Reduced-Complexity Algorithms for Data Assimilation of Large-Scale Systems

by

Jaganath Chandrasekar

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 Harris N. McClamroch
Professor Pierre T. Kabamba

Karmanye Vadhi Karasthe Maa Phaleshu Kadachana,
Maa Karmaphal Hetur Bhurma, Te Sanghastva Akarmani

- Bhagavad Gita

To Amma and Daddy, the beginning.
To my advisor, Dennis Bernstein, the middle.
To Sweta, the end.

ACKNOWLEDGEMENTS

I would like to thank Mr N. Sivaramakrishnan for his valuable insights and inputs. I would also like to thank my sister Charulata. Thanks to Shaq, Shrav, TT, Ravi, Unni and Bharti for cheering me up. I would also like to thank Suki and Giridhar for their support.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF SYMBOLS

$\triangleq$	equals by definition
$\mathbb{R}$	real numbers
$\lvert x \rvert$	absolute value of $x \in \mathbb{R}$
$\mathbb{R}^{n \times m}$	$n \times m$ real matrices
$\operatorname{rank}(A)$	rank of $A \in \mathbb{R}^{n \times m}$
A^{T}	transpose of $A \in \mathbb{R}^{n \times m}$
A^{-1}	inverse of $A \in \mathbb{R}^{n \times n}$
$\lVert x \rVert_2$	Eucledian norm $\sqrt{x^{\mathrm{T}} x}$
$\lVert A \rVert_{\mathrm{F}}$	Frobenius norm $\sqrt{\operatorname{tr}(A A^{\mathrm{T}})}$
$\mathcal{E}[X]$	expected value of $X \in \mathbb{R}^{n \times m}$
$\operatorname{diag}(a_1, \ldots, a_n)$	diagonal matrix $\begin{bmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{bmatrix}$

CHAPTER I

Introduction

Data assimilation refers to the process of using measurement data along with model information to estimate the value of a certain variable. We come across various data assimilation applications in our daily life. For example, before crossing a road, we estimate the speed of oncoming vehicles by using visual images of their position at different instances in time. These visual images serve as measurements, while our knowledge relating quick changes in the position to greater speeds serves as the model. GPS systems use estimation algorithms to determine the location of GPS receivers using signals from GPS satellites. In many feedback control applications, whenever the exact value of a feedback variable is unknown, controllers use an estimate of that variable for feedback. Hence, the performance and stability of the controller depends on the accuracy of the estimates. For example, guidance and navigation algorithms in satellites and spacecraft use critical orbital parameters that are obtained using estimation algorithms. Terrestrial weather agencies use estimation algorithms that run on supercomputers to predict the daily weather and issue critical meteorological warnings. Finally, estimation algorithms are used as fault diagnostic tools in fuel cell monitoring and many industrial applications like semiconductor manufacturing.

There are many ways to estimate an unknown quantity using available data. Most of these estimation techniques use either a deterministic or statistical framework for estimation, that is, the unknown variable x is assumed to be either a random quantity or a deterministic variable. Most estimation techniques use a model framework to capture the relationship between the available measurements y, the unknown variable x, and the model parameters and known inputs. Finally, many estimation techniques involve minimizing a certain performance criteria. Specifically, if $\hat{x}$ is an estimate of x, so that the error in the estimate is given by $x - \hat{x}$, then the objective of most estimation algorithms is to obtain an estimate $\hat{x}$ that results in a small magnitude of the error $x - \hat{x}$.

One of the earliest estimation techniques, the least-squares method, was developed by Carl Friedrich Gauss in 1809. Consider a static model

$$y = u^{\mathrm{T}} x,$$

where x is the unknown variable, u contains the known inputs and model parameters, and y is the available measurement. Assume that n measurements, $y_1, \ldots, y_n$, corresponding to n inputs $u_1, \ldots, u_n$ are available so that

$$Y_n = U_n x,$$

where

$$Y_n \triangleq \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}^{\mathrm{T}}, \quad U_n \triangleq \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix}^{\mathrm{T}}.$$

Assuming $U_n^{\mathrm{T}} U_n$ is invertible, the estimate $\hat{x}$ that minimizes

$$J_{\mathrm{LS}} \triangleq \|Y_n - U_n \hat{x}\|_2,$$

is given by $\hat{x}_n = \hat{x}_{\mathrm{LS},n}$, where

$$\hat{x}_{\mathrm{LS},n} = (U_n^{\mathrm{T}} U_n)^{-1} U_n^{\mathrm{T}} Y_n.$$

The least-squares technique is still widely used for estimation because of its simplicity [1–3]. The subscript n in $\hat{x}_{\mathrm{LS},n}$ denotes the fact that $\hat{x}_{\mathrm{LS},n}$ is the best estimate obtained using n measurements and input data, Y_n and U_n, respectively. Whenever a new measurement y_{n+1} and input value u_{n+1} are available, the new measurement and input value are appended to Y_n and U_n, and a new least-squares estimate $\hat{x}_{\mathrm{LS},n+1}$ can be obtained. However, when the number of measurements n becomes large, the size of U_n increases, and constructing $U_n^{\mathrm{T}}U_n$ is computationally expensive. Alternatively, the recursive-least-squares (RLS) procedure can be used to improve the least-squares estimate of x by updating the previously obtained least-squares estimate using only the new set of measurements [4]. RLS is a computationally efficient procedure for incorporating new measurements to improve prior state estimates.

In many cases, the relationship between the input u, the unknown variable x, and the measurement y, is more complicated. Furthermore, all the inputs that affect the model are not known, and sensors that produce measurements are inherently noisy. One simple framework that models such a scenario is the linear Gauss-Markov model given by the following dynamical system

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k + w_k, \quad k \geqslant 0 \\ y_k &= C_k x_k + v_k, \end{aligned}$$

where k indicates the time step, x_k is the unknown random variable, u_k is the known input, y_k is the measured output, w_k is the unknown external disturbance affecting the plant, v_k is the sensor noise, and A_k, B_k, and C_k are matrices containing known model parameters. A number of systems are modeled by the linear Gauss-Markov model. For example, consider rigid-body motion governed by Newton's second law. The state x comprises of the position and velocity of the body, while inputs u and

w denote known and unknown forces acting on the body, and A_k and B_k contains physical parameters like the mass of the body. Often, the dynamics of nonlinear systems like an aeroplane in flight is linearized about a mean trajectory, and a linear model is used. In this case, the state x contains altitude and pitch deviations from the nominal trajectory, whereas w denotes unknown forces acting on the aeroplane, like turbulence effects.

The objective of *state estimation* is to obtain estimates of the state x_k using measurements y_k. If $w_k \equiv 0$ and x_0 is known, then the estimator

$$\hat{x}_{k+1} = A_k\hat{x}_k + B_k u_k, \quad k \geqslant 0$$

with $\hat{x}_0 = x_0$ yields $\hat{x}_k = x_k$ for all $k \geqslant 0$. Hence, if all the inputs to a dynamical system and the value of the initial state are known, exact estimates of the state can be obtained without using any measurement data. However, since there are always external disturbances affecting the plant, generally $w_k \neq 0$ and since direct measurements of the state x are unavailable, one generally has only a poor estimate of the initial state. In this case, the measurement y_k is used along with model information to obtain better estimates of the state x_k. The use of measurement data and model information to obtain better estimates of the state is referred to as *data assimilation*.

A linear estimator has the structure

$$\begin{aligned} \hat{x}_{k+1} &= A_k\hat{x}_k + B_k u_k + K_k(y_k - \hat{y}_k), \quad k \geqslant 0 \\ \hat{y}_k &= C_k\hat{x}_k, \end{aligned}$$

where K_k is the estimator gain that injects the difference between the measured data and estimated measurement to improve the state estimates. If w_k and v_k are zero-mean normally distributed white noise, the Kalman filter provides optimal estimate

of the state x_k [5, 6]. The Kalman filter is a linear estimator with a special estimator gain. Specifically, in the Kalman filter, K_k depends on the error covariance P_k defined by

$$P_k \triangleq \mathcal{E}\left[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^{\mathrm{T}}\right].$$

Therefore, in order to provide optimal estimates of the state x_k at every time step k, the Kalman filter updates the error covariance P_k using the Riccati equation

$$P_{k+1} = (A - K_k C_k) P_k (A_k - K_k C_k)^{\mathrm{T}} + K_k R_k K_k^{\mathrm{T}} + Q_k,$$

where Q_k and R_k are the variances of w_k and v_k. For low-dimensional systems, the Kalman filter is a simple and efficient tool to obtain optimal state estimates. Owing to its simplicity, the Kalman filter has been used in a number of applications ranging from econometric analysis to the Apollo missions.

When the order of the dynamical system is high, for example, the dimension of x_k can be greater than 10^5 in terrestrial weather and ocean-climate models, implementing the Kalman filter is computationally intractable. Various extensions of the Kalman filter have been developed to address these computational issues. In many cases, estimates of only a certain subset of the state are required, and one approach that is employed in such a case is the reduced-order estimator. In these reduced-complexity estimators, a reduced-order model of the dynamics is used to propagate the state estimates instead of the full-order model. In [7, 8], a projection process is used to obtain the optimal reduced-order estimator dynamics, while the full-order dynamics are used to propagate the error covariance. Hence, although the computational burden of updating the state estimates is less, covariance propagation remains a computationally demanding task.

Alternatively, reduced-order estimators that use a reduced-order covariance are developed in [9]. In these estimators, model-reduction is first performed using various techniques like truncation and balancing, and an estimator is designed using the reduced-order model. Although such a construction does not yield optimal reduced-order estimators, the computational advantage of propagating a reduced-order covariance outweighs the degradation in performance.

Next, consider the following system with nonlinear dynamics and measurement map

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, w_k, k), \quad k \geqslant 0 \\ y_k &= h(x_k, v_k, k). \end{aligned}$$

The Kalman filter provides optimal estimates only when the dynamics and measurement map are linear. Estimators for nonlinear systems are an area of active research [10–13]. Optimal estimators for nonlinear systems are usually infinite-dimensional and cannot be easily implemented. Furthermore, propagating the error covariance of nonlinear estimators is difficult even for scalar nonlinear systems [10, 12]. However, a number of suboptimal techniques are used to deal with nonlinear systems. Amongst these, the extended Kalman filter and SDRE filter are some of the most simple approaches to nonlinear state estimation [14, 15]. In these extensions of the Kalman filter, the estimator state is propagated using the nonlinear model

$$\begin{aligned} \hat{x}_{k+1} &= f(\hat{x}_k, u_k, 0, k) + K_k(y_k - \hat{y}_k), \quad k \geqslant 0 \\ \hat{y}_k &= h(\hat{x}_k, 0, k). \end{aligned}$$

The estimator gain depends on the pseudo-error covariance that is propagated using the Riccati equation with either the Jacobians of the dynamics and measurement

maps or state-dependent factorizations taking the place of A_k and C_k. Although these estimators are not optimal, they have been used successfully in a number of areas.

Since these filters are extension of the Kalman filter, they suffer from the same computational disadvantages when used for large-scale systems. Moreover, since the dynamics are nonlinear, the projection and balancing techniques used for linear systems cannot be used to obtain a reduced-order model. Furthermore, in systems based on spatially distributed models or spatially discretized partial differential equations, for example, such systems arise in weather forecasting and atmospheric applications, it is difficult to obtain the Jacobian or a parametrization of the nonlinear dynamics.

Another approach to state estimation of nonlinear systems involves running multiple copies of the model in parallel. Such techniques are commonly referred to as particle filters [16]. In particle filters, the Kalman filter estimator gain expression is used for data injection. However, the error covariance is calculated from the collection of state estimates instead of the Riccati equation. The ensemble Kalman filter, developed in [17], injects randomly generated noise into multiple copies of the model to simulate the effect of the external disturbance w_k on the plant dynamics. In [18, 19], a deterministic approach is used to generate the collection of state estimates. Specifically, the columns of the pseudo-error covariance matrix is used to re-initialize the multiple copies of the model at every time step. In all the variations of the particle filter, the ensemble size, that is, the number of copies of the model, determines the computational requirements. The ensemble size of the deterministic particle filters is determined by the size of the pseudo-error covariance matrix. For example, the ensemble size of the unscented Kalman filter is $2n + 1$, where n is the dimension of the state to be estimated. However, computational resources place a

limit on the number of copies of the model that can be simulated in parallel.

One of the methods used to reduce the ensemble size is to apply the particle filtering algorithms to a truncated model. Specifically, these localized approaches construct ensembles of only the subset of the state whose estimates are desired [20]. The localized ensemble members are then used to construct a reduced-order pseudo-error covariance that is then used to construct the localized estimator gain. For example, in weather prediction applications, if estimates of certain atmospheric variables in only a specific region are required, then multiple copies of a model of only that region are created and used for data assimilation. Moreover, data injection is also restricted to state estimates corresponding to the local region. However, constraining data injection to a certain subset of the state in an ad-hoc manner may result in poor estimates of the state in other regions.

Yet another technique to reduce the ensemble size is given in [21, 22]. A common feature shared by these algorithms is that a low-rank approximation of the pseudo-error covariance is first constructed and then certain columns of this approximation are truncated. Since the ensemble members are re-initialized at every time step using the truncated low-rank approximation of the pseudo-error covariance, the truncation method influences the performance of these reduced ensemble estimation algorithms. Furthermore, these truncation algorithms involve an additional computational burden that is not present in the original full ensemble algorithms.

This dissertation addresses the problem of developing reduced-complexity algorithms for data-assimilation of large-scale linear and nonlinear systems. Throughout this discussion, we assume that we have a discrete-time model of the underlying dynamics. The remainder of this introduction summarizes the contents of each chapter, and outlines the original contributions of each chapter.

Chapter II Summary

The original contribution of Chapter II is an optimal linear estimator that constrains output injection to a specific subset of the state estimate. Two versions of the new linear estimator are presented and their performance is quantified. Results on the stability of the new estimator when used for state estimation of linear time-invariant systems are also presented.

Chapter III Summary

Reduced-order estimators for linear time-varying systems is considered in Chapter III. Specifically, we derive the optimal filter using a finite-horizon cost so that, unlike the infinite-horizon approach [7, 8], the resulting estimator does not require the solution of algebraic Riccati or Lyapunov equations.

Chapter IV Summary

In Chapter IV, we present a new reduced-rank square-root filter for linear systems that is based on the Cholesky factorization of the pseudo-error covariance. Specifically, Chapter IV provides a filter whose performance, in many cases, is better than the reduced-rank square-root filters in [21, 22] that use the singular value decomposition. Furthermore, the filter presented is also computationally more efficient compared to the reduced-rank square-root filters that use the singular value decomposition. Finally, we present cases when the new reduced-rank square-root filter that uses the Cholesky factorization is equivalent to the Kalman filter.

Chapter V Summary

The performance of two nonlinear estimation algorithms, the extended Kalman filter and the unscented Kalman filter, is compared in Chapter V for various nonlinear systems that contain nondifferentiable dynamics. Specifically, we are interested in data assimilation of one-dimensional compressible flow using a finite-volume model,

and the comparisons performed in Chapter V show the superiority of the unscented Kalman filter over the extended Kalman filter when the nonlinearities in a system become severe.

Chapter VI Summary

Within Chapter VI, we extend the results of Chapter IV for state estimation of nonlinear systems. Specifically, we incorporate the reduced-rank square-root filter presented in Chapter IV within the framework of the unscented Kalman filter presented in Chapter V, thus reducing the ensemble size and hence the computational requirements to propagate the error covariance. We compare the performance of this new filter with an analogous version that uses the singular value decomposition. The comparisons performed shows the superiority of this new filter in both estimation accuracy and computational requirements.

Chapter VII Summary

In Chapter VII, we present a technique that extends the localized data assimilation algorithms presented in [9]. The algorithms in [9] inject data into only a certain subset of the state and propagate a reduced-order error covariance. Hence, correlations between certain subsets of the state and the measured subspace are neglected. In Chapter VII, we compensate for the neglected correlation by using a static estimator gain based on steady-state correlations. Thus, using this new technique we are able to significantly improve estimates without a significant increase in the online computational requirements. We use this new estimation technique for data assimilation of two-dimensional magnetohydrodynamic flow using a finite-volume model that is implemented on parallel processors.

CHAPTER II

Kalman Filtering With Constrained Output Injection

This chapter considers an extension of the Kalman filter that uses measurement data to directly update the estimates of only a specific subset of the state. Specifically, we consider state estimation of discrete-time linear systems with time-varying state dimension. In the first part of this chapter, we derive the one-step and two-step versions of the new filter. The one-step version of the filter uses both the model information and measurement data in a single step, while the two-step version of the filter uses the model information and measurement data in two distinct steps. We derive bounds on the performance of both versions of the new filter, and also present a condition that guarantees their equivalence. The last part of this chapter deals with conditions that guarantee the asymptotic stability of the new filter for linear time-invariant systems. The results presented in this chapter are published in [23, 24].

2.1 Introduction

The classical Kalman filter provides optimal least-squares estimates of all of the states of a linear time-varying system under process and measurement noise. In

many applications, however, optimal estimates are desired for a specified subset of the system states, rather than all of the system states. For example, for systems arising from discretized partial differential equations, the chosen subset of states can represent a subregion of the spatial domain. However, it is well known that the optimal state estimator for a subset of system states coincides with the classical Kalman filter [14, pp. 104-109].

For applications involving high-order systems, it is often difficult to implement the classical Kalman filter, and thus it is of interest to consider computationally simpler filters that yield suboptimal estimates of a specified subset of states. One approach to this problem is to consider reduced-order Kalman filters. These reduced-complexity filters provide state estimates that are suboptimal relative to the classical Kalman filter [7, 8, 25, 26]. Alternative variants of the classical Kalman filter have been developed for computationally demanding applications such as weather forecasting [27–30], where the classical Kalman filter gain and covariance are modified so as to reduce the computational requirements.

The present chapter is motivated by computationally demanding applications such as those discussed in [27–30]. For such applications, a high-order simulation model is assumed to be available, but the derivation of a reduced-order filter in the sense of [7, 8, 25, 26] is not feasible due to the high dimensionality of the analytic model. Instead, we use a full-order state estimator based directly on the simulation model. However, rather than implementing the classical Kalman filter, we derive an optimal *spatially localized Kalman filter* in which the structure of the filter gain is constrained to reflect the desire to estimate a specified subset of states. Our development is also more general than the classical treatment since the state dimension can be time varying, which is useful for variable-resolution discretizations of partial

differential equations. Some of the results in this chapter appeared in [31].

The use of a spatially localized Kalman filter in place of the classical Kalman filter is also motivated by computational architecture constraints arising from a multiprocessor implementation of the Kalman filter [32] in which the Kalman filter operations can be confined to the subset of processors associated with the states whose estimates are desired.

2.2 Spatially Localized Kalman Filter

We consider the discrete-time dynamical system

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \quad k \geqslant 0, \tag{2.2.1}$$

with output

$$y_k = C_k x_k + v_k, \tag{2.2.2}$$

where $x_k \in \mathbb{R}^{n_k}$, $u_k \in \mathbb{R}^{m_k}$, $y_k \in \mathbb{R}^{l_k}$, and A_k, B_k, C_k are known real matrices of appropriate size. The input u_k and output y_k are assumed to be measured, and $w_k \in \mathbb{R}^{n_{k+1}}$ and $v_k \in \mathbb{R}^{l_k}$ are zero-mean white noise processes with variances and correlation

$$\mathcal{E}[w_k w_j^{\mathrm{T}}] = Q_k \delta_{kj}, \;\; \mathcal{E}[w_k v_j^{\mathrm{T}}] = S_k \delta_{kj}, \;\; \mathcal{E}[v_k v_j^{\mathrm{T}}] = R_k \delta_{kj}, \tag{2.2.3}$$

where δ_{kj} is the Kronecker delta, and $\mathcal{E}[\cdot]$ denotes expected value. We assume that R_k is positive definite. The initial state x_0 is assumed to be uncorrelated with w_k and v_k. Note that the dimension n_k of the state x_k can be time varying, and thus $A_k \in \mathbb{R}^{n_{k+1} \times n_k}$ is not necessarily square.

For the system (2.2.1) and (2.2.2), we consider a state estimator of the form

$$\hat{x}_{k+1} = A_k \hat{x}_k + B_k u_k + \mathit{\Gamma}_k K_k (y_k - \hat{y}_k), \quad k \geqslant 0, \tag{2.2.4}$$

with output

$$\hat{y}_k = C_k \hat{x}_k, \tag{2.2.5}$$

where $\hat{x}_k \in \mathbb{R}^{n_k}$, $\hat{y}_k \in \mathbb{R}^{l_k}$, $\Gamma_k \in \mathbb{R}^{n_{k+1} \times p_k}$, and $K_k \in \mathbb{R}^{p_k \times l_k}$. The nontraditional feature of (2.2.4) is the presence of the term Γ_k, which, in the classical case is the identity matrix. Here, Γ_k constrains the state estimator so that only estimator states in the range of Γ_k are directly affected by the gain K_k. For example, Γ_k can have the form

$$\Gamma_k = \begin{bmatrix} 0 \\ I_{p_k} \\ 0 \end{bmatrix}, \tag{2.2.6}$$

where I_r denotes the $r \times r$ identity matrix. We assume that Γ_k has full column rank for all $k \geq 0$.

Next, define the *state-estimation error state* e_k by

$$e_k \stackrel{\triangle}{=} x_k - \hat{x}_k, \tag{2.2.7}$$

which satisfies

$$e_{k+1} = \tilde{A}_k e_k + \tilde{w}_k, \quad k \geq 0, \tag{2.2.8}$$

where

$$\tilde{A}_k \stackrel{\triangle}{=} A_k - \Gamma_k K_k C_k, \quad \tilde{w}_k \stackrel{\triangle}{=} w_k - \Gamma_k K_k v_k. \tag{2.2.9}$$

Furthermore, we define the state-estimation error

$$J_k(K_k) \stackrel{\triangle}{=} \mathcal{E}[(L_k e_{k+1})^{\mathrm{T}} L_k e_{k+1}], \tag{2.2.10}$$

where $L_k \in \mathbb{R}^{q_k \times n_{k+1}}$ determines the weighted error components. Then,

$$J_k(K_k) = \mathrm{tr}\,[P_{k+1} M_k]\,, \tag{2.2.11}$$

where the error covariance $P_k \in \mathbb{R}^{n_k \times n_k}$ is defined by

$$P_k \stackrel{\triangle}{=} \mathcal{E}[e_k e_k^{\mathrm{T}}] \tag{2.2.12}$$

and $M_k \stackrel{\triangle}{=} L_k^{\mathrm{T}} L_k \in \mathbb{R}^{n_{k+1} \times n_{k+1}}$. We assume that M_k is positive definite for all $k \geqslant 0$. The following lemma will be useful.

Lemma 2.2.1 *The error* (2.2.7) *satisfies*

$$\mathcal{E}[e_k \tilde{w}_k^{\mathrm{T}}] = 0. \tag{2.2.13}$$

It thus follows from (2.2.8) and (2.2.13) that

$$\mathcal{E}[e_{k+1} e_{k+1}^{\mathrm{T}}] = \tilde{A}_k \mathcal{E}[e_k e_k^{\mathrm{T}}] \tilde{A}_k^{\mathrm{T}} + \mathcal{E}[\tilde{w}_k \tilde{w}_k^{\mathrm{T}}]. \tag{2.2.14}$$

Note that (2.2.3) and (2.2.9) imply that

$$\mathcal{E}[\tilde{w}_k \tilde{w}_k^{\mathrm{T}}] = \tilde{Q}_k, \tag{2.2.15}$$

where

$$\tilde{Q}_k \stackrel{\triangle}{=} Q_k - \mathit{\Gamma}_k K_k S_k^{\mathrm{T}} - S_k K_k^{\mathrm{T}} \mathit{\Gamma}_k^{\mathrm{T}} + \mathit{\Gamma}_k K_k R_k K_k^{\mathrm{T}} \mathit{\Gamma}_k^{\mathrm{T}}. \tag{2.2.16}$$

It thus follows from (2.2.12), (2.2.14), and (2.2.15) that P_k satisfies

$$P_{k+1} = \tilde{A}_k P_k \tilde{A}_k^{\mathrm{T}} + \tilde{Q}_k. \tag{2.2.17}$$

Therefore,

$$J_k(K_k) = \mathrm{tr}[(\tilde{A}_k P_k \tilde{A}_k^{\mathrm{T}} + \tilde{Q}_k) M_k]. \tag{2.2.18}$$

It follows from (2.2.9) and (2.2.16) that $J_k(K_k)$ can be expressed as

$$J_k(K_k) = \mathrm{tr}\left[\left((A_k - \mathit{\Gamma}_k K_k C_k) P_k (A_k - \mathit{\Gamma}_k K_k C_k)^{\mathrm{T}} + \tilde{Q}_k\right) M_k\right]. \tag{2.2.19}$$

2.3 Removing the Noise Correlation

In the classical case where $n_k = n$ and $\Gamma_k = I_n$ for all $k \geqslant 0$, the correlation S_k can be removed by introducing a linear combination of the measurements as deterministic inputs to the plant [34, pp. 181-183]. For the case $\Gamma_k \neq I_n$, we now state a condition under which we can derive an equivalent system with uncorrelated process and sensor noise.

Proposition 2.3.1 *Let $k \geqslant 0$ and suppose there exists $H_k \in \mathbb{R}^{p_k \times l_k}$ such that*

$$\Gamma_k H_k R_k = S_k. \tag{2.3.1}$$

Then

$$J_k(K_k) = \overline{J}_k(\overline{K}_k), \tag{2.3.2}$$

where

$$\overline{J}_k(\overline{K}_k) \triangleq \operatorname{tr}\left[\left((\overline{A}_k - \Gamma_k\overline{K}_kC_k)P_k(\overline{A}_k - \Gamma_k\overline{K}_kC_k)^{\mathrm{T}} + \overline{Q}_k + \Gamma_k\overline{K}_kR_k\overline{K}_k^{\mathrm{T}}\Gamma_k^{\mathrm{T}}\right)M_k\right], \tag{2.3.3}$$

$$\overline{K}_k \triangleq K_k - H_k, \quad \overline{A}_k \triangleq A_k - \Gamma_kH_kC_k, \tag{2.3.4}$$

and

$$\overline{Q}_k \triangleq Q_k - \Gamma_kH_kS_k^{\mathrm{T}} - S_kH_k^{\mathrm{T}}\Gamma_k^{\mathrm{T}} + \Gamma_kH_kR_kH_k^{\mathrm{T}}\Gamma_k^{\mathrm{T}}. \tag{2.3.5}$$

Proof. It follows from (2.3.5) that (2.2.18) can be expressed as

$$\begin{aligned} J_k(\overline{K}_k) = \operatorname{tr}\Big[\Big((\overline{A}_k - \Gamma_k\overline{K}_kC_k)P_k(\overline{A}_k - \Gamma_k\overline{K}_kC_k)^{\mathrm{T}} + \overline{Q}_k + \Gamma\overline{K}_kR_k\overline{K}_k^{\mathrm{T}}\Gamma_k^{\mathrm{T}} \\ -\Gamma_k\overline{K}_kS_k^{\mathrm{T}} - S_k\overline{K}_k^{\mathrm{T}}\Gamma_k^{\mathrm{T}} + \Gamma_k\overline{K}_kR_kH_k^{\mathrm{T}}\Gamma_k^{\mathrm{T}} + \Gamma_kH_kR_k\overline{K}_k^{\mathrm{T}}\Gamma_k^{\mathrm{T}}\Big)M_k\Big]. \end{aligned}$$

Using (2.3.1) yields (2.3.3). □

Note that replacing A_k, Q_k, and K_k in (2.2.18) by $\overline{A}_k$, $\overline{Q}_k$, and $\overline{K}_k$, respectively, and setting $S_k = 0$ in (2.2.18) yields (2.3.3). Hence, $\overline{J}_k(\overline{K}_k)$ is the cost of a system with uncorrelated process and sensor noise. It follows from (2.3.2) that $\overline{J}_k(\overline{K}_k)$ can be minimized with respect to $\overline{K}_k$, and K_k can be determined by using (2.3.4). If Γ_k is square and thus invertible by assumption, then $H_k = \Gamma_k^{-1} S_k R_k^{-1}$. In general, however, there may not exist a matrix H_k satisfying (2.3.1).

2.4 One-Step Spatially Constrained Kalman Filter

In this section we derive a one-step spatially constrained Kalman filter that minimizes the state-estimation error (2.2.18). For convenience, we define

$$\hat{S}_k \triangleq A_k P_k C_k^{\mathrm{T}} + S_k, \quad \hat{R}_k \triangleq R_k + C_k P_k C_k^{\mathrm{T}}, \tag{2.4.1}$$

and $\pi_k \in \mathbb{R}^{n_{k+1} \times n_{k+1}}$ by

$$\pi_k \stackrel{\triangle}{=} \Gamma_k (\Gamma_k^{\mathrm{T}} M_k \Gamma_k)^{-1} \Gamma_k^{\mathrm{T}} M_k. \tag{2.4.2}$$

Note that π_k is an oblique projector, that is, $\pi_k^2 = \pi_k$, but is not necessarily symmetric. Next, define the complementary oblique projector $\pi_{k\perp}$ by

$$\pi_{k\perp} \stackrel{\triangle}{=} I_{n_{k+1}} - \pi_k. \tag{2.4.3}$$

Proposition 2.4.1 *The gain K_k that minimizes the cost $J_k(K_k)$ in* (2.2.18) *is given by*

$$K_k = (\Gamma_k^{\mathrm{T}} M_k \Gamma_k)^{-1} \Gamma_k^{\mathrm{T}} M_k \hat{S}_k \hat{R}_k^{-1}, \tag{2.4.4}$$

where the error covariance P_k is updated using

$$P_{k+1} = A_k P_k A_k^{\mathrm{T}} + \pi_{k\perp} \hat{S}_k \hat{R}_k^{-1} \hat{S}_k^{\mathrm{T}} \pi_{k\perp}^{\mathrm{T}} + Q_k - \hat{S}_k \hat{R}_k^{-1} \hat{S}_k^{\mathrm{T}}. \tag{2.4.5}$$

Proof. Setting $J_k'(K_k) = 0$ and using the fact that $\Gamma_k^{\mathrm{T}} M_k \Gamma_k$ is positive definite for all $k \geqslant 0$ yields (2.4.4). It follows from [36, p. 286] that, for all $0 < \alpha < 1$, all distinct $A_1, A_2 \in \mathbb{R}^{n \times m}$, and positive-definite $B \in \mathbb{R}^{m \times m}$, $\operatorname{tr}\left[\alpha(1-\alpha)(A_1 - A_2)B(A_1 - A_2)^{\mathrm{T}}\right] > 0$. Hence, the mapping $A \to \operatorname{tr}(ABA^{\mathrm{T}})$ is strictly convex. It thus follows that $J_k(K_k)$ is strictly convex, and hence K_k in (2.4.4) is the unique global minimizer of $J_k(K_k)$. To update the error covariance, we first note that

$$\Gamma_k K_k = \pi_k \hat{S}_k \hat{R}_k^{-1}, \tag{2.4.6}$$

where π_k is defined by (2.4.2). Now, using (2.4.6) with (2.2.17) yields (2.4.5). □

If either $M_k = I_{n_{k+1}}$ or $L_k = \Gamma_k^{\mathrm{T}}$, then π_k is the orthogonal projector

$$\pi_k = \Gamma_k (\Gamma_k^{\mathrm{T}} \Gamma_k)^{-1} \Gamma_k^{\mathrm{T}}, \tag{2.4.7}$$

and it follows from (2.4.4) that

$$K_k = (\Gamma_k^{\mathrm{T}} \Gamma_k)^{-1} \Gamma_k^{\mathrm{T}} \hat{S}_k \hat{R}_k^{-1}. \tag{2.4.8}$$

Alternatively, specializing to the case in which Γ_k is square yields $\pi_k = I_n$ and $\pi_{k\perp} = 0$, as well as the standard Riccati update equation

$$P_{k+1} = A_k P_k A_k^{\mathrm{T}} + Q_k - (A_k P_k C_k^{\mathrm{T}} + S_k)(R_k + C_k P_k C_k^{\mathrm{T}})^{-1}(C_k P_k A_k^{\mathrm{T}} + S_k^{\mathrm{T}}). \tag{2.4.9}$$

In this case the Kalman filter gain is given by

$$K_k = (A_k P_k C_k^{\mathrm{T}} + S_k)(R_k + C_k P_k C_k^{\mathrm{T}})^{-1} \tag{2.4.10}$$

and the estimator equation is

$$\hat{x}_{k+1} = A_k \hat{x}_k + B_k u_k + K_k(y_k - \hat{y}_k). \tag{2.4.11}$$

Furthermore, the one-step filter provides optimal estimates of all of the states, that is, the filter does not depend on the state-estimate error weighting L_k.

Next, we show that increasing the number of estimator states that are directly injected with the output improves the filter performance. Define $\hat{\pi}_k$ and $\hat{\pi}_{k\perp}$ by

$$\hat{\pi}_k \triangleq \hat{\Gamma}_k(\hat{\Gamma}_k^{\mathrm{T}} M_k \hat{\Gamma}_k)^{-1}\hat{\Gamma}_k^{\mathrm{T}} M_k, \quad \hat{\pi}_{k\perp} \triangleq I - \hat{\pi}_k. \tag{2.4.12}$$

where $\hat{\Gamma}_k$ has full column rank. Next, let $\hat{K}_k$ be the optimal gain given by (2.4.4) with Γ_k replaced by $\hat{\Gamma}_k$, that is,

$$\hat{K} \triangleq (\hat{\Gamma}_k^{\mathrm{T}} M_k \hat{\Gamma}_k)^{-1}\hat{\Gamma}_k^{\mathrm{T}} M_k \hat{S}_k \hat{R}_k^{-1}, \tag{2.4.13}$$

and let $\hat{P}_{k+1}$ be the corresponding error covariance when $\hat{K}_k$ is used, that is,

$$\hat{P}_{k+1} = A_k P_k A_k^{\mathrm{T}} + \hat{\pi}_{k\perp}\hat{S}_k \hat{R}_k^{-1}\hat{S}_k^{\mathrm{T}}\hat{\pi}_{k\perp}^{\mathrm{T}} + Q_k - \hat{S}_k \hat{R}_k^{-1}\hat{S}_k^{\mathrm{T}}. \tag{2.4.14}$$

Proposition 2.4.2 *Assume that $M_k = I$, let $\hat{\Gamma}_k = [\Gamma_k \;\; G_k]$, and assume $\hat{\Gamma}_k$ has full column rank. Then*

$$\mathrm{tr}(\hat{P}_{k+1}) \leqslant \mathrm{tr}(P_{k+1}). \tag{2.4.15}$$

Proof. Noting that π_k and $\hat{\pi}_k$ are symmetric, it follows from (2.4.12) that

$$\hat{\pi}_k = \pi_k + \pi_{k\perp} G_k (G_k^{\mathrm{T}} \pi_{k\perp} G_k)^{-1} G_k^{\mathrm{T}} \pi_{k\perp}. \tag{2.4.16}$$

Therefore,

$$\pi_{k\perp} = \hat{\pi}_{k\perp} + \pi_{k\perp} G_k (G_k^{\mathrm{T}} \pi_{k\perp} G_k)^{-1} G_k^{\mathrm{T}} \pi_{k\perp}. \tag{2.4.17}$$

Hence, subtracting (2.4.14) from (2.4.5) yields

$$\mathrm{tr}(P_{k+1} - \hat{P}_{k+1}) = \mathrm{tr}((\pi_{k\perp} - \hat{\pi}_{k\perp})\hat{S}_k \hat{R}_k^{-1}\hat{S}_k^{\mathrm{T}}) \geqslant 0. \qquad \square$$

2.5 Two-Step Spatially Constrained Kalman Filter

In this section, we consider a two-step state estimator. The *data assimilation step* is given by

$$w_k^{\mathrm{da}} = \Upsilon_k K_{w,k}(y_k - y_k^{\mathrm{f}}), \quad k \geqslant 0, \tag{2.5.1}$$

and

$$x_k^{\mathrm{da}} = x_k^{\mathrm{f}} + \Gamma_k K_{x,k}(y_k - y_k^{\mathrm{f}}), \quad k \geqslant 0, \tag{2.5.2}$$

where $w_k^{\mathrm{da}} \in \mathbb{R}^{n_k}$ is the *data assimilation estimate* of w_k, $x_k^{\mathrm{da}} \in \mathbb{R}^{n_k}$ is the *data assimilation estimate* of x_k, and $x_k^{\mathrm{f}} \in \mathbb{R}^{n_k}$ is the *forecast estimate* of x_k. The *forecast step* or physics update is given by

$$x_{k+1}^{\mathrm{f}} = A_k x_k^{\mathrm{da}} + B_k u_k + w_k^{\mathrm{da}}, \quad k \geqslant 0, \tag{2.5.3}$$

$$y_k^{\mathrm{f}} = C_k x_k^{\mathrm{f}}. \tag{2.5.4}$$

Here, Υ_k is analogous to Γ_k in ensuring that only components of the process noise estimate in the range of Υ_k are directly affected by the gain $K_{w,k}$. We assume that Υ_k has full column rank for all $k \geq 0$. In traditional notation, x_k^{da} is denoted by $\hat{x}_{k|k}$ to indicate that $\hat{x}_{k|k}$ is the estimate of x_k obtained by using the measurements $y_0, \ldots, y_k$, while x_k^{f} is denoted by $\hat{x}_{k|k-1}$ to indicate that $\hat{x}_{k|k-1}$ is the estimate of x_k obtained by using the measurements $y_0, \ldots, y_{k-1}$. The notation x_k^{f} and x_k^{da} is motivated by the data assimilation literature [35].

Define the *forecast state error* e_k^{f} by

$$e_k^{\mathrm{f}} \triangleq x_k - x_k^{\mathrm{f}} \tag{2.5.5}$$

and the *forecast error covariance* P_k^{f} by

$$P_k^{\mathrm{f}} \triangleq \mathcal{E}[e_k^{\mathrm{f}}(e_k^{\mathrm{f}})^{\mathrm{T}}]. \tag{2.5.6}$$

It follows from (2.2.1) and (2.5.3) that

$$e_{k+1}^{\mathrm{f}} = A_k e_k^{\mathrm{da}} + w_k - w_k^{\mathrm{da}}, \quad k \geq 0, \tag{2.5.7}$$

where the *data assimilation error state* e_k^{da} is defined by

$$e_k^{\mathrm{da}} \triangleq x_k - x_k^{\mathrm{da}}. \tag{2.5.8}$$

Lemma 2.5.1 *The forecast error e_k^{f} satisfies*

$$\mathcal{E}[e_k^{\mathrm{f}} w_k^{\mathrm{T}}] = 0, \tag{2.5.9}$$

$$\mathcal{E}[e_k^{\mathrm{f}} v_k^{\mathrm{T}}] = 0. \tag{2.5.10}$$

Now, define the process noise estimation error

$$J_{w,k}(K_{w,k}) \triangleq \mathcal{E}\left[\left(H_k(w_k - w_k^{\mathrm{da}})\right)^{\mathrm{T}} H_k(w_k - w_k^{\mathrm{da}})\right], \tag{2.5.11}$$

where $H_k \in \mathbb{R}^{d_k \times n_{k+1}}$ determines the weighted error components. For convenience, define

$$N_k \triangleq H_k^{\mathrm{T}} H_k, \quad \chi_k \triangleq \Upsilon_k (\Upsilon_k^{\mathrm{T}} N_k \Upsilon_k)^{-1} \Upsilon_k^{\mathrm{T}} N_k, \quad \chi_{k\perp} \triangleq I_{n_{k+1}} - \chi_k. \tag{2.5.12}$$

Proposition 2.5.1 *The gain $K_{w,k}$ that minimizes the cost $J_{w,k}(K_{w,k})$ is given by*

$$K_{w,k} = (\Upsilon_k^{\mathrm{T}} N_k \Upsilon_k)^{-1} \Upsilon_k^{\mathrm{T}} N_k S_k (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1}. \tag{2.5.13}$$

Proof. Substituting (2.5.1) into (2.5.11), and using (2.2.3) and (2.5.9) in the resulting expression yields

$$J_{w,k}(K_{w,k}) = \mathrm{tr}\left[\left(Q_k - S_k K_{w,k}^{\mathrm{T}} \Upsilon_k^{\mathrm{T}} - \Upsilon_k K_{w,k} S_k^{\mathrm{T}} + \Upsilon_k K_{w,k} (C_k P_k^{\mathrm{f}} C_k + R_k) K_{w,k}^{\mathrm{T}} \Upsilon_k^{\mathrm{T}}\right) N_k\right]. \tag{2.5.14}$$

As in the proof of Proposition 2.4.1, $J_{w,k}(K_{w,k})$ is strictly convex. To obtain the optimal gain $K_{w,k}$, we set $J'_{w,k}(K_{w,k}) = 0$, which yields (2.5.13), the unique global minimizer of $J_{w,k}(K_{w,k})$. □

Next, define the state-estimation error

$$J_{x,k}(K_{x,k}) \triangleq \mathcal{E}[(L_k e_k^{\mathrm{da}})^{\mathrm{T}} L_k e_k^{\mathrm{da}}] \tag{2.5.15}$$

so that

$$J_{x,k}(K_{x,k}) = \mathrm{tr}\left[P_k^{\mathrm{da}} M_k\right], \tag{2.5.16}$$

where the *data assimilation error covariance* $P_k^{\mathrm{da}} \in \mathbb{R}^{n_k \times n_k}$ is defined by

$$P_k^{\mathrm{da}} \triangleq \mathcal{E}[e_k^{\mathrm{da}} (e_k^{\mathrm{da}})^{\mathrm{T}}]. \tag{2.5.17}$$

It follows from (2.5.2), (2.5.5), and (2.5.8) that

$$e_k^{\mathrm{da}} = \tilde{K}_{x,k} e_k^{\mathrm{f}} - \mathit{\Gamma}_k K_{x,k} v_k, \tag{2.5.18}$$

where

$$\tilde{K}_{x,k} \triangleq I - \mathit{\Gamma}_k K_{x,k} C_k. \tag{2.5.19}$$

Substituting (2.5.1) and (2.5.18) into (2.5.7) yields

$$e_{k+1}^{\mathrm{f}} = (A_k \tilde{K}_{x,k} - \mathit{\Upsilon}_k K_{w,k} C_k) e_k^{\mathrm{f}} + w_k - (A_k \mathit{\Gamma}_k K_{x,k} + \mathit{\Upsilon}_k K_{w,k}) v_k. \tag{2.5.20}$$

Next, define

$$R_k^{\mathrm{f}} \triangleq R_k + C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} \tag{2.5.21}$$

and

$$\begin{aligned} Q_k^{\mathrm{f}} \triangleq{}& Q_k - (A_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + S_k)(R_k^{\mathrm{f}})^{-1} (A_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + S_k)^{\mathrm{T}} \\ &+ \left(A_k \pi_{k\perp} P_k^{\mathrm{f}} C_k^{\mathrm{T}} + \chi_{k\perp} S_k\right) (R_k^{\mathrm{f}})^{-1} \left(A_k \pi_{k\perp} P_k^{\mathrm{f}} C_k^{\mathrm{T}} + \chi_{k\perp} S_k\right)^{\mathrm{T}} \\ &+ A_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} (R_k^{\mathrm{f}})^{-1} C_k P_k^{\mathrm{f}} A_k^{\mathrm{T}} - A_k \pi_{k\perp} P_k^{\mathrm{f}} C_k^{\mathrm{T}} (R_k^{\mathrm{f}})^{-1} C_k P_k^{\mathrm{f}} \pi_{k\perp}^{\mathrm{T}} A_k^{\mathrm{T}}. \end{aligned} \tag{2.5.22}$$

Proposition 2.5.2 *The gain $K_{x,k}$ that minimizes the cost $J_{x,k}(K_{x,k})$ is given by*

$$K_{x,k} = (\Gamma_k^{\mathrm{T}} M_k \Gamma_k)^{-1} \Gamma_k^{\mathrm{T}} M_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} (R_k^{\mathrm{f}})^{-1}, \tag{2.5.23}$$

where P_k^{da} and P_k^{f} are given by

$$P_k^{\mathrm{da}} = P_k^{\mathrm{f}} - P_k^{\mathrm{f}} C_k^{\mathrm{T}} (R_k^{\mathrm{f}})^{-1} C_k P_k^{\mathrm{f}} + \pi_{k\perp} P_k^{\mathrm{f}} C_k^{\mathrm{T}} (R_k^{\mathrm{f}})^{-1} C_k P_k^{\mathrm{f}} \pi_{k\perp}^{\mathrm{T}} \tag{2.5.24}$$

and

$$P_{k+1}^{\mathrm{f}} = A_k P_k^{\mathrm{da}} A_k^{\mathrm{T}} + Q_k^{\mathrm{f}}. \tag{2.5.25}$$

Proof. Using (2.5.17) and (2.5.18), P_k^{da} satisfies

$$P_k^{\mathrm{da}} = \tilde{K}_{x,k} P_k^{\mathrm{f}} \tilde{K}_{x,k}^{\mathrm{T}} - \tilde{K}_{x,k} \mathcal{E}[e_k^{\mathrm{f}} v_k^{\mathrm{T}}] K_{x,k}^{\mathrm{T}} \Gamma_k^{\mathrm{T}} - \Gamma_k K_{x,k} \mathcal{E}[v_k (e_k^{\mathrm{f}})^{\mathrm{T}}] \tilde{K}_{x,k}^{\mathrm{T}} + \Gamma_k K_{x,k} R_k K_{x,k}^{\mathrm{T}} \Gamma_k^{\mathrm{T}}. \tag{2.5.26}$$

Substituting (2.5.10) into (2.5.26) and substituting the resulting equation into (2.5.16) yields

$$J_{x,k}(K_{x,k}) = \mathrm{tr}[(\tilde{K}_{x,k} P_k^{\mathrm{f}} \tilde{K}_{x,k}^{\mathrm{T}} + \Gamma_k K_{x,k} R_k K_{x,k}^{\mathrm{T}} \Gamma_k^{\mathrm{T}}) M_k]. \tag{2.5.27}$$

To obtain the optimal gain $K_{x,k}$, we set $J'_{x,k}(K_{x,k}) = 0$, which yields (2.5.23). As in the proof of Proposition 2.4.1, it can be shown that $J_{x,k}(K_{x,k})$ is strictly convex, and hence $K_{x,k}$ in (2.5.23) is the unique global minimizer of $J_{x,k}(K_{x,k})$. Substituting (2.5.9) and (2.5.23) into (2.5.26) yields (2.5.24).

