k} C_1^{\mathrm{T}} + R_k)^{-1}, \tag{3.28}$$

$$P^{\mathrm{da}}_{1,k} = (I - K_{1,k} C_1) P^{\mathrm{f}}_{1,k}, \tag{3.29}$$

$$x^{\mathrm{da}}_{1,k} = x^{\mathrm{f}}_{1,k} + K_{1,k} (y_k - C_1 x^{\mathrm{f}}_{1,k}), \tag{3.30}$$

$$x^{\mathrm{da}}_{2,k} = x^{\mathrm{f}}_{2,k}. \tag{3.31}$$

In (3.27)-(3.29), $P_{1,k}$ is defined as the state-error covariance of the truncated system (3.24), (3.25), that is,

$$P^{\mathrm{f}}_{1,k} \triangleq \mathcal{E}[e^{\mathrm{f}}_{1,k} e^{\mathrm{fT}}_{1,k}], \tag{3.32}$$

where $e^{\mathrm{f}}_{1,k} \triangleq x_{1,k} - x^{\mathrm{f}}_{1,k}$.

3.5 Localized Kalman Filter with Balanced Reduction (LKFBR)

To apply LKF to the balanced system (3.10), (3.11), we first partition the transformed system (3.10), (3.11) such that

$$\begin{bmatrix} \hat{x}_{1,k+1} \\ \hat{x}_{2,k+1} \end{bmatrix} = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{bmatrix} \begin{bmatrix} \hat{x}_{1,k} \\ \hat{x}_{2,k} \end{bmatrix} + \begin{bmatrix} \hat{G}_1 w_k \\ \hat{G}_2 w_k \end{bmatrix}, \tag{3.33}$$

$$y_k = \begin{bmatrix} \hat{C}_1 & \hat{C}_2 \end{bmatrix} \begin{bmatrix} \hat{x}_{1,k} \\ \hat{x}_{2,k} \end{bmatrix} + v_k, \tag{3.34}$$

where the dimension of $\hat{x}_{1,k}$ is determined according to the Hankel singular values. Truncating (3.33), (3.34) yields

$$\hat{x}_{1,k+1} = \hat{A}_{11} x_{1,k} + \hat{G}_1 w_k, \tag{3.35}$$

$$y_{\mathrm{r},k} = \hat{C}_1 x_{1,k} + v_k, \tag{3.36}$$

which is used for error-covariance propagation and data injection using the LKF procedures (3.26) - (3.31). Finally, in order to compare the estimates to those of LKF without balanced model reduction given in (3.26), we transform the estimates back to the original coordinates using $x^{\mathrm{f}}_k = T\hat{x}^{\mathrm{f}}_k$.

3.6 Localized Kalman Filter with Residual-Subspace Balanced Truncation (LKFRBT)

We can account for the $x_{2,k}$ subsystem in the LKF algorithm by reducing the $x_{2,k}$ subsystem and then augmenting the $x_{1,k}$ subsystem with the reduced $x_{2,k}$ subsystem.

To do this, the dynamics of $x_{2,k} \in \mathbb{R}^{n_2}$ in (3.22) are expressed as

$$x_{2,k+1} = A_{22}x_{2,k} + [A_{21}\ G_2] \begin{bmatrix} x_{1,k} \\ w_k \end{bmatrix}, \tag{3.37}$$

$$z_k = A_{12}x_{2,k}, \tag{3.38}$$

to which we apply balanced realization and reduction. The resulting reduced-order model is

$$\hat{x}_{2,\mathrm{r},k+1} = (\hat{T}^{-1})_{\mathrm{r}}A_{22}\hat{T}_{\mathrm{r}}\hat{x}_{2,\mathrm{r},k} + (\hat{T}^{-1})_{\mathrm{r}}\,[A_{21}\ G_2] \begin{bmatrix} x_{1,k} \\ w_k \end{bmatrix}, \tag{3.39}$$

$$z_{\mathrm{r},k} = A_{12}\hat{T}_{\mathrm{r}}x_{2,\mathrm{r},k}, \tag{3.40}$$

where $\hat{T}$ is the balanced transformation for $x_{2,k}$ subsystem (3.37), (3.38), $\hat{x}_{2,\mathrm{r},k} \in \mathbb{R}^{n_{\mathrm{r}}}$, where $n_{\mathrm{r}} < n_2$ is the reduced approximation of $\hat{x}_{2,k} \triangleq \hat{T}^{-1}x_{2,k}$, $\hat{T}_{\mathrm{r}} \triangleq \hat{T}\left[I_{n_{\mathrm{r}}}\ 0_{n_{\mathrm{r}}\times(n_2-n_{\mathrm{r}})}\right]^{\mathrm{T}}$, and $(\hat{T}^{-1})_{\mathrm{r}} \triangleq \left[I_{n_{\mathrm{r}}}\ 0_{n_{\mathrm{r}}\times(n_2-n_{\mathrm{r}})}\right]\hat{T}^{-1}$.

By replacing the corresponding terms of (3.22), (3.23) with terms of (3.39), (3.40), we obtain

$$\begin{bmatrix} x_{1,k+1} \\ \hat{x}_{2,\mathrm{r},k+1} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12}\hat{T}_{\mathrm{r}} \\ (\hat{T}^{-1})_{\mathrm{r}}A_{21} & (\hat{T}^{-1})_{\mathrm{r}}A_{22}\hat{T}_{\mathrm{r}} \end{bmatrix} \begin{bmatrix} x_{1,k} \\ \hat{x}_{2,\mathrm{r},k} \end{bmatrix} + \begin{bmatrix} G_1 \\ (\hat{T}^{-1})_{\mathrm{r}}G_2 \end{bmatrix} w_k, \tag{3.41}$$

$$y_k = \begin{bmatrix} C_1\ 0 \end{bmatrix} \begin{bmatrix} x_{1,k} \\ \hat{x}_{2,\mathrm{r},k} \end{bmatrix} + v_k. \tag{3.42}$$

The error covariance is propagated by the reduced system (3.41), (3.42) whose dimension is $n_1 + n_{\mathrm{r}}$. The forecast and data assimilation steps are the same as those of

KFEBRM (3.16)-(3.21), replacing the reduced-order system (3.13), (3.14) with the system (3.41), (3.42), where $T_{\mathrm{r}} \in \mathbb{R}^{n\times(n_1+n_{\mathrm{r}})}$ is defined by

$$T_{\mathrm{r}} \triangleq \begin{bmatrix} I_{n_1\times n_1} & 0 \\ 0 & \hat{T}_{\mathrm{r}} \end{bmatrix}. \tag{3.43}$$

3.7 Complementary Static Open-Loop Steady-State (OLSS) Error-Covariance-Based Gain

KFEBRM, LKFBR, and LKFRBT account for interactions with the truncated subsystem by means of balanced reduction. Rather than using balanced reduction, we now compensate the reduced-order error-covariance of LKF with a complementary open-loop or closed-loop steady-state error-covariance. We begin by proving that the performance of an estimator that uses a steady-state open-loop error-covariance-based static gain is better than or equal to the open-loop performance. The proofs provide a justification for the complementary steady-state error-covariance approach.

Consider the system (3.1), (3.2), where A is asymptotically stable, the plant disturbance Gw_k and the measurement noise v_k are mutually independent, stationary random so that Q is positive semidefinite and R is positive definite.

Consider the corresponding open-loop estimator

$$\hat{x}_{\mathrm{OL},k+1} = A\hat{x}_{\mathrm{OL},k}, \tag{3.44}$$

and define the open-loop state-estimation error

$$e_{\mathrm{OL},k} \triangleq x_k - \hat{x}_{\mathrm{OL},k} \tag{3.45}$$

and the open-loop error covariance

$$P_{\mathrm{OL},k} \triangleq \mathcal{E}\left[e_{\mathrm{OL},k}e_{\mathrm{OL},k}^{\mathrm{T}}\right]. \tag{3.46}$$

Note that

$$P_{\mathrm{OL},k+1} = AP_{\mathrm{OL},k}A^{\mathrm{T}} + Q. \tag{3.47}$$

Since A is asymptotically stable, the steady-state open-loop error covariance

$$P_{\mathrm{OL}} \triangleq \lim_{k\to\infty} P_{\mathrm{OL},k} \tag{3.48}$$

exists and satisfies

$$P_{\mathrm{OL}} = AP_{\mathrm{OL}}A^{\mathrm{T}} + Q. \tag{3.49}$$

Next, using the steady-state open-loop error covariance, we define the estimator gain

$$K_{\mathrm{OL}} \triangleq AP_{\mathrm{OL}}C^{\mathrm{T}}\hat{R}_{\mathrm{OL}}^{-1}, \tag{3.50}$$

where $\hat{R}_{\mathrm{OL}} \triangleq CP_{\mathrm{OL}}C^{\mathrm{T}} + R$.

Consider the estimator

$$\hat{x}_{k+1} = A\hat{x}_k + K_{\mathrm{OL}}(y_k - C\hat{x}_k) \tag{3.51}$$

and define the state-estimation error

$$e_k \triangleq x_k - \hat{x}_k, \tag{3.52}$$

and the error covariance

$$P_k \triangleq \mathcal{E}\left[e_k e_k^{\mathrm{T}}\right]. \tag{3.53}$$

Note that we use the one-step estimator (3.51), which is equivalent to the two-step estimator composed of the forecast step and the data assimilation step. Next, subtracting (3.51) from (3.1) yields

$$e_{k+1} = (A - K_{\mathrm{OL}}C)e_k + Gw_k - K_{\mathrm{OL}}v_k. \tag{3.54}$$

Hence, (3.53) implies that error covariance P_k satisfies

$$\begin{aligned} P_{k+1} = (A - K_{\mathrm{OL}}C)P_k(A - K_{\mathrm{OL}}C)^{\mathrm{T}} + Q \\ + K_{\mathrm{OL}}RK_{\mathrm{OL}}^{\mathrm{T}}. \end{aligned} \tag{3.55}$$

The following result guarantees that the performance of the estimator based on the open-loop error covariance is better than the performance of the open-loop estimator.

Proposition 3.7.1. *Assume that $P_k \leqslant P_{\mathrm{OL}}$. Then $P_{k+1} \leqslant P_{\mathrm{OL}}$.*

Proof. Subtracting (3.55) from (3.49) yields

$$\begin{aligned} P_{\mathrm{OL}} - P_{k+1} = AP_{\mathrm{OL}}A^{\mathrm{T}} - (A - K_{\mathrm{OL}}C)P_k(A - K_{\mathrm{OL}}C)^{\mathrm{T}} \\ - K_{\mathrm{OL}}RK_{\mathrm{OL}}^{\mathrm{T}}. \end{aligned} \tag{3.56}$$

Adding and subtracting $(A-K_{\mathrm{OL}}C)P_{\mathrm{OL}}(A-K_{\mathrm{OL}}C)^{\mathrm{T}}$ to the right hand side of (3.56) yields

$$\begin{aligned} P_{\mathrm{OL}} - P_{k+1} =& (A - K_{\mathrm{OL}}C)(P_{\mathrm{OL}} - P_k)(A - K_{\mathrm{OL}}C)^{\mathrm{T}} \\ &+ AP_{\mathrm{OL}}C^{\mathrm{T}}K_{\mathrm{OL}}^{\mathrm{T}} + K_{\mathrm{OL}}CP_{\mathrm{OL}}A^{\mathrm{T}} \\ &- K_{\mathrm{OL}}\hat{R}_{\mathrm{OL}}K_{\mathrm{OL}}^{\mathrm{T}}. \end{aligned} \tag{3.57}$$

Substituting (3.50) into (3.57) yields

$$\begin{aligned} P_{\mathrm{OL}} - P_{k+1} =& (A - K_{\mathrm{OL}}C)(P_{\mathrm{OL}} - P_k)(A - K_{\mathrm{OL}}C)^{\mathrm{T}} \\ &+ AP_{\mathrm{OL}}C^{\mathrm{T}}\hat{R}_{\mathrm{OL}}^{-1}CP_{\mathrm{OL}}A^{\mathrm{T}}. \end{aligned} \tag{3.58}$$

Hence, it follows from (3.50) that

$$\begin{aligned} P_{\mathrm{OL}} - P_{k+1} =& (A - K_{\mathrm{OL}}C)(P_{\mathrm{OL}} - P_k)(A - K_{\mathrm{OL}}C)^{\mathrm{T}} \\ &+ K_{\mathrm{OL}}\hat{R}_{\mathrm{OL}}K_{\mathrm{OL}}^{\mathrm{T}}. \end{aligned} \tag{3.59}$$

Therefore, $P_{k+1} \leqslant P_{\mathrm{OL}}$. □

Hence, if $P_0 \leqslant P_{\mathrm{OL}}$, then, for all $k \geqslant 0$, $P_k \leqslant P_{\mathrm{OL}}$, which implies that the performance of the estimator based on the open-loop error covariance is never worse than the performance of the open-loop estimator. Next, we present a related result in which the estimator gain is based not on the open-loop error covariance but on a closed-loop error covariance obtained by using initial arbitrary gain $K \in \mathbb{R}^{n \times p}$.

3.8 Complementary Static Closed-Loop Steady-State (CLSS) Error-Covariance-Based Gain

Consider the closed-loop estimator

$$\hat{x}_{\mathrm{CL},k+1} = A\hat{x}_{\mathrm{CL},k} + K(y_k - C\hat{x}_{\mathrm{CL},k}), \tag{3.60}$$

where K is an estimator gain chosen so that $A - KC$ is asymptotically stable. Define the closed-loop state-estimation error

$$e_{\mathrm{CL},k} \triangleq x_k - \hat{x}_k, \tag{3.61}$$

and the closed-loop error covariance

$$P_{\mathrm{CL},k} \triangleq \mathcal{E}\left[e_{\mathrm{CL},k} e_{\mathrm{CL},k}^{\mathrm{T}}\right]. \tag{3.62}$$

Subtracting (3.60) from (3.1) yields

$$e_{\mathrm{CL},k+1} = (A - KC)e_{\mathrm{CL},k} + Gw_k - Kv_k \tag{3.63}$$

so that

$$\begin{aligned} P_{\mathrm{CL},k+1} = {} & (A - KC)P_{\mathrm{CL},k}(A - KC)^{\mathrm{T}} + Q \\ & + KRK^{\mathrm{T}}. \end{aligned} \tag{3.64}$$

Since $A - KC$ is asymptotically stable, the steady-state closed-loop error covariance defined by

$$P_{\mathrm{CL}} \triangleq \lim_{k \to \infty} P_{\mathrm{CL},k} \tag{3.65}$$

exists and satisfies the Lyapunov equation

$$P_{\mathrm{CL}} = (A - KC)P_{\mathrm{CL}}(A - KC)^{\mathrm{T}} + Q + KRK^{\mathrm{T}}. \tag{3.66}$$

Next, using steady-state closed-loop error covariance we define the estimator gain

$$K_{\mathrm{CL}} \triangleq AP_{\mathrm{CL}}C^{\mathrm{T}}\hat{R}_{\mathrm{CL}}^{-1}, \tag{3.67}$$

where

$$\hat{R}_{\mathrm{CL}} \triangleq CP_{\mathrm{CL}}C^{\mathrm{T}} + R. \tag{3.68}$$

Consider an estimator based on the estimator gain in (3.67), that is,

$$\hat{x}_{k+1} = A\hat{x}_k + K_{\mathrm{CL}}(y_k - C\hat{x}_k). \tag{3.69}$$

Subtracting (3.69) from (3.1) yields the error dynamics

$$e_{k+1} = (A - K_{\mathrm{CL}}C)e_k + Gw_k - K_{\mathrm{CL}}v_k. \tag{3.70}$$

Then the error covariance defined in (3.53) is propagated using

$$\begin{aligned} P_{k+1} = {} & (A - K_{\mathrm{CL}}C)P_k(A - K_{\mathrm{CL}}C)^{\mathrm{T}} + Q \\ & + K_{\mathrm{CL}}RK_{\mathrm{CL}}^{\mathrm{T}}. \end{aligned} \tag{3.71}$$

The following result shows that the performance of the estimator based on the closed-loop error covariance P_{CL} is better than the performance of the estimator in (3.60).

Proposition 3.8.1. *Assume that $P_k \leqslant P_{\mathrm{CL}}$. Then, $P_{k+1} \leqslant P_{\mathrm{CL}}$.*

Proof. Subtracting (3.71) from (3.66) yields

$$\begin{aligned} P_{\mathrm{CL}} - P_{k+1} = {} & (A - KC)P_{\mathrm{CL}}(A - KC)^{\mathrm{T}} \\ & - (A - K_{\mathrm{CL}}C)P_k(A - K_{\mathrm{CL}}C)^{\mathrm{T}} \\ & + KRK^{\mathrm{T}} - K_{\mathrm{CL}}RK_{\mathrm{CL}}^{\mathrm{T}}. \end{aligned} \tag{3.72}$$

Adding and subtracting $(A-K_{\mathrm{CL}}C)P_{\mathrm{CL}}(A-K_{\mathrm{CL}}C)^{\mathrm{T}}$ to the right hand side of (3.72) yields

$$\begin{aligned} P_{\mathrm{CL}} - P_{k+1} &= (A - K_{\mathrm{CL}}C)(P_{\mathrm{CL}} - P_k)(A - K_{\mathrm{CL}}C)^{\mathrm{T}} \\ &\quad + KRK^{\mathrm{T}} - K_{\mathrm{CL}}RK_{\mathrm{CL}}^{\mathrm{T}} - KCP_{\mathrm{CL}}A^{\mathrm{T}} \\ &\quad - AP_{\mathrm{CL}}C^{\mathrm{T}}K^{\mathrm{T}} + KCP_{\mathrm{CL}}C^{\mathrm{T}}K^{\mathrm{T}} \\ &\quad + K_{\mathrm{CL}}CP_{\mathrm{CL}}A^{\mathrm{T}} + AP_{\mathrm{CL}}C^{\mathrm{T}}K_{\mathrm{CL}}^{\mathrm{T}} \\ &\quad - K_{\mathrm{CL}}CP_{\mathrm{CL}}C^{\mathrm{T}}K_{\mathrm{CL}}^{\mathrm{T}}. \end{aligned} \tag{3.73}$$

Using (3.67) and (3.68) in (3.73) yields

$$\begin{aligned} P_{\mathrm{CL}} - P_{k+1} &= (A - K_{\mathrm{CL}}C)(P_{\mathrm{CL}} - P_k)(A - K_{\mathrm{CL}}C)^{\mathrm{T}} \\ &\quad + K\hat{R}_{\mathrm{CL}}K^{\mathrm{T}} + K_{\mathrm{CL}}\hat{R}_{\mathrm{CL}}K_{\mathrm{CL}}^{\mathrm{T}} \\ &\quad - K\hat{R}_{\mathrm{CL}}K_{\mathrm{CL}}^{\mathrm{T}} - K_{\mathrm{CL}}\hat{R}_{\mathrm{CL}}K^{\mathrm{T}}. \end{aligned} \tag{3.74}$$

Hence, $P_{\mathrm{CL}}-P_{k+1} = (A-K_{\mathrm{CL}}C)(P_{\mathrm{CL}}-P_k)(A-K_{\mathrm{CL}}C)^{\mathrm{T}}+(K_{\mathrm{CL}}-K)\hat{R}_{\mathrm{CL}}(K_{\mathrm{CL}}-K)^{\mathrm{T}}$. □

Hence, if $P_0 \leqslant P_{\mathrm{CL}}$, then for all $k \geqslant 0$, $P_k \leqslant P_{\mathrm{CL}}$. Furthermore, note that substituting $K = 0$ in (3.60) yields the open-loop estimator (3.44), and hence Proposition 3.7.1 follows from Proposition 3.8.1 when $K = 0$.

Based on Proposition 3.7.1 and 3.8.1, we combine LKF gain with the steady-state error-covariance-based gain to inject data into all of the states for potentially better performance than that of LKF alone.

3.9 LKF with Complementary Open-Loop Steady-State Error Covariance (LKFCOLC)

At each time step, the local-system error-covariance $P^{\mathrm{f}}_{1,k}$ is propagated by (3.27), (3.28), and (3.29), whereas the open-loop steady-state error covariance is given by

$$P_{\mathrm{OL}} = \begin{bmatrix} P_{\mathrm{OL},11} & P_{\mathrm{OL},12} \\ P_{\mathrm{OL},12} & P_{\mathrm{OL},22} \end{bmatrix}, \tag{3.75}$$

where P_{OL} is the steady-state error covariance that satisfies

$$AP_{\mathrm{OL}}A^{\mathrm{T}} - P_{\mathrm{OL}} + Q = 0. \tag{3.76}$$

Note that P_{OL} is partitioned in 3.75 according to (3.22), (3.23).

Next, we inject data into the forecast state $x^{\mathrm{f}}_{2,k}$ of LKF using the open-loop steady-state covariance. That is, (3.31) is modified as

$$x^{\mathrm{da}}_{2,k} = x^{\mathrm{f}}_{2,k} + K_2(y_k - C_1 x^{\mathrm{f}}_{1,k}), \tag{3.77}$$

where

$$K_2 \triangleq P_{\mathrm{OL},12}C_1^{\mathrm{T}}(C_1 P_{\mathrm{OL},11}C_1^{\mathrm{T}} + R_k)^{-1}. \tag{3.78}$$

Finally, the estimator gain K_k for full-state data injection composed of (3.30), (3.77) is given by

$$K_k \triangleq \begin{bmatrix} K_{1,k} \\ K_2 \end{bmatrix} = \begin{bmatrix} P^{\mathrm{f}}_{1,k}C_1^{\mathrm{T}}(C_1 P^{\mathrm{f}}_{1,k}C_1^{\mathrm{T}} + R_k)^{-1} \\ P_{\mathrm{OL},12}C_1^{\mathrm{T}}(C_1 P_{\mathrm{OL},11}C_1^{\mathrm{T}} + R_k)^{-1} \end{bmatrix}. \tag{3.79}$$

3.10 LKF with Complementary Closed-Loop Steady-State Error Covariance (LKFCCLC)

The LKFCOLC technique may not have good performance when the complementary open-loop steady-state error covariance and optimal error covariance are

significantly different. In this case, we use the complementary closed-loop steady-state covariance when LKF is applied. Hence, let $K_{\mathrm{LKF}} = [K_1^{\mathrm{T}}\ 0]^{\mathrm{T}}$, where K_1 is the steady-state Kalman gain of LKF given by (3.28), and let P_{CL} satisfy

$$\begin{aligned}(A - K_{\mathrm{LKF}}C)P_{\mathrm{CL}}(A - K_{\mathrm{LKF}}C)^{\mathrm{T}} - P_{\mathrm{CL}} \\ + K_{\mathrm{LKF}}RK_{\mathrm{LKF}}^{\mathrm{T}} + Q = 0.\end{aligned} \tag{3.80}$$

Now partition P_{CL} as

$$P_{\mathrm{CL}} = \begin{bmatrix} P_{\mathrm{CL},11} & P_{\mathrm{CL},12} \\ P_{\mathrm{CL},12} & P_{\mathrm{CL},22} \end{bmatrix}, \tag{3.81}$$

We obtain the estimator gain K_k by means of (3.79) replacing $P_{\mathrm{OL},11}$, $P_{\mathrm{OL},12}$ with $P_{\mathrm{CL},11}$, $P_{\mathrm{CL},12}$, respectively.

3.11 Simulation Example

We apply the methods introduced in sections 3.2-3.10 to a compartmental model [56], which involves states whose values are nonnegative quantities. This compartmental model is based on the physics of the processes by which material or energy is exchanged among coupled subsystems. In addition, conservation laws account for the flow of such quantities among subsystems.

A schematic diagram of the compartmental model is shown at Figure 3.1. The total number of cells n is 25 for simulations with one state per cell. We assume that the states of the first five cells are measured. Hence, the size of the localized system n_1 is set to 5. All σ_{ii} 's are set to 0.1 and all σ_{ij} $(i \neq j)$ are set to 0.44.

We simulate two cases. Case 1 involves a single-input disturbance in which the input matrix G is the $n \times 1$ ones matrix. Hence, Figure 3.2(a) shows one dominant Hankel singular value. In Case 2, n mutually independent disturbances are spread

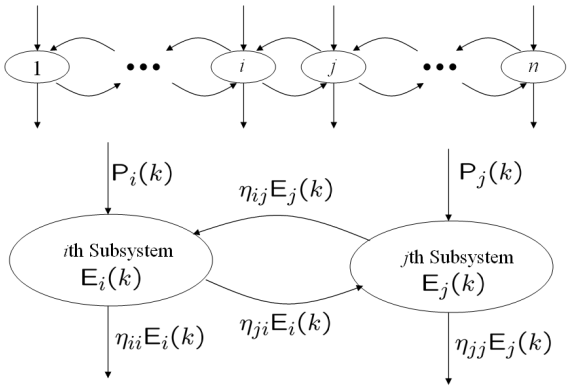


Figure 3.1: Compartmental model involving interconnected subsystems.

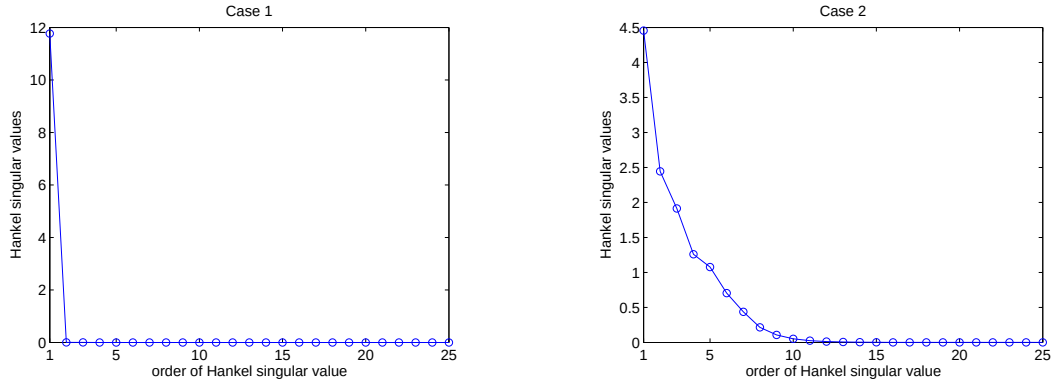
out over all of the cells, and thus the disturbance input matrix G of (3.1) is the $n \times n$ identity matrix. In Case 2, as can be seen in Figure 3.2(b), the Hankel singular values decrease gradually and thus there is no definite model-truncation threshold.

Simulation results for KFEBRM, LKF, LKFBR, LKFRBT, LKFCOLC, and LKFCCLC are shown in Figure 3.3 and 3.4. LKFCCLC shows the best performance in Case 2. In Case 1, KFEBRM and LKFBR show the best performance. Balanced-model-based methods perform well for Case 1 because of the rapidly decreasing Hankel singular values. However, the performance of LKFCOLC and LKFCCLC are comparable to that of the balanced-model-based methods. Moreover, KFCLC converges rapidly to the optimal Kalman filter with higher model order whereas KFOLC does not. We summarize the properties and performance ranks of each method in Table 3.1.

Estimators with an OLSS covariance-based static gain and CLSS covariance-based static gain consistently perform better than without the static gain as shown in Figure 3.5. Moreover, Figure 3.6 shows that LKF compensated by either OLSS or CLSS covariance show improved performance than LKF alone even when an erroneous Q is used to obtain the OLSS and CLSS covariances.

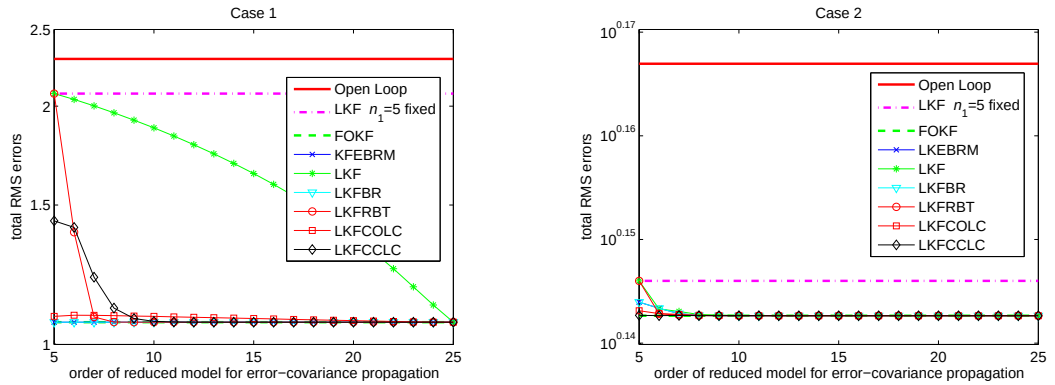
Table 3.1: Comparisons of reduced-order error-covariance Kalman filters.

Properties of Methods Method	number of states for data injection	model order for covariance propagation	requires balancing transform?	requires local measurements?	performance rank for Case 1	performance rank for Case 2
KFEBRM	25	5	yes	no	1	3
LKF	5	5	no	yes	5	5
LKFBR	5	5	yes	yes	1	3
LKFRBT	25	5	yes	yes	5	5
LKFCOLC	25	5	no	yes	3	2
LKFCCLC	25	5	no	yes	4	1



(a) (b)

Figure 3.2: Case 1 Hankel singular values (left) and Case 2 Hankel singular values (right).



(a) (b)

Figure 3.3: Total RMS errors when the order of the reduced model for error-covariance propagation is increased. Note that the number of measurements is fixed at 5. It can be seen that LKFCCLC shows the best performance in Case 2. In Case 1, KFEBRM and LKFBR show the best performance. Balanced-model-based methods perform well for Case 1 because of the rapidly decreasing Hankel singular values. However, note that the performance of LKFCOLC and LKFCCLC are comparable to that of the balanced-model-based methods. Moreover, LKFCCLC converges rapidly to the optimal Kalman filter with higher model order, whereas LKFCOLC has a small initial error but slowly converges to the optimal Kalman filter.

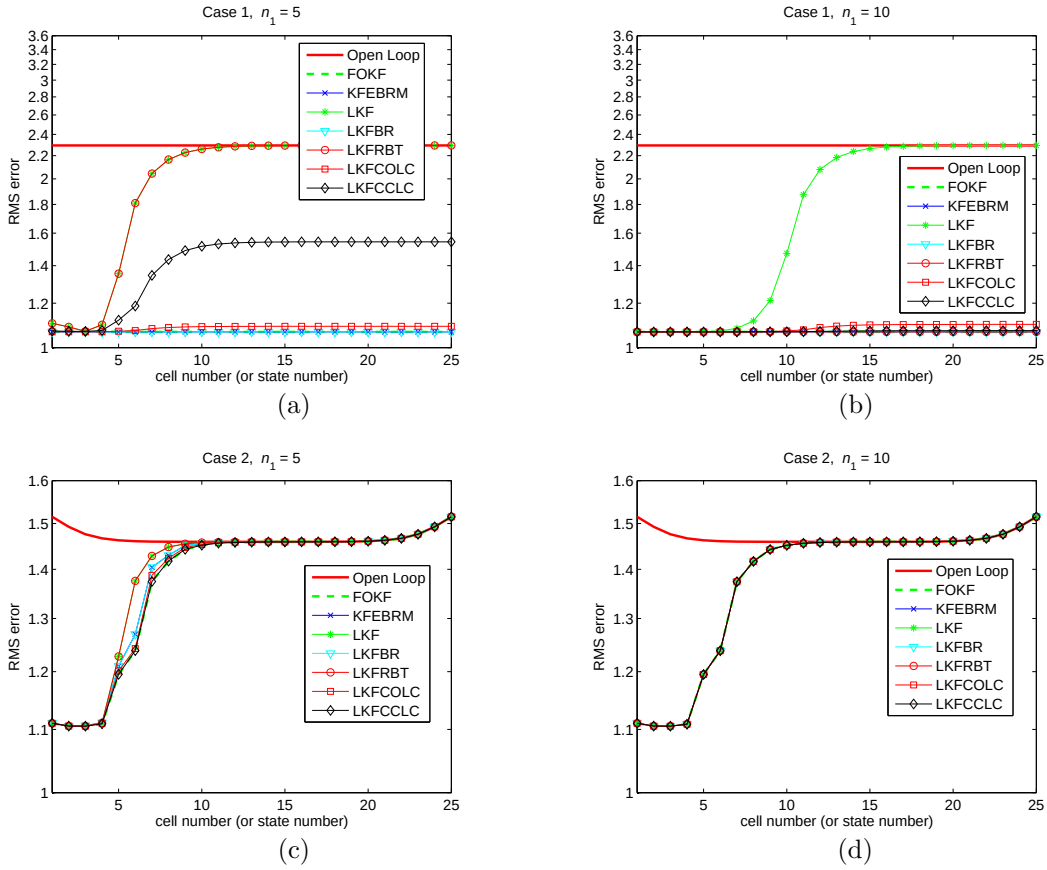


Figure 3.4: RMS error of each cell (spatial distribution of errors) with respect to each method when the order of the reduced model is $n_1 = 5, 10$. (a) and (b) are for Case 1 while (c) and (d) are for Case 2. Note that LKF and LKFRBT are identical when $n_1 = 5$.

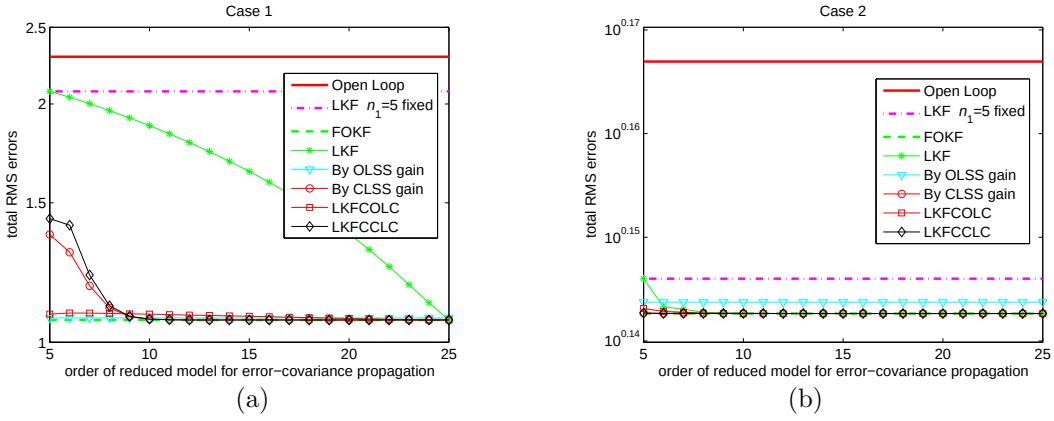
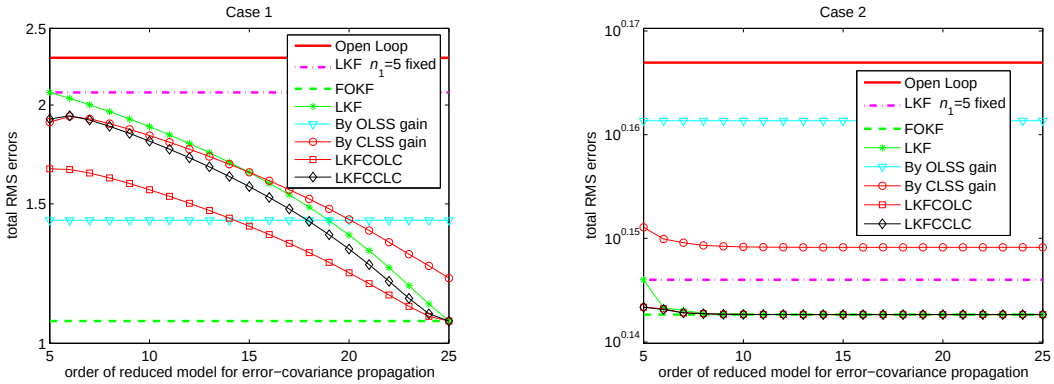


Figure 3.5: Total RMS errors of estimators with open-loop steady-state (OLSS) covariance-based gain, closed-loop steady-state (CLSS) covariance-based gain, LKFCOLC and LKFCCLC when the order of the reduced model for error-covariance propagation is increased. LKF compensated by OLSS and CLSS covariance show significantly improved performance over LKF alone.



(a) (b)

Figure 3.6: Total RMS errors of estimators with open-loop steady-state (OLSS) covariance-based gain, closed-loop steady-state (CLSS) covariance-based gain, LKFCOLC and LKFCCLC when the order of the reduced model for error-covariance propagation is increased and $0.01Q$ is taken as an erroneous disturbance covariance to obtain OLSS and CLSS covariances. LKF compensated by OLSS and CLSS covariance show improved performance over LKF alone although the erroneous disturbance covariance is used in obtaining the OLSS and CLSS covariances.

3.12 Conclusions

We presented several Kalman filters for reduced-order error-covariance propagation and compared them through numerical studies. We conducted numerical studies for two extreme cases of Hankel singular values. In both cases, LKFCOLC and LKFCCLC show good performance. When there are a few dominant Hankel singular values, LKFCCLC can be applied efficiently without the need for a similarity transformation that may be prohibitive in large-scale systems.

CHAPTER IV

Reduced-Order Covariance-Based Unscented Kalman Filtering with Complementary Steady-State Correlation

4.1 Introduction

State estimation for very large scale systems remains an area of interest research. These systems arise in applications based on spatially distributed models or spatially discretized partial differential equations. Weather forecasting and related atmospheric applications are the main driver for this line of research [57, 58]. Although the literature on reduced-order filtering extends back several decades [36, 59], the challenge in addressing very large scale systems is to propagate the covariance efficiently, especially in view of the fact that covariance propagation is $O(n^3)$ in computational complexity, where n is the number of states.

To address the problem of computational complexity, a reduced-order error-covariance propagation algorithm is developed in [60] based on balanced reduction, and this algorithm is compared to several alternative reduced-order error-covariance propagation algorithms in [61]. Some of these algorithms use an initial balancing transformation, while others use an initial model truncation along with a steady-state covariance. Algorithms that avoid the need for a balancing step are desirable when the system order is sufficiently high that balancing and transformation are prohibitive.

In the present paper we extend the approaches considered in [61] to nonlinear

systems by using the unscented Kalman filter [62]. This extension is necessitated by the fact that large-scale applications are also typically nonlinear. Since balancing is usually not feasible for systems of very large order, we consider nonlinear extensions of only the algorithms studied in [61] that avoid the need for balancing. These algorithms include the localized unscented Kalman filter (LUKF), which is essentially an unscented Kalman filter applied to a truncated model that includes all states that affect the measurements, as well as LUKF augmented by complementary steady-state error correlations. This augmentation can be performed either without LUKF present or with LUKF present. The former case is referred to as the localized unscented Kalman filter with complementary open-loop steady-state correlations (LUKFCOLC), while the latter case is referred to as the localized unscented Kalman filter with complementary closed-loop steady-state correlation (LUKFCCLC). The paper describes the LUKF, LUKFCOLC, and LUKFCCLC algorithms in detail.

To compare the performance of the LUKF, LUKFCOLC, and LUKFCCLC algorithms, we consider three examples that are computationally tractable on single-processor machines. First, we consider a finite-volume compressible hydrodynamic simulation for one-dimensional. Next, we consider a two-dimensional finite-volume magnetohydrodynamic (MHD) simulation using the BATSRUS MHD code developed in [63]. Extended Kalman filter and state-dependent Riccati equation techniques were applied to these problems in [30, 64]. Finally, we consider a one-dimensional model of the Global Ionosphere-Thermosphere Model (GITM), which considers the effect of solar flux on the dynamics of the atmosphere. A state-dependent Riccati equation technique for data assimilation was applied to this problem in [31].

4.2 The Unscented Kalman Filter

Consider the discrete-time nonlinear system with dynamics

$$x_{k+1} = f(x_k, u_k, k) + w_k \tag{4.1}$$

and measurements

$$y_k = h(x_k, k) + v_k, \tag{4.2}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, and $y_k \in \mathbb{R}^p$. The input u_k and output y_k are assumed to be measured, and $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^p$ are uncorrelated zero-mean white noise processes with covariances Q_k and R_k, respectively. We assume that R_k is positive definite. If the dynamics and the measurement map are linear, the Kalman filter yields the optimal (minimum variance) estimates of the state x_k. The Kalman filter depends on the error covariance which is propagated using the Riccati equation [65].

In this paper, we consider the unscented Kalman filter (UKF) [62]. Unlike the extended Kalman filter [64] and SDRE estimator [30], UKF does not use the Jacobian of the dynamics or a factorization of the dynamics to propagate a pseudo-error covariance. The starting point for UKF is a set of sample points, that is, a collection of state estimates that capture the initial probability distribution of the state. First, we describe the unscented transformation procedure used to obtain sample points with a specified mean and variance.

Assume that $\overline{x} \in \mathbb{R}^n$, $\overline{P} \in \mathbb{R}^{n \times n}$ is positive semidefinite and $\lambda > 0$. The unscented transformation is used to obtain $2n + 1$ sample points $X_i \in \mathbb{R}^n$ and corresponding weights $\gamma_{\mathrm{x},i}$ and $\gamma_{\mathrm{P},i}$, for $i = 1, \ldots, 2n+1$, so that the weighted mean and the weighted variance of the sample points is $\overline{x}$ and $\overline{P}$, respectively. The unscented transformation

$X = \Psi(\overline{x}, \overline{P}, \lambda)$ of $\overline{x}$ with covariance $\overline{P}$ is defined by

$$X_i = \begin{cases} \overline{x}, & \text{if } i = 0, \\ \overline{x} + \tilde{P}_i, & \text{if } i = 1, \ldots, n, \\ \overline{x} - \tilde{P}_{i-n}, & \text{if } i = n+1, \ldots, 2n, \end{cases} \tag{4.3}$$

where $\tilde{P} \triangleq \left(\lambda \overline{P}\right)^{1/2}$, $\tilde{P}_i$ is the ith column of $\tilde{P} \in \mathbb{R}^{n \times n}$, $X \in \mathbb{R}^{n \times 2n+1}$ has entries

$$X = \begin{bmatrix} X_0 & \cdots & X_{2n} \end{bmatrix} \tag{4.4}$$

and $\lambda > 0$ determines the spread of the sample points around $\overline{x}$. Note that

$$\sum_{i=0}^{2n} \gamma_{\mathrm{x},i} X_i = \overline{x}, \quad \sum_{i=0}^{2n} \gamma_{\mathrm{P},i} (X_i - \overline{x})(X_i - \overline{x})^{\mathrm{T}} = \overline{P}, \tag{4.5}$$

where the weights defined by

$$\gamma_{\mathrm{x},1} \triangleq 1 - \frac{n}{\lambda}, \quad \gamma_{\mathrm{P},1} \triangleq 1 - \frac{n}{\lambda} + (1 - \frac{\lambda}{n} + \beta), \tag{4.6}$$

$$\gamma_{\mathrm{x},i} = \gamma_{\mathrm{P},i} \triangleq \frac{1}{2\lambda}, \quad i = 2, \ldots, 2n+1, \tag{4.7}$$

depend on λ and $\beta > 0$.

UKF uses the unscented transformation to capture the mean and covariance of the error dynamics. UKF involves simulating $2n + 1$ copies of the model and using these ensembles to approximate the mean and error covariance. We assume that an initial estimate x_0^{f} of the state x_0 is given, and the covariance of error in the initial condition is $P_0^{\mathrm{f}} \in \mathbb{R}^{n \times n}$.

For all $k \geqslant 0$, the analysis step of UKF is given by

$$x_k^{\mathrm{da}} = x_k^{\mathrm{f}} + K_k (y_k - y_k^{\mathrm{f}}), \tag{4.8}$$

$$y_k^{\mathrm{f}} = h(x_k^{\mathrm{f}}, k), \tag{4.9}$$

$$X_k^{\mathrm{da}} = \Psi(x_k^{\mathrm{da}}, P_k^{\mathrm{da}}, \lambda), \tag{4.10}$$

$$P_k^{\mathrm{da}} = P_k^{\mathrm{f}} - K_k P_{\mathrm{yy},k} K_k^{\mathrm{T}}, \tag{4.11}$$

where

$$K_k = P_{\mathrm{xy},k} P_{\mathrm{yy},k}^{-1}, \tag{4.12}$$

$$P_{\mathrm{xy},k} = \sum_{i=0}^{2n} \gamma_{\mathrm{P},i} (X_{i,k}^{\mathrm{f}} - x_k^{\mathrm{f}})(Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})^{\mathrm{T}}, \tag{4.13}$$

$$P_{\mathrm{yy},k} = \sum_{i=0}^{2n} \gamma_{\mathrm{P},i} (Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})(Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})^{\mathrm{T}} + R_k, \tag{4.14}$$

$$X_k^{\mathrm{f}} = \Psi(x_k^{\mathrm{f}}, P_k^{\mathrm{f}}, \lambda), \tag{4.15}$$

$$Y_{i,k}^{\mathrm{f}} = h(X_{i,k}^{\mathrm{f}}, k), \tag{4.16}$$

and the forecast step of UKF is given by

$$\tilde{X}_{i,k+1}^{\mathrm{f}} = f(X_{i,k}^{\mathrm{da}}, u_k, k), \tag{4.17}$$

$$x_{k+1}^{\mathrm{f}} = \sum_{i=0}^{2n} \gamma_{\mathrm{x},i} \tilde{X}_{i,k+1}^{\mathrm{f}}, \tag{4.18}$$

$$P_{k+1}^{\mathrm{f}} = \sum_{i=0}^{2n} \gamma_{\mathrm{P},i} (\tilde{X}_{i,k+1}^{\mathrm{f}} - x_{k+1}^{\mathrm{f}})(\tilde{X}_{i,k+1}^{\mathrm{f}} - x_{k+1}^{\mathrm{f}})^{\mathrm{T}} + Q_k. \tag{4.19}$$

If the dynamics in (4.1) and (4.2) are linear, then UKF is equivalent to the Kalman filter [62].Since UKF involves $2n+1$ model updates given by (4.17), the computational burden of UKF is of the order $(2n+1)n^2 = 2n^3 + n^2$. If n is large, for example, in finite volume discretization of partial differential equations, then the computational burden of implementing UKF is enormous.