To update the forecast error covariance, we substitute (2.5.1) into (2.5.7) so that

$$e_{k+1}^{\mathrm{f}} = A_k e_k^{\mathrm{da}} - \Upsilon_k K_{w,k} C_k e_k^{\mathrm{f}} + w_k - \Upsilon_k K_{w,k} v_k.$$

Hence,

$$\begin{aligned}P_{k+1}^{\mathrm{f}} = {} & A_k P_k^{\mathrm{da}} A_k^{\mathrm{T}} + Q_k + \Upsilon_k K_{w,k}(C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k) K_{w,k}^{\mathrm{T}} \Upsilon_k^{\mathrm{T}} \\ & + A_k \mathcal{E}[e_k^{\mathrm{da}} w_k^{\mathrm{T}}] + \mathcal{E}[w_k (e_k^{\mathrm{da}})^{\mathrm{T}}] A_k^{\mathrm{T}} - A_k \mathcal{E}[e_k^{\mathrm{da}} (e_k^{\mathrm{f}})^{\mathrm{T}}] C_k^{\mathrm{T}} K_{w,k}^{\mathrm{T}} \Upsilon_k^{\mathrm{T}} \\ & - \Upsilon_k K_{w,k} C_k \mathcal{E}[e_k^{\mathrm{f}} (e_k^{\mathrm{da}})^{\mathrm{T}}] A_k^{\mathrm{T}} - A_k \mathcal{E}[e_k^{\mathrm{da}} v_k^{\mathrm{T}}] K_{w,k}^{\mathrm{T}} \Upsilon_k^{\mathrm{T}} \\ & - \Upsilon_k K_{w,k} \mathcal{E}[v_k (e_k^{\mathrm{da}})^{\mathrm{T}}] A_k^{\mathrm{T}} - \mathcal{E}[w_k (e_k^{\mathrm{f}})^{\mathrm{T}}] C_k^{\mathrm{T}} K_{w,k}^{\mathrm{T}} \Upsilon_k^{\mathrm{T}} \\ & - \Upsilon_k K_{w,k} C_k \mathcal{E}[e_k^{\mathrm{f}} w_k^{\mathrm{T}}] - \mathcal{E}[w_k v_k^{\mathrm{T}}] K_{w,k}^{\mathrm{T}} \Upsilon_k^{\mathrm{T}} - \Upsilon_k K_{w,k} \mathcal{E}[v_k w_k^{\mathrm{T}}] \\ & + \Upsilon_k K_{w,k} (C_k \mathcal{E}[e_k^{\mathrm{f}} v_k^{\mathrm{T}}] + \mathcal{E}[v_k (e_k^{\mathrm{f}})^{\mathrm{T}}] C_k^{\mathrm{T}}) K_{w,k}^{\mathrm{T}} \Upsilon_k^{\mathrm{T}}.\end{aligned} \tag{2.5.28}$$

Substituting (2.5.18) into (2.5.28), and using (2.5.9) and (2.5.10) in the resulting expression yields (2.5.25). □

The two-step estimator can be summarized as follows:

Data assimilation step:

$$w_k^{\mathrm{da}} = \Upsilon_k K_{w,k}(y_k - y_k^{\mathrm{f}}), \tag{2.5.29}$$

$$K_{w,k} = \left(\Upsilon_k^{\mathrm{T}} N_k \Upsilon_k\right)^{-1} \Upsilon_k^{\mathrm{T}} N_k S_k (R_k^{\mathrm{f}})^{-1}, \tag{2.5.30}$$

$$x_k^{\mathrm{da}} = x_k^{\mathrm{f}} + \Gamma_k K_{x,k}(y_k - y_k^{\mathrm{f}}), \tag{2.5.31}$$

$$K_{x,k} = (\Gamma_k^{\mathrm{T}} M_k \Gamma_k)^{-1} \Gamma_k^{\mathrm{T}} M_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} (R_k^{\mathrm{f}})^{-1}, \tag{2.5.32}$$

$$P_k^{\mathrm{da}} = P_k^{\mathrm{f}} - P_k^{\mathrm{f}} C_k^{\mathrm{T}} (R_k^{\mathrm{f}})^{-1} C_k P_k^{\mathrm{f}} + \pi_{k\perp} P_k^{\mathrm{f}} C_k^{\mathrm{T}} (R_k^{\mathrm{f}})^{-1} C_k P_k^{\mathrm{f}} \pi_{k\perp}^{\mathrm{T}}. \tag{2.5.33}$$

Forecast step:

$$x_{k+1}^{\mathrm{f}} = A_k x_k^{\mathrm{da}} + B_k u_k + w_k^{\mathrm{da}}, \tag{2.5.34}$$

$$P_{k+1}^{\mathrm{f}} = A_k P_k^{\mathrm{da}} A_k^{\mathrm{T}} + Q_k^{\mathrm{f}}. \tag{2.5.35}$$

Assume that Γ_k and Υ_k are square for all $k \geqslant 0$. Substituting (2.5.29) and (2.5.31)

into (2.5.34) yields the familiar one-step Kalman filter

$$x_{k+1}^{\mathrm{f}} = A_k x_k^{\mathrm{f}} + B_k u_k + (A_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + S_k)(R_k + C_k P_k^{\mathrm{f}} C_k)^{-1}(y_k - y_k^{\mathrm{f}}). \tag{2.5.36}$$

Furthermore, substituting (2.5.33) into (2.5.35) yields

$$P_{k+1}^{\mathrm{f}} = A_k P_k^{\mathrm{f}} A_k^{\mathrm{T}} - (A_k P_k^{\mathrm{f}} C_k + S_k)(R_k + C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}})^{-1}(C_k P_k^{\mathrm{f}} A_k^{\mathrm{T}} + S_k^{\mathrm{T}}) + Q_k. \tag{2.5.37}$$

Next, as in Proposition 2.4.2, we show that when additional estimator states are directly injected with the output data, the performance of the two-step filter improves. Define $\hat{K}_{x,k}$ by (2.5.23) with Γ_k replaced by $\hat{\Gamma}_k$, that is,

$$\hat{K}_{x,k} = (\hat{\Gamma}_k^{\mathrm{T}} M_k \hat{\Gamma}_k)^{-1} \hat{\Gamma}_k^{\mathrm{T}} M_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} (R_k^{\mathrm{f}})^{-1}. \tag{2.5.38}$$

Furthermore, let $\hat{P}_k^{\mathrm{da}}$ be the corresponding data assimilation error covariance when $\hat{K}_{x,k}$ is used instead of $K_{x,k}$, that is,

$$\hat{P}_k^{\mathrm{da}} \triangleq P_k^{\mathrm{f}} - P_k^{\mathrm{f}} C_k^{\mathrm{T}} (R_k^{\mathrm{f}})^{-1} C_k P_k^{\mathrm{f}} + \hat{\pi}_{k\perp} P_k^{\mathrm{f}} C_k^{\mathrm{T}} (R_k^{\mathrm{f}})^{-1} C_k P_k^{\mathrm{f}} \hat{\pi}_{k\perp}^{\mathrm{T}}. \tag{2.5.39}$$

Proposition 2.5.3 *Let $M_k = I$, $\hat{\Gamma}_k = [\Gamma_k \;\; G_k]$, and assume that $\hat{\Gamma}_k$ has full column rank. Then*

$$\mathrm{tr}(\hat{P}_k^{\mathrm{da}}) \leqslant \mathrm{tr}(P_k^{\mathrm{da}}). \tag{2.5.40}$$

Proof. Subtracting (2.5.39) from (2.5.24) and using the fact from (2.4.17) that $\pi_{k\perp} - \hat{\pi}_{k\perp}$ is positive semi-definite, it follows that

$$\mathrm{tr}(P_k^{\mathrm{da}} - \hat{P}_k^{\mathrm{da}}) = \mathrm{tr}\left[(\pi_{k\perp} - \hat{\pi}_{k\perp}) P_k^{\mathrm{f}} C_k^{\mathrm{T}} (R_k^{\mathrm{f}})^{-1} C_k P_k^{\mathrm{f}}\right] \geqslant 0. \qquad \square$$

2.6 Comparison of the One-Step and Two-Step Filters

When Γ_k and Υ_k are square, comparing (2.4.9) with (2.5.37) and (2.4.11) with (2.5.34) shows that the two-step filter is equivalent to the one-step filter with $K_k = AK_{x,k}+K_{w,k}$, $\hat{x}_k = x_k^{\mathrm{f}}$ and $P_k = P_k^{\mathrm{f}}$. When Γ_k and Υ_k are not square, we obtain a sufficient condition under which the one-step and two-step spatially constrained Kalman filters are equivalent.

Proposition 2.6.1 *Suppose that* $\hat{x}_0 = x_0^{\mathrm{f}}$ *and* $P_0 = P_0^{\mathrm{f}}$*, and, for all* $k \geqslant 0$*,*

$$A_k \pi_{k\perp} P_k^{\mathrm{f}} C_k^{\mathrm{T}} + \chi_{k\perp} S_k = \pi_{k\perp} \left(A_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + S_k \right). \tag{2.6.1}$$

Then the one-step filter (2.4.4), (2.4.5) *and the two-step filter in* (2.5.29)-(2.5.35) *are equivalent, that is, for all* $k > 0$*,* $\hat{x}_k = x_k^{\mathrm{f}}$ *and* $P_k = P_k^{\mathrm{f}}$*.*

Proof. Substituting (2.5.22) and (2.5.33) into (2.5.35) yields

$$\begin{aligned} P_{k+1}^{\mathrm{f}} = A_k P_k^{\mathrm{f}} A_k^{\mathrm{T}} + \left(A_k \pi_{k\perp} P_k^{\mathrm{f}} C_k^{\mathrm{T}} + \chi_{k\perp} S_k \right) (R_k^{\mathrm{f}})^{-1} \left(A_k \pi_{k\perp} P_k^{\mathrm{f}} C_k^{\mathrm{T}} + \chi_{k\perp} S_k \right)^{\mathrm{T}} \\ - (A_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + S_k)(R_k^{\mathrm{f}})^{-1} (A_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + S_k)^{\mathrm{T}} + Q_k. \end{aligned} \tag{2.6.2}$$

Substituting (2.6.7) into (2.6.2) yields

$$\begin{aligned} P_{k+1}^{\mathrm{f}} = A_k P_k^{\mathrm{f}} A_k^{\mathrm{T}} + \pi_{k\perp} (A_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + S_k)(R_k^{\mathrm{f}})^{-1} (A_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + S_k)^{\mathrm{T}} \pi_{k\perp}^{\mathrm{T}} \\ + Q_k - (A_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + S_k)(R_k^{\mathrm{f}})^{-1} (A_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + S_k)^{\mathrm{T}}. \end{aligned} \tag{2.6.3}$$

Since $P_0^{\mathrm{f}} = P_0$, it follows from (2.4.1), (2.4.5), and (2.5.21) that, for all $k > 0$, $P_k^{\mathrm{f}} = P_k$.

Next, substituting (2.5.1) and (2.5.31) into (2.5.34) yields

$$x_{k+1}^{\mathrm{f}} = A_k x_k^{\mathrm{f}} + B_k u_k + (A_k \pi_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + \chi_k S_k)(R_k^{\mathrm{f}})^{-1} (y_k - y_k^{\mathrm{f}}). \tag{2.6.4}$$

Now, (2.5.21) and (2.6.7) imply that

$$x_{k+1}^{\mathrm{f}} = A_k x_k^{\mathrm{f}} + B_k u_k + \pi_k (A_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + S_k)(C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} (y_k - C_k x_k^{\mathrm{f}}). \tag{2.6.5}$$

It follows from (2.2.4) and (2.4.4) that, for all $k \geqslant 0$,

$$\hat{x}_{k+1} = A_k\hat{x}_k + B_k u_k + \pi_k(A_k P_k C_k^{\mathrm{T}} + S_k)(C_k P_k C_k^{\mathrm{T}} + R_k)^{-1}(y_k - C_k\hat{x}_k). \quad (2.6.6)$$

Since $\hat{x}_0 = x_0^{\mathrm{f}}$ and $P_k^{\mathrm{f}} = P_k$ for all $k \geqslant 0$, (2.6.5) and (2.6.6) imply that $\hat{x}_k = x_k^{\mathrm{f}}$ for all $k \geqslant 0$. □

Note that, if Γ_k and Υ_k are square, then $\pi_{k\perp} = 0$ and $\chi_{k\perp} = 0$, and thus (2.6.7) is satisfied. Furthermore, if $S_k = 0$ or $\pi_k = \chi_k$, then Proposition 2.6.1 specializes to the following result.

Corollary 2.6.1 *Suppose that* $\hat{x}_0 = x_0^{\mathrm{f}}$, $P_0 = P_0^{\mathrm{f}}$, *and, for all* $k \geqslant 0$, *either* $S_k = 0$ *or* $\pi_k = \chi_k$. *If*

$$A_k\pi_{k\perp} = \pi_{k\perp}A_k, \quad (2.6.7)$$

for all $k \geqslant 0$, *then the one-step filter* (2.4.4), (2.4.5) *and the two-step filter in* (2.5.29)-(2.5.35) *are equivalent, that is, for all* $k > 0$, $\hat{x}_k = x_k^{\mathrm{f}}$ *and* $P_k = P_k^{\mathrm{f}}$.

Next, we present a converse of Proposition 6.1.

Proposition 2.6.2 *Assume that the one-step filter* (2.4.4), (2.4.5) *and the two-step filter in* (2.5.29)-(2.5.35) *are equivalent, that is, for all* $k \geqslant 0$, $\hat{x}_k = x_k^{\mathrm{f}}$ *and* $P_k = P_k^{\mathrm{f}}$. *Then, for all* $k \geqslant 0$, *there exists an orthogonal matrix* $U_k \in \mathbb{R}^{l_k \times l_k}$ *such that*

$$(A_k\pi_{k\perp}P_k^{\mathrm{f}}C_k^{\mathrm{T}} + \chi_{k\perp}S_k)(R_k^{\mathrm{f}})^{-1/2}U_k = \pi_{k\perp}(A_k P_k C_k^{\mathrm{T}} + S_k)(R_k^{\mathrm{f}})^{-1/2}. \quad (2.6.8)$$

Proof. Since $P_k = P_k^{\mathrm{f}}$ for all $k \geqslant 0$, subtracting (2.4.5) from (2.6.3) yields

$$\pi_{k\perp}\hat{S}_k\hat{R}_k^{-1}\hat{S}_k^{\mathrm{T}}\pi_{k\perp}^{\mathrm{T}} = (A_k\pi_{k\perp}P_k^{\mathrm{f}}C_k^{\mathrm{T}} + \chi_{k\perp}S_k)(R_k^{\mathrm{f}})^{-1}(A_k\pi_{k\perp}P_k^{\mathrm{f}}C_k^{\mathrm{T}} + \chi_{k\perp}S_k)^{\mathrm{T}}. \quad (2.6.9)$$

Hence, (2.6.8) follows from (2.4.1) and [36, p. 193]. □

Neither the one-step nor the two-step filter performs consistently better than the other. However, there are special cases when the performance of one filter is better than the other.

Proposition 2.6.3 *Assume that $C_k = 0$ and $P_k = P_k^{\mathrm{f}}$. If Γ_k is square and Υ_k is not square, then*

$$P_{k+1} \leqslant P_{k+1}^{\mathrm{f}}. \tag{2.6.10}$$

Alternatively, if Γ_k is not square and Υ_k is square, then

$$P_{k+1}^{\mathrm{f}} \leqslant P_{k+1}. \tag{2.6.11}$$

Proof. Assume that Γ_k is square and Υ_k is not square. It then follows from (2.4.2), (2.4.3) and (2.5.12) that

$$\pi_{k\perp} = 0, \quad \chi_{k\perp} \neq 0.$$

Substituting (2.5.33) and (2.5.22) into (2.5.35), and using $C_k = 0$ and $\pi_{k\perp} = 0$ yields

$$P_{k+1}^{\mathrm{f}} = A_k P_k^{\mathrm{f}} A_k^{\mathrm{T}} + \chi_{k\perp} S_k (R_k^{\mathrm{f}})^{-1} S_k^{\mathrm{T}} \chi_{k\perp}^{\mathrm{T}} - S_k (R_k^{\mathrm{f}})^{-1} S_k^{\mathrm{T}} + Q_k. \tag{2.6.12}$$

Substituting $C_k = 0$ and $\pi_{k\perp} = 0$ into (2.4.5) yields

$$P_{k+1} = A_k P_k A_k^{\mathrm{T}} - S_k (C_k P_k C_k^{\mathrm{T}} + R_k)^{-1} S_k^{\mathrm{T}} + Q_k. \tag{2.6.13}$$

Subtracting (2.6.13) from (2.6.12) yields (2.6.10).

Alternatively, if Υ_k is square and Γ_k is not square, then

$$\pi_{k\perp} \neq 0, \quad \chi_{k\perp} = 0.$$

Substituting (2.5.33) and (2.5.22) into (2.5.35), and using $C_k = 0$ and $\chi_{k\perp} = 0$ yields

$$P_{k+1}^{\mathrm{f}} = A_k P_k^{\mathrm{f}} A_k^{\mathrm{T}} - S_k (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} S_k^{\mathrm{T}} + Q_k. \tag{2.6.14}$$

Substituting $C_k = 0$ into (2.4.5) yields

$$P_{k+1} = A_k P_k A_k^{\mathrm{T}} + \pi_{k\perp} S_k \hat{R}_k^{-1} S_k^{\mathrm{T}} \pi_{k\perp}^{\mathrm{T}} - S_k \hat{R}_k^{-1} S_k^{\mathrm{T}} + Q_k. \tag{2.6.15}$$

Subtracting (2.6.14) from (2.6.15) yields (2.6.11). □

2.7 Comparison of the Open-Loop and Closed-Loop Covariances

Next, we consider the zero-gain filter

$$\hat{x}_{\mathrm{ol},k+1} = A_k \hat{x}_{\mathrm{ol},k} + B_k u_k \tag{2.7.1}$$

with the *zero-gain state-estimation error state*

$$e_{\mathrm{ol},k} \triangleq x_k - \hat{x}_{\mathrm{ol},k}. \tag{2.7.2}$$

It follows from (2.2.1), (2.7.1) and (2.7.2) that

$$P_{\mathrm{ol},k+1} = A_k P_{\mathrm{ol},k} A_k^{\mathrm{T}} + Q_k, \tag{2.7.3}$$

where the zero-gain error covariance $P_{\mathrm{ol},k} \in \mathbb{R}^{n_k \times n_k}$ is defined by $P_{\mathrm{ol},k} \triangleq \mathcal{E}\left[e_{\mathrm{ol},k} e_{\mathrm{ol},k}^{\mathrm{T}}\right]$. First, we show that the performance of the Kalman filter is better than the performance of the zero-gain filter.

Proposition 2.7.1 *If $\pi_k = I_{n_{k+1}}$ and $P_k \leqslant P_{\mathrm{ol},k}$, then $P_{k+1} \leqslant P_{\mathrm{ol},k+1}$.*

Proof. Since $\pi_k = I_{n_{k+1}}$, it follows from (2.4.3) that $\pi_{k\perp} = 0$, and hence (2.4.5) implies that

$$P_{k+1} = A_k P_k A_k^{\mathrm{T}} + Q_k - \hat{S}_k \hat{R}_k^{-1} \hat{S}_k^{\mathrm{T}}. \tag{2.7.4}$$

Subtracting (2.7.4) from (2.7.3) yields

$$P_{\mathrm{ol},k+1} - P_{k+1} = A_k (P_{\mathrm{ol},k} - P_k) A_k^{\mathrm{T}} + \hat{S}_k \hat{R}_k^{-1} \hat{S}_k \geqslant 0. \qquad \square$$

If $\pi_k \neq I_{n_{k+1}}$, then $\pi_{k\perp} \neq 0$, and subtracting (2.4.5) from (2.7.3) yields

$$P_{\mathrm{ol},k+1} - P_{k+1} = A_k(P_{\mathrm{ol},k} - P_k)A_k^{\mathrm{T}} + \hat{S}_k\hat{R}_k^{-1}\hat{S}_k^{\mathrm{T}} - \pi_{k\perp}\hat{S}_k\hat{R}_k^{-1}\hat{S}_k^{\mathrm{T}}\pi_{k\perp}^{\mathrm{T}}, \qquad (2.7.5)$$

which suggests the following negative result.

Proposition 2.7.2 *If $\pi_k \neq I_{n_{k+1}}$ and $P_k = P_{\mathrm{ol},k}$, then $P_{k+1} \leqslant P_{\mathrm{ol},k+1}$ is not always true.*

Proof. Let $k \geqslant 0$, $n_k = n_{k+1} = 2$, and

$$A_k = \begin{bmatrix} 0 & \alpha \\ 0 & 0.5 \end{bmatrix}, \; C_k = \begin{bmatrix} 0 & 1 \end{bmatrix},$$

where $24\alpha^2 + 2\alpha < 1$, and

$$Q_k = 0, \; S_k = 0, \; R_k = I, \; L_k = I, \; \Gamma_k = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Furthermore, let P_k and $P_{\mathrm{ol},k}$ have the scalar entries

$$P_k = \begin{bmatrix} p_{1,k} & p_{12,k} \\ p_{12,k} & p_{2,k} \end{bmatrix}, \quad P_{\mathrm{ol},k} = \begin{bmatrix} p_{\mathrm{ol},1,k} & p_{\mathrm{ol},12,k} \\ p_{\mathrm{ol},12,k} & p_{\mathrm{ol},2,k} \end{bmatrix}.$$

It follows from (2.4.5) and (2.7.3) that, if $P_k = P_{\mathrm{ol},k}$, then

$$p_{\mathrm{ol},1,k+1} - p_{1,k+1} = \left(\frac{24\alpha^2 + 2\alpha - 1}{25}\right)\frac{p_{2,k}^2}{1 + p_{2,k}}.$$

Hence, $p_{\mathrm{ol},1,k+1} < p_{1,k+1}$, and thus $P_{\mathrm{ol},k+1} - P_{k+1}$ is not positive semidefinite. □

The following result guarantees that the performance of the constrained filter is better than the performance of the zero-gain filter.

Proposition 2.7.3 *If $P_k \leqslant P_{\mathrm{ol},k}$, then*

$$\mathrm{tr}(P_{k+1}M_k) \leqslant \mathrm{tr}(P_{\mathrm{ol},k+1}M_k). \tag{2.7.6}$$

Proof. It follows from (2.4.3) and (2.7.5) that

$$\begin{aligned}\mathrm{tr}((P_{\mathrm{ol},k+1} - P_{k+1})M_k) =& \mathrm{tr}(A_k(P_{\mathrm{ol},k} - P_k)A_k^{\mathrm{T}}M_k) + \mathrm{tr}(\pi_k\hat{S}_k\hat{R}_k^{-1}\hat{S}_k^{\mathrm{T}}M_k \\ &+ M_k\hat{S}_k\hat{R}_k^{-1}\hat{S}_k^{\mathrm{T}}\pi_k^{\mathrm{T}} - \pi_k\hat{S}_k\hat{R}_k^{-1}\hat{S}_k^{\mathrm{T}}\pi_k^{\mathrm{T}}M_k).\end{aligned} \tag{2.7.7}$$

Since $\pi_k^{\mathrm{T}}M_k\pi_k = M_k\pi_k = \pi_k^{\mathrm{T}}M_k$, it follows that

$$\begin{aligned}\mathrm{tr}((P_{\mathrm{ol},k+1} - P_{k+1})M_k) &= \mathrm{tr}(A_k(P_{\mathrm{ol},k} - P_k)A_k^{\mathrm{T}}M_k) + \mathrm{tr}(\pi_k\hat{S}_k\hat{R}_k^{-1}\hat{S}_k^{\mathrm{T}}\pi_k^{\mathrm{T}}M_k) \\ &= \mathrm{tr}(L_kA_k(P_{\mathrm{ol},k} - P_k)A_k^{\mathrm{T}}L_k^{\mathrm{T}}) + \mathrm{tr}(L_k\pi_k\hat{S}_k\hat{R}_k^{-1}\hat{S}_k^{\mathrm{T}}\pi_k^{\mathrm{T}}L_k^{\mathrm{T}}) \geqslant 0.\end{aligned}$$

□

In fact, in the example in Proposition 2.7.2, $M_k = I$ and

$$\mathrm{tr}(P_{\mathrm{ol},k+1}) - \mathrm{tr}(P_{k+1}) = \left[\frac{22}{25}\left(\alpha + \frac{3}{22}\right)^2 + \frac{5}{44}\right]\frac{p_{2,k}^2}{1 + p_{2,k}} \geqslant 0. \tag{2.7.8}$$

Hence, $\mathrm{tr}(P_{k+1}) \leqslant \mathrm{tr}(P_{\mathrm{ol},k+1})$, and the one-step filter with constrained output injection performs better than the zero-gain filter. Although Proposition 2.7.3 guarantees that the performance of the one-step filter with constrained output injection is better than the zero-gain filter at time $k+1$, it follows from Proposition 2.7.2 that $P_{k+1} \leqslant P_{\mathrm{ol},k+1}$ may not be true. Hence, Proposition 2.7.3 does not guarantee that the performance of the one-step filter with constrained output injection is better than the zero-gain filter at time $k+2$, that is, $\mathrm{tr}(P_{k+2}) \leqslant \mathrm{tr}(P_{\mathrm{ol},k+2})$ may not be true.

The following result gives a condition under which the state estimates in the range of Γ_k are better than the corresponding estimates from the zero-gain filter.

Proposition 2.7.4 *If $P_k \leqslant P_{\mathrm{ol},k}$, then*

$$\Gamma_k^{\mathrm{T}}M_kP_{k+1}M_k\Gamma_k \leqslant \Gamma_k^{\mathrm{T}}M_kP_{\mathrm{ol},k+1}M_k\Gamma_k. \tag{2.7.9}$$

Proof. Note that

$$\begin{aligned}\Gamma_k^{\mathrm{T}} M_k (P_{k+1} - P_{\mathrm{ol},k+1}) M_k \Gamma_k =& \Gamma_k^{\mathrm{T}} M_k A_k (P_k - P_{\mathrm{ol},k}) A_k^{\mathrm{T}} M_k \Gamma_k - \Gamma_k^{\mathrm{T}} M_k \pi_k \hat{S}_k \hat{R}_k^{-1} \hat{S}_k^{\mathrm{T}} M_k \Gamma_k \\ & - \Gamma_k^{\mathrm{T}} M_k \hat{S}_k \hat{R}_k^{-1} \hat{S}_k^{\mathrm{T}} \pi_k^{\mathrm{T}} M_k \Gamma_k + \Gamma_k^{\mathrm{T}} M_k \pi_k \hat{S}_k \hat{R}_k^{-1} \hat{S}_k^{\mathrm{T}} \pi_k^{\mathrm{T}} M_k \Gamma_k. \end{aligned} \tag{2.7.10}$$

It follows from (2.4.2) that

$$\Gamma_k^{\mathrm{T}} M_k \pi_k = \Gamma_k^{\mathrm{T}} M_k. \tag{2.7.11}$$

Substituting (2.7.11) into (2.7.10) yields

$$\begin{aligned}\Gamma_k^{\mathrm{T}} M_k (P_{k+1} - P_{\mathrm{ol},k+1}) M_k \Gamma_k &= \Gamma_k^{\mathrm{T}} M_k A_k (P_k - P_{\mathrm{ol},k}) A_k^{\mathrm{T}} M_k \Gamma_k - \Gamma_k^{\mathrm{T}} M_k \hat{S}_k \hat{R}_k^{-1} \hat{S}_k^{\mathrm{T}} M_k \Gamma_k \\ &\leqslant 0. \end{aligned}$$

□

Assume that Γ_k has the form (2.2.6). Then, it follows from Proposition 2.7.4 that, if $M_k = I$, that is, all of the states are weighted, then the state estimate in the range of Γ_k obtained using the Kalman filter with constrained output injection are better than the state estimates obtained when data assimilation is not performed. However, state estimates that are not in the range of Γ_k may be worse than estimates obtained when no data assimilation is performed.

2.8 Steady-State Filters for Linear Time-Invariant Systems

Next, we discuss the steady-state behavior of the one-step spatially constrained Kalman filter for linear time-invariant systems. For all $k \geqslant 0$, let $A_k = A$, $B_k = B$, $C_k = C$, $\Gamma_k = \Gamma$, $L_k = L$, $Q_k = Q$, $S_k = 0$ and $R_k = R$. Assuming R is positive definite, it follows from Proposition 4.1 that the optimal gain K_k that minimizes J_k is given by

$$K_k = (\Gamma^{\mathrm{T}} M \Gamma)^{-1} \Gamma^{\mathrm{T}} M A P_k C^{\mathrm{T}} \hat{R}_k, \tag{2.8.1}$$

where

$$\hat{R}_k \triangleq CP_kC^{\mathrm{T}} + R, \quad M \triangleq L^{\mathrm{T}}L. \tag{2.8.2}$$

Furthermore, the covariance update is given by

$$P_{k+1} = AP_kA^{\mathrm{T}} + Q + \pi_\perp AP_kC^{\mathrm{T}}\hat{R}_k^{-1}CP_kA^{\mathrm{T}}\pi_\perp^{\mathrm{T}} - AP_kC^{\mathrm{T}}\hat{R}_k^{-1}CP_kA^{\mathrm{T}}, \tag{2.8.3}$$

where

$$\pi \triangleq \Gamma(\Gamma^{\mathrm{T}}M\Gamma)^{-1}\Gamma^{\mathrm{T}}M, \quad \pi_\perp \triangleq I - \pi. \tag{2.8.4}$$

If $\lim_{k\to\infty} P_k$ exists, then the filtering process reaches statistical steady state. If Γ is square and thus by assumption nonsingular, then $y_k - \hat{y}_k$ is directly injected into all of the estimator states. In this case, the following lemma guarantees the existence of $\lim_{k\to\infty} P_k$.

Lemma 2.8.1 *If Γ is square and (A, C) is detectable, then $P \triangleq \lim_{k\to\infty} P_k$ exists and is positive semidefinite. If, in addition, (A, Q) is stabilizable, then P is positive definite and $A-\Gamma KC$ is asymptotically stable, where $K \triangleq \Gamma^{-1}APC^{\mathrm{T}}(CPC^{\mathrm{T}}+R)^{-1}$.*

Proof. Since Γ is square, it follows from (2.4.2) and (2.4.3) that $\pi = I$ and $\pi_\perp = 0$. Hence, it follows from (2.8.3) that

$$P_{k+1} = AP_kA^{\mathrm{T}} - AP_kC^{\mathrm{T}}(CP_kC^{\mathrm{T}} + R)^{-1}CP_kA^{\mathrm{T}} + Q. \tag{2.8.5}$$

Since (A, C) is detectable, it follows from [34, pp. 100-101] that, if P_0 is positive semidefinite, then $P \triangleq \lim_{k\to\infty} P_k$ exists and satisfies the algebraic Riccati equation

$$P = APA^{\mathrm{T}} - APC^{\mathrm{T}}(CPC^{\mathrm{T}} + R)^{-1}CPA^{\mathrm{T}} + Q. \tag{2.8.6}$$

If (A, C) is detectable and (A, Q) is stabilizable, it follows from [34, pp. 101-103] that P is positive definite and $A - \Gamma KC$ is asymptotically stable. □

When Γ is not square, the existence of $\lim_{k\to\infty} P_k$ is not guaranteed. In fact, we have the following negative result when $\pi \neq I_n$.

Proposition 2.8.1 *Assume that $\pi \neq I_n$ and A is asymptotically stable. Then $\lim_{k\to\infty} P_k$ does not always exist.*

Proof. Consider the example in Proposition 7.2. It follows from (2.8.3) that

$$p_{2,k+1} = p_{2,k}\left(\frac{1}{4} + \frac{1}{100}\left[8(\alpha-1)^2 - 25\right]\frac{p_{2,k}}{1+p_{2,k}}\right). \tag{2.8.7}$$

Hence, if α satisfies

$$(\alpha-1)^2 > 25 \tag{2.8.8}$$

and

$$p_{2,0} > \frac{175}{8(\alpha-1)^2 - 200}, \tag{2.8.9}$$

then, for all $k > 0$, $p_{2,k+1} > 2p_{2,k}$, which implies that $\lim_{k\to\infty} p_{2,k} = \infty$. Hence, if $P_0 \in \mathbb{R}^{2\times 2}$ satisfies (2.8.9), then $\lim_{k\to\infty} P_k$ does not exist. □

Next, we present a converse result concerning the existence of $\lim_{k\to\infty} P_k$. For all $M \in \mathbb{R}^{n\times m}$, let $\mathcal{R}(M)$ denote the range of M.

Proposition 2.8.2 *Assume that (A, Γ) is stabilizable. If $P = \lim_{k\to\infty} P_k$ exists and $\mathcal{R}(\pi APC^{\mathrm{T}}) = \mathcal{R}(\Gamma)$, then (A, Γ, C) is output feedback stabilizable.*

Proof. Letting $k \to \infty$ in (2.8.3) yields

$$P = APA + Q + \pi_\perp APC^{\mathrm{T}}\hat{R}^{-1}CPA^{\mathrm{T}}\pi_\perp^{\mathrm{T}} - APC^{\mathrm{T}}\hat{R}^{-1}CPA^{\mathrm{T}}, \tag{2.8.10}$$

where $\hat{R} \triangleq CPC^{\mathrm{T}} + R$. We can rewrite (2.8.10) as

$$P = APA^{\mathrm{T}} + Q - \Gamma KCPA^{\mathrm{T}} - APC^{\mathrm{T}}K^{\mathrm{T}}\Gamma^{\mathrm{T}} + \Gamma K\hat{R}K^{\mathrm{T}}\Gamma^{\mathrm{T}}, \tag{2.8.11}$$

where

$$K \triangleq (\Gamma^{\mathrm{T}} M \Gamma)^{-1} \Gamma^{\mathrm{T}} M A P C^{\mathrm{T}} \hat{R}^{-1}. \tag{2.8.12}$$

Hence, (2.8.11) can be expressed as

$$P = (A - \Gamma K C) P (A - \Gamma K C)^{\mathrm{T}} + Q + \Gamma K R K^{\mathrm{T}} \Gamma^{\mathrm{T}}. \tag{2.8.13}$$

Next, define $\tilde{A}$ and $\tilde{\Gamma}$ by

$$\tilde{A} \triangleq A - \Gamma K C, \ \tilde{\Gamma} \triangleq \Gamma K R^{1/2}. \tag{2.8.14}$$

Since (A, Γ) is stabilizable and $\mathcal{R}(\Gamma) = \mathcal{R}(\pi A P C^{\mathrm{T}})$, it follows from [36, pp. 510, 551] that $(\tilde{A}, \tilde{\Gamma})$ is also stabilizable. Let $\lambda \in \mathbb{C}$ be an eigenvalue of $\tilde{A}$. Then, there exists an eigenvector $x \in \mathbb{C}^n$ of $\tilde{A}$ such that

$$x^* \tilde{A} = \lambda x^*. \tag{2.8.15}$$

Furthermore, (2.8.13) implies that

$$x^* P x = x^* \tilde{A} P \tilde{A}^{\mathrm{T}} x + x^* (Q + \tilde{\Gamma} \tilde{\Gamma}^{\mathrm{T}}) x. \tag{2.8.16}$$

Substituting (2.8.15) into (2.8.16) yields

$$(1 - |\lambda|^2) x^* P x = x^* (Q + \tilde{\Gamma} \tilde{\Gamma}^{\mathrm{T}}) x. \tag{2.8.17}$$

If $|\lambda| \geqslant 1$, then (2.8.17) implies that

$$x^* (Q + \tilde{\Gamma} \tilde{\Gamma}^{\mathrm{T}}) x = 0 \tag{2.8.18}$$

and hence

$$x^* \tilde{\Gamma} = 0. \tag{2.8.19}$$

It follows from (2.8.15) and (2.8.19) that λ is an unstable and uncontrollable eigenvalue of $(\tilde{A}, \tilde{\Gamma})$, which contradicts the fact that $(\tilde{A}, \tilde{\Gamma})$ is stabilizable. Hence, $|\lambda| < 1$ and $\tilde{A}$ is asymptotically stable. Since K given by (2.8.12) renders $A - \Gamma KC$ asymptotically stable, (A, Γ, C) is output feedback stabilizable. □

The following result provides a sufficient condition for P_k to be bounded when C is square and nonsingular.

Proposition 2.8.3 *Assume that C is square and nonsingular. If*

$$\operatorname{sprad}(\pi_\perp A) < 1, \tag{2.8.20}$$

then P_k is bounded.

Proof. Since C is nonsingular, (2.8.3) can be expressed as

$$P_{k+1} = AP_kA^{\mathrm{T}} + Q + \pi_\perp AP_k(P_k + C^{-1}RC^{-\mathrm{T}})^{-1}P_kA^{\mathrm{T}}\pi_\perp^{\mathrm{T}} - AP_k(P_k + C^{-1}RC^{-\mathrm{T}})^{-1}P_kA^{\mathrm{T}}. \tag{2.8.21}$$

Next, consider the Lyapunov equation

$$\tilde{P}_{k+1} = (A - \Gamma\tilde{K})\tilde{P}_k(A - \Gamma\tilde{K})^{\mathrm{T}} + Q + \Gamma\tilde{K}\tilde{K}^{\mathrm{T}}\Gamma^{\mathrm{T}} + A\tilde{R}A^{\mathrm{T}}, \tag{2.8.22}$$

where

$$\tilde{K} \triangleq (\Gamma^{\mathrm{T}}M\Gamma)^{-1}\Gamma MA \tag{2.8.23}$$

and

$$\tilde{R} \triangleq C^{-1}RC^{-\mathrm{T}}. \tag{2.8.24}$$

Using (2.8.23), we rewrite (2.8.22) as

$$\tilde{P}_{k+1} = \pi_\perp A\tilde{P}_kA^{\mathrm{T}}\pi_\perp^{\mathrm{T}} + Q + \pi AA^{\mathrm{T}}\pi^{\mathrm{T}} + A\tilde{R}A^{\mathrm{T}}. \tag{2.8.25}$$

Since $\pi_\perp A$ is asymptotically stable and $Q+\pi AA^{\mathrm{T}}\pi^{\mathrm{T}}+A\tilde{R}A^{\mathrm{T}}$ is positive semidefinite, $\tilde{P} = \lim_{k\to\infty}\tilde{P}_k$ exists for all positive-semidefinite $\tilde{P}_0$. Subtracting (2.8.21) from (2.8.25) yields

$$\tilde{P}_{k+1} - P_{k+1} = A\tilde{R}(\tilde{R}+P_k)^{-1}\tilde{R}A^{\mathrm{T}} + \pi AA^{\mathrm{T}}\pi^{\mathrm{T}} + \pi_\perp AP_k(P_k+\tilde{R})^{-1}\tilde{R}A^{\mathrm{T}}\pi_\perp^{\mathrm{T}} + \pi_\perp A(\tilde{P}_k - P_k)A^{\mathrm{T}}\pi_\perp^{\mathrm{T}}. \tag{2.8.26}$$

It follows from (2.8.26) that, if $\tilde{P}_k \geqslant \tilde{P}_k$, then $\tilde{P}_{k+1} \geqslant P_{k+1}$. Hence, if $P_0 \leqslant \tilde{P}_0$, then $P_k \leqslant \tilde{P}_k$ for all $k > 0$. Furthermore, since $\tilde{P}_k$ converges to $\tilde{P}$ for every choice of $\tilde{P}_0$, it follows that P_k is bounded. □

Numerical results suggest that the following strengthening of Proposition 8.3 is true.

Conjecture 2.8.1 *Assume that C is square and nonsingular. If*

$$\mathrm{sprad}(\pi_\perp A) < 1, \tag{2.8.27}$$

then $\lim_{k\to\infty} P_k$ exists.

Example 2.8.1 *Let*

$$A = \begin{bmatrix} 0 & 5 \\ 0 & 3 \end{bmatrix}, C = I, Q = 0, R = I, \tag{2.8.28}$$

and choose

$$\Gamma = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}, \tag{2.8.29}$$

where $\gamma_1, \gamma_2 \in \mathbb{R}$ so that

$$\pi = \frac{1}{\gamma_1^2+\gamma_2^2}\begin{bmatrix} \gamma_1^2 & \gamma_1\gamma_2 \\ \gamma_1\gamma_2 & \gamma_2^2 \end{bmatrix}, \quad \pi_\perp = \frac{1}{\gamma_1^2+\gamma_2^2}\begin{bmatrix} \gamma_2^2 & -\gamma_1\gamma_2 \\ -\gamma_1\gamma_2 & \gamma_1^2 \end{bmatrix}. \tag{2.8.30}$$

Note that

$$\pi_\perp A = \frac{1}{\gamma_1^2+\gamma_2^2}\begin{bmatrix} 0 & 5\gamma_2^2-3\gamma_1\gamma_2 \\ 0 & 3\gamma_1^2-5\gamma_1\gamma_2 \end{bmatrix} \tag{2.8.31}$$

and hence

$$\mathrm{sprad}(\pi_\perp A) = \frac{1}{\gamma_1^2+\gamma_2^2}|3\gamma_1^2-5\gamma_1\gamma_2|. \tag{2.8.32}$$

It follows from Conjecture 2.8.1 that, if

$$-(\gamma_1^2+\gamma_2^2) < 3\gamma_1^2-5\gamma_1\gamma_2 < \gamma_1^2+\gamma_2^2, \tag{2.8.33}$$

then $\lim_{k\to\infty} P_k$ exists. The shaded region in Figure 2.1 indicates values of γ_1 and γ_2 that satisfy (2.8.33). Next, we choose various values of γ_1, γ_2 and numerically evaluate P_k as $k \to \infty$ using (2.8.3). The values of γ_1, γ_2 for which $\lim_{k\to\infty} P_k$ exists, are indicated by '•' and the values of γ_1, γ_2 for which $\lim_{k\to\infty} P_k$ does not exist are indicated by '×'. The numerical results are consistent with Conjecture 8.1.

2.9 N-Mass System Example

Consider the N-mass system shown in Figure 2.2 with stiffnesses $k_1, \ldots, k_{N+1} > 0$ and dashpots $c_1, \ldots, c_{N+1} > 0$. Let q_i denote the position of mass m_i. Define

$$q \triangleq \begin{bmatrix} q_1 & \cdots & q_N \end{bmatrix}^{\mathrm{T}}, \; M \triangleq \mathrm{diag}(m_1, \ldots, m_N). \tag{2.9.1}$$

$$K \triangleq \begin{bmatrix} k_1+k_2 & -k_2 & 0 & \cdots & 0 & 0 \\ -k_2 & k_2+k_3 & -k_3 & \cdots & 0 & 0 \\ 0 & -k_3 & k_3+k_4 & \cdots & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -k_N & k_N+k_{N+1} \end{bmatrix}, \tag{2.9.2}$$

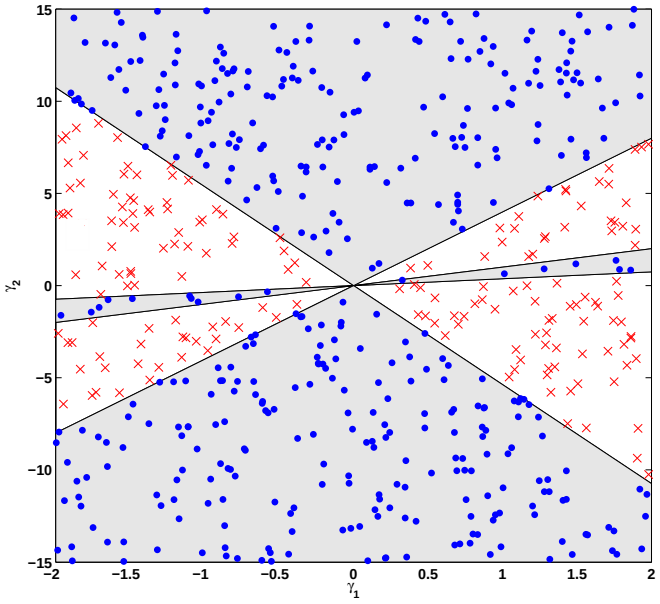


Figure 2.1: The shaded region indicates the values of γ_1, γ_2 that satisfy (2.8.33). The dots indicate the values of γ_1, γ_2 for which $\lim_{k\to\infty} P_k$ exists, whereas the values of γ_1, γ_2 for which $\lim_{k\to\infty} P_k$ does not exist are indicated by '$\times$'. These numerical results are consistent with Conjecture 2.8.1.

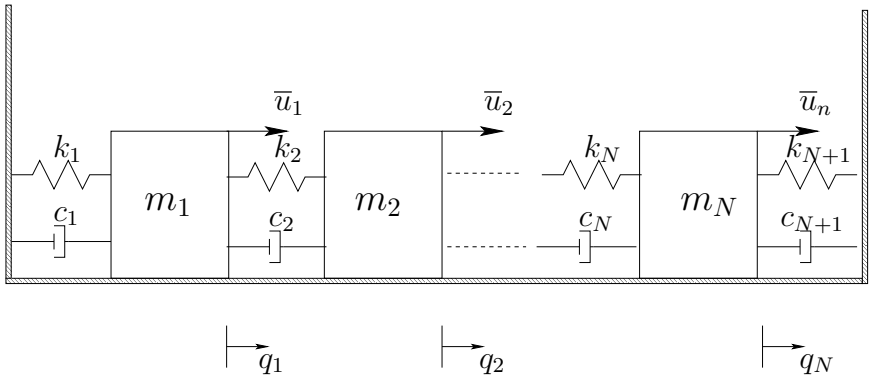


Figure 2.2: N-Mass System

$$C \triangleq \begin{bmatrix} c_1 + c_2 & -c_2 & 0 & \cdots & 0 & 0 \\ -c_2 & c_2 + c_3 & -c_3 & \cdots & 0 & 0 \\ 0 & -c_3 & c_3 + c_4 & \cdots & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -c_N & c_N + c_{N+1} \end{bmatrix}, \tag{2.9.3}$$

We assume that d masses are disturbed by unknown force inputs $w \in \mathbb{R}^d$, which are zero-mean white noise with unit intensity, while p masses are actuated by known force inputs $u \in \mathbb{R}^p$. Let u and w have entries

$$u = \begin{bmatrix} u_1 & \cdots & u_p \end{bmatrix}^{\mathrm{T}}, \quad w \triangleq \begin{bmatrix} w_1 & \cdots & w_d \end{bmatrix}^{\mathrm{T}} \tag{2.9.4}$$

and let $\mathcal{B}_u$ and $\mathcal{D}_w$ have entries

$$\mathcal{B}_u = \begin{bmatrix} \mathcal{B}_{u,1} & \cdots & \mathcal{B}_{u,p} \end{bmatrix}, \ \mathcal{D}_w = \begin{bmatrix} \mathcal{D}_{w,1} & \cdots & \mathcal{D}_{w,d} \end{bmatrix}, \tag{2.9.5}$$

where, for all $i = 1, \ldots, p$ and $j = 1, \ldots, d$, $\mathcal{B}_{u,i}$ and $\mathcal{D}_{w,j}$ are defined by

$$\mathcal{B}_{u,i} = \begin{bmatrix} 0_{1\times\hat{i}-1} & \frac{1}{m_{\hat{i}}} & 0_{1\times N-\hat{i}} \end{bmatrix}^{\mathrm{T}}, \ \mathcal{D}_{w_j} = \begin{bmatrix} 0_{1\times\hat{j}-1} & \frac{1}{m_{\hat{j}}} & 0_{1\times N-\hat{j}} \end{bmatrix}^{\mathrm{T}} \tag{2.9.6}$$

and $\hat{i}$ and $\hat{j}$ correspond to the masses on which forces u_i and w_j act, respectively. The equations of motion can be written in first-order form as

$$\dot{x} = \mathcal{A}x + \mathcal{B}u + \mathcal{D}_1 w, \tag{2.9.7}$$

where $\mathcal{A} \in \mathbb{R}^{2N\times 2N}$, $\mathcal{B} \in \mathbb{R}^{2N\times m}$, $\mathcal{D}_1 \in \mathbb{R}^{2N\times d}$, and $x \in \mathbb{R}^{2N}$ are defined by

$$\begin{aligned} \mathcal{A} &\triangleq \begin{bmatrix} 0_N & I_N \\ -M^{-1}K & -M^{-1}C \end{bmatrix}, \ \mathcal{B} \triangleq \begin{bmatrix} 0_N \\ \mathcal{B}_u \end{bmatrix}, \ \mathcal{D}_1 \triangleq \begin{bmatrix} 0_N \\ \mathcal{D}_w \end{bmatrix}, \\ x &\triangleq \begin{bmatrix} q_1 & \cdots & q_N & \dot{q}_1 & \cdots & \dot{q}_N \end{bmatrix}^{\mathrm{T}}. \end{aligned} \tag{2.9.8}$$

Next, we assume that measurements of the positions of l masses are available so that the output $y \in \mathbb{R}^l$ can be expressed as

$$y = C_{\text{pos}} x + v, \tag{2.9.9}$$

where $C_{\text{pos}} \in \mathbb{R}^{l \times 2N}$ has entries

$$C_{\text{pos}} = \begin{bmatrix} C_{\text{pos}}^{[1]} \\ \vdots \\ C_{\text{pos}}^{[l]} \end{bmatrix} \tag{2.9.10}$$

and, for all $i = 1, \ldots, N$, $C_{\text{pos}}^{[i]} \in \mathbb{R}^{1 \times 2N}$ is defined by

$$C_{\text{pos}}^{[i]} \triangleq \begin{bmatrix} 0_{1\times(\hat{i}-1)} & 1 & 0_{1\times(N-\hat{i})} & 0_{1\times N} \end{bmatrix}, \tag{2.9.11}$$

where $\hat{i}$ corresponds to the index of the mass whose position measurements are available. With the sampling time $t_{\text{s}} = 0.1$ s, we obtain the zero-order-hold discrete-time model of (2.9.7) and (2.9.9) given by

$$x_{k+1} = Ax_k + Bu_k + D_1 w_k, \tag{2.9.12}$$

$$y_k = C_{\text{pos}} x_k + v_k. \tag{2.9.13}$$

Signal	Masses
Known force input u	m_1, m_5, m_{10}
Unknown force input w	m_4, m_{15}, m_{18}
Position measurement y	m_9, m_{12}

Table 2.1: Forcing and measurement signals in the N-mass system.

Let $N = 20$, so that the (2.9.7) has order $n = 40$ with known inputs $u \in \mathbb{R}^3$ and unknown inputs $w \in \mathbb{R}^3$. We assume that w is zero-mean white Gaussian noise

with unit covariance, and the known inputs $u \in \mathbb{R}^3$ are chosen to be sinusoids. The masses on which w and u act and the available measurements are given in Table 1. We assume that the process noise and the measurement sensor noise are uncorrelated and hence $S_k = 0$. The values of the masses $m_1, \ldots, m_{20}$, damping coefficients $c_1, \ldots, c_{21}$, and spring constants $k_1, \ldots, k_{21}$ are $m_i = 10$ kg for $i = 1, \ldots, 20$, $c_i = 0.8$ N-s/m and $k_i = 5$ N/m for $i = 1, \ldots, 21$. Finally, we assume that the process noise and sensor noise are uncorrelated, that is, $S_k = 0$ for all $k \geqslant 0$. Next, we obtain estimates of the position and velocity of $m_1, \ldots, m_{20}$ using two sets of measurements y, one with a signal to noise ratio (SNR) of 20 dB and another with a SNR of 1 dB. The measurements of position of m_9 and m_{12} with different signal to noise rations are shown in Figure 2.3.

We first choose $\mathit{\Gamma}_k = I_{2N}$ and $L_k = I_{2N}$, that is, the available measurements are injected into all of the states of the estimator, and the errors between all of the states and the corresponding state estimates are weighted. In this case, the one-step and two-step Kalman filters are equivalent. The state estimates are obtained using the two-step filter (2.5.31)-(2.5.34). The root mean square (RMS) value of the error in the estimates of position of m_4 when measurements with a signal to noise ratio of 20 dB and 1 dB, respectively, are used is shown in Figure 2.4. The RMS value of the errors in position and velocity estimates of $m_1, \ldots, m_{20}$ are plotted in Figure 2.5 and Figure 2.6, respectively.

Next, we obtain estimates by constraining the output injection into only some of the states of the estimator. First, we choose $\mathit{\Gamma}_k = \mathit{\Lambda}_1$ for all $k \geqslant 0$, where

$$\mathit{\Lambda}_1 \triangleq \begin{bmatrix} 0_{24\times 8} & I_{24} & 0_{24\times 8} \end{bmatrix}^{\mathrm{T}} \tag{2.9.14}$$

so that the measurements are injected into only the estimates of the positions and

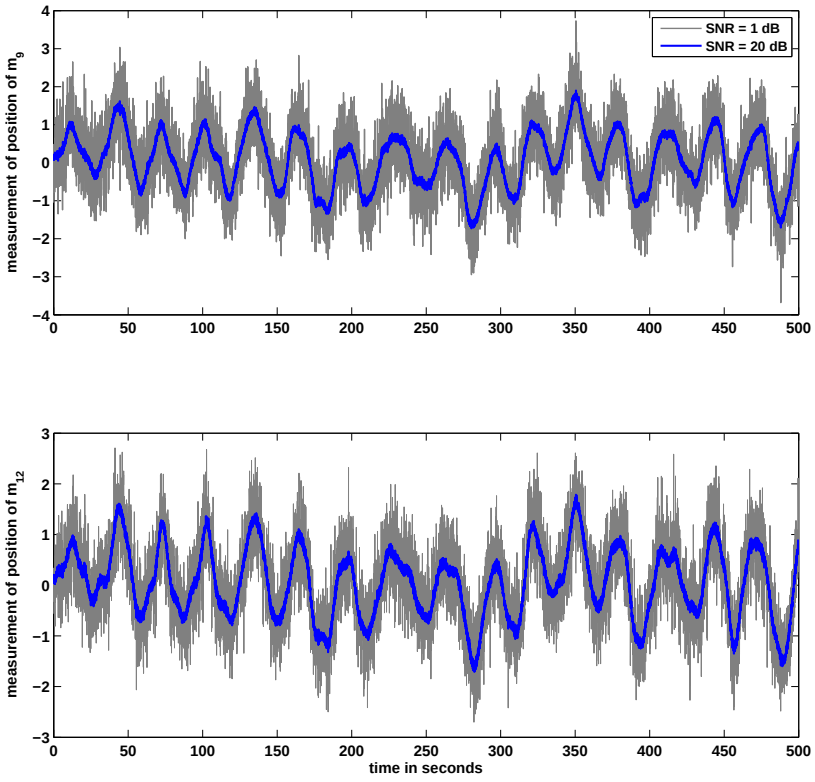


Figure 2.3: Noisy measurements of the positions of m_9 and m_{12} with SNR = 20 db and SNR = 1 dB. These measurements are used to estimate the positions and velocities of masses $m_1, \ldots, m_{20}$.

velocities of $m_5, \ldots, m_{16}$. Furthermore, we choose $L_k = I_{2N}$ so that the errors in all of the state estimates are weighted equally. The RMS value of the error in the position estimate of m_4 obtained when $\Gamma_k = \Lambda_1$ for all $k \geqslant 0$ is shown in Figure 2.4. The RMS value of the errors in position and velocity estimates of $m_1, \ldots, m_{20}$, are shown in Figure 2.5 and Figure 2.6, respectively. Finally, we choose $\Gamma_k = \Lambda_2$ for all $k \geqslant 0$, where

$$\Lambda_2 \triangleq \begin{bmatrix} 0_{8\times 16} & I_8 & 0_{8\times 16} \end{bmatrix}^{\mathrm{T}} \tag{2.9.15}$$

so that only the estimates of the positions and velocities of $m_9, \ldots, m_{12}$ are directly affected by the measurements y. Again, we choose $L_k = I_{2N}$ for all $k \geqslant 0$, and the performance of the estimator with $\Gamma_k = \Lambda_2$ for all $k \geqslant 0$ is shown in Figure 2.4, Figure 2.5 and Figure 2.6.

When $\Gamma_k = I_{2N}$, the measurements are injected directly into all of the states of the estimator, and Figure 2.4 confirms the expected fact that the performance of the classical Kalman filter with $\Gamma_k = I_{2N}$ is better than the estimators with $\Gamma_k \neq I_{2N}$. Note that the number of states into which measurements are injected when $\Gamma_k = \Lambda_2$ is less than the number of states that are directly affected by measurements when $\Gamma_k = \Lambda_1$, and it follows from Figure 2.4 that reducing the number of estimator states that are directly affected by measurements degrades the performance of the estimator. These observations are consistent with Proposition 2.5.3.

Although the errors in the position and velocity estimates of all of the masses are weighted in all three cases $\Gamma_k = I_{2N}$, $\Gamma_k = \Lambda_1$, and $\Gamma_k = \Lambda_2$, Figure 2.5 and Figure 2.6 demonstrate that the error in the position and velocity estimates of all of the masses is the least when $\Gamma_k = I_{2N}$ and the measurements are directly injected into all of the estimator states. Finally, it can be seen that when the measurements

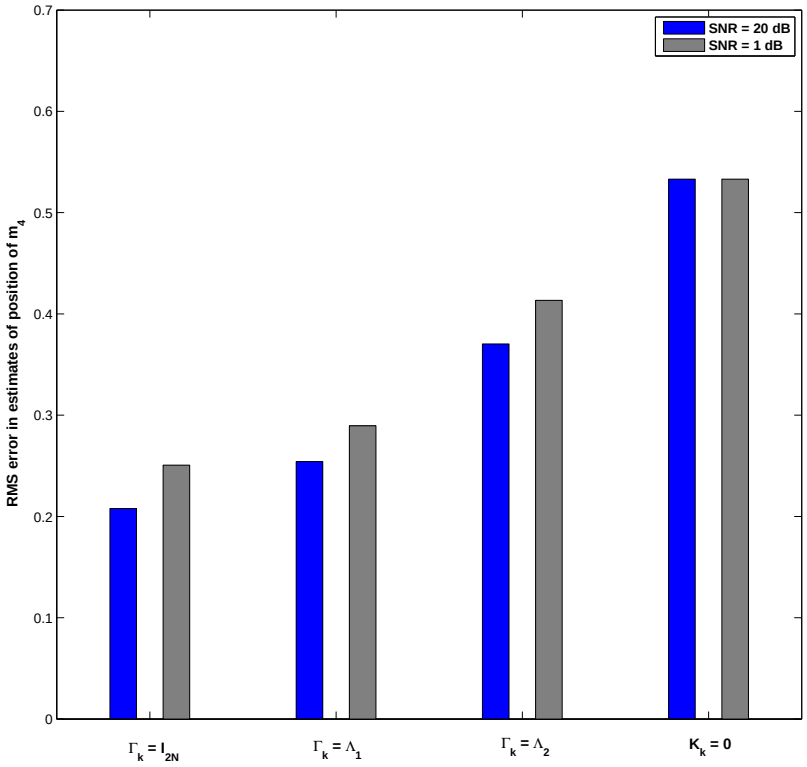


Figure 2.4: Root mean square value of the error in estimating the position of m_4 obtained using the two-step filter with $\Gamma_k = I_{2N}$ (classical Kalman filter) and $\Gamma_k \neq I_{2N}$ using two different sets of measurements, one with SNR= 20 dB and another with SNR = 1 dB. When $\Gamma_k = \Lambda_1$, measurements are directly injected into the estimates of only the positions and velocities of masses $m_5, \ldots, m_{16}$, whereas when $\Gamma_k = \Lambda_2$, measurements are directly injected into estimates of only the positions and velocities of masses $m_9, \ldots, m_{12}$. As expected, the performance of the estimators with constrained output injection ($\Gamma_k \neq I$) is not as good as the estimator with $\Gamma_k = I_{2N}$. Since the zero-gain filter does not use the measurements, its performance does not depend on the value of the SNR of the measurements.

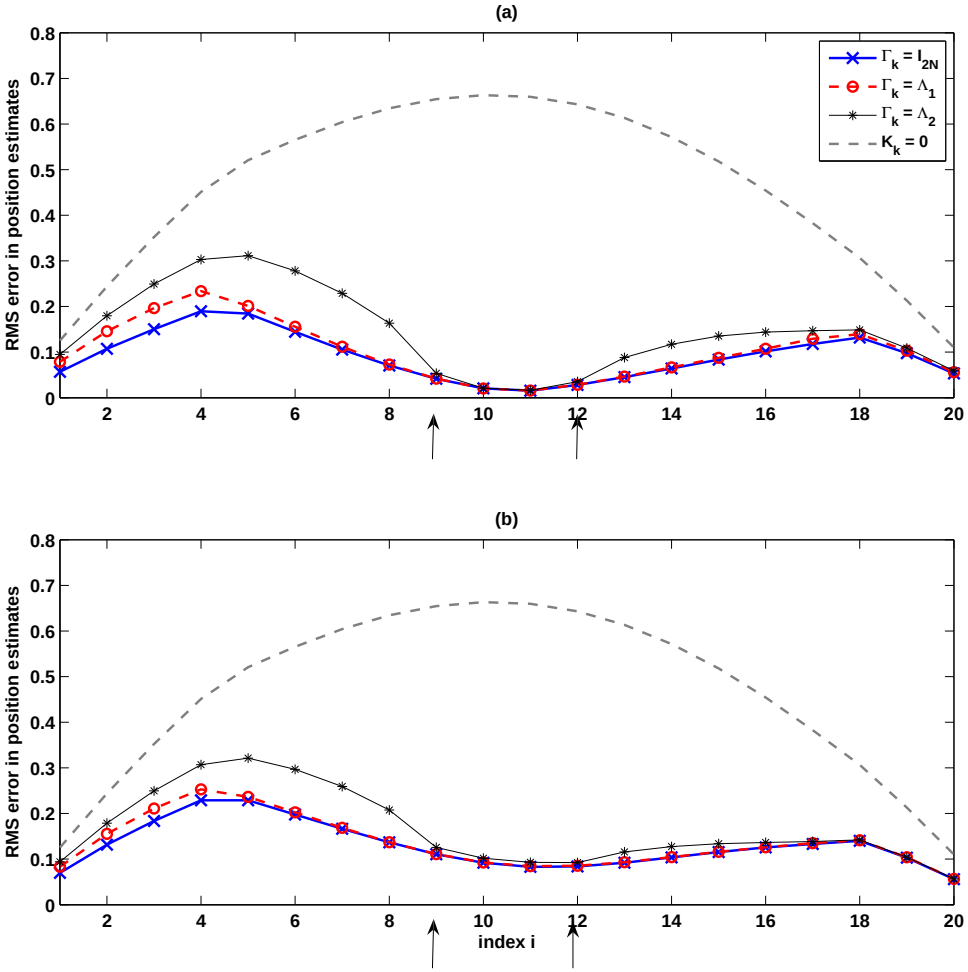


Figure 2.5: RMS value of the errors in the position estimates of all of the masses when measurements with (a) SNR =20 dB and (b) SNR = 1 dB are injected into all of the state estimates ($\Gamma_k = I_{2N}$) and when measurements are injected into only the position and velocity estimates of some of the masses ($\Gamma_k \neq I_{2N}$). The performance of the zero-gain filter with $K_k \equiv 0$ is also shown for comparison. When measurements are injected into a larger number of the estimator states, the performance of the estimator improves. The arrows indicate the masses whose position measurements are available. As the SNR of the measurement increases, the difference in the performance of the filters with $\Gamma_k = I_{2N}$ and $\Gamma_k \neq I_{2N}$ decreases.

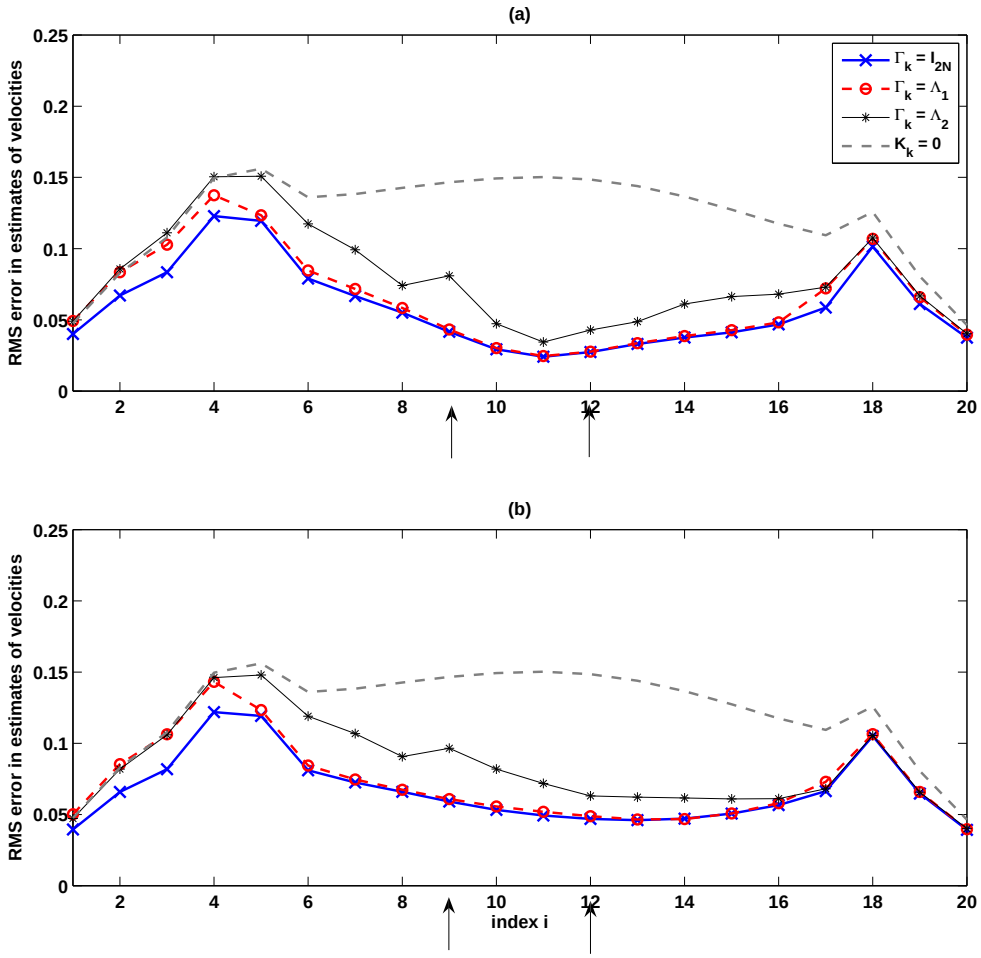


Figure 2.6: RMS value of the errors in the velocity estimates from the optimal filter with $\Gamma_k = I_{2N}$ and $\Gamma_k \neq I_{2N}$ when measurements with (a) SNR = 20 dB and (b) SNR = 1 dB are used. When $\Gamma_k \neq I_{2N}$, the one-step and two-step filters are not equivalent, and the results presented here are obtained using the two-step estimator. The performance of the estimators with $\Gamma_k \neq I_{2N}$ improves when additional states of the estimator are directly injected with measurements.

are injected into a subset of the estimator states, then the estimates of the states that are not directly affected by the measurements improve. The performance of the zero-gain filter with $K_k = 0$ for all $k \geqslant 0$ is also plotted in figures 2.4, 2.5 and 2.6 for comparison.

2.10 Conclusions

In this chapter, we presented an extension of the Kalman filter that constrains data injection into only a specified subset of state estimates rather than the entire state estimate. This extension accounts for correlation between the process noise and the sensor noise. Conditions are given under which the one-step and two-step forms of the filter are equivalent. Future work will consider reduced-rank square root formulations of this filter to reduce the computational burden of propagating the covariance. More general conditions that guarantee the existence of a steady-state covariance for linear time-invariant dynamics are also of interest. Although we constrain output injection, the order of the estimator dynamics is equal to the order of the plant dynamics. In the next chapter, we do not constrain output injection. Instead, we obtain state estimates of a specific subset of the state by using a reduced-order model of the plant dynamics.

CHAPTER III

Reduced-Order Kalman Filtering for Time-Varying Systems

The previous chapter considered a full-order estimator, that is, the order of the dynamics of the estimator was the same as the order of the plant dynamics. In this chapter, we consider a reduced-order estimator for state estimation of linear time-varying systems with time-varying state dimension. A reduced-order estimator provides an estimate of a specific subset of the state, and uses a reduce-order model of the plant dynamics to propagate the state estimates. We assume that a white noise process affects the plant dynamics and also assume that measurements are corrupted by sensor noise. In this chapter, we derive the optimal reduced-order estimator using a finite-horizon approach. The resulting reduced-order estimator involves two covariance update equations, one that resembles the discrete-time Lyapunov equation, and another that resembles the discrete-time Riccati equation. The results presented in this chapter can be found in [37].

3.1 Introduction

Since 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 [38].

One approach to this problem is to consider reduced-order Kalman filters. These reduced-complexity filters provide state estimates that are suboptimal relative to the classical Kalman filter [7, 8, 25, 26]. Alternative variants of the classical Kalman filter have been developed for computationally demanding applications such as weather forecasting [27, 29, 30, 35], where the classical Kalman filter gain and covariance are modified so as to reduce the computational requirements. A comparison of various techniques is given in [9].

An alternative approach to reducing complexity is to restrict the data-injection subspace to obtain a spatially localized Kalman filter. This approach is developed in [23, 31] and discussed in Chapter II.

In this chapter, we revisit the approach of [7, 39], which consider 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 by coupled Riccati and Lyapunov equations, whose solution requires iterative techniques.

We extend the results of [7, 39] by adopting the finite-horizon optimization technique used in [23, 24] to obtain reduced-order filters that are applicable to time-varying systems. The time-varying filter gains are given by recursive update equations that account for the restricted order of the filter but do not require iterative solution methods. This technique also avoids the periodicity constraint associated with the multirate filter derived in [40]. Related techniques are used in [41].

3.2 Finite-Horizon Discrete-Time Optimal Reduced-Order Estimator

Consider the system

$$x_{k+1} = A_k x_k + D_{1,k} w_k, \tag{3.2.1}$$

$$y_k = C_k x_k + D_{2,k} w_k, \tag{3.2.2}$$

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. We assume for convenience that $D_{1,k} D_{2,k}^{\mathrm{T}} = 0$.

We consider a reduced-order estimator with dynamics

$$x_{\mathrm{e},k+1} = A_{\mathrm{e},k} x_{\mathrm{e},k} + B_{\mathrm{e},k} y_k, \tag{3.2.3}$$

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

$$\tilde{Q}_k \triangleq \mathcal{E}[\tilde{x}_k \tilde{x}_k^{\mathrm{T}}], \tag{3.2.4}$$

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

$$\tilde{x}_k \triangleq \begin{bmatrix} x_k \\ x_{\mathrm{e},k} \end{bmatrix}. \tag{3.2.5}$$

Consider the cost function

$$J_k \triangleq \mathcal{E}\Big[\big(L_k x_{k+1} - x_{\mathrm{e},k+1}\big)^{\mathrm{T}} \big(L_k x_{k+1} - x_{\mathrm{e},k+1}\big)\Big], \tag{3.2.6}$$

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 (3.2.4) and (3.2.5) that J_k can be expressed as

$$J_k = \mathrm{tr}\left(\tilde{Q}_{k+1} \tilde{R}_k\right), \tag{3.2.7}$$

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

$$\tilde{R}_k \triangleq \begin{bmatrix} L_k^{\mathrm{T}} L_k & -L_k^{\mathrm{T}} \\ -L_k & I \end{bmatrix}. \tag{3.2.8}$$

Note that (3.2.1) and (3.2.3) imply that

$$\tilde{x}_{k+1} = \tilde{A}_k \tilde{x}_k + \tilde{D}_{1,k} w_k, \tag{3.2.9}$$

where

$$\tilde{A}_k \triangleq \begin{bmatrix} A_k & 0 \\ B_{\mathrm{e},k} C_k & A_{\mathrm{e},k} \end{bmatrix}, \quad \tilde{D}_{1,k} \triangleq \begin{bmatrix} D_{1,k} \\ B_{\mathrm{e},k} D_{2,k} \end{bmatrix}. \tag{3.2.10}$$

Therefore,

$$\tilde{Q}_{k+1} = \tilde{A}_k \tilde{Q}_k \tilde{A}_k^{\mathrm{T}} + \tilde{V}_{1,k}, \tag{3.2.11}$$

where

$$\tilde{V}_{1,k} \triangleq \begin{bmatrix} V_{1,k} & 0 \\ 0 & B_{\mathrm{e},k} V_{2,k} B_{\mathrm{e},k}^{\mathrm{T}} \end{bmatrix}, \tag{3.2.12}$$

and

$$V_{1,k} \triangleq D_{1,k} D_{1,k}^{\mathrm{T}}, \quad V_{2,k} \triangleq D_{2,k} D_{2,k}^{\mathrm{T}}. \tag{3.2.13}$$

Partition $\tilde{Q}_k$ as

$$\tilde{Q}_k = \begin{bmatrix} \tilde{Q}_{1,k} & \tilde{Q}_{12,k} \\ \tilde{Q}_{12,k}^{\mathrm{T}} & \tilde{Q}_{2,k} \end{bmatrix}. \tag{3.2.14}$$

Hence, it follows from (3.2.11) that

$$\tilde{Q}_{1,k+1} = A_k \tilde{Q}_{1,k} A_k^{\mathrm{T}} + V_{1,k}, \tag{3.2.15}$$

$$\tilde{Q}_{12,k+1} = A_k \tilde{Q}_{1,k} C_k^{\mathrm{T}} B_{\mathrm{e},k}^{\mathrm{T}} + A_k \tilde{Q}_{12,k} A_{\mathrm{e},k}^{\mathrm{T}}, \tag{3.2.16}$$

$$\begin{aligned} \tilde{Q}_{2,k+1} = {} & B_{\mathrm{e},k} \left(C_k \tilde{Q}_{1,k} C_k^{\mathrm{T}} + V_{2,k} \right) B_{\mathrm{e},k}^{\mathrm{T}} \\ & + A_{\mathrm{e},k} \tilde{Q}_{12,k}^{\mathrm{T}} C_k^{\mathrm{T}} B_{\mathrm{e},k}^{\mathrm{T}} + B_{\mathrm{e},k} C_k \tilde{Q}_{12,k} A_{\mathrm{e},k}^{\mathrm{T}} + A_{\mathrm{e},k} \tilde{Q}_{2,k} A_{\mathrm{e},k}. \end{aligned} \tag{3.2.17}$$

Therefore, (3.2.7) and (3.2.8) imply that J_k can be expressed as

$$\begin{aligned} J\left(A_{\mathrm{e},k}, B_{\mathrm{e},k}\right) = \mathrm{tr}\left[L_k\left(A_k\tilde{Q}_{1,k}A_k^{\mathrm{T}} + V_{1,k}\right)L_k^{\mathrm{T}}\right] - 2\mathrm{tr}\left[B_{\mathrm{e},k}C_k\tilde{Q}_{1,k}A_k^{\mathrm{T}}L_k^{\mathrm{T}}\right] \\ - 2\mathrm{tr}\left[A_{\mathrm{e},k}\tilde{Q}_{12,k}^{\mathrm{T}}A_k^{\mathrm{T}}L_k^{\mathrm{T}}\right] + \mathrm{tr}\left[B_{\mathrm{e},k}\left(C_k\tilde{Q}_{1,k}C_k^{\mathrm{T}} + V_{2,k}\right)B_{\mathrm{e},k}^{\mathrm{T}}\right] \\ + \mathrm{tr}\left[A_{\mathrm{e},k}\tilde{Q}_{2,k}A_{\mathrm{e},k}^{\mathrm{T}}\right] + 2\mathrm{tr}\left[A_{\mathrm{e},k}\tilde{Q}_{12,k}^{\mathrm{T}}C_k^{\mathrm{T}}B_{\mathrm{e},k}^{\mathrm{T}}\right]. \end{aligned} \tag{3.2.18}$$

Proposition 3.2.1 *Assume that $A_{\mathrm{e},k}$ and $B_{\mathrm{e},k}$ minimize J_k. Then, $A_{\mathrm{e},k}$ and $B_{\mathrm{e},k}$ satisfy*

$$A_{\mathrm{e},k}\tilde{Q}_{2,k} = \left(L_kA_k - B_{\mathrm{e},k}C_k\right)\tilde{Q}_{12,k}, \tag{3.2.19}$$

$$B_{\mathrm{e},k} = \left(L_kA_k\tilde{Q}_{1,k} - A_{\mathrm{e},k}\tilde{Q}_{12,k}^{\mathrm{T}}\right)C_k^{\mathrm{T}}\left(C_k\tilde{Q}_{1,k}C_k^{\mathrm{T}} + V_{2,k}\right)^{-1}. \tag{3.2.20}$$

Proof. Setting $\frac{\partial J_k}{\partial A_{\mathrm{e},k}} = 0$ and $\frac{\partial J_k}{\partial B_{\mathrm{e},k}} = 0$ yields the result. □

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

$$Q_k \triangleq \tilde{Q}_{1,k} - \tilde{Q}_{12,k}\tilde{Q}_{2,k}^{-1}\tilde{Q}_{12,k}^{\mathrm{T}}, \quad \hat{Q}_k \triangleq \tilde{Q}_{12,k}\tilde{Q}_{2,k}^{-1}\tilde{Q}_{12,k}^{\mathrm{T}}, \tag{3.2.21}$$

$\tilde{V}_{2,k} \in \mathbb{R}^{p_k\times p_k}$ by

$$\tilde{V}_{2,k} \triangleq C_kQ_kC_k^{\mathrm{T}} + V_{2,k}, \tag{3.2.22}$$

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

$$G_k \triangleq \tilde{Q}_{2,k}^{-1}\tilde{Q}_{12,k}^{\mathrm{T}}. \tag{3.2.23}$$

Proposition 3.2.2 *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*

$$A_{\mathrm{e},k} = L_kA_k\left(I - Q_kC_k^{\mathrm{T}}\tilde{V}_{2,k}^{-1}C_k\right)G_k^{\mathrm{T}}, \tag{3.2.24}$$

$$B_{\mathrm{e},k} = L_kA_kQ_kC_k^{\mathrm{T}}\tilde{V}_{2,k}^{-1}. \tag{3.2.25}$$

Proof. It follows from (3.2.19) that

$$A_{\mathrm{e},k} = (L_k A_k - B_{\mathrm{e},k} C_k)\, \tilde{Q}_{12,k} \tilde{Q}_{2,k}^{-1}. \tag{3.2.26}$$

Substituting (3.2.26) into (3.2.20) yields (3.2.25). Finally, substituting (3.2.25) into (3.2.26) yields (3.2.24). □

Proposition 3.2.3 *Assume that $A_{\mathrm{e},k}$ and $B_{\mathrm{e},k}$ satisfy Proposition 3.2.2. Then,*

$$L_k \tilde{Q}_{12,k+1} = \tilde{Q}_{2,k+1}, \tag{3.2.27}$$

$$\tilde{Q}_{12,k+1} = \hat{Q}_{k+1} L_k^{\mathrm{T}}, \tag{3.2.28}$$

$$\tilde{Q}_{2,k+1} = L_k \hat{Q}_{k+1} L_k^{\mathrm{T}}. \tag{3.2.29}$$

Proof. Substituting (3.2.24) and (3.2.25) into (3.2.16) and (3.2.17) yields

$$\tilde{Q}_{12,k+1} = A_k \left[\hat{Q}_k + Q_k C_k^{\mathrm{T}} \tilde{V}_{2,k}^{-1} C_k Q_k \right] A_k^{\mathrm{T}} L_k^{\mathrm{T}}, \tag{3.2.30}$$

$$\tilde{Q}_{2,k+1} = L_k A_k \left[\hat{Q}_k + Q_k C_k^{\mathrm{T}} \tilde{V}_{2,k}^{-1} C_k Q_k \right] A_k^{\mathrm{T}} L_k^{\mathrm{T}}. \tag{3.2.31}$$

Pre-multiplying (3.2.30) by L_k yields $L_k \tilde{Q}_{12,k+1} = \tilde{Q}_{2,k+1}$. Using (3.2.21) 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

$$M_k \triangleq A_k \left(\hat{Q}_k + Q_k C_k^{\mathrm{T}} \tilde{V}_{2,k}^{-1} C_k Q_k \right) A_k^{\mathrm{T}}, \tag{3.2.32}$$

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

$$\tau_k \triangleq G_k^{\mathrm{T}} L_{k-1}, \quad \tau_{k\perp} \triangleq I - \tau_k. \tag{3.2.33}$$

Proposition 3.2.4 *Assume that $A_{\mathrm{e},k}$ and $B_{\mathrm{e},k}$ satisfy Proposition 3.2.2. Then, $\tau_{k+1}^2 = \tau_{k+1}$, that is, τ_{k+1} is an oblique projector.*

Proof. It follows from (3.2.32) that (3.2.30) and (3.2.31) can be expressed as

$$\tilde{Q}_{12,k+1} = M_k L_k^{\mathrm{T}}, \tag{3.2.34}$$

$$\tilde{Q}_{2,k+1} = L_k M_k L_k^{\mathrm{T}}. \tag{3.2.35}$$

Hence, (3.2.23) and (3.2.33) imply that

$$\tau_{k+1} = M_k L_k^{\mathrm{T}} \left(L_k M_k L_k^{\mathrm{T}}\right)^{-1} L_k. \tag{3.2.36}$$

Therefore,

$$\tau_{k+1}^2 = \tau_{k+1}. \qquad \square$$

Proposition 3.2.5 *Assume that $A_{\mathrm{e},k}$ and $B_{\mathrm{e},k}$ satisfy Proposition 3.2.2. Then,*

$$\tau_{k+1}\hat{Q}_{k+1} = \hat{Q}_{k+1}. \tag{3.2.37}$$

Proof. It follows from (3.2.21) that

$$\hat{Q}_{k+1} = \tilde{Q}_{12,k+1}\tilde{Q}_{2,k+1}^{-1}\tilde{Q}_{12,k+1}^{\mathrm{T}}. \tag{3.2.38}$$

Substituting (3.2.34) and (3.2.35) into (3.2.38) yields

$$\hat{Q}_{k+1} = M_k L_k^{\mathrm{T}} \left(L_k M_k L_k^{\mathrm{T}}\right)^{-1} L_k M_k. \tag{3.2.39}$$

Hence, pre-multiplying (3.2.39) by τ_{k+1} and substituting (3.2.36) into the resulting expression yields

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

Proposition 3.2.6 *Assume that $A_{\mathrm{e},k}$ and $B_{\mathrm{e},k}$ satisfy Proposition 3.2.2. Then,*

$$\begin{aligned} Q_{k+1} &= A_k Q_k A_k^{\mathrm{T}} + V_{1,k} - A_k Q_k C_k^{\mathrm{T}} \tilde{V}_{2,k}^{-1} C_k Q_k A_k^{\mathrm{T}} \\ &\quad + \tau_{k+1\perp} \left[A_k \hat{Q}_k A_k^{\mathrm{T}} + A_k Q_k C_k^{\mathrm{T}} \tilde{V}_{2,k}^{-1} C_k Q_k A_k^{\mathrm{T}} \right] \tau_{k+1\perp}^{\mathrm{T}}, \end{aligned} \tag{3.2.40}$$

$$\hat{Q}_{k+1} = \tau_{k+1} \left[A_k \hat{Q}_k A_k^{\mathrm{T}} + A_k Q_k C_k^{\mathrm{T}} \tilde{V}_{2,k}^{-1} C_k Q_k A_k^{\mathrm{T}} \right] \tau_{k+1}^{\mathrm{T}}, \tag{3.2.41}$$

$$\tau_{k+1} = M_k L_k^{\mathrm{T}} \left(L_k M_k L_k \right)^{-1} L_k. \tag{3.2.42}$$

Proof. It follows from (3.2.27) and (3.2.31) that

$$L_k \hat{Q}_{k+1} L_k^{\mathrm{T}} = L_k \left[A_k \hat{Q}_k A_k^{\mathrm{T}} + A_k Q_k C_k^{\mathrm{T}} \tilde{V}_{2,k}^{-1} C_k Q_k A_k^{\mathrm{T}} \right] L_k^{\mathrm{T}}. \tag{3.2.43}$$

Pre-multiplying and post-multiplying (3.2.43) by G_{k+1}^{T} and G_{k+1}, respectively, yields

$$\tau_{k+1} \hat{Q}_{k+1} \tau_{k+1}^{\mathrm{T}} = \tau_{k+1} \left[A_k \hat{Q}_k A_k^{\mathrm{T}} + A_k Q_k C_k^{\mathrm{T}} \tilde{V}_{2,k}^{-1} C_k Q_k A_k^{\mathrm{T}} \right] \tau_{k+1}^{\mathrm{T}}. \tag{3.2.44}$$

Hence, (3.2.41) follows from Proposition 3.2.5.

Since $\tilde{Q}_{12,k+1} = \hat{Q}_{k+1} L_k$, (3.2.30) and (3.2.33) imply that

$$\tau_{k+1} \hat{Q}_{k+1} = \tau_{k+1} \left[A_k \hat{Q}_k A_k^{\mathrm{T}} + A_k Q_k C_k^{\mathrm{T}} \tilde{V}_{2,k}^{-1} C_k Q_k A_k^{\mathrm{T}} \right]. \tag{3.2.45}$$

Therefore, (3.2.41) imply that

$$\tau_{k+1} \left[A_k \hat{Q}_k A_k^{\mathrm{T}} + A_k Q_k C_k^{\mathrm{T}} \tilde{V}_{2,k}^{-1} C_k Q_k A_k^{\mathrm{T}} \right] = \tau_{k+1} \left[A_k \hat{Q}_k A_k^{\mathrm{T}} + A_k Q_k C_k^{\mathrm{T}} \tilde{V}_{2,k}^{-1} C_k Q_k A_k^{\mathrm{T}} \right] \tau_{k+1}^{\mathrm{T}}. \tag{3.2.46}$$

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

$$\begin{aligned} \hat{Q}_{k+1} &= A_k \hat{Q}_k A_k^{\mathrm{T}} + A_k Q_k C_k^{\mathrm{T}} \tilde{V}_{2,k}^{-1} C_k Q_k A_k^{\mathrm{T}} \\ &\quad - \tau_{k+1\perp} \left[A_k \hat{Q}_k A_k^{\mathrm{T}} + A_k Q_k C_k^{\mathrm{T}} \tilde{V}_{2,k}^{-1} C_k Q_k A_k^{\mathrm{T}} \right] \tau_{k+1\perp}^{\mathrm{T}}. \end{aligned} \tag{3.2.47}$$

It follows from (3.2.15) and (3.2.21) that

$$Q_{k+1} = A Q_k A^{\mathrm{T}} + V_{1,k} + A \hat{Q}_k A^{\mathrm{T}} - \hat{Q}_{k+1}. \tag{3.2.48}$$

Therefore, substituting (3.2.47) into (3.2.48) yields (3.2.40). □

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 3.2.3 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 (3.2.40) and (3.2.41), respectively.

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

State update:

$$G_k = (L_k\hat{Q}_kL_k)^{-1}L_k\hat{Q}_k, \tag{3.2.49}$$

$$x_{\mathrm{e},k+1} = L_kA_k\left(I - Q_kC_k^{\mathrm{T}}\tilde{V}_{2,k}^{-1}C_k\right)G_k^{\mathrm{T}}x_{\mathrm{e},k} + L_kA_kQ_kC_k^{\mathrm{T}}\tilde{V}_{2,k}^{-1}y_k. \tag{3.2.50}$$

Covariance update:

$$\begin{aligned}
M_k &= A_k\left(\hat{Q}_k + Q_kC_k^{\mathrm{T}}\tilde{V}_{2,k}^{-1}C_kQ_k\right)A_k^{\mathrm{T}}, && (3.2.51)\\
\tau_{k+1} &= M_kL_k^{\mathrm{T}}\left(L_kM_kL_k\right)^{-1}L_k, && (3.2.52)\\
\hat{Q}_{k+1} &= \tau_{k+1}\left[A_k\hat{Q}_kA_k^{\mathrm{T}} + A_kQ_kC_k^{\mathrm{T}}\tilde{V}_{2,k}^{-1}C_kQ_kA_k^{\mathrm{T}}\right]\tau_{k+1}^{\mathrm{T}}, && (3.2.53)\\
Q_{k+1} &= A_kQ_kA_k^{\mathrm{T}} + V_{1,k} - A_kQ_kC_k^{\mathrm{T}}\tilde{V}_{2,k}^{-1}C_kQ_kA_k^{\mathrm{T}} && (3.2.54)\\
&\quad +\tau_{k+1\perp}\left[A_k\hat{Q}_kA_k^{\mathrm{T}} + A_kQ_kC_k^{\mathrm{T}}\tilde{V}_{2,k}^{-1}C_kQ_kA_k^{\mathrm{T}}\right]\tau_{k+1\perp}^{\mathrm{T}}.
\end{aligned}$$

3.3 Two-Step Estimator

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

$$x_{\mathrm{e},k}^{\mathrm{da}} = C_{\mathrm{e},k}^{\mathrm{f}}x_{\mathrm{e},k}^{\mathrm{f}} + D_{\mathrm{e},k}^{\mathrm{f}}y_k, \tag{3.3.1}$$

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

$$x_{\mathrm{e},k+1}^{\mathrm{f}} = A_{\mathrm{e},k}^{\mathrm{da}}x_{\mathrm{e},k}^{\mathrm{da}}. \tag{3.3.2}$$

First, we 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

$$\tilde{Q}_k^{\mathrm{f}} \triangleq \mathcal{E}\left[\tilde{x}_k^{\mathrm{f}}(\tilde{x}_k^{\mathrm{f}})^{\mathrm{T}}\right], \quad \tilde{Q}_k^{\mathrm{da}} \triangleq \mathcal{E}\left[\tilde{x}_k^{\mathrm{da}}(\tilde{x}_k^{\mathrm{da}})^{\mathrm{T}}\right], \tag{3.3.3}$$

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

$$\tilde{x}_k^{\mathrm{f}} \triangleq \begin{bmatrix} x_k \\ x_{\mathrm{e},k}^{\mathrm{f}} \end{bmatrix}, \quad \tilde{x}_k^{\mathrm{da}} \triangleq \begin{bmatrix} x_k \\ x_{\mathrm{e},k}^{\mathrm{da}} \end{bmatrix}. \tag{3.3.4}$$

Define the *data assimilation cost* by

$$J_k^{\mathrm{da}} \triangleq \mathcal{E}\left[\left(L_k x_k - x_{\mathrm{e},k}^{\mathrm{da}}\right)^{\mathrm{T}}\left(L_k x_k - x_{\mathrm{e},k}^{\mathrm{da}}\right)\right]. \tag{3.3.5}$$

Hence, (3.3.3) implies that

$$J_k^{\mathrm{da}} = \mathrm{tr}(\tilde{Q}_k^{\mathrm{da}} \tilde{R}_k), \tag{3.3.6}$$

where $\tilde{R}_k$ is defined by

$$\tilde{R}_k \triangleq \begin{bmatrix} L_k^{\mathrm{T}} L_k & -L_k^{\mathrm{T}} \\ -L_k & I \end{bmatrix}. \tag{3.3.7}$$

It follows from (3.2.1), (3.3.1), and (3.3.4) that

$$\tilde{x}_k^{\mathrm{da}} = \tilde{A}_k^{\mathrm{f}} \tilde{x}_k^{\mathrm{f}} + \tilde{D}_{1,k}^{\mathrm{f}} w_k, \tag{3.3.8}$$

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

$$\tilde{A}_k^{\mathrm{f}} \triangleq \begin{bmatrix} I & 0 \\ D_{\mathrm{e},k}^{\mathrm{f}} C_k & C_{\mathrm{e},k}^{\mathrm{f}} \end{bmatrix}, \quad \tilde{D}_{1,k}^{\mathrm{f}} \triangleq \begin{bmatrix} 0 \\ D_{\mathrm{e},k}^{\mathrm{f}} D_{2,k} \end{bmatrix}. \tag{3.3.9}$$

Therefore,

$$\tilde{Q}_k^{\mathrm{da}} = \tilde{A}_k^{\mathrm{f}} \tilde{Q}_k^{\mathrm{f}} (\tilde{A}_k^{\mathrm{f}})^{\mathrm{T}} + \tilde{D}_{1,k}^{\mathrm{f}} (\tilde{D}_{1,k}^{\mathrm{f}})^{\mathrm{T}}. \tag{3.3.10}$$

Hence, J_k^{da} can be expressed as

$$J_k^{\mathrm{da}} = \mathrm{tr}\left[\left(\tilde{A}_k^{\mathrm{f}}\tilde{Q}_k^{\mathrm{f}}(\tilde{A}_k^{\mathrm{f}})^{\mathrm{T}} + \tilde{D}_{1,k}^{\mathrm{f}}(\tilde{D}_{1,k}^{\mathrm{f}})^{\mathrm{T}}\right)\tilde{R}_k\right]. \tag{3.3.11}$$

Partition $\tilde{Q}_k^{\mathrm{f}}$ as

$$\tilde{Q}_k^{\mathrm{f}} = \begin{bmatrix} \tilde{Q}_{1,k}^{\mathrm{f}} & \tilde{Q}_{12,k}^{\mathrm{f}} \\ (\tilde{Q}_{12,k}^{\mathrm{f}})^{\mathrm{T}} & \tilde{Q}_{2,k}^{\mathrm{f}} \end{bmatrix} \tag{3.3.12}$$

so that substituting (3.3.9) into (3.3.11) yields

$$\begin{aligned} J_k^{\mathrm{da}} = {} & \mathrm{tr}\left[L_k\tilde{Q}_{1,k}^{\mathrm{f}}L_k^{\mathrm{T}}\right] - 2\mathrm{tr}\left[D_{\mathrm{e},k}^{\mathrm{f}}C_k\tilde{Q}_{1,k}^{\mathrm{f}}L_k^{\mathrm{T}}\right] - 2\mathrm{tr}\left[L_k\tilde{Q}_{12,k}^{\mathrm{f}}(C_{\mathrm{e},k}^{\mathrm{f}})^{\mathrm{T}}\right] \\ & + \mathrm{tr}\left[C_{\mathrm{e},k}^{\mathrm{f}}\tilde{Q}_{2,k}^{\mathrm{f}}(C_{\mathrm{e},k}^{\mathrm{f}})^{\mathrm{T}}\right] + 2\mathrm{tr}\left[D_{\mathrm{e},k}^{\mathrm{f}}C_k\tilde{Q}_{12,k}^{\mathrm{f}}(C_{\mathrm{e},k}^{\mathrm{f}})^{\mathrm{T}}\right] \\ & + \mathrm{tr}\left[D_{\mathrm{e},k}^{\mathrm{f}}\left(C_k\tilde{Q}_{1,k}^{\mathrm{f}}C_k^{\mathrm{T}} + V_{2,k}\right)(D_{\mathrm{e},k}^{\mathrm{f}})^{\mathrm{T}}\right]. \end{aligned} \tag{3.3.13}$$

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

Proposition 3.3.1 *Assume that $C_{\mathrm{e},k}^{\mathrm{f}}$ and $D_{\mathrm{e},k}^{\mathrm{f}}$ minimize J_k^{da}. Then, $C_{\mathrm{e},k}^{\mathrm{f}}$ and $D_{\mathrm{e},k}^{\mathrm{f}}$ satisfy*

$$C_{\mathrm{e},k}^{\mathrm{f}}\tilde{Q}_{2,k}^{\mathrm{f}} = \left(L_k - D_{\mathrm{e},k}^{\mathrm{f}}C_k\right)\tilde{Q}_{12,k}^{\mathrm{f}}, \tag{3.3.14}$$

$$D_{\mathrm{e},k}^{\mathrm{f}} = \left(L\tilde{Q}_{1,k}^{\mathrm{f}} - C_{\mathrm{e},k}^{\mathrm{f}}(\tilde{Q}_{12,k}^{\mathrm{f}})^{\mathrm{T}}\right)C_k^{\mathrm{T}}\left(C_k\tilde{Q}_{1,k}^{\mathrm{f}}C_k^{\mathrm{T}} + V_{2,k}\right)^{-1}. \tag{3.3.15}$$

Proof. Setting $\frac{\partial J_k^{\mathrm{da}}}{\partial C_{\mathrm{e},k}^{\mathrm{f}}} = 0$ and $\frac{\partial J_k^{\mathrm{da}}}{\partial D_{\mathrm{e},k}^{\mathrm{f}}} = 0$ yields the result. □

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

$$\begin{aligned} Q_k^{\mathrm{f}} &\triangleq \tilde{Q}_{1,k}^{\mathrm{f}} - \tilde{Q}_{12,k}^{\mathrm{f}}(\tilde{Q}_{2,k}^{\mathrm{f}})^{-1}(\tilde{Q}_{12,k}^{\mathrm{f}})^{\mathrm{T}}, \\ \hat{Q}_k^{\mathrm{f}} &\triangleq \tilde{Q}_{12,k}^{\mathrm{f}}(\tilde{Q}_{2,k}^{\mathrm{f}})^{-1}(\tilde{Q}_{12,k}^{\mathrm{f}})^{\mathrm{T}}. \end{aligned} \tag{3.3.16}$$

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

$$V_{2,k}^{\mathrm{f}} \triangleq C_k Q_k^{\mathrm{f}} C_k^{\mathrm{T}} + V_{2,k}. \tag{3.3.17}$$

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

$$G_k^{\mathrm{f}} \triangleq (\tilde{Q}_{2,k}^{\mathrm{f}})^{-1}(\tilde{Q}_{12,k}^{\mathrm{f}})^{\mathrm{T}}. \tag{3.3.18}$$

Proposition 3.3.2 *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,*

$$C^{\mathrm{f}}_{\mathrm{e},k} = L_k \left(I - Q^{\mathrm{f}}_k C^{\mathrm{T}}_k (V^{\mathrm{f}}_{2,k})^{-1} C_k\right) (G^{\mathrm{f}}_k)^{\mathrm{T}}, \tag{3.3.19}$$

$$D^{\mathrm{f}}_{\mathrm{e},k} = L_k Q^{\mathrm{f}}_k C^{\mathrm{T}}_k (V^{\mathrm{f}}_{2,k})^{-1}. \tag{3.3.20}$$

Proof. It follows from (3.3.14) that

$$C^{\mathrm{f}}_{\mathrm{e},k} = \left(L_k - D^{\mathrm{f}}_{\mathrm{e},k} C_k\right) (G^{\mathrm{f}}_k)^{\mathrm{T}}. \tag{3.3.21}$$

Substituting (3.3.21) into (3.3.15) yields

$$\begin{aligned} D^{\mathrm{f}}_{\mathrm{e},k} = [&L_k \tilde{Q}^{\mathrm{f}}_{1,k} - L_k \tilde{Q}^{\mathrm{f}}_{12,k} (\tilde{Q}^{\mathrm{f}}_{2,k})^{-1} (\tilde{Q}^{\mathrm{f}}_{12,k})^{\mathrm{T}} C^{\mathrm{T}}_k \\ &+ D^{\mathrm{f}}_{\mathrm{e},k} C_k \tilde{Q}^{\mathrm{f}}_{12,k} (\tilde{Q}^{\mathrm{f}}_{2,k})^{-1} (\tilde{Q}^{\mathrm{f}}_{12,k})^{\mathrm{T}} C^{\mathrm{T}}_k] \left(C_k \tilde{Q}^{\mathrm{f}}_{1,k} C^{\mathrm{T}}_k + V_{2,k}\right)^{-1}. \end{aligned} \tag{3.3.22}$$

Therefore, (3.3.20) follows from (3.3.16) and (3.3.17). Finally, substituting (3.3.20) into (3.3.21) yields (3.3.19). □