In many data assimilation applications involving finite volume models, the dynamics involve nearest neighbor interactions (banded dynamics), and hence measurements available in a certain spatial region seem to influence the estimates in only a certain neighborhood around the measurement location. Next, we consider an extension of UKF that approximates the error covariance corresponding to only a specific subspace of the state and not the entire state, thereby reducing the number of ensembles needed.

4.3 Localized Unscented Kalman Filter (LUKF)

Assume that the state $x_k \in \mathbb{R}^n$ has components

$$x_k = \begin{bmatrix} x_{\mathrm{L},k}^{\mathrm{T}} & x_{\mathrm{E},k}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{4.20}$$

where $x_{\mathrm{L},k} \in \mathbb{R}^{n_{\mathrm{L}}}$ and $x_{\mathrm{E},k} \in \mathbb{R}^{n_{\mathrm{E}}}$, and $n_{\mathrm{L}} + n_{\mathrm{E}} = n$. Also, assume that the measurements depend on the state x_{L} so that y_k can be expressed as

$$y_k = h(x_{\mathrm{L},k}, k) + v_k. \tag{4.21}$$

Finally, let Q_k and P_0^{f} have entries

$$Q_k = \begin{bmatrix} Q_{\mathrm{L},k} & Q_{\mathrm{LE},k} \\ Q_{\mathrm{LE},k}^{\mathrm{T}} & Q_{\mathrm{E},k} \end{bmatrix}, P_0^{\mathrm{f}} = \begin{bmatrix} P_{\mathrm{L},0}^{\mathrm{f}} & P_{\mathrm{LE},0}^{\mathrm{f}} \\ (P_{\mathrm{LE},0}^{\mathrm{f}})^{\mathrm{T}} & P_{\mathrm{E},0}^{\mathrm{f}} \end{bmatrix}. \tag{4.22}$$

The objective is to directly inject the measurement data y_k into only the states corresponding to the estimate of $x_{\mathrm{L},k}$ by using a reduced-order surrogate error covariance. For example, in weather prediction models involving spatial dimensions, $x_{\mathrm{L},k}$ may represent the states corresponding to a small region surrounding the location where measurements are available, and $x_{\mathrm{E},k}$ may represent the states that are outside this localized region.

Assume that for all $k \geqslant 0$, the error covariance P_k^{f} of UKF has the structure

$$P_k^{\mathrm{f}} = \begin{bmatrix} P_{\mathrm{L},k}^{\mathrm{f}} & 0 \\ 0 & 0 \end{bmatrix}, \tag{4.23}$$

where $P_{\mathrm{L},k}^{\mathrm{f}} \in \mathbb{R}^{n_{\mathrm{L}} \times n_{\mathrm{L}}}$ represents the covariance of error corresponding to the state $x_{\mathrm{L},k}$. Hence, it follows from (4.3) and (4.23) that if $X_k^{\mathrm{f}} = \Psi(x_k^{\mathrm{f}}, P_k^{\mathrm{f}}, \lambda)$ then for $i = n_{\mathrm{L}} + 1, \ldots, n, n + n_{\mathrm{L}} + 1, \ldots, 2n$,

$$X_{i,k}^{\mathrm{f}} = X_{1,k}^{\mathrm{f}} = x_k^{\mathrm{f}}. \tag{4.24}$$

Since $2n_{\mathrm{E}}+1$ ensembles are exactly the same, it suffices to retain only one such ensemble. Therefore, the number of ensembles required is reduced from $2n+1$ to $(2n+1)-2n_{\mathrm{E}}=2n_{\mathrm{L}}+1$. Furthermore, it follows from (4.23) that instead of a $n\times n$ error covariance only a $n_{\mathrm{L}}\times n_{\mathrm{L}}$ reduced-order error covariance has to be estimated using the $2n_{\mathrm{L}}+1$ ensembles. Applying these simplifying assumptions to UKF yields the localized unscented Kalman filter (LUKF).

The data assimilation step of LUKF is given by

$$x^{\mathrm{da}}_{\mathrm{L},k} = x^{\mathrm{f}}_{\mathrm{L},k} + K_{\mathrm{L},k}(y_k - y^{\mathrm{f}}_k), \tag{4.25}$$

$$x^{\mathrm{da}}_{\mathrm{E},k} = x^{\mathrm{f}}_{\mathrm{E},k} \tag{4.26}$$

$$y^{\mathrm{f}}_k = h(x^{\mathrm{f}}_{\mathrm{L},k}, k), \tag{4.27}$$

$$X^{\mathrm{da}}_{\mathrm{L},k} = \Psi(x^{\mathrm{da}}_{\mathrm{L},k}, P^{\mathrm{da}}_{\mathrm{L},k}, \lambda), \tag{4.28}$$

$$P^{\mathrm{da}}_{\mathrm{L},k} = P^{\mathrm{f}}_{\mathrm{L},k} - K_{\mathrm{L},k}P_{\mathrm{yy},k}K^{\mathrm{T}}_{\mathrm{L},k}, \tag{4.29}$$

where

$$K_{\mathrm{L},k} = P_{\mathrm{x_L y},k}P^{-1}_{\mathrm{yy},k}, \tag{4.30}$$

$$P_{\mathrm{x_L y},k} = \sum_{i=0}^{2n_{\mathrm{L}}} \gamma_{\mathrm{P},i}(X^{\mathrm{f}}_{\mathrm{L},i,k} - x^{\mathrm{f}}_{\mathrm{L},k})(Y^{\mathrm{f}}_{i,k} - y^{\mathrm{f}}_k)^{\mathrm{T}}, \tag{4.31}$$

$$P_{\mathrm{yy},k} = \sum_{i=0}^{2n_{\mathrm{L}}} \gamma_{\mathrm{P},i}(Y^{\mathrm{f}}_{i,k} - y^{\mathrm{f}}_k)(Y^{\mathrm{f}}_{i,k} - y^{\mathrm{f}}_k)^{\mathrm{T}} + R_k, \tag{4.32}$$

$$X^{\mathrm{f}}_{\mathrm{L},k} = \Psi(x^{\mathrm{f}}_{\mathrm{L},k}, P^{\mathrm{f}}_{\mathrm{L},k}, \lambda), \tag{4.33}$$

$$Y^{\mathrm{f}}_{i,k} = h(X^{\mathrm{f}}_{\mathrm{L},i,k}, k), \tag{4.34}$$

and for $i=0,\ldots,2n_{\mathrm{L}}$, $X^{\mathrm{f}}_{\mathrm{L},i,k}\in\mathbb{R}^{n_{\mathrm{L}}}$ is the $(i+1)$th column of $X^{\mathrm{f}}_{\mathrm{L},k}$. Note that only $2n_{\mathrm{L}}+1$ ensembles are used compared to the $2n+1$ ensembles in the UKF, and (4.25)-(4.26) imply that the measurement data is injected directly into only the estimates of the state corresponding to the subspace $x_{\mathrm{L},k}$.

Next, for all $i = 0, \ldots, 2n_{\mathrm{L}}$, define $X^{\mathrm{da}}_{i,k} \in \mathbb{R}^n$ by

$$X^{\mathrm{da}}_{i,k} \triangleq \begin{bmatrix} X^{\mathrm{da}}_{\mathrm{L},i,k} \\ x^{\mathrm{da}}_{\mathrm{E},k} \end{bmatrix}, \tag{4.35}$$

where $X^{\mathrm{da}}_{\mathrm{L},i,k} \in \mathbb{R}^{n_{\mathrm{L}}}$ is the $(i+1)$th column of $X^{\mathrm{da}}_{\mathrm{L},k}$. It follows from (4.23) that the correlations corresponding to the error in the state $x_{\mathrm{E},k}$ are assumed to be zero, and therefore, the estimate $x^{\mathrm{da}}_{\mathrm{E},k}$ of the state $x_{\mathrm{E},k}$ in all the ensembles of LUKF in (4.35) is the same. However, the estimate of the state $x_{\mathrm{L},k}$ is different in each ensemble. The forecast step of LUKF is given by

$$\tilde{X}^{\mathrm{f}}_{i,k+1} = f(X^{\mathrm{da}}_{i,k}, u_k, k). \tag{4.36}$$

The forecast estimate of the state x_k is obtained by

$$x^{\mathrm{f}}_{k+1} = \sum_{i=0}^{2n_{\mathrm{L}}} \gamma_{\mathrm{x},i} \tilde{X}^{\mathrm{f}}_{i,k+1}. \tag{4.37}$$

Next, for $i = 0, \ldots, 2n_{\mathrm{L}}$, let $X^{\mathrm{f}}_{i,k+1} \in \mathbb{R}^n$ have entries

$$\tilde{X}^{\mathrm{f}}_{i,k+1} = \begin{bmatrix} \tilde{X}^{\mathrm{f}}_{\mathrm{L},i,k+1} \\ \tilde{X}^{\mathrm{f}}_{\mathrm{E},i,k+1} \end{bmatrix} \tag{4.38}$$

with $\tilde{X}^{\mathrm{f}}_{\mathrm{L},i,k+1} \in \mathbb{R}^{n_{\mathrm{L}}}$ and $\tilde{X}^{\mathrm{f}}_{\mathrm{E},i,k+1} \in \mathbb{R}^{n_{\mathrm{E}}}$. Finally, to account for the increase in the error covariance due to the process noise, represented by $Q_{\mathrm{L},k}$, the surrogate covariance of the error in the estimate of $x_{\mathrm{L},k}$ is given by

$$P^{\mathrm{f}}_{\mathrm{L},k+1} = \sum_{i=0}^{2n} \gamma_{\mathrm{P},i} (\tilde{X}^{\mathrm{f}}_{\mathrm{L},i,k+1} - x^{\mathrm{f}}_{\mathrm{L},k+1})(\tilde{X}^{\mathrm{f}}_{\mathrm{L},i,k+1} - x^{\mathrm{f}}_{\mathrm{L},k+1})^{\mathrm{T}} + Q_{\mathrm{L},k}. \tag{4.39}$$

Although (4.26) implies that data is not directly injected into the state estimates corresponding to the subspace $x_{\mathrm{E},k}$, it follows from (4.35)-(4.37) that the measurement data affect the estimates of the state $x_{\mathrm{E},k}$ through the dynamic coupling between $x_{\mathrm{L},k}$ and $x_{\mathrm{E},k}$. LUKF involves $2n_{\mathrm{L}} + 1$ model updates and therefore the number of computations involved is of the order $(2n_{\mathrm{L}} + 1)n$. Hence, if $n_{\mathrm{L}} \ll n$, then LUKF is computationally efficient compared to UKF.

4.4 Complementary Steady-State Correlation

Although LUKF provides estimates of all of the states, (4.26) implies that LUKF injects data directly into only that states corresponding to the estimate of $x_{\mathrm{L},k}$. On the other hand, UKF injects data directly into the all of states of the estimator. Since ignoring the correlation between the error in the estimates of the states $x_{\mathrm{L},k}$ and $x_{\mathrm{E},k}$ in LUKF may result in poor estimates, we consider a modification of LUKF that uses a constant correlation between the error in the estimates of the states $x_{\mathrm{L},k}$ and $x_{\mathrm{E},k}$. In the following sections, we assume that $Q_k = Q$ and $R_k = R$ for all $k \geqslant 0$.

If the dynamics and the measurement map in (4.1) and (4.2) are linear and time-invariant, then, the error covariance is propagated using the Riccati equation, and under certain detectability and stabilizability assumptions, the error covariance converges to a steady-state value that is the solution of an algebraic Riccati equation. If the dynamics are nonlinear, then there is no guarantee that UKF or LUKF will reach a statistical steady-state. However, simulations may indicate that after a certain period of time, the performance of the estimators do not vary significantly, and in that case, we assume that the estimator has almost reached statistical steady-state.

4.4.1 LUKF with Complementary Open-Loop Correlation (LUKFCOLC)

First, we determine a static estimator gain that is based on the steady-state correlation between the measurements y_k and the state x_k. If the dynamics are linear and time-invariant, that is $f(x, u, k) = Ax + Bu$ and $h(x, k) = Cx$ for all $k \geqslant 0$, and (A, Q) is stabilizable, then the steady-state state covariance P_{xx} is the solution of the Lyapunov equation

$$P_{\mathrm{xx}} = AP_{\mathrm{xx}}A^{\mathrm{T}} + Q. \tag{4.40}$$

Furthermore, the steady state correlation $P_{\rm xy}$ between the measurement y_k and the state x_k is given by $P_{\rm xy} = P_{\rm xx}C^{\rm T}$.

However, since the dynamics are nonlinear, we approximate the steady-state state covariance by using Monte Carlo simulations. Consider N copies of the open-loop model of the system (4.1)-(4.2) so that for $i = 1, \ldots, N$,

$$\begin{aligned} \tilde{x}_{i,k+1} &= f(\tilde{x}_{i,k}, u_k, k) + \tilde{w}_{i,k}, \\ \tilde{y}_{i,k} &= h(\tilde{x}_{i,k}, k) + \tilde{v}_{i,k}, \end{aligned} \tag{4.41}$$

where $\tilde{x}_{i,0}$ is a random variable with the specified mean x_0 and variance $P_0^{\rm f}$, and $\tilde{w}_{i,k}$ and $\tilde{v}_{i,k}$ are sampled from zero-mean white processes with variances Q and R, respectively. Next, we define an approximation of the steady state open-loop correlation $P_{\rm OL,xy}$ and $P_{\rm OL,yy}$ by

$$P_{\rm OL,xy} \triangleq \lim_{k\to\infty} \frac{1}{N-1} \sum_{i=1}^{N} (\tilde{x}_{i,k} - \overline{x}_k)(\tilde{y}_{i,k} - \overline{y}_k)^{\rm T}, \tag{4.42}$$

$$P_{\rm OL,yy} \triangleq \lim_{k\to\infty} \frac{1}{N-1} \sum_{i=1}^{N} (\tilde{y}_{i,k} - \overline{y}_k)(\tilde{y}_{i,k} - \overline{y}_k)^{\rm T}, \tag{4.43}$$

where

$$\overline{x}_k \triangleq \frac{1}{N} \sum_{i=1}^{N} \tilde{x}_{i,k}, \quad \overline{y}_k \triangleq \frac{1}{N} \sum_{i=1}^{N} \tilde{y}_{i,k}. \tag{4.44}$$

Alternatively, the unscented transformation can also be used to approximate the steady state open-loop state covariance. Note that the state covariance of (4.1) is the same as the open-loop error covariance, that is the covariance of error of an estimator when the estimator gain is zero. Hence, we use (4.8)-(4.19) with $K_k = 0$ for all $k \geqslant 0$, and define $P_{\rm OL,xy}$ and $P_{\rm OL,yy}$ by

$$P_{\rm OL,xx} \triangleq \lim_{k\to\infty} P_{\rm xy,k}, \quad P_{\rm OL,yy} \triangleq \lim_{k\to\infty} P_{\rm yy,k}. \tag{4.45}$$

If n is small, then the computational burden of using the open-loop unscented Kalman filter to estimate the open-loop error correlation is small. However, when n is large,

approximating the error covariance by using Monte Carlo simulations with a small N is computationally more efficient.

Finally, we define the static estimator gain $K_{\mathrm{OL}} \in \mathbb{R}^{n\times p}$ based on the steady-state open-loop correlations by

$$K_{\mathrm{OL}} \triangleq P_{\mathrm{OL,xy}} P_{\mathrm{OL,yy}}^{-1}. \tag{4.46}$$

and let K_{OL} have entries

$$K_{\mathrm{OL}} = \begin{bmatrix} K_{\mathrm{OL,L}} \\ K_{\mathrm{OL,E}} \end{bmatrix}, \tag{4.47}$$

where $K_{\mathrm{OL,L}} \in \mathbb{R}^{n_{\mathrm{L}}\times p}$ and $K_{\mathrm{OL,E}} \in \mathbb{R}^{n_{\mathrm{E}}\times p}$. The forecast step of LUKFCOLC is given by (4.35) - (4.39). The analysis step of the LUKFCOLC is given by

$$x_{\mathrm{L},k}^{\mathrm{da}} = x_{\mathrm{L},k}^{\mathrm{f}} + K_{\mathrm{L},k}(y_k - y_k^{\mathrm{f}}), \tag{4.48}$$

$$x_{\mathrm{E},k}^{\mathrm{da}} = x_{\mathrm{E},k}^{\mathrm{f}} + K_{\mathrm{OL,E}}(y_k - y_k^{\mathrm{f}}), \tag{4.49}$$

$$y_k^{\mathrm{f}} = h(x_{\mathrm{L},k}^{\mathrm{f}}, k), \tag{4.50}$$

$$X_{\mathrm{L},k}^{\mathrm{da}} = \Psi(x_{\mathrm{L},k}^{\mathrm{da}}, P_{\mathrm{L},k}^{\mathrm{da}}, \lambda), \tag{4.51}$$

$$P_{\mathrm{L},k}^{\mathrm{da}} = P_{\mathrm{L},k}^{\mathrm{f}} - K_{\mathrm{L},k} P_{\mathrm{yy},k} K_{\mathrm{L},k}^{\mathrm{T}}, \tag{4.52}$$

where $K_{\mathrm{L},k}$ and $P_{\mathrm{yy},k}$ are defined in (4.30)-(4.34).

Note that injecting measurement data y_k in an estimator affects the error covariances and hence, the actual closed-loop error correlation between y_k and the error in estimates $x_k^{\mathrm{f}} - x_k$ will be different from the open-loop error correlation $P_{\mathrm{OL,xy}}$ with no data injection. However, (4.49) implies that the estimator gain corresponding to the estimate $x_{\mathrm{E},k}^{\mathrm{da}}$ is based on only the open-loop error correlation and is not aware of the change in correlation due to data injection. On the other hand, UKF always updates the closed-loop error covariances, thus accounting for the change in the correlation due to data injection.

4.4.2 LUKF with Complementary Closed-Loop Correlation (LUKFCCLC)

Next, instead of using a static estimator gain that is based on the open-loop steady-state correlations, we use a static estimator gain that is based on the closed-loop steady-state correlations. Specifically, we estimate the steady-state correlations between the error in the estimates when LUKF is used for state estimation. We assume that LUKF has reached a statistical steady-state when the performance of LUKF does not change significantly.

The Monte-Carlo procedure to determine the steady-state closed-loop correlation is as follows. First, we simulate N copies of the open-loop model of the system as shown in (4.41) and obtain outputs $\tilde{y}_{i,k}$. Next, for $i = 1, \ldots, N$, we perform state estimation using LUKF with the outputs $\tilde{y}_{i,k}$. Let $\tilde{x}^{\mathrm{f}}_{i,k}$ be the estimate of $\tilde{x}_{i,k}$ provided by the ith simulation of LUKF. We approximate the steady-state closed-loop correlations by

$$P_{\mathrm{CL,x\overline{y}}} \triangleq \lim_{k\to\infty} \frac{1}{N-1}\sum_{i=1}^{N}\big[\tilde{x}_{i,k} - \tilde{x}^{\mathrm{f}}_{i,k}\big]\big[\tilde{y}_{i,k} - h(\tilde{x}^{\mathrm{f}}_{i,k})\big]^{\mathrm{T}}, \tag{4.53}$$

$$P_{\mathrm{CL,y\overline{y}}} \triangleq \lim_{k\to\infty} \frac{1}{N-1}\sum_{i=1}^{N}\big[\tilde{y}_{i,k} - h(\tilde{x}^{\mathrm{f}}_{i,k})\big]\big[\tilde{y}_{i,k} - h(\tilde{x}^{\mathrm{f}}_{i,k})\big]^{\mathrm{T}}. \tag{4.54}$$

Note that $\tilde{x}_{i,k}$ and $\tilde{x}^{\mathrm{f}}_{i,k}$ are all simulation outputs and hence $P_{\mathrm{CL,xy}}$ and $P_{\mathrm{CL,yy}}$ in (4.53) and (4.54), respectively, can be evaluated.

Alternatively, the unscented transformation can also be used to obtain an estimate of the closed-loop error correlations. To do this, we first use LUKF with the simulated measurement data $\tilde{y}_{1,k}$ to obtain estimates $\tilde{x}^{\mathrm{f}}_{1,k}$ of the state $\tilde{x}_{1,k}$ for $k \geqslant 0$. Assuming $K_{\mathrm{L,k}}$ does not vary significantly after a sufficiently long time interval, we define the steady-state LUKF estimator gain K_{L} by

$$K_{\mathrm{L}} \triangleq \lim_{k\to\infty} K_{\mathrm{L},k}, \tag{4.55}$$

where $K_{\mathrm{L},k}$ is the estimator gain given by (4.30) when obtaining the estimate $\tilde{x}^{\mathrm{f}}_{1,k}$. Note that LUKF ignores correlations between certain states and hence cannot be used to estimate the closed-loop error correlation. Instead, we use the unscented transformation to estimate the closed-loop steady-state error correlations. Specifically, we use (4.8)-(4.19) with

$$K_k = \begin{bmatrix} K_{\mathrm{L}} \\ 0 \end{bmatrix}, \tag{4.56}$$

for all $k \geqslant 0$, and view the correlations $P_{\mathrm{xy},k}$ and $P_{\mathrm{yy},k}$ in (4.13) and (4.14) as an estimate of the closed-loop error correlations of LUKF. We then estimate the closed-loop steady-state error correlations $P_{\mathrm{CL,xy}}$ and $P_{\mathrm{CL,yy}}$ by

$$P_{\mathrm{CL,xy}} = \lim_{k\to\infty} P_{\mathrm{xy},k}, \quad P_{\mathrm{CL,yy}} = \lim_{k\to\infty} P_{\mathrm{yy},k}. \tag{4.57}$$

Finally, the static estimator gain that is based on the steady-state closed-loop error correlations is given by

$$K_{\mathrm{CL}} = P_{\mathrm{CL,xy}} P^{-1}_{\mathrm{CL,yy}} \tag{4.58}$$

with entries

$$K_{\mathrm{CL}} = \begin{bmatrix} K_{\mathrm{CL,L}} \\ K_{\mathrm{CL,E}} \end{bmatrix}, \tag{4.59}$$

where $K_{\mathrm{CL,L}} \in \mathbb{R}^{n_{\mathrm{L}} \times p}$ and $K_{\mathrm{CL,E}} \in \mathbb{R}^{n_{\mathrm{E}} \times p}$.

The forecast step of LUKFCCLC is given by (4.35) - (4.39), and the analysis step of LUKFCCLC is given by (4.48)-(4.52) with $K_{\mathrm{OL,E}}$ replaced by $K_{\mathrm{CL,E}}$ in (4.49).

Next, we compare the performance of UKF, LUKF, LUKFCOLC, and LUKFCCLC on the one dimensional global ionosphere-thermosphere model.

4.5 One dimensional Global Ionosphere-Thermosphere Model

The Global Ionosphere-Thermosphere Model (GITM) is a 3-dimensional, parallel, spherical code that models the Earth's thermosphere and ionosphere system, which has an altitude range from about 100 km to 1000 km, using a stretched altitude grid [10]. This framework allows the model to have nonhydrostatic solutions, resolving sound and gravity waves in both the vertical and horizontal directions. Inputs to GITM include solar ultraviolet (UV) photons, auroral energetic particles, electric fields, and electric currents. The code explicitly solves for the neutral-particle densities of O, O_2, N(^{2}D), N(^{2}P), N(^{4}S), N_2, NO, H, and He (rather than mass-mixing ratios); ion species O^+(^{4}S), O^+(^{2}D), O^+(^{2}P), O_2^+ , N^+, N_2^+ , NO^+, H^+, and He^+; and neutral-particle, ion, and electron temperatures. The bulk horizontal neutral-particle winds are solved for. In the vertical direction, individual winds are solved for with a bulk vertical wind given by the mass-density-weighted average of the individual vertical velocities. The ion velocity is determined by balancing the drag between ion and neutral particles, gravity, pressure gradients, and external electric fields. The electron velocity is considered to be the $E \times B$ drift. At the core, GITM separates vertical and horizontal advection to faciliate numerical solution of the exponential fall-off of the atmosphere. Many of these details are discussed in the classic reference [66].

Most parameters and physical effects can be turned on or off in GITM. We can use this capability to simplify GITM when necessary or to experiment with the physics that constitute GITM. For example, solar EUV heating, Joule heating, and auroral heating can be turned on or off to study thermal sources. Likewise, ion drag, viscosity, Coriolis force, gravity, electromagnetic force, ion pressure, and neutral drag can be

switched on or off to examine their effect on neutral and ion motion. In addition, ion chemistry, ion advection and neutral chemistry can be switched on or off to study the effect of specific chemical reactions. In particular, individual neutral and ion species can be turned on or off to determine the effects of less-dominant species.

GITM can run in 1D as well as 3D. One-dimensional simulations can be carried out efficiently on a personal computer. This ability allows us to quickly and inexpensively investigate long time periods to determine the stability of the code, and to test new physics or data assimilation algorithms. When run in 1D, horizontal transport is ignored, and only the vertical advection is taken into account. Since source terms are included, processes such as ion drag are taken into account. Other external driving terms, such as the electric potential, are specified on a 1°-by-1° grid surrounding the 1D domain. An electric field can then be derived at the cell center, driving the horizontal ion velocities. The horizontal ion flows cause horizontal neutral flows to develop and Joule heating to occur. The data assimilation algorithms verified for 1D GITM will be extended to 3D GITM, which is significantly more computationally intensive.

We apply LUKF compensated by complementary open-loop and closed-loop steady-state covariance to data assimilation for vertical 1D GITM with 50 cells defined along the vertical direction, dominant neutrals O, O_2, N_2, $N(^4S)$, and dominant ions $O^+(^4S)$, O_2^+ , NO^+. We consider only the solar irradiation represented by $F_{10.7}$ as an input to GITM. We summarize the features of 1D GITM for data assimilation in Table 4.1. In addition, the 1D GITM cell structure for UKF, LUKF, and LUKF with open-loop or closed-loop steady-state covariance compensation is shown in Figure 4.1. Table 4.2 summarizes the conditions for the truth model from which the measurement data are taken as well as the data-assimilation model.

Table 4.1: Summary of one-dimensional GITM for data assimilation.

items	description
location	Millstone Hill, MA USA
grid	50 grid cells stretched in vertical direction from 100 km to 750 km altitude
input	daily $F_{10.7}$
states	natural logarithm of number density of O, O_2, N_2, N(^{4}S) vertical velocities of O, O_2, N_2, N(^{4}S) eastward horizontal velocity of neutrals northward horizontal velocity of neutrals normalized temperature of neutrals T_n natural logarithm of number density of O^+(^{4}S), O_2^+ , NO^+ electron temperature T_e 15 states per cell and $15 \times 50 = 750$ states in the model
outputs	electron temperature T_e number density of electron N_e vertical velocity of ions V_i ions temperature T_i in measurement cells 10, 20, 30, and 40

Table 4.2: Conditions for the truth model and assimilation model. Here, MSIS is an empirical model that relates the neutral-particle densities and temperature to the integrated solar flux approximation $F_{10.7}$ and activity level A_p. IRI models the ionospheric density and temperatures for all latitudes and local times. $F_{10.7}$ is the flux at 2800 MHz or 10.7-cm wavelength over the entire solar disk. $F_{10.7}$ is a measure of solar irradiation strength and one of the inputs to GITM.

model conditions	the truth model	the model to be assimilated
$F_{10.7}$	daily random $F_{10.7}$ with average 210	$F_{10.7} = 210$ constant
initial condition	determined by MSIS, IRI with $F_{10.7} = 210$	determined by MSIS, IRI with $F_{10.7} = 100$

The open-loop and closed-loop steady-state covariance of the data assimilation model are obtained by using the unscented transformation (UT). That is, open-loop and closed-loop steady-state covariances are obtained from $2n + 1$ (where n is the total number of states in 1D GITM) ensembles of the 1D model while the model is propagated open-loop and closed-loop, respectively. In particular, the closed-loop steady-state covariance is obtained in two steps as follows:

1. Run LUKF GITM and store local estimator gains over the entire time interval.

2. Run GITM with the unscented transformation while local states are updated

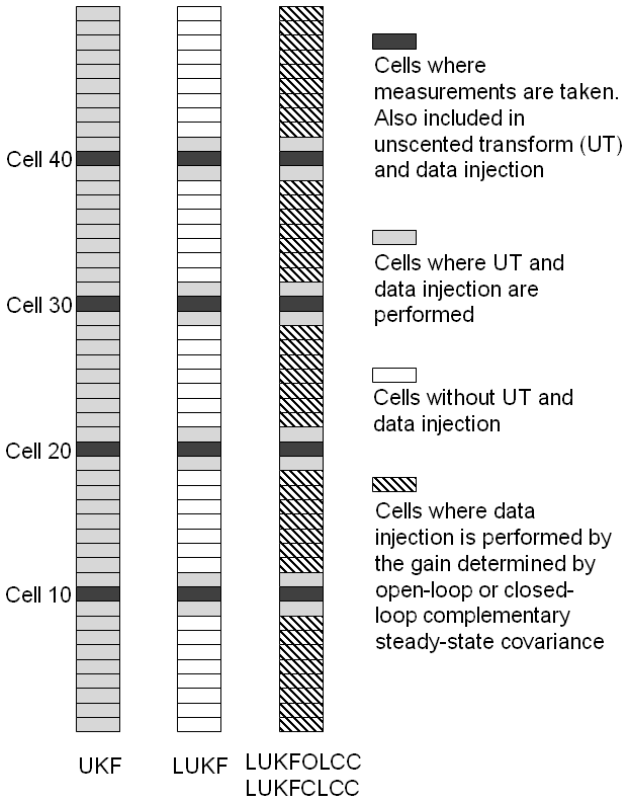


Figure 4.1: 1D GITM cell structure for UKF, LUKF, and LUKFOLCC or LUKFCLCC. In all 3 cases, measurements of number density of electrons, electron temperature, ion-particle temperature, and ion velocity are taken in cells 10, 20, 30, and 40. UKF performs the unscented transformation (UT) and data injection on all 50 cells whereas LUKF performs UT and data injection only on the local groups comprised of 12 cells that include measurement cells. For LUKFOLCC and LUKFCLCC, LUKF is implemented along with an update of the remaining cells using the constant gain determined from an estimate of the open-loop and closed-loop complementary steady-state covariance, respectively.

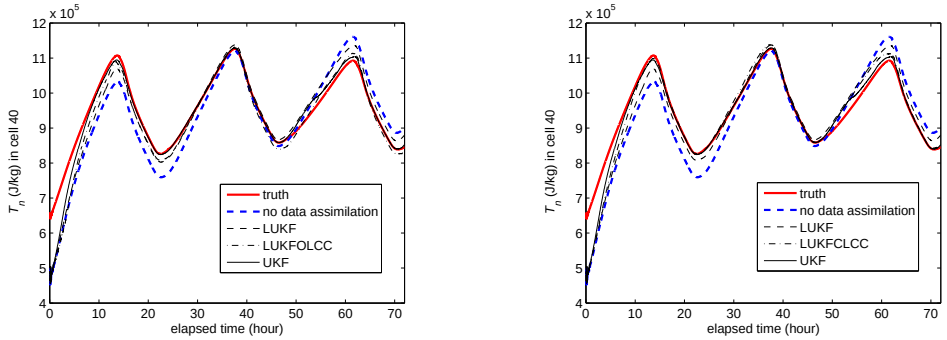


Figure 4.2: Comparisons of the normalized neutral-particle temperature in cell 40. Left plot is for the LUKF with open-loop steady-state covariance compensation (LUKFOLCC), while the right plot is for the LUKF with closed-loop steady-state covariance compensation (LUKFCLCC). The LUKFCLCC is closer to the truth model than the LUKFOLCC.

using the stored local gain and obtain the resulting steady-state covariance.

4.5.1 Comparison of Performance and Computation Time for UKF, LUKF, LUKFOLCC, and LUKFCLCC

We compare the performance and computation time of UKF, LUKF, LUKFOLCC, and LUKFCLCC for 1D GITM. We choose the normalized neutral-particle temperature T_n for performance evaluation. In particular, for a given cell, $T_n \triangleq p/\rho$, where p is the total neutral pressure, $\rho = \sum_s M_s N_s$ is the neutral-particle mass density, M_s is the molecular mass of neutral species s, and N_s is the number density of the neutral species s.

Figure 4.2 compares normalized neutral-particle temperature profiles in cell 40. It can be seen that LUKF with either open-loop or closed-loop steady-state covariance compensation performs better than LUKF alone. Furthermore, LUKFCLCC is closer to the truth model than LUKFOLCC. The errors are more directly compared in Figure 4.3. Computation times (excluding offline covariance estimation) of LUKFOLCC and LUKFCLCC are close to that of LUKF.

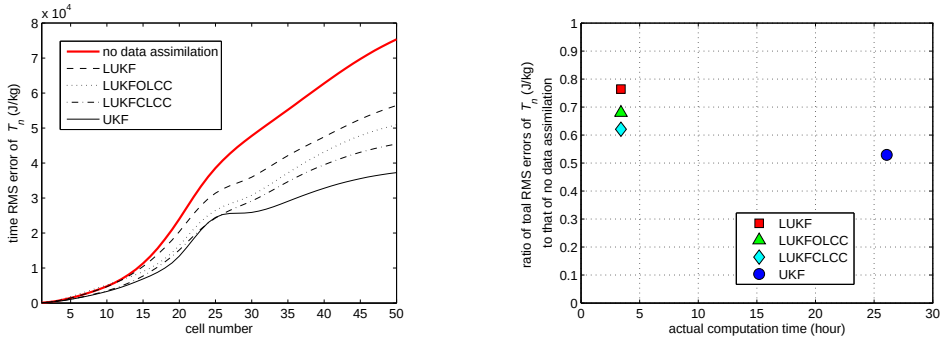


Figure 4.3: RMS errors for the normalized neutral-particle temperatures (a) and ratio of total RMS errors for data assimilation to errors without data assimilation versus computation time plot (b). The right plot is for normalized neutral-particle temperatures. The error for LUKFCLCC is less than for LUKFOLCC. Moreover, note that the LUKFOLCC and LUKFCLCC show better performance than LUKF while computation times are close to that of LUKF.

4.6 Conclusion

We present extensions of the the unscented Kalman filter that propagate a reduced-order pseudo error covariance. To compensate for the neglected correlation between certain states and the measurement, we present two methods that use a complementary static estimator gain based on correlations between the measurements and the neglected states. The use of a static estimator gain based on the open-loop and closed-loop correlations helps improve estimation performance without a significant increase in the online computational burden.

CHAPTER V

Cholesky-Based Reduced-Rank Square-Root Kalman Filtering

5.1 Introduction

The problem of state estimation for large-scale systems has gained increasing attention due to computationally intensive applications such as weather forecasting [34], where state estimation is commonly referred to as data assimilation. For these problems, there is a need for algorithms that are computationally tractable despite the enormous dimension of the state. These problems also typically entail nonlinear dynamics and model uncertainty [67], although these issues are outside the scope of this paper.

One approach to obtaining more tractable algorithms is to consider reduced-order Kalman filters. These reduced-complexity filters provide state estimates that are sub-optimal relative to the classical Kalman filter [35–37, 43, 68]. Alternative reduced-order variants of the classical Kalman filter have been developed for computationally demanding applications [38–40, 69], where the classical Kalman filter gain and covariance are modified so as to reduce the computational requirements. A comparison of several techniques is given in [41].

A widely studied technique for reducing the computational requirements of the Kalman filter for large scale systems is the *reduced-rank filter* [70–73]. In this method,

the error-covariance matrix is factored to obtain a square root, whose rank is then reduced through truncation. This factorization-and-truncation method has direct application to the problem of generating a reduced ensemble for use in particle filter methods [74, 75].

Reduced-rank filters are closely related to the classical factorization techniques [76, 77], which provide numerical stability and computational efficiency, as well as a starting point for reduced-rank approximation.

The primary technique for truncating the error-covariance matrix is the singular value decomposition (SVD) [70–75], wherein the singular values provide guidance as to which components of the error covariance are most relevant to the accuracy of the state estimates. Approximation based on the SVD is largely motivated by the fact that error-covariance truncation is optimal with respect to approximation in unitarily invariant norms, such as the Frobenius norm. Despite this theoretical grounding, there appear to be no theoretical criteria to support the optimality of approximation based on the SVD within the context of recursive state estimation. The difficult is due to the fact that optimal approximation depends on the dynamics and measurement maps in addition to the components of the error covariance.

In the present paper we begin by observing that the Kalman filter update depends on the combination of terms $C_k P_k$, where C_k is the measurement map and P_k is the error covariance. This observation suggests that approximation of $C_k P_k$ may be more suitable than approximation based on P_k alone.

To develop this idea, we show that approximation of $C_k P_k$ leads directly to truncation based on the Cholesky decomposition. Unlike the SVD, however, the Cholesky decomposition does not possess a natural measure of magnitude that is analogous to the singular values arising in the SVD. Nevertheless, filter reduction based on the

Cholesky decomposition provides state-estimation accuracy that is competitive with, and in many cases superior to, that of the SVD. In particular, we show that, in special cases, the accuracy of the Cholesky-decomposition-based reduced-rank filter is equal to the accuracy of the full-rank filter, and we demonstrate examples for which the Cholesky-decomposition-based reduced-rank filter provides acceptable accuracy, whereas the SVD-based reduced-rank filter provides arbitrarily poor accuracy.

A fortuitous advantage of using the Cholesky decomposition in place of the SVD is the fact that the Cholesky decomposition is computationally less expensive than the SVD, specifically, $O(n^3/6)$ [78], and thus an asymptotic computational advantage over SVD by a factor of 12. An additional advantage is that the entire matrix need not be factored; instead, by arranging the states so that those states that contribute directly to the measurement correspond to the initial columns of the lower triangular square root, then only the leading submatrix of the error covariance must be factored, yielding yet further savings over the SVD. Once the factorization is performed, the algorithm effectively retains only the initial "tall" columns of the full Cholesky factorization and truncates the "short" columns.

5.2 The Kalman filter

Consider the time-varying discrete-time system

$$x_{k+1} = A_k x_k + G_k w_k, \tag{5.1}$$

$$y_k = C_k x_k + H_k v_k, \tag{5.2}$$

where $x_k \in \mathbb{R}^{n_k}$, $w_k \in \mathbb{R}^{d_w}$, $y_k \in \mathbb{R}^{p_k}$, $v_k \in \mathbb{R}^{d_v}$, and A_k, G_k, C_k, and H_k are known real matrices of appropriate sizes. We assume that w_k and v_k are zero-mean white processes with unit covariances. Define $Q_k \triangleq G_k G_k^{\mathrm{T}}$ and $R_k \triangleq H_k H_k^{\mathrm{T}}$, and assume that R_k is positive definite for all $k \geqslant 0$. Furthermore, we assume that w_k and v_k

are uncorrelated for all $k \geqslant 0$. The objective is to obtain an estimate of the state x_k using the measurements y_k.

The Kalman filter provides the optimal minimum-variance estimate of the state x_k. The Kalman filter can be expressed in two steps, namely, the *data assimilation step*, where the measurements are used to update the states, and the *forecast step*, which uses the model. These steps can be summarized as follows:

Data Assimilation Step

$$K_k = P_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1}, \tag{5.3}$$

$$P_k^{\mathrm{da}} = P_k^{\mathrm{f}} - P_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} C_k P_k^{\mathrm{f}}, \tag{5.4}$$

$$x_k^{\mathrm{da}} = x_k^{\mathrm{f}} + K_k (y_k - C_k x_k^{\mathrm{f}}). \tag{5.5}$$

Forecast Step

$$x_{k+1}^{\mathrm{f}} = A_k x_k^{\mathrm{da}}, \tag{5.6}$$

$$P_{k+1}^{\mathrm{f}} = A_k P_k^{\mathrm{da}} A_k^{\mathrm{T}} + Q_k. \tag{5.7}$$

The matrices $P_k^{\mathrm{f}} \in \mathbb{R}^{n \times n}$ and $P_k^{\mathrm{da}} \in \mathbb{R}^{n \times n}$ are the state-error covariances, that is,

$$P_k^{\mathrm{f}} = \mathcal{E}[e_k^{\mathrm{f}} (e_k^{\mathrm{f}})^{\mathrm{T}}], \quad P_k^{\mathrm{da}} = \mathcal{E}[e_k^{\mathrm{da}} (e_k^{\mathrm{da}})^{\mathrm{T}}], \tag{5.8}$$

where

$$e_k^{\mathrm{f}} \triangleq x_k - x_k^{\mathrm{f}}, \quad e_k^{\mathrm{da}} \triangleq x_k - x_k^{\mathrm{da}}. \tag{5.9}$$

In the following sections, we consider reduced-rank square-root filters that propagate approximations of a square-root of the error covariance instead of the actual error covariance.

5.3 SVD-Based Reduced-Rank Square-Root Filter

Note that the Kalman filter uses the error covariances P_k^{da} and P_k^{f}, which are updated using (5.4) and (5.7). For computational efficiency, we construct a suboptimal filter that uses reduced-rank approximations of the error covariances P_k^{da} and P_k^{f}. Specifically, we consider reduced-rank approximations $\hat{P}_k^{\mathrm{da}}$ and $\hat{P}_k^{\mathrm{f}}$ of the error covariances P_k^{da} and P_k^{f} such that $\|P_k^{\mathrm{da}} - \hat{P}_k^{\mathrm{da}}\|_{\mathrm{F}}$ and $\|P_k^{\mathrm{f}} - \hat{P}_k^{\mathrm{f}}\|_{\mathrm{F}}$ are minimized, where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. To achieve this approximation, we compute singular value decompositions of the error covariances at each time step.

Let $P \in \mathbb{R}^{n\times n}$ be positive semidefinite, let $\sigma_1 \geqslant \cdots \geqslant \sigma_n$ be the singular values of P, and let $u_1, \ldots, u_n \in \mathbb{R}^n$ be corresponding orthonormal eigenvectors. Next, define $U_q \in \mathbb{R}^{n\times q}$ and $\Sigma_q \in \mathbb{R}^{q\times q}$ by

$$U_q \triangleq \begin{bmatrix} u_1 & \cdots & u_q \end{bmatrix}, \quad \Sigma_q \triangleq \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_q \end{bmatrix}. \tag{5.10}$$

With this notation, the singular value decomposition of P is given by

$$P = U_n \Sigma_n U_n^{\mathrm{T}}, \tag{5.11}$$

where U_n is orthogonal. For $q \leqslant n$, let $\Phi_{\mathrm{SVD}}(P, q) \in \mathbb{R}^{n\times q}$ denote the SVD-based rank-q approximation of the square root $\Sigma_q^{1/2}$ of P given by

$$\Phi_{\mathrm{SVD}}(P, q) \triangleq U_q \Sigma_q^{1/2}. \tag{5.12}$$

The following standard result shows that SS^{T}, where $S \triangleq \Phi_{\mathrm{SVD}}(P, q)$, is the best rank-$q$ approximation of P in the Frobenius norm.

Lemma 5.3.1. *Let $P \in \mathbb{R}^{n\times n}$ be positive semidefinite, and let $\sigma_1 \cdots \geqslant \sigma_n$ be the*

singular values of P. *If* $S = \Phi_{\mathrm{SVD}}(P, q)$, *then*

$$\min_{\mathrm{rank}(\hat{P})=q} \|P - \hat{P}\|_{\mathrm{F}}^2 = \|P - SS^{\mathrm{T}}\|_{\mathrm{F}}^2 = \sigma_{q+1}^2 + \cdots + \sigma_n^2. \tag{5.13}$$

The data assimilation and forecast steps of the SVD-based rank-q square-root filter are given by the following steps:

Data Assimilation step

$$K_{\mathrm{s},k} = \hat{P}_{\mathrm{s},k}^{\mathrm{f}} C_k^{\mathrm{T}} \left(C_k \hat{P}_{\mathrm{s},k}^{\mathrm{f}} C_k^{\mathrm{T}} + R_k \right)^{-1}, \tag{5.14}$$

$$\tilde{P}_{\mathrm{s},k}^{\mathrm{da}} = \hat{P}_k^{\mathrm{f}} - \hat{P}_k^{\mathrm{f}} C_k^{\mathrm{T}} \left(C_k \hat{P}_{\mathrm{s},k}^{\mathrm{f}} C_k^{\mathrm{T}} + R_k \right)^{-1} C_k \hat{P}_{\mathrm{s},k}^{\mathrm{f}}, \tag{5.15}$$

$$x_{\mathrm{s},k}^{\mathrm{da}} = x_{\mathrm{s},k}^{\mathrm{f}} + K_{\mathrm{s},k}(y_k - C_k x_{\mathrm{s},k}^{\mathrm{f}}), \tag{5.16}$$

where

$$\tilde{S}_{\mathrm{s},k}^{\mathrm{f}} \triangleq \Phi_{\mathrm{SVD}}(\tilde{P}_{\mathrm{s},k}^{\mathrm{f}}, q), \tag{5.17}$$

$$\hat{P}_{\mathrm{s},k}^{\mathrm{f}} \triangleq \tilde{S}_{\mathrm{s},k}^{\mathrm{f}} (\tilde{S}_{\mathrm{s},k}^{\mathrm{f}})^{\mathrm{T}}. \tag{5.18}$$

Forecast step

$$x_{\mathrm{s},k+1}^{\mathrm{f}} = A_k x_{\mathrm{s},k}^{\mathrm{da}}, \tag{5.19}$$

$$\tilde{P}_{\mathrm{s},k+1}^{\mathrm{f}} = A_k \hat{P}_{\mathrm{s},k}^{\mathrm{da}} A_k^{\mathrm{T}} + Q_k, \tag{5.20}$$

where

$$\tilde{S}_{\mathrm{s},k}^{\mathrm{da}} \triangleq \Phi_{\mathrm{SVD}}(\tilde{P}_{\mathrm{s},k}^{\mathrm{da}}, q), \tag{5.21}$$

$$\hat{P}_{\mathrm{s},k}^{\mathrm{da}} \triangleq \tilde{S}_{\mathrm{s},k}^{\mathrm{da}} (\tilde{S}_{\mathrm{s},k}^{\mathrm{da}})^{\mathrm{T}}, \tag{5.22}$$

and $\tilde{P}_{\mathrm{s},0}^{\mathrm{f}}$ is positive semidefinite.