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

$$\tilde{Q}^{\mathrm{da}}_k = \begin{bmatrix} \tilde{Q}^{\mathrm{da}}_{1,k} & \tilde{Q}^{\mathrm{da}}_{12,k} \\ (\tilde{Q}^{\mathrm{da}}_{12,k})^{\mathrm{T}} & \tilde{Q}^{\mathrm{da}}_{2,k} \end{bmatrix}. \tag{3.3.23}$$

Proposition 3.3.3 *Assume that $x^{\mathrm{da}}_{\mathrm{e},k}$ is given by* (3.3.1)*, and $C^{\mathrm{f}}_{\mathrm{e},k}$ and $D^{\mathrm{f}}_{\mathrm{e},k}$ satisfy* (3.3.19)*,* (3.3.20)*. Then,*

$$\tilde{Q}^{\mathrm{da}}_{1,k} = \tilde{Q}^{\mathrm{f}}_{1,k}, \tag{3.3.24}$$

$$\tilde{Q}^{\mathrm{da}}_{12,k} = \left(\hat{Q}^{\mathrm{f}}_k + Q^{\mathrm{f}}_k C^{\mathrm{T}}_k (V^{\mathrm{f}}_{2,k})^{-1} C_k Q^{\mathrm{f}}_k\right) L^{\mathrm{T}}_k, \tag{3.3.25}$$

$$\tilde{Q}^{\mathrm{da}}_{2,k} = L_k \left(\hat{Q}^{\mathrm{f}}_k + Q^{\mathrm{f}}_k C^{\mathrm{T}}_k (V^{\mathrm{f}}_{2,k})^{-1} C_k Q^{\mathrm{f}}_k\right) L^{\mathrm{T}}_k. \tag{3.3.26}$$

Proof. It follows from (3.3.10) that $\tilde{Q}^{\mathrm{da}}_{1,k} = \tilde{Q}^{\mathrm{f}}_{1,k}$ and

$$\tilde{Q}^{\mathrm{da}}_{12,k} = \tilde{Q}^{\mathrm{f}}_{12,k} (C^{\mathrm{f}}_{\mathrm{e},k})^{\mathrm{T}} + \tilde{Q}^{\mathrm{f}}_{1,k} C^{\mathrm{T}}_k (D^{\mathrm{f}}_{\mathrm{e},k})^{\mathrm{T}}. \tag{3.3.27}$$

Substituting (3.3.19) and (3.3.20) into (3.3.27) yields (3.3.25). Similarly, it follows from (3.3.10) and (3.3.23) that

$$\begin{aligned}\tilde{Q}_{2,k}^{\mathrm{da}} = C_{\mathrm{e},k}^{\mathrm{f}}\tilde{Q}_{1,k}^{\mathrm{f}}(C_{\mathrm{e},k}^{\mathrm{f}})^{\mathrm{T}} + C_{\mathrm{e},k}^{\mathrm{f}}(\tilde{Q}_{12,k}^{\mathrm{f}})^{\mathrm{T}}C_k^{\mathrm{T}}(D_{\mathrm{e},k}^{\mathrm{f}})^{\mathrm{T}} \\ + D_{\mathrm{e},k}^{\mathrm{f}}C_k\tilde{Q}_{12,k}^{\mathrm{f}}(C_{\mathrm{e},k}^{\mathrm{f}})^{\mathrm{T}} + D_{\mathrm{e},k}^{\mathrm{f}}\left(C_k\tilde{Q}_{1,k}^{\mathrm{f}}C_k^{\mathrm{T}} + V_{2,k}\right)(D_{\mathrm{e},k}^{\mathrm{f}})^{\mathrm{T}}.\end{aligned} \tag{3.3.28}$$

Finally, substituting (3.3.19) and (3.3.20) into (3.3.28) yields (3.3.26). □

Next, define Q_k^{da} and $\hat{Q}_k^{\mathrm{da}}$ by

$$\begin{aligned}Q_k^{\mathrm{da}} &\triangleq \tilde{Q}_{1,k}^{\mathrm{da}} - \tilde{Q}_{12,k}^{\mathrm{da}}(\tilde{Q}_{2,k}^{\mathrm{da}})^{-1}(\tilde{Q}_{12,k}^{\mathrm{da}})^{\mathrm{T}}, \\ \hat{Q}_k^{\mathrm{da}} &\triangleq \tilde{Q}_{12,k}^{\mathrm{da}}(\tilde{Q}_{2,k}^{\mathrm{da}})^{-1}(\tilde{Q}_{12,k}^{\mathrm{da}})^{\mathrm{T}}.\end{aligned} \tag{3.3.29}$$

Corollary 3.3.1 *Assume that $C_{\mathrm{e},k}^{\mathrm{f}}$ and $D_{\mathrm{e},k}^{\mathrm{f}}$ satisfy Proposition 3.3.2. Then,*

$$L_k\tilde{Q}_{12,k}^{\mathrm{da}} = \tilde{Q}_{2,k}^{\mathrm{da}}, \quad \tilde{Q}_{12,k}^{\mathrm{da}} = \hat{Q}_k^{\mathrm{da}}L_k^{\mathrm{T}}, \quad \tilde{Q}_{2,k}^{\mathrm{da}} = L_k\hat{Q}_k^{\mathrm{da}}L_k^{\mathrm{T}}. \tag{3.3.30}$$

Next, define G_k^{da} by

$$G_k^{\mathrm{da}} \triangleq (\tilde{Q}_{2,k}^{\mathrm{da}})^{-1}(\tilde{Q}_{12,k}^{\mathrm{da}})^{\mathrm{T}}. \tag{3.3.31}$$

Also, define M_k^{da} by

$$M_k^{\mathrm{da}} \triangleq \hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}}C_k^{\mathrm{T}}(V_{2,k}^{\mathrm{f}})^{-1}C_kQ_k^{\mathrm{f}} \tag{3.3.32}$$

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

$$\tau_k^{\mathrm{da}} \triangleq (G_k^{\mathrm{da}})^{\mathrm{T}}L_k, \quad \tau_{k\perp}^{\mathrm{da}} \triangleq I - \tau_k^{\mathrm{da}}. \tag{3.3.33}$$

Proposition 3.3.4 *Assume that $C_{\mathrm{e},k}^{\mathrm{f}}$ and $D_{\mathrm{e},k}^{\mathrm{f}}$ satisfy Proposition 3.3.2. Then, τ_k^{da} is an oblique projector.*

Proof. It follows from (3.3.25) and (3.3.26) that

$$\tilde{Q}_{12,k}^{\mathrm{da}} = M_k^{\mathrm{da}}L_k^{\mathrm{T}}, \quad \tilde{Q}_{2,k}^{\mathrm{da}} = L_kM_k^{\mathrm{da}}L_k^{\mathrm{T}}. \tag{3.3.34}$$

Substituting (3.3.34) into (3.3.31) yields

$$G_k^{\mathrm{da}} = \left(L_k M_k^{\mathrm{da}} L_k^{\mathrm{T}}\right)^{-1} L_k M_k^{\mathrm{da}}. \tag{3.3.35}$$

Therefore, it follows from (3.3.33) that

$$\tau_k^{\mathrm{da}} = M_k^{\mathrm{da}} L_k^{\mathrm{T}} \left(L_k M_k^{\mathrm{da}} L_k^{\mathrm{T}}\right)^{-1} L_k. \tag{3.3.36}$$

Hence, $(\tau_k^{\mathrm{da}})^2 = \tau_k^{\mathrm{da}}$. □

Proposition 3.3.5 *Assume that $C_{\mathrm{e},k}^{\mathrm{f}}$ and $D_{\mathrm{e},k}^{\mathrm{f}}$ satisfy Proposition 3.3.2. Then,*

$$\tau_k^{\mathrm{da}} \hat{Q}_k^{\mathrm{da}} = \hat{Q}_k^{\mathrm{da}}. \tag{3.3.37}$$

Proof. It follows from (3.3.29) and (3.3.34) that

$$\hat{Q}_k^{\mathrm{da}} = M_k^{\mathrm{da}} L_k^{\mathrm{T}} \left(L_k M_k^{\mathrm{da}} L_k^{\mathrm{T}}\right)^{-1} L_k M_k^{\mathrm{da}}. \tag{3.3.38}$$

Hence, (3.3.37) follows from (3.3.36). □

Proposition 3.3.6 *Assume that $x_{\mathrm{e},k}^{\mathrm{da}}$ is given by* (3.3.1)*, and $C_{\mathrm{e},k}^{\mathrm{f}}$ and $D_{\mathrm{e},k}^{\mathrm{f}}$ satisfy Proposition 3.3.2. Then,*

$$\hat{Q}_k^{\mathrm{da}} = \tau_k^{\mathrm{da}} \left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}}\right) (\tau_k^{\mathrm{da}})^{\mathrm{T}}, \tag{3.3.39}$$

$$Q_k^{\mathrm{da}} = Q_k^{\mathrm{f}} - Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} + \tau_{k\perp}^{\mathrm{da}} \left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}}\right) (\tau_{k\perp}^{\mathrm{da}})^{\mathrm{T}}. \tag{3.3.40}$$

Proof. It follows from (3.3.26) and (3.3.30) that

$$L_k \hat{Q}_k^{\mathrm{da}} L_k^{\mathrm{T}} = L_k \left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}}\right) L_k^{\mathrm{T}}. \tag{3.3.41}$$

Pre-multiplying and post-multiplying (3.3.41) by $(G_k^{\mathrm{da}})^{\mathrm{T}}$ and G_k^{da}, respectively, yields (3.3.39).

Next, it follows from (3.3.25), (3.3.30), and (3.3.33) that

$$\tau_k^{\mathrm{da}}\hat{Q}_k^{\mathrm{da}} = \tau_k^{\mathrm{da}}\left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}}C_k^{\mathrm{T}}(V_{2,k}^{\mathrm{f}})^{-1}C_kQ_k^{\mathrm{f}}\right). \tag{3.3.42}$$

Therefore, Proposition 3.3.4 and (3.3.39) imply that

$$\tau_k^{\mathrm{da}}\left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}}C_k^{\mathrm{T}}(V_{2,k}^{\mathrm{f}})^{-1}C_kQ_k^{\mathrm{f}}\right) = \tau_k^{\mathrm{da}}\left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}}C_k^{\mathrm{T}}(V_{2,k}^{\mathrm{f}})^{-1}C_kQ_k^{\mathrm{f}}\right)(\tau_k^{\mathrm{da}})^{\mathrm{T}}. \tag{3.3.43}$$

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

$$\hat{Q}_k^{\mathrm{da}} = \hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}}C_k^{\mathrm{T}}(V_{2,k}^{\mathrm{f}})^{-1}C_kQ_k^{\mathrm{f}} - \tau_{k\perp}^{\mathrm{da}}\left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}}C_k^{\mathrm{T}}(V_{2,k}^{\mathrm{f}})^{-1}C_kQ_k^{\mathrm{f}}\right)(\tau_{k\perp}^{\mathrm{da}})^{\mathrm{T}}. \tag{3.3.44}$$

Finally, note that (3.3.24) implies that

$$Q_k^{\mathrm{da}} = \tilde{Q}_{1,k}^{\mathrm{f}} - \hat{Q}_k^{\mathrm{da}}. \tag{3.3.45}$$

Substituting (3.3.44) into (3.3.45) yields (3.3.40). □

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

$$J_k^{\mathrm{f}} \triangleq \mathcal{E}\left[\left(L_kx_{k+1} - x_{\mathrm{e},k+1}^{\mathrm{f}}\right)\left(L_kx_{k+1} - x_{\mathrm{e},k+1}^{\mathrm{f}}\right)^{\mathrm{T}}\right]. \tag{3.3.46}$$

Hence, it follows from (3.3.3) that

$$J_k^{\mathrm{f}} = \mathrm{tr}\left(\tilde{Q}_{k+1}^{\mathrm{f}}\tilde{R}_k\right). \tag{3.3.47}$$

It follows from (3.2.1) and (3.3.2) that

$$\tilde{x}_{k+1}^{\mathrm{f}} = \tilde{A}_k^{\mathrm{da}}\tilde{x}_k^{\mathrm{da}} + \tilde{D}_{1,k}^{\mathrm{da}}w_k, \tag{3.3.48}$$

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

$$\tilde{A}_k^{\mathrm{da}} \triangleq \begin{bmatrix} A_k & 0 \\ 0 & A_{\mathrm{e},k}^{\mathrm{da}} \end{bmatrix}, \quad \tilde{D}_{1,k}^{\mathrm{da}} \triangleq \begin{bmatrix} D_{1,k} \\ 0 \end{bmatrix}. \tag{3.3.49}$$

Therefore,

$$\tilde{Q}^{\mathrm{f}}_{k+1} = \tilde{A}^{\mathrm{da}}_k \tilde{Q}^{\mathrm{da}}_k (\tilde{A}^{\mathrm{da}}_k)^{\mathrm{T}} + \tilde{D}^{\mathrm{da}}_{1,k} (\tilde{D}^{\mathrm{da}}_{1,k})^{\mathrm{T}}. \tag{3.3.50}$$

Substituting (3.3.50) into (3.3.47) and using (3.3.49) yields

$$\begin{aligned} J^{\mathrm{f}}_k = \mathrm{tr} \left[L_k \left(A_k \tilde{Q}^{\mathrm{da}}_{1,k} A^{\mathrm{T}}_k + V_{1,k} \right) L^{\mathrm{T}}_k \right] - \mathrm{tr} \left[L_k A_k \tilde{Q}^{\mathrm{da}}_{12,k} (A^{\mathrm{da}}_{\mathrm{e},k})^{\mathrm{T}} \right] \\ - \mathrm{tr} \left[A^{\mathrm{da}}_{\mathrm{e},k} (\tilde{Q}^{\mathrm{da}}_{12,k})^{\mathrm{T}} A^{\mathrm{T}}_k L^{\mathrm{T}}_k \right] + \mathrm{tr} \left[A^{\mathrm{da}}_{\mathrm{e},k} \tilde{Q}^{\mathrm{da}}_{2,k} (A^{\mathrm{da}}_{\mathrm{e},k})^{\mathrm{T}} \right]. \end{aligned} \tag{3.3.51}$$

Proposition 3.3.7 *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*

$$A^{\mathrm{da}}_{\mathrm{e},k} = L_k A_k (G^{\mathrm{da}}_k)^{\mathrm{T}}. \tag{3.3.52}$$

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

Assume that $A^{\mathrm{da}}_{\mathrm{e},k}$ is given by (3.3.52). Then the following result concerns relationships among the covariances $\tilde{Q}^{\mathrm{f}}_{12,k+1}$, $\tilde{Q}^{\mathrm{f}}_{2,k+1}$, and $\hat{Q}^{\mathrm{f}}_{k+1}$.

Proposition 3.3.8 *Assume that $A^{\mathrm{da}}_{\mathrm{e},k}$ satisfies* (3.3.52)*. Then,*

$$L_k \tilde{Q}^{\mathrm{f}}_{12,k+1} = \tilde{Q}^{\mathrm{f}}_{2,k+1}, \;\; \tilde{Q}^{\mathrm{f}}_{12,k+1} = \hat{Q}^{\mathrm{f}}_{k+1} L^{\mathrm{T}}_k, \;\; \tilde{Q}^{\mathrm{f}}_{2,k+1} = L_k \hat{Q}^{\mathrm{f}}_{k+1} L^{\mathrm{T}}_k. \tag{3.3.53}$$

Proof. It follows from (3.3.49) and (3.3.50) that

$$\tilde{Q}^{\mathrm{f}}_{12,k+1} = A_k \tilde{Q}^{\mathrm{da}}_{12,k} (A^{\mathrm{da}}_{\mathrm{e},k})^{\mathrm{T}}. \tag{3.3.54}$$

Substituting (3.3.52) into (3.3.54) yields

$$\tilde{Q}^{\mathrm{f}}_{12,k+1} = A_k \hat{Q}^{\mathrm{da}}_k A^{\mathrm{T}}_k L^{\mathrm{T}}_k. \tag{3.3.55}$$

Similarly, (3.3.49) and (3.3.50) imply that

$$\tilde{Q}^{\mathrm{f}}_{2,k+1} = A^{\mathrm{da}}_{\mathrm{e},k} \tilde{Q}^{\mathrm{da}}_{2,k} (A^{\mathrm{da}}_{\mathrm{e},k})^{\mathrm{T}}. \tag{3.3.56}$$

Substituting (3.3.52) into (3.3.56) yields

$$\tilde{Q}^{\mathrm{f}}_{2,k+1} = L_k A_k \hat{Q}^{\mathrm{da}}_k A_k^{\mathrm{T}} L_k^{\mathrm{T}}. \tag{3.3.57}$$

Therefore, (3.3.55) and (3.3.57) imply that $L\tilde{Q}^{\mathrm{f}}_{12,k+1} = \tilde{Q}^{\mathrm{f}}_{2,k+1}$.

Assuming $\tilde{Q}^{\mathrm{f}}_{2,k+1}$ is invertible, $L_k \tilde{Q}^{\mathrm{f}}_{12,k+1} (\tilde{Q}^{\mathrm{f}}_{2,k+1})^{-1} = I$. Therefore, it follows from (3.3.16) that $\tilde{Q}^{\mathrm{f}}_{12,k+1} = \hat{Q}^{\mathrm{f}}_{k+1} L_k^{\mathrm{T}}$ and $\tilde{Q}^{\mathrm{f}}_{2,k+1} = L_k \hat{Q}^{\mathrm{f}}_{k+1} L_k^{\mathrm{T}}$. □

Next, define M^{f}_k by

$$M^{\mathrm{f}}_k \triangleq A_k \hat{Q}^{\mathrm{da}}_k A_k^{\mathrm{T}}. \tag{3.3.58}$$

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

$$\tau^{\mathrm{f}}_k \triangleq (G^{\mathrm{f}}_k)^{\mathrm{T}} L_{k-1}, \quad \tau^{\mathrm{f}}_{k\perp} \triangleq I - \tau^{\mathrm{f}}_k. \tag{3.3.59}$$

Proposition 3.3.9 *Assume that $A^{\mathrm{da}}_{\mathrm{e},k}$ satisfies* (3.3.52)*. Then, τ^{f}_{k+1} is an oblique projector, that is, $(\tau^{\mathrm{f}}_{k+1})^2 = \tau^{\mathrm{f}}_{k+1}$.*

Proof. It follows from (3.3.55), (3.3.57), and (3.3.58) that

$$\tilde{Q}^{\mathrm{f}}_{12,k+1} = M^{\mathrm{f}}_k L_k^{\mathrm{T}}, \quad \tilde{Q}^{\mathrm{f}}_{2,k} = L_k M^{\mathrm{f}}_k L_k^{\mathrm{T}}. \tag{3.3.60}$$

Substituting (3.3.60) into (3.2.23) yields

$$\tau^{\mathrm{f}}_{k+1} = M^{\mathrm{f}}_k L_k^{\mathrm{T}} \left(L_k M^{\mathrm{f}}_k L_k^{\mathrm{T}} \right)^{-1} L_k. \tag{3.3.61}$$

Therefore, $(\tau^{\mathrm{f}}_{k+1})^2 = \tau^{\mathrm{f}}_{k+1}$. □

Proposition 3.3.10 *Assume that $A^{\mathrm{da}}_{\mathrm{e},k}$ satisfies* (3.3.52)*. Then,*

$$\tau^{\mathrm{f}}_{k+1} \hat{Q}^{\mathrm{f}}_{k+1} = \hat{Q}^{\mathrm{f}}_{k+1}. \tag{3.3.62}$$

Proof. Note that (3.3.16) and (3.3.60) imply that

$$\hat{Q}^{\mathrm{f}}_{k+1} = M^{\mathrm{f}}_k L^{\mathrm{T}}_k \left(L_k M^{\mathrm{f}}_k L^{\mathrm{T}}_k\right)^{-1} L_k M^{\mathrm{f}}_k. \tag{3.3.63}$$

Hence, (3.3.62) follows from (3.3.61) and (3.3.63). □

Proposition 3.3.11 *Assume that* $A^{\mathrm{da}}_{\mathrm{e},k}$ *satisfies* (3.3.52)*. Then,*

$$\hat{Q}^{\mathrm{f}}_{k+1} = \tau^{\mathrm{f}}_{k+1} A_k \hat{Q}^{\mathrm{da}}_k A^{\mathrm{T}}_k (\tau^{\mathrm{f}}_{k+1})^{\mathrm{T}}, \tag{3.3.64}$$

$$Q^{\mathrm{f}}_{k+1} = A_k Q^{\mathrm{da}}_k A^{\mathrm{T}}_k + V_{1,k} + \tau^{\mathrm{f}}_{k+1\perp} \left(A_k \hat{Q}^{\mathrm{da}}_k A^{\mathrm{T}}_k\right) (\tau^{\mathrm{f}}_{k+1\perp})^{\mathrm{T}}. \tag{3.3.65}$$

Proof. It follows from (3.3.53) and (3.3.57) that

$$L_k \hat{Q}^{\mathrm{f}}_{k+1} L^{\mathrm{T}}_k = L_k A_k \hat{Q}^{\mathrm{da}}_k A^{\mathrm{T}}_k L^{\mathrm{T}}_k. \tag{3.3.66}$$

Pre-multiplying and post-multiplying (3.3.66) by $(G^{\mathrm{f}}_k)^{\mathrm{T}}$ and G^{f}_k, respectively, and using Proposition 3.3.10 yields (3.3.64). Note that (3.3.53) and (3.3.55) imply that

$$L_k \hat{Q}^{\mathrm{f}}_{k+1} = L_k A_k \hat{Q}^{\mathrm{da}}_k A^{\mathrm{T}}_k. \tag{3.3.67}$$

Pre-multiplying (3.3.67) by $(G^{\mathrm{f}}_{k+1})^{\mathrm{T}}$ and using (3.3.62) yields

$$\tau^{\mathrm{f}}_{k+1} A_k \hat{Q}^{\mathrm{da}}_k A^{\mathrm{T}}_k = \tau^{\mathrm{f}}_{k+1} A_k \hat{Q}^{\mathrm{da}}_k A^{\mathrm{T}}_k (\tau^{\mathrm{f}}_{k+1})^{\mathrm{T}}. \tag{3.3.68}$$

Therefore, $\hat{Q}^{\mathrm{f}}_{k+1}$ can be expressed as

$$\hat{Q}^{\mathrm{f}}_{k+1} = A_k \hat{Q}^{\mathrm{da}}_k A^{\mathrm{T}}_k - \tau^{\mathrm{f}}_{k+1\perp} A_k \hat{Q}^{\mathrm{da}}_k A^{\mathrm{T}}_k (\tau^{\mathrm{f}}_{k+1\perp})^{\mathrm{T}}. \tag{3.3.69}$$

It follows from (3.2.1) and (3.3.16) that

$$Q^{\mathrm{f}}_{k+1} = A_k \tilde{Q}^{\mathrm{f}}_{1,k} A^{\mathrm{T}}_k + V_{1,k} - \hat{Q}^{\mathrm{f}}_{k+1}. \tag{3.3.70}$$

Therefore, substituting (3.3.24) into (3.3.70) yields

$$Q^{\mathrm{f}}_{k+1} = A_k \tilde{Q}^{\mathrm{da}}_{1,k} A^{\mathrm{T}}_k + V_{1,k} - \hat{Q}^{\mathrm{f}}_{k+1}. \tag{3.3.71}$$

Finally, substituting (3.3.69) into (3.3.71) yields (3.3.65). □

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

Data assimilation step:

$$x_{\mathrm{e},k}^{\mathrm{da}} = L_k \left(I - Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k\right) (G_k^{\mathrm{f}})^{\mathrm{T}} x_{\mathrm{e},k}^{\mathrm{f}} + L_k Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} y_k, \tag{3.3.72}$$

$$\hat{Q}_k^{\mathrm{da}} = \tau_k^{\mathrm{da}} \left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}}\right) (\tau_k^{\mathrm{da}})^{\mathrm{T}}, \tag{3.3.73}$$

$$Q_k^{\mathrm{da}} = Q_k^{\mathrm{f}} - Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}} + \tau_{k\perp}^{\mathrm{da}} \left(\hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}}\right) (\tau_{k\perp}^{\mathrm{da}})^{\mathrm{T}}, \tag{3.3.74}$$

$$\tau_k^{\mathrm{da}} = M_k^{\mathrm{da}} L_k^{\mathrm{T}} \left(L_k M_k^{\mathrm{da}} L_k^{\mathrm{T}}\right)^{-1} L_k, \tag{3.3.75}$$

$$M_k^{\mathrm{da}} = \hat{Q}_k^{\mathrm{f}} + Q_k^{\mathrm{f}} C_k^{\mathrm{T}} (V_{2,k}^{\mathrm{f}})^{-1} C_k Q_k^{\mathrm{f}}. \tag{3.3.76}$$

Forecast step:

$$x_{\mathrm{e},k+1}^{\mathrm{f}} = L_k A_k (G_k^{\mathrm{da}})^{\mathrm{T}} x_{\mathrm{e},k}^{\mathrm{da}}, \tag{3.3.77}$$

$$\hat{Q}_{k+1}^{\mathrm{f}} = \tau_{k+1}^{\mathrm{f}} A_k \hat{Q}_k^{\mathrm{da}} A_k^{\mathrm{T}} (\tau_{k+1}^{\mathrm{f}})^{\mathrm{T}}, \tag{3.3.78}$$

$$Q_{k+1}^{\mathrm{f}} = A_k Q_k^{\mathrm{da}} A_k^{\mathrm{T}} + V_{1,k} + \tau_{k+1\perp}^{\mathrm{f}} \left(A_k \hat{Q}_k^{\mathrm{da}} A_k^{\mathrm{T}}\right) (\tau_{k+1\perp}^{\mathrm{f}})^{\mathrm{T}}, \tag{3.3.79}$$

$$\tau_{k+1}^{\mathrm{f}} = M_k^{\mathrm{f}} L_k^{\mathrm{T}} \left(L_k M_k^{\mathrm{f}} L_k^{\mathrm{T}}\right)^{-1} L_k, \tag{3.3.80}$$

$$M_k^{\mathrm{f}} = A_k \hat{Q}_k^{\mathrm{da}} A_k^{\mathrm{T}}. \tag{3.3.81}$$

3.4 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 3.1 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 $c_j = 0.05$ Ns/m. We set the initial error covariance $P_0 = 100I$ and assume that $V_{1,k} = I$, $V_{2,k} = I$ for all $k \geqslant 0$.

Let q_i denote the position of the ith mass so that

$$x \triangleq \begin{bmatrix} q_1 & \dot{q}_1 & \cdots & q_{10} & \dot{q}_{10} \end{bmatrix}. \tag{3.4.1}$$

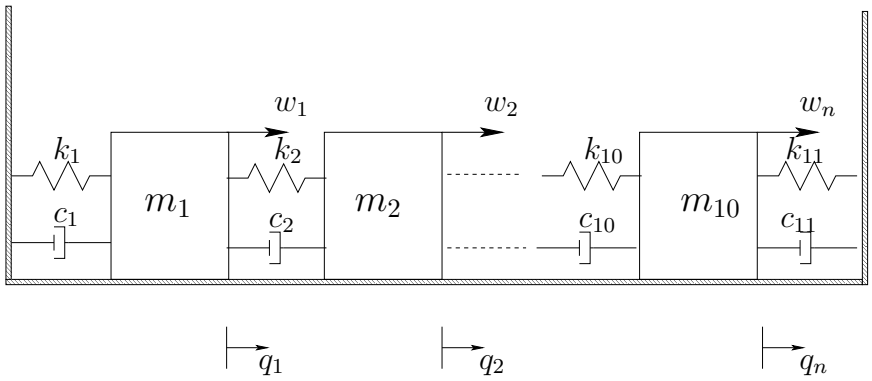


Figure 3.1: Mass-spring-dashpot system

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 $n_{\mathrm{e}} = 8$. For the subspace estimator, we consider a change of basis so that the system has a block upper-triangular structure. Recall that the costs for the estimator is defined by (3.2.6) 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 3.2. As expected, the performance of the reduced-order filter is never better than the full-order Kalman filter (indicated by ratios greater than 1). 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 $n_{\mathrm{e}} = 16$ is shown in Figure 3.2. The objective in both the cases is to obtain estimates of Lx_k, where for $i = 1, \ldots, n_{\mathrm{e}}$, $j = 1, \ldots, n$, the (i, j)th entry of $L \in \mathbb{R}^{n_{\mathrm{e}}\times(n-n_{\mathrm{e}})}$ is given by

$$L_{(i,i)} = \begin{cases} 1, & \text{if } \ i = j, \\ 0.05, & \text{else.} \end{cases} \tag{3.4.2}$$

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

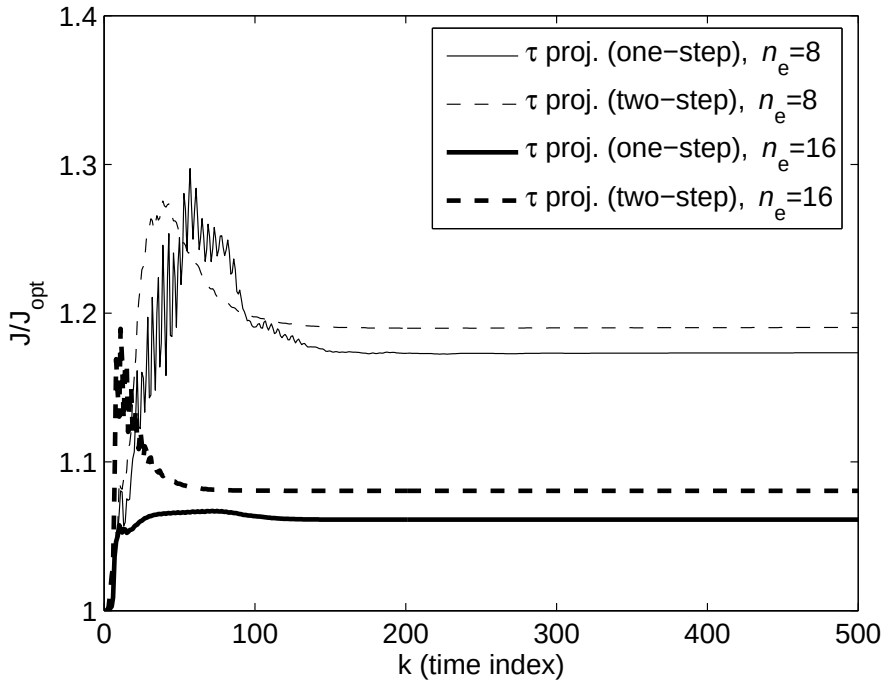


Figure 3.2: Ratios of J to the corresponding full-order cost when the reduced-order estimator is applied to the asymptotically stable mass-spring-dashpot system for $n_{\mathrm{e}} = 8, 16$. The plots demonstrate that the one-step and two-step estimators are not equivalent.

3.5 Conclusion

Using the finite-horizon optimization, an optimal reduced-order estimator was obtained in the form of recursive update equations for time-varying systems. These estimator is characterized by the τ projector, in the recursive update equations. Moreover, we derived one-step and two-step update equations for the reduced-order estimator. When the order of the estimator is equal to the order of the system, the oblique projection becomes the identity and the estimator is equivalent to the classical optimal recursive full-order filter. We demonstrated the performance of the reduced-order estimator for an asymptotically stable lumped-structure. Since the reduced-order estimator does not reduce the computational requirements of propagating the error covariance, we introduce an estimator in the next chapter that reduces the computational requirement of the full-order estimator by propagating a few columns of the square root of the error covariance instead of the entire error covariance matrix.

CHAPTER IV

Cholesky-Based Reduced-Rank Square-Root Kalman Filtering

Although, the reduced-order estimator in the previous chapter used a reduced-order model to update the state estimates, the full-order covariance had to be updated to obtain the optimal estimator gain. In this chapter, we consider a reduced-rank square-root Kalman filter based on the Cholesky decomposition of the state-error covariance. This filter propagates only a few columns of the square root of the state-error covariance. Specifically, the columns are chosen from the Cholesky factor of the state-error covariance. We compare the performance of this filter with the reduced-rank square-root filter based on the singular value decomposition. The results in this chapter are presented in [42].

4.1 Introduction

The problem of state estimation for large-scale systems has gained increasing attention due to computationally intensive applications such as weather forecasting [17, 38], 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, although these issues will not be dealt with in this chapter.

One approach to obtaining more tractable algorithms is to consider reduced-order Kalman filters. These reduced-complexity filters provide state estimates that are suboptimal relative to the classical Kalman filter [7, 8, 25, 26, 39]. Alternative reduced-order variants of the classical Kalman filter have been developed for computationally demanding applications [27, 29, 30, 35], 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 [9].

A widely studied technique for reducing the computational requirements of the Kalman filter for large scale systems is the *reduced-rank filter* [21, 28, 43, 44]. 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 [22, 45].

Reduced-rank filters are closely related to the classical factorization techniques [46, 47], 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) [21, 22, 28, 43–45], 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 criteria to support the optimality of approximation

based on the SVD within the context of recursive state estimation. The difficulty 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 this chapter, we begin by observing that the Kalman filter update depends on the product $C_k P_k$, 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, $O(n^3/6)$ [48], 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.

4.2 The Kalman filter

Consider the discrete-time system

$$x_{k+1} = A_k x_k + G_k w_k, \tag{4.2.1}$$

$$y_k = C_k x_k + H_k v_k, \tag{4.2.2}$$

where $x_k \in \mathbb{R}^n$, $w_k \in \mathbb{R}^{d_w}$, $y_k \in \mathbb{R}^p$, $v_k \in \mathbb{R}^{d_v}$, 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 [5, 6] 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

$$K_k = P_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1}, \tag{4.2.3}$$

$$P_k^{\mathrm{da}} = P_k^{\mathrm{f}} - P_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} C_k P_k^{\mathrm{f}}, \tag{4.2.4}$$

$$x_k^{\mathrm{da}} = x_k^{\mathrm{f}} + K_k (y_k - C_k x_k^{\mathrm{f}}). \tag{4.2.5}$$

Forecast Step

$$x_{k+1}^{\mathrm{f}} = A_k x_k^{\mathrm{da}}, \tag{4.2.6}$$

$$P_{k+1}^{\mathrm{f}} = A_k P_k^{\mathrm{da}} A_k^{\mathrm{T}} + Q_k. \tag{4.2.7}$$

The states x_k^{f} and x_k^{da} are the forecast and data assimilation estimates of the state x_k, while 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,

$$P_k^{\mathrm{f}} = \mathcal{E}[e_k^{\mathrm{f}}(e_k^{\mathrm{f}})^{\mathrm{T}}], \quad P_k^{\mathrm{da}} = \mathcal{E}[e_k^{\mathrm{da}}(e_k^{\mathrm{da}})^{\mathrm{T}}], \tag{4.2.8}$$

where

$$e_k^{\mathrm{f}} \triangleq x_k - x_k^{\mathrm{f}}, \quad e_k^{\mathrm{da}} \triangleq x_k - x_k^{\mathrm{da}}. \tag{4.2.9}$$

Next, we consider two reduced-rank square-root filters for state estimation that propagate approximations of a square-root of the error covariance instead of the actual error covariance.

4.3 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 (4.2.4) and (4.2.7). To reduce the computational requirements, we consider a filter that uses reduced-rank approximations of the error covariances. Instead of updating the error covariances, we propagate predicted error covariances $\tilde{P}_{\mathrm{s},k}^{\mathrm{da}}$ and $\tilde{P}_{\mathrm{s},k}^{\mathrm{f}}$ using reduced-rank approximations $\hat{P}_{\mathrm{s},k}^{\mathrm{da}}$ and $\hat{P}_{\mathrm{s},k}^{\mathrm{f}}$. The reduced-rank approximations are chosen so that $\mathrm{rank}(\hat{P}_{\mathrm{s},k}^{\mathrm{da}}) < n$ and $\mathrm{rank}(\hat{P}_{\mathrm{s},k}^{\mathrm{f}}) < n$, and such that $\|\tilde{P}_{\mathrm{s},k}^{\mathrm{da}} - \hat{P}_{\mathrm{s},k}^{\mathrm{da}}\|_{\mathrm{F}}$ and $\|\tilde{P}_{\mathrm{s},k}^{\mathrm{f}} - \hat{P}_k^{\mathrm{f}}\|_{\mathrm{F}}$ are minimized. To achieve this, we perform a singular value decomposition on the predicted error covariances at every 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 $u_1, \ldots, u_n \in \mathbb{R}^n$ be the corresponding orthogonal singular vectors so that, for $i = 1, \ldots, n$,

$$Pu_i = \sigma_i u_i \tag{4.3.1}$$

and

$$u_i u_j^{\mathrm{T}} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{else.} \end{cases} \tag{4.3.2}$$

Next, define $U_q \in \mathbb{R}^{n\times q}$ and $\Sigma_q \in \mathbb{R}^{q\times q}$ by

$$U_q \triangleq \begin{bmatrix} u_1 & \cdots & u_q \end{bmatrix}, \quad \Sigma_q \triangleq \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_q \end{bmatrix}. \tag{4.3.3}$$

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

$$P = U_n \Sigma_n U_n^{\mathrm{T}}, \tag{4.3.4}$$

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 a square-root of P given by

$$\Phi_{\mathrm{SVD}}(P, q) \triangleq U_q \Sigma_q^{1/2}. \tag{4.3.5}$$

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

Lemma 4.3.1 *Let $P \in \mathbb{R}^{n\times n}$ be positive semidefinite, and let $\sigma_1 \geqslant \cdots \geqslant \sigma_n$ be the singular values of P. If $S = \Phi_{\mathrm{SVD}}(P, q)$, then*

$$\min_{\mathrm{rank}(\hat{P})=q} \|P - \hat{P}\|_{\mathrm{F}} = \|P - SS^{\mathrm{T}}\|_{\mathrm{F}}^2 = \sigma_{q+1}^2 + \cdots + \sigma_n^2. \tag{4.3.6}$$

Proof. See [36]. □

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

Data Assimilation step

$$K_{\mathrm{s},k} = \hat{P}^{\mathrm{f}}_{\mathrm{s},k} C_k^{\mathrm{T}} \left(C_k \hat{P}^{\mathrm{f}}_{\mathrm{s},k} C_k^{\mathrm{T}} + R_k \right)^{-1}, \tag{4.3.7}$$

$$\tilde{P}^{\mathrm{da}}_{\mathrm{s},k} = \hat{P}^{\mathrm{f}}_{\mathrm{s},k} - \hat{P}^{\mathrm{f}}_{\mathrm{s},k} C_k^{\mathrm{T}} \left(C_k \hat{P}^{\mathrm{f}}_{\mathrm{s},k} C_k^{\mathrm{T}} + R_k \right)^{-1} C_k \hat{P}^{\mathrm{f}}_{\mathrm{s},k}, \tag{4.3.8}$$

$$x^{\mathrm{da}}_{\mathrm{s},k} = x^{\mathrm{f}}_{\mathrm{s},k} + K_{\mathrm{s},k}(y_k - C_k x^{\mathrm{f}}_{\mathrm{s},k}), \tag{4.3.9}$$

Forecast step

$$x^{\mathrm{f}}_{\mathrm{s},k+1} = A_k x^{\mathrm{da}}_{\mathrm{s},k}, \tag{4.3.10}$$

$$\tilde{P}^{\mathrm{f}}_{\mathrm{s},k+1} = A_k \hat{P}^{\mathrm{da}}_{\mathrm{s},k} A_k^{\mathrm{T}} + Q_k, \tag{4.3.11}$$

where

$$\hat{P}^{\mathrm{f}}_{\mathrm{s},k} \triangleq \tilde{S}^{\mathrm{f}}_{\mathrm{s},k} (\tilde{S}^{\mathrm{f}}_{\mathrm{s},k})^{\mathrm{T}}, \quad \hat{P}^{\mathrm{da}}_{\mathrm{s},k} \triangleq \tilde{S}^{\mathrm{da}}_{\mathrm{s},k} (\tilde{S}^{\mathrm{da}}_{\mathrm{s},k})^{\mathrm{T}}, \tag{4.3.12}$$

$$\tilde{S}^{\mathrm{f}}_{\mathrm{s},k} \triangleq \Phi_{\mathrm{SVD}}(\tilde{P}^{\mathrm{f}}_{\mathrm{s},k}, q), \quad \tilde{S}^{\mathrm{da}}_{\mathrm{s},k} \triangleq \Phi_{\mathrm{SVD}}(\tilde{P}^{\mathrm{da}}_{\mathrm{s},k}, q), \tag{4.3.13}$$

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

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

$$P^{\mathrm{f}}_{\mathrm{s},k} \triangleq \mathcal{E}\left[(x_k - x^{\mathrm{f}}_{\mathrm{s},k})(x_k - x^{\mathrm{f}}_{\mathrm{s},k})^{\mathrm{T}} \right], \quad P^{\mathrm{da}}_{\mathrm{s},k} \triangleq \mathcal{E}\left[(x_k - x^{\mathrm{da}}_{\mathrm{s},k})(x_k - x^{\mathrm{da}}_{\mathrm{s},k})^{\mathrm{T}} \right]. \tag{4.3.14}$$

Using (4.2.1), (4.3.9) and (4.3.10), it follows that

$$P^{\mathrm{da}}_{\mathrm{s},k} = (I - K_{\mathrm{s},k} C_k) P^{\mathrm{f}}_{\mathrm{s},k} (I - K_{\mathrm{s},k} C_k)^{\mathrm{T}} + K_{\mathrm{s},k} R_k K_{\mathrm{s},k}^{\mathrm{T}}, \tag{4.3.15}$$

$$P^{\mathrm{f}}_{\mathrm{s},k} = A_k P^{\mathrm{da}}_{\mathrm{s},k} A_k^{\mathrm{T}} + Q_k. \tag{4.3.16}$$

Note that $\tilde{P}^{\mathrm{f}}_{\mathrm{s},k}$ and $\tilde{P}^{\mathrm{da}}_{\mathrm{s},k}$ are predicted error covariances and not covariances of the state error. Specifically, even if $\tilde{P}^{\mathrm{f}}_{\mathrm{s},0} = P^{\mathrm{f}}_0$, it does not necessarily follow that $\tilde{P}^{\mathrm{f}}_{\mathrm{s},k} = P^{\mathrm{f}}_k$ for all $k > 0$. Furthermore, since $K_{\mathrm{s},k} \neq K_k$, the SVD-based rank-$q$ square-root filter is a suboptimal 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 4.3.1 *Assume that* $\tilde{P}^{\mathrm{f}}_{\mathrm{s},k} = P^{\mathrm{f}}_k$ *and* $\operatorname{rank}(P^{\mathrm{f}}_k) \leqslant q$. *Then,* $K_{\mathrm{s},k} = K_k$, $\tilde{P}^{\mathrm{da}}_{\mathrm{s},k} = P^{\mathrm{da}}_k$, *and* $\tilde{P}^{\mathrm{f}}_{\mathrm{s},k+1} = P^{\mathrm{f}}_{k+1}$.

Proof. Since $\operatorname{rank}(\tilde{P}^{\mathrm{f}}_{\mathrm{s},k}) \leqslant q$, it follows from Lemma 4.3.1 that

$$\hat{P}^{\mathrm{f}}_{\mathrm{s},k} = \tilde{S}^{\mathrm{f}}_{\mathrm{s},k} \left(\tilde{S}^{\mathrm{f}}_{\mathrm{s},k}\right)^{\mathrm{T}} = \tilde{P}^{\mathrm{f}}_{\mathrm{s},k}. \tag{4.3.17}$$

Hence, it follows from (4.3.7) that $K_{\mathrm{s},k} = K_k$. Furthermore, it follows from (4.2.4), (4.3.8), and (4.3.17) that

$$\tilde{P}^{\mathrm{da}}_{\mathrm{s},k} = P^{\mathrm{da}}_k. \tag{4.3.18}$$

Since $\operatorname{rank}(P^{\mathrm{f}}_k) \leqslant q$, it follows from (4.2.4) that $\operatorname{rank}(P^{\mathrm{da}}_k) \leqslant q$ and hence (4.3.18) implies that $\operatorname{rank}(\tilde{P}^{\mathrm{da}}_{\mathrm{s},k}) \leqslant q$. Therefore, Lemma 4.3.1, (4.3.12) and (4.3.13) imply that

$$\hat{P}^{\mathrm{da}}_{\mathrm{s},k} = \tilde{S}^{\mathrm{da}}_{\mathrm{s},k} \left(\tilde{S}^{\mathrm{da}}_{\mathrm{s},k}\right)^{\mathrm{T}} = \tilde{P}^{\mathrm{da}}_{\mathrm{s},k}. \tag{4.3.19}$$

Hence, it follows from (4.3.18) and (4.3.19) that $\hat{P}^{\mathrm{da}}_{\mathrm{s},k} = P^{\mathrm{da}}_k$, and therefore (4.2.7) and (4.3.11) imply that $\tilde{P}^{\mathrm{f}}_{\mathrm{s},k+1} = P^{\mathrm{f}}_{k+1}$. □

Corollary 4.3.1 *Assume that* $x^{\mathrm{f}}_{\mathrm{s},0} = x^{\mathrm{f}}_0$, $\tilde{P}^{\mathrm{f}}_{\mathrm{s},0} = P^{\mathrm{f}}_0$, *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. It follows from (4.2.4) and (4.2.7) that $\operatorname{rank}(P_k^{\mathrm{f}}) \leqslant q$ for all k. Hence, using Proposition 4.3.1 and induction, it can be shown that $K_{\mathrm{s},k} = K_k$ for all $k \geqslant 0$. Therefore, (4.2.5), (4.2.6), (4.3.9) and (4.3.10) imply that $x_{\mathrm{s},k}^{\mathrm{f}} = x_k^{\mathrm{f}}$ for all $k \geqslant 0$. □

4.4 Cholesky-Factorization-Based Reduced-Rank Square-Root Filter

The Kalman filter gain K_k depends on a particular subspace of the error covariance. Specifically, K_k depends only on the correlation $C_k P_k^{\mathrm{f}}$ between the error in the measured states and the unmeasured states. We thus have the following observation.

Lemma 4.4.1 *Assume that $\hat{P}_k \in \mathbb{R}^{n\times n}$ is positive semidefinite. Partition $\hat{P}_k$ and P_k^{f} as*

$$\hat{P}_k = \begin{bmatrix} \hat{P}_{q,k} & (\hat{P}_{\overline{q}q,k})^{\mathrm{T}} \\ \hat{P}_{\overline{q}q,k} & \hat{P}_{\overline{q},k} \end{bmatrix}, \quad P_k^{\mathrm{f}} = \begin{bmatrix} P_{q,k}^{\mathrm{f}} & (P_{\overline{q}q,k}^{\mathrm{f}})^{\mathrm{T}} \\ P_{\overline{q}q,k}^{\mathrm{f}} & P_{\overline{q},k}^{\mathrm{f}} \end{bmatrix}, \tag{4.4.1}$$

where $\hat{P}_{q,k}, P_{q,k}^{\mathrm{f}} \in \mathbb{R}^{q\times q}$ and $\hat{P}_{\overline{q},k}, P_{\overline{q},k}^{\mathrm{f}} \in \mathbb{R}^{\overline{q}\times\overline{q}}$, assume that C_k has the form

$$C_k = \begin{bmatrix} I_q & 0 \end{bmatrix}, \tag{4.4.2}$$

and define $\hat{K}_k$ by

$$\hat{K}_k \triangleq \hat{P}_k C_k^{\mathrm{T}} (C_k \hat{P}_k C_k^{\mathrm{T}} + R_k)^{-1}. \tag{4.4.3}$$

Furthermore, let $\begin{bmatrix} \hat{P}_{q,k} & \left(\hat{P}_{\overline{q}q,k}\right)^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} P_{q,k}^{\mathrm{f}} & \left(P_{\overline{q}q,k}^{\mathrm{f}}\right)^{\mathrm{T}} \end{bmatrix}$. Then, $\hat{K}_k = K_k$.