Next, define the forecast and data assimilation error covariances $P_{\mathrm{s},k}^{\mathrm{f}}$ and $P_{\mathrm{s},k}^{\mathrm{da}}$ of the SVD-based rank-q square-root filter by

$$P_{\mathrm{s},k}^{\mathrm{f}} \triangleq \mathcal{E}\left[(x_k - x_{\mathrm{s},k}^{\mathrm{f}})(x_k - x_{\mathrm{s},k}^{\mathrm{f}})^{\mathrm{T}} \right], \tag{5.23}$$

$$P_{\mathrm{s},k}^{\mathrm{da}} \triangleq \mathcal{E}\left[(x_k - x_{\mathrm{s},k}^{\mathrm{da}})(x_k - x_{\mathrm{s},k}^{\mathrm{da}})^{\mathrm{T}} \right]. \tag{5.24}$$

Using (5.1), (5.16) and (5.19), it can be shown that

$$P_{\mathrm{s},k}^{\mathrm{da}} = (I - K_{\mathrm{s},k}C)P_{\mathrm{s},k}^{\mathrm{f}}(I - K_{\mathrm{s},k}C)^{\mathrm{T}} + K_{\mathrm{s},k}R_kK_{\mathrm{s},k}^{\mathrm{T}}, \tag{5.25}$$

$$P_{\mathrm{s},k+1}^{\mathrm{f}} = A_kP_{\mathrm{s},k}^{\mathrm{da}}A_k^{\mathrm{T}} + Q_k. \tag{5.26}$$

Note that $\tilde{S}_{\mathrm{s},k}^{\mathrm{f}}\left(\tilde{S}_{\mathrm{s},k}^{\mathrm{f}}\right)^{\mathrm{T}} \leqslant \tilde{P}_{\mathrm{s},k}^{\mathrm{f}}$ and $\tilde{S}_{\mathrm{s},k}^{\mathrm{da}}\left(\tilde{S}_{\mathrm{s},k}^{\mathrm{da}}\right)^{\mathrm{T}} \leqslant \tilde{P}_{\mathrm{s},k}^{\mathrm{da}}$. Hence, even if $\tilde{P}_{\mathrm{s},0}^{\mathrm{f}} = P_{\mathrm{s},0}^{\mathrm{f}}$, it does not necesarily follow that $\tilde{P}_{\mathrm{s},k}^{\mathrm{f}} = P_{\mathrm{s},k}^{\mathrm{f}}$ and $\tilde{P}_{\mathrm{s},k}^{\mathrm{da}} = P_{\mathrm{s},k}^{\mathrm{da}}$ for all $k > 0$. Therefore, since $K_{\mathrm{s},k}$ does not use the true error covariance $P_{\mathrm{s},k}^{\mathrm{f}}$, the SVD-based rank-$q$ square-root filter is generally not equivalent to the Kalman filter. However, under certain conditions, the SVD-based rank-q square-root filter is equivalent to the Kalman filter. Specifically, we have the following result.

Proposition 5.3.1. *Assume that* $\tilde{P}_{\mathrm{s},k}^{\mathrm{f}} = P_{\mathrm{s},k}^{\mathrm{f}}$ *and* $\mathrm{rank}(P_{\mathrm{s},k}^{\mathrm{f}}) \leqslant q$. *Then,* $\tilde{P}_{\mathrm{s},k}^{\mathrm{da}} = P_{\mathrm{s},k}^{\mathrm{da}}$ *and* $\tilde{P}_{\mathrm{s},k+1}^{\mathrm{f}} = P_{\mathrm{s},k+1}^{\mathrm{f}}$. *If, in addition,* $P_{\mathrm{s},k}^{\mathrm{f}} = P_k^{\mathrm{f}}$, *then* $K_{\mathrm{s},k} = K_k$ *and* $P_{\mathrm{s},k+1}^{\mathrm{f}} = P_{k+1}^{\mathrm{f}}$.

Proof. Since $\mathrm{rank}(\tilde{P}_k^{\mathrm{f}}) \leqslant q$, it follows from Lemma 5.3.1 that

$$\hat{P}_{\mathrm{s},k}^{\mathrm{f}} = \tilde{S}_{\mathrm{s},k}^{\mathrm{f}}\left(\tilde{S}_{\mathrm{s},k}^{\mathrm{f}}\right)^{\mathrm{T}} = \tilde{P}_{\mathrm{s},k}^{\mathrm{f}}. \tag{5.27}$$

Hence, it follows from (5.14) that

$$K_{\mathrm{s},k} = P_{\mathrm{s},k}^{\mathrm{f}}C_k^{\mathrm{T}}\left(C_kP_{\mathrm{s},k}^{\mathrm{f}}C_k^{\mathrm{T}} + R_k\right)^{-1}, \tag{5.28}$$

while substituting (5.27) into (5.15) yields

$$\tilde{P}_{\mathrm{s},k}^{\mathrm{da}} = \tilde{P}_{\mathrm{s},k}^{\mathrm{f}} - \tilde{P}_{\mathrm{s},k}^{\mathrm{f}}C_k^{\mathrm{T}}(C_k\tilde{P}_{\mathrm{s},k}^{\mathrm{f}}C_k^{\mathrm{T}} + R_k)^{-1}C_k\tilde{P}_{\mathrm{s},k}^{\mathrm{f}}. \tag{5.29}$$

Next, substituting (5.28) into (5.25) and using $\tilde{P}_{\mathrm{s},k}^{\mathrm{f}} = P_{\mathrm{s},k}^{\mathrm{f}}$ in the resulting expression yields

$$\tilde{P}_{\mathrm{s},k}^{\mathrm{da}} = P_{\mathrm{s},k}^{\mathrm{da}}. \tag{5.30}$$

Since $\operatorname{rank}(\tilde{P}^{\mathrm{f}}_{\mathrm{s},k}) \leqslant q$, it follows from (5.29) that $\operatorname{rank}(\tilde{P}^{\mathrm{da}}_{\mathrm{s},k}) \leqslant q$, and therefore Lemma 5.3.1 implies that

$$\hat{P}^{\mathrm{da}}_{\mathrm{s},k} = \tilde{S}^{\mathrm{da}}_{\mathrm{s},k} \left(\tilde{S}^{\mathrm{da}}_{\mathrm{s},k}\right)^{\mathrm{T}} = \tilde{P}^{\mathrm{da}}_{\mathrm{s},k}. \tag{5.31}$$

Hence, it follows from (5.20), (5.26), and (5.30) that

$$\tilde{P}^{\mathrm{f}}_{\mathrm{s},k+1} = P^{\mathrm{f}}_{\mathrm{s},k+1}. \tag{5.32}$$

Finally, it follows from (5.3) and (5.28) that

$$K_{\mathrm{s},k} = K_k. \tag{5.33}$$

Therefore, (5.4) and (5.25) imply that

$$P^{\mathrm{da}}_{\mathrm{s},k} = P^{\mathrm{da}}_k. \tag{5.34}$$

Hence, it follows from (5.7), (5.26) and (5.34) that

$$P^{\mathrm{f}}_{\mathrm{s},k+1} = P^{\mathrm{f}}_{k+1}. \qquad \square$$

Corollary 5.3.1. *Assume that* $x^{\mathrm{f}}_{\mathrm{s},0} = x^{\mathrm{f}}_0$, $\tilde{P}^{\mathrm{f}}_{\mathrm{s},0} = P^{\mathrm{f}}_{\mathrm{s},0}$, *and* $\operatorname{rank}(P^{\mathrm{f}}_0) \leqslant q$. *Furthermore, assume that, for all* $k \geqslant 0$, $\operatorname{rank}(A_k) + \operatorname{rank}(Q_k) \leqslant q$. *Then, for all* $k \geqslant 0$, $K_{\mathrm{s},k} = K_k$ *and* $x^{\mathrm{f}}_{\mathrm{s},k} = x^{\mathrm{f}}_k$.

Proof. Since $x^{\mathrm{f}}_{\mathrm{s},0} = x^{\mathrm{f}}_0$, (5.8) and (5.23) imply that $P^{\mathrm{f}}_{\mathrm{s},0} = P^{\mathrm{f}}_0$. It follows from (5.25) that if $\operatorname{rank}(P^{\mathrm{f}}_{\mathrm{s},k}) \leqslant q$, then $\operatorname{rank}(P^{\mathrm{da}}_{\mathrm{s},k}) \leqslant q$, and hence (5.26) implies that $\operatorname{rank}(P^{\mathrm{f}}_{\mathrm{s},k+1}) \leqslant q$. Therefore, using Proposition 5.3.1 and induction, it follows that $K_{\mathrm{s},k} = K_k$ for all $k \geqslant 0$. Therefore, (5.5), (5.6), (5.16) and (5.19) imply that $x^{\mathrm{f}}_{\mathrm{s},k} = x^{\mathrm{f}}_k$ for all $k \geqslant 0$. $\square$

5.4 Cholesky-Decomposition-Based Reduced-Rank Square-Root Filter

The Kalman filter gain K_k depends on a particular subspace of the range of the error covariance. Specifically, K_k depends only on $C_k P_k^{\mathrm{f}}$ and not on the entire error covariance. We thus consider a filter that uses reduced-rank approximations $\hat{P}_k^{\mathrm{da}}$ and $\hat{P}_k^{\mathrm{f}}$ of the error covariances P_k^{da} and P_k^{f} such that $\|C_k(P_k^{\mathrm{da}} - \hat{P}_k^{\mathrm{da}})\|_{\mathrm{F}}$ and $\|C_k(P_k^{\mathrm{f}} - \hat{P}_k^{\mathrm{f}})\|_{\mathrm{F}}$ are minimized. To achieve this minimization, we compute a Cholesky decomposition of both error covariances at each time step.

Since $P \in \mathbb{R}^{n \times n}$ is positive semidefinite, the Cholesky decomposition yields a lower triangular Cholesky factor $L \in \mathbb{R}^{n \times n}$ of P that satisfies

$$LL^{\mathrm{T}} = P. \tag{5.35}$$

Partition L as

$$L = \left[\begin{array}{ccc} L_1 & \cdots & L_n \end{array} \right], \tag{5.36}$$

where, for $i = 1, \ldots, n$, $L_i \in \mathbb{R}^n$ has real entries

$$L_i = \left[\begin{array}{ccccc} 0_{1 \times (i-1)} & L_{i,1} & \cdots & L_{i,n-i+1} \end{array} \right]^{\mathrm{T}}. \tag{5.37}$$

Truncating the last $n - q$ columns of L yields the reduced-rank Cholesky factor

$$\Phi_{\mathrm{CHOL}}(P, q) \triangleq \left[\begin{array}{ccc} L_1 & \cdots & L_q \end{array} \right] \in \mathbb{R}^{n \times q}. \tag{5.38}$$

Lemma 5.4.1. *Let $P \in \mathbb{R}^{n \times n}$ be positive definite, define $S \triangleq \Phi_{\mathrm{CHOL}}(P, q)$ and $\hat{P} \triangleq SS^{\mathrm{T}}$, and partition P and $\hat{P}$ as*

$$P = \left[\begin{array}{cc} P_1 & P_{12} \\ (P_{12})^{\mathrm{T}} & P_2 \end{array} \right], \quad \hat{P} = \left[\begin{array}{cc} \hat{P}_1 & \hat{P}_{12} \\ (\hat{P}_{12}) & \hat{P}_2 \end{array} \right], \tag{5.39}$$

where $P_1, \hat{P}_1 \in \mathbb{R}^{q \times q}$ and $P_2, \hat{P}_2 \in \mathbb{R}^{r \times r}$. Then, $\hat{P}_1 = P_1$ and $\hat{P}_{12} = P_{12}$.

Proof. Let L be the Cholesky factor of P. It follows from (5.37) that $L_i L_i^{\mathrm{T}} \in \mathbb{R}^n$ has the structure

$$L_i L_i^{\mathrm{T}} = \begin{bmatrix} 0_{i-1} & 0_{(i-1)\times(n-i+1)} \\ 0_{(n-i+1)\times(i-1)} & X_i \end{bmatrix}, \tag{5.40}$$

where $X_i \in \mathbb{R}^{(n-i+1)\times(n-i+1)}$. Therefore,

$$\sum_{i=q+1}^{n} L_i L_i^{\mathrm{T}} = \begin{bmatrix} 0_{q\times q} & 0_{q\times r} \\ 0_{r\times q} & Y_r \end{bmatrix}, \tag{5.41}$$

where $Y_r \in \mathbb{R}^{r\times r}$. Since

$$P = \sum_{i=1}^{n} L_i L_i^{\mathrm{T}}, \tag{5.42}$$

it follows from (5.38) that

$$P = \hat{P} + \sum_{i=q+1}^{n} L_i L_i^{\mathrm{T}}. \tag{5.43}$$

Substituting (5.41) into (5.43) yields $\hat{P}_1 = P_1$ and $\hat{P}_{12} = P_{12}$. □

Lemma 5.4.1 implies that, if $S = \Phi_{\mathrm{CHOL}}(P, q)$, then the first q rows and columns of SS^{T} and P are equal.

The data assimilation and forecast steps of the Cholesky-based rank-q square-root filter are given by the following steps:

Data Assimilation step

$$K_{\mathrm{c},k} = \hat{P}^{\mathrm{f}}_{\mathrm{c},k} C_k^{\mathrm{T}} \left(C_k \hat{P}^{\mathrm{f}}_{\mathrm{c},k} C_k^{\mathrm{T}} + R_k \right)^{-1}, \tag{5.44}$$

$$\tilde{P}^{\mathrm{da}}_{\mathrm{c},k} = \hat{P}^{\mathrm{f}}_{\mathrm{c},k} - \hat{P}^{\mathrm{f}}_{\mathrm{c},k} C_k^{\mathrm{T}} \left(C_k \hat{P}^{\mathrm{f}}_{\mathrm{c},k} C_k^{\mathrm{T}} + R_k \right)^{-1} C_k \hat{P}^{\mathrm{f}}_{\mathrm{c},k}, \tag{5.45}$$

$$x^{\mathrm{da}}_{\mathrm{c},k} = x^{\mathrm{f}}_{\mathrm{c},k} + K_{\mathrm{c},k}(y_k - C_k x^{\mathrm{f}}_{\mathrm{c},k}), \tag{5.46}$$

where

$$\tilde{S}^{\mathrm{f}}_{\mathrm{c},k} \triangleq \Phi_{\mathrm{SVD}}(\tilde{P}^{\mathrm{f}}_{\mathrm{c},k}, q), \tag{5.47}$$

$$\hat{P}^{\mathrm{f}}_{\mathrm{c},k} \triangleq \tilde{S}^{\mathrm{f}}_{\mathrm{c},k} (\tilde{S}^{\mathrm{f}}_{\mathrm{s},k})^{\mathrm{T}}. \tag{5.48}$$

Forecast step

$$x^{\mathrm{f}}_{\mathrm{c},k+1} = A_k x^{\mathrm{da}}_{\mathrm{c},k}, \tag{5.49}$$

$$\tilde{P}^{\mathrm{f}}_{\mathrm{c},k+1} = A_k \hat{P}^{\mathrm{da}}_{\mathrm{c},k} A_k^{\mathrm{T}} + Q_k, \tag{5.50}$$

where

$$\tilde{S}^{\mathrm{da}}_{\mathrm{c},k} \triangleq \Phi_{\mathrm{SVD}}(\tilde{P}^{\mathrm{da}}_{\mathrm{c},k}, q), \tag{5.51}$$

$$\hat{P}^{\mathrm{da}}_{\mathrm{c},k} \triangleq \tilde{S}^{\mathrm{da}}_{\mathrm{c},k} (\tilde{S}^{\mathrm{da}}_{\mathrm{c},k})^{\mathrm{T}}, \tag{5.52}$$

and $\tilde{P}^{\mathrm{f}}_{\mathrm{c},0}$ is positive semidefinite.

Next, define the forecast and data assimilation error covariances $P^{\mathrm{f}}_{\mathrm{c},k}$ and $P^{\mathrm{da}}_{\mathrm{c},k}$, respectively, of the Cholesky-based rank-q square-root filter by

$$P^{\mathrm{f}}_{\mathrm{c},k} \triangleq \mathcal{E}\left[(x_k - x^{\mathrm{f}}_{\mathrm{c},k})(x_k - x^{\mathrm{f}}_{\mathrm{c},k})^{\mathrm{T}}\right], \tag{5.53}$$

$$P^{\mathrm{da}}_{\mathrm{s},k} \triangleq \mathcal{E}\left[(x_k - x^{\mathrm{da}}_{\mathrm{c},k})(x_k - x^{\mathrm{da}}_{\mathrm{c},k})^{\mathrm{T}}\right], \tag{5.54}$$

that is, $P^{\mathrm{f}}_{\mathrm{c},k}$ and $P^{\mathrm{da}}_{\mathrm{c},k}$ are the error covariances when the Cholesky-based rank-q square-root filter is used. Using (5.1), (5.46) and (5.49), it can be shown that

$$P^{\mathrm{da}}_{\mathrm{c},k} = (I - K_{\mathrm{c},k} C) P^{\mathrm{f}}_{\mathrm{c},k} (I - K_{\mathrm{c},k} C)^{\mathrm{T}} + K_{\mathrm{c},k} R_k K^{\mathrm{T}}_{\mathrm{c},k}, \tag{5.55}$$

$$P^{\mathrm{f}}_{\mathrm{c},k} = A_k P^{\mathrm{da}}_{\mathrm{c},k} A_k^{\mathrm{T}} + Q_k. \tag{5.56}$$

Note that $\tilde{S}^{\mathrm{f}}_{\mathrm{c},k} \left(\tilde{S}^{\mathrm{f}}_{\mathrm{c},k}\right)^{\mathrm{T}} \leqslant \tilde{P}^{\mathrm{f}}_k$ and $\tilde{S}^{\mathrm{da}}_{\mathrm{c},k} \left(\tilde{S}^{\mathrm{da}}_{\mathrm{c},k}\right)^{\mathrm{T}} \leqslant \tilde{P}^{\mathrm{da}}_k$. Hence, even if $\tilde{P}^{\mathrm{f}}_{\mathrm{c},0} = P^{\mathrm{f}}_{\mathrm{c},0}$, the Cholesky-based rank-q square-root filter is generally not equivalent to the Kalman filter. However, in certain cases, the Cholesky-based rank-q square root filter is equivalent to the Kalman filter.

Proposition 5.4.1. *Let A_k and C_k have the structure*

$$A_k = \begin{bmatrix} A_{1,k} & 0 \\ A_{21,k} & A_{2,k} \end{bmatrix}, \quad C_k = \begin{bmatrix} I_p & 0 \end{bmatrix}, \tag{5.57}$$

where $A_{1,k} \in \mathbb{R}^{p\times p}$ *and* $A_{2,k} \in \mathbb{R}^{r\times r}$*, partition* P_k^{f} *and* $\tilde{P}_{\mathrm{c},k}^{\mathrm{f}}$ *as*

$$P_k^{\mathrm{f}} = \begin{bmatrix} P_{1,k}^{\mathrm{f}} & (P_{21,k}^{\mathrm{f}})^{\mathrm{T}} \\ P_{21,k}^{\mathrm{f}} & P_{2,k}^{\mathrm{f}} \end{bmatrix}, \quad \tilde{P}_{\mathrm{c},k}^{\mathrm{f}} = \begin{bmatrix} \tilde{P}_{\mathrm{c},1,k}^{\mathrm{f}} & (\tilde{P}_{\mathrm{c},21,k}^{\mathrm{f}})^{\mathrm{T}} \\ \tilde{P}_{\mathrm{c},21,k}^{\mathrm{f}} & \tilde{P}_{\mathrm{c},2,k}^{\mathrm{f}} \end{bmatrix}, \tag{5.58}$$

where $P_{1,k}^{\mathrm{f}}, \tilde{P}_{\mathrm{c},1,k}^{\mathrm{f}} \in \mathbb{R}^{p\times p}$ *and* $P_{2,k}^{\mathrm{f}}, \tilde{P}_{\mathrm{c},2,k}^{\mathrm{f}} \in \mathbb{R}^{r\times r}$*, and assume that* $q = p$*,* $\tilde{P}_{\mathrm{c},1,k}^{\mathrm{f}} = P_{1,k}^{\mathrm{f}}$*, and* $\tilde{P}_{\mathrm{c},21,k}^{\mathrm{f}} = P_{21,k}^{\mathrm{f}}$*. Then,* $K_{\mathrm{c},k} = K_k$*,* $\tilde{P}_{\mathrm{c},1,k+1}^{\mathrm{f}} = P_{1,k+1}^{\mathrm{f}}$*, and* $\tilde{P}_{\mathrm{c},21,k+1}^{\mathrm{f}} = P_{21,k+1}^{\mathrm{f}}$*.*

Proof. Let P_k^{da} have entries

$$P_k^{\mathrm{da}} = \begin{bmatrix} P_{1,k}^{\mathrm{da}} & (P_{21,k}^{\mathrm{da}})^{\mathrm{T}} \\ P_{21,k}^{\mathrm{da}} & P_{2,k}^{\mathrm{da}} \end{bmatrix}, \tag{5.59}$$

where $P_{1,k}^{\mathrm{da}} \in \mathbb{R}^{p\times p}$ is positive semidefinite and $P_{2,k}^{\mathrm{da}} \in \mathbb{R}^{r\times r}$. It follows from (5.4) that

$$P_{1,k}^{\mathrm{da}} = P_{1,k}^{\mathrm{f}} - P_{1,k}^{\mathrm{f}}(P_{1,k}^{\mathrm{f}} + R_k)^{-1}P_{1,k}^{\mathrm{f}}, \tag{5.60}$$

$$P_{21,k}^{\mathrm{da}} = P_{21,k}^{\mathrm{f}} - P_{21,k}^{\mathrm{f}}(P_{1,k}^{\mathrm{f}} + R_k)^{-1}P_{1,k}^{\mathrm{f}}. \tag{5.61}$$

Substituting (5.57) into (5.3) yields

$$K_k = \begin{bmatrix} P_{1,k}^{\mathrm{f}} \\ P_{21,k}^{\mathrm{f}} \end{bmatrix} (P_{1,k}^{\mathrm{f}} + R_k)^{-1}. \tag{5.62}$$

Furthermore, (5.7) and (5.57) imply that

$$P_{1,k+1}^{\mathrm{f}} = A_{1,k}P_{1,k}^{\mathrm{da}}A_{1,k}^{\mathrm{T}} + Q_{1,k}, \tag{5.63}$$

$$P_{21,k+1}^{\mathrm{f}} = A_{2,k}P_{21,k}^{\mathrm{da}}A_{1,k}^{\mathrm{T}} + A_{21,k}P_{1,k}^{\mathrm{da}}A_{1,k}^{\mathrm{T}} + Q_{21,k}, \tag{5.64}$$

where Q_k has entries

$$Q_k = \begin{bmatrix} Q_{1,k} & (Q_{21,k})^{\mathrm{T}} \\ Q_{21,k} & Q_{2,k} \end{bmatrix}. \tag{5.65}$$

Define $\hat{P}_{\mathrm{c},k}^{\mathrm{f}}$ and $\hat{P}_{\mathrm{c},k}^{\mathrm{da}}$ by (5.48) and (5.52), and let $\hat{P}_{\mathrm{c},k}^{\mathrm{f}}$ and $\hat{P}_{\mathrm{c},k}^{\mathrm{da}}$ have entries

$$\hat{P}_{\mathrm{c},k}^{\mathrm{f}} = \begin{bmatrix} \hat{P}_{\mathrm{c},1,k}^{\mathrm{f}} & (\hat{P}_{\mathrm{c},21,k}^{\mathrm{f}})^{\mathrm{T}} \\ \hat{P}_{\mathrm{c},21,k}^{\mathrm{f}} & \hat{P}_{\mathrm{c},2,k}^{\mathrm{f}} \end{bmatrix}, \hat{P}_{\mathrm{c},k}^{\mathrm{da}} = \begin{bmatrix} \hat{P}_{\mathrm{c},1,k}^{\mathrm{da}} & (\hat{P}_{\mathrm{c},21,k}^{\mathrm{da}})^{\mathrm{T}} \\ \hat{P}_{\mathrm{c},21,k}^{\mathrm{da}} & \hat{P}_{\mathrm{c},2,k}^{\mathrm{da}} \end{bmatrix}, \tag{5.66}$$

where $\hat{P}^{\mathrm{da}}_{1,k}, \hat{P}^{\mathrm{f}}_{1,k} \in \mathbb{R}^{p\times p}$ are positive semidefinite and $\hat{P}^{\mathrm{da}}_{2,k}, \hat{P}^{\mathrm{f}}_{2,k} \in \mathbb{R}^{r\times r}$. Substituting (5.66) into (5.44) yields

$$K_{\mathrm{c},k} = \begin{bmatrix} \hat{P}^{\mathrm{f}}_{1,k} \\ \hat{P}^{\mathrm{f}}_{21,k} \end{bmatrix} (\hat{P}^{\mathrm{f}}_{1,k} + R_k)^{-1}. \tag{5.67}$$

Since $S^{\mathrm{f}}_{\mathrm{c},k} = \Phi_{\mathrm{CHOL}}(\tilde{P}^{\mathrm{f}}_{\mathrm{c},k}, q)$ and we assume that $q = p$, $\tilde{P}^{\mathrm{f}}_{\mathrm{c},1,k} = P^{\mathrm{f}}_{1,k}$, and $\tilde{P}^{\mathrm{f}}_{\mathrm{c},21,k} = P^{\mathrm{f}}_{21,k}$, it follows from Lemma 5.4.1 that

$$\hat{P}^{\mathrm{f}}_{\mathrm{c},1,k} = P^{\mathrm{f}}_{\mathrm{c},1,k}, \quad \hat{P}^{\mathrm{f}}_{\mathrm{c},21,k} = P^{\mathrm{f}}_{\mathrm{c},21,k}. \tag{5.68}$$

Hence, $K_{\mathrm{c},k} = K_k$.

Next, substituting (5.48) into (5.45) yields

$$\tilde{P}^{\mathrm{da}}_{\mathrm{c},k} = \hat{P}^{\mathrm{f}}_{\mathrm{c},k} - \hat{P}^{\mathrm{f}}_{\mathrm{c},k} C_k^{\mathrm{T}} (C_k \hat{P}^{\mathrm{f}}_{\mathrm{c},k} C_k^{\mathrm{T}} + R_k)^{-1} C_k \hat{P}^{\mathrm{f}}_{\mathrm{c},k}. \tag{5.69}$$

Let $\tilde{P}^{\mathrm{da}}_{\mathrm{c},k}$ have entries

$$\tilde{P}^{\mathrm{da}}_{\mathrm{c},k} = \begin{bmatrix} \tilde{P}^{\mathrm{da}}_{\mathrm{c},1,k} & (\tilde{P}^{\mathrm{da}}_{\mathrm{c},21,k})^{\mathrm{T}} \\ \tilde{P}^{\mathrm{da}}_{\mathrm{c},21,k} & \tilde{P}^{\mathrm{da}}_{\mathrm{c},2,k} \end{bmatrix}, \tag{5.70}$$

where $\tilde{P}^{\mathrm{da}}_{\mathrm{c},1,k} \in \mathbb{R}^{p\times p}$ is positive semidefinite and $\tilde{P}^{\mathrm{da}}_{\mathrm{c},2,k} \in \mathbb{R}^{r\times r}$. Therefore, it follows from (5.57) and (5.66) that

$$\tilde{P}^{\mathrm{da}}_{\mathrm{c},1,k} = \hat{P}^{\mathrm{f}}_{\mathrm{c},1,k} - \hat{P}^{\mathrm{f}}_{\mathrm{c},1,k} (\hat{P}^{\mathrm{f}}_{\mathrm{c},1,k} + R_k)^{-1} \hat{P}^{\mathrm{f}}_{\mathrm{c},1,k}, \tag{5.71}$$

$$\tilde{P}^{\mathrm{da}}_{\mathrm{c},21,k} = \hat{P}^{\mathrm{f}}_{\mathrm{c},21,k} - \hat{P}^{\mathrm{f}}_{\mathrm{c},21,k} (\hat{P}^{\mathrm{f}}_{\mathrm{c},1,k} + R_k)^{-1} \hat{P}^{\mathrm{f}}_{\mathrm{c},1,k}. \tag{5.72}$$

Substituting (5.68) into (5.71) and using (5.59) and (5.60) yields

$$\tilde{P}^{\mathrm{da}}_{\mathrm{c},1,k} = P^{\mathrm{da}}_{1,k}, \quad \tilde{P}^{\mathrm{da}}_{\mathrm{c},21,k} = P^{\mathrm{da}}_{21,k}. \tag{5.73}$$

Moreover, since $\tilde{S}^{\mathrm{da}}_{\mathrm{c},k} = \Phi_{\mathrm{CHOL}}(\tilde{P}^{\mathrm{da}}_{\mathrm{c},k}, q)$, it follows from Lemma 5.4.1 that

$$\hat{P}^{\mathrm{da}}_{\mathrm{c},1,k} = \tilde{P}^{\mathrm{da}}_{\mathrm{c},1,k}, \quad \hat{P}^{\mathrm{da}}_{\mathrm{c},21,k} = \tilde{P}^{\mathrm{da}}_{\mathrm{c},21,k}. \tag{5.74}$$

It follows from (5.50) and (5.57) that

$$\tilde{P}^{\mathrm{f}}_{\mathrm{c},1,k+1} = A_{1,k}\hat{P}^{\mathrm{da}}_{1,k}A^{\mathrm{T}}_{1,k} + Q_{1,k}, \tag{5.75}$$

$$\tilde{P}^{\mathrm{f}}_{\mathrm{c},21,k+1} = A_{2,k}\hat{P}^{\mathrm{da}}_{21,k}A^{\mathrm{T}}_{1,k} + A_{21,k}\hat{P}^{\mathrm{da}}_{1,k}A^{\mathrm{T}}_{1,k} + Q_{21,k}. \tag{5.76}$$

Therefore, (5.63), (5.73), and (5.74) imply that

$$\tilde{P}^{\mathrm{f}}_{\mathrm{c},1,k+1} = P^{\mathrm{f}}_{1,k+1}, \quad \tilde{P}^{\mathrm{f}}_{\mathrm{c},21,k+1} = P^{\mathrm{f}}_{21,k+1}. \qquad \square$$

Corollary 5.4.1. *Assume that A_k and C_k have the structure in* (5.57)*. Let $q = p$, $\tilde{P}^{\mathrm{f}}_{\mathrm{c},1,0} = P^{\mathrm{f}}_{1,0}$, $\tilde{P}^{\mathrm{f}}_{\mathrm{c},21,0} = P^{\mathrm{f}}_{21,0}$, and $x^{\mathrm{f}}_{\mathrm{c},0} = x^{\mathrm{f}}_0$. Then for all $k \geqslant 0$, $K_{\mathrm{c},k} = K_k$, and hence $x^{\mathrm{f}}_{\mathrm{c},k} = x^{\mathrm{f}}_k$.*

Proof. Using induction and Proposition 5.4.1 yields $K_{\mathrm{c},k} = K_k$ for all $k \geqslant 0$. Hence, it follows from (5.5), (5.6), (5.46), and (5.49) that $x^{\mathrm{f}}_{\mathrm{c},k} = x^{\mathrm{f}}_k$ for all $k \geqslant 0$. $\square$

5.5 Examples

We compare performance of the SVD-based and Cholesky-based reduced-rank square-root Kalman filters using a compartmental model [56] and 10-DOF mass-spring-damper system.

A schematic diagram of the compartmental model is shown in Fig 5.1. The number n of cells is 20 with one state per cell. All η_{ii} and all η_{ij} $(i \neq j)$ are set to 0.1. We assume that the state of the 9th cell is measured, and disturbances enter all cells so that the disturbance covariance Q has full rank.

We simulate three cases of disturbance covariance and compare costs $J \triangleq \mathcal{E}(e^{\mathrm{T}}_k e_k)$ in Figure 5.3 where (a) shows the cost when $Q = I$, (b) shows the cost when $Q = I$ is

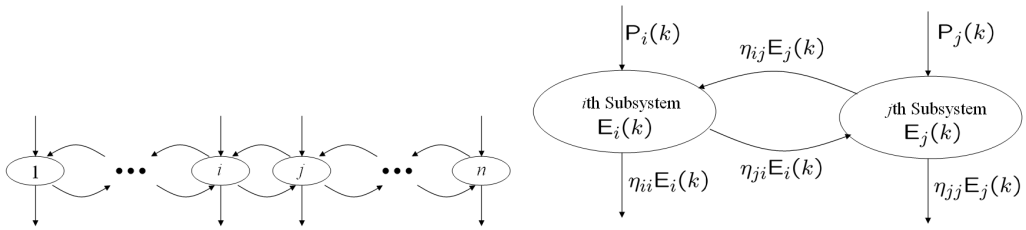


Figure 5.1: Compartmental model involving interconnected subsystems.

changed so that only $Q_{(9,9)}$ is 4.0, and (c) shows cost when $Q = I$ is changed so that only $Q_{(9,9)}$ is 0.25. We reduce the rank of the square-root covariance to 2 in all three cases. In (a) and (b) of Figure 5.3, the Cholesky-based reduced-rank square-root Kalman filter exhibits almost the same performance as the optimal filter whereas the SVD-based reduced-rank square-root Kalman filter shows degraded performance in (a). Meanwhile, in (c), the SVD-based has a large transient and large steady-state offset from the optimal, whereas the Cholesky-based reduced-rank square-root Kalman filter behaves close to the full-order filter.

The mass-spring-damper model is shown in Figure 5.2. The total number of masses is 10 with two states (displacement and velocity) per mass. For $i = 1, \ldots, 10$, $m_i = 1$, and $k_j = 1$, $c_j = 0.01$, $j = 1, \ldots, 11$. We assume that the displacement of the 5th mass is measured and disturbances enter all states so that disturbance covariance Q has full rank.

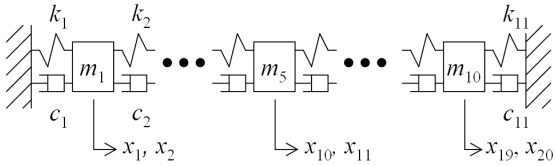


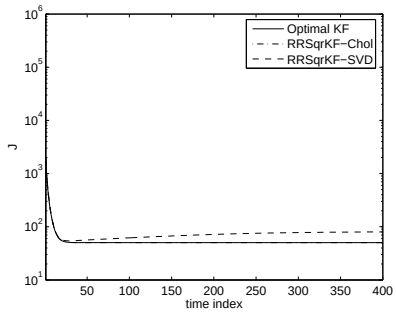
Figure 5.2: Mass-spring-damper system.

We simulate three cases of disturbance covariance and compare costs J in (a),(b) and (c) of Figure 5.4. (a) shows costs when $Q = I$, (b) shows costs when $Q = I$ is

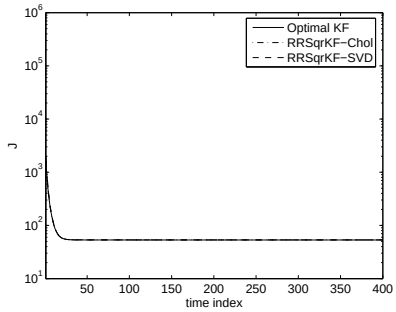
changed so that only $Q_{(9,9)}$ is 4.0 and (c) shows costs when $Q = I$ is changed so that only $Q_{(9,9)}$ is 0.25. We reduce the rank of square-root covariance to 2 in all three cases. In (d) and (e), the costs for the Cholesky-based and SVD-based reduced-rank square-root Kalman filters are close to each other but larger than the optimal. Meanwhile, in (f), the SVD-based shows unstable behavior, while the Cholesky-based filter remains stable and nearly optimal.

5.6 Conclusions

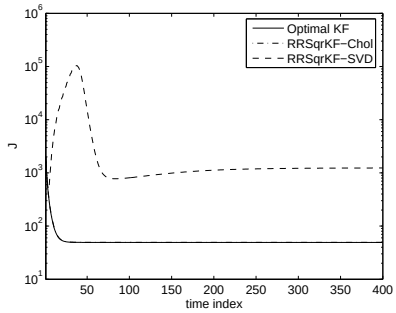
We developed a Cholesky decomposition method to obtain reduced-rank square-root Kalman filters. We showed that the SVD-based reduced-rank square-root Kalman filter is equivalent to the optimal filter when the reduced-rank is equal to or greater than the rank of error-covariance, while the Cholesky-based is equivalent to the optimal when the system is lower triangular block-diagonal according to the observation matrix C which has the form $[I_{p\times p}\ 0]$ and p is equal to the reduced-rank q. Furthermore, the Cholesky-based rank-q square root filter is equivalent to the optimal Kalman filter for a specific number of time steps. In general cases, the Cholesky-based does not always perform better than the SVD-based and vice versa. Finally, using simulation examples, we showed that the Cholesky-based exhibits more stable performance than the SVD-based filter, which can become unstable when the strong disturbances enter the system states that are not measured.



(a)

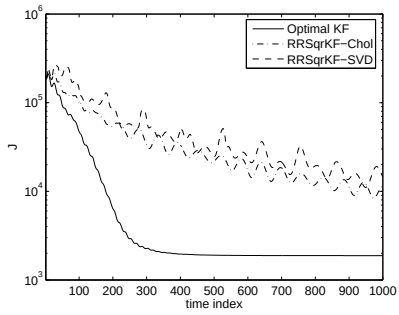


(b)

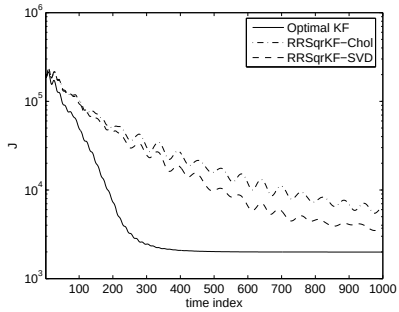


(c)

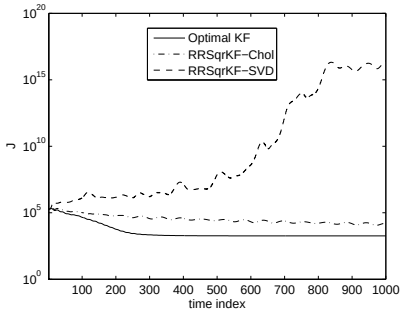
Figure 5.3: Time evolutions of the cost $J \triangleq \mathcal{E}(e_k^{\mathrm{T}} e_k)$ for the compartmental system. For this example, $R = 10^{-4}$, $P_0 = 10^2 I$, the rank of reduced rank filters is fixed at 2 and energy measurement is taken at compartment 9. Disturbances enter compartments 1,2,...,20. In (a), the cost J is when disturbance covariance is $Q = I_{20\times 20}$, (b) is for the case when $Q_{(9,9)}$ is changed to 4.0 and (c) is for the case when $Q_{(9,9)}$ is changed to 0.25.



(a)



(b)



(c)

Figure 5.4: Time evolutions of the cost $J = \mathcal{E}(e_k^{\mathrm{T}} e_k)$ of the mass-spring-damper system. Here $R = 10^{-4}$, $P_0 = 100^2 I$, the rank of reduced rank filters is fixed at 2, and a displacement measurement of m_5 is available. Velocity and force disturbances enter mass 1,2,. . .,10. (a) shows the cost J when disturbance covariance is $Q = I_{20\times 20}$ while (b) is for the case when $Q_{(9,9)}$ is changed to 4.0 and (c) is for the case when $Q_{(9,9)}$ is changed to 0.25.

CHAPTER VI

Localized Data Assimilation in the Ionosphere-Thermosphere Using a Sampled-Data Unscented Kalman Filter

6.1 Introduction

For nonlinear estimation and data assimilation, the terrestrial weather forecasting community has largely adopted the ensemble Kalman filter (EnKF) [79–84]. This technique retains the data injection form of the Kalman filter but does not propagate the error covariance in the classical manner. Instead, EnKF propagates an ensemble of systems under random forcing and initial states to estimate the error covariance. This technique is applicable in principle to highly nonlinear systems. Although the size of the ensemble affects the accuracy of the estimates, there are no theoretical guidelines for determining the size of the ensemble.

For applications involving nonlinear and non-Gaussian systems [85], particle filters are used. A particle filter obtains estimates through Monte Carlo simulation at each step using an assumed probability density function and resampling technique [86]. Although particle filters can estimate the state of a nonlinear system with a non-Gaussian probability density, there is no definitive guideline for determining the number of sample points for achieving good accuracy.

In contrast with the ensemble Kalman filter and particle filters, the unscented

Kalman filter (UKF) uses a deterministic number of ensemble members (specifically, $2n+1$ ensemble members, where n is the number of states of the system) to estimate the error covariance and obtain the data-injection gain [33, 87]. The fundamental component of UKF is the unscented transformation, which uses a minimal set of specially chosen weighted points to parameterize the mean and covariance of the state probability distribution. These sample points, which capture the mean and covariance of a Gaussian random variable, are propagated through the model to capture the posterior mean and covariance to second order for smooth but arbitrary nonlinearities. Furthermore, UKF treats the model and its software implementation as a black box, which eliminates the need to construct a Jacobian as required by the extended Kalman filter (XKF) [88].

In view of these advantages, the goal of the present paper is to apply UKF to data assimilation for space weather applications. In particular, we focus on the Earth's atmosphere between 100 km and 1000 km altitude, a region known as the ionosphere-thermosphere. For this objective we use the parallel global ionosphere-thermosphere model (GITM) code as the basis of data assimilation. Using UKF, we eliminate the need for either the Jacobian required by XKF or a dynamics factorization required by the state-dependent Riccati equation (SDRE) filter [89]. In addition, for flow problems, UKF is significantly more accurate than XKF under highly nonlinear conditions [88].

For large-scale systems, however, the $2n+1$ ensemble size of UKF presents a significant computational burden. For example, in vertical (altitude-only) 1D GITM, with $n = 700$, the total number of states in the UKF ensemble for GITM can reach 900,000, while, for the 3D case with 5° resolution in longitude and latitude, with $n = 1,814,400$, the total number of states exceeds 10^{13} (10 trillion). The result-

ing computational requirement necessitates localized UKF, wherein data injection is confined to a specified region, with coupling to data-free simulation in the exterior region [90, 91]. With this approach, data assimilation based on 3D GITM is feasible through parallel implementation on a multiprocessor cluster.

Data for GITM are provided by ground-based or space-based sensors. Since measurements are not available at every integration time step, we perform the unscented transformation to update the covariance and state only when measurement data are available. Between measurement update times, GITM runs in data-free simulation mode. Another relevant issue in data assimilation based on GITM is that data assimilation performance depends on the accuracy of the disturbance and measurement noise covariances. Due to the high nonlinearity of GITM, UKF is sensitive to the disturbance covariance and may become unstable or yield poor performance. Although the measurement noise covariance is known, a disturbance covariance must be constructed to capture the effect of external drivers. For the case of solar irradiation, we approximate the disturbance covariance by means of a Monte Carlo method. Alternative techniques are discussed in [92]

6.2 Sampled-Data UKF

Consider the discrete-time nonlinear system

$$x_{k+1} = f(x_k, u_k, k) + w_k \tag{6.1}$$

and measurements

$$y_k = h(x_k, k) + v_k, \tag{6.2}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, and $y_k \in \mathbb{R}^p$. The input u_k and output y_k are assumed to be measured, and $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^p$ are uncorrelated zero-mean white noise processes

with covariances Q_k and R_k, respectively. We assume that R_k is positive definite. The inputs u_k and w_k represent known and unknown physics drivers, respectively.

The starting point for UKF is a collection of state estimates or sample points that capture the initial probability distribution of the state. The unscented transformation is used to construct these sample points with a specified mean and variance.