Proof. It follows from (4.4.1) and (4.4.2) that

$$C_k \hat{P}_k = \begin{bmatrix} \hat{P}_{q,k} & (\hat{P}_{\overline{q}q,k})^{\mathrm{T}} \end{bmatrix}, \quad C_k P_k^{\mathrm{f}} = \begin{bmatrix} P_{q,k}^{\mathrm{f}} & (\hat{P}_{\overline{q}q,k})^{\mathrm{T}} \end{bmatrix}, \tag{4.4.4}$$

and

$$C_k \hat{P}_k C_k^{\mathrm{T}} = \hat{P}_{q,k}, \;\; C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} = P_{q,k}^{\mathrm{f}}. \tag{4.4.5}$$

Hence, it follows from (4.2.3) and (4.4.3) that $\hat{K}_k = K_k$. □

Next, we consider a filter that updates the predicted error covariances $\tilde{P}_{\mathrm{c},k}^{\mathrm{da}}$ and $\tilde{P}_{\mathrm{c},k}^{\mathrm{f}}$ using reduced-rank approximations $\hat{P}_{\mathrm{c},k}^{\mathrm{da}}$ and $\hat{P}_{\mathrm{c},k}^{\mathrm{f}}$ such that $\mathrm{rank}(\hat{P}_{\mathrm{c},k}^{\mathrm{da}}) < n$ and $\mathrm{rank}(\hat{P}_{\mathrm{c},k}^{\mathrm{f}}) < n$, and such that $\|C_k(\tilde{P}_{\mathrm{c},k}^{\mathrm{da}} - \hat{P}_{\mathrm{c},k}^{\mathrm{da}})\|_{\mathrm{F}}$ and $\|C_k(\tilde{P}_{\mathrm{c},k}^{\mathrm{f}} - \hat{P}_{\mathrm{c},k}^{\mathrm{f}})\|_{\mathrm{F}}$ are minimized. To achieve this, we perform a Cholesky factorization of the predicted error covariances at every time step.

Let $P \in \mathbb{R}^{n \times n}$ be positive definite. The Cholesky factorization yields a lower triangular Cholesky factor $L \in \mathbb{R}^{n \times n}$ that satisfies

$$LL^{\mathrm{T}} = P. \tag{4.4.6}$$

Partition L as

$$L = \left[\begin{array}{ccc} L_1 & \cdots & L_n \end{array} \right], \tag{4.4.7}$$

so that truncating the last $n - q$ columns of L yields the rank-q Cholesky factor

$$\Phi_{\mathrm{CHOL}}(P, q) \triangleq \left[\begin{array}{ccc} L_1 & \cdots & L_q \end{array} \right] \in \mathbb{R}^{n \times q}. \tag{4.4.8}$$

Lemma 4.4.2 *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*

$$P = \left[\begin{array}{cc} P_q & P_{q\bar{q}} \\ (P_{q\bar{q}})^{\mathrm{T}} & P_{\bar{q}} \end{array} \right], \;\; \hat{P} = \left[\begin{array}{cc} \hat{P}_q & \hat{P}_{q\bar{q}} \\ (\hat{P}_{q\bar{q}})^{\mathrm{T}} & \hat{P}_{\bar{q}} \end{array} \right], \tag{4.4.9}$$

where $P_q, \hat{P}_q \in \mathbb{R}^{q \times q}$ and $P_{\bar{q}}, \hat{P}_{\bar{q}} \in \mathbb{R}^{\bar{q} \times \bar{q}}$. Then, $\left[\begin{array}{cc} \hat{P}_q & \hat{P}_{q\bar{q}} \end{array} \right] = \left[\begin{array}{cc} P_q & P_{q\bar{q}} \end{array} \right]$.

Proof. Let L be the Cholesky factor of P. Since L is lower triangular, $L_i L_i^{\mathrm{T}}$ has the structure

$$L_i L_i^{\mathrm{T}} = \begin{bmatrix} 0_{i-1} & 0_{(i-1)\times(n-i+1)} \\ 0_{(n-i+1)\times(i-1)} & X_i \end{bmatrix}, \tag{4.4.10}$$

and therefore

$$\sum_{i=q+1}^{n} L_i L_i^{\mathrm{T}} = \begin{bmatrix} 0_q & 0_{q\times\overline{q}} \\ 0_{\overline{q}\times q} & Y_{\overline{q}} \end{bmatrix}, \tag{4.4.11}$$

where $Y_{\overline{q}} \in \mathbb{R}^{\overline{q}\times\overline{q}}$. Since

$$P = \sum_{i=1}^{n} L_i L_i^{\mathrm{T}}, \tag{4.4.12}$$

it follows from (4.4.8) that

$$P = \hat{P} + \sum_{i=q+1}^{n} L_i L_i^{\mathrm{T}}. \tag{4.4.13}$$

Substituting (4.4.11) into (4.4.13) yields $\hat{P}_q = P_q$ and $\hat{P}_{q\overline{q}} = P_{q\overline{q}}$. □

Lemma 4.4.2 implies that, if $S = \Phi_{\mathrm{CHOL}}(P, q)$, then the first q columns and rows 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

$$K_{\mathrm{c},k} = \hat{P}^{\mathrm{f}}_{\mathrm{c},k} C_k^{\mathrm{T}} \left(C_k \hat{P}^{\mathrm{f}}_{\mathrm{c},k} C_k^{\mathrm{T}} + R_k \right)^{-1}, \tag{4.4.14}$$

$$\tilde{P}^{\mathrm{da}}_{\mathrm{c},k} = \hat{P}^{\mathrm{f}}_{\mathrm{c},k} - \hat{P}^{\mathrm{f}}_{\mathrm{c},k} C_k^{\mathrm{T}} \left(C_k \hat{P}^{\mathrm{f}}_{\mathrm{c},k} C_k^{\mathrm{T}} + R_k \right)^{-1} C_k \hat{P}^{\mathrm{f}}_{\mathrm{c},k}, \tag{4.4.15}$$

$$x^{\mathrm{da}}_{\mathrm{c},k} = x^{\mathrm{f}}_{\mathrm{c},k} + K_{\mathrm{c},k}(y_k - C_k x^{\mathrm{f}}_{\mathrm{c},k}). \tag{4.4.16}$$

Forecast step

$$x^{\mathrm{f}}_{\mathrm{c},k+1} = A_k x^{\mathrm{da}}_{\mathrm{c},k}, \tag{4.4.17}$$

$$\tilde{P}^{\mathrm{f}}_{\mathrm{c},k+1} = A_k \hat{P}^{\mathrm{da}}_{\mathrm{c},k} A_k^{\mathrm{T}} + Q_k, \tag{4.4.18}$$

where

$$\hat{P}^{\mathrm{f}}_{\mathrm{c},k} \triangleq \tilde{S}^{\mathrm{f}}_{\mathrm{c},k}\left(\tilde{S}^{\mathrm{f}}_{\mathrm{c},k}\right)^{\mathrm{T}}, \quad \hat{P}^{\mathrm{da}}_{\mathrm{c},k} \triangleq \tilde{S}^{\mathrm{da}}_{\mathrm{c},k}\left(\tilde{S}^{\mathrm{da}}_{\mathrm{c},k}\right)^{\mathrm{T}}, \tag{4.4.19}$$

$$\tilde{S}^{\mathrm{f}}_{\mathrm{c},k} \triangleq \Phi_{\mathrm{CHOL}}(\tilde{P}^{\mathrm{f}}_{\mathrm{c},k}, q), \quad \tilde{S}^{\mathrm{da}}_{\mathrm{c},k} \triangleq \Phi_{\mathrm{CHOL}}(\tilde{P}^{\mathrm{da}}_{\mathrm{c},k}, q), \tag{4.4.20}$$

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

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

$$P^{\mathrm{f}}_{\mathrm{c},k} \triangleq \mathcal{E}\left[(x_k - x^{\mathrm{f}}_{\mathrm{c},k})(x_k - x^{\mathrm{f}}_{\mathrm{c},k})^{\mathrm{T}}\right], \quad P^{\mathrm{da}}_{\mathrm{s},k} \triangleq \mathcal{E}\left[(x_k - x^{\mathrm{da}}_{\mathrm{c},k})(x_k - x^{\mathrm{da}}_{\mathrm{c},k})^{\mathrm{T}}\right], \tag{4.4.21}$$

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 (4.2.1), (4.4.16) and (4.4.17), it can be shown that

$$P^{\mathrm{da}}_{\mathrm{c},k} = (I - K_{\mathrm{c},k}C_k)P^{\mathrm{f}}_{\mathrm{c},k}(I - K_{\mathrm{c},k}C_k)^{\mathrm{T}} + K_{\mathrm{c},k}R_kK^{\mathrm{T}}_{\mathrm{c},k}, \tag{4.4.22}$$

$$P^{\mathrm{f}}_{\mathrm{c},k} = A_kP^{\mathrm{da}}_{\mathrm{c},k}A^{\mathrm{T}}_k + Q_k. \tag{4.4.23}$$

Again, like the SVD-based rank-q square-root filter, $\tilde{P}^{\mathrm{f}}_{\mathrm{c},k}$ and $\tilde{P}^{\mathrm{da}}_{\mathrm{c},k}$ are predicted error covariances and not covariances of the state error. Hence, even if $\tilde{P}^{\mathrm{f}}_{\mathrm{c},0} = P^{\mathrm{f}}_0$, the Cholesky-based rank-q square-root filter is suboptimal and generally not equivalent to the Kalman filter. However, the following result shows that, in certain cases, the Cholesky-based rank-q square-root filter is equivalent to the Kalman filter.

Proposition 4.4.1 *Assume that $p = q$, C_k has the form*

$$C_k = \begin{bmatrix} I_q & 0 \end{bmatrix}, \tag{4.4.24}$$

partition P^{f}_k and $\tilde{P}^{\mathrm{f}}_{\mathrm{c},k}$ as

$$P^{\mathrm{f}}_k = \begin{bmatrix} P^{\mathrm{f}}_{q,k} & (P^{\mathrm{f}}_{\overline{q}q,k})^{\mathrm{T}} \\ P^{\mathrm{f}}_{\overline{q}q,k} & P^{\mathrm{f}}_{\overline{q},k} \end{bmatrix}, \quad \tilde{P}^{\mathrm{f}}_{\mathrm{c},k} = \begin{bmatrix} \tilde{P}^{\mathrm{f}}_{\mathrm{c},q,k} & (\tilde{P}^{\mathrm{f}}_{\mathrm{c},\overline{q}q,k})^{\mathrm{T}} \\ \tilde{P}^{\mathrm{f}}_{\mathrm{c},\overline{q}q,k} & \tilde{P}^{\mathrm{f}}_{\mathrm{c}\overline{q},k} \end{bmatrix}, \tag{4.4.25}$$

where $P^{\rm f}_{q,k}, \tilde{P}^{\rm f}_{{\rm c},q,k} \in \mathbb{R}^{q\times q}$ *and* $P^{\rm f}_{\overline{q},k}, \tilde{P}^{\rm f}_{{\rm c},\overline{q},k} \in \mathbb{R}^{\overline{q}\times\overline{q}}$, *and assume that* $\left[\begin{array}{cc} \tilde{P}^{\rm f}_{{\rm c},q,k} & \tilde{P}^{\rm f}_{{\rm c},\overline{q}q,k} \end{array}\right] = \left[\begin{array}{cc} P^{\rm f}_{q,k} & P^{\rm f}_{\overline{q}q,k} \end{array}\right]$. *Then,* $K_{{\rm c},k} = K_k$. *If, in addition,* A_k *has the form*

$$A_k = \left[\begin{array}{cc} A_{q,k} & 0 \\ A_{\overline{q}q,k} & A_{\overline{q},k} \end{array}\right], \tag{4.4.26}$$

where $A_{q,k} \in \mathbb{R}^{q\times q}$ *and* $A_{\overline{q},k} \in \mathbb{R}^{\overline{q}\times\overline{q}}$, *then* $\left[\begin{array}{cc} \tilde{P}^{\rm f}_{{\rm c},q,k+1} & \tilde{P}^{\rm f}_{{\rm c},\overline{q}q,k+1} \end{array}\right] = \left[\begin{array}{cc} P^{\rm f}_{q,k+1} & P^{\rm f}_{\overline{q}q,k+1} \end{array}\right]$.

Proof. Partition $\hat{P}^{\rm f}_{{\rm c},k}$ as

$$\hat{P}^{\rm f}_{{\rm c},k} = \left[\begin{array}{cc} \hat{P}^{\rm f}_{{\rm c},q,k} & (\hat{P}^{\rm f}_{{\rm c},\overline{q}q,k})^{\rm T} \\ \hat{P}^{\rm f}_{{\rm c},\overline{q}q,k} & \hat{P}^{\rm f}_{{\rm c},\overline{q},k} \end{array}\right], \tag{4.4.27}$$

where $\hat{P}^{\rm f}_{q,k} \in \mathbb{R}^{q\times q}$ is positive semidefinite and $\hat{P}^{\rm f}_{\overline{q},k} \in \mathbb{R}^{\overline{q}\times\overline{q}}$. It follows from Lemma 4.4.2 and (4.4.20) that

$$\hat{P}^{\rm f}_{{\rm c},q,k} = \tilde{P}^{\rm f}_{{\rm c},q,k}, \quad \hat{P}^{\rm f}_{{\rm c},\overline{q}q,k} = \tilde{P}^{\rm f}_{{\rm c},\overline{q}q,k}. \tag{4.4.28}$$

Therefore, it follows from Lemma 4.4.1 and (4.4.14) that $K_{{\rm c},k} = K_k$.

Next, partition $P^{\rm da}_k$ as

$$P^{\rm da}_k = \left[\begin{array}{cc} P^{\rm da}_{q,k} & (P^{\rm da}_{\overline{q}q,k})^{\rm T} \\ P^{\rm da}_{\overline{q}q,k} & P^{\rm da}_{\overline{q},k} \end{array}\right], \tag{4.4.29}$$

where $P^{\rm da}_{q,k} \in \mathbb{R}^{q\times q}$ is positive semidefinite and $P^{\rm da}_{\overline{q},k} \in \mathbb{R}^{\overline{q}\times\overline{q}}$. It follows from (4.2.4) that

$$P^{\rm da}_{q,k} = P^{\rm f}_{q,k} - P^{\rm f}_{q,k}(P^{\rm f}_{q,k} + R_k)^{-1}P^{\rm f}_{q,k}, \tag{4.4.30}$$

$$P^{\rm da}_{\overline{q}q,k} = P^{\rm f}_{\overline{q}q,k} - P^{\rm f}_{\overline{q}q,k}(P^{\rm f}_{q,k} + R_k)^{-1}P^{\rm f}_{q,k}. \tag{4.4.31}$$

Now, partition $\tilde{P}^{\rm da}_{{\rm c},k}$ and $\hat{P}^{\rm da}_{{\rm c},k}$ as

$$\tilde{P}^{\rm da}_{{\rm c},k} = \left[\begin{array}{cc} \tilde{P}^{\rm da}_{{\rm c},q,k} & (\tilde{P}^{\rm da}_{{\rm c},\overline{q}q,k})^{\rm T} \\ \tilde{P}^{\rm da}_{{\rm c},\overline{q}q,k} & \tilde{P}^{\rm da}_{{\rm c},\overline{q},k} \end{array}\right], \quad \hat{P}^{\rm da}_{{\rm c},k} = \left[\begin{array}{cc} \hat{P}^{\rm da}_{{\rm c},q,k} & (\hat{P}^{\rm da}_{{\rm c},\overline{q}q,k})^{\rm T} \\ \hat{P}^{\rm da}_{{\rm c},\overline{q}q,k} & \hat{P}^{\rm da}_{{\rm c},\overline{q},k} \end{array}\right], \tag{4.4.32}$$

where $\tilde{P}^{\mathrm{da}}_{q,k}, \hat{P}^{\mathrm{da}}_{q,k} \in \mathbb{R}^{q\times q}$ are positive semidefinite and $\tilde{P}^{\mathrm{da}}_{\overline{q},k}, \hat{P}^{\mathrm{da}}_{\overline{q},k} \in \mathbb{R}^{\overline{q}\times\overline{q}}$. Therefore, it follows from (4.4.15), (4.4.24), (4.4.27), and (4.4.32) that

$$\tilde{P}^{\mathrm{da}}_{\mathrm{c},q,k} = \hat{P}^{\mathrm{f}}_{\mathrm{c},q,k} - \hat{P}^{\mathrm{f}}_{\mathrm{c},q,k}(\hat{P}^{\mathrm{f}}_{\mathrm{c},q,k} + R_k)^{-1}\hat{P}^{\mathrm{f}}_{\mathrm{c},q,k}, \tag{4.4.33}$$

$$\tilde{P}^{\mathrm{da}}_{\mathrm{c},\overline{q}q,k} = \hat{P}^{\mathrm{f}}_{\mathrm{c},\overline{q}q,k} - \hat{P}^{\mathrm{f}}_{\mathrm{c},\overline{q}q,k}(\hat{P}^{\mathrm{f}}_{\mathrm{c},q,k} + R_k)^{-1}\hat{P}^{\mathrm{f}}_{\mathrm{c},q,k}. \tag{4.4.34}$$

Hence, comparing (4.4.30) with (4.4.33) and (4.4.31) with (4.4.34), and using

$$\begin{bmatrix} \tilde{P}^{\mathrm{f}}_{\mathrm{c},q,k} & \left(\tilde{P}^{\mathrm{f}}_{\mathrm{c},\overline{q}q,k}\right)^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} P^{\mathrm{f}}_{q,k} & \left(P^{\mathrm{f}}_{\overline{q}q,k}\right)^{\mathrm{T}} \end{bmatrix} \tag{4.4.35}$$

and (4.4.28) yields

$$\tilde{P}^{\mathrm{da}}_{\mathrm{c},q,k} = P^{\mathrm{da}}_{q,k}, \quad \tilde{P}^{\mathrm{da}}_{\mathrm{c},\overline{q}q,k} = P^{\mathrm{da}}_{\overline{q}q,k}. \tag{4.4.36}$$

Moreover, since $S^{\mathrm{da}}_{\mathrm{c},k} = \Phi_{\mathrm{CHOL}}(\tilde{P}^{\mathrm{da}}_{\mathrm{c},k}, q)$, it follows from Lemma 4.4.2 that

$$\hat{P}^{\mathrm{da}}_{\mathrm{c},q,k} = \tilde{P}^{\mathrm{da}}_{\mathrm{c},q,k}, \quad \hat{P}^{\mathrm{da}}_{\mathrm{c},\overline{q}q,k} = \tilde{P}^{\mathrm{da}}_{\mathrm{c},\overline{q}q,k}. \tag{4.4.37}$$

Therefore, (4.4.36) implies that

$$\hat{P}^{\mathrm{da}}_{\mathrm{c},q,k} = P^{\mathrm{da}}_{q,k}, \quad \hat{P}^{\mathrm{da}}_{\mathrm{c},\overline{q}q,k} = P^{\mathrm{da}}_{\overline{q}q,k}. \tag{4.4.38}$$

Now assume that A_k has the form (4.4.26). Then (4.2.7) implies that

$$P^{\mathrm{f}}_{q,k+1} = A_{q,k}P^{\mathrm{da}}_{q,k}A^{\mathrm{T}}_{q,k} + Q_{q,k}, \tag{4.4.39}$$

$$P^{\mathrm{f}}_{\overline{q}q,k+1} = A_{\overline{q},k}P^{\mathrm{da}}_{\overline{q}q,k}A^{\mathrm{T}}_{q,k} + A_{\overline{q}q,k}P^{\mathrm{da}}_{q,k}A^{\mathrm{T}}_{q,k} + Q_{\overline{q}q,k}, \tag{4.4.40}$$

where Q_k has entries

$$Q_k = \begin{bmatrix} Q_{q,k} & (Q_{\overline{q}q,k})^{\mathrm{T}} \\ Q_{\overline{q}q,k} & Q_{\overline{q},k} \end{bmatrix}. \tag{4.4.41}$$

Furthermore, it follows from (4.4.18), (4.4.26) and (4.4.32) that

$$\tilde{P}^{\mathrm{f}}_{\mathrm{c},q,k+1} = A_{q,k}\hat{P}^{\mathrm{da}}_{\mathrm{c},q,k}A^{\mathrm{T}}_{q,k} + Q_{q,k}, \tag{4.4.42}$$

$$\tilde{P}^{\mathrm{f}}_{\mathrm{c},\overline{q}q,k+1} = A_{\overline{q},k}\hat{P}^{\mathrm{da}}_{\mathrm{c},\overline{q}q,k}A^{\mathrm{T}}_{q,k} + A_{\overline{q}q,k}\hat{P}^{\mathrm{da}}_{\mathrm{c},q,k}A^{\mathrm{T}}_{q,k} + Q_{\overline{q}q,k}. \tag{4.4.43}$$

Therefore, (4.4.38), (4.4.39), (4.4.40), (4.4.42), and (4.4.43) imply that $\tilde{P}^{\mathrm{f}}_{\mathrm{c},q,k+1} = P^{\mathrm{f}}_{q,k+1}$ and $\tilde{P}^{\mathrm{f}}_{\mathrm{c},\bar{q}q,k+1} = P^{\mathrm{f}}_{\bar{q}q,k+1}$. □

Corollary 4.4.1 *Assume that C_k and A_k are of the form* (4.4.24) *and* (4.4.26). *Let $\tilde{P}^{\mathrm{f}}_{\mathrm{c},q,0} = P^{\mathrm{f}}_{q,0}$, $\tilde{P}^{\mathrm{f}}_{\mathrm{c},\bar{q}q,0} = P^{\mathrm{f}}_{\bar{q}q,0}$, 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 4.4.1 yields $K_{\mathrm{c},k} = K_k$ for all $k \geqslant 0$. Hence, it follows from (4.2.5), (4.2.6), (4.4.16), and (4.4.17) that $x^{\mathrm{f}}_{\mathrm{c},k} = x^{\mathrm{f}}_k$ for all $k \geqslant 0$. □

4.4.1 Linear Time-Invariant Systems

Next, we consider linear time-invariant systems and hence assume that, for all $k \geqslant 0$, $A_k = A$, $C_k = C$, $G_k = G$, $H_k = H$, $Q_k = Q$, and $R_k = R$. Next, we assume that $p < n$ and (A, C) is observable so that the observability matrix $\mathcal{O} \in \mathbb{R}^{pn \times n}$ defined by

$$\mathcal{O} \triangleq \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{4.4.44}$$

has full column rank. Next, without loss of generality we consider a basis such that

$$\mathcal{O} = \begin{bmatrix} I_n \\ 0_{(p-1)n \times n} \end{bmatrix}. \tag{4.4.45}$$

Therefore, (4.4.44) and (4.4.45) imply that, for every positive integer i such that $ip \leqslant n$,

$$CA^{i-1} = \begin{bmatrix} 0_{p \times p(i-1)} & I_p & 0_{p \times (n-pi)} \end{bmatrix}. \tag{4.4.46}$$

Next, we present a result that shows that the Cholesky-based rank-q square-root filter is equivalent to the Kalman filter for a specific number of time steps. To do this, we first present the following results.

Lemma 4.4.3 *Let i be a positive integer, and for all $k > 0$, let $\hat{P}_k \in \mathbb{R}^{n\times n}$ satisfy*

$$CA^{i-1}\hat{P}_{k+1} = CA^i\hat{P}_kA^{\mathrm{T}} - CA^i\hat{P}_kC^{\mathrm{T}}(C\hat{P}_kC + R)^{-1}C\hat{P}_kA^{\mathrm{T}} + CA^{i-1}Q. \quad (4.4.47)$$

Assume that $CA^i\hat{P}_k = CA^iP_k^{\mathrm{f}}$ and $C\hat{P}_k = CP_k^{\mathrm{f}}$. Then, $CA^{i-1}\hat{P}_{k+1} = CA^{i-1}P_{k+1}^{\mathrm{f}}$.

Proof. Substituting (4.2.4) into (4.2.7) yields

$$P_{k+1}^{\mathrm{f}} = AP_k^{\mathrm{f}}A^{\mathrm{T}} - AP_k^{\mathrm{f}}C^{\mathrm{T}}(CP_k^{\mathrm{f}}C^{\mathrm{T}} + R)^{-1}CP_k^{\mathrm{f}}A^{\mathrm{T}} + Q. \quad (4.4.48)$$

Pre-multiplying (4.4.48) by CA^{i-1} and comparing the resulting equation with (4.4.47) yields the result. □

Lemma 4.4.4 *Assume that $\hat{P}_k \in \mathbb{R}^{n\times n}$ satisfies (4.4.47) for all $k > 0$ and $i = 1, \ldots, r$. Let $CA^{i-1}\hat{P}_0 = CA^{i-1}P_0^{\mathrm{f}}$ for $i = 1, \ldots, r$. Then, for all $k = 0, \ldots, r$, $C\hat{P}_k = CP_k^{\mathrm{f}}$.*

Proof. It follows from Lemma 4.4.3 that, for $i = 0, \ldots, r-2$, $CA^i\hat{P}_1 = CA^iP_1^{\mathrm{f}}$. The result follows from repeated application of Lemma 4.4.3. □

Proposition 4.4.2 *Let $r > 0$ be an integer such that $0 < q = pr < n$. Furthermore, assume that $\tilde{P}_{\mathrm{c},0}^{\mathrm{f}} = P_0^{\mathrm{f}}$. Then, for all $k = 0, \ldots, r$, $K_{\mathrm{c},k} = K_k$. If, in addition, $x_{\mathrm{c},0}^{\mathrm{f}} = x_0^{\mathrm{f}}$, then for all $k = 0, \ldots, r$, $x_{\mathrm{c},k}^{\mathrm{f}} = x_k^{\mathrm{f}}$.*

Proof. It follows from Lemma 4.4.2 and (4.4.46) that, for all $k \geqslant 0$ and $i = 1, \ldots, r$,

$$CA^{i-1}\hat{P}_{\mathrm{c},k}^{\mathrm{f}} = CA^{i-1}\tilde{P}_{\mathrm{c},k}^{\mathrm{f}}, \quad CA^{i-1}\hat{P}_{\mathrm{c},k}^{\mathrm{da}} = CA^{i-1}\tilde{P}_{\mathrm{c},k}^{\mathrm{da}}. \quad (4.4.49)$$

Note that

$$\tilde{P}^{\mathrm{f}}_{\mathrm{c},k+1} = A\hat{P}^{\mathrm{da}}_{\mathrm{c},k}A^{\mathrm{T}} + Q. \tag{4.4.50}$$

Multiplying (4.4.50) by CA^{i-1} yields

$$CA^{i-1}\tilde{P}^{\mathrm{f}}_{\mathrm{c},k+1} = CA^{i}\hat{P}^{\mathrm{da}}_{\mathrm{c},k}A^{\mathrm{T}} + CA^{i-1}Q. \tag{4.4.51}$$

Substituting (4.4.49) into (4.4.51) yields

$$CA^{i-1}\hat{P}^{\mathrm{f}}_{\mathrm{c},k+1} = CA^{i}\tilde{P}^{\mathrm{da}}_{\mathrm{c},k}A^{\mathrm{T}} + CA^{i-1}Q, \tag{4.4.52}$$

for $i = 1, \ldots, r$. Using (4.4.15) in (4.4.52) yields

$$CA^{i-1}\hat{P}^{\mathrm{f}}_{\mathrm{c},k+1} = CA^{i}\left[\hat{P}^{\mathrm{f}}_{\mathrm{c},k} - \hat{P}^{\mathrm{f}}_{\mathrm{c},k}C^{\mathrm{T}}(C\hat{P}^{\mathrm{f}}_{\mathrm{c},k}C^{\mathrm{T}} + R)^{-1}C\hat{P}^{\mathrm{f}}_{\mathrm{c},k}\right]A^{\mathrm{T}} + CA^{i-1}Q, \tag{4.4.53}$$

for all $k \geqslant 0$ and $i = 1, \ldots, r$. Since $\tilde{P}^{\mathrm{f}}_{\mathrm{c},0} = P^{\mathrm{f}}_{0}$, it follows from Lemma 4.4.2 that, for $i = 1, \ldots, r$,

$$CA^{i-1}\hat{P}^{\mathrm{f}}_{\mathrm{c},0} = CA^{i-1}P^{\mathrm{f}}_{0}. \tag{4.4.54}$$

Hence, it follows from (4.4.53) and Lemma 4.4.4 that, for $k = 0, \ldots, r$,

$$C\hat{P}^{\mathrm{f}}_{k} = CP^{\mathrm{f}}_{k}. \tag{4.4.55}$$

Finally, (4.2.3) and (4.4.14) imply that, for $k = 0, \ldots, r$,

$$K_{\mathrm{c},k} = K_{k}. \tag{4.4.56}$$

Hence, it follows from (4.2.5), (4.2.6), (4.4.16), and (4.4.17) that for all $k = 0, \ldots, r$, $x^{\mathrm{f}}_{\mathrm{c},k} = x^{\mathrm{f}}_{k}$. □

Hence, the Cholesky-based rank-q square-root filter is equivalent to the Kalman filter for a fixed number of time steps that depend on the rank q of the approximations

$\hat{P}^{\mathrm{da}}_{\mathrm{c},k}$ and $\hat{P}^{\mathrm{f}}_{\mathrm{c},k}$ of the predicted error covariances $\tilde{P}^{\mathrm{da}}_{\mathrm{c},k}$ and $\tilde{P}^{\mathrm{f}}_{\mathrm{c},k}$, as well as the dimension p of the output. However, in general $\tilde{P}^{\mathrm{f}}_{\mathrm{c},k}$ and P^{f}_{k} are not equal for all $k = 0, \ldots, r$ even though Proposition 4.4.2 implies that $K_{\mathrm{c},k}$ and K_k are equal. Moreover, $K_{\mathrm{c},k}$ and K_k are generally not equal for $k > r$.

4.5 Examples

We compare the performance of the SVD-based rank-q square-root filter and the Cholesky-based rank-q square-root filter with the Kalman filter for two linear time-invariant systems.

4.5.1 Compartmental Model

A schematic diagram of the compartmental model [49] is shown in Figure 4.1. The n compartments or subsystems exchange energy through mutual interaction. Applying conservation of energy yields, for $i = 1, \ldots, n$,

$$x_{i,k+1} = x_{i,k} - \beta x_{i,k} - \alpha \left(x_{i+1,k} - x_{i,k} \right) - \alpha \left(x_{i,k} - x_{i-1,k} \right) + g_i w_{i,k}, \tag{4.5.1}$$

where $x_{i,k}$ is the energy in the i-th compartment, $w_{i,k}$ is the external disturbance affecting the i-th compartment, $0 < \beta < 1$ is the loss coefficient, and $0 < \alpha < 1$ is the flow coefficient. It follows from (4.5.1) that

$$x_{k+1} = Ax_k + Gw_k, \tag{4.5.2}$$

where

$$x_k \triangleq \left[\begin{array}{ccc} x_{1,k} & \cdots & x_{n,k} \end{array} \right]^{\mathrm{T}}, \quad w_k \triangleq \left[\begin{array}{ccc} w_{1,k} & \cdots & w_{n,k} \end{array} \right]^{\mathrm{T}}, \tag{4.5.3}$$

and $A \in \mathbb{R}^{n\times n}$ and $G \in \mathbb{R}^{n\times n}$ are defined by

$$A \triangleq \begin{bmatrix} 1-\beta-\alpha & \alpha & 0 & 0 & \cdots & 0 \\ \alpha & 1-\beta-2\alpha & \alpha & 0 & \cdots & 0 \\ 0 & \alpha & 1-\beta-2\alpha & \alpha & \cdots & 0 \\ \vdots & & \ddots & & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \alpha & 1-\beta-\alpha \end{bmatrix}, \tag{4.5.4}$$

and

$$G \triangleq \operatorname{diag}(g_1, \ldots, g_n). \tag{4.5.5}$$

Let $n = 20$, $\alpha = 0.35$ and $\beta = 0.5$. We assume that the disturbance w_k affects all of the compartments so that $g_i \neq 0$ for $i = 1, \ldots, n$, and hence $Q = GG^{\mathrm{T}}$ has full rank. The external disturbance w_k is modeled as a white-noise process with unit covariance. Finally, we use measurements of the energy in the 10th and 11th compartments to estimate the energy in all of the compartments, that is,

$$y_k = \begin{bmatrix} x_{10,k} & x_{11,k} \end{bmatrix}^{\mathrm{T}} + v_k. \tag{4.5.6}$$

To evaluate the performance of the SVD-based and Cholesky-based reduced-rank square-root filters, we compare the costs J_k, $J_{\mathrm{s},k}$ and $J_{\mathrm{c},k}$, where

$$J_k \triangleq \operatorname{tr}(P_k^{\mathrm{f}}), \quad J_{\mathrm{s},k} = \operatorname{tr}(P_{\mathrm{s},k}^{\mathrm{f}}), \quad J_{\mathrm{c},k} = \operatorname{tr}(P_{\mathrm{c},k}^{\mathrm{f}}). \tag{4.5.7}$$

Recall that $P_{\mathrm{s},k}^{\mathrm{f}}$ and $P_{\mathrm{c},k}^{\mathrm{f}}$, which are the true error covariances when the reduced-rank square-root filters are used, are given by (4.3.15)-(4.3.16) and (4.4.22)-(4.4.23), respectively. In all cases, we initialize the three filters with $x_0^{\mathrm{f}} = x_{\mathrm{c},0}^{\mathrm{f}} = x_{\mathrm{s},0}^{\mathrm{f}} = 0$ and $P_0^{\mathrm{f}} = \tilde{P}_{\mathrm{c},0}^{\mathrm{f}} = \tilde{P}_{\mathrm{s},0}^{\mathrm{f}} = I_{20}$.

We compare the performance of the SVD-based and Cholesky-based filters for $q = 2, 5, 10$. The steady-state performance $\lim_{k\to\infty} J_{\mathrm{s},k}$ and $\lim_{k\to\infty} J_{\mathrm{c},k}$ of the

SVD-based rank-q square-root filter and the Cholesky-based rank-q square-root filter, respectively, is shown in Figure 4.2. Figure 4.3 shows the performance of the SVD-based reduced-rank square-root filter $J_{\mathrm{s},k}$ and the Cholesky-based reduced-rank square-root filter $J_{\mathrm{c},k}$, when $q = 2$ in both cases. The cost J_k of the Kalman filter is also plotted for comparison. Finally, we plot $J_{\mathrm{c},k}/J_k$ and $J_{\mathrm{s},k}/J_k$ when $q = 10$. Note that $p = 2$, and hence, $r = 5$ satisfies $q = pr$. Therefore, it follows from Proposition 4.4.2 that the Cholesky-based rank-q square-root filter is equivalent to the Kalman filter for $k = 0, \ldots, 5$, as confirmed by Figure 4.4. In fact, the performance of the Cholesky-based reduced-rank square-root filter with $q = 10$ is indistinguishable from the performance of the Kalman filter for all $k = 0, \ldots, 10$.

4.5.2 N-mass system

Next, we consider the mass-spring-damper model shown in Figure 4.5. The number of masses is 10 with two states (displacement and velocity) per mass so that $n = 20$. For $i = 1, \ldots, 10$, $m_i = 1$ kg, while $k_j = 1$ N/m and $c_j = 0.2$ N-s/m for $j = 1, \ldots, 11$. We assume that an external force $w_{i,k}$ acts on the mass m_i, where $w_{i,k}$ is a white-noise process with unit covariance so that

$$x_{k+1} = Ax_k + w_k, \tag{4.5.8}$$

where

$$x \triangleq \begin{bmatrix} q_1 & \dot{q}_1 & \cdots & q_{10} & \dot{q}_{10} \end{bmatrix}^{\mathrm{T}}, \quad w \triangleq \begin{bmatrix} w_1 & \cdots & w_{10} \end{bmatrix}^{\mathrm{T}}, \tag{4.5.9}$$

and $A \in \mathbb{R}^{20\times 20}$ is obtained using a zero-order-hold discretization of the continuous-time dynamics. We assume that the displacement of the 5th mass is measured so that,

$$y_k = q_{5,k} + v_k, \tag{4.5.10}$$

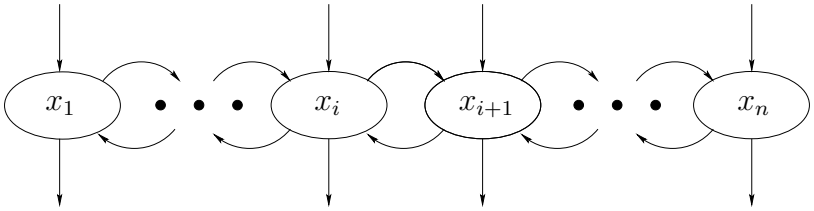
where v_k is white-noise process with unit covariance. Again, we initialize the Kalman filter and the reduced-rank square-root filters with $x_0^{\mathrm{f}} = x_{\mathrm{c},0}^{\mathrm{f}} = x_{\mathrm{s},0}^{\mathrm{f}} = 0$ and $P_0^{\mathrm{f}} = \tilde{P}_{\mathrm{c},0}^{\mathrm{f}} = \tilde{P}_{\mathrm{s},0}^{\mathrm{f}} = I_{20}$.

We compare the performance of the reduced-rank square-root filters for $q = 4$ and $q = 8$. The mean-square-error (MSE) in the estimates of the position of the masses is shown in Figure 4.6. It can be seen that, for a specific choice of q, the performance of the Cholesky-based rank-q square-root filter is better than the performance of the SVD-based rank-q square-root filter. The MSE in the estimates of the velocities of the masses is shown in Figure 4.7. The performance of the Kalman filter is plotted for comparison. Finally, we plot the ratio $J_{\mathrm{c},k}/J_k$, where J_k and $J_{\mathrm{c},k}$ are defined in (4.5.7), for the case $q = 4$. It can be seen from Figure 4.9 that, in accordance with Proposition 4.4.2, the Cholesky-based rank-q square-root filter is equivalent to the Kalman filter for $k = 0, \ldots, r = q = 4$ because $p = 1$.

4.6 Conclusions

We developed a reduced-rank square-root Kalman filter based on the Cholesky factorization. We presented conditions under which the SVD-based reduced-rank square-root Kalman filter and the Cholesky-based reduced-rank square-root Kalman filter are equivalent to the Kalman filter. In general, neither the Cholesky-based nor the SVD-based reduced-rank square-root filter consistently outperforms the other. However, in this chapter, we presented two examples where the Cholesky-based reduced-rank square-root filter performs better than the SVD-based reduced-rank square-root filter. Since the Cholesky factorization is a computationally efficient algorithm compared to the singular value decomposition, the Cholesky-based reduced-rank square-root filter provides a computationally efficient alternative method for

reduced-rank square-root filtering. In chapters II-IV, we considered reduced-complexity algorithms for state estimation of linear systems. In the next chapter, we compare two algorithms for state estimation of nonlinear systems.



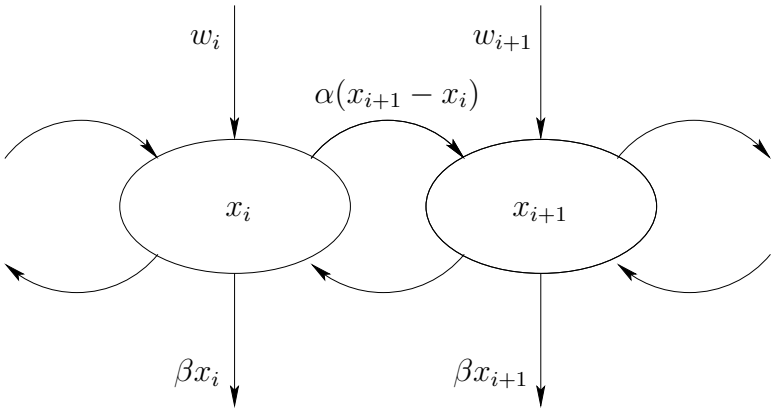


Figure 4.1: Compartmental model where energy is exchanged between neighboring compartments

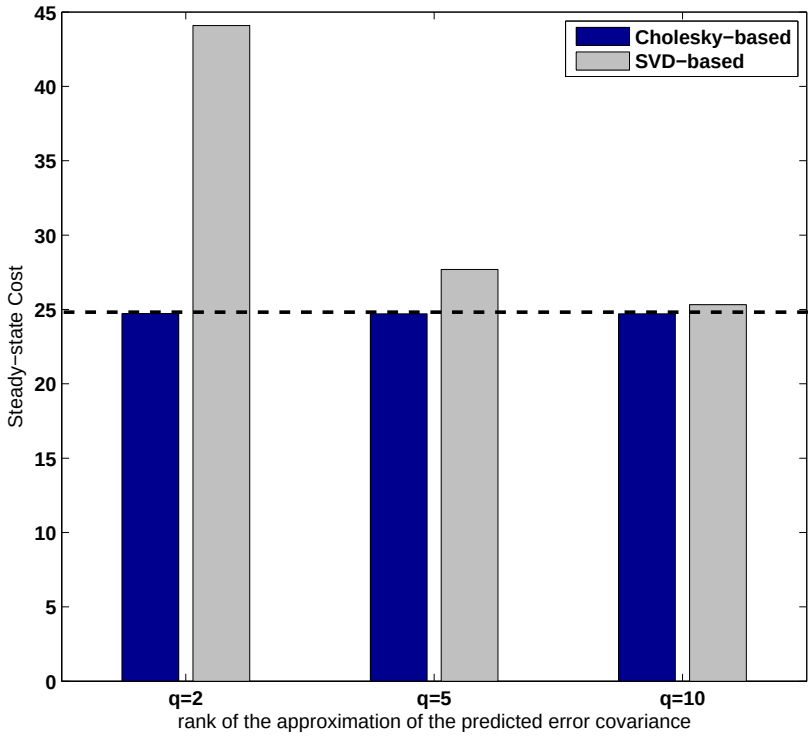


Figure 4.2: Steady-state performance of the SVD-based and Cholesky-based reduced-rank square-root filters for $q = 2, 5, 10$. As q increases, the performance of the reduced-rank square-root filters improves. Moreover, note that $n = 20$ and even when $q = 2$, the performance of the Cholesky-based reduced-rank square-root filter is similar to that of the Kalman filter. The steady-state performance of the Kalman filter is shown as the dashed line for comparison.

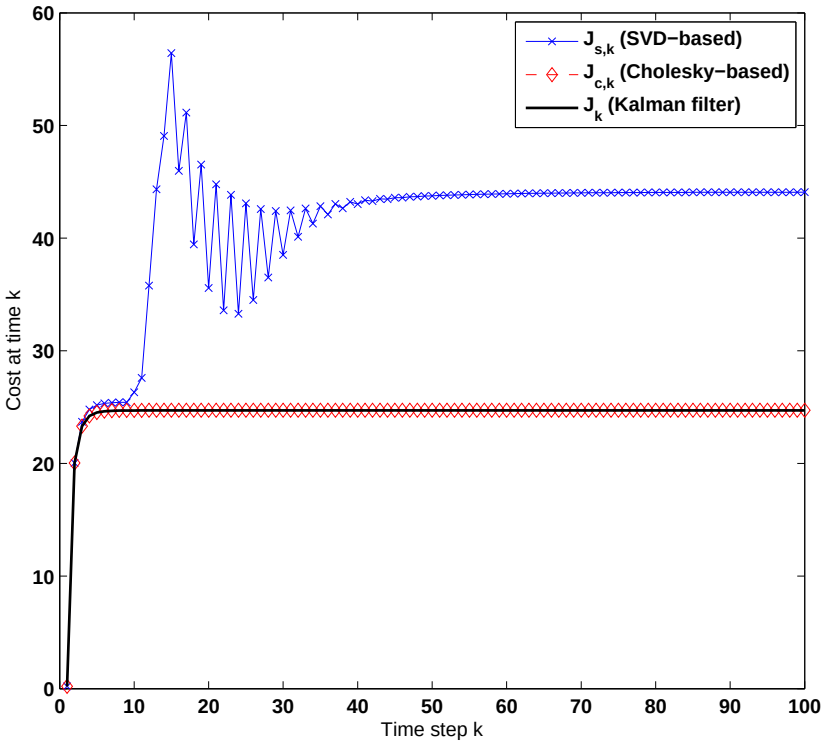


Figure 4.3: The costs $J_{s,k}$ and $J_{c,k}$ of the SVD-based and Cholesky-based reduced-rank square-root filters, respectively, with $q = 2$. The performance of the Cholesky-based rank-q square-root filter is close to that of the Kalman filter. However, the performance of the SVD-based filter is poor.

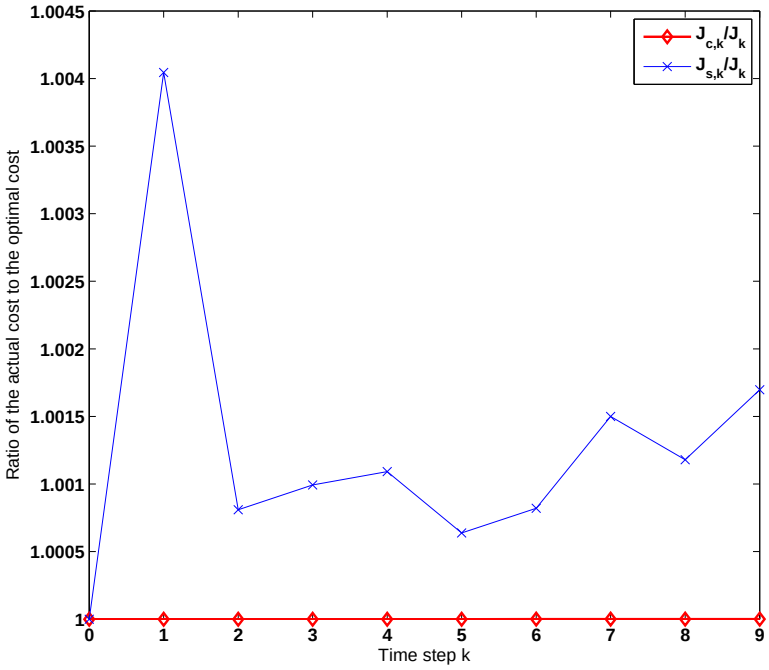


Figure 4.4: Ratio of the costs $J_{\mathrm{s},k}$ and $J_{\mathrm{c},k}$ of the reduced-rank filters with $q = 10$ and the Kalman filter. The Cholesky-based rank-q square-root filter is equivalent to the Kalman filter for $k = 0, \ldots, r = 5$. In fact, the performance of the Cholesky-based rank-q square-root filter is close to the performance of the Kalman filter for $k = 0, \ldots, 10$.

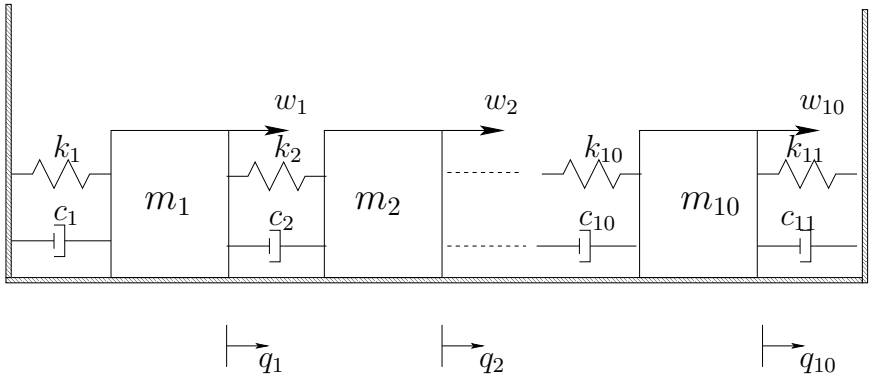


Figure 4.5: Mass-spring-dashpot system

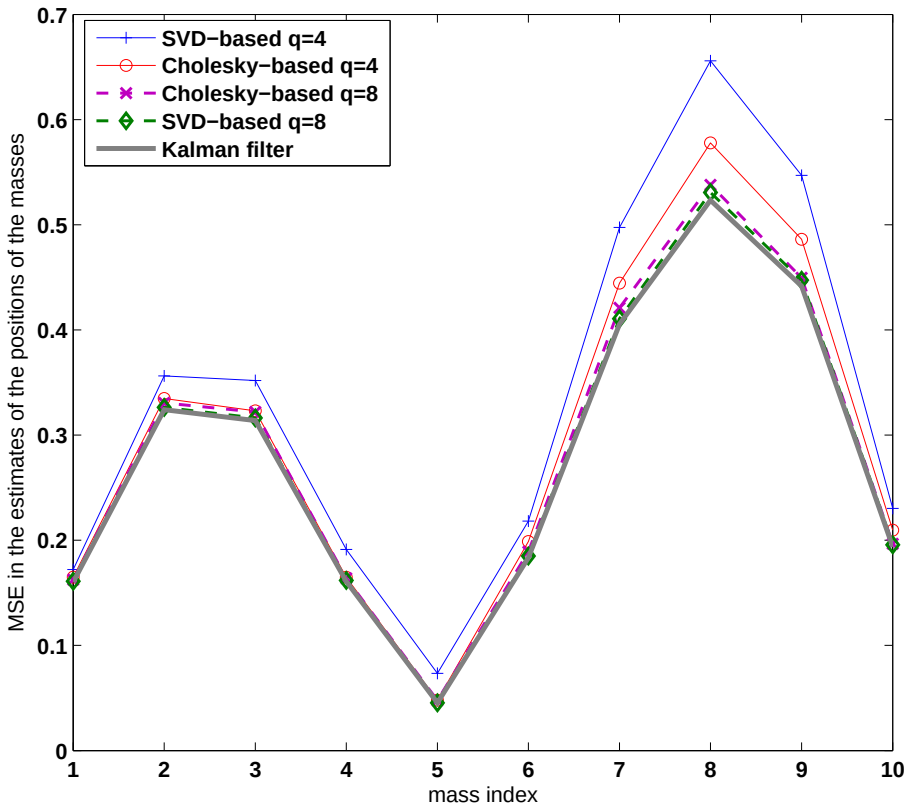


Figure 4.6: Steady-state MSE in the estimates of the positions of the masses $m_1, \ldots, m_{10}$ using the Cholesky-based and SVD-based reduced-rank square-root filters for $q = 4$ and $q = 8$ when $k \to \infty$. The performance of the reduced-rank square-root filters improves as q increases, while, for $q = n$, both reduced-rank square-root filters are equivalent to the Kalman filter.

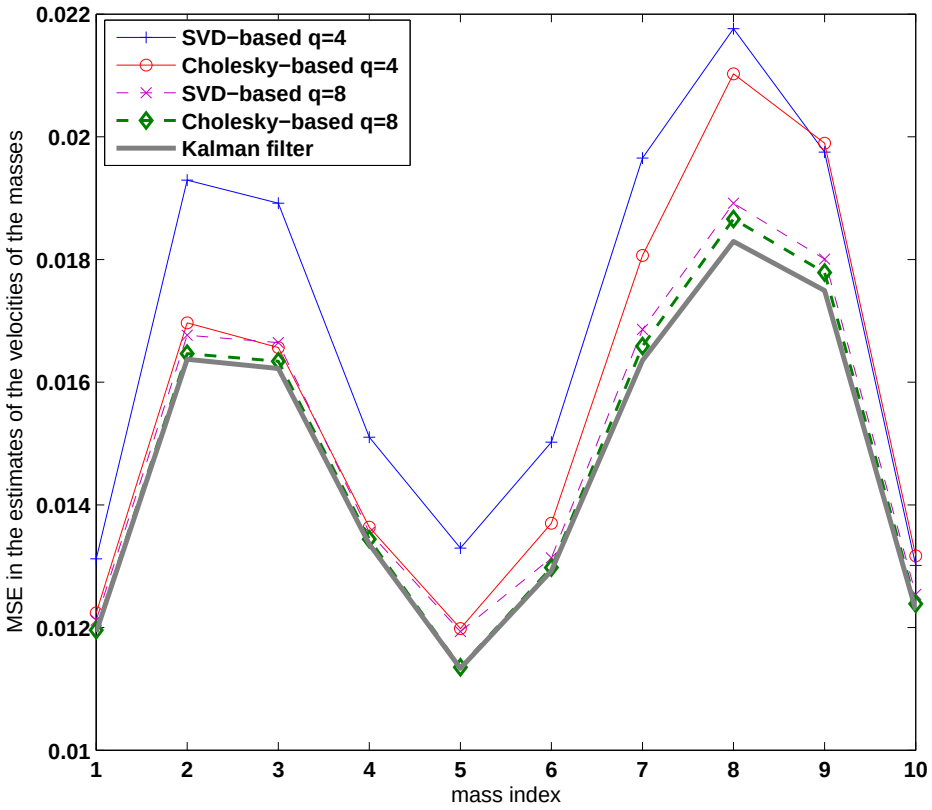


Figure 4.7: Steady-state MSE in the estimates of the velocities of the masses $m_1, \ldots, m_{10}$ using the Cholesky-based and SVD-based reduced-rank square-root filters with $q = 4$ and $q = 8$.

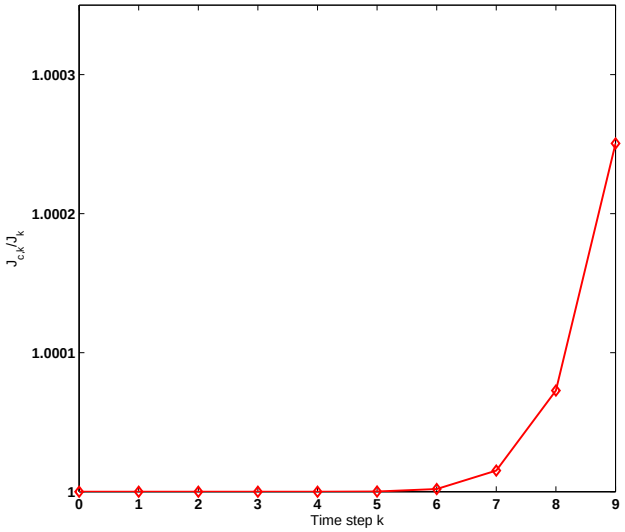


Figure 4.8: Ratio of the cost $J_{\mathrm{c},k}$ of the Cholesky-based reduced-rank filter with $q = 4$ to the cost J_k of the Kalman filter. Since the Cholesky-based rank-q square-root filter is equivalent to the Kalman filter for $k = 0, \ldots, 4$, the ratio is equal to 1 at these time steps.

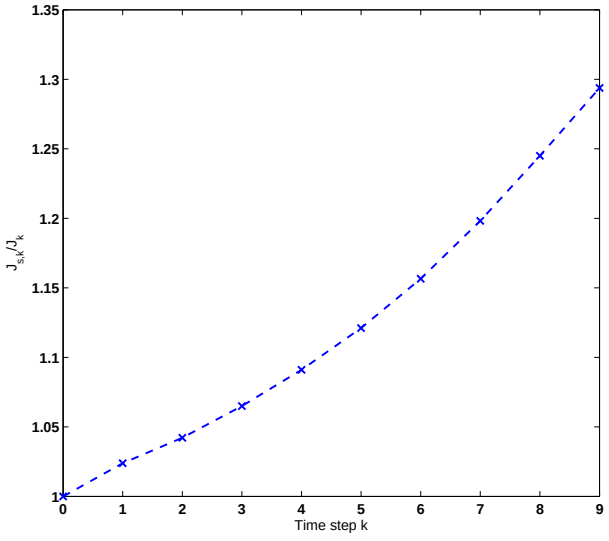


Figure 4.9: Ratio of the cost $J_{\mathrm{s},k}$ of the SVD-based reduced-rank filter with $q = 4$ to the cost J_k of the Kalman filter. The performance of the SVD-based reduced-rank square-root filter is inferior to the performance of the Kalman filter for all $k > 0$.

CHAPTER V

A Comparison of the Extended and Unscented Kalman Filters for Discrete-Time Systems with Nondifferentiable Dynamics

In this chapter, we consider state estimation of discrete-time nonlinear systems with nondifferentiable dynamics. Due to the presence of nonlinear dynamics, designing optimal estimators is difficult and hence we use suboptimal algorithms for state estimation. Specifically, we compare the performances of the extended Kalman filter and unscented Kalman filter. The extended Kalman filter uses the Jacobian of the dynamics to propagate a pseudo-error covariance, whereas the unscented Kalman filter is a particle based filter that calculates a pseudo-error covariance from a collection of state estimates. Finally, we consider H_∞ filter based extensions of the extended Kalman filter and unscented Kalman filter. The results presented in this chapter are given in [50].

5.1 Introduction

Because of the widespread need for nonlinear observers and estimators, this area of research remains one of the most active [51–53]. One of the main drivers of research in this area is applications to distributed, large scale systems, the most visible of which is weather forecasting [38, 54, 55]. This area is often referred to as

data assimilation.

The classical Kalman filter for linear systems is often applied to nonlinear systems in the form of the extended Kalman filter (XKF) [14, 56]. In the XKF, the state is propagated using the nonlinear dynamics, while the pseudo-covariance is propagated using the Jacobians of the dynamics and measurement maps. We use the phrase "pseudo-covariance" to stress the fact that the error covariance matrix in the linear case is generally not the covariance of the error in the nonlinear case. The XKF can be implemented in either the one-step or two-step forms, where the latter form involves a physics update followed by a data-assimilation step.

A variation of the XKF is the state-dependent Riccati equation (SDRE) approach, in which, in place of the Jacobians, the dynamics and output map are exactly factored, and the factors are used for the pseudo-covariance update [15, 57]. This approach has been studied by solving the algebraic Riccati equation and by updating the pseudo-covariance. An interesting aspect of the SDRE approach is the fact that, in the non-scalar case, the factorizations are not unique, while guidelines for selecting advantageous factorizations have not been developed. Our own numerical experiments suggest that the best SDRE factorizations are close to the Jacobian, suggesting that the SDRE filter might have limited advantages, if any, over the XKF. In our opinion, advantages of the SDRE over the XKF have not been definitively demonstrated.

Another approach to state estimation of linear systems are the H_∞ filters [58]. Unlike the classical Kalman filter, these filters do not require the stringent Gaussian distribution assumption of the process and sensor noise affecting the system and guarantee a performance bound. Estimation with uncertainty in the model has also been performed using the H_∞ filter [59]. We apply the H_∞ filter to nonlinear systems

by using the Jacobians of the dynamics and measurement maps and call the resulting filter the extended H_∞ filter (XHF).

Yet another approach to nonlinear estimation involves particle filters. Here the idea is to propagate a collection of state estimates from which statistics can be computed. Among the various techniques that have been developed are the unscented Kalman filter (UKF) [18, 19, 60], which deterministically constructs the collection of state estimates, as well as the ensemble Kalman filter (EnKF) [61, 62], which uses a stochastic construction. Although particle filters do not require the propagation of a covariance (or pseudo-covariance) in the usual (Riccati) way, the size of the collection determines the computational requirements [63]. Finally, we combine the H_∞-filter gain expression with the particle filter framework to obtain the unscented H_∞ filter (UHF).

This chapter focuses on discrete-time systems with dynamics that are not differentiable. The main motivation is state estimation based on computational fluid dynamics (CFD) models for space weather forecasting [64, 65]. In particular we focus on CFD models for hydrodynamics (HD) and magnetohydrodynamics (MHD) in which the equations of fluid motion are approximated by finite volume schemes. In [57, 63] we have considered SDRE and XKF methods for state estimation.

In HD and MHD, the CFD models involve nondifferentiable functions as part of the discretization of the underlying partial differential equations [66, 67]. Consequently, to avoid the need for the Jacobian, we developed SDRE filters for 1-dimensional HD in [57]. In the present chapter, we consider an alternative approach in which we apply XKF and XHF despite the lack of differentiability. In particular, we compute the Jacobian at all points at which it exists, and we employ an averaged value at points at which the dynamics are not differentiable.

To demonstrate the accuracy of XKF, XHF, UKF, and UHF when the dynamics are not differentiable, we consider several examples. For each example, we compare the performance of XKF, XHF, UKF, and UHF.

5.2 The H_∞ Filter

Consider the discrete-time linear time-invariant system with dynamics

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{5.2.1}$$

and measurements

$$y_k = Cx_k + v_k, \tag{5.2.2}$$

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 is assumed to be measured, and $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^p$ are unknown process and measurement noise, respectively.

Consider the cost function

$$J(K_k) = \frac{\sum_{i=0}^{N}(x_i - x_i^{\mathrm{f}})^{\mathrm{T}} M (x_i - x_i^{\mathrm{f}})}{(x_0 - x_0^{\mathrm{f}})^{\mathrm{T}} P_0^{\mathrm{f}} (x_1 - x_1^{\mathrm{f}}) + \sum_{i=0}^{N} w_i^{\mathrm{T}} Q w_i + \sum_{i=0}^{N} v_i^{\mathrm{T}} R v_i}. \tag{5.2.3}$$

The H_∞ filter ensures that inspite of the worst possible process and sensor noise, the cost $J(K_k)$ satisfies

$$J(K_k) \leqslant \frac{1}{\gamma}. \tag{5.2.4}$$

The data assimilation step of the robust H_∞ filter is given by

$$x_k^{\mathrm{da}} = x_k^{\mathrm{f}} + K_k(y_k - y_k^{\mathrm{f}}), \tag{5.2.5}$$

$$y_k^{\mathrm{f}} = Cx_k^{\mathrm{f}}, \tag{5.2.6}$$

$$P_k^{\mathrm{da}} = (I - K_k C)\tilde{P}_k^{\mathrm{f}}(I - K_k C)^{\mathrm{T}} + K_k R K_k^{\mathrm{T}}, \tag{5.2.7}$$

where

$$K_k = \tilde{P}_k^{\mathrm{f}} C^{\mathrm{T}} (C \tilde{P}_k^{\mathrm{f}} C^{\mathrm{T}} + R)^{-1} \tag{5.2.8}$$

and

$$\tilde{P}_k^{\mathrm{f}} \triangleq P_k^{\mathrm{f}} (I - \gamma M P_k^{\mathrm{f}})^{-1}. \tag{5.2.9}$$

The forecast step of the H_∞ filter is given by

$$x_{k+1}^{\mathrm{f}} = A x_k^{\mathrm{da}}, \tag{5.2.10}$$

$$P_{k+1}^{\mathrm{f}} = A P_k^{\mathrm{da}} A^{\mathrm{T}} + Q. \tag{5.2.11}$$

Note that unlike the Kalman filter, w_k and v_k may not be white noise processes and hence Q and R are not their covariances, but a weighting on the uncertainty associated with the process and sensor noise. Moreover, P_k^{f} and P_k^{da} in (5.2.5)-(5.2.11) are not the error covariances. Hence, although the Kalman filter provides optimal estimates when the process and sensor noise are white-processes, the H_∞ filter guarantees a certain performance bound irrespective of the magnitude of the process and sensor noise encountered.

5.3 The Extended Kalman Filter

Next, we consider the discrete-time nonlinear system with dynamics

$$x_{k+1} = f(x_k, u_k, k) + w_k \tag{5.3.1}$$

and measurements

$$y_k = h(x_k, k) + v_k, \tag{5.3.2}$$

where $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^p$ are unknown process and measurement noise with covariance Q_k and R_k, respectively. Furthermore, we assume that R_k is positive

definite. Since the dynamics and measurements are nonlinear functions of the state, the discrete-time Riccati equation cannot be used to propagate the error covariance P_k. We thus consider the extended Kalman filter (XKF) for estimating x_k in (5.3.1) using measurements (5.3.2). The two-step XKF is given by

$$x_{k+1}^{\mathrm{f}} = f(x_k^{\mathrm{da}}, u_k, k), \tag{5.3.3}$$

$$x_k^{\mathrm{da}} = x_k^{\mathrm{f}} + K_k(y_k - y_k^{\mathrm{f}}), \tag{5.3.4}$$

$$y_k^{\mathrm{f}} = h(x_k^{\mathrm{f}}, k), \tag{5.3.5}$$

where K_k, P_k^{da} and P_k^{f} are given by (4.2.3), (4.2.4) and (4.2.7), respectively, with

$$A_k \triangleq \frac{\partial f(x,u,k)}{\partial x}\Big|_{x=x_k^{\mathrm{da}},u=u_k}, \quad C_k \triangleq \frac{\partial h(x,k)}{\partial x}\Big|_{x=x_k^{\mathrm{da}}}. \tag{5.3.6}$$

A one-step version of the XKF exists and note that the one-step and the two-step extended Kalman filters are not necessarily equivalent.

If $f(x,u,k)$ and $h(x,k)$ are not differentiable with respect to x, the two-step XKF (5.3.3)-(5.3.6) cannot be used to obtain an estimate of the state x_k because A_k and C_k defined in (5.3.6) may not exist for all x_k^{da}. However, we assume that the first order symmetric partial derivatives [68, 69] of $f(x,u,k)$ and $h(x,k)$ exist everywhere, that is, for all $x \in \mathbb{R}^n$,

$$\frac{\partial_{\mathrm{s}} f(\xi,u,k)}{\partial_{\mathrm{s}}\xi_i}\Big|_{\xi=x} \triangleq \lim_{\delta\to 0} \frac{f(x+\delta e_i,u,k) - f(x-\delta e_i,u,k)}{2\delta} \tag{5.3.7}$$

and

$$\frac{\partial_{\mathrm{s}} h(\xi,k)}{\partial_{\mathrm{s}}\xi_i}\Big|_{\xi=x} \triangleq \lim_{\delta\to 0} \frac{h(x+\delta e_i,k) - h(x-\delta e_i,k)}{2\delta} \tag{5.3.8}$$

exist, where $\xi \in \mathbb{R}^n$ has scalar entries $\xi = \begin{bmatrix} \xi_1 & \cdots & \xi_n \end{bmatrix}^{\mathrm{T}}$ and $e_i \in \mathbb{R}^n$ is the ith column of the $n \times n$ identity matrix. Hence, for example, although $f(x) = |x|$ does

not have a derivative at $x = 0$, it follows from (5.3.7) that $\frac{\partial_{\mathrm{s}} f}{\partial_{\mathrm{s}} x}(0) = 0$. Furthermore, if $g : \mathbb{R}^n \to \mathbb{R}$ is a differentiable function, then the symmetric partial derivative and the partial derivative are equal, that is, for all $x \in \mathbb{R}^n$,

$$\frac{\partial_{\mathrm{s}} g(\xi)}{\partial_{\mathrm{s}} \xi_i}\Big|_{\xi=x} = \frac{\partial g(\xi)}{\partial \xi_i}\Big|_{\xi=x}. \tag{5.3.9}$$

It follows from the symmetry of (5.3.7) that the one-sided limits are equivalent. Specifically,

$$\lim_{\delta \uparrow 0} \frac{g(x+\delta) - g(x-\delta)}{2\delta} = \lim_{\delta \downarrow 0} \frac{g(x+\delta) - g(x-\delta)}{2\delta}. \tag{5.3.10}$$

Moreover, the symmetric derivative is the average of the left and right directional derivatives.

Next, we define the (i, j) entry of the averaged Jacobian $F_{\mathrm{s}}(x, u, k) \in \mathbb{R}^{n \times n}$ and $H_{\mathrm{s}}(x, k) \in \mathbb{R}^{p \times n}$ of $f(\cdot)$ and $h(\cdot)$, respectively, by

$$F_{\mathrm{s},i,j}(x, u, k) \triangleq \frac{\partial_{\mathrm{s}} f_i(\xi, u, k)}{\partial_{\mathrm{s}} \xi_j}\Big|_{\xi=x}, H_{\mathrm{s},i,j}(x, k) \triangleq \frac{\partial_{\mathrm{s}} h_i(\xi, k)}{\partial_{\mathrm{s}} \xi_j}\Big|_{\xi=x}, \tag{5.3.11}$$

where $f_i(x, u, k)$ and $h_i(x, k)$ are the scalar entries of $f(x, u, k) \in \mathbb{R}^n$ and $h(x, k) \in \mathbb{R}^p$, respectively. It follows from (5.3.9) that if $f(\cdot)$ and $h(\cdot)$ are differentiable, then, for all $x \in \mathbb{R}^n$, the averaged Jacobians F_{s} and H_{s} are equal to the true Jacobians. Hence, the two-step XKF for (5.3.1)-(5.3.2) when $f(\cdot)$ and $h(\cdot)$ satisfy (5.3.7) and (5.3.8) is given by (5.3.3), where K_k, P_k^{da} and P_k^{f} are given by (4.2.3), (4.2.4) and (4.2.7), respectively, with

$$A_k = F_{\mathrm{s}}(x_k^{\mathrm{da}}, u_k, k), \quad C_k = H_{\mathrm{s}}(x_k^{\mathrm{da}}, k). \tag{5.3.12}$$

5.4 The Extended H_∞ Filter

An alternative approach to state estimation of (5.3.1)-(5.3.2) is based on the H_∞ filter. Although, the H-infitiy filter is derived for linear time-invariant systems,

like the extended Kalman filter, the Jacobian of the dynamics and measurements maps can be used in the filter equations. However, the performance bounds guaranteed in the linear case are not valid anymore.

The extended H_∞ filter is given by (5.3.3)-(5.3.5), where K_k, P_k^{da} and P_k^{f} are given by (5.2.8), (5.2.7), and (5.2.11), with A and C replaced by A_k and C_k, respectively, where A_k and C_k are defined in (5.3.12). Note that since the Jacobians are based on the symmetric derivatives, the extended H_∞ filter that uses the averaged Jacobians can be used on nonlinear systems with nondifferentiable dynamics. Finally, we use γ, Q and R in the H_∞ filter as tuning parameters to improve the estimates. Note that XHF may not be stable for all values of γ and hence the value of γ must be tuned carefully.

5.5 The Unscented Kalman Filter

Another approach to state estimation of nonlinear systems is the unscented Kalman filter (UKF). Unlike the XKF and SDRE estimator, the 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 the UKF is a set of sample points, that is, a collection of state estimates that capture the initial probability distribution of the state [18, 19].

Let $x \in \mathbb{R}^n$, and let $P \in \mathbb{R}^{n\times n}$ be positive semidefinite. The unscented transformation provides $2n+1$ ensembles $X_i \in \mathbb{R}^n$ and corresponding weights $\gamma_{x,i}$ and $\gamma_{P,i}$, for $0 = 1, \ldots, 2n$, such that the weighted mean and weighted variance of the ensembles are x and P, respectively. Specifically, let $S \in \mathbb{R}^{n\times n}$ satisfy

$$SS^{\mathrm{T}} = P, \tag{5.5.1}$$

and, for all $i = 1, \ldots, n$, let S_i denote the ith column of S. For $\alpha > 0$, the unscented

transformation $X = \Psi(x, S, \alpha) \in \mathbb{R}^{n\times(2n+1)}$ of x with covariance $P = SS^{\mathrm{T}}$ is defined by

$$X \triangleq \begin{bmatrix} X_0 & \cdots & X_{2n} \end{bmatrix}, \tag{5.5.2}$$

where

$$X_i = \begin{cases} x, & i = 0, \\ x + \sqrt{\alpha}S_i, & i = 1, \ldots, n, \\ x - \sqrt{\alpha}S_{i-n}, & i = n+1, \ldots, 2n. \end{cases} \tag{5.5.3}$$

The parameter α determines the spread of the ensembles around x. Next, define the weights $\gamma_i \in \mathbb{R}$ by

$$\gamma_0 \triangleq \frac{\alpha - n}{\alpha}, \quad \gamma_i \triangleq \frac{1}{2\alpha}, \quad i = 1, \ldots, 2n \tag{5.5.4}$$

Then,

$$\sum_{i=0}^{2n} \gamma_i X_i = x, \quad \sum_{i=0}^{2n} \gamma_i (X_i - x)(X_i - x)^{\mathrm{T}} = P. \tag{5.5.5}$$

Note that the unscented transformation described above is the scaled unscented transformation given in [70] and ensures that the distance between the sample point X_i and x does not increase as n increases.

UKF uses the unscented transformation to approximate the error covariance and estimate the state x_k. Letting x_0^{f} be an initial estimate of x_0 with error covariance P_0^{f}, the data assimilation step of UKF is given by

$$x_k^{\mathrm{da}} = x_k^{\mathrm{f}} + K_k(y_k - y_k^{\mathrm{f}}), \tag{5.5.6}$$

$$y_k^{\mathrm{f}} = C_k x_k^{\mathrm{f}}, \tag{5.5.7}$$

$$X_k^{\mathrm{da}} = \Psi(x_k^{\mathrm{da}}, S_k^{\mathrm{da}}, \alpha), \tag{5.5.8}$$

$$P_k^{\mathrm{da}} = P_k^{\mathrm{f}} - K_k P_{yy,k} K_k^{\mathrm{T}}, \tag{5.5.9}$$

where

$$K_k = P_{xy,k} P_{yy,k}^{-1}, \tag{5.5.10}$$

$$P_{xy,k} = \sum_{i=0}^{2n} \gamma_i (X_{i,k}^{\mathrm{f}} - x_k^{\mathrm{f}})(Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})^{\mathrm{T}}, \tag{5.5.11}$$

$$P_{yy,k} = \sum_{i=0}^{2n} \gamma_i (Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})(Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})^{\mathrm{T}} + R_k, \tag{5.5.12}$$

$$Y_{i,k}^{\mathrm{f}} = h(X_{i,k}^{\mathrm{f}}, k), \quad i = 0, \ldots, 2n \tag{5.5.13}$$

and $S_k^{\mathrm{da}} \in \mathbb{R}^{n \times n}$ satisfies

$$S_k^{\mathrm{da}} (S_k^{\mathrm{da}})^{\mathrm{T}} = P_k^{\mathrm{da}}. \tag{5.5.14}$$

The forecast step of UKF is given by

$$X_{i,k+1}^{\mathrm{f}} = f(X_{i,k}^{\mathrm{da}}, u_k, k), \quad i = 0, \ldots, 2n, \tag{5.5.15}$$

$$x_{k+1}^{\mathrm{f}} = \sum_{i=0}^{2n} \gamma_i X_{i,k+1}^{\mathrm{f}}, \tag{5.5.16}$$

$$P_{k+1}^{\mathrm{f}} = \sum_{i=0}^{2n} \gamma_i (X_{i,k+1}^{\mathrm{f}} - x_{k+1}^{\mathrm{f}})(X_{i,k+1}^{\mathrm{f}} - x_{k+1}^{\mathrm{f}})^{\mathrm{T}} + Q_k. \tag{5.5.17}$$

When the dynamics in (5.3.1) are linear, UKF is equivalent to the Kalman filter [19]. Furthermore, in the linear case, P_k^{da} and P_k^{f} are the covariances of the error $x_k - x_k^{\mathrm{da}}$ and $x_k - x_k^{\mathrm{f}}$, respectively. However, in the nonlinear case, P_k^{da} and P_k^{f} are pseudo-error covariances. The case when the process noise w_k in (5.3.1) does not enter linearly is discussed in [71]. However, since we assume that the process noise affects the system affinely, we use the covariance Q_k of w_k in (5.5.17) to account for uncertainty in the state estimates.

At every time step k, the ensemble X_k^{da} is constructed in (5.5.8) using the unscented transformation based on a square root S_k^{da} of P_k^{da} satisfying (5.5.14). However, S_k^{da} satisfying (5.5.14) is not unique. For example, the singular value decomposition or the Cholesky factorization can be used to obtain a square root of the

pseudo-error covariance P_k^{da}. Moreover, if $S_k^{\mathrm{da}} = \hat{S}_k^{\mathrm{da}}$ satisfies (5.5.14), then, for any orthogonal matrix $U \in \mathbb{R}^{n\times n}$, $S_k^{\mathrm{da}} = \hat{S}_k^{\mathrm{da}} U$ also satisfies (5.5.14). For linear dynamics, UKF is equivalent to the Kalman filter, and the performance of UKF does not depend on the choice of S_k^{da}. However, for nonlinear dynamics, the performance of UKF depends on the choice of S_k^{da}, although simulation results indicate that the performance of UKF is similar for different choices of S_k^{da}.

Since the UKF involves $2n + 1$ model update, the computational burden of the UKF is of the order $(2n + 1)n^2 = 2n^3 + n^2$. On the other hand, the XKF involves a single model update and covariance propagation using the Riccati equation and hence the computational burden of the XKF is of the order $n^3 + n^2$. Hence, when n is large the computational burden of the UKF is approximately twice that of the XKF. The performance of the UKF and XKF are compared in [18, 19, 72].

5.6 The Unscented H_∞ Filter

Finally, we consider an extension of the UKF that is based on the H_∞ filter. The analysis step of the unscented H_∞ filter (UHF) is given by

$$x_k^{\mathrm{da}} = x_k^{\mathrm{f}} + K_k(y_k - y_k^{\mathrm{f}}), \tag{5.6.1}$$

$$y_k^{\mathrm{f}} = h(x_k^{\mathrm{f}}, k), \tag{5.6.2}$$

$$X_k^{\mathrm{da}} = \Psi(x_k^{\mathrm{da}}, P_k^{\mathrm{da}}, \lambda), \tag{5.6.3}$$

$$P_k^{\mathrm{da}} = \tilde{P}_k^{\mathrm{f}} - K_k \tilde{P}_{yy,k} K_k^{\mathrm{T}}, \tag{5.6.4}$$

where

$$K_k = \tilde{P}_{xy,k}\tilde{P}_{yy,k}^{-1}, \tag{5.6.5}$$

$$\tilde{P}_{xy,k} = \sum_{i=0}^{2n} \gamma_i (X_{i,k}^{\mathrm{f}} - x_k^{\mathrm{f}})(Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})^{\mathrm{T}}, \tag{5.6.6}$$

$$\tilde{P}_{yy,k} = \sum_{i=0}^{2n} \gamma_i (Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})(Y_{i,k}^{\mathrm{f}} - y_k^{\mathrm{f}})^{\mathrm{T}} + R_k, \tag{5.6.7}$$

$$Y_{i,k}^{\mathrm{f}} = h(X_{i,k}^{\mathrm{f}}, k), \tag{5.6.8}$$

and the forecast step of the unscented Kalman filter is given by

$$X_{i,k+1}^{\mathrm{f}} = f(X_{i,k}^{\mathrm{da}}, k), \tag{5.6.9}$$

$$x_{k+1}^{\mathrm{f}} = \sum_{i=0}^{2n} \gamma_i X_{i,k+1}^{\mathrm{f}}, \tag{5.6.10}$$

$$P_{k+1}^{\mathrm{f}} = \sum_{i=0}^{2n} \gamma_i (X_{i,k+1}^{\mathrm{f}} - x_{k+1}^{\mathrm{f}})(X_{i,k+1}^{\mathrm{f}} - x_{k+1}^{\mathrm{f}})^{\mathrm{T}} + Q_k, \tag{5.6.11}$$

$$\tilde{P}_{k+1}^{\mathrm{f}} = P_{k+1}^{\mathrm{f}}(I - \gamma M P_{k+1}^{\mathrm{f}})^{-1}. \tag{5.6.12}$$

Note that when the dynamics are linear, then the unscented H_∞ filter is equivalent to the H_∞ filter presented in Section 3. Note that P_k^{f} and P_k^{da} are not the error covariances and no performance bounds are guaranteed by UHF. Also, like XHF, although the parameter γ can be chosen so that the filter yields good estimates of the state x_k, stability of UHF is not guaranteed for all values of γ.

5.7 Examples

Next, we use the XKF, XHF, UKF, and UHF for state estimation of low-dimensional discrete-time systems with nondifferentiable nonlinearities. Specifically, we consider nonlinearities that are not differentiable but have symmetric derivatives everywhere. Hence, we use XKF and XHF with the averaged Jacobian and compare the performance of XKF and XHF with UKF and UHF.

5.7.1 Absolute Value Function

First, we consider nonlinearities that commonly occur in finite volume discretization of hyperbolic partial differential equations [66, 67]. For example, the absolute value function appears in the first-order upwind discretization of an advection equation [66]. Let $x \in \mathbb{R}^4$ and

$$\begin{aligned} x_{k+1} &= \mathrm{abs}(\sin(Mx_k)) + w_k, \\ y_k &= Cx_k + v_k, \end{aligned} \tag{5.7.1}$$

where $M \in \mathbb{R}^{4\times 4}$ and

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.7.2}$$

and w_k and v_k are zero-mean white processes with covariances $Q_k = 0.1I_4$ and $R_k = 0.01I_2$, respectively. Note that for all $x \in \mathbb{R}$,

$$\frac{\partial_{\mathrm{s}}\mathrm{abs}(\xi)}{\partial_{\mathrm{s}}\xi}\Big|_{\xi=x} = \begin{cases} 1, & \text{if } x > 0, \\ -1, & \text{if } x < 0, \\ 0, & \text{if } x = 0. \end{cases} \tag{5.7.3}$$

Hence, it follows from (5.3.11), (5.7.1) and (5.7.3) that for $i, j = 1, \ldots, n$, the (i, j) entry row of $F_{\mathrm{s}}(x)$ is given by

$$F_{\mathrm{s},i,j}(x) = \begin{cases} \cos(\mathrm{row}_i(M)x)M_{i,j}, & \text{if } \sin(\mathrm{row}_i(M)x) > 0, \\ -\cos(\mathrm{row}_i(M)x)M_{i,j}, & \text{if } \sin(\mathrm{row}_i(M)x) < 0, \\ 0_{1\times 4}, \text{if } \sin(\mathrm{row}_i(M)x) = 0, \end{cases} \tag{5.7.4}$$

and $H_{\mathrm{s}}(x) = C$.

Figure 5.1 shows a plot of $\mathrm{abs}(\sin(mx))$ and it can be seen that as m increases, the nonlinearities become more prominent, that is, the variation in the slope increases.

Next, we compare the state estimates from XKF, XHF, UKF, and UHF for various choices of M. The logarithm of the sum of the Euclidean norms of the errors in the state estimates for 50 different choices of M with $\mathrm{sprad}(M) = 0.5$ is shown in Figure 5.4. Note that although the performance of the estimators varies depending on the choice of M, numerical simulations suggest that the performance of XKF, XHF, UKF, and UHF is almost indistinguishable for all choices of M. The error in the state estimates when no data assimilation is performed, that is, $K_k = 0$ for $k \geqslant 0$ in XKF, is also plotted for comparison. Next, we compare the performance of all the estimators for 50 different choices of M with $\mathrm{sprad}(M) = 10$. The performance of XKF, XHF, UKF, and UHF is shown in Figure 5.5. It can be seen that, in the case of more severe nonlinearities, the performance of UKF and UHF is better than the performance of XKF and XHF. The values of γ in all the cases were chosen such that XHF and UHF are both stable for all the choices of M with a specified spectral radius. However, the performance of XHF and UHF is very similar to the performance of XKF and UKF, respectively.

5.7.2 Minmod Function

Next, we consider discrete-time systems involving the minmod function, which is used in second-order upwind finite volume schemes as a slope limiter to reduce the diffusion effects [67]. For $\alpha, \beta \in \mathbb{R}$, define

$$\mathrm{minmod}(\alpha, \beta) = \frac{1}{2}\left(\mathrm{sign}(\beta) + \mathrm{sign}(\beta)\right) \min\{|\alpha|, |\beta|\}, \tag{5.7.5}$$

see Figure 5.2. Let $x \in \mathbb{R}^{10}$ and

$$\begin{aligned} x_{k+1} &= \sin(Mx_k) + \mathrm{minmod}(M_{\mathrm{L}}x_k, M_{\mathrm{R}}x_k) + w_k, \\ y_k &= Cx_k + v_k. \end{aligned} \tag{5.7.6}$$

We choose $M \in \mathbb{R}^{10\times 10}$ so that $\text{sprad}(M) < 1$, and for $i, j = 1, \ldots, 10$, the (i, j) entry of $M_{\text{L}} \in \mathbb{R}^{10\times 10}$ is given by

$$(M_{\text{L}})_{i,i} = 1, \quad (M_{\text{L}})_{i,i-1} = -1, \tag{5.7.7}$$

$$(M_{\text{L}})_{i,j} = 0 \ \text{ if } \ j \notin \{i, i-1\}, \tag{5.7.8}$$

$M_{\text{R}} = -M_{\text{L}}^{\text{T}}$, and for all k, $C_k = C \in \mathbb{R}^{2\times 10}$ is chosen to be

$$C = \begin{bmatrix} 1 & 0_{1\times 9} \\ 0_{1\times 9} & 1 \end{bmatrix}. \tag{5.7.9}$$

We assume that w_k and v_k are zero-mean white processes with covariances $Q_k = Q = 0.1I_{10}$ and $R_k = R = 0.01I_2$, respectively. Note that for all $u, v \in \mathbb{R}$,

$$\frac{\partial_{\text{s}}}{\partial_{\text{s}}\alpha}\text{minmod}(\alpha, \beta)\Big|_{(u,v)} = \begin{cases} 0, \ \text{ if } \ uv < 0 \ \text{ or } \ u = v = 0, \\ 0, \ \text{ if } \ uv > 0 \ \text{ and } \ |u| > |v|, \\ 0, \ \text{ if } \ u \neq 0, \ \ v = 0, \\ 0.5, \ \text{ if } \ uv > 0 \ \text{ and } \ |u| = |v|, \\ 0.5, \ \text{ if } \ u = 0, \ \ v \neq 0, \\ 1, \ \text{ if } \ uv > 0 \ \text{ and } \ |u| < |v|. \end{cases} \tag{5.7.10}$$

Furthermore, using a procedure similar to the previous example, the (i, j) entry of $F_{\text{s}}(x) \in \mathbb{R}^{10\times 10}$ can be obtained by using (5.7.10) and the chain rule for differentiation, and (5.7.9) implies that $H_{\text{s}}(x) = C$.

The sum of the Euclidean norm of the error in the state estimates obtained from XKF, XHF, UKF, and UHF for 50 different choices of M with $\text{sprad}(M) = 0.5$, is shown in Figure 5.6. The performance of the four estimators for 50 different choices of M with $\text{sprad}(M) = 10.0$ is shown in Figure 5.7. Again, the performance of UKF

and UHF is better than the performance of XKF and XHF when the nonlinearities become severe. However, the use of XHF or UHF seems to have no significant advantage over XKF or UKF, respectively.

5.8 Simulation Example : One-dimensional Hydrodynamics

Finally, we consider state estimation of one-dimensional hydrodynamic flow based on a finite volume model. The flow of an inviscid, compressible fluid along a one-dimensional channel is governed by Euler's equations

$$\frac{\partial \rho}{\partial t} = -\frac{\partial}{\partial x}\rho v, \tag{5.8.1}$$

$$\frac{d}{dt}\left(\frac{p}{\rho^\gamma}\right) = 0, \tag{5.8.2}$$

$$\rho\frac{\partial v}{\partial t} = -\rho v\frac{\partial v}{\partial x} - \frac{\partial p}{\partial x}, \tag{5.8.3}$$

where $\rho \in \mathbb{R}$ is the density, $v \in \mathbb{R}$ is the velocity, $p \in \mathbb{R}$ is the pressure of the fluid, and $\gamma = \frac{5}{3}$ is the heat capacity ratio of the fluid. A discrete-time model of hydrodynamic flow can be obtained by using a finite-volume based spatial and temporal discretization.

Assume that the channel consists of n identical cells as shown in Figure 3. For all $i = 1, \ldots, n$, let $\rho^{[i]}$, $v^{[i]}$, and $p^{[i]}$ be the density, velocity, and pressure at the center of the ith cell. For all $i = 1 \ldots, n$, define $U^{[i]} \in \mathbb{R}^3$ by

$$U^{[i]} = \begin{bmatrix} \rho^{[i]} & m^{[i]} & \mathcal{E}^{[i]} \end{bmatrix}^{\mathrm{T}}, \tag{5.8.4}$$

where the momentum $m^{[i]}$ and energy $\mathcal{E}^{[i]}$ in the ith cell are given by

$$m^{[i]} = \rho^{[i]}v^{[i]}, \;\; \mathcal{E}^{[i]} = \frac{1}{2}\rho^{[i]}(v^{[i]})^2 + \frac{p^{[i]}}{\gamma - 1}. \tag{5.8.5}$$

We use a second-order Rusanov scheme [67] to discretize (5.8.1)-(5.8.2) and obtain a discrete-time model that enables us to update the flow variables at the center of each cell.

Define the flux dyad $F^{[i]} \in \mathbb{R}^3$ at the ith cell by

$$\left[\begin{array}{cc} m_x^{[i]} & \frac{3-\gamma}{2}\frac{(m_x^{[i]})^2}{\rho^{[i]}} + (\gamma-1)\mathcal{E}^{[i]} \end{array} \quad -\frac{\gamma-1}{2}\frac{(m_x^{[i]})^3}{(\rho^{[i]})^2} + \frac{\gamma m_x^{[i]}\mathcal{E}^{[i]}}{\rho^{[i]}} \right]^{\mathrm{T}}. \tag{5.8.6}$$

Next, define $U_{\mathrm{L}}^{[i]}$ and $U_{\mathrm{R}}^{[i]}$ by

$$U_{\mathrm{L}}^{[i]} \triangleq U^{[i]} + \tfrac{1}{2}\mathrm{minmod}(U^{[i+1]} - U^{[i]}, U^{[i]} - U^{[i-1]}), \tag{5.8.7}$$

$$U_{\mathrm{R}}^{[i]} \triangleq U^{[i]} - \tfrac{1}{2}\mathrm{minmod}(U^{[i+1]} - U^{[i]}, U^{[i]} - U^{[i-1]}). \tag{5.8.8}$$

The left and right flux dyad $F_{\mathrm{L}}^{[i]}$ and $F_{\mathrm{R}}^{[i]}$ is given by (5.8.6) with $U^{[i]} = U_{\mathrm{L}}^{[i]}$ and $U^{[i]} = U_{\mathrm{R}}^{[i]}$, respectively. Finally, define the second-order Rusanov flux $\overline{F}_{\mathrm{Rus}}^{[i]}$ by

$$\overline{F}_{\mathrm{Rus}}^{[i]} \triangleq \frac{1}{2}\left(F_{\mathrm{L}}^{[i]} + F_{\mathrm{R}}^{[i+1]}\right) - c^{[i]}\frac{1}{2}\left(U_{\mathrm{R}}^{[i+1]} - U_{\mathrm{L}}^{[i]}\right), \tag{5.8.9}$$

where

$$c^{[i]} \triangleq \mathrm{abs}(v^{[i]}) + \sqrt{\frac{\gamma p^{[i]}}{\rho^{[i]}}}. \tag{5.8.10}$$

The discrete-time state update equation [66, 67] is given by

$$U_{k+1}^{[i]} = U_k^{[i]} - \frac{t_{\mathrm{s}}}{\Delta x}\left[\overline{F}_{\mathrm{Rus},k}^{[i]} - \overline{F}_{\mathrm{Rus},k}^{[i-1]}\right], \tag{5.8.11}$$

where $t_{\mathrm{s}} < 0$ is the sampling time and Δx is the width of each cell. It follows from (5.8.7)-(5.8.11) that $U_{k+1}^{[i]}$ depends on $U_k^{[i-2]}, \ldots, U_k^{[i+2]}$, as expected for a second-order scheme.

Next, define the state vector $x \in \mathbb{R}^{3(n-4)}$ by

$$x \triangleq \left[\begin{array}{ccc} (U_k^{[3]})^{\mathrm{T}} & \cdots & (U_k^{[n-2]})^{\mathrm{T}} \end{array} \right]^{\mathrm{T}}. \tag{5.8.12}$$

For all $k \geqslant 0$, let $u_k \in \mathbb{R}^3$ denote the boundary condition for the first two cells, so that

$$u_k = (U_k^{[1]})^{\mathrm{T}} = (U_k^{[2]})^{\mathrm{T}}. \tag{5.8.13}$$

Furthermore, we assume Neumann boundary conditions at cells with indices $n-1$ and n so that, for all $k \geqslant 0$,

$$U_k^{[n]} = U_k^{[n-1]} = U_k^{[n-2]}. \tag{5.8.14}$$

It follows from (5.8.11) that the second-order Rusanov scheme yields a nonlinear discrete-time update model of the form

$$x_{k+1} = f(x_k, u_k). \tag{5.8.15}$$

Let $n = 54$ so that $x \in \mathbb{R}^{150}$. For all $k \geqslant 0$, let $\varrho_k^{[1]} = \varrho_k^{[2]} = 1$ kg/m^3, $v_k^{[1]} = v_k^{[2]} = v_{\text{in}} + \frac{v_{\text{in}}}{4}\sin(k)$ m/s, and $p_k^{[1]} = p_k^{[2]} = 1$ N/m^2, where v_{in} is the inlet velocity. We assume that the truth model is given by

$$x_{k+1} = f(x_k, u_k) + w_k, \tag{5.8.16}$$

where $w_k \in \mathbb{R}^{3(n-4)}$ represents unmodeled drivers and is assumed to be zero-mean white Gaussian process noise with covariance matrix $Q \in \mathbb{R}^{3(n-4)\times 3(n-4)}$, where

$$Q = \operatorname{diag}(Q^{[3]}, Q^{[4]}, \ldots, Q^{[n-2]}) \tag{5.8.17}$$

and, for $i = 3, \ldots, n-2$, $Q^{[i]} \in \mathbb{R}^{3\times 3}$ is defined by

$$Q^{[i]} = \begin{cases} \operatorname{diag}(0.05, 0.05, 0.05), & \text{if } i = 10, 25, 40, \\ 0_{3\times 3}, & \text{else.} \end{cases} \tag{5.8.18}$$

It follows from (5.8.16)-(5.8.17) that the flow variables in the 10th, 25th and 40th cell are directly affected by w_k. Next, for $i = 3, \ldots, n-2$, define $C^{[i]} \in \mathbb{R}^{3\times 3(n-4)}$

$$C^{[i]} \triangleq \begin{bmatrix} 0_{3\times 3(n-4-i)} & I_{3\times 3} & 0_{3\times 3(i-1)} \end{bmatrix} \tag{5.8.19}$$

so that the measurement $y_k \in \mathbb{R}^6$ of density, momentum and energy at cells with indices 6, 16, 26, 35, and 42 is given by

$$y_k = Cx_k + v_k, \tag{5.8.20}$$

where

$$C = \begin{bmatrix} (C^{[6]})^{\mathrm{T}} & (C^{[16]})^{\mathrm{T}} & (C^{[26]})^{\mathrm{T}} & (C^{[35]})^{\mathrm{T}} & (C^{[42]})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{5.8.21}$$

and v_k is zero-mean white Gaussian noise with covariance matrix $R = 0.01 I_{30\times 30}$.

Let $t_{\mathrm{s}} = 0.05$ s and $\Delta x = 1$ m. We simulate the truth model (5.8.16) from an arbitrary initial condition $x_0 \in \mathbb{R}^{3(n-4)}$ and obtain measurements y_k from (5.8.20) for various choices of $v_{\mathrm{in}} \in \{0.0, 1.0, 2.0, \ldots, 10.0\}$ m/s. Note that

$$\sqrt{\frac{\gamma p^{[1]}}{\rho^{[1]}}} = 1.29 \text{ m/s}, \tag{5.8.22}$$

and hence, if $v_{\mathrm{in}} > 1.29$ m/s, then the flow is supersonic. The objective is to estimate the density, momentum, and energy at the cells where measurements of flow variables are unavailable using XKF and UKF. It follows from (5.2.9) and (5.6.12) that XHF and UHF involve inverting a $n \times n$ matrix which is computationally intensive when n is large which is the case in finite volume discretization of partial differential equations. Moreover, in the previous examples, no significant improvement in performance was noticed when the XHF and UHF were used instead of XKF and UKF, respectively. Hence, we do not use XHF or UHF for state estimation in the one-dimensional hydrodynamic flow example. To obtain estimates, we initialize the three estimators with the same initial condition $\tilde{x}_0 \neq x_0$. Note that $f(x, u)$ in (5.8.15) contains the nondifferentiable functions abs$(\cdot)$ and minmod$(\cdot, \cdot)$. Hence, we use the averaged Jacobian defined in (5.3.11) in the two-step XKF. Finally, we perform state estimation using UKF.

The error in the estimates of the energy $\mathcal{E}_k^{[30]}$ in cell 30, when measurements y_k are used in XKF and UKF with $v_{\mathrm{in}} = 1$ m/s is shown in Figure 5.8. The error in estimates of the energy $\mathcal{E}_k^{[30]}$ in cell 30, when $v_{\mathrm{in}} = 10$ m/s is shown in Figure 5.9. The sum of the Euclidean norm of error in the state estimates for different values of

v_{in} is shown in Figure 5.10. Note that at low inlet velocities v_{in}, the performance of XKF and UKF is very similar. However, at higher inlet velocities, the nonlinearities are more severe and the performance of UKF is better than that of XKF.

5.9 Conclusion

In this chapter we compared the performance of the extended Kalman filter, the extended H_∞ filter, the unscented Kalman filter, and the unscented H_∞ filter for nonlinear systems with nondifferentiable nonlinearities. Whenever the Jacobian fails to exist, we use an averaged Jacobian based on the symmetric derivatives in the extended Kalman filter. We perform state estimation of one-dimensional hydrodynamic flow based on a finite volume discretization and as the inlet velocity increases the nonlinearities become severe and the performance of UKF is better than that of XKF. For all the examples that we considered, whenever the nonlinearities are not severe, the performance of XKF with the averaged Jacobian and UKF is similar. However, whenever the nonlinearities become more severe, UKF performs better than XKF. No significant improvement in the performance was noticed when either the extended H_∞ filter or the unscented H_∞ filter was used over the extended Kalman filter and unscented Kalman filter, respectively.