To define this procedure, let $\overline{x} \in \mathbb{R}^n$, let $\overline{P} \in \mathbb{R}^{n\times n}$ be positive semidefinite, and let $\lambda > 0$. The unscented transformation provides $2n+1$ sample points $X_i \in \mathbb{R}^n$ and corresponding weights $\gamma_{\mathrm{s},i}$ and $\gamma_{\mathrm{P},i}$ so that $\overline{x}$ and $\overline{P}$ are the weighted mean and weighted variance of the sample points, respectively. The unscented transformation

$$X = \Psi(\overline{x}, \overline{P}, \lambda) \in \mathbb{R}^{n\times(2n+1)} \tag{6.3}$$

of $\overline{x}$ with covariance $\overline{P}$ is defined by

$$X_i = \begin{cases} \overline{x}, & \text{if } \ i = 0, \\ \overline{x} + \sqrt{\lambda}\tilde{P}_i, & \text{if } \ i = 1, \ldots, n, \\ \overline{x} - \sqrt{\lambda}\tilde{P}_{i-n}, & \text{if } \ i = n+1, \ldots, 2n, \end{cases} \tag{6.4}$$

where $\tilde{P}_i$ is the ith column of $\tilde{P} \in \mathbb{R}^{n\times n}$, which satisfies $\tilde{P}^{\mathrm{T}}\tilde{P} = \overline{P}$, and X_i is the ith column of X. The parameter $\lambda > 0$ determines the spread of the sample points around $\overline{x}$ but can otherwise be chosen arbitrarily. Note that

$$\sum_{i=0}^{2n} \gamma_{\mathrm{s},i} X_i = \overline{x}, \ \ \sum_{i=0}^{2n} \gamma_{\mathrm{P},i}(X_i - \overline{x})(X_i - \overline{x})^{\mathrm{T}} = \overline{P}, \tag{6.5}$$

where the weights $\gamma_{\mathrm{s},i}$ and $\gamma_{\mathrm{P},i}$ are defined by

$$\gamma_{\mathrm{s},0} \ \triangleq \ 1 - \frac{n}{\lambda}, \ \ \gamma_{\mathrm{P},0} \triangleq 1 - \frac{n}{\lambda} + (1 - \frac{\lambda}{n} + \beta), \tag{6.6}$$

$$\gamma_{\mathrm{s},i} \ = \ \gamma_{\mathrm{P},i} \triangleq \frac{1}{2\lambda}, \ \ i = 1, \ldots, 2n. \tag{6.7}$$

The parameter $\beta \geq 0$ can be chosen arbitrarily; it is customary to set $\beta = 2$.

UKF uses the unscented transformation to update the state estimate by simulating $2n+1$ copies of the model with the initial conditions $X_0, \ldots, X_{2n}$, and by using the propagated states to approximate the mean and covariance of the state error. We assume that an initial estimate x_0^{f} of the state x_0 is given along with an initial error covariance $P_0^{\mathrm{f}} \in \mathbb{R}^{n \times n}$.

For data assimilation based on GITM, we consider simulated data that are representative of an incoherent scatter radar (ISR). The ISR data update rate is typically much slower than the GITM integration time step. For the present study, we implement UKF for GITM with state and error-covariance measurement updates occurring every 60 seconds, whereas the integration time step for the GITM advection equations (see (6.39)-(6.41)) is 1 second.

Between measurement updates, the standard approach is to propagate the error-covariance in open loop. However, since the major dynamics of GITM are slow compared to the measurement update rate, we freeze the error covariance between measurement updates. A similar technique is used in [93, 94].

The sampled-data UKF with data available every N steps and with frozen inter-sample error covariance is illustrated in Figure 6.1. We assume that measurements are available at the sample instants $k = N, 2N, 3N, \ldots$.

The UKF data assimilation step for $k = iN$ given $x_k^{\mathrm{f}}, P_k^{\mathrm{f}}, \lambda, y_k$, and R_k is given by

$$X_k^{\mathrm{f}} \triangleq \Psi(x_k^{\mathrm{f}}, P_k^{\mathrm{f}}, \lambda), \tag{6.8}$$

$$Y_{i,k}^{\mathrm{f}} \triangleq h(X_{i,k}^{\mathrm{f}}, k), \tag{6.9}$$

$$y_k^{\mathrm{f}} \triangleq \sum_{i=0}^{2n} \gamma_{\mathrm{s},i} Y_{i,k}^{\mathrm{f}}, \tag{6.10}$$

$$P_{\mathrm{xy},k} \triangleq \sum_{i=0}^{2n} \gamma_{\mathrm{P},i} (X_{i,k}^{\mathrm{f}} - x_k^{\mathrm{f}})(Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})^{\mathrm{T}}, \tag{6.11}$$

$$P_{\mathrm{yy},k} \triangleq \sum_{i=0}^{2n} \gamma_{\mathrm{P},i} (Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})(Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})^{\mathrm{T}} + R_k, \tag{6.12}$$

$$K_k \triangleq P_{\mathrm{xy},k} P_{\mathrm{yy},k}^{-1}, \tag{6.13}$$

$$x_k^{\mathrm{da}} = x_k^{\mathrm{f}} + K_k(y_k - y_k^{\mathrm{f}}), \tag{6.14}$$

$$P_k^{\mathrm{da}} = P_k^{\mathrm{f}} - K_k P_{\mathrm{yy},k} K_k^{\mathrm{T}}. \tag{6.15}$$

The UKF forecast step for $k = iN$ given $x_k^{\mathrm{da}}, P_k^{\mathrm{da}}, \lambda, u_k$, and Q_k is given by

$$X_k^{\mathrm{da}} \triangleq \Psi(x_k^{\mathrm{da}}, P_k^{\mathrm{da}}, \lambda), \tag{6.16}$$

$$\tilde{X}_{i,k+1}^{\mathrm{f}} = f(X_{i,k}^{\mathrm{da}}, u_k, k), \tag{6.17}$$

$$x_{k+1}^{\mathrm{f}} \triangleq \sum_{i=0}^{2n} \gamma_{\mathrm{s},i} \tilde{X}_{i,k+1}^{\mathrm{f}}, \tag{6.18}$$

$$P_{k+1}^{\mathrm{f}} = \sum_{i=0}^{2n} \gamma_{\mathrm{P},i} (\tilde{X}_{i,k+1}^{\mathrm{f}} - x_{k+1}^{\mathrm{f}})(\tilde{X}_{i,k+1}^{\mathrm{f}} x_{k+1}^{\mathrm{f}})^{\mathrm{T}} + Q_k. \tag{6.19}$$

The UKF forecast step for $k = iN + 1, \ldots, (i+1)N - 1$ given x_k^{f}, u_k, and P_k^{f} is given by

$$x_{k+1}^{\mathrm{f}} = f(x_k^{\mathrm{f}}, u_k, k), \tag{6.20}$$

$$P_{k+1}^{\mathrm{f}} = P_k^{\mathrm{f}} \text{ (frozen)}. \tag{6.21}$$

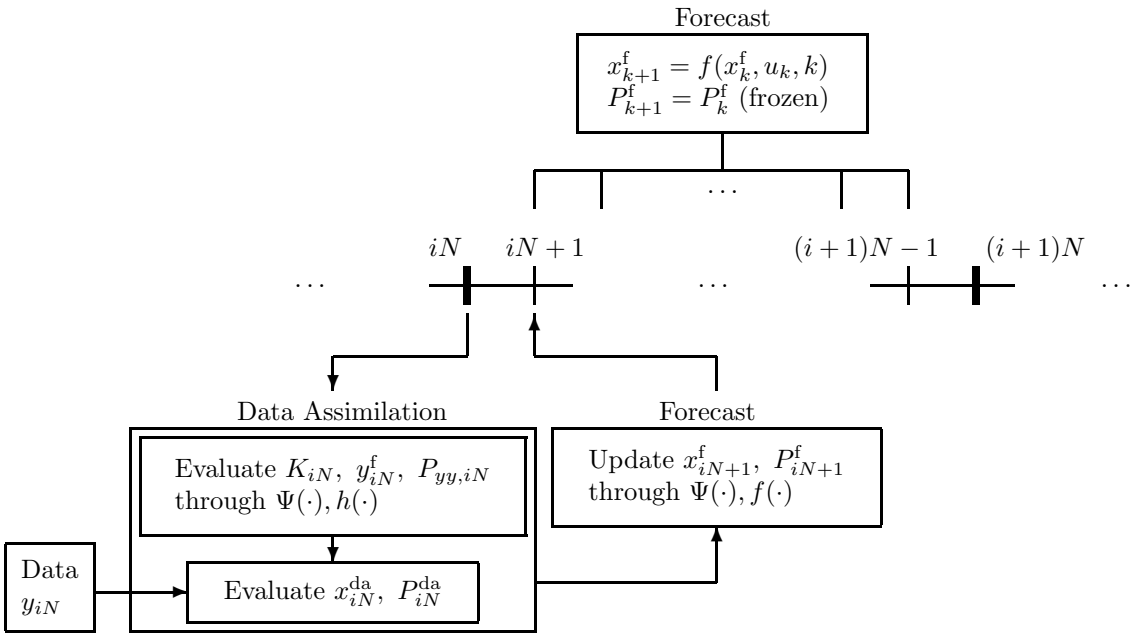


Figure 6.1: Timing diagram of sampled-data UKF with frozen intersample error covariance.

6.3 Localized UKF

UKF estimates all states through data injection. In particular, for a system with n states, UKF requires $2n + 1$ simulation model updates, which are used to update the $n \times n$ covariance. If n is large, as in the case of GITM, then the computational burden of implementing UKF is enormous. We address this problem by performing UKF updates of both the states and the error covariance locally while treating the subsystem coupling terms as known inputs. The benefit of data assimilation thus reaches the entire system through coupling between the localized and exterior regions.

Assume that the state $x_k \in \mathbb{R}^n$ has the components

$$x_k = \begin{bmatrix} x_{\mathrm{L},k} \\ \\ x_{\mathrm{E},k} \end{bmatrix}, \tag{6.22}$$

where $x_{\mathrm{L},k} \in \mathbb{R}^{n_\mathrm{L}}$ and $x_{\mathrm{E},k} \in \mathbb{R}^{n_\mathrm{E}}$ denote the states of the localized and exterior regions, respectively, and $n_\mathrm{L} + n_\mathrm{E} = n$. We, assume that the measurements depend entirely on the state $x_{\mathrm{L},k}$ so that y_k can be expressed as

$$y_k = h(x_{\mathrm{L},k}, k) + v_k. \tag{6.23}$$

Finally, partition Q_k and P_k as

$$Q_k = \begin{bmatrix} Q_{\mathrm{L},k} & Q_{\mathrm{LE},k} \\ Q_{\mathrm{LE},k}^{\mathrm{T}} & Q_{\mathrm{E},k} \end{bmatrix}, P_k = \begin{bmatrix} P_{\mathrm{L},k} & P_{\mathrm{LE},k} \\ (P_{\mathrm{LE},k})^{\mathrm{T}} & P_{\mathrm{E},k} \end{bmatrix}. \tag{6.24}$$

The objective is to directly inject the measurement data y_k into only the states corresponding to the estimate of $x_{\mathrm{L},k}$ by using a reduced-order error covariance.

In the data assimilation step of UKF, we inject data into the $x_{\mathrm{L},k}$ subsystem and update $P_{\mathrm{L},k}^{\mathrm{f}}$ as if only the $x_{\mathrm{L},k}$ subsystem were present. However, in the forecast step (6.17)-(6.19), we update the full state vector $X_{i,k} \in \mathbb{R}^n$ through the dynamics $f(\cdot)$ in (6.17) but with the number of ensembles reduced from $2n + 1$ to $2n_\mathrm{L} + 1$ since the ensembles are determined from $P_{\mathrm{L},k}^{\mathrm{da}} \in \mathbb{R}^{n_\mathrm{L} \times n_\mathrm{L}}$. This technique is the localized unscented Kalman filter (LUKF) [90, 91].

LUKF data assimilation step for $k = N, 2N, 3N, \ldots$ is given by

$$X_{\mathrm{L},k}^{\mathrm{f}} \triangleq \Psi(x_{\mathrm{L},k}^{\mathrm{f}}, P_{\mathrm{L},k}^{\mathrm{f}}, \lambda), \tag{6.25}$$

$$Y_{i,k}^{\mathrm{f}} \triangleq h(X_{\mathrm{L},i,k}^{\mathrm{f}}, k), \tag{6.26}$$

$$y_k^{\mathrm{f}} \triangleq \sum_{i=0}^{2n_\mathrm{L}} \gamma_{\mathrm{s},i} Y_{i,k}^{\mathrm{f}}, \tag{6.27}$$

$$P_{\mathrm{x_L y},k} \triangleq \sum_{i=0}^{2n_\mathrm{L}} \gamma_{\mathrm{P},i} (X_{\mathrm{L},i,k}^{\mathrm{f}} - x_{\mathrm{L},k}^{\mathrm{f}})(Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})^{\mathrm{T}}, \tag{6.28}$$

$$P_{\mathrm{yy},k} \triangleq \sum_{i=0}^{2n_\mathrm{L}} \gamma_{\mathrm{P},i} (Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})(Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})^{\mathrm{T}} + R_k, \tag{6.29}$$

$$K_{\mathrm{L},k} \triangleq P_{\mathrm{x_L y},k} P_{\mathrm{yy},k}^{-1}, \tag{6.30}$$

$$x_{\mathrm{L},k}^{\mathrm{da}} = x_{\mathrm{L},k}^{\mathrm{f}} + K_{\mathrm{L},k}(y_k - y_k^{\mathrm{f}}), \tag{6.31}$$

$$x_{\mathrm{E},k}^{\mathrm{da}} = x_{\mathrm{E},k}^{\mathrm{f}} \tag{6.32}$$

$$P_{\mathrm{L},k}^{\mathrm{da}} = P_{\mathrm{L},k}^{\mathrm{f}} - K_{\mathrm{L},k} P_{\mathrm{yy},k} K_{\mathrm{L},k}^{\mathrm{T}}, \tag{6.33}$$

where, for $i = 0, \ldots, 2n_{\mathrm{L}}$, $X_{\mathrm{L},i,k}^{\mathrm{f}} \in \mathbb{R}^{n_{\mathrm{L}}}$ is the $(i+1)$th column of $X_{\mathrm{L},k}^{\mathrm{f}}$. Note that only $2n_{\mathrm{L}} + 1$ ensembles are used rather than $2n + 1$ ensembles as in UKF, while (6.31)-(6.32) indicate that measurement data are injected directly into only the estimates of the state $x_{\mathrm{L},k}$ corresponding to the localized region.

The LUKF forecast step is given by

$$X_{\mathrm{L},k}^{\mathrm{da}} \triangleq \Psi(x_{\mathrm{L},k}^{\mathrm{da}}, P_{\mathrm{L},k}^{\mathrm{da}}, \lambda), \tag{6.34}$$

$$\tilde{X}_{i,k+1}^{\mathrm{f}} = f(X_{i,k}^{\mathrm{da}}, u_k, k), \tag{6.35}$$

$$x_{k+1}^{\mathrm{f}} \triangleq \sum_{i=0}^{2n_{\mathrm{L}}} \gamma_{\mathrm{s},i} \tilde{X}_{i,k+1}^{\mathrm{f}}, \tag{6.36}$$

$$P_{\mathrm{L},k+1}^{\mathrm{f}} = \sum_{i=0}^{2n_{\mathrm{L}}} \gamma_{\mathrm{P},i} (\tilde{X}_{\mathrm{L},i,k+1}^{\mathrm{f}} - x_{\mathrm{L},k+1}^{\mathrm{f}})(\tilde{X}_{\mathrm{L},i,k+1}^{\mathrm{f}} - x_{\mathrm{L},k+1}^{\mathrm{f}})^{\mathrm{T}} + Q_{\mathrm{L},k}. \tag{6.37}$$

where, for all $i = 0, \ldots, 2n_{\mathrm{L}}$, $X_{i,k}^{\mathrm{da}} \in \mathbb{R}^n$, $\tilde{X}_{i,k+1}^{\mathrm{f}} \in \mathbb{R}^n$ have the form

$$X_{i,k}^{\mathrm{da}} \triangleq \begin{bmatrix} X_{\mathrm{L},i,k}^{\mathrm{da}} \\ x_{\mathrm{E},k}^{\mathrm{da}} \end{bmatrix}, \ \tilde{X}_{i,k+1}^{\mathrm{f}} = \begin{bmatrix} \tilde{X}_{\mathrm{L},i,k+1}^{\mathrm{f}} \\ \tilde{X}_{\mathrm{E},i,k+1}^{\mathrm{f}} \end{bmatrix}, \tag{6.38}$$

where $X_{\mathrm{L},i,k}^{\mathrm{da}} \in \mathbb{R}^{n_{\mathrm{L}}}$ is the $(i+1)$th column of $X_{\mathrm{L},k}^{\mathrm{da}}$, and $\tilde{X}_{i,k+1}^{\mathrm{f}}$ has components $\tilde{X}_{\mathrm{L},i,k+1}^{\mathrm{f}} \in \mathbb{R}^{n_{\mathrm{L}}}$ and $\tilde{X}_{\mathrm{E},i,k+1}^{\mathrm{f}} \in \mathbb{R}^{n_{\mathrm{E}}}$. Notice that the estimate $x_{\mathrm{E},k}^{\mathrm{da}}$ of the state $x_{\mathrm{E},k}$ in all of the ensembles of LUKF in (6.38) is the same, whereas the estimate $X_{\mathrm{L},i,k}^{\mathrm{da}}$ of the state $x_{\mathrm{L},k}$ is different in each ensemble.

Although (6.32) implies that data are not directly injected into the state estimates corresponding to $x_{\mathrm{E},k}$, it follows from $X^{\mathrm{da}}_{\mathrm{L},i,k}$ in (6.38), (6.34), and (6.36) that the measurement data affect the estimates of the state $x_{\mathrm{E},k}$ through the dynamic coupling between $x_{\mathrm{L},k}$ and $x_{\mathrm{E},k}$. Since LUKF involves $2n_{\mathrm{L}}+1$ model updates, the number of states involved is of the order $(2n_{\mathrm{L}}+1)n$. Hence, when $n_{\mathrm{L}} \ll n$, LUKF is computationally less demanding than UKF.

6.4 GITM Model

The GITM is a fully parallel three-dimensional finite-volume model that simulates the coupled ionosphere-thermosphere system over the entire surface of the Earth in spherical coordinates. GITM for data assimilation is based on 14 state variables per cell, namely, the number densities of the neutral species O, O_2, N_2, N, the vertical velocities of each neutral species, the eastward and northward bulk neutral velocities u_ϕ, u_θ, respectively, the normalized neutral mean temperature $\mathcal{T}_{\mathrm{n}}$, and the number densities of the ions O^+, O_2^+, and NO^+. Here, number density denotes the number of neutral molecules or ions per cubic meter. These state variables, which are updated by solving the equations of continuity, momentum, and energy, are used to compute the number density of electrons N_{e}, the ion velocity $\mathbf{v}_{\mathrm{ion}}$, and the ion temperature T_{ion}, which correspond to ISR data. Some of the major features of the model are described here. For a detailed description of the physics, dynamics, and numerical schemes, see [95].

GITM uses a block-based, two-dimensional uniform domain decomposition in latitude and longitude as well as a nonuniform grid in altitude, where cell height increases with altitude. This approach facilitates parallel computation over the entire domain, with information passed among cells after each iteration. The horizontal (longitude

and latitude) grid resolution is set at run time, which thus specifies the number of blocks used in the simulation. The grid is fixed for each simulation. GITM can be run in one-dimensional 1D vertical mode by using one block in both latitude and longitude. In this case, the horizontal dynamics are not taken into account. The ability to run in 1D renders long-term studies of the upper atmosphere feasible, while facilitating quick debugging of the code, since 1D GITM can also be run inexpensively on a personal computer. However 3D GITM, which is necessarily global, requires parallel implementation on a multiprocessor platform.

The altitude grid size is based on the scale height, which is the exponential rate at which the density decays. The code is initialized using the mass spectrometer incoherent scatter (MSIS) [96] and international reference ionosphere (IRI) models [97] to specify the neutral and ion densities and temperatures. The lower boundary of the simulated region is specified by the user, as is the number of vertical grid points. The scale height is calculated during initialization, and the vertical grid spacing is set to 1/3 of the scale height. The upper boundary then depends on the number of grid points requested.

The model can use a variety of externally imposed magnetic field configurations, including a dipole field, a tilted dipole, or the internal geomagnetic reference field (IGRF) with apex coordinates [98].

GITM is programmed so that it can be restarted from a previously saved state. This feature allows long time periods to be simulated in disconnected segments of time without any break in the physics.

Several features render GITM distinct from alternative models of the ionosphere-thermosphere system. First, GITM does not assume that gravity is constant over the domain. Next, GITM uses an altitude-based coordinate system rather than pressure-

based coordinates. Also, GITM does not assume hydrostatic equilibrium, where the gradient of pressure in the vertical direction is assumed to equal gravity. Instead, GITM uses a more realistic pressure model that includes terms due to Coriolis, centrifugal acceleration, and ion drag in the vertical momentum equation.

The ionosphere-thermosphere system is highly coupled to the surrounding regions. The incident solar radiation acts to heat the dayside atmosphere, while, at high latitudes, the magnetosphere deposits energy into the atmosphere. GITM can use various high-latitude and solar radiation drivers. Typically, the high latitudes are driven by the Weimer potential model [99] or using results from the assimilative mapping of ionospheric electrodynamics technique (AMIE) [100]. In order to specify the solar flux in the extreme ultraviolet (EUV) range of the spectrum, GITM uses either an empirical model based on the daily measure of the 10.7-cm solar flux $F_{10.7}$, or data from the solar EUV experiment (SEE) [101].

GITM solves two major sources of dynamics, namely, advection and chemistry. Advection of neutral and ion species is modeled by

$$\frac{\partial N}{\partial t} + N\nabla \cdot \mathbf{u} + \mathbf{u} \cdot \nabla N = \mathcal{S}, \tag{6.39}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla \mathcal{T} + \frac{\mathcal{T}}{\rho}\nabla \rho = \mathcal{F}, \tag{6.40}$$

$$\frac{\partial \mathcal{T}}{\partial t} + \mathbf{u} \cdot \nabla \mathcal{T} + (\gamma - 1)\mathcal{T}\nabla \cdot \mathbf{u} = \mathcal{Q}, \tag{6.41}$$

where (6.39), (6.40), and (6.41) are the continuity, momentum, and energy equations, respectively, N is the number density of a neutral or an ion; $\mathbf{u}$ is the wind velocity vector; $\mathcal{S}$ is the corresponding source term; $\mathcal{T} = p/\rho$ is the normalized temperature, where p is pressure, ρ is mass density; $\mathcal{F}$ is the forcing input due to gravity, pressure gradients, Coriolis and centrifugal forces, and ion-neutral and neutral-neutral friction; γ is the ratio of specific heats, and $\mathcal{Q}$ is the thermal energy input rate.

For chemistry, the ionization of the neutral species due to solar EUV, which is the primary source of ions on the dayside, is expressed as

$$\mathrm{O} + h\nu \longrightarrow \mathrm{O}^+ + \mathrm{e}, \tag{6.42}$$

$$\mathrm{O}_2 + h\nu \longrightarrow \mathrm{O}_2^+ + \mathrm{e}, \tag{6.43}$$

$$\mathrm{N}_2 + h\nu \longrightarrow \mathrm{N}_2^+ + \mathrm{e}, \tag{6.44}$$

$$\mathrm{N} + h\nu \longrightarrow \mathrm{N}^+ + \mathrm{e}, \tag{6.45}$$

where h is the Planck constant, ν is the frequency of the radiation, and e is an electron. Note that $h\nu$ represents the quantized energy of the photons of radiation having frequency ν.

In GITM, the advection equations are discretized using the second-order Lax-Friedrichs scheme [102], while the stoichiometric equations are solved by a subcycling technique [95, 103], which reduces the chemistry time step within the one-step advection when the change in any species exceeds 25% in the advective time step.

6.5 Incoherent Scatter Radar Measurements

While ISRs provide several data products [104], we consider the number density N_e of electron, the line-of-sight component of ion velocity $\mathbf{v}_\mathrm{ion}$, and the ion temperature T_ion. All ions are assumed to move in same velocity $\mathbf{v}_\mathrm{ion}$, while T_ion is the average value over the ion species. For 1D GITM, we assume that the radar is pointed vertically to measure the vertical component of ion velocity.

The number density N_e of electrons is given by

$$N_\mathrm{e} = N_{\mathrm{O}^+} + N_{\mathrm{O}_2^+} + N_{\mathrm{NO}^+}, \tag{6.46}$$

where $N_{\mathrm{O}^+}, N_{\mathrm{O}_2^+}$, and N_{NO^+} are the number densities of $\mathrm{O}^+, \mathrm{O}_2^+$, and NO^+, respectively. For data assimilation we use the logarithm of the number densities, which are

typically 10^{12} particles/m^3.

Next, the ion velocity $\mathbf{v}_{\text{ion}}$ is given by

$$\mathbf{v}_{\text{ion}} = \frac{\mathbf{A}\cdot\mathbf{b}}{\rho_{\text{ion}}\nu_{\text{ion/n}}} + \frac{\rho_{\text{ion}}^2\nu_{\text{ion/n}}\mathbf{A}_\perp + eN_{\text{e}}\mathbf{A}_\perp\times\mathbf{B}}{\rho_{\text{ion}}^2\nu_{\text{ion/n}}^2 + e^2N_{\text{e}}^2B^2}, \tag{6.47}$$

where

$$\mathbf{A} = \rho_{\text{ion}}\mathbf{g} + eN_{\text{e}}\mathbf{E} - \nabla(P_{\text{ion}} + P_{\text{e}}) + \rho_{\text{ion}}\nu_{\text{ion/n}}\mathbf{u}, \tag{6.48}$$

ρ_{ion} is the ion mass density, $\mathbf{g}$ is the Earth's gravity vector, e is the electron charge, $\mathbf{E}$ is the externally generated (magnetospheric) electric field, P_{ion} and P_{e} are the ion and electron pressures, respectively, $\nu_{\text{ion/n}}$ is the ion-neutral collision frequency, $\mathbf{u}$ is the bulk neutral velocity, $\mathbf{B}$ is the magnetic field, $B \triangleq |\mathbf{B}|$ is the magnitude of the magnetic field, $\mathbf{b} \triangleq \mathbf{B}/B$ is the direction of the magnetic field, and $\mathbf{A}_\perp$ is the component of $\mathbf{A}$ perpendicular to $\mathbf{B}$. Finally, the ion temperature T_{ion} is given by

$$T_{\text{ion}} = \frac{aT_{\text{e}} + bT_{\text{n}} + \frac{3}{2}\rho_{\text{n}}\frac{M_{\text{n}}}{M_{\text{ion}}}Q_{Jh}}{a+b}, \tag{6.49}$$

where a and b are weighted sums of densities given by

$$a = 15\times 3.2\text{e(-8)}\frac{N_{\text{e}}}{T_{\text{e}}^{1.5}}(N_{\text{O}^+} + 0.5N_{\text{O}_2^+} + 0.53N_{\text{NO}^+}), \tag{6.50}$$

$$\begin{aligned} b = {} & \left\{6.6\text{e(-14)}N_{\text{N}_2} + 5.8\text{e(-14)}N_{\text{O}_2} + 0.21\text{e(-14)}N_{\text{O}}(2T_{\text{n}})^{\frac{1}{2}}\right\}N_{\text{O}^+} \\ & + \left\{5.9\text{e(-14)}N_{\text{N}_2} + 5.45\text{e(-14)}N_{\text{O}_2} + 4.5\text{e(-14)}N_{\text{O}}\right\}N_{\text{NO}^+} \\ & + \left\{5.8\text{e(-14)}N_{\text{N}_2} + 0.14\text{e(-14)}N_{\text{O}_2}T_{\text{n}}^{\frac{1}{2}} + 4.4\text{e(-14)}N_{\text{O}}\right\}N_{\text{O}_2}^+, \end{aligned} \tag{6.51}$$

T_{e} is the electron temperature, T_{n} is the neutral temperature, ρ_{n} is the mass density of the neutral species given by

$$\rho_{\text{n}} = M_{\text{O}}N_{\text{O}} + M_{\text{O}_2}N_{\text{O}_2} + M_{\text{N}_2}N_{\text{N}_2} + M_{\text{N}}N_{\text{N}}, \tag{6.52}$$

where M_{S} is the mass of the species S, M_{n} is the mean mass of the neutral species, M_{ion} is the mean mass of the ions, and Q_{Jh} is the Joule heating [105].

6.6 Approximating the Process Noise Covariance Q

The disturbance covariance Q, which determines the range of UKF perturbations, is a critical parameter for strongly nonlinear systems such as GITM. An inappropriate value of Q yields poor data assimilation performance including instability. Filter performance can be improved by taking Q to be proportional to the error covariance, a technique known as covariance inflation [106–108].

For GITM, we approximate Q based on the measurement update time step, which is much longer than the GITM integration time step. Basically, we assume that error-covariance propagation is dominated by the disturbance rather than the system dynamics during the measurement-update time interval, which implies that the error-covariance can be frozen between measurement updates.

We estimate Q based on the solar EUV irradiation $F_{10.7}$, which is one of several inputs to GITM. $F_{10.7}$ is a main GITM driver during calm periods, that is, when there are few geomagnetic storms. Since $F_{10.7}$ is largely unknown in the sense that only its daily average is known [109, 110], its variation is appropriately represented by w_k. Consequently, the process noise or disturbance input of GITM is assumed to come from the random variation of $F_{10.7}$.

6.6.1 Estimating the Covariance Q

At time step k, consider GITM with mean input $\overline{\mu}_k$, which represents mean value of $F_{10.7}$. The resulting state $\overline{x}_k$ propagates according to

$$\overline{x}_{k+N} = f^{(N)}(\overline{x}_k, \overline{\mu}_k), \tag{6.53}$$

where N is the number of steps in a sample period and $f^{(N)}(\cdot)$ is the mapping from time k to $k+N$.

Now consider GITM with inputs $\mu_{k,i} = \overline{\mu}_k + \Delta\mu_{k,i},\ i = 1, \ldots, m$, where m is number of samples for Monte Carlo simulation, $\mu_{k,i}$ is $F_{10.7}$, $\overline{\mu}_k$ is the mean $F_{10.7}$, and $\Delta\mu_{k,i}$ is the deviation from the mean. The mean of $\Delta\mu_{k,i}$ is zero, and the corresponding standard deviation is chosen to be 3 times larger than the standard deviation of the daily $F_{10.7}$ variations. Hence

$$x_{k+N,i} = f^{(N)}(\overline{x}_k, \overline{\mu}_k + \Delta\mu_{k,i}) \tag{6.54}$$

$$\approx f^{(N)}(\overline{x}_k, \overline{\mu}_k) + \Delta\mu_{k,i}\eta_k, \tag{6.55}$$

where $\eta_k \triangleq \frac{\partial f^{(N)}}{\partial \mu_k}(\overline{x}_k, \overline{\mu}_k) \in \mathbb{R}^n$. In (6.55), the term $\Delta\mu_{k,i}\eta_k$ serves as the disturbance input. Using (6.53) and (6.55), we obtain

$$\Delta\mu_{k,i}\eta_k \approx x_{k+N,i} - f^{(N)}(\overline{x}_k, \overline{\mu}_k) \tag{6.56}$$

$$= x_{k+N,i} - \overline{x}_{k+N}. \tag{6.57}$$

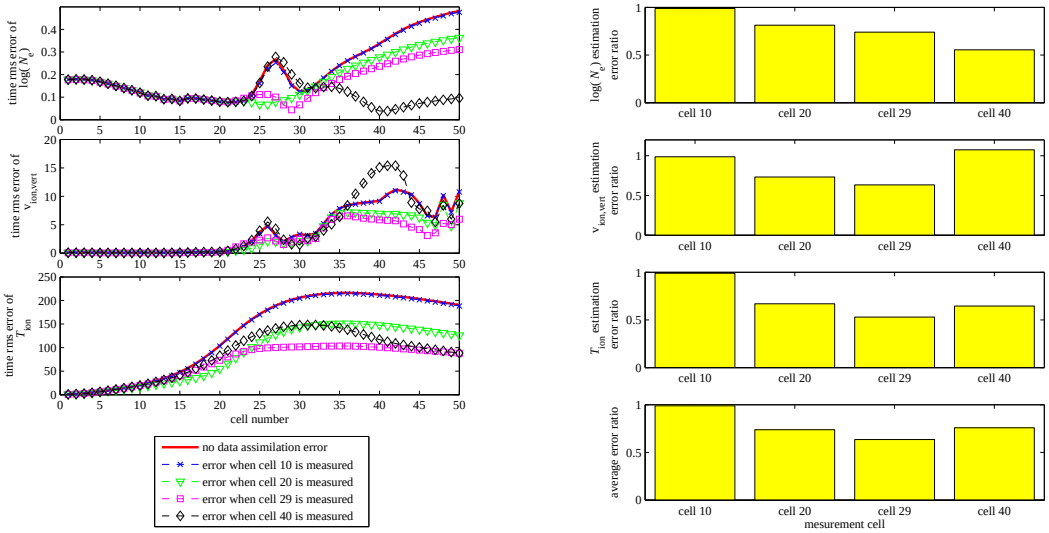
Then, the process noise covariance $\tilde{Q}_k$ can be approximated as

$$\tilde{Q}_k \approx \frac{1}{m}\sum_{i=1}^{m}(\eta_k\Delta\mu_{k,i})(\eta_k\Delta\mu_{k,i})^{\mathrm{T}}. \tag{6.58}$$

To obtain a constant value of Q, we take Q to be the diagonal part of $\tilde{Q}_k$, where k is chosen such that $\mathrm{tr}\tilde{Q}_k$ achieves its maximum value over the simulation interval.

6.7 Observability of GITM for Data Assimilation

We assess the observability of GITM by changing the measurement locations and quantities used in data assimilation performed by UKF. We can thus select measurement locations and quantities to optimize estimation accuracy. For this study, we use a 1D vertical GITM model with 50 grid cells covering 100 km to 750 km in altitude at the location of Millstone Hill, MA, USA, where the Haystack Observatory is located [http://www.haystack.mit.edu/].



(a) (b)

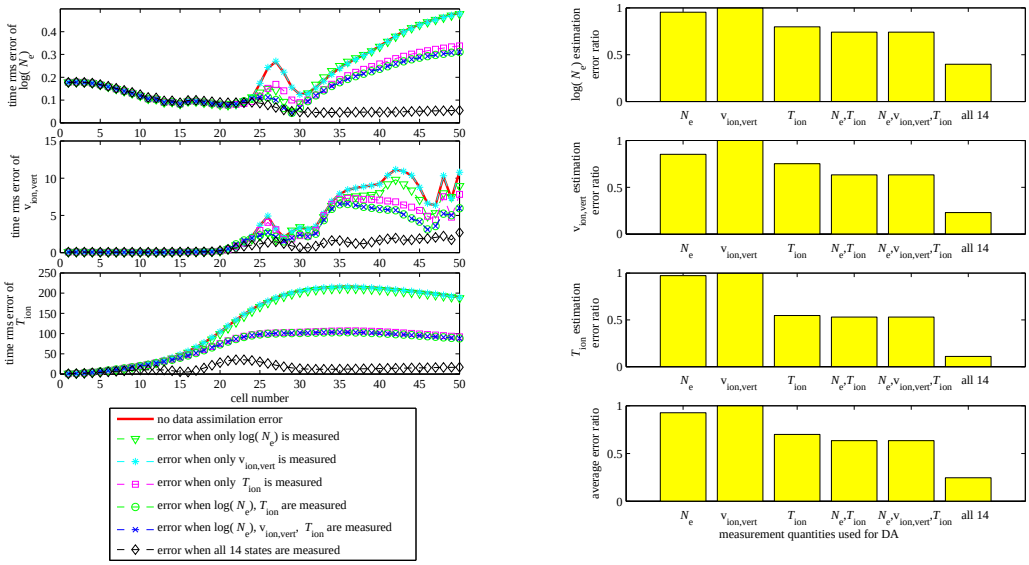
Figure 6.2: Effects of measurement locations on the observability. (a) Cell number versus RMS errors. (b) Ratio of sum of data assimilation RMS errors to the sum of data-free simulation errors. The errors are shown with respect to the measurement locations (cell 10, 20, 29, and 40)

6.7.1 Effect of Measurement Locations on Observability

To determine the effects of various measurement locations, we change the measurement cell from the lower altitudes to the higher altitudes with three measurement quantities, that is, the logarithm $\log(N_e)$ of the number density of electrons, the vertical ion velocity $v_{ion,vert}$, and the ion temperature T_{ion}.

The problem objective is to estimate $\log(N_e)$, $v_{ion,vert}$, and T_{ion} in all 50 cells using measurements of $\log(N_e)$, $v_{ion,vert}$, and T_{ion} in either of the cells 10, 20, 29, or 40, which correspond to 118 km, 187 km, 315 km, and 512 km in altitude, respectively. We assess the observability of each case by comparing the time-averaged RMS estimation errors.

Figure 6.2(a) shows the spatial distribution of the RMS errors with respect to various measurement locations, while, for each measurement location, 6.2(b) shows the ratio of the sum of data assimilation RMS errors to the sum of RMS errors of



(a) (b)

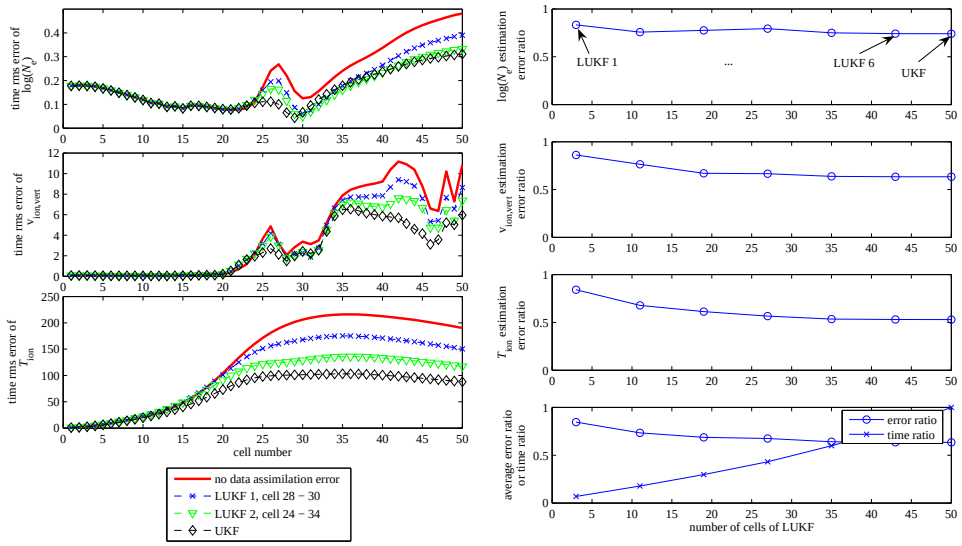
Figure 6.3: Effects of measurement quantity combinations on the observability. (a) Cell number versus RMS errors. (b) Ratio of sum of data assimilation RMS errors to the sum of data-free simulation errors. The errors are shown with respect to various combinations of measurements obtained from cell 29.

data-free simulation with mean value of $F_{10.7}$. By equally weighting the errors of the estimated quantities, we can see from the lowest plot of Figure 6.2(b) that the overall errors are the smallest for measurements taken in cell 29.

6.7.2 Effect of Measurement Quantities on Observability

We now consider all 7 combinations of the available measurement quantities $\log(N_e)$, $v_{ion,vert}$, and T_{ion} while fixing the measurement location at cell 29. We can thus assess the contribution of each measurement or combination of measurements to estimation accuracy.

Figure 6.3(a) shows the spatial distribution of the RMS errors for various combinations of the measurements, while Figure 6.3(b) shows for each case the ratio of the sum of data assimilation RMS errors to the sum of no-data-assimilation errors, that is, when data are not used and the mean value of $F_{10.7}$ is assumed.



(a) (b)

Figure 6.4: [LUKF performance. (a) Cell number versus RMS errors. (b) Ratio of sum of data assimilation RMS errors to the sum of no-data-assimilation errors. The LUKF performance for regions of various sizes are compared with the performance of UKF. All cases use measurements of log number density of electron and ion temperature of cell 29. Spatial regions of LUKF's are, LUKF 1 : cell 28 - 30 (3 cells), LUKF 2 : cell 24 - 34 (11 cells), LUKF 3 : cell 24 - 42 (19 cells), LUKF 4 : cell 24 - 50 (27 cells), LUKF 5 : cell 16 - 50 (35 cells), LUKF 6 : cell 8 - 50(43 cells), Full UKF : cell 1 - 50 (50 cells).

Figure 6.3 shows that the measurements of the vertical component of the ion velocity $\mathrm{v}_{\mathrm{ion,vert}}$ have a negligible effect on the data assimilation accuracy. Therefore, we can exclude $\mathrm{v}_{\mathrm{ion,vert}}$ from measurements for data assimilation without a noticeable loss of accuracy. The cases of $N_{\mathrm{e}}, \mathrm{v}_{\mathrm{ion,vert}}$ and $\mathrm{v}_{\mathrm{ion,vert}}, T_{\mathrm{ion}}$ are similar to the the cases of N_{e} measurement and T_{ion} measurement, respectively.

Finally, we perform data assimilation using all 14 states in cell 29, which provides the best accuracy at all altitudes.

6.8 LUKF Performance

From the previous section, it is shown that the measurements of $N_{\mathrm{e}}, T_{\mathrm{ion}}$ in cell 29 are effective for data assimilation. Therefore, it is expected that we can obtain the most effective LUKF performance when we choose the LUKF region to include

cell 29 and use measurements from cell 29.

Figure 6.4 compares the UKF and LUKF for localized region of various sizes. The estimation error decreases while the size of the LUKF region increases. Figure 6.4(b) shows that the errors of LUKF with the number of cells greater than 11 do not change appreciably, whereas the computation time increases rapidly.

The accuracy of the LUKF estimators with the lower number of LUKF cells is quite close to the accuracy of UKF because of the highly coupled upper cells of vertical 1D GITM. In other words, the ionosphere-thermosphere is highly observable at higher altitudes. Consequently, LUKF can be effectively applied to this kind of example.

6.9 Conclusion

We used a localized, sampled-data update scheme with frozen-intersample error covariance to reduce the computational complexity of the vertical 1D GITM data assimilation based on the unscented Kalman filter. We performed the numerical studies to obtain effective measurement locations and quantities for the sampled-data UKF, and then applied the sampled-data LUKF. The sampled-data LUKF with a small local region showed good estimation accuracy in much shorter computation time for data assimilation on the highly coupled vertical 1D GITM.

CHAPTER VII

Ensemble-On-Demand Kalman Filter for Large-Scale Systems with Time-Sparse Measurements

7.1 Introduction

State estimation for spatially distributed systems typically entails nonlinear, high-dimensional dynamics. For these applications, state estimation is known in practice as data assimilation. Applications range from weather forecasting, to oceanography, to structural dynamics [111–113].

Data assimilation methods use variations of the basic formalism of the classical Kalman filter. The most popular methods replace Riccati-equation-based error covariance propagation of the classical filter with an ensemble of models that approximate the second moment of the error covariance, which is subsequently used to determine a data injection gain. Two such methods are the ensemble Kalman filter (EnKF) [32], which is based on stochastically sampled drivers, and the unscented Kalman filter (UKF) [33], which is based on deterministically determined drivers for an ensemble of $2n+1$ members, where n is the number of states. In the case of linear systems, UKF exactly reproduces the results of the classical Kalman filter.

EnKF and UKF have three main advantages over classical techniques. First, they are often effective for nonlinear systems when the extended Kalman filter is not effective. Second, even when the extended Kalman filter is effective, EnKF and UKF

obviate the need to linearize the system; in fact, EnKF and UKF can be based on a computer program without requiring an explicit mathematical description of the system dynamics. Finally, EnKF and UKF are parallelizable since each ensemble member can be updated independently.

Many of the EnKF or UKF applications of interest arise from extremely high-order dynamics. In particular, we are interested in the global ionosphere-thermosphere model (GITM) [95], whose 10^6 states require a several-hundred-node computing cluster for real-time simulation. Real-time data assimilation based on UKF would require several million nodes, which is not feasible in the foreseeable future.

For very large scale systems, EnKF is distinct from UKF in that the number of ensemble members is not specified. However, useful guidance for the appropriate size of the EnKF ensemble based on linearized analysis is given in [114]. This analysis plays a key role in the present paper, as explained below.

In the present paper we are motivated by the need to perform data assimilation on a system such as GITM, where propagation of an ensemble throughout the data assimilation process is computationally prohibitive. In particular, as is often the case in practice, we assume that the available measurements are time sparse, that is, occur infrequently. When measurements are available at every time step, UKF methods for systems with underlying continuous-time dynamics are given in [115]. However, these methods are prohibitive for large scale systems, and are not needed for systems in which the underlying dynamics are given in time-discretized form.

Under the assumption of time-sparse measurements, we avoid propagating the ensemble members for all of the time steps by creating an ensemble of models only when a new measurement is made available. We then propagate this ensemble into the future, thereby generating an error-covariance matrix, which, in turn, is used to

create a data injection gain, which, finally, is used to assimilate the measurements at the time step at which the measurements became available. Once the measurements are assimilated, only a single simulated model is updated until new measurements become available. We call this algorithm the ensemble-on-demand Kalman filter (EnODKF).

EnODKF is suboptimal since the past history of the error covariance is lost each time the ensemble is collapsed and thus disbanded. However, the computational advantages of not updating the complete ensemble throughout the process can facilitate data assimilation in applications that would otherwise be prohibitive.

The goal of the present paper is to present EnODKF and numerically investigate its properties within the context of linear systems. Nonlinear applications are readily addressed, but are deferred to future work. A key element of our investigations is the analysis of ensemble size based on the work of [114]. We provide a self-contained proof of a result given in [114] that provides guidance on the size of the ensemble needed to accurately estimate the error covariance. We use this guidance within the context of EnODKF, and demonstrate the performance of EnODKF for representative examples, specifically, a lumped vibration problem and a heat flow problem.