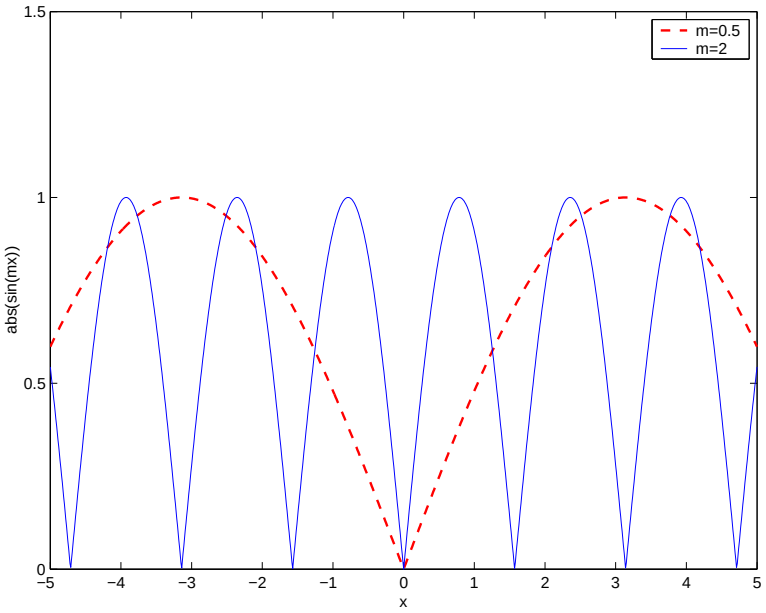


Figure 5.1: Plot of abs(sin(mx)) for $m = 0.5$ and $m = 2$

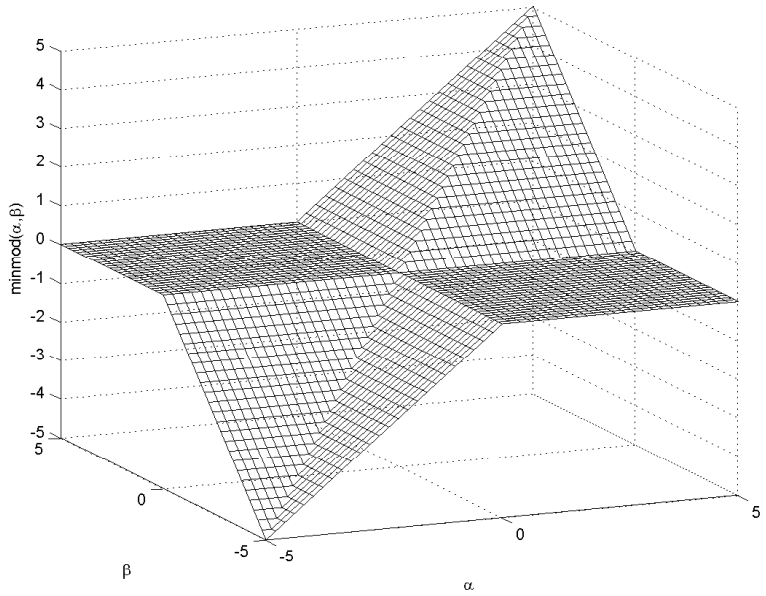


Figure 5.2: Plot of minmod(α, β) for $-5 \leqslant \alpha, \beta < 5$

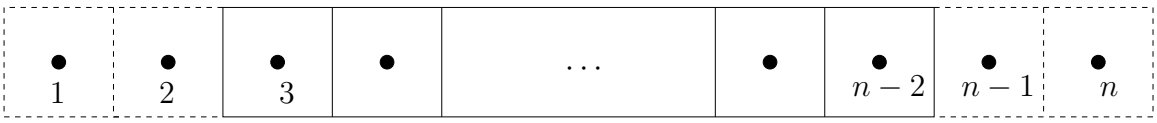


Figure 5.3: One-dimensional grid used in the finite volume scheme

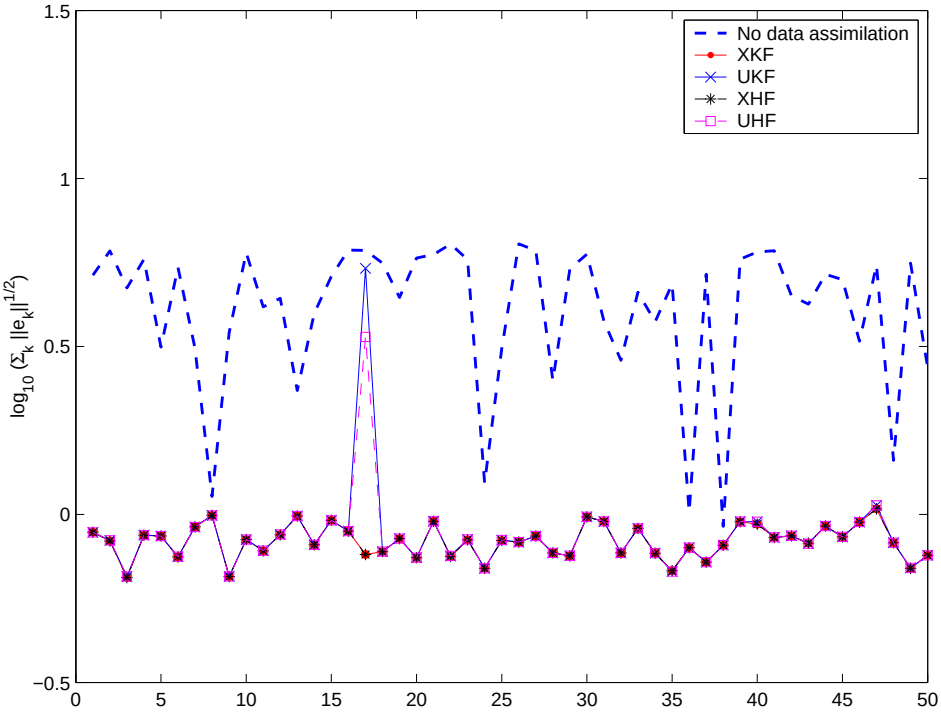


Figure 5.4: Logarithm of the sum of Euclidean norms of the errors in state estimates obtained using XKF, XHF, UKF, and UHF for the system (5.7.1). The performance is compared for 50 different choices of M with $\mathrm{sprad}(M) = 0.5$. The chosen value of $\gamma = 0.4$ is approximately the maximal value for which XHF and UHF are stable. The error in the estimates when no data assimilation is performed, that is, $K_k = 0$ for all $k \geqslant 0$ in XKF is also shown for comparison. The performance of all four estimators is similar and better than the no data assimilation case.

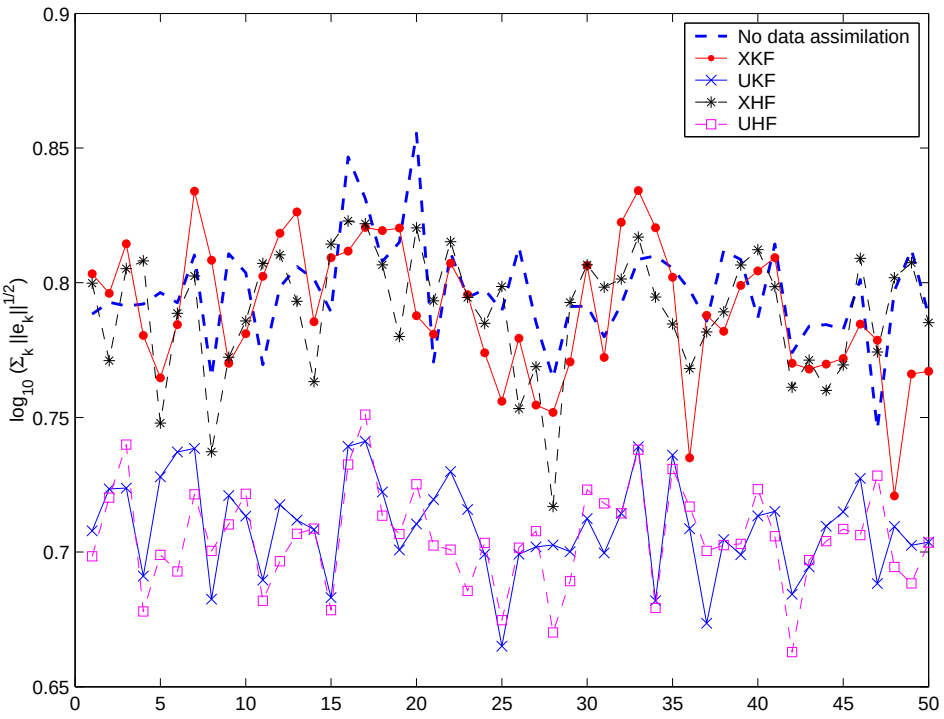


Figure 5.5: Logarithm of the sum of the Euclidean norms of the errors in state estimates obtained using XKF, XHF, UKF, and UHF for the system (5.7.1). The performance is compared for 50 different choices of M with $\text{sprad}(M) = 10$. In this case, the performance of UKF and UHF is much better than the performance of XKF or XHF. In fact, there are cases when the performance of XKF and XHF is worse than the no data assimilation case. However, the performance of UKF is very similar to the performance of UHF, and the performance of XKF is very similar to the performance of XHF.

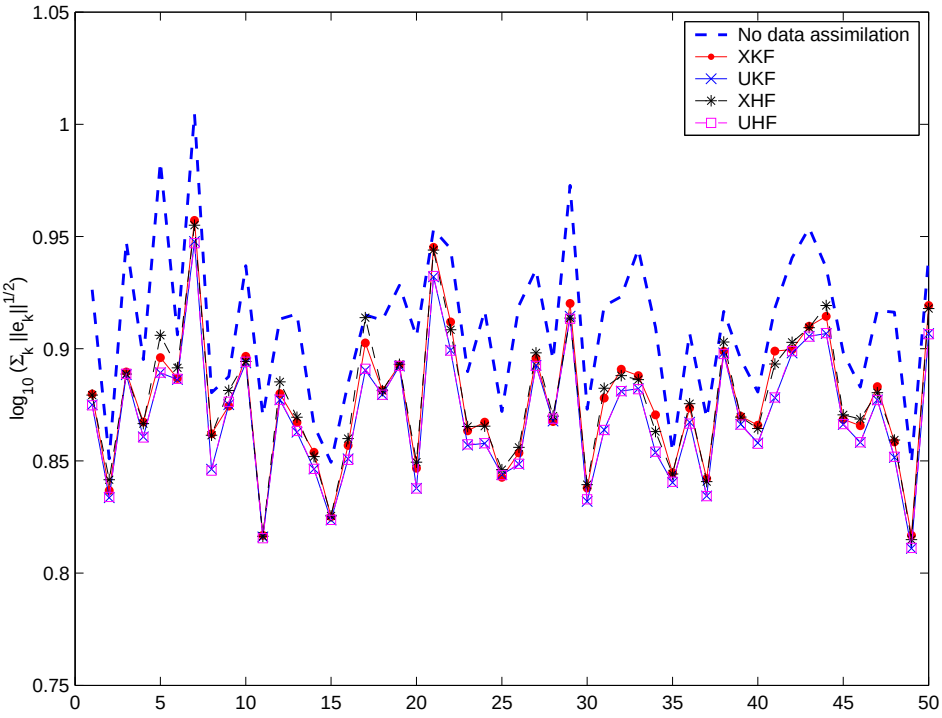


Figure 5.6: Logarithm of the sum of the Euclidean norms of the errors in state estimates obtained using XKF, XHF, UKF, and UHF for the system (5.7.6). The performance of the four estimators are compared for different choices of M with $\text{sprad}(M) = 0.5$. The performance of all four estimators is similar and better than the case when no data assimilation is performed. We choose the largest possible γ (=1.5) such that both XHF and UHF are stable for all choices of M.

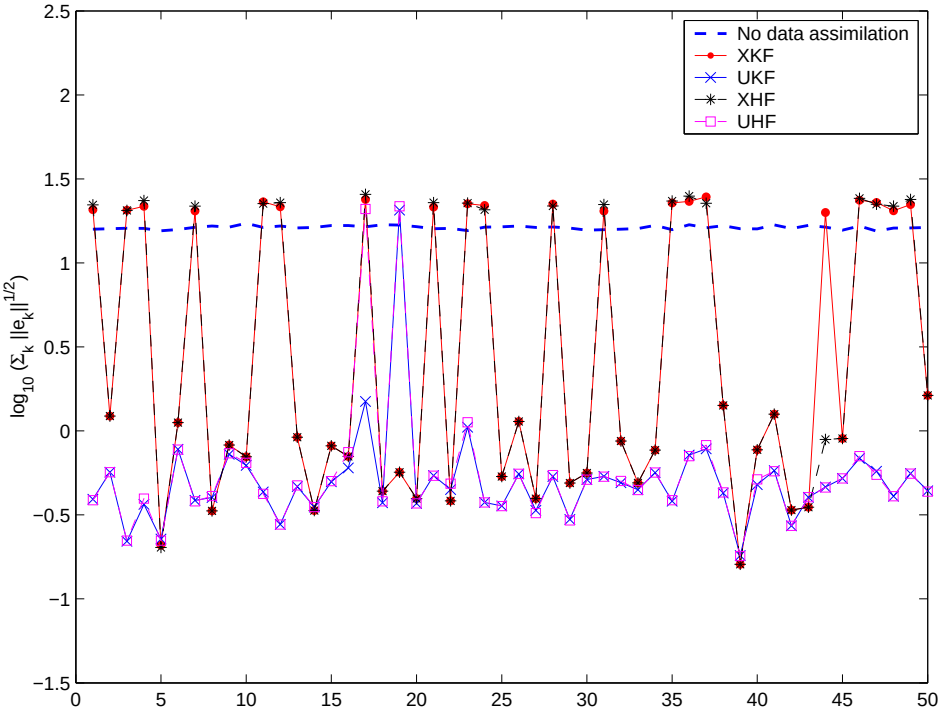


Figure 5.7: Logarithm of the sum of the Euclidean norms of the errors in state estimates obtained using XKF, XHF, UKF, and UHF for the system (5.7.6). The performance of the two estimators is compared for 50 different choices of M with $\text{sprad}(M) = 10.0$. There seems to be no significant improvement in the performance when the H_∞ filters (XHF and UHF) are used over XKF and UKF, respectively.

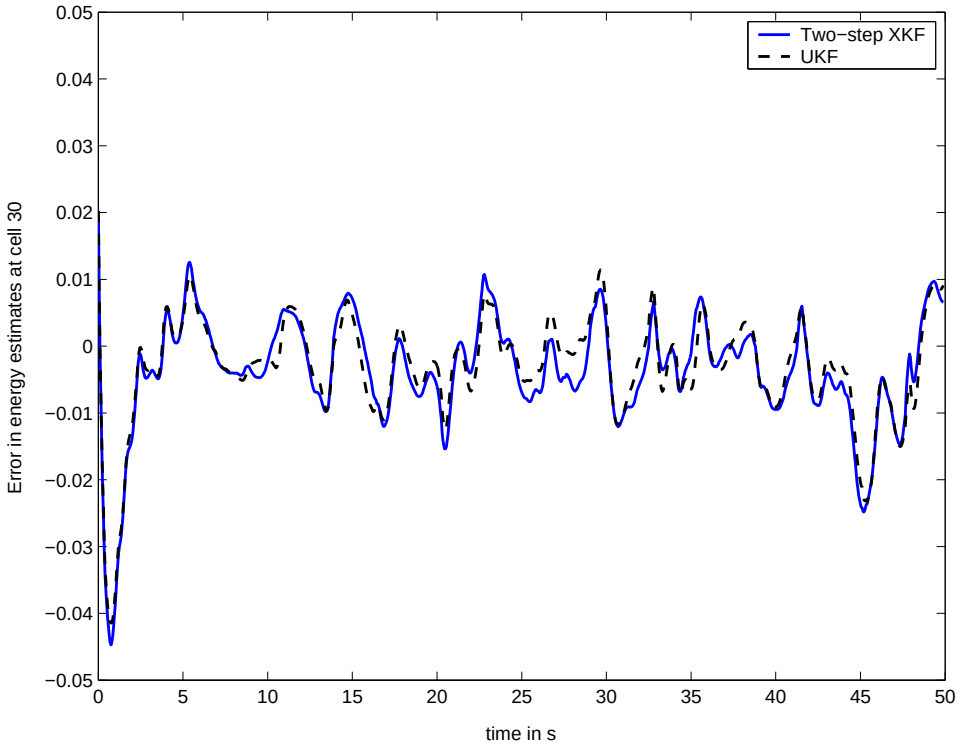


Figure 5.8: The error in the estimates of energy at cell 30 obtained using XKF and UKF when $v_{\text{in}} = 1$ m/s and the flow is subsonic.

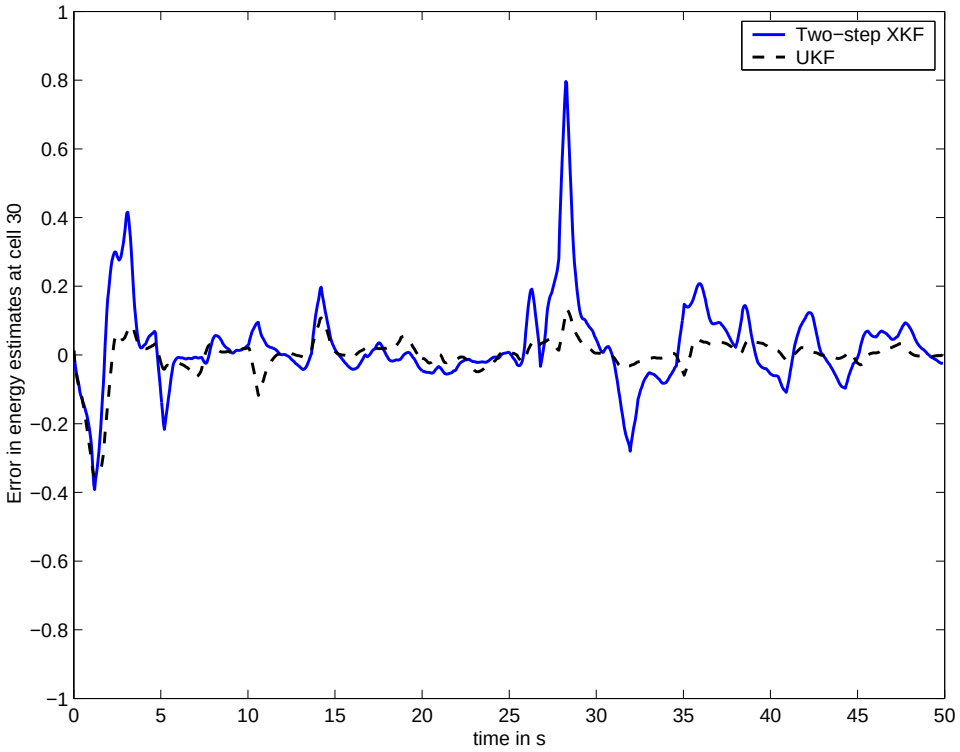


Figure 5.9: The error in the estimates of velocity at cell 30 obtained using XKF and UKF when $v_{\text{in}} = 10$ m/s and the flow is supersonic with Mach number 7.75.

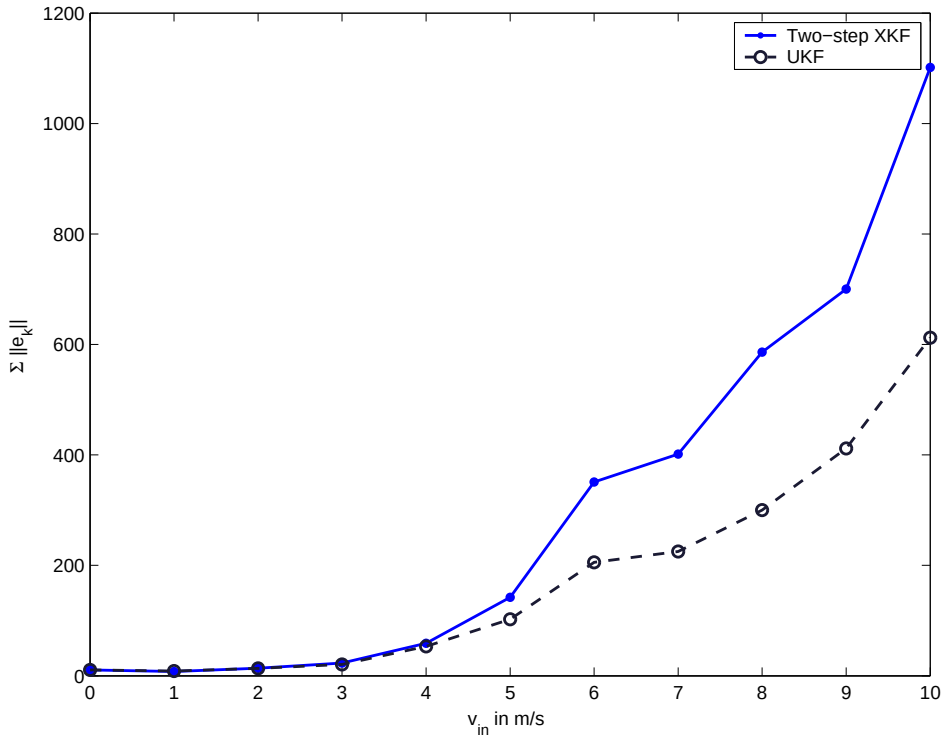


Figure 5.10: The square root of the sum of the Euclidean norms of the errors in state estimates, obtained using XKF and UKF for different choices of the inlet velocity v_{in}. The performance of UKF is better that the performance of XKF for high inlet velocities, with a computational burden that is twice that of XKF.

CHAPTER VI

Reduced-Rank Unscented Kalman Filtering Using Cholesky-Based Decomposition

In the previous chapter, we demonstrated the superiority of the unscented Kalman filter over the extended Kalman filter when the nonlinearity in the dynamical system becomes severe. However, the unscented Kalman filter performs $2n+1$ model update at every time step, where n is the order of the system. In this chapter, we use the results presented in Chapter IV to reduce the ensemble of the unscented Kalman filter. Specifically, we consider a reduced-rank square-root unscented Kalman filter based on the Cholesky decomposition of the state-error covariance. The performance of this filter is compared with an analogous filter based on the singular value decomposition. We evaluate the performance of these filters for illustrative linear and nonlinear systems. The results of this chapter are published in [73].

6.1 Introduction

Data assimilation for large-scale systems has gained increasing attention due to nonlinear and computationally intensive applications such as weather forecasting [38, 78]. These problems require algorithms that are computationally tractable despite the enormous dimension of the state. Reduced-order variants of the classi-

cal Kalman filter have been developed for computationally demanding applications [27, 29, 30, 35], 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 [9].

An alternative technique for reducing the computational requirements of data assimilation for high-dimensional systems is the *reduced-rank filter* [21, 28, 43, 74–76]. In this method, the error-covariance matrix is factored to obtain a square root, whose rank is then reduced through truncation. The truncated square-root is then propagated by the data assimilation algorithm. This technique is closely related to classical decomposition techniques [46, 47], which provide numerical stability and computational efficiency. Factorization-and-truncation methods have direct application to the problem of generating a reduced ensemble for use in particle filter methods [28, 45].

The primary technique for truncating the error-covariance matrix is the singular value decomposition (SVD), wherein the singular values are used to determine which components of the error covariance are most relevant to the accuracy of the state estimates [21, 28, 43]. Despite the intuitively appealing nature of this approach, the optimality of approximation based on the SVD within the context of recursive state estimation is not guaranteed The difficulty 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 related work [42], it is observed that the Kalman filter estimate update depends on the product $C_k P_k$, where C_k is the measurement map and P_k is the error covariance. Consequently, the approximation technique developed in [42] focuses on $C_k P_k$ rather than P_k alone. In particular, it is shown in [42] 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, the accuracy of the Cholesky-decomposition-based reduced-rank filter is typically equal to the accuracy of the full-rank filter, while examples show that, in special cases, 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, $O(n^3/6)$ [48], 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 left-most columns of the lower triangular square root, only the leading submatrix of the error covariance must be factored, yielding yet further savings over the SVD. Once the decomposition is performed, the algorithm effectively retains only the initial "tall" columns of the full Cholesky decomposition and truncates the "short" columns.

To assimilate data in nonlinear systems, particle filters are used to propagate a collection of state estimates from which statistics can be computed. These techniques include the ensemble Kalman filter (EnKF) [61–63], which uses a stochastic construction, as well as the unscented Kalman filter (UKF) [18, 19, 60], which deterministically constructs the collection of state estimates by perturbing the nominal state estimate. Specifically, UKF constructs the ensemble members by using the

columns of the square root of the error covariance to perturb the nominal state estimate. For a model of order n, the n columns and their negatives result in $2n+1$ ensemble members and thus $2n+1$ model updates.

A straightforward approach to reducing the UKF ensemble size is to use a factorization-and-truncation method to truncate $n-q$ columns of the square root of the error covariance and construct the ensemble members using the remaining q columns. In [22, 28, 45], SVD-based decomposition-and-truncation is used to construct reduced-rank approximations of the square root of the error covariance, which are then used to construct the ensemble members resulting in a ensemble size $2q+1$.

In this paper, we use the Cholesky-based decomposition technique developed in [42] to construct the reduced ensemble members. Specifically, we use the Cholesky decomposition to obtain a square root of the error-covariance and select columns of the Cholesky factor to approximate $C_k P_k$. The retained columns of the Cholesky factor are used to construct the ensemble members. We compare the performance of the Cholesky-decomposition-based reduced-rank UKF and the SVD-based reduced-rank UKF on a linear advection model and a nonlinear system that exhibits chaotic dynamics.

6.2 The Reduced-Rank Unscented Transformation

We consider the discrete-time system with nonlinear dynamics

$$x_{k+1} = f(x_k, u_k, k) + w_k \tag{6.2.1}$$

and linearly dependent measurements

$$y_k = C_k x_k + v_k, \tag{6.2.2}$$

where $x_k, w_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, and $y_k, v_k \in \mathbb{R}^p$. The input u_k and output y_k are assumed to be measured, and w_k and v_k are uncorrelated zero-mean white noise processes with covariances Q_k and R_k, respectively. We assume that R_k is positive definite. The objective is to obtain estimates of the state x_k using measurements y_k. When the dynamics (6.2.1) are linear, the Kalman filter provides estimates that minimize the mean-square-error (MSE) in the state estimates [5, 6]. However, for nonlinear dynamics, we approximate the state error covariance using ensembles that are constructed deterministically according to UKF. The starting point for UKF is a set of sample points, that is, a collection of state estimates that capture the probability distribution of the state [18, 19]. Letting x_0^{f} be an initial estimate of x_0 with error covariance P_0^{f}, UKF is given by the following steps:

UKF data assimilation step:

$$x_k^{\mathrm{da}} = x_k^{\mathrm{f}} + K_k(y_k - y_k^{\mathrm{f}}), \tag{6.2.3}$$

$$y_k^{\mathrm{f}} = C_k x_k^{\mathrm{f}}, \tag{6.2.4}$$

$$X_k^{\mathrm{da}} = \Psi(x_k^{\mathrm{da}}, S_k^{\mathrm{da}}, \alpha), \tag{6.2.5}$$

$$P_k^{\mathrm{da}} = P_k^{\mathrm{f}} - P_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} C_k P_k^{\mathrm{f}}, \tag{6.2.6}$$

where

$$K_k = P_k^{\mathrm{f}} C_k^{\mathrm{T}} (C_k P_k^{\mathrm{f}} C_k^{\mathrm{T}} + R_k)^{-1} \tag{6.2.7}$$

and $S_k^{\mathrm{da}} \in \mathbb{R}^{n \times n}$ satisfies

$$S_k^{\mathrm{da}} (S_k^{\mathrm{da}})^{\mathrm{T}} = P_k^{\mathrm{da}}. \tag{6.2.8}$$

UKF forecast step:

$$X^{\mathrm{f}}_{i,k+1} = f(X^{\mathrm{da}}_{i,k}, u_k, k), \quad i = 0, \ldots, 2n, \tag{6.2.9}$$

$$x^{\mathrm{f}}_{k+1} = \sum_{i=0}^{2n} \gamma_i X^{\mathrm{f}}_{i,k+1}, \tag{6.2.10}$$

$$P^{\mathrm{f}}_{k+1} = \sum_{i=0}^{2n} \gamma_i (X^{\mathrm{f}}_{i,k+1} - x^{\mathrm{f}}_{k+1})(X^{\mathrm{f}}_{i,k+1} - x^{\mathrm{f}}_{k+1})^{\mathrm{T}} + Q_k. \tag{6.2.11}$$

It follows from (6.2.9) that UKF involves $2n + 1$ model updates, and hence the computational burden of UKF is of the order $(2n + 1)n^2 = 2n^3 + n^2$. Therefore, when n is large, UKF is computationally expensive. We thus define an unscented transformation for a reduced ensemble. Let $x \in \mathbb{R}^n$ and $S \in \mathbb{R}^{n \times q}$, where $0 < q \leqslant n$. The *rank-q unscented transformation* $X = \Psi_q(x, S, \alpha) \in \mathbb{R}^{n \times (2q+1)}$ of x with covariance $P = SS^{\mathrm{T}}$ is defined by

$$X \triangleq \left[\begin{array}{ccc} X_0 & \cdots & X_{2q} \end{array} \right], \tag{6.2.12}$$

where

$$X_i = \begin{cases} x, & i = 0, \\ x + \sqrt{\alpha} S_i, & i = 1, \ldots, q, \\ x - \sqrt{\alpha} S_{i-q}, & i = q + 1, \ldots, 2q. \end{cases} \tag{6.2.13}$$

Also, defining the weights

$$\gamma_{q,0} \triangleq \frac{\alpha - q}{\alpha}, \quad \gamma_{q,i} \triangleq \frac{1}{2\alpha}, \quad i = 1, \ldots, 2q, \tag{6.2.14}$$

it follows that

$$\sum_{i=0}^{2q} \gamma_{q,i} X_i = x, \quad \sum_{i=0}^{2q} \gamma_{q,i} (X_i - x)(X_i - x)^{\mathrm{T}} = SS^{\mathrm{T}} = P. \tag{6.2.15}$$

Next, we present a case in which the unscented transformation and rank-q unscented transformation are equivalent. The following result is a consequence of (5.5.3) and (6.2.13).

Lemma 6.2.1 *Let $x \in \mathbb{R}^n$, let $P \in \mathbb{R}^{n\times n}$ be positive semidefinite with* $\operatorname{rank}(P) \leqslant q \leqslant n$, *and let $\hat{S} \in \mathbb{R}^{n\times q}$ satisfy $\hat{S}\hat{S}^{\mathrm{T}} = P$. Furthermore, let $S \triangleq \left[\hat{S}\ 0_{n\times(n-q)}\right]$, $\hat{X} \triangleq \Psi_q(x, \hat{S}, \alpha)$, and $X \triangleq \Psi(x, S, \alpha)$. Then, $X_i = x$, for all $i = q+1, \ldots, n, n+q+1, \ldots, 2n$. Moreover, $\hat{X}_0 = X_0$, and for all $i = 1, \ldots, q$, $\hat{X}_i = X_i$ and $\hat{X}_{q+i} = X_{n+q+i}$, where $\hat{X}_i$ is the ith column of $\hat{X}$.*

Lemma 6.2.2 *Assume that* $\operatorname{rank}(P_k^{\mathrm{f}}) \leqslant q \leqslant n$. *Then,* $\operatorname{rank}(P_k^{\mathrm{da}}) \leqslant q$.

Proof. Since $\operatorname{rank}(P_k^{\mathrm{f}}) \leqslant q$, it follows that there exists $S_k^{\mathrm{f}} \in \mathbb{R}^{n\times q}$ satisfying

$$S_k^{\mathrm{f}}(S_k^{\mathrm{f}})^{\mathrm{T}} = P_k^{\mathrm{f}}. \tag{6.2.16}$$

In fact, $S_k^{\mathrm{f}} = \Phi_{\mathrm{SVD}}(P_k^{\mathrm{f}}, q)$ satisfies (6.2.16). Therefore, (6.2.6) implies that P_k^{da} can be expressed as

$$P_k^{\mathrm{da}} = S_k^{\mathrm{f}}\left[I - (C_kS_k^{\mathrm{f}})^{\mathrm{T}}(C_kS_k^{\mathrm{f}}(C_kS_k^{\mathrm{f}})^{\mathrm{T}} + R_k)^{-1}C_kS_k^{\mathrm{f}}\right](S_k^{\mathrm{f}})^{\mathrm{T}}. \tag{6.2.17}$$

Hence, (6.2.17) implies that $\operatorname{rank}(P_k^{\mathrm{da}}) \leqslant q$. Since $\operatorname{rank}(P_k^{\mathrm{da}}) \leqslant q$, there exists $\hat{S}_k^{\mathrm{da}} \in \mathbb{R}^{n\times q}$ satisfying $\hat{S}_k^{\mathrm{da}}(\hat{S}_k^{\mathrm{da}})^{\mathrm{T}} = P_k^{\mathrm{da}}$. □

Hence, if P_k^{f} is rank deficient, then P_k^{da} is also rank deficient. The following result shows that the ensemble size can be reduced from $2n+1$ to $2q+1$ when $\operatorname{rank}(P_k^{\mathrm{f}}) = q$.

Proposition 6.2.1 *Assume* $\operatorname{rank}(P_k^{\mathrm{f}}) \leqslant q \leqslant n$, *and define $S_k^{\mathrm{da}} \triangleq \left[\hat{S}_k^{\mathrm{da}}\ 0_{n\times(n-q)}\right]$, where $\hat{S}_k^{\mathrm{da}} \in \mathbb{R}^{n\times q}$ satisfies $\hat{S}_k^{\mathrm{da}}(\hat{S}_k^{\mathrm{da}})^{\mathrm{T}} = P_k^{\mathrm{da}}$. Define $\hat{X}_k^{\mathrm{da}} \triangleq \Psi_q(x_k^{\mathrm{da}}, \hat{S}_k^{\mathrm{da}}, \alpha)$, and let $\hat{x}_{k+1}^{\mathrm{f}} \in \mathbb{R}^n$ and $\hat{P}_{k+1}^{\mathrm{f}} \in \mathbb{R}^{n\times n}$ be given by*

$$\hat{x}_{k+1}^{\mathrm{f}} = \sum_{i=0}^{2q} \gamma_{q,i}\hat{X}_{i,k+1}^{\mathrm{f}}, \tag{6.2.18}$$

$$\hat{P}_{k+1}^{\mathrm{f}} = \sum_{i=0}^{2q} \gamma_{q,i}(\hat{X}_{i,k+1}^{\mathrm{f}} - \hat{x}_{k+1}^{\mathrm{f}})(\hat{X}_{i,k+1}^{\mathrm{f}} - \hat{x}_{k+1}^{\mathrm{f}})^{\mathrm{T}} + Q_k, \tag{6.2.19}$$

where $\hat{X}^{\mathrm{f}}_{i,k+1} \in \mathbb{R}^n$ *is given by*

$$\hat{X}^{\mathrm{f}}_{i,k+1} = f(\hat{X}^{\mathrm{da}}_{i,k}, u_k, k), \quad i = 0, \ldots, 2q, \tag{6.2.20}$$

and $\hat{X}^{\mathrm{da}}_{i,k} \in \mathbb{R}^n$ *is the* i*th column of* $\hat{X}^{\mathrm{da}}_k$*. Then,* $\hat{x}^{\mathrm{f}}_{k+1} = x^{\mathrm{f}}_{k+1}$ *and* $\hat{P}^{\mathrm{f}}_{k+1} = P^{\mathrm{f}}_{k+1}$*.*

Proof. It follows from Lemma 6.2.1 that $\hat{X}^{\mathrm{da}}_{0,k} = X^{\mathrm{da}}_{0,k}$, for all $i = 1, \ldots, q$, $\hat{X}^{\mathrm{da}}_{i,k} = X^{\mathrm{da}}_{i,k}$ and $\hat{X}^{\mathrm{da}}_{q+i,k} = X^{\mathrm{da}}_{n+q+i,k}$, and for all $i = q+1, \ldots, n, n+q+1, \ldots, 2n$, $X^{\mathrm{da}}_{i,k} = x^{\mathrm{da}}_k$. Therefore, the (6.2.9) and (6.2.20) imply that $\hat{X}^{\mathrm{f}}_{i,k+1} = X^{\mathrm{f}}_{i,k+1}$ and $\hat{X}^{\mathrm{f}}_{q+i,k+1} = X^{\mathrm{f}}_{n+q+i,k+1}$, and for all $i = q+1, \ldots, n, n+q+1, \ldots, 2n$, $X^{\mathrm{f}}_{i,k+1} = X^{\mathrm{f}}_{0,k+1}$. Finally, the result follows from (5.5.4), (6.2.10), (6.2.11), (6.2.14), and (6.2.18). □

Hence, when $\mathrm{rank}(P^{\mathrm{f}}_k) = q < n$, the ensemble size can be reduced from $2n + 1$ to $2q + 1$, and thus, using the rank-q unscented transformation instead of the unscented transformation in (6.2.5) of UKF does not degrade the performance of UKF. However, when P^{f}_k has full rank, P^{da}_k generally has full rank. In this case, we construct rank-q approximations of the pseudo-error covariances and perform estimation using the rank-q unscented transformation based on a square root of the low-rank approximation of the pseudo-error covariance.

6.3 SVD-Based Reduced-Rank Unscented Kalman Filter

To reduce the ensemble size, we use a reduced-rank approximation $\hat{P}^{\mathrm{f}}_{\mathrm{s},k}$ of $P^{\mathrm{f}}_{\mathrm{s},k}$. The reduced-rank approximations are chosen such that $\|\hat{P}^{\mathrm{f}}_{\mathrm{s},k} - P^{\mathrm{f}}_{\mathrm{s},k}\|_{\mathrm{F}}$ is minimized subject to $\mathrm{rank}(\hat{P}^{\mathrm{f}}_{\mathrm{s},k}) = q$, where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. 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 $u_1, \ldots, u_n \in \mathbb{R}^n$ be the corresponding orthogonal singular vectors. Next, define $U_q \in \mathbb{R}^{n\times q}$ and $\Sigma_q \in \mathbb{R}^{q\times q}$ by

$$U_q \triangleq \left[\begin{array}{ccc} u_1 & \cdots & u_q \end{array}\right], \quad \Sigma_q \triangleq \mathrm{diag}(\sigma_1, \ldots, \sigma_q). \tag{6.3.1}$$

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

$$P = U_n \Sigma_n U_n^{\mathrm{T}}, \tag{6.3.2}$$

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 $U_n \Sigma_n^{1/2}$ of P given by

$$\Phi_{\mathrm{SVD}}(P, q) \triangleq U_q \Sigma_q^{1/2}. \tag{6.3.3}$$

As noted in [36], SS^{T}, where $S \triangleq \Phi_{\mathrm{SVD}}(P, q)$, is the best rank-$q$ approximation of P in the Frobenius norm .

Next, we use the singular value decomposition at each time step to obtain a reduced-rank approximation of the pseudo-error covariance, and this reduction in rank enables us to reduce the ensemble size. The SVD-based reduced-rank square-root unscented Kalman filter (SVDRRUKF) is summarized as follows:

SVDRRUKF data assimilation step:

$$x_{\mathrm{s},k}^{\mathrm{da}} = x_{\mathrm{s},k}^{\mathrm{f}} + K_{\mathrm{s},k}(y_k - y_{\mathrm{s},k}^{\mathrm{f}}), \tag{6.3.4}$$

$$y_{\mathrm{s},k}^{\mathrm{f}} = C_k x_{\mathrm{s},k}^{\mathrm{f}}, \tag{6.3.5}$$

$$X_{\mathrm{s},k}^{\mathrm{da}} = \Psi_q(x_{\mathrm{s},k}^{\mathrm{da}}, S_{\mathrm{s},k}^{\mathrm{da}}, \alpha), \tag{6.3.6}$$

$$S_{\mathrm{s},k}^{\mathrm{da}} = S_{\mathrm{s},k}^{\mathrm{f}} H_{\mathrm{s},k}^{\mathrm{f}}, \tag{6.3.7}$$

where

$$K_{\mathrm{s},k} = S_{\mathrm{s},k}^{\mathrm{f}} (C_k S_{\mathrm{s},k}^{\mathrm{f}})^{\mathrm{T}} \left(C_k S_{\mathrm{s},k}^{\mathrm{f}} (C_k S_{\mathrm{s},k}^{\mathrm{f}})^{\mathrm{T}} + R_k \right)^{-1} \tag{6.3.8}$$

and $H_{\mathrm{s},k}^{\mathrm{f}} \in \mathbb{R}^{q \times q}$ satisfies

$$H_{\mathrm{s},k}^{\mathrm{f}} (H_{\mathrm{s},k}^{\mathrm{f}})^{\mathrm{T}} = I_q - (C_k S_{\mathrm{s},k}^{\mathrm{f}})^{\mathrm{T}} \left(C_k S_{\mathrm{s},k}^{\mathrm{f}} (C_k S_{\mathrm{s},k}^{\mathrm{f}})^{\mathrm{T}} + R_k \right)^{-1} C_k S_{\mathrm{s},k}^{\mathrm{f}}. \tag{6.3.9}$$

SVDRRUKF forecast step:

$$X^{\mathrm{f}}_{\mathrm{s},i,k+1} = f(X^{\mathrm{da}}_{\mathrm{s},i,k}, u_k, k), \quad i = 0, \ldots, 2q, \tag{6.3.10}$$

$$x^{\mathrm{f}}_{k+1} = \sum_{i=0}^{2q} \gamma_{q,i} X^{\mathrm{f}}_{\mathrm{s},i,k+1}, \tag{6.3.11}$$

$$P^{\mathrm{f}}_{\mathrm{s},k+1} = \sum_{i=0}^{2q} \gamma_{q,i} (X^{\mathrm{f}}_{\mathrm{s},i,k+1} - x^{\mathrm{f}}_{\mathrm{s},k+1})(X^{\mathrm{f}}_{\mathrm{s},i,k+1} - x^{\mathrm{f}}_{\mathrm{s},k+1})^{\mathrm{T}} + Q_k, \tag{6.3.12}$$

$$S^{\mathrm{f}}_{\mathrm{s},k+1} = \Phi_{\mathrm{SVD}}(P^{\mathrm{f}}_{\mathrm{s},k+1}, q). \tag{6.3.13}$$

Next, define $\hat{P}^{\mathrm{f}}_{\mathrm{s},k}, \hat{P}^{\mathrm{da}}_{\mathrm{s},k} \in \mathbb{R}^{n \times n}$ by

$$\hat{P}^{\mathrm{f}}_{\mathrm{s},k} \triangleq S^{\mathrm{f}}_{\mathrm{s},k} (S^{\mathrm{f}}_{\mathrm{s},k})^{\mathrm{T}}, \quad \hat{P}^{\mathrm{da}}_{\mathrm{s},k} \triangleq \hat{P}^{\mathrm{f}}_{\mathrm{s},k} - \hat{P}^{\mathrm{f}}_{\mathrm{s},k} C_k^{\mathrm{T}} (C_k \hat{P}^{\mathrm{f}}_{\mathrm{s},k} C_k^{\mathrm{T}} + R_k)^{-1} C_k \hat{P}^{\mathrm{f}}_{\mathrm{s},k}. \tag{6.3.14}$$

It then follows from (6.3.7) that $S^{\mathrm{da}}_{\mathrm{s},k} (S^{\mathrm{da}}_{\mathrm{s},k})^{\mathrm{T}} = \hat{P}^{\mathrm{da}}_{\mathrm{s},k}$. Furthermore, (6.3.8) and (6.3.14) imply that

$$K_{\mathrm{s},k} = \hat{P}^{\mathrm{f}}_{\mathrm{s},k} C_k^{\mathrm{T}} (C_k \hat{P}^{\mathrm{f}}_{\mathrm{s},k} C_k^{\mathrm{T}} + R_k)^{-1}. \tag{6.3.15}$$

Furthermore, since $\mathrm{rank}(S^{\mathrm{f}}_{\mathrm{s},k}) \leqslant q$, it follows from (6.3.14) that $\mathrm{rank}(\hat{P}^{\mathrm{f}}_{\mathrm{s},k}) \leqslant q$ and $\mathrm{rank}(\hat{P}^{\mathrm{da}}_{\mathrm{s},k}) \leqslant q$. Hence, (6.3.15) implies that the filter gain $K_{\mathrm{s},k}$ depends on $\hat{P}^{\mathrm{f}}_{\mathrm{s},k}$, the reduced-rank approximation of $P^{\mathrm{f}}_{\mathrm{s},k}$, and the ensemble X^{da}_k depends on $\hat{P}^{\mathrm{da}}_{\mathrm{s},k}$, the reduced-rank approximation of $P^{\mathrm{da}}_{\mathrm{s},k}$. Also, as shown in Section 6.8, the matrix $H^{\mathrm{f}}_{\mathrm{s},k}$ satisfying (6.3.9) is not unique. Since the singular value decomposition in (6.3.13) is computationally intensive [48], we introduce an alternative method to obtain a reduced-rank approximation of a square root of the pseudo-error covariance.

6.4 Cholesky-Factorization-Based Reduced-Rank Unscented Kalman Filter

The filter gain K_k of UKF depends on a particular subspace of the forecast error covariance P^{f}_k. Specifically, K_k depends only on the correlation $C_k P^{\mathrm{f}}_k$ between

the error in the measured states and unmeasured states. Since $\mathrm{rank}(C_k) = p$, there exists a transformation matrix $T_k \in \mathbb{R}^{n\times n}$ such that the change of basis $\tilde{x}_k = T_k x_k$ ensures that (see Section 6.9) C_k has the form

$$C_k = \begin{bmatrix} I_p & 0 \end{bmatrix}. \tag{6.4.1}$$

The following result is given in [42].

Lemma 6.4.1 *Partition* P_k^{f} *as*

$$P_k^{\mathrm{f}} = \begin{bmatrix} P_{p,k}^{\mathrm{f}} & (P_{\overline{p}p,k}^{\mathrm{f}})^{\mathrm{T}} \\ P_{\overline{p}p,k}^{\mathrm{f}} & P_{\overline{p},k}^{\mathrm{f}} \end{bmatrix}, \tag{6.4.2}$$

where $P_{p,k}^{\mathrm{f}} \in \mathbb{R}^{p\times p}$ *and* $P_{\overline{p},k}^{\mathrm{f}} \in \mathbb{R}^{\overline{p}\times\overline{p}}$*, and assume that* C_k *has the form* (6.4.1)*. Then,*

$$K_k = \begin{bmatrix} P_{p,k}^{\mathrm{f}} \\ P_{\overline{p}p,k}^{\mathrm{f}} \end{bmatrix} (P_{p,k}^{\mathrm{f}} + R_k)^{-1}. \tag{6.4.3}$$

Next, to reduce the ensemble size, we construct a filter that uses a reduced-rank approximation $\hat{P}_{\mathrm{c},k}^{\mathrm{f}}$ of $P_{\mathrm{c},k}^{\mathrm{f}}$ such that $\mathrm{rank}(\hat{P}_{\mathrm{c},k}^{\mathrm{f}}) < n$ and $\|C_k(\hat{P}_{\mathrm{c},k}^{\mathrm{f}} - P_{\mathrm{c},k}^{\mathrm{f}})\|_{\mathrm{F}}$ is minimized. To obtain $\hat{P}_{\mathrm{c},k}^{\mathrm{f}}$, we perform a Cholesky factorization of the pseudo-error covariance $P_{\mathrm{c},k}^{\mathrm{f}}$ at each time step. Assuming that $P \in \mathbb{R}^{n\times n}$ is positive definite, the Cholesky factorization of P yields a unique lower triangular Cholesky factor $L \in \mathbb{R}^{n\times n}$ satisfying

$$LL^{\mathrm{T}} = P. \tag{6.4.4}$$

Truncating the last $n-q$ columns of $L = \begin{bmatrix} L_1 & \cdots & L_n \end{bmatrix}$ yields the rank-q Cholesky factor

$$\Phi_{\mathrm{CHOL}}(P, q) \triangleq \begin{bmatrix} L_1 & \cdots & L_q \end{bmatrix} \in \mathbb{R}^{n\times q}. \tag{6.4.5}$$

The following result is given in [42].