7.2 Ensemble Kalman Filter (EnKF)

Consider the discrete-time nonlinear system with dynamics

$$x_{k+1} = f(x_k, u_k, k) + w_k,\ k = 0, 1, 2, \ldots, \tag{7.1}$$

and measurements

$$y_k = h(x_k, k) + v_k,\ k \in \mathcal{K}_{\mathrm{d}}, \tag{7.2}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, $y_k \in \mathbb{R}^p$, and $\mathcal{K}_{\mathrm{d}}$ denotes the set of time steps at which

measurements y_k are available. The input u_k is assumed to be known, and $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^p$ are uncorrelated zero-mean white noise processes with covariances Q_k and R_k, respectively. We assume that R_k is positive definite.

Equation (7.2) denotes that measurements are not available at every time step. When data are not available, the ensemble members are updated by means of a forecast step only. However, when data are available, the ensemble members are updated by both a data assimilation step and a forecast step. We now summarize the steps of the ensemble Kalman filter. For an ensemble consisting of q members at the kth step, EnKF is given by the following procedure:

For $k \notin \mathcal{K}_{\mathrm{d}}$:

Forecast Step

$$x_{k+1}^{\mathrm{f},i} = f(x_k^{\mathrm{f},i}, u_k, k) + w_k^i,\ i = 1, \ldots, q, \tag{7.3}$$

$$x_{k+1}^{\mathrm{f}} = \frac{1}{q}\sum_{i=1}^{q} x_{k+1}^{\mathrm{f},i}. \tag{7.4}$$

For $k \in \mathcal{K}_{\mathrm{d}}$:

Data Assimilation Step

$$E_{x,k}^{\mathrm{f}} \triangleq \left[x_k^{\mathrm{f},1} - x_k^{\mathrm{f}}\ \cdots\ x_k^{\mathrm{f},q} - x_k^{\mathrm{f}}\right], \tag{7.5}$$

$$E_{y,k}^{\mathrm{f}} \triangleq \left[y_k^{\mathrm{f},1} - y_k^{\mathrm{f}}\ \cdots\ y_k^{\mathrm{f},q} - y_k^{\mathrm{f}}\right], \tag{7.6}$$

$$P_{xy,k}^{\mathrm{f}} = \frac{1}{q-1} E_{x,k}^{\mathrm{f}} (E_{y,k}^{\mathrm{f}})^{\mathrm{T}}, \tag{7.7}$$

$$P_{yy,k}^{\mathrm{f}} = \frac{1}{q-1} E_{y,k}^{\mathrm{f}} (E_{y,k}^{\mathrm{f}})^{\mathrm{T}} \tag{7.8}$$

$$K_k = P_{xy,k}^{\mathrm{f}} (P_{yy,k}^{\mathrm{f}})^{-1}, \tag{7.9}$$

$$x_k^{\mathrm{da},i} = x_k^{\mathrm{f},i} + K_k\left(y_k - h(x_k^{\mathrm{f},i}, k) + v_k^i\right),\ i = 1, \ldots, q, \tag{7.10}$$

$$x_k^{\mathrm{da}} = \frac{1}{q}\sum_{i=1}^{q} x_k^{\mathrm{da},i}. \tag{7.11}$$

Forecast Step

$$x_{k+1}^{\mathrm{f},i} = f(x_k^{\mathrm{da},i}, u_k, k) + w_k^i, \;\; i = 1, \ldots, q, \tag{7.12}$$

$$x_{k+1}^{\mathrm{f}} = \frac{1}{q} \sum_{i=1}^{q} x_{k+1}^{\mathrm{f},i}. \tag{7.13}$$

To reproduce the process noise statistics, the noise term w_k^i, which drives the i^{th} ensemble member, is generated deterministically or is sampled from a normal distribution with mean zero and covariance Q_k. Likewise, v_k^i is sampled from a normal distribution with mean zero and covariance R_k and added to the residual $y_k - h(x_k^{\mathrm{f},i})$ in order to reproduce the measurement noise statistics.

Figure 7.1 illustrates EnKF. Each ensemble member is updated by time-sparse measurement data, and is propagated independently when data are not available.

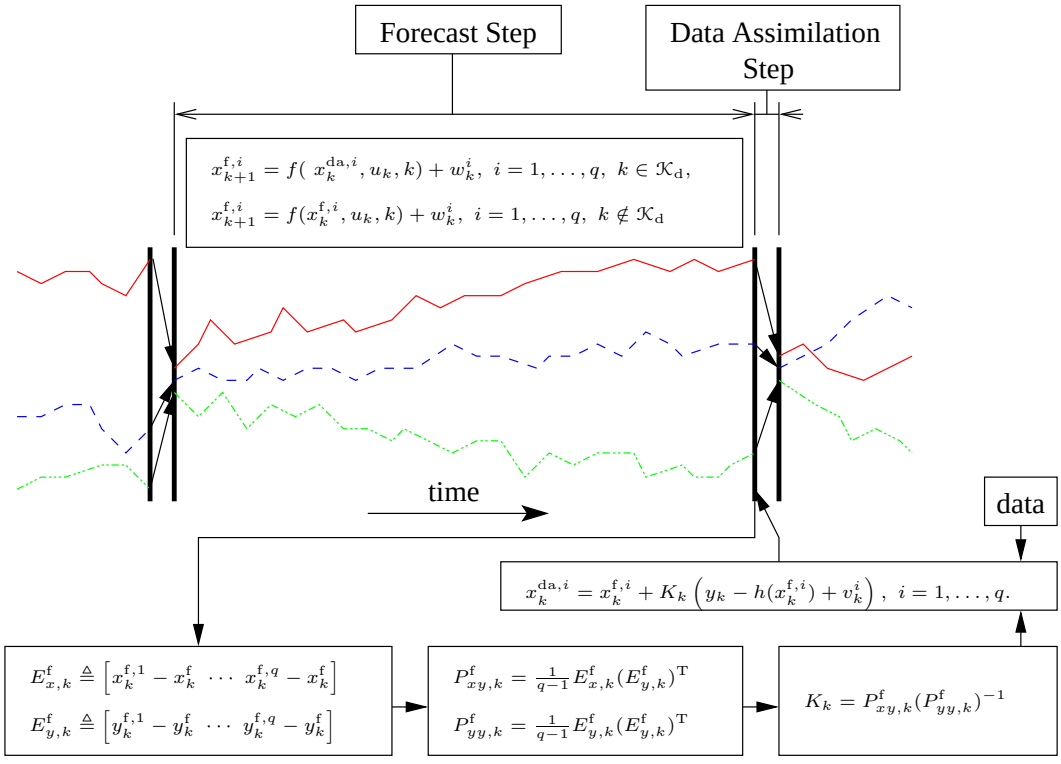


Figure 7.1: Diagram of the ensemble Kalman filter.

7.3 Ensemble Size for Linear Systems

The accuracy of EnKF generally improves as the number of ensemble members is increased. However, a large number of ensembles may be computationally intractable in terms of computation time and memory. Therefore, it is necessary to determine the minimum ensemble size that can adequately approximate the mean of the states and the error covariance. We now examine the required number of ensemble members for the case of linear dynamics. Specifically, we consider the linear system

$$x_{k+1} = A_k x_k + w_k, \tag{7.14}$$

where $x_k \in \mathbb{R}^n$, $A_k \in \mathbb{R}^{n\times n}$, and $w_k \in \mathbb{R}^n$ is a random disturbance with mean zero and covariance Q_k. The following lemma is needed.

Lemma 7.3.1. *Let* $\mathcal{S}_1 \subseteq \mathbb{R}^l$ *and* $\mathcal{S}_2 \subseteq \mathbb{R}^l$ *be subspaces, and assume that*

$$\dim\, \mathcal{S}_1 + \dim\, \mathcal{S}_2 \leq l, \tag{7.15}$$

where dim *denotes dimension. Then there exists an orthogonal matrix* $S \in \mathbb{R}^{l\times l}$ *such that*

$$S\mathcal{S}_2 \subseteq \mathcal{S}_1^{\perp}. \tag{7.16}$$

Proof. Let $n_1 = \dim \mathcal{S}_1$ and $n_2 = \dim \mathcal{S}_2$. Let $M_1 \in \mathbb{R}^{l\times n_1}$, $M_2 \in \mathbb{R}^{l\times n_2}$ be matrices whose columns are an orthonormal basis for $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Next, let $M_1^{\mathrm{c}} \in \mathbb{R}^{l\times(l-n_1)}$ be a matrix composed of $l-n_1$ orthonormal vectors that are also orthogonal to each column vector of M_1, and let $M_2^{\mathrm{c}} \in \mathbb{R}^{l\times(l-n_2)}$ be defined similarly. Now define $S \triangleq [\, M_1^{\mathrm{c}}\ M_1][\, M_2\ M_2^{\mathrm{c}}]^{-1} \in \mathbb{R}^{l\times l}$. Hence $[\, SM_2\ SM_2^{\mathrm{c}}] = [\, M_1^{\mathrm{c}}\ M_1]$. Since $l - n_1 \geq n_2$, it follows that $S\mathcal{S}_1 = \mathcal{R}(SM_2) \subseteq \mathcal{R}(M_1^{\mathrm{c}}) = \mathcal{S}_1^{\perp}$, where $\mathcal{R}$ denotes range. □

The following result is stated without proof in [114].

Fact 7.3.1. *Let $x_k^{\mathrm{f},i}, \ldots, x_k^{\mathrm{f},q} \in \mathbb{R}^n$, and define*

$$H_k \triangleq [A_k x_k^{\mathrm{f},i} - A_k x_k^{\mathrm{f}},\ \ldots\ , A_k x_k^{\mathrm{f},q} - A_k x_k^{\mathrm{f}}] \in \mathbb{R}^{n \times q}, \tag{7.17}$$

where $x_k^{\mathrm{f}} \triangleq \frac{1}{q} \sum_{i=1}^{q} x_k^{\mathrm{f},i}$. Then there exist $w_k^1, \ldots, w_k^q \in \mathbb{R}^n$ such that

$$\sum_{i=1}^{q} w_k^i = 0, \tag{7.18}$$

$$\sum_{i=1}^{q} w_k^i \left(A_k x_k^{\mathrm{f},i} - A_k x_k^{\mathrm{f}} \right)^{\mathrm{T}} = 0, \tag{7.19}$$

$$\frac{1}{q-1} \sum_{i=1}^{q} w_k^i w_k^{i\mathrm{T}} = Q_k, \tag{7.20}$$

if and only if

$$\mathrm{rank}(H_k) + \mathrm{rank}(Q_k) + 1 \leq q. \tag{7.21}$$

Now, let

$$x_{k+1}^{\mathrm{f},i} = A_k x_k^{\mathrm{f},i} + w_k^i,\ i = 1, \ldots, q. \tag{7.22}$$

Then

$$x_{k+1}^{\mathrm{f}} = A_k x_k^{\mathrm{f}}, \tag{7.23}$$

$$P_{xx,k+1}^{\mathrm{f}} = A_k P_{xx,k}^{\mathrm{f}} A_k^{\mathrm{T}} + Q_k, \tag{7.24}$$

where $x_{k+1}^{\mathrm{f}} \triangleq \frac{1}{q} \sum_{i=1}^{q} x_{k+1}^{\mathrm{f},i}$, $P_{xx,k}^{\mathrm{f}} \triangleq \frac{1}{q-1} \sum_{i=1}^{q} \left(x_k^{\mathrm{f},i} - x_k^{\mathrm{f}} \right) \left(x_k^{\mathrm{f},i} - x_k^{\mathrm{f}} \right)^{\mathrm{T}}$, and $P_{xx,k+1}^{\mathrm{f}} \triangleq \frac{1}{q-1} \sum_{i=1}^{q} \left(x_{k+1}^{\mathrm{f},i} - x_{k+1}^{\mathrm{f}} \right) \left(x_{k+1}^{\mathrm{f},i} - x_{k+1}^{\mathrm{f}} \right)^{\mathrm{T}}$.

Proof. Defining

$$W_k \triangleq [w_k^1\ \cdots\ w_k^q] \in \mathbb{R}^{n \times q}, \tag{7.25}$$

(7.18)-(7.20) can be written as

$$\begin{bmatrix} 1_{1\times q} \\ H_k \\ W_k \end{bmatrix} W_k^{\mathrm{T}} = \begin{bmatrix} 0 \\ 0 \\ (q-1)Q_k \end{bmatrix}, \tag{7.26}$$

where $1_{1\times q}$ is the $1 \times q$ ones matrix. Letting $\mathrm{col}_j(M)$ denote the j^{th} column of M, (7.26) can be written as

$$\begin{bmatrix} 1_{1\times q} \\ H_k \\ W_k \end{bmatrix} \mathrm{col}_j(W_k^{\mathrm{T}}) = \begin{bmatrix} 0 \\ 0 \\ (q-1)\ \mathrm{col}_j(Q_k) \end{bmatrix},\ j = 1, \ldots, n. \tag{7.27}$$

To prove necessity, note that, using (7.27), Theorem 2.6.3 in [47] implies that, for all $j = 1, \ldots, n$,

$$\mathrm{rank} \begin{bmatrix} 1_{1\times q} \\ H_k \\ W_k \end{bmatrix} = \mathrm{rank} \begin{bmatrix} 1_{1\times q} & 0 \\ H_k & 0 \\ W_k & (q-1)\mathrm{col}_j(Q_k) \end{bmatrix} \leq q. \tag{7.28}$$

Since $H_k 1_{q\times 1} = 0$, it follows that $1_{1\times q}$ is orthogonal to every row of H_k. Therefore,

$$\mathrm{rank} \begin{bmatrix} 1_{1\times q} \\ H_k \end{bmatrix} = \mathrm{rank}(H_k) + 1. \tag{7.29}$$

Furthermore, it follows from (7.27) that every row of W_k is orthogonal to every row of $[1_{1\times q}^{\mathrm{T}}\ H_k^{\mathrm{T}}]^{\mathrm{T}}$. Finally, since $\mathrm{rank}(W_k) = \mathrm{rank}(W_k W_k^{\mathrm{T}}) = \mathrm{rank}(Q_k)$, it follows that

$$\mathrm{rank} \begin{bmatrix} 1_{1\times q} \\ H_k \\ W_k \end{bmatrix} = \mathrm{rank}(H_k) + \mathrm{rank}(Q_k) + 1 \leq q. \tag{7.30}$$

To prove sufficiency, let $\hat{W}_k \in \mathbb{R}^{n\times q}$ be such that $\hat{W}_k\hat{W}_k^{\mathrm{T}} = (q-1)Q_k$, and define the matrix

$$\overline{H}_k \triangleq \left[\begin{array}{c} 1_{1\times q} \\ H_k \end{array}\right] \in \mathbb{R}^{(n+1)\times q}. \tag{7.31}$$

Let $\mathcal{S}_1 \triangleq \mathcal{R}(\overline{H}_k^{\mathrm{T}}) \subseteq \mathbb{R}^q$ and $\mathcal{S}_2 \triangleq \mathcal{R}(W_k^{\mathrm{T}}) \subseteq \mathbb{R}^q$. Since $\dim \mathcal{S}_1 + \dim \mathcal{S}_2 = 1 + \mathrm{rank}(H_k) + \mathrm{rank}(Q_k) \leq q$, Lemma 7.3.1 implies that there exists an orthogonal matrix $S \in \mathbb{R}^{q\times q}$ such that

$$W_k^{\mathrm{T}} = S\hat{W}_k, \tag{7.32}$$

$$\overline{H}_k W_k^{\mathrm{T}} = 0, \tag{7.33}$$

$$W_k W_k^{\mathrm{T}} = \hat{W}_k S^{\mathrm{T}} S \hat{W}_k^{\mathrm{T}} = \hat{W}_k \hat{W}_k^{\mathrm{T}} = (q-1)Q_k. \tag{7.34}$$

Hence (7.26) follows. Finally, (7.23) and (7.24) follow from (7.26). □

Let

$$\mathcal{Q} \triangleq \{q : (7.21) \text{ is satisfied}\}. \tag{7.35}$$

Then, Fact 7.3.1 shows that the minimum number of ensemble members needed to achieve (7.23), (7.24) is

$$q_{\min} \triangleq \min \mathcal{Q}. \tag{7.36}$$

Furthermore, the maximum value of $q_{\min}$ given by (7.36) is $2n+1$, where n is the number of states of the system. This value is the number of ensemble members used by UKF [33]. However, in many cases, H_k and the disturbance covariance Q_k have low rank, which means that the required ensemble size q may be substantially less than $2n+1$.

We now summarize the numerical algorithm given in [114] for generating $w_k^1, \ldots, w_k^q$ that satisfy (7.18)-(7.20). This algorithm is used for the numerical examples in sections 7.7 and 7.8. For $z_j = [z_{j,1} \ \cdots \ z_{j,j}]^{\mathrm{T}} \in \mathbb{R}^j$, define the Householder matrix $\mathcal{H}(z_j) \in \mathbb{R}^{j \times j}$ by

$$\mathcal{H}(z_j) \triangleq I - \frac{1}{1 + |z_{j,j}|} \begin{bmatrix} z_{j,1} \\ \vdots \\ z_{j,j-1} \\ z_{j,j} + \operatorname{sign}(z_{j,j}) \end{bmatrix} \begin{bmatrix} z_{j,1} & \cdots & z_{j,j-1} & z_{j,j} + \operatorname{sign}(z_{j,j}) \end{bmatrix}, \tag{7.37}$$

and let $\hat{\mathcal{H}}(z_j) \in \mathbb{R}^{j \times (j-1)}$ denote $\mathcal{H}(z_j)$ with its last column deleted. Next, define $\Gamma \in \mathbb{R}^{q \times (n+1)}$ by

$$\Gamma \triangleq \overline{H}_k^{\mathrm{T}}, \tag{7.38}$$

and let $h_k \triangleq \operatorname{rank}(H_k)$ and $q_k \triangleq \operatorname{rank}(Q_k)$.

Fact 7.3.2. *Let $q \geq h_k + q_k + 1$ be an integer, and let $\hat{W}_k \in \mathbb{R}^{n \times q}$ satisfy $\hat{W}_k \hat{W}_k^{\mathrm{T}} = (q-1)Q_k$. Let γ be a nonzero column of Γ, and define $z_q \triangleq \gamma / \|\gamma\|_2$. Form Γ_0 by removing γ from Γ. For $i = 0, \ldots, h_k - 1$, let $\gamma_i \in \mathbb{R}^{q-i-1}$ be a nonzero column of $\hat{\mathcal{H}}^{\mathrm{T}}(z_{q-i})\Gamma_i \in \mathbb{R}^{(q-i-1)\times(n-i-1)}$, and define $z_{q-i-1} \triangleq \gamma_i / \|\gamma_i\|_2$. Remove γ_i from $\hat{\mathcal{H}}^{\mathrm{T}}(z_{q-i})\Gamma_i$ and denote the resulting matrix by Γ_{i+1}. Finally, let $\Omega \in \mathbb{R}^{(q-1-h_k)\times q_k}$ satisfy $\Omega^{\mathrm{T}}\Omega = I$. Then*

$$W_k = [w_k^1 \cdots w_k^q] \triangleq \hat{W}_k \hat{\mathcal{H}}(z_q) \cdots \hat{\mathcal{H}}(z_{q-h_k})\Omega \tag{7.39}$$

satisfies (7.18)-(7.20).

7.4 Ensemble Reduction for Linear System Data Assimilation

Consider the linear system

$$x_{k+1} = A_k x_k + w_k \tag{7.40}$$

with measurements

$$y_k = C_k x_k + v_k,\ k \in \mathcal{K}_{\mathrm{d}}, \tag{7.41}$$

where $x_k \in \mathbb{R}^n$, $y_k \in \mathbb{R}^p$, and $\mathcal{K}_{\mathrm{d}}$ denotes the set of time steps at which measurements y_k are available. As in Section 7.2, $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^p$ are uncorrelated zero-mean white noise processes with covariances Q_k and R_k, respectively. We assume that R_k is positive definite. For this linear system, the Kalman filter is given by the following procedure:

For $k \notin \mathcal{K}_{\mathrm{d}}$:

$$x_{k+1}^{\mathrm{f}} = A_k x_k^{\mathrm{f}}, \tag{7.42}$$

$$P_{k+1}^{\mathrm{f}} = A_k P_k^{\mathrm{f}} A_k^{\mathrm{T}} + Q_k. \tag{7.43}$$

For $k \in \mathcal{K}_{\mathrm{d}}$:

Data Assimilation Step

$$K_k = P_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1}, \tag{7.44}$$

$$x_k^{\mathrm{da}} = x_k^{\mathrm{f}} + K_k (y_k - y_k^f), \tag{7.45}$$

$$P_k^{\mathrm{da}} = P_k^{\mathrm{f}} - P_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} C_k P_k^{\mathrm{f}}. \tag{7.46}$$

Forecast Step

$$x_{k+1}^{\mathrm{f}} = A_k x_k^{\mathrm{da}}, \tag{7.47}$$

$$P_{k+1}^{\mathrm{f}} = A_k P_k^{\mathrm{da}} A_k^{\mathrm{T}} + Q_k. \tag{7.48}$$

The following result shows that disturbances that do not affect the observable subspace can be ignored by the data assimilation procedure.

Proposition 7.4.1. *Consider the linear system*

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \begin{bmatrix} A_{1,k} & 0 \\ A_{21,k} & A_{2,k} \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + \begin{bmatrix} w_{1,k} \\ w_{2,k} \end{bmatrix}, \tag{7.49}$$

$$y_k = \begin{bmatrix} C_{1,k} & 0 \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + v_k, \tag{7.50}$$

where $A_{2,k}$ is asymptotically stable. Let $w_k \triangleq \begin{bmatrix} w_{1,k}^{\mathrm{T}} & w_{2,k}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, assume w_k and v_k are uncorrelated, and define

$$Q_k \triangleq \mathcal{E}(w_k w_k^{\mathrm{T}}) = \begin{bmatrix} Q_{1,k} & Q_{12,k} \\ Q_{12,k}^{\mathrm{T}} & Q_{2,k} \end{bmatrix}, \ \hat{Q}_k \triangleq \begin{bmatrix} Q_{1,k} & 0 \\ 0 & 0 \end{bmatrix}. \tag{7.51}$$

Now, let $\hat{x}_{k,Q_k}$ be the state estimate of the Kalman filter that uses Q_k, and let $\hat{x}_{k,\hat{Q}_k}$ be the state estimate of the suboptimal estimator whose gain is obtained by replacing Q_k by $\hat{Q}_k$ in (7.43), (7.48). Define $\mathcal{P}_k \triangleq \mathcal{E}[(x_k - \hat{x}_{k,Q_k})(x_k - \hat{x}_{k,Q_k})^{\mathrm{T}}]$ and $\hat{\mathcal{P}}_k \triangleq \mathcal{E}[(x_k - \hat{x}_{k,\hat{Q}_k})(x_k - \hat{x}_{k,\hat{Q}_k})^{\mathrm{T}}]$, and let the corresponding costs J_{Q_k} of the Kalman filter and $J_{\hat{Q}_k}$ of the suboptimal estimator be

$$J_{Q_k} \triangleq \mathrm{tr}\mathcal{P}_k, \ J_{\hat{Q}_k} \triangleq \mathrm{tr}\hat{\mathcal{P}}_k. \tag{7.52}$$

Assume that the Kalman filter and the suboptimal estimator have same initial conditions and initial error covariance. Then, for all k,

$$J_{Q_k} \leq J_{\hat{Q}_k}. \tag{7.53}$$

Furthermore, if $Q_{12,k} = 0$, then, for all k,

$$J_{Q_k} = J_{\hat{Q}_k}. \tag{7.54}$$

Proof. We denote the gains of the Kalman filter with Q_k and the suboptimal estimator with $\hat{Q}_k$ by K_k and $\hat{K}_k$, respectively, where K_k is given by (7.44) and $\hat{K}_k$ is given by

$$\hat{K}_k = \hat{P}_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k \hat{P}_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1}, \tag{7.55}$$

$$\hat{P}_k^{\mathrm{da}} = \hat{P}_k^{\mathrm{f}} - \hat{P}_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k \hat{P}_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} C_k \hat{P}_k^{\mathrm{f}}, \tag{7.56}$$

$$\hat{P}_{k+1}^{\mathrm{f}} = A_k \hat{P}_k^{\mathrm{da}} A_k^{\mathrm{T}} + \hat{Q}_k. \tag{7.57}$$

Then the error covariance $\mathcal{P}_k$ of the Kalman filter and the pseudo-error covariance $\hat{\mathcal{P}}_k$ of the suboptimal estimator satisfy

$$\mathcal{P}_{k+1} = A_k (I - K_k C_k) \mathcal{P}_k (I - K_k C_k)^{\mathrm{T}} A_k^{\mathrm{T}} + A_k K_k R_k K_k^{\mathrm{T}} A_k^{\mathrm{T}} + Q_k, \tag{7.58}$$

$$\hat{\mathcal{P}}_{k+1} = A_k (I - \hat{K}_k C_k) \hat{\mathcal{P}}_k (I - \hat{K}_k C_k)^{\mathrm{T}} A_k^{\mathrm{T}} + A_k \hat{K}_k R_k \hat{K}_k^{\mathrm{T}} A_k^{\mathrm{T}} + Q_k. \tag{7.59}$$

Subtracting (7.58) from (7.59), adding and subtracting $A_k(I-\hat{K}_k C_k)\mathcal{P}_k(I-\hat{K}_k C_k)^{\mathrm{T}} A_k^{\mathrm{T}}$, and using $K_k = \mathcal{P}_k C_k^{\mathrm{T}} \tilde{R}_k^{-1}$, where $\tilde{R}_k \triangleq C_k \mathcal{P}_k C_k^{\mathrm{T}} + R_k$, yields

$$\begin{aligned}
&\hat{\mathcal{P}}_{k+1} - \mathcal{P}_{k+1} \\
&= A_k (I - \hat{K}_k C_k)(\hat{\mathcal{P}}_k - \mathcal{P}_k)(I - \hat{K}_k C_k)^{\mathrm{T}} A_k^{\mathrm{T}} \\
&\quad + A_k \hat{K}_k \tilde{R}_k \hat{K}_k^{\mathrm{T}} A_k^{\mathrm{T}} + A_k K_k \tilde{R}_k K_k^{\mathrm{T}} A_k^{\mathrm{T}} - A_k^{\mathrm{T}} K_k \tilde{R}_k \hat{K}_k^{\mathrm{T}} A_k^{\mathrm{T}} - A_k \hat{K}_k \tilde{R}_k K_k^{\mathrm{T}} A_k^{\mathrm{T}} \\
&= A_k (I - \hat{K}_k C_k)(\hat{\mathcal{P}}_k - \mathcal{P}_k)(I - \hat{K}_k C_k)^{\mathrm{T}} A_k^{\mathrm{T}} + A_k (\hat{K}_k - K_k) \tilde{R}_k (\hat{K}_k - K_k)^{\mathrm{T}} A_k^{\mathrm{T}} \\
&\geq 0,
\end{aligned} \tag{7.60}$$

which implies (7.53).

Now, assume $Q_{12,k} = 0$. Let P_k^{f} and $\hat{P}_k^{\mathrm{f}}$ denote the forecast-step error covariance and pseudo-error covariance of the Kalman filter with Q_k and the suboptimal estimator with Q_k replaced by $\hat{Q}_k$, respectively. Next, partition P_k^{f} and $\hat{P}_k^{\mathrm{f}}$ according to

(7.49) as

$$P_k^{\mathrm{f}} = \begin{bmatrix} P_{1,k}^{\mathrm{f}} & P_{12,k}^{\mathrm{f}} \\ P_{12,k}^{\mathrm{fT}} & P_{2,k}^{\mathrm{f}} \end{bmatrix}, \ \hat{P}_k^{\mathrm{f}} = \begin{bmatrix} \hat{P}_{1,k}^{\mathrm{f}} & \hat{P}_{12,k}^{\mathrm{f}} \\ \hat{P}_{12,k}^{\mathrm{fT}} & \hat{P}_{2,k}^{\mathrm{f}} \end{bmatrix}. \tag{7.61}$$

Similarly, define and partition P_k^{da} and $\hat{P}_k^{\mathrm{da}}$. Let P_0 denote the initial error covariance, and define the initial forecast step error covariances P_0^{f}, $\hat{P}_0^{\mathrm{f}}$ by

$$P_0^{\mathrm{f}} = \hat{P}_0^{\mathrm{f}} = P_0. \tag{7.62}$$

Now, the gains K_k and $\hat{K}_k$ are given by

$$K_k = \begin{bmatrix} P_{1,k}^{\mathrm{f}} C_{1,k}^{\mathrm{T}} \\ P_{12,k}^{\mathrm{f}} C_{1,k}^{\mathrm{T}} \end{bmatrix} V_k^{-1}, \ \hat{K}_k = \begin{bmatrix} \hat{P}_{1,k}^{\mathrm{f}} C_{1,k}^{\mathrm{T}} \\ \hat{P}_{12,k}^{\mathrm{f}} C_{1,k}^{\mathrm{T}} \end{bmatrix} \hat{V}_k^{-1}, \tag{7.63}$$

where $V_k \triangleq C_{1,k} P_{1,k}^{\mathrm{f}} C_{1,k}^{\mathrm{T}} + R_k$ and $\hat{V}_k \triangleq C_{1,k} \hat{P}_{1,k}^{\mathrm{f}} C_{1,k}^{\mathrm{T}} + R_k$. Using the gains K_k and $\hat{K}_k$, $P_{1,k}^{\mathrm{da}}$, $P_{12,k}^{\mathrm{daT}}$ and $\hat{P}_{1,k}^{\mathrm{da}}$, $\hat{P}_{12,k}^{\mathrm{daT}}$ are given by

$$P_{1,k}^{\mathrm{da}} = P_{1,k}^{\mathrm{f}} - P_{1,k}^{\mathrm{f}} C_{1,k}^{\mathrm{T}} V_k^{-1} C_{1,k} P_{1,k}^{\mathrm{f}}, \ P_{12,k}^{\mathrm{daT}} = P_{12,k}^{\mathrm{fT}} - P_{12,k}^{\mathrm{fT}} C_{1,k}^{\mathrm{T}} V_k^{-1} C_{1,k} P_{1,k}^{\mathrm{f}}, \tag{7.64}$$

$$\hat{P}_{1,k}^{\mathrm{da}} = \hat{P}_{1,k}^{\mathrm{f}} - \hat{P}_{1,k}^{\mathrm{f}} C_{1,k}^{\mathrm{T}} V_k^{-1} C_{1,k} \hat{P}_{1,k}^{\mathrm{f}}, \ \hat{P}_{12,k}^{\mathrm{daT}} = \hat{P}_{12,k}^{\mathrm{fT}} - \hat{P}_{12,k}^{\mathrm{fT}} C_{1,k}^{\mathrm{T}} V_k^{-1} C_{1,k} \hat{P}_{1,k}^{\mathrm{f}}. \tag{7.65}$$

Consequently, P_{k+1}^{f} and $\hat{P}_{k+1}^{\mathrm{f}}$ are given by

$$P_{1,k+1}^{\mathrm{f}} = A_{1,k} P_{1,k}^{\mathrm{da}} A_{1,k}^{\mathrm{T}} + Q_{1,k}, \ P_{12,k+1}^{\mathrm{fT}} = A_{21,k} P_{1,k}^{\mathrm{da}} A_{1,k}^{\mathrm{T}} + A_{2,k} P_{12,k}^{\mathrm{daT}} A_{1,k}^{\mathrm{T}}, \tag{7.66}$$

$$\hat{P}_{1,k+1}^{\mathrm{f}} = A_{1,k} \hat{P}_{1,k}^{\mathrm{da}} A_{1,k}^{\mathrm{T}} + Q_{1,k}, \ \hat{P}_{12,k+1}^{\mathrm{fT}} = A_{21,k} \hat{P}_{1,k}^{\mathrm{da}} A_{1,k}^{\mathrm{T}} + A_{2,k} \hat{P}_{12,k}^{\mathrm{daT}} A_{1,k}^{\mathrm{T}}. \tag{7.67}$$

Hence, $P_{1,k}^{\mathrm{f}} = \hat{P}_{1,k}^{\mathrm{f}}$, $P_{12,k}^{\mathrm{fT}} = \hat{P}_{12,k}^{\mathrm{fT}}$, and $K_k = \hat{K}_k$ for all k, which implies $\hat{x}_{k,Q_k} = \hat{x}_{k,\hat{Q}_k}$ and thus (7.54). □

Proposition 7.4.1 implies that, for EnKF, it is not necessary to generate $w_{2,k}^i$, which does not affect the observable subspace of (A_k, C_k).

Now, for the disturbance $\begin{bmatrix} w_{1,k}^{\mathrm{T}} & 0 \end{bmatrix}^{\mathrm{T}}$, let the corresponding matrices for H_k and Q_k be $\hat{H}_k$ and $\hat{Q}_k$. Then, assuming that $\mathrm{rank}(\hat{H}_k) \leq \mathrm{rank}(H_k)$, the minimum ensemble size needed by the ensemble Kalman filter to satisfy constraints (7.18)-(7.20) can be reduced by $\mathrm{rank}(Q_k) - \mathrm{rank}(\hat{Q}_k)$.

7.5 Ensemble-On-Demand Kalman Filter (EnODKF)

EnKF requires in (7.3) that q ensemble members be updated in parallel at every time step whether or not data are available. When $q_{\min}$ given by (7.36) is large, real-time estimation for acceptable accuracy is computationally expensive. To partially overcome the excessive computational complexity of the ensemble Kalman filter, we consider the *ensemble on-demand Kalman filter* (EnODKF), which propagates the ensemble members over a small number $N_{\rm s}$ of steps only when data are available. EnODKF is described by the following procedure:

For $k \notin \mathcal{K}_{\rm d}$:

Forecast Step

$$x_{k+1}^{\rm f} = f(x_k^{\rm f}, u_k, k). \tag{7.68}$$

For $k \in \mathcal{K}_{\rm d}$:

Data Assimilation Step

$$x_k^{\mathrm{f},i} = x_k^{\mathrm{f}},\ i = 1, \ldots, q,$$

$$x_{k+j}^{\mathrm{f},i} = f(x_{k+j-1}^{\mathrm{f},i}, u_{k+j-1}, k) + w_{k+j-1}^{i},\ i = 1, \ldots, q,\ j = 1, \ldots, N_{\mathrm{s}}, \tag{7.69}$$

$$y_{k+N_{\mathrm{s}}}^{\mathrm{f},i} = h(x_{k+N_{\mathrm{s}}}^{\mathrm{f},i}, k + N_{\mathrm{s}}), \tag{7.70}$$

$$x_{k+N_{\mathrm{s}}}^{\mathrm{f}} = \frac{1}{q}\sum_{i=1}^{q} x_{k+N_{\mathrm{s}}}^{\mathrm{f},i},\ y_{k+N_{\mathrm{s}}}^{\mathrm{f}} = \frac{1}{q}\sum_{i=1}^{q} y_{k+N_{\mathrm{s}}}^{\mathrm{f},i}, \tag{7.71}$$

$$E_{x,k+N_{\mathrm{s}}}^{\mathrm{f}} \triangleq \left[x_{k+N_{\mathrm{s}}}^{\mathrm{f},1} - x_{k+N_{\mathrm{s}}}^{\mathrm{f}} \ \cdots \ x_{k+N_{\mathrm{s}}}^{\mathrm{f},q} - x_{k+N_{\mathrm{s}}}^{\mathrm{f}}\right], \tag{7.72}$$

$$E_{y,k+N_{\mathrm{s}}}^{\mathrm{f}} \triangleq \left[y_{k+N_{\mathrm{s}}}^{\mathrm{f},1} - y_{k+N_{\mathrm{s}}}^{\mathrm{f}} \ \cdots \ y_{k+N_{\mathrm{s}}}^{\mathrm{f},q} - y_{k+N_{\mathrm{s}}}^{\mathrm{f}}\right], \tag{7.73}$$

$$P_{xy,k}^{\mathrm{f}} = \frac{1}{q-1} E_{x,k+N_{\mathrm{s}}}^{\mathrm{f}} (E_{y,k+N_{\mathrm{s}}}^{\mathrm{f}})^{\mathrm{T}}, \tag{7.74}$$

$$P_{yy,k}^{\mathrm{f}} = \frac{1}{q-1} E_{y,k+N_{\mathrm{s}}}^{\mathrm{f}} (E_{y,k+N_{\mathrm{s}}}^{\mathrm{f}})^{\mathrm{T}} + R_{k_{\mathrm{d}}}, \tag{7.75}$$

$$K_k = P_{xy,k}^{\mathrm{f}} (P_{yy,k}^{\mathrm{f}})^{-1}, \tag{7.76}$$

$$x_k^{\mathrm{da}} = x_k^{\mathrm{f}} + K_k \left(y_k - h(x_k^{\mathrm{f}})\right). \tag{7.77}$$

Forecast Step

$$x_{k+1}^{\mathrm{f}} = f(x_k^{\mathrm{da}}, u_k, k). \tag{7.78}$$

Figure 7.2 illustrates EnODKF. Each ensemble member propagates for N_{s} steps when data are available in order to generate an approximate error covariance matrices and data assimilation gain. Then, the states are updated using the available data at the measurement time.

The ensemble size q for EnODKF is ideally chosen such that $q \geq 1 + \mathrm{rank}(H_{k'}) + \mathrm{rank}(Q_{k'})$, where $k \leq k' < k + N_{\mathrm{s}}$ and $k \in \mathcal{K}_{\mathrm{d}}$. Next, the choice of N_{s} is governed by the tradeoff between computation time and accuracy. That is, larger N_{s} ensures better accuracy but increased computation time.

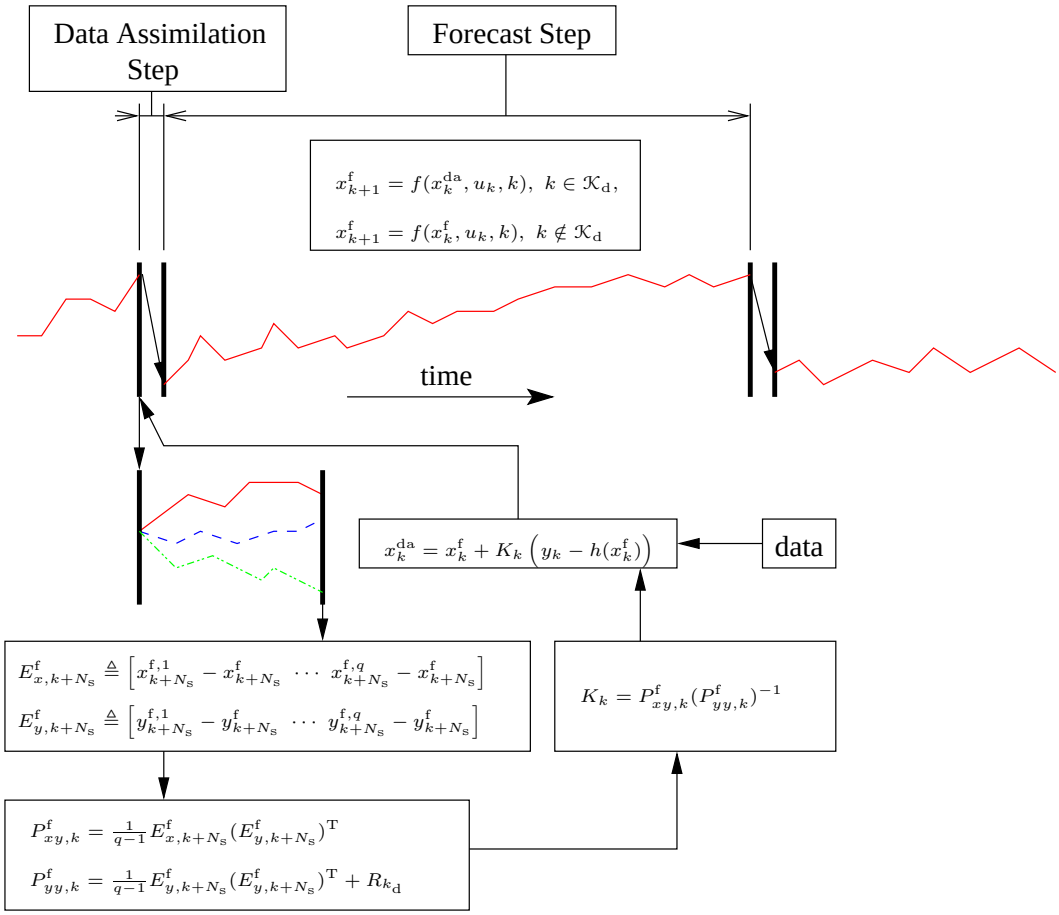


Figure 7.2: Diagram of the ensemble-on-demand Kalman filter.

7.6 Numerical Results

We consider EnKF and EnODKF for the linear system (7.40) with measurements (7.41). As a baseline reference, we also compute estimates using the Kalman filter (7.42)-(7.48). For all simulations, the truth model is the model with stochastic drivers, the no data assimilation (NoDA) model is the model with each driver replaced by its mean value, and the data assimilation (DA) model is the NoDA model with data assimilation using simulated measurements from the truth model simulation.

7.7 Lumped Vibration Example

Consider the mass-spring-dashpot system consisting of 10 masses shown in Figure 7.3. For $i = 1, \ldots, 10$, $m_i = 1.0$, while, for $j = 1, \ldots, 11$, $k_j = 1.0$ and $c_j = 0.05$. We choose low damping coefficients so that each vibration mode is underdamped with damping ratio less than 0.05. The order of the system is $n = 20$.

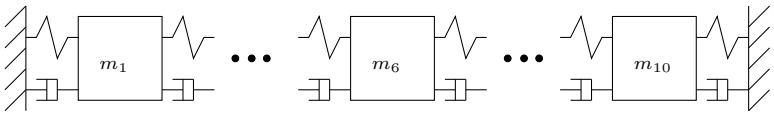


Figure 7.3: Mass-spring-dashpot system.

We consider three kinds of disturbance inputs, specifically, several local disturbances in which an independent force is applied to each mass, one local disturbance in which a single force is applied to the 6th mass, and one global disturbance applied to all of the masses.

We consider two measurement cases involving the velocity of the 6th mass and the velocity of the 5th mass. We assume that measurements are available every $N_{\mathrm{d}} = 4$ steps, that is, $\mathcal{K}_{\mathrm{d}} = \{1, 5, 9, \ldots\}$, and we apply EnODKF with several values of N_{s} to assess EnODKF performance as a function of N_{s}.

First, we examine the minimum number of ensemble members required for EnKF in light of (7.21). In Figure 7.5, we compare the performance of EnKF with stochastic and deterministic disturbances, respectively. The left column of plots corresponds to the case where the number of independent disturbance force inputs is 10. The plots in the center row show the singular values of Q_k in order of descending magnitude. The top row of plots show the singular values of H_k. It can be seen that $\mathrm{rank}(H_k)$ for this case is 20. The bottom row of plots show that the ratio J/J_{opt} of EnKF costs J to the Kalman filter cost J_{opt} fluctuates but decreases with increasing ensemble size.

Specifically, J/J_{opt} for EnKF with deterministic disturbance sampling remains near 1 beyond an ensemble size of about $1+\text{rank}(H_k)+\text{rank}(Q_k)$. Although the profiles of J/J_{opt} of EnKF using a stochastic disturbance fluctuate more than the EnKF with deterministic disturbance sampling, we observe that the expected value of J/J_{opt} with stochastic disturbance sampling converges when ensemble size is greater than $1 + \text{rank}(H_k) + \text{rank}(Q_k)$.

The right column of plots corresponds to the case in which a single global disturbance force is applied to all 10 masses so that $\text{rank}(Q_k) = 1$. In this case, $\text{rank}(H_k)$ is 20 using a small singular-value tolerance for determining rank. However, the singular values of H_k decrease rapidly, which indicates that the effective rank of H_k is less than the first case. The plot of J/J_{opt} has the same trend as the singular value plot of H_k, that is, the value of J/J_{opt} converges faster than the previous cases.

Next, we demonstrate EnODKF. Figure 7.7 shows the results of data assimilation with measurement update period $N_{\text{d}} = 4$, and with $N_{\text{s}} = 1, 2, 3$, and 4. It can be seen that EnODKF has poor performance in (a), (b) and (c), but the fluctuation of J/J_{opt} with respect to ensemble size is less than EnKF. As illustrated by (a), (c), and (d), the number N_{s} of propagation steps has a stronger impact on estimation when the measurements are affected by disturbances from other locations. The EnODKF performance is best for the case in which the system is subject to a single global disturbance as shown in (d).

7.8 Two-Dimensional Heat Conduction Example

Consider the heat conduction in a two-dimensional plate, governed by

$$\frac{\partial T(x,y,t)}{\partial t} = \alpha \left(\frac{\partial^2 T(x,y,t)}{\partial x^2} + \frac{\partial^2 T(x,y,t)}{\partial y^2} \right) + w(x,y,t), \tag{7.79}$$

where $T(x, y, t)$ is the temperature at position (x, y) and time t, $w(x, y, t)$ represents disturbance heat sources or sinks acting on the plate, and α is the heat conduction coefficient. We discretize (7.79) over a spatial grid of size $n_x \times n_y = 20 \times 20$, where n_x and n_y denote the number of grid points in the horizontal and vertical directions, respectively. The random initial conditions $\mathcal{N}(0, I_n)$ are given, and all boundary conditions are free.

We consider five kinds of disturbance inputs, specifically, 1) n_y-independent disturbances to the left boundary edge, 2) a single disturbance to the center of the left boundary edge, 3) a single disturbance to all cells at the left edge (A-type disturbance), 4) two independent sets of disturbances to each half of the left boundary edge where all disturbances within a set are identical (AB-type disturbance), and, 5) finally, two independent sets of disturbances to each half of the left boundary edge and another two independent sets of disturbances to each half of the right boundary edge (ABCD-type disturbance).

We consider five cases of measurements, where single, two, and four measurement points are selected with different distances from the left and the right boundary edges. We assume that measurements are available every $N_{\mathrm{d}} = 6$ steps, and we consider $N_{\mathrm{s}} = 1$ and $N_{\mathrm{s}} = 4$ for the EnODKF. The disturbances and measurements are illustrated in Figure 7.4.

Proposition 7.4.1 shows that we do not need to include in the ensemble Kalman filter the disturbance sources that do not affect the observable subspace of (A_k, C_k). Consequently, the ensemble size needed to achieve acceptable accuracy is less than $1 + \mathrm{rank}(H_k) + \mathrm{rank}(Q_k)$.

To illustrate Proposition 7.4.1, we consider estimation for two-dimensional heat conduction in a square plate composed of two regions that have different heat con-

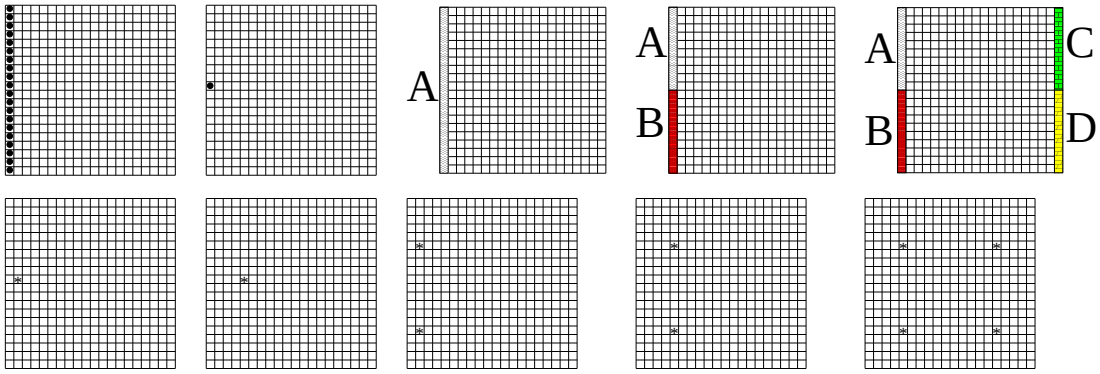


Figure 7.4: Illustration of types of disturbances (upper row) and measurements (lower row) used for 2D heat conduction estimation. The leftmost disturbance in the upper row indicates that 20 independent disturbances act on all of the cells of the left boundary edge.

duction coefficients. We take $\alpha_{\text{small}} = 0.2\alpha_{\text{large}}$ so that the system matrix can be approximated by (7.49). The regions of the different heat conduction coefficients are shown in Figure 7.6(a). Next, let the 2D plate be subject to $n_y = 20$ independent heat sources placed along the left edge of the system and use one point measurement as shown in Figure 7.6(a) and Figure 7.6(b).

We assume that the states of the α_{large} region are observable from the measurement, and are reachable by the 10 independent disturbance sources that are in the α_{large} region, whereas the remaining 10 disturbance sources in the α_{small} region do not affect the α_{large} region due to lower conductivity. Then, we perform EnKF data assimilation with the 10 and 20 independent disturbance sources, respectively, while increasing the number of ensemble members.

It can be seen from Figure 7.6(e) that the errors of EnKF with 20 disturbance sources converge at around an ensemble size of 20, which is less than half of $1 + \text{rank}(H_k) + \text{rank}(Q_k)$ for the entire system. Next, the errors of EnKF with 10 disturbance sources that correspond to the $\hat{Q}_k$ given by (7.51), have the same estimation accuracy as EnKF with full 20 disturbance sources at an ensemble size of around 20, which means that there is no accuracy degradation with the reduced distur-

bance sources since the reduced disturbance sources affect the α_{large} region, while the remaining 10 disturbances have minimal effect on the α_{large} region.

Next, we compare EnKF and EnODKF. The 2D heat conduction system has slow dynamics, and the disturbances are damped out rapidly while passing through the cells. Therefore, EnODKF with $N_{\text{s}} = 1$ works poorly when the measurement point and the disturbance location are different by as few as one cell. This characteristic is illustrated in Figure 7.8, where all EnODKF results are identical to NoDA.

However, as shown in the 1st and the 3rd rows of plots of Figure 7.9, EnODKF with $N_{\text{s}} = 4$ is effective for the cases where the measurement locations are distant from the disturbances by 1.0 length unit, where each square cell is 0.5 length units in width. However, EnODKF fails to work when the measurement location is placed 4.1 and 4.0 length units away from the disturbances as shown in the 2nd and the 4th rows of plots of Figure 7.9, respectively.

7.9 Conclusion

In using EnKF, the main issues are how to perturb the system and how many ensemble members are required for acceptable accuracy. We showed that the ensemble size of EnKF for acceptable accuracy can be guided by the number $1 + \text{rank}(H_k) + \text{rank}(Q_k)$, which is the lower limit for generating disturbance vectors that are orthogonal to the propagated states while maintaining exact disturbance covariance.

It is evident in the linear system examples that we can avoid errors due to sampling errors by using a finite number of exact disturbances. However, the disturbance covariances must be accurately known in order to generate those disturbances. However, in large scale data assimilation, for example, for terrestrial weather, space

weather, and ocean currents, the disturbance covariance is difficult to construct even when the disturbance sources are known. Moreover, when the disturbance sources are not known, we cannot expect to obtain useful results from data assimilation. Therefore, implementation of data assimilation by EnKF involves first identifying disturbance sources and then exciting the system using the identified disturbance sources with the ensemble size guided by the number $1 + \text{rank}(H_k) + \text{rank}(Q_k)$.

For large scale systems, the number $1+\text{rank}(H_k)+\text{rank}(Q_k)$ may be prohibitively large for the available computing resources, and thus the reduction of computational complexity is needed. However, before reducing computational complexity using, for example, SVD, projection of disturbance, or model reduction, unnecessary disturbance sources should be removed. We showed the effectiveness of removing unnecessary disturbances in a 2D heat conduction example with decreased computational burden and no degradation of accuracy.

Finally, we showed that EnODKF is computationally inexpensive but provides acceptable performance for systems under a single global-type disturbance.

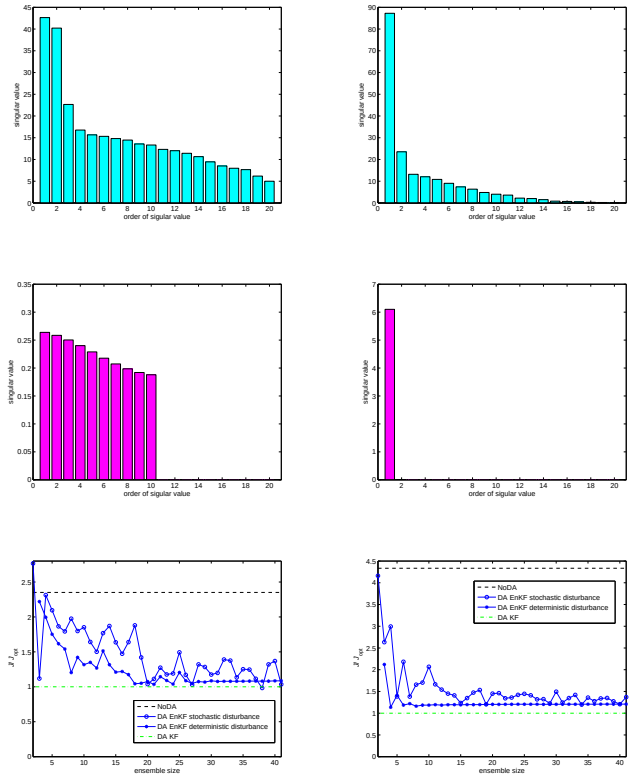
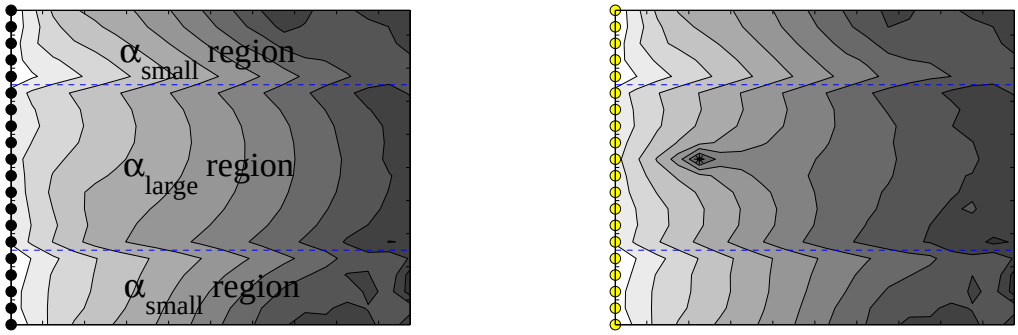
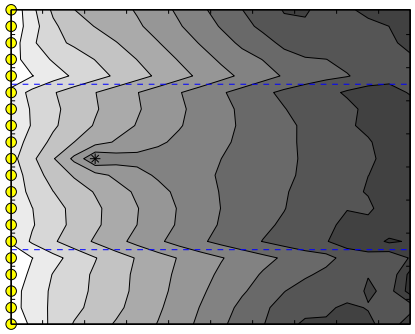


Figure 7.5: Comparison of EnKF with stochastic and deterministic disturbance samplings, respectively. The plots in the left column correspond to the case where the number of independent disturbance force inputs is 10. The plots in the right column correspond to the case in which a single global disturbance force is applied to all 10 masses. The plots in the center row show the singular values of Q_k in order of descending magnitude. The top row of plots show the singular values of H_k. The bottom row of plots show that the ratio J/J_{opt} of EnKF costs J to the Kalman filter cost J_{opt}. This ratio fluctuates but decreases with increasing ensemble size. The plots of J/J_{opt} indicate the same trends as the singular values of H_k. Although the profiles of J/J_{opt} of EnKF using a stochastic disturbance fluctuate more than the EnKF with deterministic disturbance sampling, we observe that the expected value of J/J_{opt} of stochastic-disturbance-EnKF converges when ensemble size is greater than $1 + \text{rank}(H_k) + \text{rank}(Q_k)$.

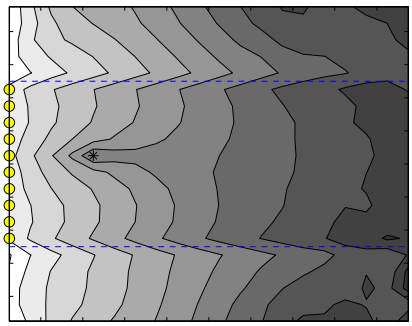


(a)

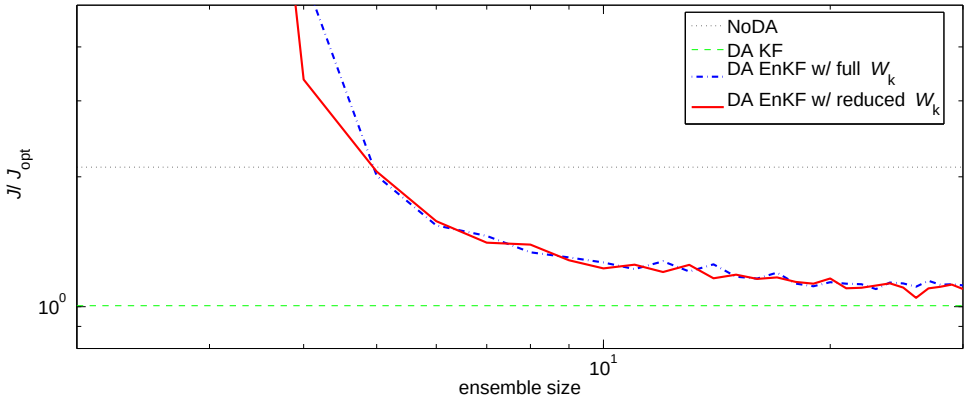
(b)



(c)



(d)



(e)

Figure 7.6: Comparisons of EnKF estimators using full disturbances and reduced disturbances for 2D heat conduction on the square plate composed of regions of two different heat conduction coefficients α_{small} and α_{large}. The α_{small} and α_{large} regions are shown in (a) divided by thin dashed lines, with NoDA rms error distributions. We take $\alpha_{\text{small}} = 0.2\alpha_{\text{large}}$. 20 filled-circles in (a) indicate the 20 independent disturbances in the truth model. (b) is the rms error distribution of the Kalman filter with the measurement at the location marked by $*$. (c) and (d) are the rms error distribution of EnKF estimators at ensemble size 30 with full 20 and 10 disturbances in the α_{large} region, respectively. It can be seen in (e) that the errors of EnKF with full disturbance converge at around the ensemble size 20, which is far less than $1 + \text{rank}(H_k) + \text{rank}(Q_k)$ for the entire system, which is greater than 40. Furthermore, the errors of the EnKF with fewer disturbances yields the same converged estimation accuracy as EnKF with full disturbances at around the ensemble size 20.

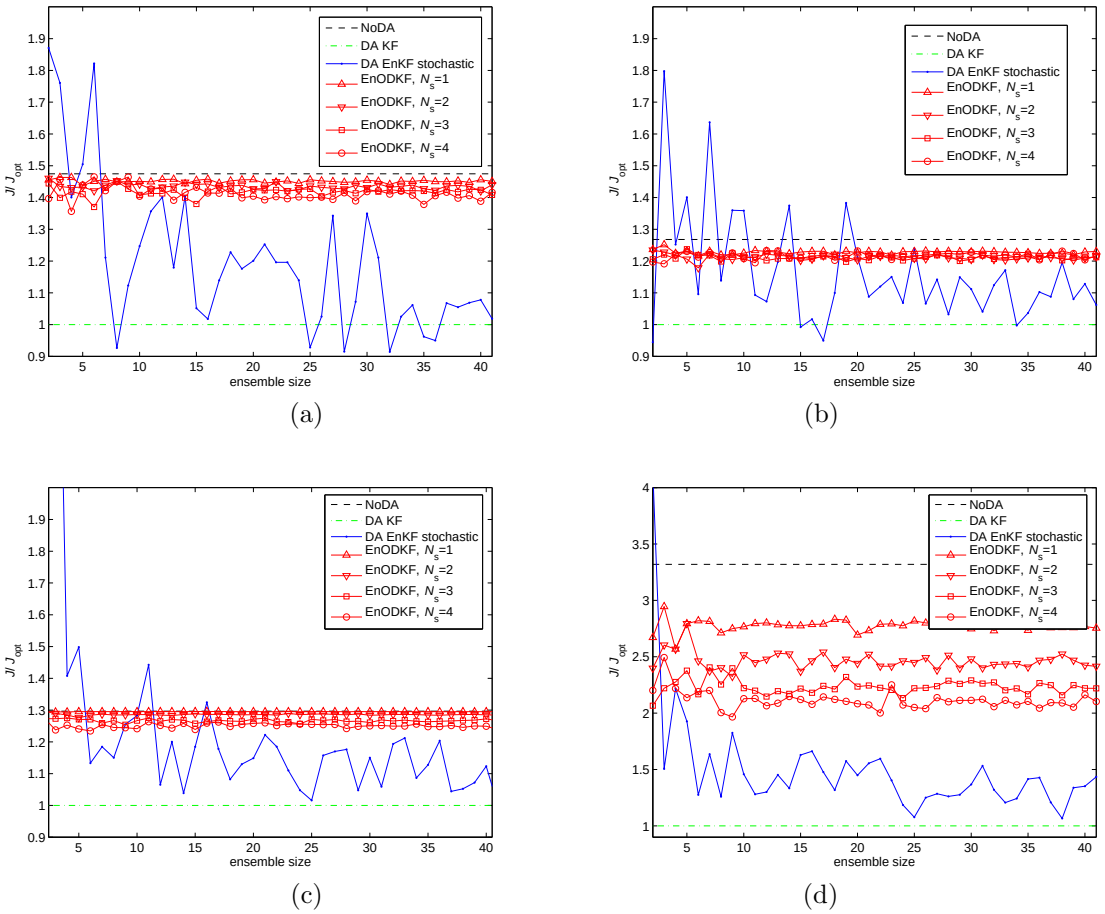


Figure 7.7: Data assimilation cost versus ensemble size for the lumped-vibration example. (a) 10 independent disturbances with measurements of x_{12}, (b) single disturbance to x_{12} with measurements of x_{12}, (c) single disturbance to x_{12} with measurements of x_{10}, (d) single global disturbance to all states with measurements of x_{12}.

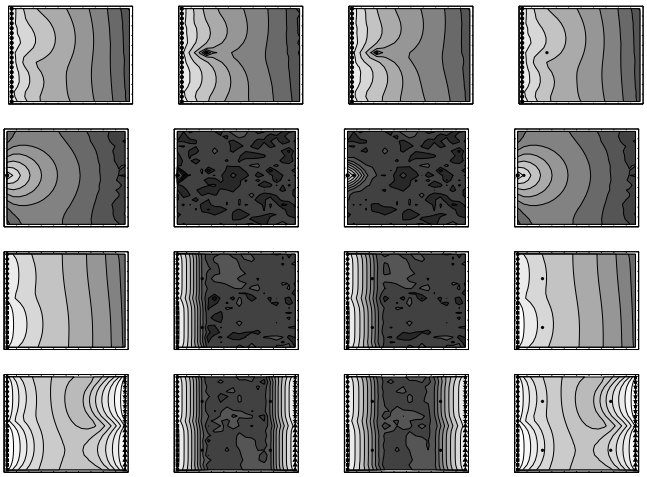


Figure 7.8: 2D heat conduction estimation rms error distribution with $N_{\mathrm{d}} = 6$ and $N_{\mathrm{s}} = 1$. We assume uniform heat conductivity. 1st to 4th columns: rms error distribution of NoDA, Kalman filter, ensemble Kalman filter, and ensemble on-demand Kalman filter, respectively. 1st row: 20 independent disturbances to the left boundary edge, 2nd row: single disturbance to the left boundary edge, 3rd row: AB-type disturbance, 4th row: ABCD-type disturbance. Darker regions around measurement points indicate that the errors are reduced by data assimilation relative to NoDA.

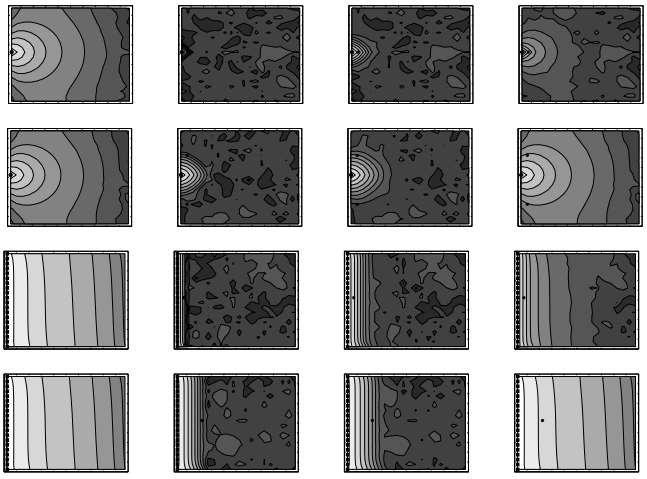


Figure 7.9: 2D heat conduction estimation rms error distribution with $N_{\mathrm{d}} = 6$ and $N_{\mathrm{s}} = 4$. We assume uniform heat conductivity. 1st to 4th columns: rms error distribution of NoDA, Kalman filter, ensemble Kalman filter, and ensemble on-demand Kalman filter, respectively. 1st-2nd rows: single disturbance to the left boundary edge, 3rd-4th rows: A-type disturbance.

CHAPTER VIII

A Numerical Investigation of Data Assimilation Using the Global Ionosphere-Thermosphere Model

8.1 Introduction

Studies of the upper atmosphere have led to an in-depth understanding of the dominant features of the global ionosphere and thermosphere. The incident solar Extreme Ultraviolet (EUV) radiation, which is a highly variable part of the solar spectrum, determines, to a large part, the amount of energy absorbed in the atmosphere. The major neutral species O_2, N_2, and NO absorb this radiation, and the energy is used either to heat the ambient gas or ionize the neutrals. Once ionized, the resulting photoelectrons receive the remaining energy, and in turn pass the energy on to the thermal electrons and ultimately the ions.

Due to the configuration of the Earth's magnetic field, as well as the interaction between the magnetic field and the solar wind, the upper atmosphere has a strong latitudinal dependence. At lower latitudes, the atmosphere is not strongly influenced by these features, and the solar photons are the dominant source of energy. At higher latitudes, however, auroral heating and particle precipitation are often the primary sources of energy to the atmosphere. Still, there exists a fundamental understanding of these processes and their effect on the ionosphere-thermosphere system, especially when transient features are ignored.

Perturbations to this background state of the atmosphere, or space weather, can have dramatic effects on both space- and ground-based technology. For instance, satellites in low-Earth orbit can experience significant changes in velocity when the thermospheric drag on the spacecraft changes. Furthermore, abrupt changes in the total electron content (TEC) of the ionosphere can lead to substantial errors in ground-based Global Positioning System (GPS) measurements. For these reasons, finding methods to improve the accuracy in forecasting the state of the ionosphere-thermosphere system has become an active area of research [116–118].

When attempting to accurately forecast the state of the upper atmosphere, it is important to start with initial states that are as close as possible to the true initial states. This is because preconditioning of the atmosphere has a large effect on the future state of the system. For example, the electron profile in the thermosphere after a solar flare is highly dependent on the O/N^2 ratio before the flare, since this ratio largely determines the production and loss rates for the electrons. Since the upper atmosphere is an externally driven system, simulated results depend strongly on the inputs used to drive the system. Therefore, even if the atmospheric model itself is accurate, the correct solution cannot be obtained if the drivers are uncertain. The best way to ensure, then, that the initial state of the system is as accurate as possible, given uncertainty in the drivers as well as the model, is to use available measurements of the system to correct the model. Data assimilation facilitates the use of point or integral measurements of the upper atmosphere to adjust the approximate system towards reality. By having an understanding of how different parts of the system correlate with each other, this technique allows the simulation to be corrected on a large scale, even if measurements are taken locally.

Various studies have performed ionospheric data assimilation based on measure-

ments of TEC [3–5] and bottom side electron density profiles [6]. TEC measurements are available both from GPS ground stations, as well as from GPS occultation between LEO satellites and GPS receivers, which means that data coverage is continuous and global in nature. The immense amount of TEC data available from these sources makes assimilating TEC measurements extremely practical and useful. More recently, with the availability of precise thermospheric density data from the Champ [7] and Grace [8] experiments, interest has arisen in assimilating thermospheric variables. Thermospheric data assimilation using simulated measurements of thermospheric composition is investigated in [9].

In the present study, data assimilation techniques are used with the Global Ionosphere-Thermosphere Model (GITM) [119] to estimate the ionosphere-thermosphere system under uncertain solar EUV conditions as well as during geomagnetic storm conditions. Preliminary results are presented for uncertain solar EUV conditions using simulated incoherent scatter radar (ISR) measurements of electron number density and ion temperature, as well as simulated satellite measurements of neutral density. In addition, the behavior of the GITM-based data assimilation under geomagnetic storm conditions when using simulated ISR measurements of electron number density, ion temperature, and simulated satellite measurements of neutral density, or using simulated vertical TEC measurements is presented.

8.2 Method

8.2.1 GITM

GITM is a fully parallel three-dimensional (3D) model of the coupled ionosphere-thermosphere system in spherical coordinates. The model solves the conservation equations in altitude coordinates as opposed to pressure coordinates and does not assume hydrostatic equilibrium. Therefore, the vertical momentum equation can

be solved self-consistently, and substantial vertical flows can develop [11]. GITM uses a block-based uniform domain decomposition in the horizontal direction, along with a non-uniform altitude grid, with a resolution of 1/3 scale height. The grid is entirely flexible, so the user is free to change the horizontal resolution from run to run by specifying the number of blocks to use in a given simulation. A consequence of this flexibility is that GITM can be run in 1-dimension (1D) where only a single latitude and longitude are simulated, and horizontal transport and gradients are ignored. Since 1D GITM can be run quickly on a personal computer, this feature renders long-term studies of the upper atmosphere feasible, while facilitating quick debugging of the code.

GITM is initialized using the MSIS [120] and IRI [121] empirical models of the thermosphere and ionosphere, respectively, to specify the neutral and ion densities and temperatures. Various solar and high-latitude drivers can be used to specify the relevant inputs from the surrounding regions. The EUVAC [122] and EUV91 [123] empirical models of the EUV spectrum based on the $F_{10.7}$ proxy are generally used to drive the solar flux. The high latitudes are typically driven using Weimer potential patterns [99] or the AMIE technique [124] as well as the particle precipitation model [125].

8.2.2 Data Assimilation Algorithms

The data assimilation technique is based on 14 state variables per cell, specifically, the number densities and vertical velocities of the individual neutral species O, O_2, N, N_2, the eastward and northward bulk velocities, the normalized thermospheric temperature $\mathcal{T}_n$, which is given by

$$\mathcal{T}_n = P/\rho, \tag{8.1}$$

where P is the pressure and ρ is the neutral density, as well as the number densities of the ion species O^+, O_2^+, and NO^+. These state variables, which are updated by solving the equations of continuity, momentum, and energy, are used to compute the number density of electrons, the ion velocity, and the ion temperature, which correspond to data provided by an ISR.

In adapting GITM for data assimilation, variations of the basic formalism of the classical Kalman filter can be used. The most popular methods replace the Riccati equation error covariance propagation of the classical filter with an ensemble of models that approximate the second moment of the error covariance, which is subsequently used to determine the data injection gain. Two such methods are the ensemble Kalman filter (EnKF) [126], which is based on stochastically sampled drivers, and the unscented Kalman filter (UKF) [127], which is based on deterministically determined drivers for an ensemble of $2n+1$ members, where n is the number of states.

EnKF and UKF have three main advantages over classical techniques. First, they are often effective for nonlinear systems when the extended Kalman filter is not. Second, even when the extended Kalman filter is effective, both filters obviate the need to linearize the system. Finally, EnKF and UKF are eminently parallelizable since each ensemble member can be updated independently.

Many of the EnKF or UKF applications of interest arise from extremely high-order dynamics. In particular, GITM has a total of 10^6 states, and therefore significant computational resources are required for real-time simulation. Real-time data assimilation based on UKF would require several million processors, which is not feasible. To reduce the computational complexity of UKF, localized UKF (LUKF) propagates the error covariance of the local region with a local disturbance covariance matrix

[90, 128]. However, in applying UKF or LUKF to a large scale nonlinear system, we need to obtain an approximate disturbance covariance matrix, which requires additional computations and causes another source of uncertainty through the approximation.

For very large scale systems, EnKF has a potential advantage over UKF in that the number of ensemble members is not specified. However, useful guidance for the appropriate size of the EnKF ensemble based on linearized analysis is given by [129, 130]. Moreover, since EnKF perturbs the mean drivers, EnKF does not need a disturbance covariance matrix, which inherently eliminates the error in the approximated disturbance covariance matrix. Meanwhile, we need to propagate an ensemble of the entire system when the disturbance is global or its effects propagate throughout all regions of the system. In those cases, enforcing localization of the disturbances violates the base-line physics.

The ensemble-on-demand Kalman filter (EnODKF) [130] avoids propagating ensemble members for every time step by creating an ensemble of members only when data are available. This ensemble is propagated into the future to generate an error-covariance matrix, which, in turn, is used to create a data injection gain, which, finally, is used to assimilate the measurements at the time step at which the measurements became available. Once the measurements are assimilated, only a single simulated model is updated until new measurements become available. EnODKF with the ensemble size guideline of [129, 130] is a computationally inexpensive method that is applicable to the system perturbed by a single global disturbance. We thus apply EnODKF to the data assimilation using GITM when the ionosphere-thermosphere is driven by the solar EUV flux disturbance.

In contrast to the solar EUV flux, the geomagnetic storm drivers perturb the lo-

cal high-latitudinal regions of the ionosphere-thermosphere. Thus, the effects of the geomagnetic storm drivers propagate toward the equator from the regions around the north or the south pole. It takes 30 - 60 minutes for the geomagnetic storm drivers to affect the regions from 45°S latitude to 45°N latitude. Hence, EnODKF is not a suitable method for the geomagnetic storm case since it propagates ensemble members for a limited number of steps only when data are available while discarding previously propagated information. For this geomagnetic storm case, EnKF is effective since it continually propagates the system under disturbance drivers. Therefore, we apply EnKF to the geomagnetic storm case with small a ensemble size determined by using the ensemble size guideline [129, 130]. We summarize the EnKF and EnODKF algorithms in Section 7.2 and 7.5.

In applying EnODKF and EnKF to GITM-based data assimilation, we run GITM over the global ionosphere-thermosphere since the solar EUV flux is a global driver and the effects of geomagnetic storm propagate throughout the whole ionosphere-thermosphere. However, we inject the data into the local region where the states of the local region are well correlated with the measurements.

It is important to note in EnODKF and EnKF GITM-based data assimilations is that there are poor correlations between some states in auto and cross covariance matrices. Good correlations are guaranteed in the pairs of (neutral densities, neutral densities) and (neutral densities, air densities); (neutral velocities, neutral velocities); (neutral temperatures, neutral temperatures) and (neutral temperatures, ion temperatures); (ion densities, ion densities) and (ion densities, electron densities). We observe that the respective states of poor correlations have far different speeds of response or far different sensitivities to disturbances. Poor correlations result in poor DA performance or crashes of DA simulations. Therefore, we set the poor cor-

relations to zero, in the state-output cross covariance matrix and output-output auto covariance matrix.

8.2.3 Ensemble size

The accuracy of EnKF improves as the number of ensemble members is increased. However, a large number of ensemble members may be computationally intractable in terms of time and memory. Therefore, it is necessary to determine the minimum ensemble size that can adequately approximate the mean of the states and the error covariance. In order to obtain insight into the minimum ensemble size, consider the linear system

$$x_{k+1} = A_k x_k + w_k, \tag{8.2}$$

where $x_k \in \mathbb{R}^n$, $A_k \in \mathbb{R}^{n\times n}$, and $w_k \in \mathbb{R}^n$ is a random disturbance with mean zero and covariance Q_k. Now, let $x_k^{\mathrm{f},i}, \ldots, x_k^{\mathrm{f},q} \in \mathbb{R}^n$ denote the states of the ensemble members for the linear system (8.2), and define

$$H_k \triangleq [A_k x_k^{\mathrm{f},i} - A_k x_k^{\mathrm{f}},\ \ldots\ , A_k x_k^{\mathrm{f},q} - A_k x_k^{\mathrm{f}}] \in \mathbb{R}^{n\times q}, \tag{8.3}$$

where $x_k^{\mathrm{f}} \triangleq \frac{1}{q}\sum_{i=1}^{q} x_k^{\mathrm{f},i}$. Then, the ensemble size q for which sampled disturbances can be generated to reproduce the Kalman filter statistics is constrained by the inequality given by [129, 130]

$$\operatorname{rank}(H_k) + \operatorname{rank}(Q_k) + 1 \leq q. \tag{8.4}$$

The minimum value of q given by (8.4) provides a guideline for the ensemble size for the linear system.

In Section 7.2, (8.4) is applied to EnKF for the nonlinear system given by (7.1).

8.2.4 Implementation of the Algorithms

The original GITM code is written in fortran95 using MPICH for parallelization. The data assimilation code resides at a higher level than the GITM code and controls runs of multiple GITM codes for its purposes. The data assimilation (DA) code is also written in fortran95 using MPICH and uses LAPACK functions to perform matrix calculations. Although the original GITM code uses a variable time step, we fix the time step as 1 s in data assimilation code, for time-consistency in all ensemble members and simulated measurement data. The configuration of 3D CFD cells and processors for the simulations of this dissertation is shown in Table 8.1.

Table 8.1: Configuration of 3D cells and processors in the GITM simulations.

longitude, latitude range and resolution	0° to 360° in longitude with 5° resolution, 90°S to 90°N in latitude with 5° resolution
number altitudinal grid points	50 grid points
number of longitudinal blocks	8
number of latitudinal blocks	4
number of altitudinal blocks (fixed)	1
number of longitudinal cells per block	9
number of latitudinal cells per block	9
number of altitudinal cells per block	50
number of blocks per processor	1
total number of processors	32

The flowchart of the GITM-based DA is illustrated in Figure 8.1.

To increase the running-speed of ensemble filters, in particular, for the realtime execution of EnKF-based GITM DA code, we need to propagate an ensemble of GITM codes in parallel. To realize the parallel execution of an ensemble of GITM codes with the configuration given in Table 8.1, the required number of processors is 32 $\times$ ensemble size. However, in actual situations, the availability of computing resources in public grid computers is limited by public rules. Actually, the 32 is the largest number of processors that can be used for our GITM-based DA simulations

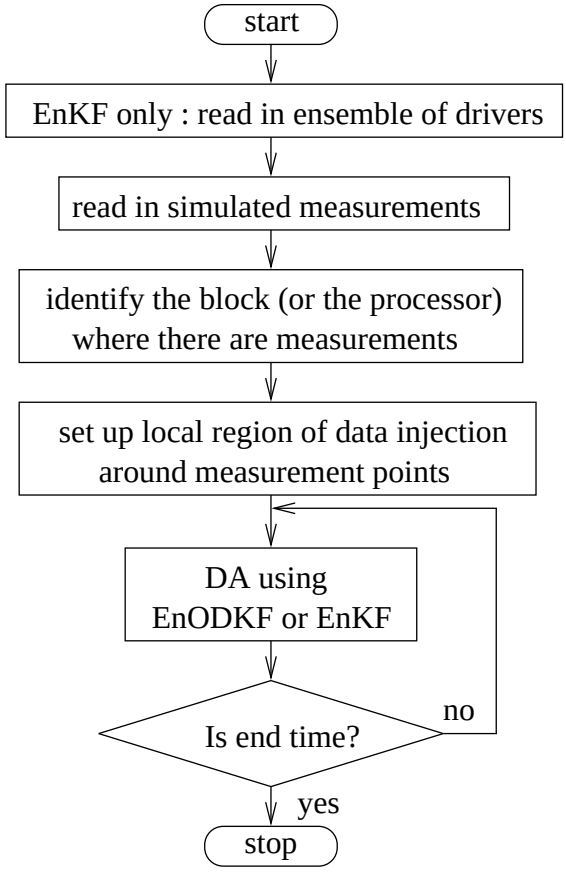


Figure 8.1: Flowchart of GITM-based DA code.

without too much waiting time when we use the public gird computer NYX at the University of Michigan. Under the environment of the limited computing resources, Figure 8.2 and 8.3 show how current implementation propagates an ensemble of members in EnODKF and EnKF, respectively.

The original GITM code runs 40% faster than realtime on the NYX grid computer. Using the implementations shown in figures 8.2 and 8.3, the typical running-speeds of EnODKF-based (measurements available every 4 minutes, 1 minute for ensemble-on-demand propagation) and EnKF-based (7 ensemble members) GITM DA codes are 0.9 $\times$ realtime and 7 $\times$ realtime, respectively.

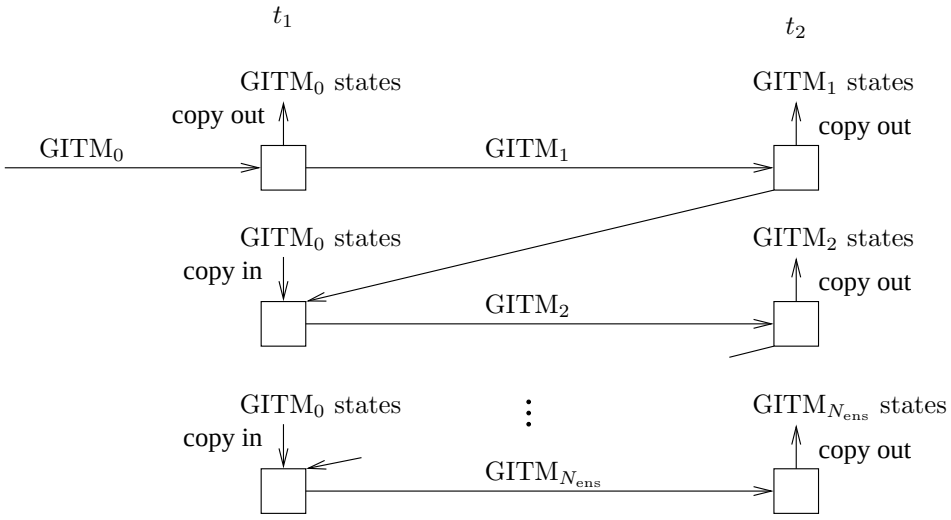


Figure 8.2: Propagation of an ensemble of GITM codes in series in EnODKF-based GITM DA. t_1 is start time and t_2 is end time of an ensemble-on-demand propagation. GITM$_0$ represents the main GITM code to which data are injected. GITM$_i$, $i = 1, \ldots, N_{\text{ens}}$ represents the GITM code for the i^{th} ensemble member.

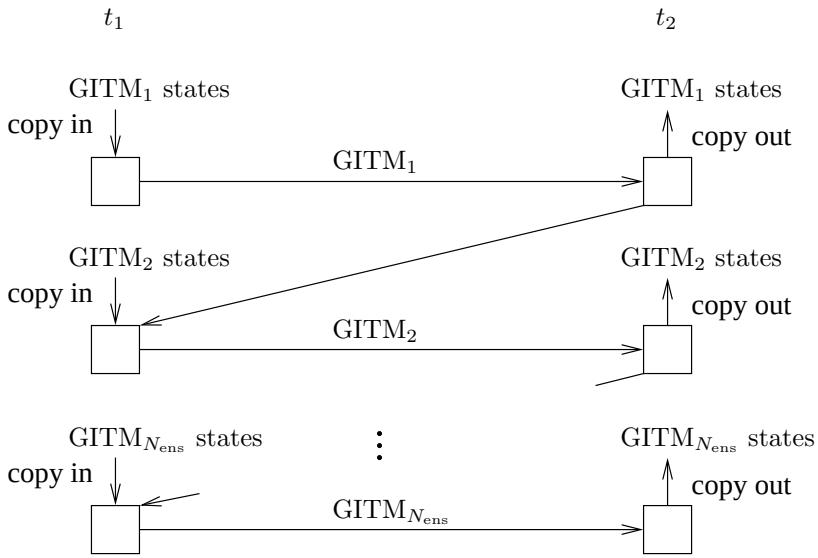


Figure 8.3: Propagation of an ensemble of GITM codes in series in EnKF-based GITM DA. t_1 is the start time and t_2 is the end time of an ensemble propagation. t_1 and t_2 correspond to times of two subsequent measurements. GITM$_i$, $i = 1, \ldots, N_{\text{ens}}$ represents the GITM code for the i^{th} ensemble member.

8.3 Results

8.3.1 Solar flux disturbance simulation

The solar EUV flux is one of the main drivers of the ionosphere-thermosphere. In practice, direct EUV flux measurements are not provided continually. Instead,

only the daily average of the EUV flux parameter $F_{10.7}$ expressed in solar flux unit (sfu), where 1 sfu = 10^{-22} watt per square meter per Hertz, is available. Hence, the EUV flux variation within a day is uncertain. We first consider data assimilation for GITM (GITM-DA) in the presence of uncertain EUV drivers.

In [131], from measured data for a solar flare event, it is shown that the directly measured EUV flux time profile is similar to the x-ray flux time profile. We can obtain 1-minute- or 5-minute-resolution x-ray flux data continuously from (http://www.swpc.noaa.gov/today.html), and we can see that the x-ray flux time profile within a day is composed of intermittent wedge or triangular-shaped random spikes with a time duration of 1 to 3 hours. Based on these observations, $F_{10.7}$ within a day is modeled by a triangular wave with 3-hour peak-to-peak variation as shown in Figure 8.4, where the peak amplitudes are random.

To obtain simulated data, GITM is run with the $F_{10.7}$ profile given in Figure 8.4 for 36 hours. The model results are then used as measurement data when performing data assimilation. The solar flux is assumed to drive the entire dayside in a consistent manner, and there are no secondary effects due to propagation. Therefore, the EUV flux can be considered a single global driver for the dayside of the Earth, which allows for use of EnODKF [130].

For the case of the EUV flux disturbance, rank(Q_k), which is equivalent to the number of independent disturbance drivers, is one. Since rank(H_k) ≥ 1 we need at least three ensemble members in the EnODKF based on (8.4). Simulated point measurements of the thermospheric density, electron number density N_{e}, and ion temperature T_{ion} are taken at 400 km altitude.

In GITM, the time constants associated with the modeled reactions vary significantly. For example, the number density of O^+ responds quickly to the variations in

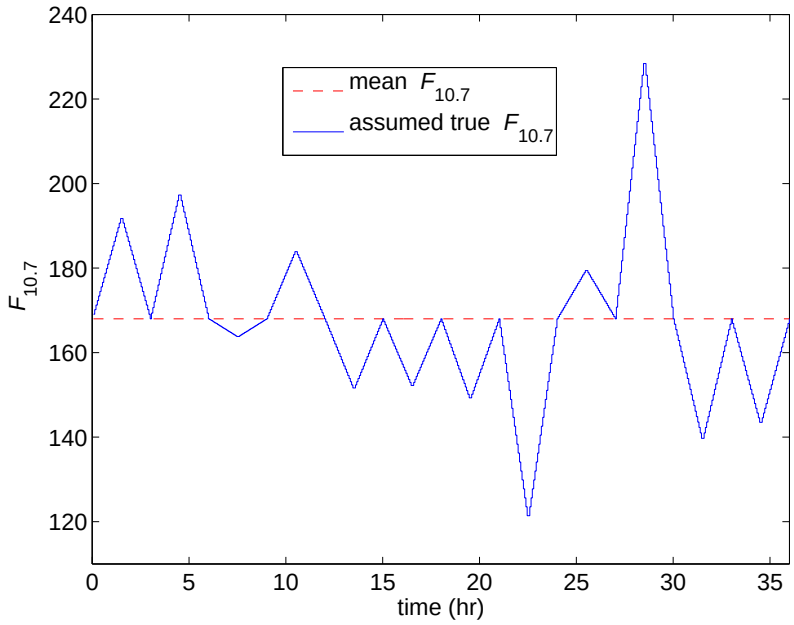


Figure 8.4: Assumed triangular true $F_{10.7}$. Mean of $F_{10.7}$ is 168.

the EUV flux, whereas the response of the neutral density is sluggish. Hence, in applying EnODKF, we constrain the auto- and cross-correlation so that the correlations between quantities with different time constants are zero to avoid filter instabilities induced by inaccurate correlations. However, since we ignore correlations between different quantities such as densities and temperatures, we cannot achieve good estimation results for all states by using only one type of measurement quantity.

We start numerical investigation of data assimilation with 2 cases of simulated measurement quantities at a single measurement location with local data injection. For each measurement case, we inject data into a local region encompassing 5 cells by 5 cells by 9 cells in longitude, latitude, and altitude, respectively, centered at the measurement point. We summarize in Table 8.2 the conditions of data assimilation for the two cases of measurement quantities.

Next, to quantify the DA accuracy, we define

$$\epsilon \triangleq \frac{|X_{\mathrm{DA}} - X_{\mathrm{truth}}|}{|X_{\mathrm{NoDA}} - X_{\mathrm{truth}}|}, \tag{8.5}$$

Table 8.2: Summary of GITM-DA for solar EUV flux disturbance.

GITM longitude, latitude range and resolution	0° to 360° in longitude with 5° resolution, 90°S to 90°N in latitude with 5° resolution
GITM altitude range and grid points	100 km to 700 km, 50 grid points
region of data injection	280° to 305° in longitude (5 cells), 10° to 35° in latitude (5 cells), 324 km to 484 km in altitude (9 cells),
measurement point	single point at 292.5° in longitude, 22.5° in latitude, 402 km in altitude
measurement quantities	case 1 : neutral mass density ρ_{n} only, case 2 : electron number density N_{e}, ion temperature T_{ion}, and neutral mass density ρ_{n}

where X is a simulated quantity, for example, density or temperature, DA indicates results from GITM-DA, NoDA are results from GITM without data assimilation but with $F_{10.7}$ given by its mean value, and truth denotes results from the truth model. $0 < \epsilon < 1$ implies GITM-DA reduces errors from NoDA.

Figure 8.5 shows the results of data assimilation when only measurements of the neutral density ρ_{n} are used. In Figure 8.5, the accuracy of the DA estimate of the thermospheric temperature is worse than the accuracy of the NoDA estimate of the thermospheric temperature. On the other hand, the DA estimate of thermospheric density inside the injection region improves. Estimation of N_{e} is worse than NoDA at the measurement point with slight error reduction below the measurement point. Outside the data injection region, the data assimilation has little effect. However, as shown in Figure 8.6, using measurements of $N_{\mathrm{e}}, T_{\mathrm{ion}}$, and ρ_n yields better estimation accuracy than data assimilation using only ρ_{n} measurements.

However, when the measurements of $N_{\mathrm{e}}, T_{\mathrm{ion}}$, and ρ_n are used, a discontinuous pressure change across the border of data-injection region is induced by data assimilation. The pressure difference results in an unrealistic disturbance in the horizontal velocities (Figure 8.7, right), which, in turn, disturbs the electron density (Figure

8.7, left).

In Figure 8.8, we enlarge the data injection region to 270° to 315° in longitude (9 cells), 0°N to 45°N in latitude (9 cells), and 261 km to 559 km in altitude (17 cells) when all measurement quantities (N_e, T_{ion}, and ρ_n) are used. The plots show that increasing the size of the injection region also increases the effective data assimilation region, but with an increase in the size of the adversely affected region. However, when the data injection region is extended to the full day-side of the Earth, the induced pressure gradient disappears since the uncertainties of the EUV flux are confined to the day-side of the Earth.

8.3.2 Geomagnetic disturbance simulation

Geomagnetic Storm Drivers

During storm conditions, the magnetospheric electric fields can become highly dynamic. Since these fields map to ionospheric altitudes, they have a strong effect on the motion of the ions, and as the fields change magnitude and direction, so do the ions. These changing flows can cause substantial increases in the Joule heating rate in the high-latitudes. Joule heating can be expressed as

$$Q_j = \sigma_p(\mathbf{E} + \mathbf{U} \times \mathbf{B}), \tag{8.6}$$

where σ_p is the Pedersen conductance, $\mathbf{E}$ is the electric field, $\mathbf{U}$ is the thermospheric neutral wind velocity, and $\mathbf{B}$ is the magnetic field. Since the neutral winds can be difficult to determine, this equation is often simplified to $Q_j = \sigma_p E^2$.

The sudden addition of energy into the high-latitude ionosphere can affect the atmosphere in various ways. First, the absorption of energy can cause the atmosphere to expand, causing increases in neutral density at a given altitude as high density air from below is pushed up. In addition, chemical reaction rates that are

temperature dependent can be affected, resulting in compositional changes. Sudden increases in energy deposition can also launch large scale gravity waves that propagate equatorward, resulting in compositional and dynamical perturbations on a global scale.

In this study, the electrodynamic potential patterns [99] and the particle precipitation patterns [125] are used to drive GITM under geomagnetic storm conditions. These inputs are determined by the B_y and B_z components of the interplanetary magnetic field (IMF), solar wind speed, and hemispheric power index (HPI). The hemispheric power is an estimate of the energy in gigawatts of all electrons precipitating into a hemisphere. The hemispheric power is obtained by comparing the energy flux observed along an orbital track of a NOAA/TIROS spacecraft with the climatology maps of the energy flux developed from the NOAA/TIROS database. The maps are based on 10 levels of geomagnetic activity. The profile and magnitude of the energy flux of precipitating electrons observed along an orbital track is matched to one of the 10 maps. The HPI is thus the number of the relevant map.

It is assumed that the uncertainties of high-latitude drivers are represented by the uncertainties of the primary drivers, that is, B_y, B_z, and HPI. The solar wind is assumed to be given.

Analysis of B_y, B_z and HPI, and Assumptions on Uncertainties

The IMF components B_y and B_z, shown in Figure 8.9, are measured by the ACE satellite at the L1 point, sunward of the Earth, from 01:00UT, October 29th, 2003 for 71 hours. Since these measurements are used to calculate the potential based on empirical relationships, and since there is ambiguity on how the measurements are propagated to the magnetosphere, there is inherent uncertainty in these IMF components and their effect on the electric potential. It is assumed that the low-

frequency components of the IMF near the Earth do not change from the measured IMF, although the high frequency components change randomly with zero mean. The measurement of the solar wind velocity is assumed to be accurate, and we assume that the solar wind speed is uniform between the ACE satellite and the Earth. Therefore, the high-frequency components of B_y and B_z are the uncertain drivers.

The remaining geomagnetic driver HPI is also shown in Figure 8.9 at the times when B_y and B_z are measured. In addition to HPI data collected by TIROS, HPI data are also obtained by the DMSP satellites. In Figure 8.10 (http://cedarweb.hao.ucar.edu/dmsp/dmspssj4_hp.html), a scatter plot of the NOAA-12 (local time at the equator 18.5 hours) and DMSP F13 (local time at equator 17.5 hours) data for one year is provided. Each point shows the HP obtained simultaneously (+/- 10 minutes) by the two spacecraft. Any value of HP from NOAA-12 or DMSP F13 that is less than 5 GW is ignored because the data may be from a pass that did not cut through the full auroral zone. A linear regression is performed with all of the data shown and the solid line shown is the result of the regression analysis. The resulting correlation coefficient of 0.817 indicates a reasonably good correlation between the two data sets. However, the dispersion of the data points around the regressed linear line implies that the HPI has a large degree of uncertainty. Hence, it is assumed that the high frequency components of HPI are another source of uncertainty during geomagnetic storm conditions.

As shown in Figure 8.12, the low and high frequency components of B_y, B_z, and HPI are separated according to Figure 8.11. The statistics of the high frequency components of B_y, B_z, and HPI are computed with respect to time, and are used to generate the perturbing disturbances to each ensemble member. The high frequency

signals of B_y, B_z, and HPI are are assumed to be stochastically independent.

EnKF for the Geomagnetic Storm Data Assimilation and Ensemble Size

EnKF is chosen for GITM-DA during geomagnetic storm conditions because these conditions strongly involve disturbance propagation, for example, gravity waves. EnODKF does not work well for this problem since it propagates the ensemble members for only a limited number of steps after each measurement time and subsequently discards the information from previous propagations.

For the ensemble size q, the sets of drivers B_y, B_z, and HPI are generated for the ensemble members by using the analyzed statistics of the high frequency components of the measured drivers. The ensemble size q is determined using (8.4). rank(Q_k) is 3 because the number of independent drivers is 3. Next, we take rank(H_k) of the local data injection region, which includes measurement locations, and we assume that rank(H_k) of the local region is 3, considering the first three dominant singular values of the H_k. Finally, the ensemble size q is set to be 7.

Data assimilation conditions for a simulated geomagnetic storm

We use GITM with same longitudinal and latitudinal resolution as the EUV case, for data assimilation under geomagnetic storm condition. Throughout the geomagnetic storm DA, we set constant $F_{10.7} = 267$ which is the three-day average of daily average $F_{10.7}$ data from October 29th, 2003 to October 31st, 2003. Then, the altitude range is set to 100 km - 857 km with 50 altitudinal grid points using the $F_{10.7}$ value. For simulated measurements, we consider six measurement locations distributed over two GITM blocks where one GITM block is composed of 9 cells by 9 cells by 50 cells. For point measurements, we consider grid points that are nearest to 400 km in altitude. We name each measurement point as m1, ... , m6. Next, 4 longitudinal by 2 latitudinal GITM blocks and 4 longitudinal by 1 latitudinal GITM blocks are considered for the size of region of data injection. The latter is used for the size of the default region of data injection for the geomagnetic storm DA. Range of altitudes of the region of data injection is from 312 km to 668 km where 17 altitudinal grid points are defined. We summarize data assimilation conditions for geomagnetic storm in Table 8.3.

Table 8.3: Data assimilation conditions for the simulated geomagnetic storm.

GITM longitude, latitude range and resolution	0° to 360° in longitude with 5° resolution, 90°S to 90°N in latitude with 5° resolution
GITM altitude range and grid points	100 km to 857 km, 50 grid points
date and time of geomagnetic storm	01:00UT, Oct. 29, 2003 to 01:00UT, Nov. 1, 2003
default region of data injection (4 longitudinal by 1 latitudinal GITM blocks)	180° to 90° in longitude (36 cells), 0° to 45° in latitude (9 cells), 312 km to 668 km in altitude (17 cells),
measurement points	6 points defined by name (longitude, latitude, altitude as follow : m1 (282.5°, 22.5°, 395 km), m2 (287.5°, 32.5°, 395 km), m3 (302.5°, 27.5°, 395 km), m4 (322.5°, 27.5°, 395 km), m5 (327.5°, 37.5°, 395 km), m6 (337.5°, 32.5°, 395 km)

Data Assimilation Results Using Simulated ISR Measurements and Neutral Mass Density

Electron number density N_e is a major ISR measurement quantity for ions. Ion temperature T_{ion} is another ion measurement quantity by ISR but its time profile is similar to neutral temperature, hence, it can be regraded as neutral measurement. Neutral mass density ρ_n can be obtained from geo-scientific satellites. We assess the performance of geomagnetic storm DA using these available ion and neutral measurement quantities N_e, T_{ion} and ρ_n. We consider semi-ideal measurement conditions for these available measurement quantities when we take simulated measurements from 6 locations. That is, we assume that ISR and satellite measurements are taken every one minute, all at the same time instant from fixed 6 measurement locations, while, in reality, one ISR cannot measure 6 points at the same time although the measurement points are within the range of ISR, and satellite at around 400 km altitude cannot stay fixed. However, the performance of geomagnetic storm DA using these semi-ideal measurements is the baseline performance that can be referenced when realistic simulated measurements or real measurements are used for the data assimilation.

We start the data assimilation from 9:00UT, October 29, 2003 with the region of data injection whose longitudinal-latitudinal size is 4 by 2 GITM blocks. However, data assimilation is stopped by error at 10:56UT, October 29, 2003, due to excessive ion generations within the region of data injection. Figure 8.13 shows the NoDA and DA error contours at 10:50UT, just before the stop. In the lower plot, the excessively increased errors are identified at the lower left and the lower right corners of the region of data injection which are distant from the measurement locations. The areas of large errors are the locations where the correlations with measurements are weak.

Hence, the erroneous correlations with measurements are reasons for the stop of DA.

We start again the geomagnetic storm DA from 11:00UT using the states of NoDA at 11:00UT for initial conditions for all 7 ensemble members. The region of data injection is reduced to the default region describe by Table 8.3.

Using reduced region of data injection, geomagnetic storm DA shows good performance for 14 simulated hours with feasible running speed. The geomagnetic storm DA is setup to use 32 processors where each processor is assigned to each GITM block. Currently, since the DA code runs on a public grid computer (NYX at the University of Michigan), using more than 32 processors is actually difficult due to too much waiting time before execution of DA code. Hence, each ensemble member runs one by one from a measurement time to the next measurement time. With this hardware and software setup, the running speed of geomagnetic storm DA is five times slower than real time when we use 7 ensemble members in EnKF. Current run speed can be reduced to real time if more than $32 \times 5 = 170$ processors are available. Even though 170 is a large number, it is not an impossible number these days.

Estimation accuracy of states of GITM using measurements N_{e}, T_{ion} and ρ_{n} at 6 points is demonstrated in Figure 8.14. In Figure 8.14, truth, NoDA and DA results for states and measurement quantities at m2 location are compared. As shown in bottom plots, DA shows good agreement with measurement quantities at m2. Moreover, number densities of all neutral species , number densities of all ion species and neutral temperature are estimated with acceptable accuracy. However, since neutral velocity estimates are not directly estimated by data assimilation and are sensitive to small changes of pressure or density, they are randomly fluctuating, but stay around the truth results.

Now, we examine spatial estimation performance of geomagnetic storm DA. We

show DA performance on longitudinal-latitudinal plane at the measurement altitude. Results are similar at other altitudes. In Figure 8.15 - 8.17, we show geomagnetic storm DA spatial estimation performance on N_e, T_{ion} and ρ_n at two different times 16:00UT and 17:50UT, October 29, 2003. Notice that all states are well estimated if N_e, T_{ion} and ρ_n are estimated correctly.

In left plots of Figure 8.15, the NoDA error of N_e at 16:00UT that overlaps the data injection region is almost completely removed by DA with slight spill-over region near the region of data injection. Meanwhile, in the right plots, the NoDA error of N_e at 17:50UT that overlaps the data injection region is also well removed by DA but there occurs a large area of significant error below the region of data injection. Similar phenomenon is observed in Figure 8.16.

We can see an interesting fact in Figure 8.17 where ρ_n estimation performance is shown. In right plots for 17:50UT, the NoDA errors outside the region of data injection are reduced as well as inside. The change of ρ_n outside the region of data injection is induced by data assimilation. That is, the DA induces helpful effects for ρ_n estimates whereas worse effect for N_e and T_{ion} estimates. Hence, we observe that the DA-induced effects may not be correct. To overcome this problem, we need to enlarge the region of data injection with more measurement locations.

Data Assimilation Results Using Simulated TEC Measurements

TEC measurements are practically very useful measurements since they are widely distributed measurements over the lands of the world. In this section, we use TEC measurements at longitudinal-latitudinal locations of 6 measurement points of Table 8.3 and investigate effectiveness and limitations of TEC measurements on data assimilation for the ionosphere-thermosphere. The region of data injection is the default region described by Table 8.3.

We perform geomagnetic storm DA for 14 hours from 11:00UT using only TEC measurements. We use the states of NoDA at 11:00UT for initial conditions of all 7 ensemble members. Figure 8.18 shows the comparison of results of truth, NoDA and the geomagnetic storm DA using TEC measurements at m2 point. It is shown that estimated N_e by DA reasonably keeps track of true N_e. However, estimated T_{ion} and ρ_n by DA do not change significantly from NoDA. Moreover, we observe that N_e estimates by DA for 12:00UT - 16:00UT, Oct. 29th and 20:00UT Oct. 29th - 01:00UT Oct. 30th are not so accurate as the case where we use N_e, T_{ion} and ρ_n measurements.

In Figure 8.19, left plots show that the most NoDA error of N_e at 16:00UT reduced by DA. However, there is significant error at the bottom-right corner of the region of data injection. Moreover, there is a region of induced error left to the region of data injection. In the right plots, similar DA performance is shown at 17:50UT.

From simulation results given by Figure 8.18 and 8.19, we observe that, within the boundary of well-chosen region of data injection, for estimating N_e, using TEC measurements is effective, but is not better than using N_e, T_{ion} and ρ_n measurements.

8.4 Conclusion and Discussion

In this paper, we applied EnODKF and EnKF algorithms with small ensemble size to data assimilation for the ionosphere-thermosphere using GITM. We used simulated measurements for all numerical investigations. Two conditions of the ionosphere-thermosphere are considered separately: uncertain EUV flux and geomagnetic storm conditions. EnODKF is used for DA of uncertain EUV flux case where the EUV flux is a single global disturbance. EnKF is used for DA of geomagnetic storm case where disturbance is propagated from high latitude regions.

EnODKF is computationally inexpensive method and we could achieve realtime data assimilation using GITM. EnKF was effectively applied to geomagnetic storm DA because we could obtain good DA results using 7 ensemble members, which is determined by ensemble size guideline [129, 130]. The ensemble size 7 is reasonably small ensemble size that is acceptable by our available computing resources.

DA results from both cases show good estimation accuracy when we use ion measurements and neutral measurements together, while the poor correlations are set to zero in calculating the estimator gain. However, DA results may be poor or unstable when the region of data injection includes regions where correlation with measurement data are weak. In addition, we should notice that DA results outside the region of data injection may not be correct.

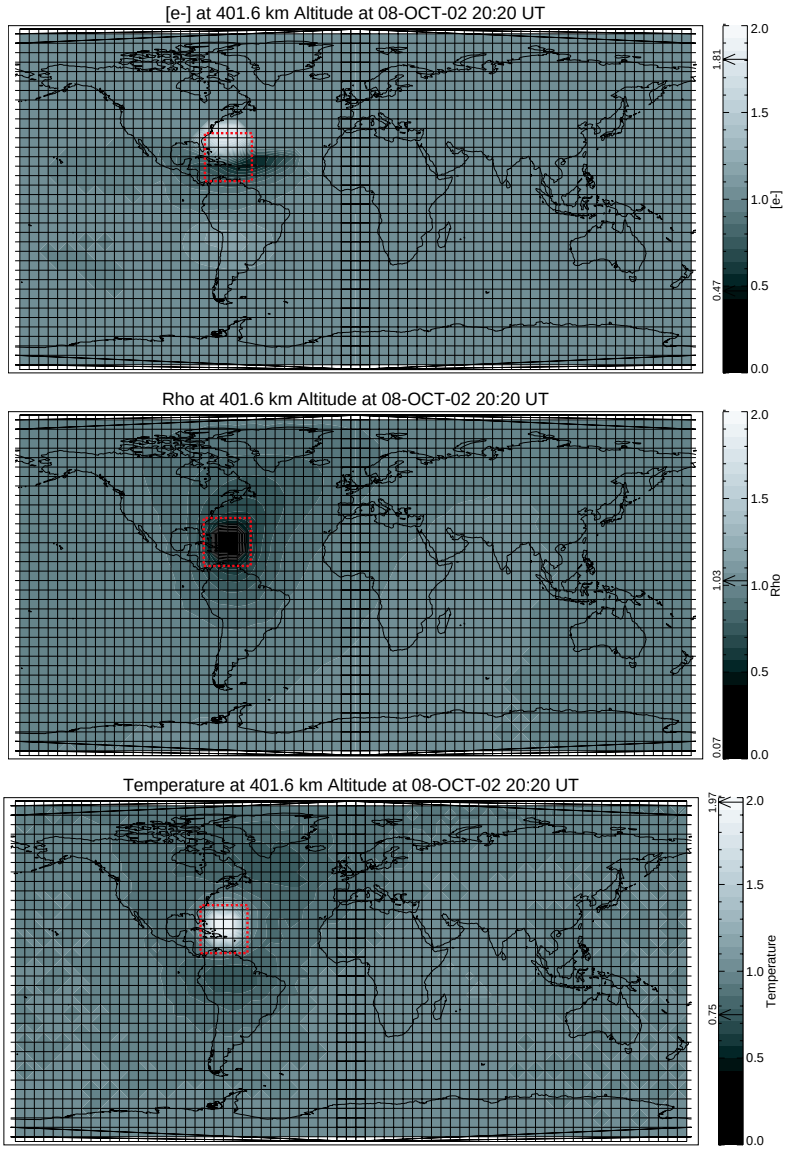


Figure 8.5: Error ratio contour plot of electron number density (top), neutral mass density (center), and error contour plot of normalized temperature (bottom), from the results of data assimilation for solar EUV flux case where only ρ_{n} measurement is taken for data assimilation. The data injection area is indicated by the dashed box whose size is 5 cells in longitude by 5 cells in latitude by 9 cells in altitude. The measurement location is the center of the box, where only ρ_{n} measurement is taken for data assimilation.

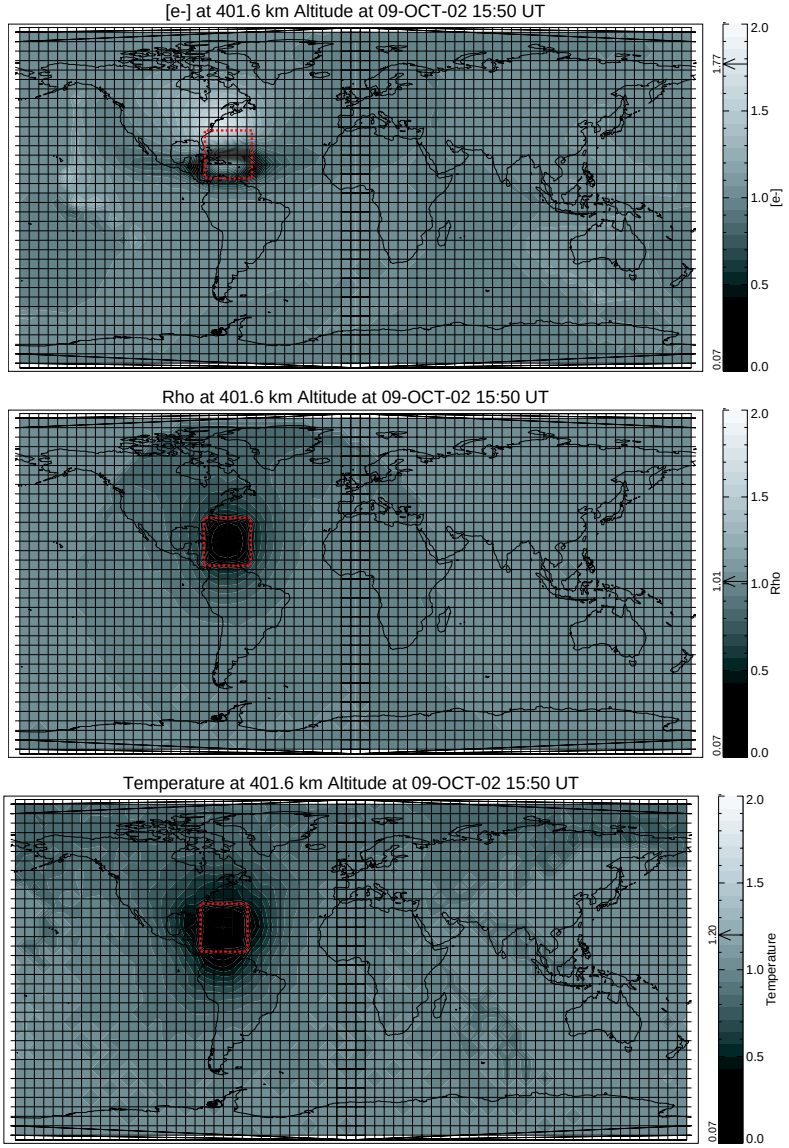


Figure 8.6: Error ratio contour plot of the electron number density (top), neutral mass density (center), and error contour plot of normalized temperature (bottom), from the results of data assimilation for solar EUV flux case where N_e, T_{ion}, and ρ_n measurements are taken for data assimilation. The data injection area is indicated by the dashed box whose size is 5 cells in longitude by 5 cells in latitude by 9 cells in altitude. Measurement location is the center of the box, where N_e, T_{ion}, and ρ_n measurements are taken for data assimilation.

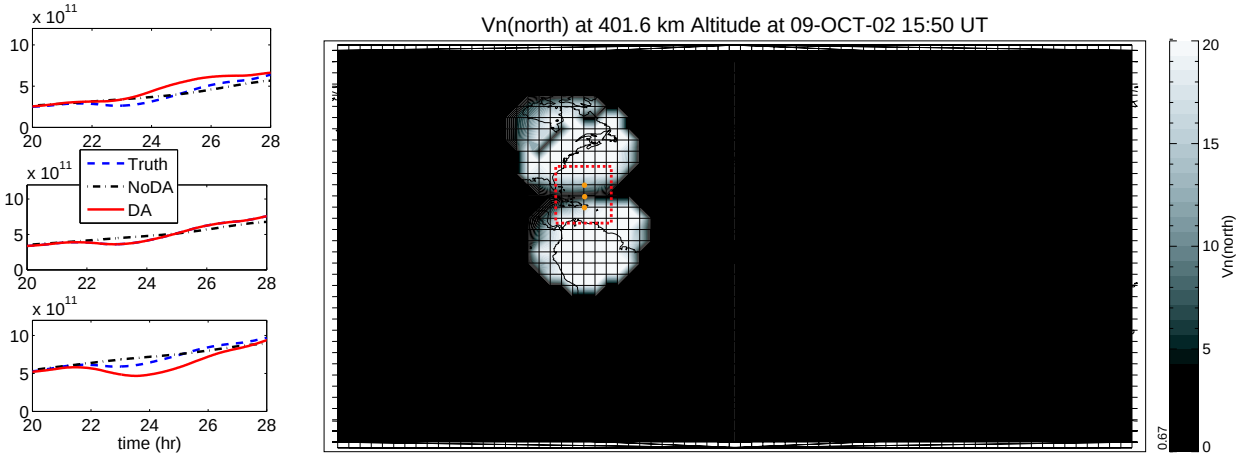


Figure 8.7: Electron number densities at 3 locations marked by dots (left), from top to bottom, and error ratio contour of northward horizontal velocity (right), from the results of data assimilation for solar EUV flux case where N_e, T_{ion}, and ρ_n measurements are taken for data assimilation. The data injection area is indicated by the dashed box whose size is 5 cells in longitude by 5 cells in latitude by 9 cells in altitude. Measurement location is the center of the box, where N_e, T_{ion}, and ρ_n measurements are taken for data assimilation.

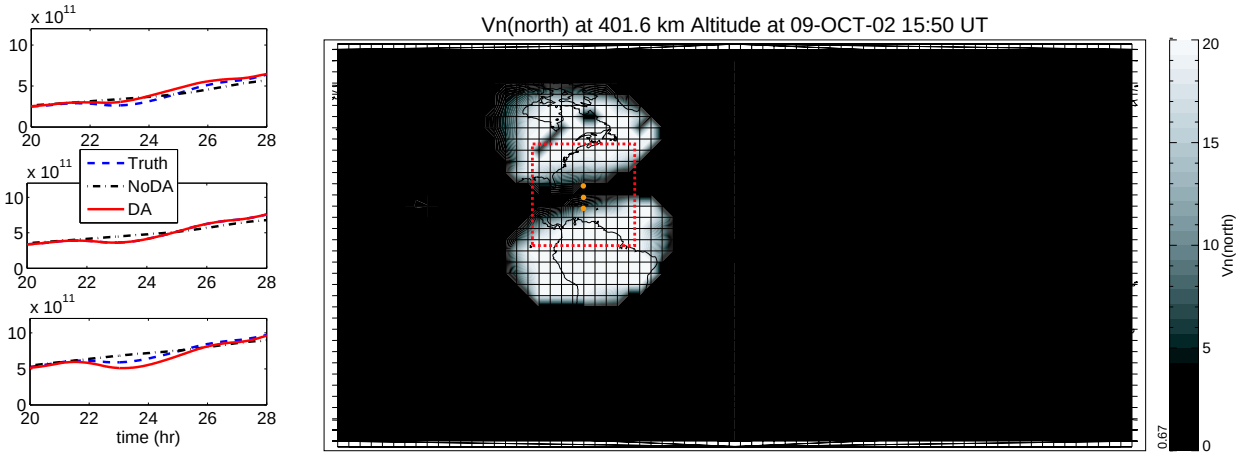


Figure 8.8: Electron number densities at 3 locations marked by dots(left), from top to bottom, and error ratio contour of northward horizontal velocity for the enlarged local region (right). The data injection area is indicated by the dashed box whose size is 9 cells in longitude by 9 cells in latitude by 17 cells in altitude. The longitude-latitude size corresponds to the one full GITM block. Measurement location is the center of the box, where N_e, T_{ion}, and ρ_n measurements are used for data assimilation.

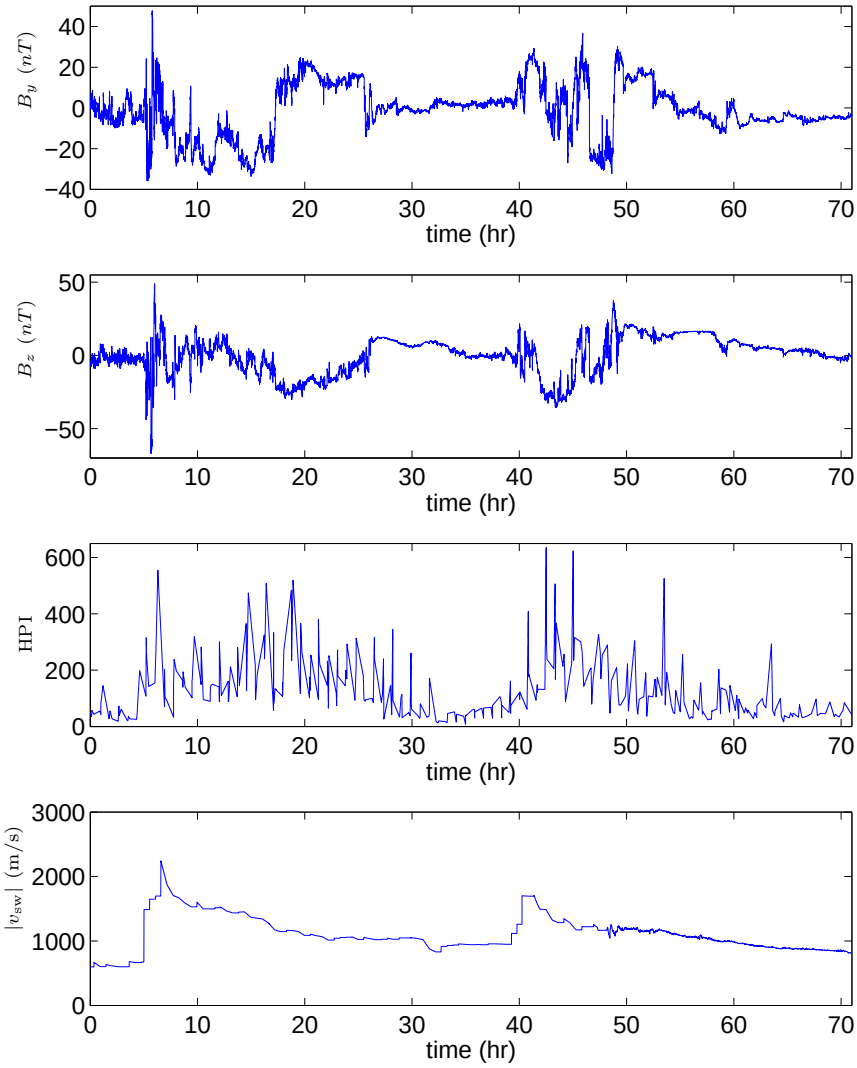


Figure 8.9: Measured values of interplanetary magnetic filed (IMF) components B_y, B_z in the Geocentric Solar Ecliptic (GSE) coordinate system, hemispheric power index HPI, and, magnitude of solar wind velocity $|v_{\mathrm{sw}}|$, respectively, which constitute geomagnetic storm drivers for the ionosphere-thermosphere.

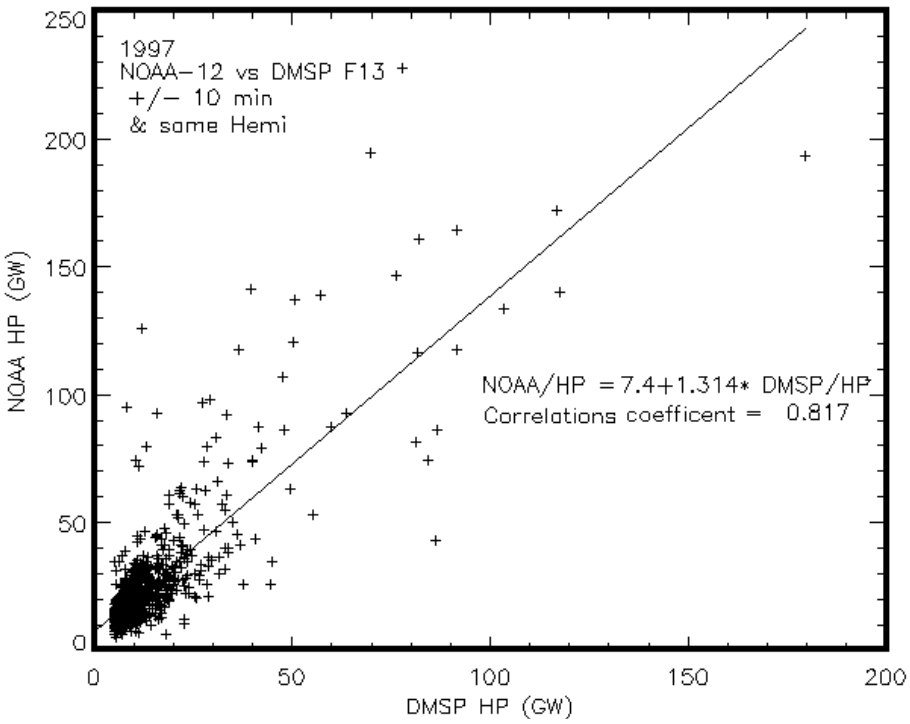


Figure 8.10: A scatter plot of the NOAA-12 and DMSP F13 data for one year. Each point shows the HP obtained simultaneously (+/- 10 minutes) from the two spacecraft. The solid line shows the result of a regression analysis.

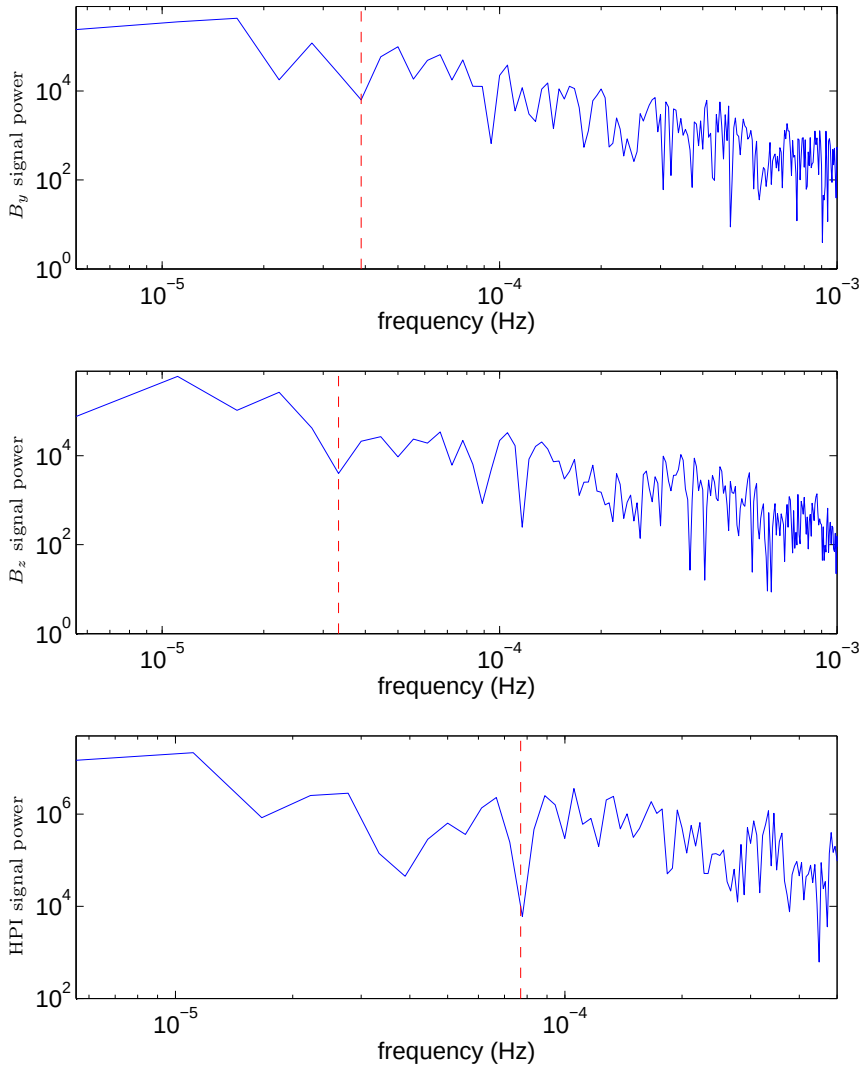


Figure 8.11: B_y, B_z, and HPI signal power spectra with boundary frequencies. We assume that signal components above the boundary frequencies are uncertain. The boundary frequencies, which are indicated by dashed vertical lines, are 3.9×10^{-5} Hz, 3.3×10^{-5} Hz, and, 7.8×10^{-5} Hz, respectively.

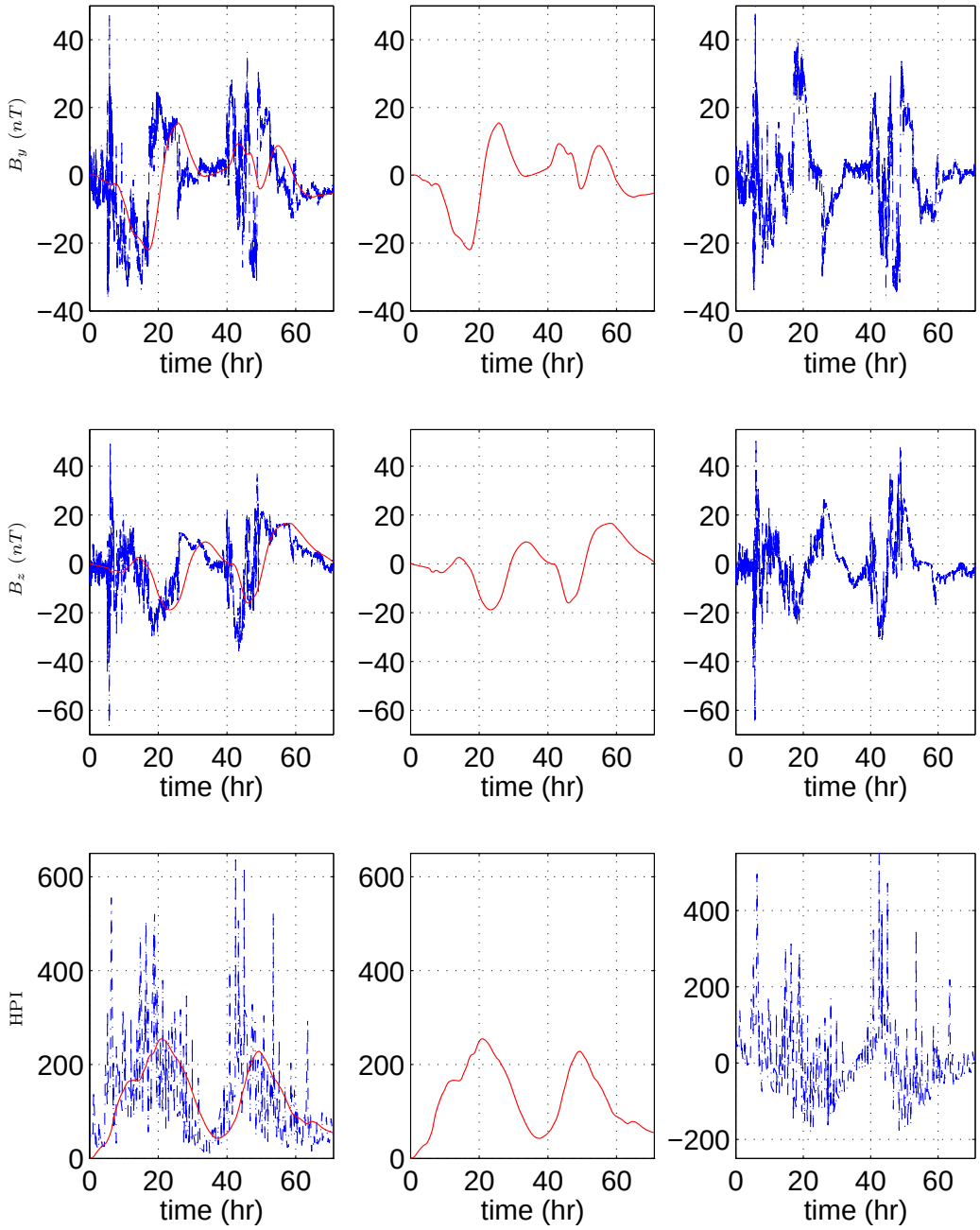


Figure 8.12: B_y, B_z, HPI and their separated low and high frequency signals, where the boundary frequencies are 3.9×10^{-5} Hz, 3.3×10^{-5} Hz, and, 7.8×10^{-5} Hz, respectively. Plots of leftmost column are the original signals overlapped with low-frequency signals.

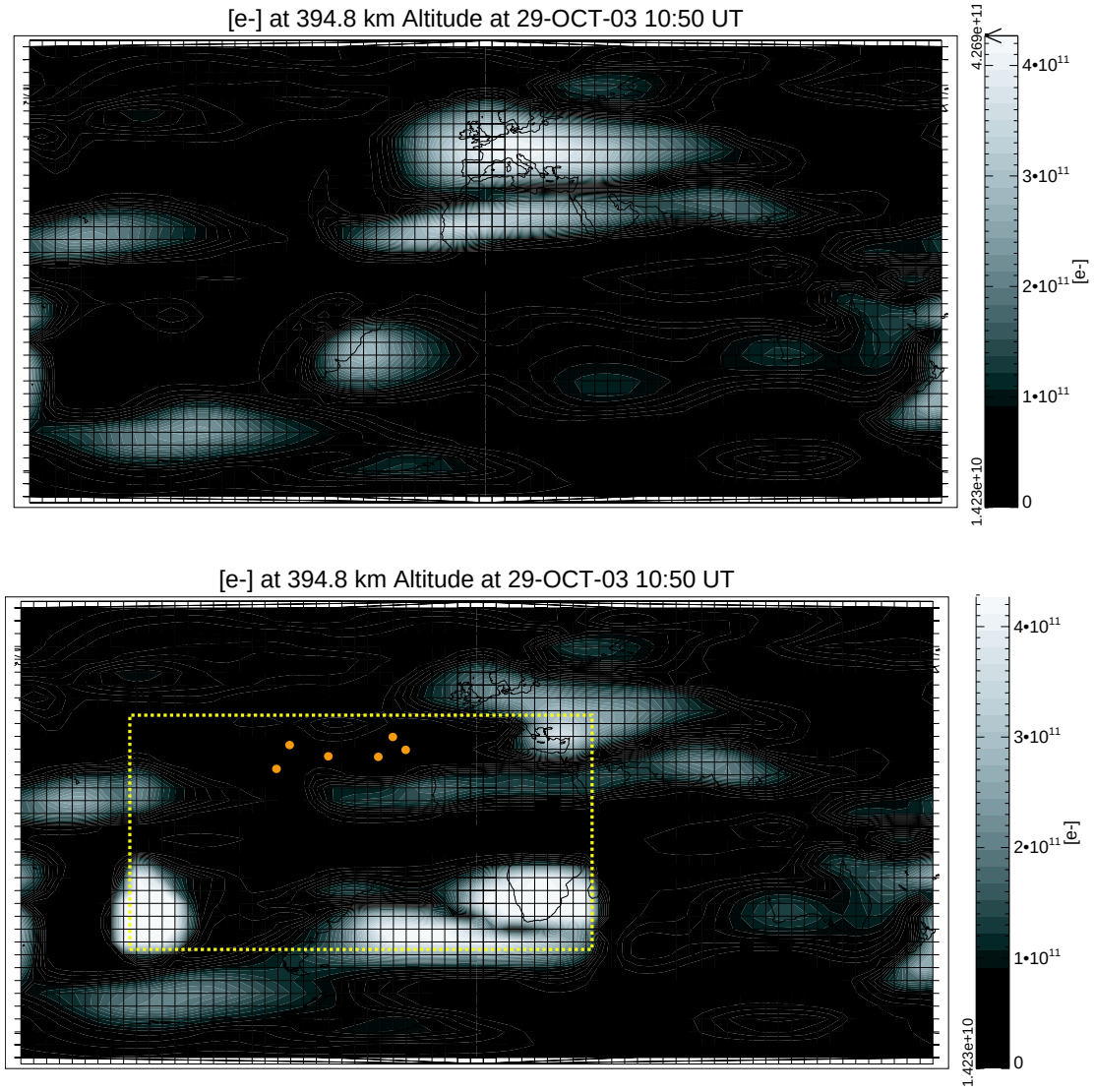


Figure 8.13: Absolute error contour plot of NoDA (top) and DA by EnKF (bottom) for geomagnetic storm conditions at 10:50UT, October 29, 2003, just before the stop by error of data assimilation. N_{e}, T_{ion}, and ρ_{n} at 6 measurement points are used for data assimilation. The region of data injection is indicated by dashed box whose longitudinal and latitudinal size is 4 by 2 GITM blocks. Longitudinal and latitudinal size of one GITM block is 9 cells by 9 cells. In the lower plot, the excessively increased errors are identified at the lower left and lower right corners of the region of data injection that are distant form the measurement locations.

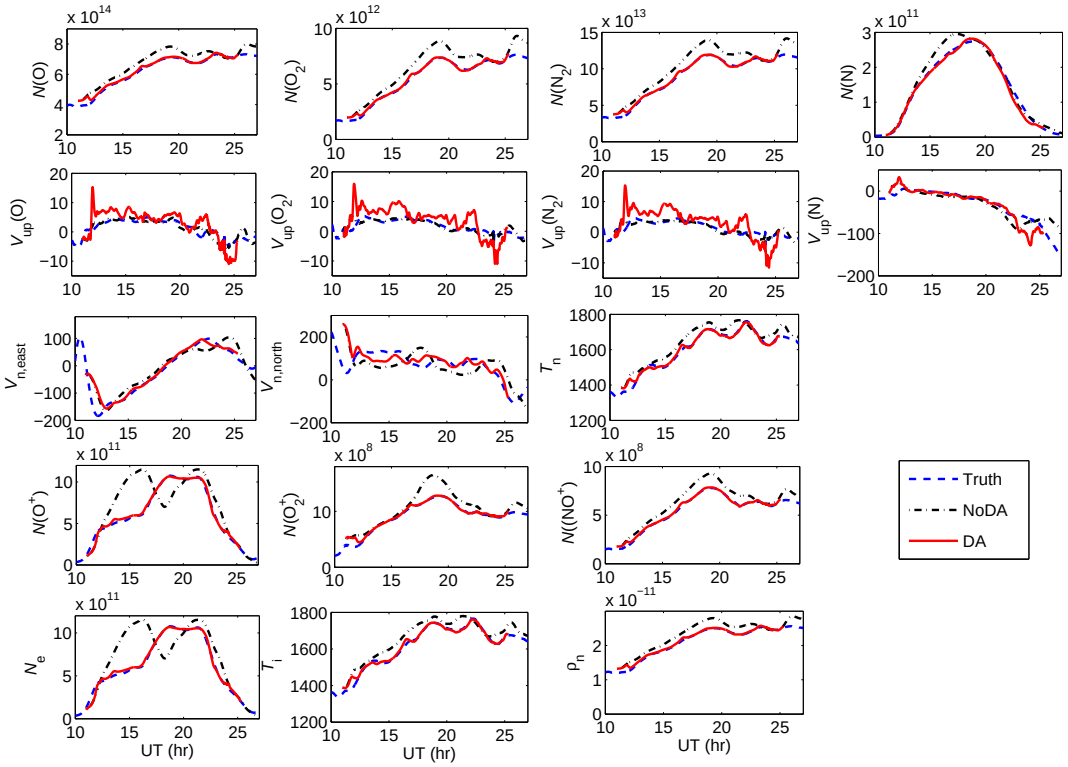


Figure 8.14: Comparison of states and outputs of truth, NoDA and DA by EnKF for geomagnetic storm conditions at m2 measurement location. N_e, T_{ion}, and ρ_n at 6 measurement points are used for data assimilation. x-axis is time (hr) from 00:00UT, October 29, 2003. The size of region of data injection is 4 by 1 GITM blocks and given by Table 8.3. $N(\cdot)$ is number density of the species in parenthesis, $V_{up}(\cdot)$ is vertical velocity of the species in parenthesis, $V_{n,east}$ is bulk neutral eastward velocity, $V_{n,north}$ is bulk neutral northward velocity, and T_n is neutral temperature.

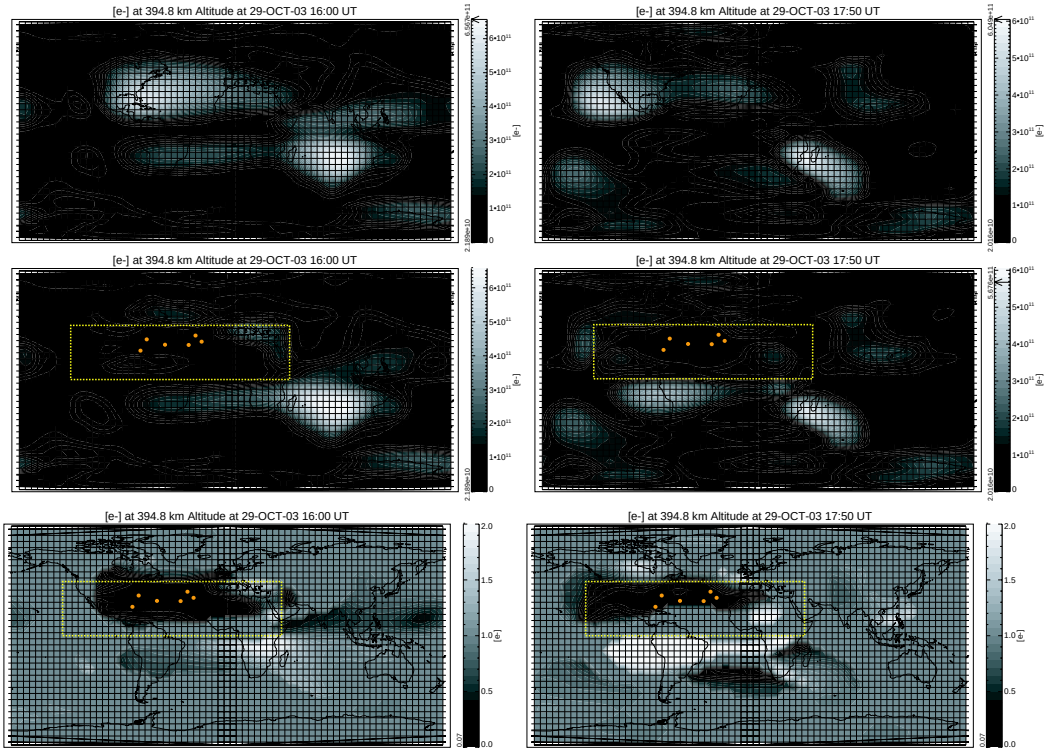


Figure 8.15: Absolute error contour plot of NoDA (top), absolute error contour plot of DA by EnKF (center), and error ratio ϵ contour plot (bottom) of electron number density N_{e} for geomagnetic storm conditions at 16:00UT, October 29, 2003 (left) and 17:50UT, October 29, 2003 (right). N_{e}, T_{ion}, and ρ_{n} simulated measurements at 6 measurement points are used for data assimilation. Each measurement point is indicated by a dot. The region of data injection is indicated by dashed box which corresponds to 4 by 1 GITM blocks

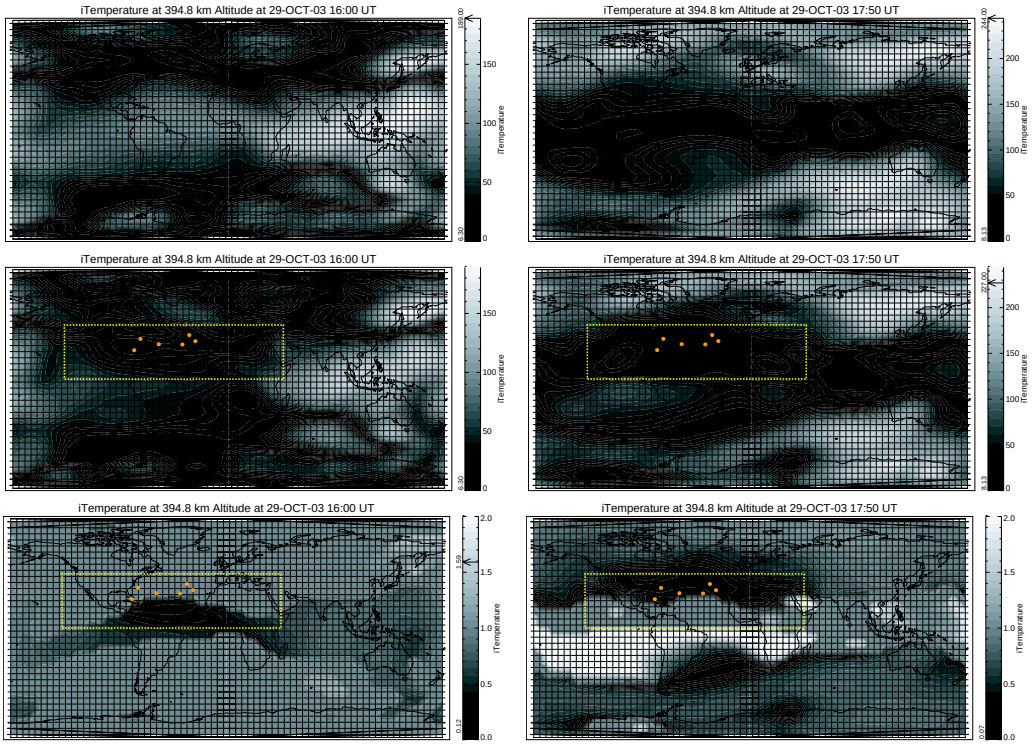


Figure 8.16: Absolute error contour plot of NoDA (top), absolute error contour plot of DA by EnKF (center), and error ratio ϵ contour plot (bottom) of ion temperature T_{ion} for geomagnetic storm conditions at 16:00UT, October 29, 2003 (left) and 17:50UT, October 29, 2003 (right). N_{e}, T_{ion}, and ρ_{n} simulated measurements at 6 measurement points are used for data assimilation. Each measurement point is indicated by a dot. The region of data injection is indicated by dashed box which corresponds to 4 by 1 GITM blocks

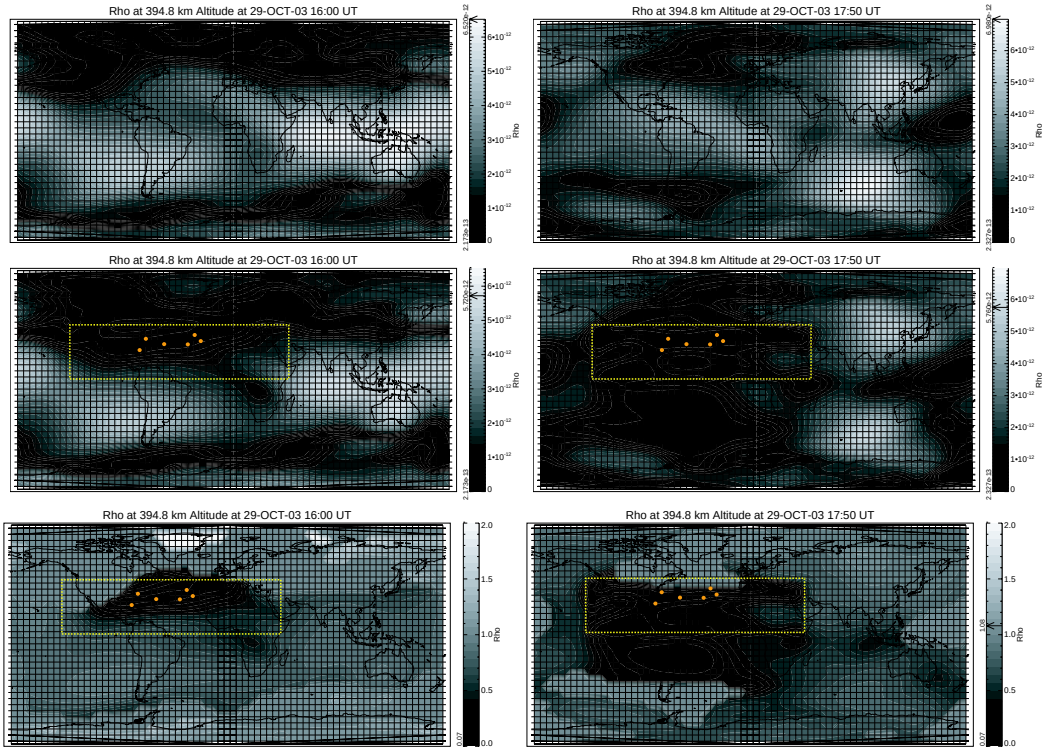


Figure 8.17: Absolute error contour plot of NoDA (top), absolute error contour plot of DA by EnKF (center), and error ratio ϵ contour plot (bottom) of neutral mass density ρ_{n} for geomagnetic storm conditions at 16:00UT, October 29, 2003 (left) and 17:50UT, October 29, 2003 (right). N_{e}, T_{ion}, and ρ_{n} simulated measurements at 6 measurement points are used for data assimilation. Each measurement point is indicated by a dot. The region of data injection is indicated by dashed box which corresponds to 4 by 1 GITM blocks

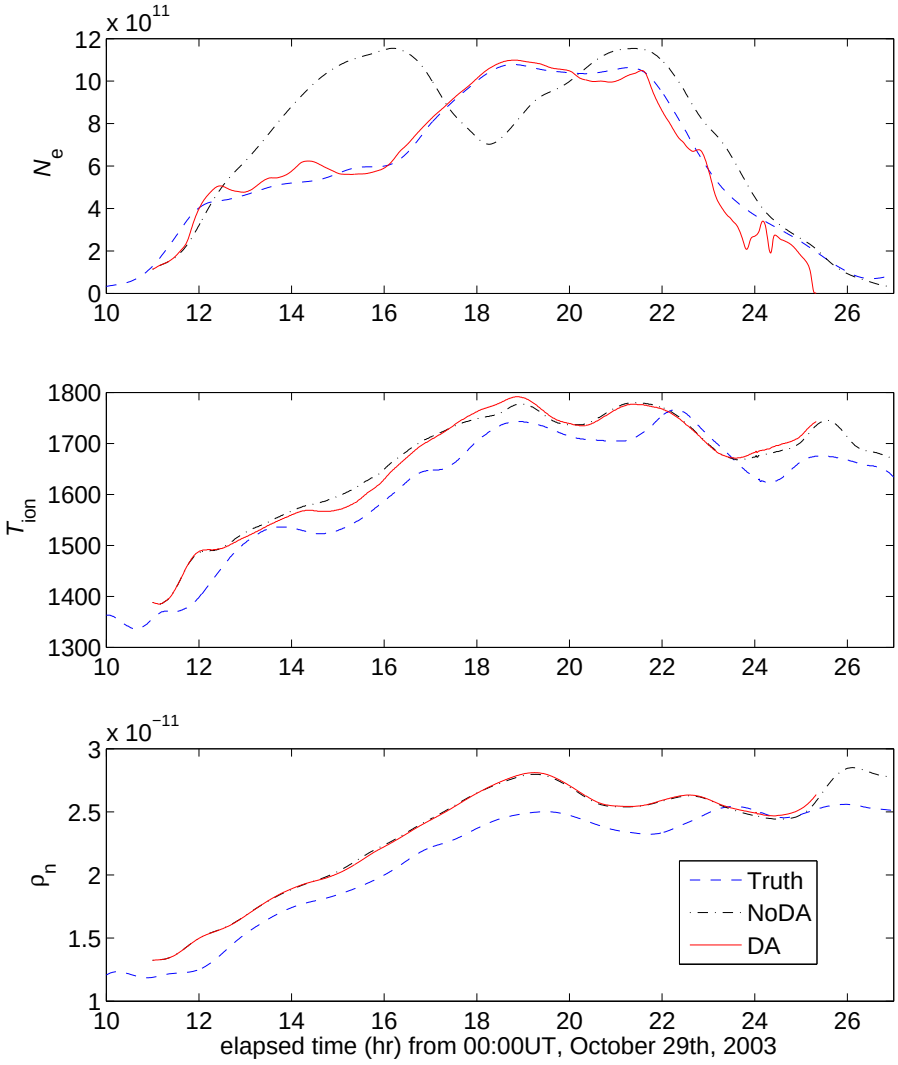


Figure 8.18: Comparison of states and outputs of truth, NoDA and DA by EnKF for geomagnetic storm conditions at m2 measurement location. Simulated TEC measurements at longitudinal-latitudinal locations of 6 measurement points are used. The size of region of data injection is 4 by 1 GITM blocks and given by Table 8.3. Only electron number density N_e is significantly affected by data assimilation.

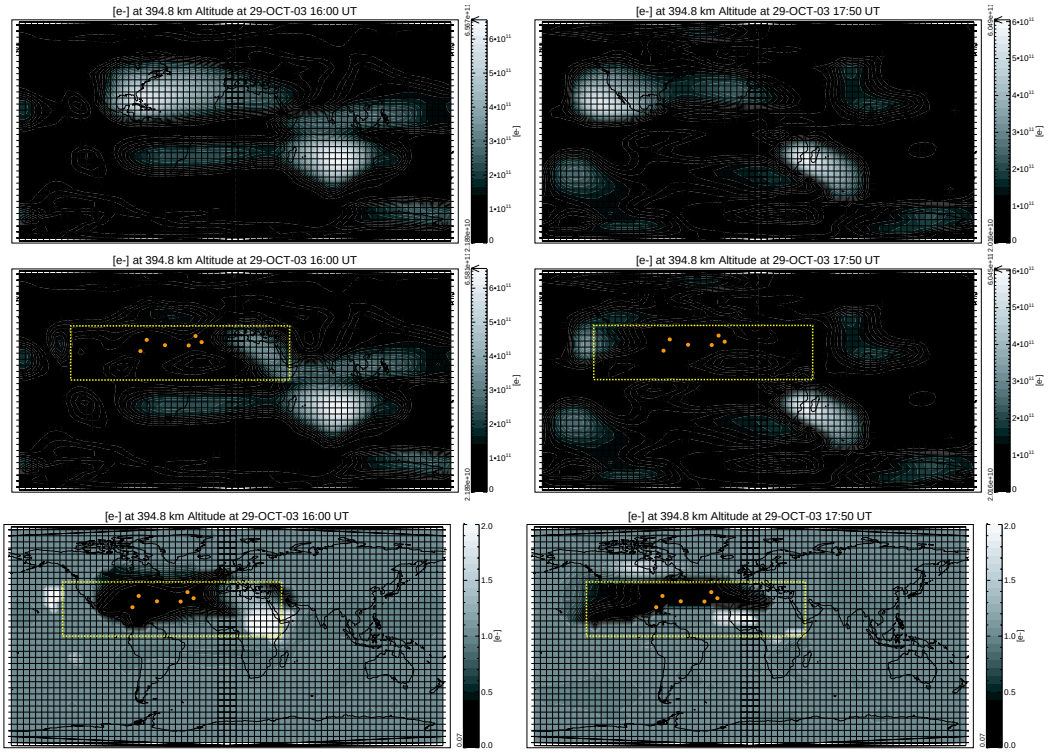


Figure 8.19: Absolute error contour plots of NoDA (top), absolute error contour plot of DA by EnKF (center), and error ratio ϵ contour plots (bottom) of electron number density N_e for geomagnetic storm conditions at 16:00UT, October 29, 2003 (left) and 17:50UT, October 29, 2003 (right). Simulated TEC at longitudinal-latitudinal locations of 6 measurement points are used for data assimilation. Each measurement location is indicated by a dot. The region of data injection is indicated by dashed box which corresponds to 4 by 1 GITM blocks.

CHAPTER IX

Conclusions

In this dissertation, we surveyed and developed data assimilation algorithms that are applicable to large scale nonlinear system. Specifically, we put high priority on developing the data assimilation algorithms for the ionosphere-thermosphere using the Global Ionosphere-Thermosphere Model (GITM).

In Chapter II, using finite-horizon optimization, we obtained the optimal reduced-order estimators and optimal fixed-structure subspace estimator in the form of recursive update equations for time-varying systems. These estimators are characterized by the oblique projectors. Moreover, we derived one-step and two-step update equations for each estimator. Even though the estimators reduce computational complexity in data injection, they still require full-order error-covariance propagation in order to be optimal.

Next, in Chapter III, noticing that the main computational complexity in large scale data assimilation is due to the propagation of the huge error covariance matrix, we presented several suboptimal reduced-order Kalman filters for discrete-time LTI systems based on reduced-order error-covariance propagation. These filters use combinations of balanced model truncation and complementary steady-state covariance compensation. We conducted numerical studies using a compartmental model for

two extreme cases of Hankel singular values. In both cases, localized Kalman filter with open loop or closed loop complementary steady-state covariance compensation showed good performance. When there are a few dominant Hankel singular values, localized Kalman filter with closed loop complementary steady-state covariance compensation can be applied efficiently without the need for a similarity transformation that may be prohibitive in large-scale systems.

In Chapter IV, we considered the unscented Kalman filter (UKF) as a large scale data assimilation method and presented extensions of the the unscented Kalman filter that propagate a reduced-order pseudo error covariance. To compensate for the neglected correlation between certain states and the measurement, we presented two methods that use a complementary static estimator gain based on correlations between the measurements and the neglected states, which are introduced in Chapter III using linear system. The use of a static estimator gain based on the open-loop and closed-loop correlations helps improve estimation performance without a significant increase in the online computational burden.

In Chapter V, we developed a Cholesky decomposition method to obtain reduced-rank square-root Kalman filters where the reduced-rank square-root of error covariance corresponds to the ensemble size reduction of UKF. We compared the Cholesky-based and the SVD-based reduced-rank square-root Kalman filters. In general cases, the Cholesky-based does not always perform better for estimation accuracy than the SVD-based and vice versa. However, using simulation examples, we showed that the Cholesky-based exhibits more computationally efficient and stable performance than the SVD-based filter, which can become unstable when the strong disturbances enter the system states that are not measured.

In Chapter VI, we used a localized, sampled-data update scheme with frozen-

intersample error covariance to reduce the computational complexity of the vertical 1D GITM data assimilation based on the unscented Kalman filter. We performed the numerical studies to obtain effective measurement locations and quantities for the sampled-data UKF, and then applied the sampled-data LUKF. The sampled-data LUKF with a small local region showed good estimation accuracy in much shorter computation time for data assimilation on the highly coupled vertical 1D GITM.

For the next step, we applied the localized UKF with approximate disturbance covariance to the 3 dimensional GITM. However, we were not able to obtain effective data assimilation results because the UKF turned out to be sensitive to the disturbance covariance. Hence, in Chapter VII, we examine ensemble Kalman filter and ensemble-on-demand Kalman filter that do not use explicit disturbance covariance. In this Chapter VII, we showed that ensemble-on-demand Kalman filter is computationally inexpensive but provides acceptable performance for systems under a single global-type disturbance. Furthermore, we demonstrated that ensemble Kalman filter is effective when we perturb the model using the well-identified disturbance statistics.

In Chapter VIII, we successfully applied EnODKF and EnKF with small ensemble size to the 3 dimensional GITM-based data assimilation, which means that full 3 dimensional GITM-based data assimilation yielded acceptable estimation accuracy with feasible computation speed. Specifically, EnODKF was used for solar EUV flux disturbance case whereas EnKF was used for geomagnetic storm condition. We used combinations of simulated measurements and injected data into a local region where correlations with measurements are good, while the poor correlations are set to zero in calculating the data-injection gain. We showed that the ion measurements and neutral measurements should be used together to provide good estimation accuracy over all states and outputs of the ionosphere-thermosphere in the region of interest.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans, ASME.J. of Basic Eng.*, 82:35–45, 1960.

[2] S. Odenwald. *The 23rd Cycle: Learning to Live with a Stormy Star*. Columbia University Press, 2001.

[3] G. A. Hajj, B. D. Wilson, C. Wang, X. Pi, and I. G. Rosen. Data assimilation of ground GPS total electron content into a physics-based ionospheric model by use of the Kalman filter. *Radio Science*, 39:RS1S05, February 2004. doi:10.1029/2002RS002859.

[4] Robert W. Schunk, Ludger Scherliess, Jan J. Sojka, Donald C. Thompson, David N. Anderson, Mihail Codrescu, Cliff Minter, Timothy J. Fuller-Rowell, Roderick A. Heelis, Marc Hairston, and Bruce M. Howe. Global assimilation of ionospheric measurements (GAIM). *Radio Science*, 39:RS1S02, 2004. doi:10.1029/2002RS002794.

[5] M. J. Angling. First assimilations of COSMIC radio occultation data into the Electron Density Assimilative Model (EDAM). *Annales Geophysicae*, 26:353–359, February 2008.

[6] J. J. Sojka, D. C. Thompson, R. W. Schunk, J. V. Eccles, J. J. Makela, M. C. Kelley, S. A. González, N. Aponte, and T. W. Bullett. Ionospheric data assimilation: recovery of strong mid-latitudinal density gradients. *Journal of Atmospheric and Solar-Terrestrial Physics*, 65:1087–1097, July 2003.

[7] Ch. Reigber, H. Luhr, and P. Schwintzer. CHAMP mission satus and perspectives. In *Suppl to EOS, Transactions, AGU, 81(48), F307*, 2000.

[8] B. Tapley, S. Bettadpur, M. Watkins, and Ch. Reigber. The gravity recovery and climate experiment: Mission overview and early results. *Geophys. Res. Lett.*, 31:L09607, 2004. doi:10.1029/2004GL019920.

[9] T. J. Fuller-Rowell, C. F. Minter, and M. V. Codrescu. Data assimilation for neutral thermospheric species during geomagnetic storms. *Radio Science*, 39:RS1S03, February 2004. 10.1029/2002RS002835.

[10] A. J. Ridley, Y. Deng, and G. Toth. The global ionosphere-thermosphere model. *J. Atmos. and Sol.-Terr. Phy.*, 68:839–864, 2006.

[11] Y. Deng, A. D. Richmond, and A. J. Ridley. Assessment of the non-hydrostatic effect on the upper atmosphere using a general circulation model (gcm). *Geophys. Res. Lett.*, 35:L01104, 2008. doi: 10.1029/2007GL032182.

[12] D. S. Bernstein and D. C. Hyland. The optimal projection equations for reduced-order state estimation. *IEEE Trans. Autom. Contr.*, AC-30:583–585, 1985.

[13] W. M. Haddad and D. S. Bernstein. Optimal reduced-order observer-estimators. *AIAA J. Guid. Contr. Dyn.*, 13:1126–1135, 1990.

[14] P. Hippe and C. Wurmthaler. Optimal reduced-order estimators in the frequency domain: The discrete-time case. *Int. J. Contr.*, 52:1051–1064, 1990.

[15] C. S. Hsieh. The unified structure of unbiased minimum-variance reduced-order filters. In *Proceedings of Contr. Dec. Conf*, pages 4871–4876, Dec. 2003.

[16] H. S. Hoang, R. Baraille, and O. Talagrand. On the design of a stable adaptive filter for state estimation in high dimensional systems. *Automatica*, 37(3):341–359, 2001.

[17] C. T. Leondes and L. M. Novak. Optimal minimal-order observers for discrete-time systems-a unified theory. *Automatica*, 8(4):379–387, 1972.

[18] Y. Halevi. The optimal reduced-order estimator for systems with singular measurement noise. *IEEE Trans. Autom. Contr.*, 34(7):777–781, 1989.

[19] R. P. Paroda and A. Kumar. Optimal reduced-order estimator design for discrete, time-varying systems. In *IEEE International Conference on Systems Engineering*, pages 367–370, Aug. 1991.

[20] D. Haessig and B. Friedland. Separate-bias estimation with reduced-order kalman filters. *IEEE Trans. Autom. Contr.*, 43(7):983–987, 1998.

[21] C. S. Hsieh and F. C. Chen. Optimal minimal-order least-squares estimators via the general two-stage kalman filter. *IEEE Trans. Autom. Contr.*, 46(11):1772–1776, 2001.

[22] K. W. Corscadden and S. R. Duncan. Reduced-order estimator for closed-loop online estimation of cross-directional parameters in a plastics extrusion process. In *IEE Proceedings of Control Theory and Applications*, volume 144, pages 549–557, 1997.

[23] M. Barut, O. S. Bogosyan, and M. Gokasan. An ekf based reduced order estimator for the sensorless control of ims. In *Proceedings of 2003 IEEE Conference on Control Applications*, volume 2, pages 1256–1261, 2003.

[24] S. Roy and P. Mookerjee. Hierarchical estimation with reduced order local observers. In *Proceedings of Contr. Dec. Conf*, volume 1, pages 835–836, Dec. 1989.

[25] A. Azemi, E. Yaz, and K. Olejniczak. Reduced-order estimation of power system harmonics. In *Proceedings of the 4th IEEE Conference on Control Applications*, pages 631–636, 1995.

[26] D. A. Haessig and B. Friedland. Nonlinear reduced-order state and parameter observer. In *Proceedings of the 2001 American Control Conference*, volume 3, pages 1978–1980, June 2001.

[27] B. F. Farrel and P. J. Ioannou. State estimation using a reduced-order kalman filter. *J. of the Atmospheric Sciences*, 58:3666–3680, 2001.

[28] A. Gelb. *Applied Optimal Estimation*. The M.I.T Press, 1974.

[29] C. P. Mracek, J. R. Cloutier, and C. A. D'Souza. A new technique for nonlinear estimation. In *Proc. Int. Conf. Contr. App.*, pages 338–343, June 1996. Dearborn, MI.

[30] J. Chandrasekar, A. Ridley, and D. S. Bernstein. A sdre-based asymptotic observer for nonlinear discrete-time systems. In *Proc. Amer. Cont. Conf.*, volume 5, pages 3630–3635, 2005.

[31] I. Kim, J. Chandrasekar, A. Ridley, and D. S. Bernstein. Data assimilation using the global ionosphere-thermosphere model. In *Proc. ICCS*, pages 489–496, May 2006. Reading, UK.

[32] G. Evensen. Sequential data assimilation with a nonlinear quasi-geotrophic model using monte carlo methods to forecast error statistics. *Geographical Research*, 99(C5):10143–10162, 1994.

[33] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. In *Proc. of the IEEE*, volume 92, pages 401–422, March 2004.

[34] M. Lewis, S. Lakshmivarahan, and S. Dhall. *Dynamic Data Assimilation: A Least Squares Approach*. Cambridge, 2006.

[35] D. S. Bernstein and D. C. Hyland. The optimal projection equations for reduced-order state estimation. *IEEE Trans. Autom. Contr.*, AC-30:583–585, 1985.

[36] P. Hippe and C. Wurmthaler. Optimal reduced-order estimators in the frequency domain: The discrete-time case. *Int. J. Contr.*, 52:1051–1064, 1990.

[37] W. M. Haddad and D. S. Bernstein. Optimal reduced-order observer-estimators. *AIAA J. Guid. Contr. Dyn.*, 13:1126–1135, 1990.

[38] L. Scherliess, R. W. Schunk, J. J. Sojka, and D. C. Thompson. Development of a physics-based reduced state kalman filter for the ionosphere. *Radio Science*, 39-RS1S04, 2004.

[39] S. F. Farrell and P. J. Ioannou. State estimation using a reduced-order kalman filter. *J. Atmos. Sci.*, 58:3666–3680, 2001.

[40] 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 nino. *J. Climate*, 14:1720–1737, 2001.

[41] I. S. Kim, J. Chandrasekar, H. J. Palanthandalam-Madapusi, A. Ridley, and D. S. Bernstein. State estimation for large-scale systems based on reduced-order error-covariance propagation. In *Proc. of the IEEE 30th Annual Northeast Bioengineering Conf.*, pages 5700–5705, June 2007.

[42] J. Chandrasekar, O. Barerro, B. De Moor, and D. S. Bernstein. Kalman filtering with constrained output injection. *Int. J. Contr.*, 80:1863–1879, 2007.

[43] D. S. Bernstein, L. D. Davis, and D. C. Hyland. The optimal projection equations for reduced-order, discrete-time modelling, estimation and control. *Int. J. Contr.*, 9:288–293, 1986.

[44] W. M. Haddad, D. S. Bernstein, and V. Kapila. Reduced-order multirate estimation. *AIAA J. Guid. Contr. Dyn.*, 17:712–721, 1994.

[45] A. Johnson. Discrete and sampled-data stochastic control problems with complete and incomplete state information. *Applied Mathematics and Optimization*, 24:289–316, 1991.

[46] W. M. Haddad and D. S. Bernstein. Optimal reduced-order subspace-observer design with a frequency-domain error bound. *Advances in Control and Dynamic Systems*, 2(2):23–38, 1990.

[47] D. S. Bernstein. *Matrix Mathematics*. Princeton University Press, 2005.

[48] B. F. Farrell and P. J. Ioannou. *Predictability of weather and climate*. Cambridge University Press, July 2006.

[49] A. W. Heemink, M. Verlaan, and A. J. Segers. Variance-reduced ensemble kalman filtering. *Mon. Weather Rev.*, 129:1718–1728, 2001.

[50] A. J. Busalacchi, J. Ballabrera-Poy, and R. Murtugudde. Application of a reduced-order kalman filter to initialize a coupled atmosphere-ocean model: Impact on the prediction of el nino. *J. Climate*, 14:1720–1737, 2001.

[51] D. Menemenlis, P. Fieguth, and I. Fukumori. Mapping and pseudo-inverse algorithms for data assimilation. In *Int. Geoscience Remote Sensing Symp.*, pages 3221–3223, 2002.

[52] A. Segers. *Data Assimilation in atmospheric chemistry models using Kalman filtering*. DUP Science, 2002.

[53] J. M. Lewis, S. Lakshmivarahan, and Sudarshan Dhall. *Dynamic Data Assimilation: A Least Squares Approach*. Cambridge University Press, 2006.

[54] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, Society for Industrial and Applied Mathematics, 2005.

[55] J. Chandrasekar, I. S. Kim, A. Ridley, and D. S. Bernstein. A localized unscented kalman filter based on reduced-order-error-covariance propagation and complementary steady-state correlation. In *Proc. of Amer. Contr. Conf.*, June 2007.

[56] D. S. Bernstein and D. C. Hyland. Compartmental modeling and second-moment analysis of state space systems. *SIAM J. Matrix Anal. Appl.*, 14(3):880–901, 1993.

[57] Y. K. Sasaki and J. S. Goerss. Satellite data assimilation using nasa data systems test 6 observations. *Mon. Wea. Rev.*, 110:16351644, 1982.

[58] J. A. Carton, G. Chepurin, , and X. Cao. A simple ocean data assimilation analysis of the global upper ocean 195095. part i: Methodology. *J. Phy. Ocean*, 30:294309, 1999.

[59] W. M. Haddad and D. S. Bernstein. Optimal reduced-order observer-estimators. *AIAA J. Guid. Dyn. Contr.*, 13:1126–1135, 1990.

[60] B. F. Farrell and P. J. Ioanannou. Optimal reduced-order observer-estimators. *J. Atmo. Sci.*, 58:3666–3658, 2001.

[61] I. S. Kim, J. Chandrasekar, H. J. Palanthandalam, A. Ridley, and D. S. Bernstein. State estimation for large-scale systems based on reduced-order error-covariance propagation. In *submitted Amer. Contr. Conf*, 2007.

[62] S. Julier. The scaled unscented transformation. In *Proc. Amer. Cont. Conf.*, pages 4555–4559, May 2002.

[63] G. Toth, A. Ridley, K. Powell, and T. Gambosi. A high-performance framework for sun-to-earth space weather modeling. In *Parallel and Distributed Processing Symposium*, 2005.

[64] J. Chandrasekar, A. Ridley, and D. S. Bernstein. A comparison of the extended and unscented kalman filters for discrete-time systems with nondifferentiable dynamics. In *submitted Amer. Contr. Conf*, 2007.

[65] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Dover Publications Inc., Mineola, NY, 1979.

[66] M. H. Rees. *Physics and Chemistry of the Upper Atmosphere*. Cambridge University Press, 1989.

[67] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer, 2006.

[68] C. S. Hsieh. The unified structure of unbiased minimum-variance reduced-order filters. In *Proc. Contr. Dec. Conf.*, pages 4871–4876, December 2003. Maui, HI.

[69] P. Fieguth, D. Menemenlis, and I. Fukumori. Mapping and pseudo-inverse algorithms for data assimilation. In *Proc. Int. Geoscience Remote Sensing Symp.*, pages 3221–3223, 2002.

[70] M. Verlaan and A. W. Heemink. Tidal flow forecasting using reduced rank square root filters. *Stochastics Hydrology and Hydraulics*, 11:349–368, 1997.

[71] S. Gillijns, D. S. Bernstein, and B. D. Moor. The reduced rank transform square root filter for data assimilation. In *Proc. of the 14th IFAC Symposium on System Identification (SYSID2006)*, volume 11, pages 349–368, 2006.

[72] A. W. Heemink, M. Verlaan, and A. J. Segers. Variance reduced ensemble kalman filtering. *Monthly Weather Review*, 129:1718–1728, 2001.

[73] D. Treebushny and H. Madsen. A new reduced rank square root kalman filter for data assimilation in mathematical models. *Lecture Notes in Computer Science*, 2657:482–491, 2003.

[74] J. L. Anderson. An ensemble adjustment kalman filter for data assimilation. *Monthly Weather Review*, 129:2884–2903, 2001.

[75] 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.

[76] G. J. Bierman. *Factorization Methods for Discrete Sequential Estimation*. reprinted by Dover, 2006.

[77] M. Morf and T. Kailath. Square-root algorithms for least-squares estimation. *IEEE Trans. Autom. Contr.*, AC-20:487–497, 1975.

[78] G. W. Stewart. *Matrix Algorithms Volume 1: Basic Decompositions*. SIAM, 1998.

[79] C. H. Bishop and Z. Toth. Ensemble transformation and adaptive observations. *Monthly Weather Rev.*, 56:1748–1765, 1999.

[80] T. M. Hamill and C. Snyder. Using improved background-error covariances from an ensemble kalman filter for adaptive observations. *Monthly Weather Rev.*, 130:1552–1572, 2002.

[81] S. J. Majumdar, C. H. Bishop, R. Buizza, and R. Gelaro. A comparison of ensemble transform kalman filter targeting guidance with ECMWF and NRL total-energy singular vector guidance. *Quart. J. Roy. Meteor. Soc.*, 128:2527–2549, 2002.

[82] C. H. Bishop, B. Etherton, and S. J. Majumdar. Adaptive sampling with the ensemble transform kalman filter, part 1: Theoretical aspects. *Monthly Weather Rev.*, 129:420–436, 2001.

[83] S. Gillijns, O. Barrero Mendoza, J. Chandrasekar, B. De Moor, D. S. Bernstein, and A. Ridley. What is the ensemble kalman filter and how well does it work? In *Proc. Amer. Contr. Conf.*, pages 4448–4453, June 2006. Minneapolis, MN.

[84] G. Evensen. Advanced data assimilation for strongly nonlinear dynamics. *Monthly Weather Rev.*, 125:1342–1354, 1997.

[85] A. Doucet, N. de Freitas, and N. J. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

[86] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter, particle filters for tracking applications*. Artech House, Boston, 2004.

[87] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, Cambridge, MA, 2005.

[88] J. Chandrasekar, A. Ridley, and D. S. Bernstein. A comparison of the extended and unscented kalman filters for discrete-time systems with nondifferentiable dynamics. In *Proc. Amer. Contr. Conf.*, pages 4431–4436, July 2007. New York, NY.

[89] J. Chandrasekar, A. Ridley, and D. S. Bernstein. An sdre-based asymptotic observer for nonlinear discrete-time systems. In *Proc. Amer. Contr. Conf.*, pages 3630–3635, June 2005. Portland, OR.

[90] J. Chandrasekar, I. S. Kim, A. J. Ridley, and D. S. Bernstein. Reduced-order covariance-based unscented kalman filtering with complementary steady-state correlation. In *Proc. Amer. Contr. Conf.*, pages 4452–4457, July 2007. New York, NY.

[91] I. S. Kim, J. Chandrasekar, H. J. Palanthandalam-Madapusi, A. J. Ridley, and D. S. Bernstein. State estimation for large-scale systems based on reduced-order error-covariance propagation. In *Proc. Amer. Contr. Conf.*, pages 5700–5705, July 2007. New York, NY.

[92] D. P. Dee. On-line estimation of error covariance paramaters for atmospheric data assimilation. *Monthly Weather Rev.*, 123:1128–1145, 1995.

[93] T. Penduff, P. Brasseur, C. E. Testut, B. Barnier, and J. Verron. A four-year eddy-permitting assimilation of sea-surface temperature and altimetric data in the south atlantic ocean. *J. of Marine Res.*, 60:805–833, 2002.

[94] C. E. Testut, P. Brasseur, J. M. Brankart, and J. Verron. Assimilation of sea-surface temperature and altimetric observations during 1992-1993 into an eddy permitting primitive equation model of the north atlantic ocean. *J. of Marine Syst.*, 40(41):291–316, 2003.

[95] A. J. Ridley, Y. Deng, and G. Toth. The global ionosphere thermosphere model. *Journal of Atmospheric and Solar-Terrestrial Physics*, 68:839–864, 2006.

[96] A. Hedin. Msis-86 thermospheric model. *J. Geophys. Res.*, 92:4649, 1987.

[97] D. Bilitza. International reference ionosphere 2000. *Radio Science*, 36:261, 2001.

[98] A. Richmond. Ionospheric electrodynamics using magnetic apex coordinates. *J. Geomagn. Geoelectr.*, 47:191, 1995.

[99] D. R. Weimer. A flexible, IMF dependent model of high latitude electric potential having space weather applications. *J. Geophys. Res. Lett.*, 23:2549, 1996.

[100] A. D. Richmond and Y. Kamide. Mapping electrodynamic features of the high-latitude ionosphere from localized observations:technique. *J. Geophys. Res.*, 93:5741, 1988.

[101] T. N. Woods, F. G. Eparvier, S. M. Bailey, P. C. Chamberlin, J. Lean, G. J. Rottman, S. C. Solomon, W. K. Tobiska, and D. L. Woodraska. Solar euv experiment (SEE): Mission overview and first results. *J. Geophys. Res.*, 110(A01312, doi:10.1029/2004JA010765), 2005.

[102] M. Brio and C. C. Wu. An upwind differencing scheme for the equations of ideal magnetohydrodynamics. *J. Comp. Phy.*, 75:400–422, 1988.

[103] S. C. O. Glover and M. M. Low. Simulating the formation of molecular clouds. ii. rapid formation from turbulent initial conditions. *The Astrophysical Journal*, 659:1317–1337, 2007.

[104] K. Suvanto. Analysis of incoherent scatter radar data from non-thermal f-region plasma. *Journal of Atmospheric and Terrestrial Physics*, 51(6):483–495, 1989.

[105] G. R. Wilson, D. R. Weimenr, J. O. Wise, and F. A. Marcos. Response of the thermosphere to joule heating and particle precipitation. *J. Geophys. Res.*, 111(A10314, doi:10.1029/2005JA011274), 2006.

[106] J. L. Anderson and S. L. Anderson. A monte carlo implementation of the nonlinear filtering problem to produce ensemble assimilation and forecasts. *Monthly Weather Rev.*, 127:2741–2758, 1999.

[107] J. M. Brankart, C. E. Testut, P. Brasseur, and J. Verron. Implementation of a multivariate data assimilation scheme for isopycnic ocean models: Application to a 1993-1996 hindcast of the north atlantic ocean circulation. *J. Geophys. Res.*, 108(C3):1–20, 2003.

[108] J. Verron, L. Gourdeau, D. T. Pham, R. Murtugudde, and A. Busalacchi. An extended kalman filter to assimilate satellite altimeter data into a nonlinear numerical model of the tropical pacific: Method and validation. *J. Geophys. Res.*, 104:5441–5458, 1999.

[109] N. Balan, G. J. Bailey, B. Jenkins, P. B. Rao, and R. J. Moffett. Variation of ionospheric ionization and related solar fluxes during an intense solar cycle. *J. Geophys. Res.*, 99(A2):2243–2254, 2004.

[110] J. Leia, L. Liua, W. Wana, and S. Zhangc. Modeling the behavior of ionosphere above millstone hill during the september 21-27, 1998 storm. *J. of Atmospheric and Solar-Terrestrial Physics*, 66:1093–1102, 2004.

[111] P. L. Houtekamer and H. L. Mitchel. A sequential ensenble kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129:796–811, 2001.

[112] L. Nerger, S. Danilov, G. Kivman, W. Hiller, and J. Schroter. Data assimilation with the ensemble kalman filter and the seik filter applied to a finite element model of the north atlantic. *Marine Systems*, 65:288–298, 2007.

[113] P. Moireau, D. Chapelle, and P. L. Tallec. Joint state and parameter estimation for distributed mechanical systems. *Comput. Methods Appl. Mech. Engrg.*, 197:659–677, 2008.

[114] D. T. Pham. Stochastic methods for sequential data assimilation in strongly nonlinear systems. *Monthly Weather Rev.*, 129:1194–1207, 2000.

[115] S. Sarkka. On unscented kalman filtering for state estimation of continuous-time nonlinear systems. *IEEE Trans. Automatic Contr.*, 52(9):1631–1641, 2007.

[116] J. Lastovicka. Monitoring and forecasting of ionospheric space weather-effects of geomagnetic storms. *Journal of Atmospheric and Solar-Terrestrial Physics*, 64:697–705, 2002.

[117] R.W. Schunk, L. Scherliess, and J. J. Sojka. Recent approaches tomodeling ionospheric weather. *Advances in Space Research*, 31:819–828, 2003.

[118] G. Tóth, I. V. Sokolov, T. I. Gombosi, D. R. Chesney, C.R. Clauer, D. L. De Zeeuw, K. C. Hansen, K. J. Kane, W. B. Manchester, K. G. Powell, A. J. Ridley, I. I. Roussev, Q. F. Stout, O. Volberg, Richard A. Wolf, S. Sazykin, A. Chan, Bin Yu, and József Kóta. Space weather modeling framework: A new tool for the space science community. *J. Geophys. Res.*, 110, 2005. Submitted.

[119] A. J. Ridley, Y. Deng, and G. Toth. The global ionosphere-thermosphere model. *J. Atmos. Sol-Terr. Phys.*, 68:839, 2006.

[120] A. E. Hedin. MSIS-86 thermospheric model. *J. Geophys. Res.*, 92:4649, 1987.

[121] D. Bilitza. International reference ionosphere 2000. *Radio Science*, 36(2):261–275, 2001.

[122] P. G. Richards, J. A. Fennelly, and D. G. Torr. EUVAC: A solar EUV flux model for aeronomic calculations. *J. Geophys. Res.*, 99(A5):8981–8992, 1994.

[123] W. K. Tobiska. Revised solar extreme ultraviolet flux model. *J. Atm. Terr. Phys.*, 53:1005, 1991.

[124] A. D. Richmond. Assimilative mapping of ionospheric electrodynamics. *Adv. Space Res.*, 12:59, 1992.

[125] T. J. Fuller-Rowell and D.S. Evans. Height-integrated Pedersen and Hall conductivity patterns inferred from TIROS–NOAA satellite data. *J. Geophys. Res.*, 92:7606, 1987.

[126] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *jgr*, 99:10143–10162, May 1994.

[127] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proc. of the IEEE*, 92:401–422, March 2004.

[128] I. S. Kim, D. J. Pawlowski, A. J. Ridley, and D. S. Bernstein. Localized data assimilation in the ionosphere-thermosphere using a sampled-data unscented kalman filter. In *Proc. Amer. Contr. Conf.*, number WeC18.2, June 2008. Seatle, WA.

[129] D. T. Pham. Stochastic methods for seuential data assimilation in strongly non-linear systems. *Monthly Weather Review*, 129:1194–1207, 2000.

[130] I. S. Kim and D. S. Bernstein. Ensemble-On-Demand Kalman Filter for Large-Scale Systems with Time-Sparse Measurements. *submitted to IEEE Conf. Dec. Contr.*, 2008.

[131] B. T. Tsurutani, F. L. Guarnier, T. Fuller-Rowell, A. J. Mannucci, B. Iijima, W. D. Gonzalez, D. L. Judge, P. Gangopadhyay, A. Saito, T. Tsuda, O. P. Verkhoglyadova, and G. A. Zambon. Extreme solar EUV flares and ICMEs and resultant extreme ionospheric effects: Comparison of the Halloween 2003 and the Bastille Day events. *RADIO SCIENCE*, 41:RS5S07, doi:10.1029/2005RS003331, 2006.