Lemma 6.4.2 *Let $P \in \mathbb{R}^{n\times n}$ be positive definite. Define $S \triangleq \Phi_{\mathrm{CHOL}}(P,q)$, where $0 < q \leqslant n$, and $\hat{P} \triangleq SS^{\mathrm{T}}$, and partition P and $\hat{P}$ as*

$$P = \begin{bmatrix} P_q & P_{q\bar{q}} \\ (P_{q\bar{q}})^{\mathrm{T}} & P_{\bar{q}} \end{bmatrix}, \quad \hat{P} = \begin{bmatrix} \hat{P}_q & \hat{P}_{q\bar{q}} \\ (\hat{P}_{q\bar{q}})^{\mathrm{T}} & \hat{P}_{\bar{q}} \end{bmatrix}, \tag{6.4.6}$$

where $P_q, \hat{P}_q \in \mathbb{R}^{q\times q}$ and $P_{\bar{q}}, \hat{P}_{\bar{q}} \in \mathbb{R}^{\bar{q}\times\bar{q}}$. Then,

$$\begin{bmatrix} \hat{P}_q & \hat{P}_{q\bar{q}} \end{bmatrix} = \begin{bmatrix} P_q & P_{q\bar{q}} \end{bmatrix}. \tag{6.4.7}$$

Lemma 6.4.2 implies that, if $S = \Phi_{\mathrm{CHOL}}(P,q)$, then the first q columns and rows of SS^{T} and P are equal. Next, we use the Cholesky factorization at each time step to obtain a reduced-rank approximation of the pseudo-error covariance, thus reducing the ensemble size. The Cholesky-based reduced-rank unscented Kalman filter (CDRRUKF) is summarized as follows:

CDRRUKF data assimilation step:

$$x^{\mathrm{da}}_{\mathrm{c},k} = x^{\mathrm{f}}_{\mathrm{c},k} + K_{\mathrm{c},k}(y_k - y^{\mathrm{f}}_{\mathrm{c},k}), \tag{6.4.8}$$

$$y^{\mathrm{f}}_{\mathrm{c},k} = C_k x^{\mathrm{f}}_{\mathrm{c},k}, \tag{6.4.9}$$

$$X^{\mathrm{da}}_{\mathrm{c},k} = \Psi_q(x^{\mathrm{da}}_{\mathrm{c},k}, S^{\mathrm{da}}_{\mathrm{c},k}, \alpha), \tag{6.4.10}$$

$$S^{\mathrm{da}}_{\mathrm{c},k} = S^{\mathrm{f}}_{\mathrm{c},k} H^{\mathrm{f}}_{\mathrm{c},k}, \tag{6.4.11}$$

where

$$K_{\mathrm{c},k} = S^{\mathrm{f}}_{\mathrm{c},k}(C_k S^{\mathrm{f}}_{\mathrm{c},k})^{\mathrm{T}} \left(C_k S^{\mathrm{f}}_{\mathrm{c},k}(C_k S^{\mathrm{f}}_{\mathrm{c},k})^{\mathrm{T}} + R_k\right)^{-1} \tag{6.4.12}$$

and $H^{\mathrm{f}}_{\mathrm{c},k} \in \mathbb{R}^{q\times q}$ satisfies

$$H^{\mathrm{f}}_{\mathrm{c},k}(H^{\mathrm{f}}_{\mathrm{c},k})^{\mathrm{T}} = I - (C_k S^{\mathrm{f}}_{\mathrm{c},k})^{\mathrm{T}} \left(C_k S^{\mathrm{f}}_{\mathrm{c},k}(C_k S^{\mathrm{f}}_{\mathrm{c},k})^{\mathrm{T}} + R_k\right)^{-1} C_k S^{\mathrm{f}}_{\mathrm{c},k}. \tag{6.4.13}$$

CDRRUKF forecast step:

$$X^{\mathrm{f}}_{\mathrm{c},i,k+1} = f(X^{\mathrm{da}}_{\mathrm{c},i,k}, u_k, k), \quad i = 0, \ldots, 2q \tag{6.4.14}$$

$$x^{\mathrm{f}}_{k+1} = \sum_{i=0}^{2q} \gamma_{q,i} X^{\mathrm{f}}_{\mathrm{c},i,k+1}, \tag{6.4.15}$$

$$P^{\mathrm{f}}_{\mathrm{c},k+1} = \sum_{i=0}^{2q} \gamma_{q,i} (X^{\mathrm{f}}_{\mathrm{c},i,k+1} - x^{\mathrm{f}}_{\mathrm{c},k+1})(X^{\mathrm{f}}_{\mathrm{c},i,k+1} - x^{\mathrm{f}}_{\mathrm{c},k+1})^{\mathrm{T}} + Q_k, \tag{6.4.16}$$

$$S^{\mathrm{f}}_{\mathrm{c},k+1} = \Phi_{\mathrm{CHOL}}(P^{\mathrm{f}}_{\mathrm{c},k+1}, q). \tag{6.4.17}$$

Next, define $\hat{P}^{\mathrm{da}}_{\mathrm{c},k}, \hat{P}^{\mathrm{f}}_{\mathrm{c},k} \in \mathbb{R}^{n \times n}$ by

$$\hat{P}^{\mathrm{da}}_{\mathrm{c},k} \triangleq \hat{P}^{\mathrm{f}}_{\mathrm{c},k} - \hat{P}^{\mathrm{f}}_{\mathrm{c},k} C_k^{\mathrm{T}} (C_k \hat{P}^{\mathrm{f}}_k C_k^{\mathrm{T}} + R_k)^{-1} C_k \hat{P}^{\mathrm{f}}_{\mathrm{c},k}, \quad \hat{P}^{\mathrm{f}}_{\mathrm{c},k} \triangleq S^{\mathrm{f}}_{\mathrm{c},k} (S^{\mathrm{f}}_{\mathrm{c},k})^{\mathrm{T}}. \tag{6.4.18}$$

It then follows from (6.4.11) that $S^{\mathrm{da}}_{\mathrm{c},k}(S^{\mathrm{da}}_{\mathrm{c},k})^{\mathrm{T}} = \hat{P}^{\mathrm{da}}_{\mathrm{c},k}$. Furthermore, (6.4.12) and (6.4.18) imply that

$$K_{\mathrm{c},k} = \hat{P}^{\mathrm{f}}_{\mathrm{c},k} C_k^{\mathrm{T}} (C_k \hat{P}^{\mathrm{f}}_k C_k^{\mathrm{T}} + R_k)^{-1}. \tag{6.4.19}$$

Hence, like the estimator gain $K_{\mathrm{s},k}$ of SVDRRUKF, the estimator gain $K_{\mathrm{c},k}$ of CDRRUKF depends on a reduced-rank approximation $\hat{P}^{\mathrm{f}}_{\mathrm{c},k}$ of the pseudo-error covariance $P^{\mathrm{f}}_{\mathrm{c},k}$. As discussed in Section 6.8, the matrix $H^{\mathrm{f}}_{\mathrm{c},k}$ satisfying (6.3.9) is not unique. Due to the rank-reduction step (6.4.17), CDRRUKF is generally not equivalent to UKF.

6.5 Linear Advection Model

Consider a linear advection model [78] with n cells, and let $x_{i,k}$ be the energy in the ith cell at time k. The energy flow satisfies

$$x_{i,k+1} = \begin{cases} x_{i-1,k}, & \text{if } i = 2, \ldots, n, \\ x_{n,k}, & \text{if } i = 1. \end{cases} \tag{6.5.1}$$

Hence, energy in the ith cell flows to the $(i+1)$th cell, while the periodic boundary condition ensures that energy is in constant circulation. We choose $n = 100$ and assume that the disturbance w_k enters selected cells, where $w_k \in \mathbb{R}^n$ is white noise process with covariance $Q_k = Q$ for all $k \geqslant 0$, and $Q \in \mathbb{R}^{n \times n}$ is diagonal with entries

$$Q_{i,i} = \begin{cases} 1, & \text{if } i \in \{10, 20, \ldots, 100\}, \\ 0, & \text{else.} \end{cases} \tag{6.5.2}$$

Next, we assume that measurements of the energy in cells 50 and 51 are available so that

$$y_k = \begin{bmatrix} x_{50,k} \\ x_{51,k} \end{bmatrix} + v_k, \tag{6.5.3}$$

where v_k is white noise process with covariance $R_k = 0.1I_2$. Note that (6.5.3) can be expressed as (6.2.2).

First, we use the measurements y_k to estimate the energy in the remaining cells using UKF, SVDRRUKF, and CDRRUKF. In all three cases, the initial estimates x_0^{f}, $x_{\mathrm{s},0}^{\mathrm{f}}$, and $x_{\mathrm{c},0}^{\mathrm{f}}$ are not equal to the initial state x_0. Moreover, we choose $P_0^{\mathrm{f}} = P_{\mathrm{s},0}^{\mathrm{f}} = P_{\mathrm{c},0}^{\mathrm{f}} = 0.1I_n$. Finally, we choose $\alpha = 0.6$ for all three filters. Note that since the dynamics in (6.5.1) are linear, UKF is equivalent to the Kalman filter and hence UKF provides the optimal estimates of the state x_k that minimize the MSE. The MSE of state estimates from UKF is shown in Figure 6.1. The MSE of state estimates when data assimilation is not performed is also shown for comparison.

Next, as shown in Figure 6.2 and Figure 6.3, data assimilation is performed using SVDRRUKF and CDRRUKF for several values of q between 5 and 100. Note that SVDRRUKF and CDRRUKF use $2q + 1$ ensemble members, whereas UKF uses $2n + 1$ ensemble members. It can be seen that the performance of SVDRRUKF

with 111 ensemble members ($q = 55$) is close to optimal, whereas the performance of CDRRUKF is close to optimal with 11 ensemble members ($q = 5$). The steady-state MSE of state estimates for various values of q is plotted in Figure 6.4 and Figure 6.5. The performance of SVDRRUKF is poor when $q < 55$, and close to optimal when $q \geqslant 55$. Thus the ensemble size can be reduced from 201 to 111 with negligible change in the performance. Finally, note that even with $q = 5$, the performance of CDRRUKF is close to optimal. Hence, the ensemble size can be reduced from 211 to 11 with negligible performance deterioration.

Next, we repeat the same procedure except with a poor estimate of the process noise covariance for data assimilation. Specifically, we replace Q_k in (6.3.12) and (6.4.16) by $\hat{Q}_k$, where $\hat{Q}_k = I$ for all $k \geqslant 0$. The steady-state MSE of state estimates for different choices of q is plotted in Figure 6.4 and Figure 6.5. SVDRRUKF with a poor estimate of the error covariance is unstable for all $q \leqslant 95$ (indicated by the X's). However, it can be seen from Figure 6.5 that even with $q = 5$ and a poor estimate of the process noise covariance, the performance of CDRRUKF is close to optimal.

Finally, we replace Q_k in (6.4.17) by $\hat{Q}_k$, where $\hat{Q}_k = \alpha I$ for all $k \geqslant 0$, and perform state estimation using CDRRUKF. The steady-state MSE of the state estimates is shown in Figure 6.6 for various values of α. The degradation in performance for smaller values of α is less when the ensemble size is large. However, for all three cases $q = 5$, $q = 15$, and $q = 15$, the performance of CDRRUKF is close to optimal when $\alpha \geqslant 1$. This suggests that it is advantageous to overestimate the process noise covariance. SVDRRUKF with $q = 5, 15, 25$ is unstable for all choices of $\alpha = 0.005, \ldots, 50$. Hence, these simulations suggest that CDRRUKF is more robust than SVDRRUKF with respect to uncertainties in the process noise covariance.

6.6 L96 Model

Next, we compare the performance of SVDRRUKF and CDRRUKF on a nonlinear model that exhibits chaotic dynamics. The L96 model mimics the propagation of an unspecified meteorological quantity along the latitude circle [79]. The dynamics are governed by

$$\frac{d}{dt}x_i(t) = (x_{i+1}(t) - x_{i-2}(t))x_{i-1}(t) - x_i(t) + u_i(t), \tag{6.6.1}$$

where $x_i(t) \in \mathbb{R}$ denotes the meteorological quantity at the ith grid point at time t, $u_i \in \mathbb{R}$ denotes an external forcing term, and w_i denotes unknown disturbances affecting the ith grid point. For all $t \geqslant 0$, the boundary conditions are defined by

$$x_0(t) = x_n(t), \;\; x_{-1}(t) = x_{n-1}(t), \;\; x_{n+1}(t) = x_1(t). \tag{6.6.2}$$

We choose $u_i(t) = 8$ for all $i = 1, \ldots, n$ and all $t \geqslant 0$. Using fourth-order Runge-Kutta discretization with a sampling time of 0.05 s, we obtain a discrete-time model of (6.6.1) that can be expressed as (6.2.1). Furthermore, we assume that the discretized model is corrupted by an unknown external disturbance that affects certain cells. We choose $n = 40$, and assume that w_k is white noise process with covariance $Q_k = Q$ for all $k \geqslant 0$, where $Q \in \mathbb{R}^{n \times n}$ is diagonal with entries

$$Q_{i,i} = \begin{cases} 0.1, & \text{if } i \in \{5, 15, 25, 35\}, \\ 0, & \text{else.} \end{cases} \tag{6.6.3}$$

Next, we assume that measurements from cells with 20 and 21 are available so that

$$y_k = \begin{bmatrix} x_{20,k} \\ x_{21,k} \end{bmatrix} + v_k, \tag{6.6.4}$$

where v_k is white noise process with covariance $R_k = 0.01 I_2$. Hence, (6.6.4) can be expressed as (6.2.2) with $C_k = C \in \mathbb{R}^{2 \times 40}$. We use the measurements y_k to estimate

the state in the cells where measurements are not available. The estimates of $x_{20}(t)$ and $x_{23}(t)$ obtained using UKF are shown in Figure 6.7. The MSE of state estimates obtained using UKF is shown in Figure 6.8. The error in the state estimates obtained when data assimilation is not performed is also shown for comparison. Since $n = 40$, UKF uses 81 $(2n + 1)$ ensembles.

Next, as shown in Figure 6.9 and Figure 6.10, we reduce the ensemble size and use SVDRRUKF and CDRRUKF with $q = 10, 20, 30$. Although the number of ensembles in SVDRRUKF and CDRRUKF is the same for a fixed value of q, it can be seen that the performance of SVDRRUKF is poor compared to the performance of CDRRUKF for both $q = 20$ and $q = 30$. Moreover, the performance of CDRRUKF with 61 $(q = 30)$ ensemble members is close to the performance of UKF with 81 ensemble members. Figure 6.11 shows the difference in the MSE of state estimates between data-free simulation and the reduced-rank filters with $q = 10$. Positive values indicate the cells and time instants at which estimates from the reduced-rank filters are better than the estimates obtained when data assimilation is not performed, while negative values indicate the cells and time instants at which estimates from the reduced-rank filters are worse than the estimates obtained from data-free simulation.

Next, since the process noise covariance Q_k is often not readily available, we assume that we have a poor estimate of the process noise covariance. Specifically, we replace Q_k in (6.3.12) and (6.4.16) by $\hat{Q}_k$, where $\hat{Q}_k = \alpha I$ for all $k \geqslant 0$. Figure 6.12 shows the time-averaged MSE of state estimates obtained using SVDRRUKF and CDRRUKF with $q = 10$ and $q = 20$ for various values of α between 0.001 and 100. The error in state estimates are averaged between 35 sec and 50 sec. It can be seen that, for all values of α, the performance of CDRRUKF is superior to the performance of SVDRRUKF. In fact, CDRRUKF with 21 ensemble members $(q = 10)$ consistently

outperforms SVDRRUKF with 41 ensemble members ($q = 20$).

6.7 Simulation Example : 1-D Compressible Flow Model

Finally, we consider state estimation of one-dimensional hydrodynamic flow based on a finite volume model. The flow of an inviscid, compressible fluid along a one-dimensional channel is governed by Euler's equations

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}\varrho v + w_\rho = 0, \tag{6.7.1}$$

$$\frac{d}{dt}\left(\frac{p}{\rho^\gamma}\right) + w_p = 0, \tag{6.7.2}$$

$$\rho\frac{\partial v}{\partial t} + \rho v\frac{\partial v}{\partial x} + \frac{\partial p}{\partial x} + w_v = 0, \tag{6.7.3}$$

where $\rho \in \mathbb{R}$ is the density, $v \in \mathbb{R}$ is the velocity, $p \in \mathbb{R}$ is the pressure of the fluid, $\gamma = \frac{5}{3}$ is the ratio of specific heats of the fluid, and w_ρ, w_v, and w_p are the unmodeled source terms that affect the density, pressure and velocity of the flow. Due to the presence of coupled partial differential equations, it is generally difficult to obtain closed-form solutions of (6.7.1)-(6.7.3). However, a discrete-time model of the flow can be obtained by using a finite-volume-based spatial and temporal discretization.

Assume that the channel consists of n identical cells. For all $i = 1, \ldots, n$, let $\rho_k^{[i]}$, $v_k^{[i]}$, and $p_k^{[i]}$ be the density, velocity, and pressure in the ith cell at time step k. For all $i = 1 \ldots, n$, define $U^{[i]} \in \mathbb{R}^3$ by

$$U^{[i]} = \begin{bmatrix} \rho_k^{[i]} & v_k^{[i]} & p_k^{[i]} \end{bmatrix}^{\mathrm{T}}. \tag{6.7.4}$$

We use a second-order Rusanov scheme [67] to discretize (6.7.1)-(6.7.3), and obtain a discrete-time model

$$U_{k+1}^{[i]} = f^{[i]}(U_k^{[i-2]}, \ldots, U_k^{[i+2]}, k) + W_k^{[i]}, \tag{6.7.5}$$

where $W_k^{[i]} \in \mathbb{R}^3$ represents unmodeled source terms that affects the density, velocity and pressure of the fluid in the ith cell, and is assumed to be zero-mean white Gaussian process noise with covariance matrix $Q^{[i]} \in \mathbb{R}^{3\times 3}$. Furthermore, for all $k \geqslant 0$, $U_k^{[-1]}$, $U_k^{[0]}$, $U_k^{[n+1]}$, and $U_k^{[n+2]}$ denote the boundary conditions. Next, define the state-vector $x_k \in \mathbb{R}^{3n}$ by

$$x_k \triangleq \left[\begin{array}{ccc} (U_k^{[1]})^{\mathrm{T}} & \cdots & (U_k^{[n]})^{\mathrm{T}} \end{array} \right]^{\mathrm{T}}. \tag{6.7.6}$$

so that (6.7.5) yields a discrete-time model of the form (6.2.1), where $w_k \in \mathbb{R}^{3n}$ is defined by

$$w_k \triangleq \left[\begin{array}{ccc} (W_k^{[1]})^{\mathrm{T}} & \cdots & (W_k^{[n]})^{\mathrm{T}} \end{array} \right]. \tag{6.7.7}$$

Since $W_k^{[i]}$ is a zero-mean white Gaussian process, (6.7.7) implies that w_k is also a zero-mean white Gaussian process with covariance $Q_k = Q \in \mathbb{R}^{3n\times 3n}$, where

$$Q \triangleq \mathrm{diag}(Q^{[1]}, \ldots, Q^{[n]}). \tag{6.7.8}$$

We assume that measurements of density, velocity and pressure from certain cells are available so that y_k is given by (6.2.2), with $C_k = C$ for all $k \geqslant 0$, where

$$C \triangleq \left[\begin{array}{ccc} (C^{[i_1]})^{\mathrm{T}} & \cdots & (C^{[i_p]})^{\mathrm{T}} \end{array} \right]^{\mathrm{T}}, \tag{6.7.9}$$

v_k is zero-mean white Gaussian noise with covariance matrix $R = 0.01 I_{3p\times 3p}$, and for all $i \in \{1, \ldots, n\}$, $C^{[i]} \in \mathbb{R}^{3\times 3n}$ is defined by

$$C^{[i]} \triangleq \left[\begin{array}{ccc} 0_{3\times 3(n-i)} & I_{3\times 3} & 0_{3\times 3(i-1)} \end{array} \right]. \tag{6.7.10}$$

Let $n = 100$ so that $x_k \in \mathbb{R}^{300}$. We assume that the discretized cells are of width 1 m and choose a sampling time of $t_s = 0.2$ s. First, we consider flow along a circular

one-dimensional channel (see Figure 6.10). Hence, the boundary conditions are given by

$$U_k^{[0]} = U_k^{[n]}, \;\; U_k^{[-1]} = U_k^{[n-1]} \;\; U_k^{[n+1]} = U_k^{[1]}, \;\; U_k^{[n+2]} = U_k^{[2]}, \;\; k \geqslant 0. \tag{6.7.11}$$

We assume that unknown source terms affect cells with indices 15, 25, 75, and 85 and therefore

$$Q^{[i]} = \begin{cases} 0.1 I_3, & \text{if } i \in \{15, 25, 75, 85\}, \\ 0, & \text{else.} \end{cases} \tag{6.7.12}$$

Furthermore, we use measurements of density, velocity, and pressure from cells 50 and 51 to estimate the flow variables in other regions. We assume that the nominal initial conditions are given by

$$\rho_0^{[i]} = \begin{cases} 1.5, & \text{if } \; i \in \{45, \ldots, 55\}, \\ 1, & \text{else.} \end{cases}, \tag{6.7.13}$$

$$v_0^{[i]} = 0, \;\; i = 1, \ldots, n, \tag{6.7.14}$$

$$p_0^{[i]} = \begin{cases} 1.5, & \text{if } \; i \in \{45, \ldots, 55\}, \\ 1, & \text{else.} \end{cases}. \tag{6.7.15}$$

We initialize the estimators with the nominal initial condition and initialize the truth model by adding random perturbations to the nominal initial condition.

The evolution of density between 50 sec and 100 sec is shown in Figure 6.14. The estimates from data-free simulation and UKF are also shown. Figure 6.15 shows the total MSE in the state-estimates when data assimilation is performed using UKF. The error in the state estimates when data assimilation is not performed is also shown in the same figure. Note that we consider 100 cells and the dimension n of

the state-vector is 300, and therefore UKF uses 601 $(2n+1)$ ensembles. Thus, we update the flow variable in 60100 cells and hence UKF is computationally expensive.

Next, we reduce the ensemble size and perform data assimilation using SVDRRUKD and CDRRUKF. Figure 6.16 shows the total MSE in the state-estimates obtained using SVDRRUKF with $q = 100, 50, 25$. Note that the dimension of the state-vector $n = 300$ and degradation in performance can be seen only when $q = 25$. The error in the state-estimates obtained using CDRRUKF with $q = 100, 50, 25$ is shown in Figure 6.17. The performance of CDRRUKF for all values of q is close to that of UKF. The difference in the MSE of state estimates between data-free simulation and the reduced-rank filters with $q = 15$ is shown in Figure 6.18. Positive values indicate the cells and time instants at which estimates from the reduced-rank filters are better than the estimates obtained when data assimilation is not performed, while negative values indicate the cells and time instants at which estimates from the reduced-rank filters are worse than the estimates obtained from data-free simulation.

Finally, Figure 6.19 shows the performance of SVDRRUKF and CDRRUKF for $q = 200, 150, 100, 50, 25, 15$. The normalized computational time and normalized estimation accuracy of the reduced-rank filters is shown. It can be seen that even with $q = 15$, the performance of CDRRUKF is close to that of UKF although CDRRUKF with $q = 15$ takes about 1/5th of the time taken by UKF. However, the performance of SVDRRUKF with $q = 15$ is worse than that of data-free simulation.

6.8 Ensemble Transformation

Note that $H^{\mathrm{f}}_{\mathrm{s},k}$ and $H^{\mathrm{f}}_{\mathrm{c},k}$ that satisfy (6.3.9) and (6.4.13), respectively, are not unique. Let $S \in \mathbb{R}^{n\times q}$, where $q \leqslant n$, $C \in \mathbb{R}^{p\times n}$, and $R \in \mathbb{R}^{p\times p}$ be positive definite.

Assume that $H \in \mathbb{R}^{q\times q}$ satisfies

$$HH^{\mathrm{T}} = I - (CS)^{\mathrm{T}}\left(CS(CS)^{\mathrm{T}} + R\right)^{-1} CS. \tag{6.8.1}$$

In fact if $H = \hat{H}$ satisfies (6.8.1), then for all unitary matrix $U \in \mathbb{R}^{q\times q}$, $H = \hat{H}U$ also satisfies (6.8.1). Note that (6.8.1) resemble (6.3.9) and (6.4.13). A comparison of the performance of ensemble-based filters for different choices of H is performed in [22]. Note that certain choices of H ensure that $\sum_{i=0}^{q} \mathrm{col}_i(SH) = 0$ whenever $\sum_{i=0}^{q} \mathrm{col}_i(S) = 0$, where $\mathrm{col}_i(M)$ denotes the ith column of a matrix M. However, in SVDRRUKF and CDRRUKF, $\sum_{i=0}^{q} S^{\mathrm{f}}_{\mathrm{s},i,k}$ and $\sum_{i=0}^{q} S^{\mathrm{f}}_{\mathrm{c},i,k}$ may not be equal to zero because of the rank reduction step (6.3.13) and (6.4.17). Hence, instead of using the results in [22, 77], we use a symmetric positive-negative pairing of the ensembles. Specifically, (5.5.3), (6.3.6), and (6.4.10) imply that, for all $i = 1, \ldots, q$, $X^{\mathrm{da}}_{\mathrm{s},i,k} - x^{\mathrm{da}}_{\mathrm{s},k} = -(X^{\mathrm{da}}_{\mathrm{s},q+1-i} - x^{\mathrm{da}}_{\mathrm{s},k})$ and $X^{\mathrm{da}}_{\mathrm{c},i,k} - x^{\mathrm{da}}_{\mathrm{c},k} = -(X^{\mathrm{da}}_{\mathrm{c},q+1-i} - x^{\mathrm{da}}_{\mathrm{c},k})$, and hence

$$\sum_{i=0}^{2q} \gamma_{x,q,i} X^{\mathrm{da}}_{\mathrm{s},i,k} = x^{\mathrm{da}}_{\mathrm{s},k}, \quad \sum_{i=0}^{2q} \gamma_{x,q,i} X^{\mathrm{da}}_{\mathrm{c},i,k} = x^{\mathrm{da}}_{\mathrm{c},k}. \tag{6.8.2}$$

Finally, using the Matrix Inversion Lemma in (6.8.1) yields

$$HH^{\mathrm{T}} = \left(I_q + (CS)^{\mathrm{T}} R^{-1} CS\right)^{-1}. \tag{6.8.3}$$

Hence, either the singular value decomposition or Cholesky factorization of (6.8.3) can be used to obtain H. Since the Cholesky factorization is computationally efficient, we use the Cholesky factorization to obtain $H^{\mathrm{f}}_{\mathrm{s},k}$ and $H^{\mathrm{f}}_{\mathrm{c},k}$ in all our simulations. Note that no rank-reduction is performed while obtaining $H^{\mathrm{f}}_{\mathrm{s},k}$ and $H^{\mathrm{f}}_{\mathrm{c},k}$. Furthermore, our simulations did not show any significant change in the performance when $H^{\mathrm{f}}_{\mathrm{s},k}$ and $H^{\mathrm{f}}_{\mathrm{c},k}$ were obtained using the singular value decomposition.

6.9 Basis Selection for CDRRUKF

The following result given in [42] shows that CDRRUKF is equivalent to UKF for a single time step when C_k has the form (6.4.1).

Proposition 6.9.1 *Assume that C_k has the structure in* (6.4.1)*, and let $\hat{P}^{\mathrm{f}}_{\mathrm{c},k} = P^{\mathrm{f}}_k$. Then, $C_k\hat{P}^{\mathrm{f}}_{\mathrm{c},k} = C_kP^{\mathrm{f}}_k$ and hence, $K_{\mathrm{c},k} = K_k$.*

Note that Proposition 6.9.1 guarantees that CDRRUKF and UKF are equivalent only for a single time step. However, if the dynamics in (6.2.1) is linear and time-invariant, that is, for all $k \geqslant 0$,

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{6.9.1}$$

$$y_k = Cx_k + v_k, \tag{6.9.2}$$

then a basis for the state x can be chosen so that CDRRUKF is equivalent to UKF for $r > 0$ time steps. We first define the observability matrix $\mathcal{O}(A, C) \in \mathbb{R}^{pn\times n}$ by

$$\mathcal{O}(A,C) \triangleq \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}. \tag{6.9.3}$$

Note that for linear systems, $\mathcal{O}(A, C)$ determines the value of the output y_k at future instances in time. Specifically, if $u_k = w_k = v_k = 0$, for all $k \geqslant 0$, then (6.9.1)-(6.9.3) imply that, for all $k \geqslant 0$,

$$\begin{bmatrix} y_k \\ \vdots \\ y_{k+n-1} \end{bmatrix} = \mathcal{O}(A,C)x_k. \tag{6.9.4}$$

Assume that $\mathcal{O}(A, C)$ has the form

$$\mathcal{O}(A, C) = \begin{bmatrix} I_n \\ 0_{(p-1)n \times n} \end{bmatrix}. \tag{6.9.5}$$

Let x_k have entries

$$x_k = \begin{bmatrix} x_{1,k} & \cdots & x_{n,k} \end{bmatrix}^{\mathrm{T}}, \tag{6.9.6}$$

Then, (6.9.4) implies that, for any integer $r > 0$ such that $pr \leqslant n$,

$$\begin{bmatrix} y_k \\ \vdots \\ y_{k+r-1} \end{bmatrix} = \begin{bmatrix} x_{1,k} \\ \vdots \\ x_{pr,k} \end{bmatrix}. \tag{6.9.7}$$

Therefore, the measurements from time step k to $k+r-1$ depend on only the value of the first pr components of the state vector x_k at time step k. The following result is given in [42].

Proposition 6.9.2 *Assume that $\mathcal{O}(A, C)$ has the form*

$$\mathcal{O}(A, C) = \begin{bmatrix} I_n \\ 0_{(p-1)n \times n} \end{bmatrix}. \tag{6.9.8}$$

Let $r > 0$ be an integer such that $pr < n$ and let $q = pr$. Furthermore, assume that $P^{\mathrm{f}}_{\mathrm{c},0} = P^{\mathrm{f}}_0$. Then, for all $k = 0, \ldots, r$, $K_{\mathrm{c},k} = K_k$. If, in addition, $x^{\mathrm{f}}_{\mathrm{c},0} = x^{\mathrm{f}}_0$, then for all $k = 0, \ldots, r$, $x^{\mathrm{f}}_{\mathrm{c},k} = x^{\mathrm{f}}_k$.

Generally, the observability matrix $\mathcal{O}(A, C)$ may not be of the form (6.9.8). However, a suitable change of basis for the state x can be found so that the observability matrix satisfies (6.9.8). Let $T \in \mathbb{R}^{n \times n}$ be invertible, and define $\tilde{A} \triangleq TAT^{-1}$ and $\tilde{C} \triangleq CT^{-1}$. Let $\tilde{x} \triangleq Tx$, so that in the new basis, (6.2.1) can be expressed as

$$\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}u_k + \tilde{w}_k, \tag{6.9.9}$$

$$y_k = \tilde{C}_k\tilde{x}_k + \tilde{v}_k. \tag{6.9.10}$$

If (A, C) is observable, then $(\tilde{A}, \tilde{C})$ is also observable, and there exists an invertible matrix $T \in \mathbb{R}^{n\times n}$ such that $\mathcal{O}(\tilde{A}, \tilde{C})$ satisfies (6.9.8) (see [36]). Hence, for linear dynamics, we use (6.9.9) and (6.9.10) to construct CDRRUKF and perform data assimilation in the new basis so that the observability matrix has the form (6.9.8), and thus ensure that the performance guaranteed in Proposition 6.9.2 is achieved. Moreover, all the results in Section 6.5 are obtained using a basis such that the observability matrix has the form (6.9.8)

Next, we consider systems with nonlinear dynamics. Specifically, we consider nonlinear systems like terrestrial-weather and ocean-climate models, where the state vector represents physical variables like temperature, pressure, and density at specific grid points that discretize a spatial region. For example, in a one-dimensional model, x_k can be expressed as

$$x_k = \begin{bmatrix} x_k^{[1]} & \cdots & x_k^{[n]} \end{bmatrix}^{\mathrm{T}}, \tag{6.9.11}$$

where $x_k^{[i]}$ denotes the physical variable in the ith grid point at time step k. Furthermore, in systems modeled by finite volume schemes, the future value of the physical variable in a particular grid point i depends only on the current value of the physical variables in its neighboring cells. Hence, the dynamics (6.2.1) can be expressed as

$$x_{k+1}^{[i]} = f^{[i]}(x_k^{[i-b]}, \ldots, x_k^{[i+b]}, u_k, k), \quad i = 1, \ldots, n, \tag{6.9.12}$$

and $b > 0$ depends on the order of the finite volume scheme [66, 67]. For example, $b = 2$ in a second-order finite volume scheme.

Next, let y_k denote measurement of the physical variable at a particular grid-point, so that

$$y_k = x_k^{[i_1]} + v_k, \tag{6.9.13}$$

where $i_1 \in \{1, \ldots, n\}$. For nonlinear systems, the notion of an observability matrix is not well developed and is an area of active research [80]. However, it follows from (6.9.12) and (6.9.13) that, if $w_k = v_k = 0$, for all $k \geqslant 0$, then

$$\begin{bmatrix} y_k \\ \vdots \\ y_{k+r-1} \end{bmatrix} = \begin{bmatrix} g_1(x_k^{[i_1-b]}, \ldots, x_k^{[i_1+b]}, u_k, k) \\ \vdots \\ g_r(x_k^{[i_1-rb]}, \ldots, x_k^{[i_1+rb]}, u_k, k) \end{bmatrix}. \tag{6.9.14}$$

Hence, (6.9.14) can be expressed as

$$\begin{bmatrix} y_k \\ \vdots \\ y_{k+r-1} \end{bmatrix} = g(x_k^{[i_1-rb]}, \ldots, x_k^{[i_1+rb]}, u_k, k). \tag{6.9.15}$$

Now define $\tilde{x}_k \in \mathbb{R}^n$ by

$$\tilde{x}_k = \begin{bmatrix} x_k^{[i_1]} & x_k^{[i_1-1]} & x_k^{[i_1+1]} & x_k^{[i_1-2]} & x_k^{[i_1+2]} & \ldots \end{bmatrix}. \tag{6.9.16}$$

Then, (6.9.15) implies that $y_k, \ldots, y_{k+r-1}$ depends on only first $2rb$ components of the state vector $\tilde{x}_k$ at time step k. Hence, while using CDRRUKF for nonlinear systems that are modeled by finite-volume schemes, we choose a basis so that the outputs $y_k, \ldots, y_{k+r-1}$ depend on only the first few components of the state vector. Although it is difficult to obtain rigorous results similar to Proposition 6.9.2 in the nonlinear case, simulation results indicate that choosing such a basis significantly improves the performance of CDRRUKF. Furthermore, we use such a basis in all our simulations in Section 6.6.

6.10 Conclusion

In this chapter, we presented a reduced-rank Unscented Kalman filter based on the Cholesky decomposition. The ensemble members are reinitialized at each time

step using the columns of Cholesky factor of the square root of the pseudo-error covariance matrix. In all the examples that we considered, the Cholesky-based reduced-rank unscented Kalman filter yielded better estimates than its counterpart based on the singular value decomposition. Moreover, the Cholesky-based filter is computationally faster than the filter based on singular value decomposition, and hence is an attractive alternative to existing reduced-rank filters that use the singular value decomposition.

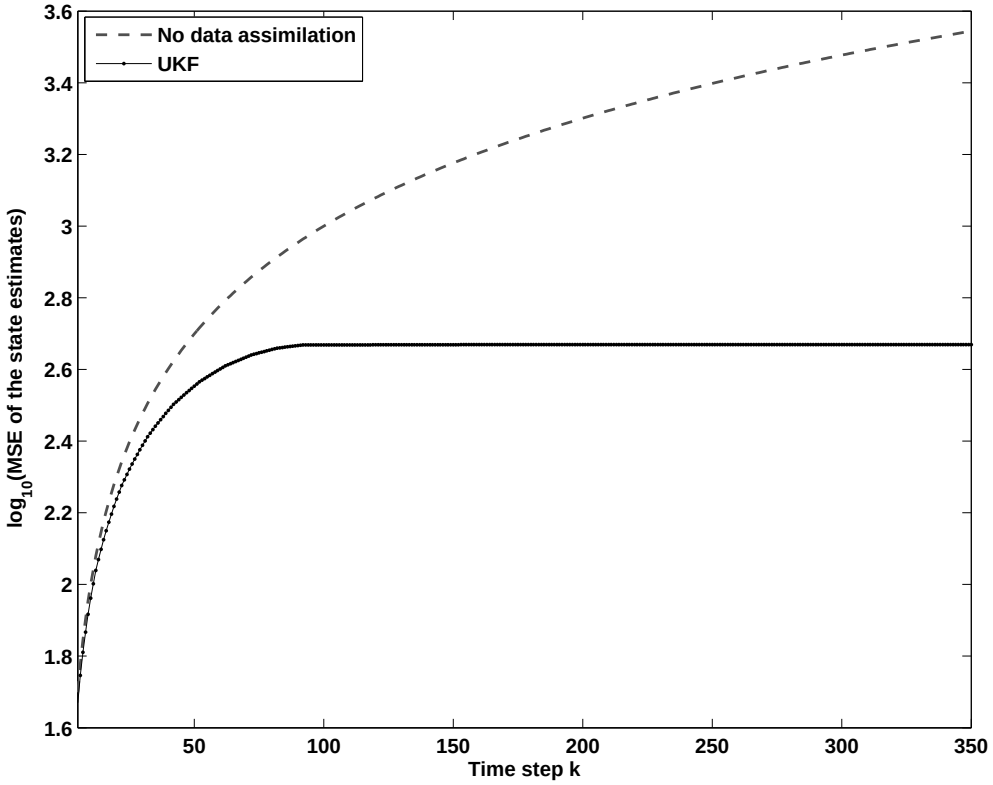


Figure 6.1: MSE of the state estimates obtained from UKF. Since the dynamics are linear, UKF is equivalent to the Kalman filter. The MSE of state estimates when no data assimilation is performed is shown for comparison.

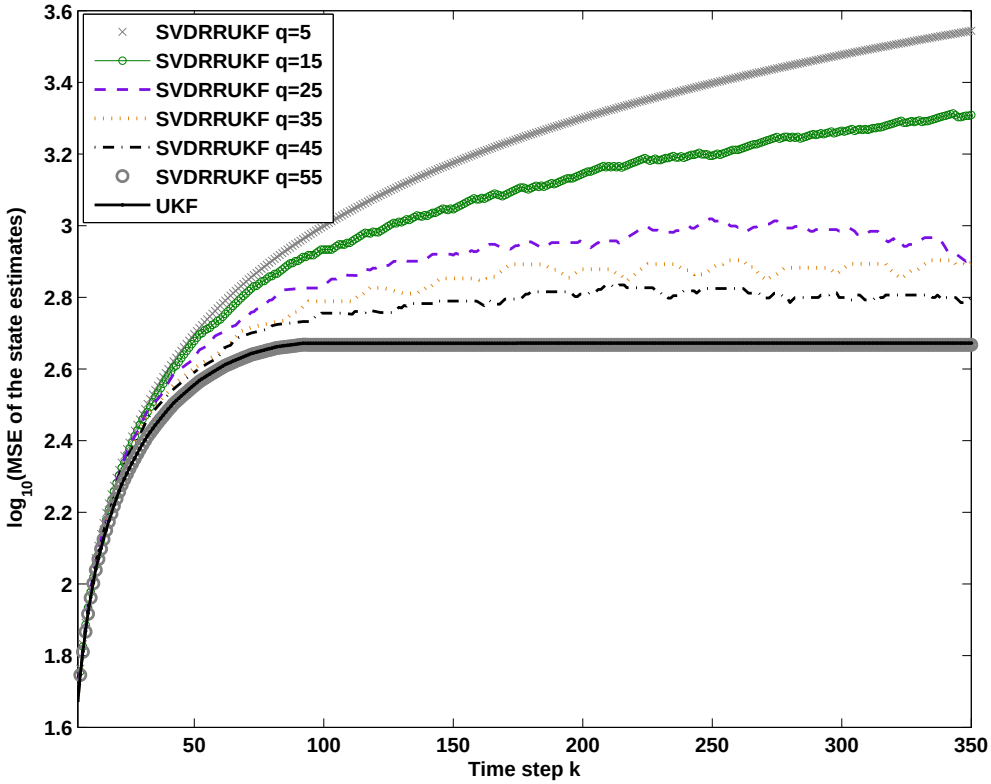


Figure 6.2: MSE of the state estimates obtained from SVDRRUKF for various values of q. SVDRRUKF with $q = 5$ is unstable, while the performance of SVDRRUKF with $q = 55$ is close to the optimal (UKF) performance. Note that SVDRRUKF with $q = 55$ uses 111 ensemble members, whereas UKF uses 201 ensemble members.

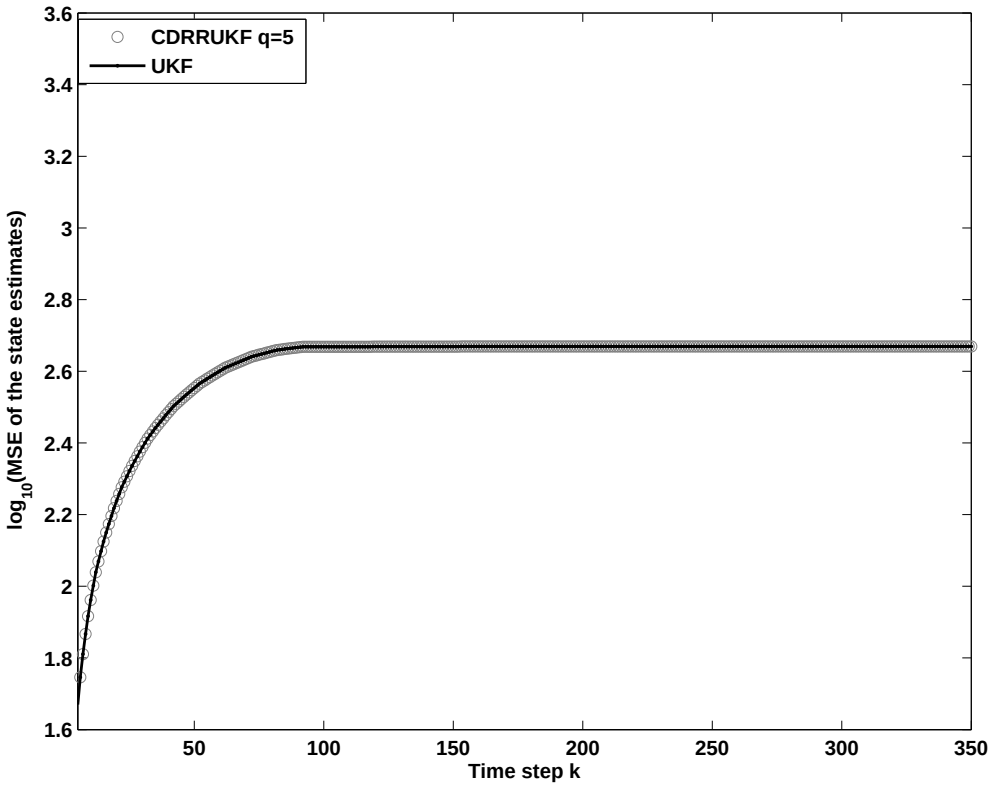


Figure 6.3: MSE of the state estimates obtained from CDRRUKF with $q = 5$. The performance of CDRRUKF with $q = 5$ is close to the optimal (UKF) performance. Note that CDRRUKF with $q = 5$ uses 11 ensemble members, while UKF uses 201 ensemble members.

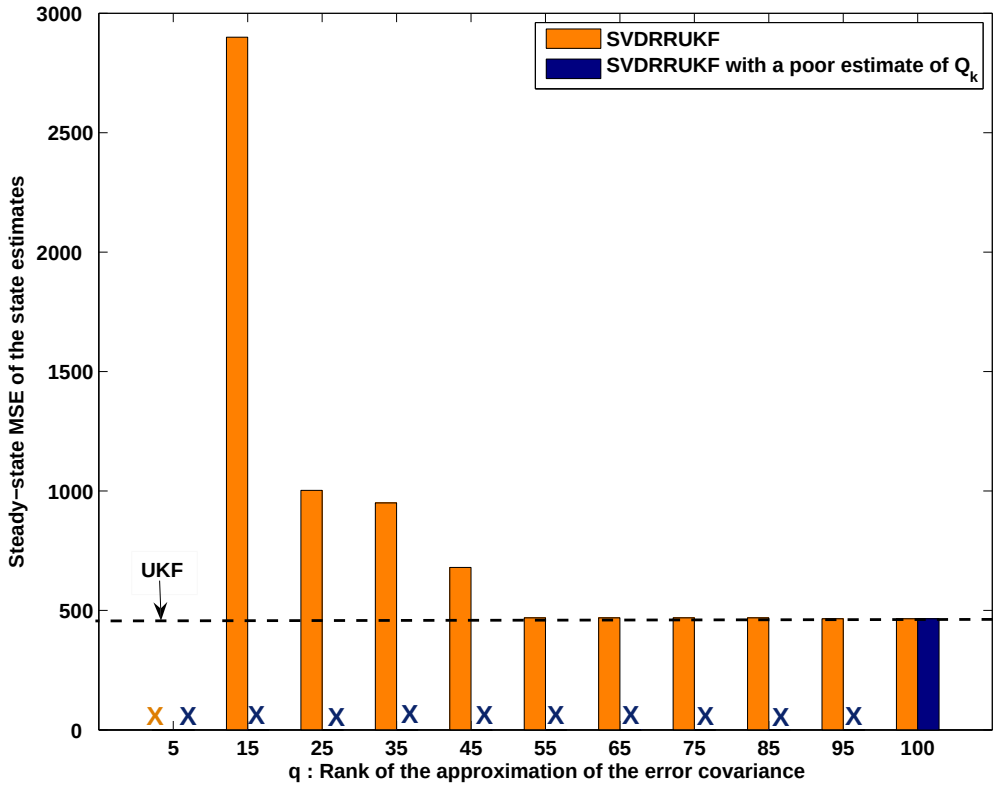


Figure 6.4: Steady-state performance of SVDRRUKF for various values of q between 5 and 100. For each value of q, we perform data assimilation with the exact value of the process noise covariance and with a poor estimate of the process noise covariance. Specifically, we replace Q_k by $\hat{Q}_k$ in (6.3.12), where $\hat{Q}_k = I$ for all $k \geqslant 0$. The performance of UKF is shown for comparison. The X's indicate cases in which the filter is unstable. SVDRRUKF is unstable when $q = 5$, irrespective of the value of the process noise covariance used for data assimilation. When the exact value of the process noise covariance is used for data assimilation, the performance of SVDRRUKF is poor when $q < 55$ and close to optimal for $q > 55$. However, when a poor estimate of the process noise covariance is used for data assimilation, SVDRRUKF is unstable for all $q = 5, \ldots, 95$. These results indicate that SVDRRUKF is sensitive to uncertainties in the estimate of the process noise covariance.

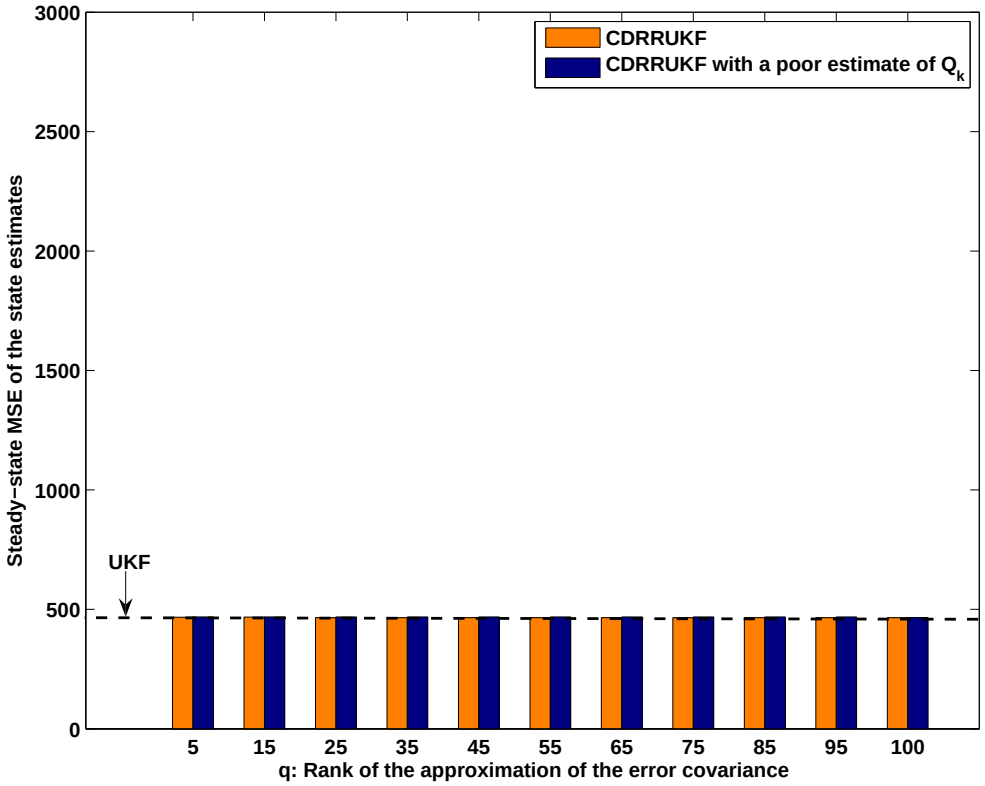


Figure 6.5: Steady-state performance of CDRRUKF for values of q between 5 and 100. We first perform data assimilation using the correct value of the process noise covariance, and then perform data assimilation with a poor estimate of the process noise covariance, that is, we replace Q_k in (6.4.16) by $\hat{Q}_k$, where $\hat{Q}_k = I$ for all $k \geqslant 0$. Note that for $q = 5$, the performance of CDRRUKF is close to optimal, irrespective of the value of the process noise covariance used for data assimilation.

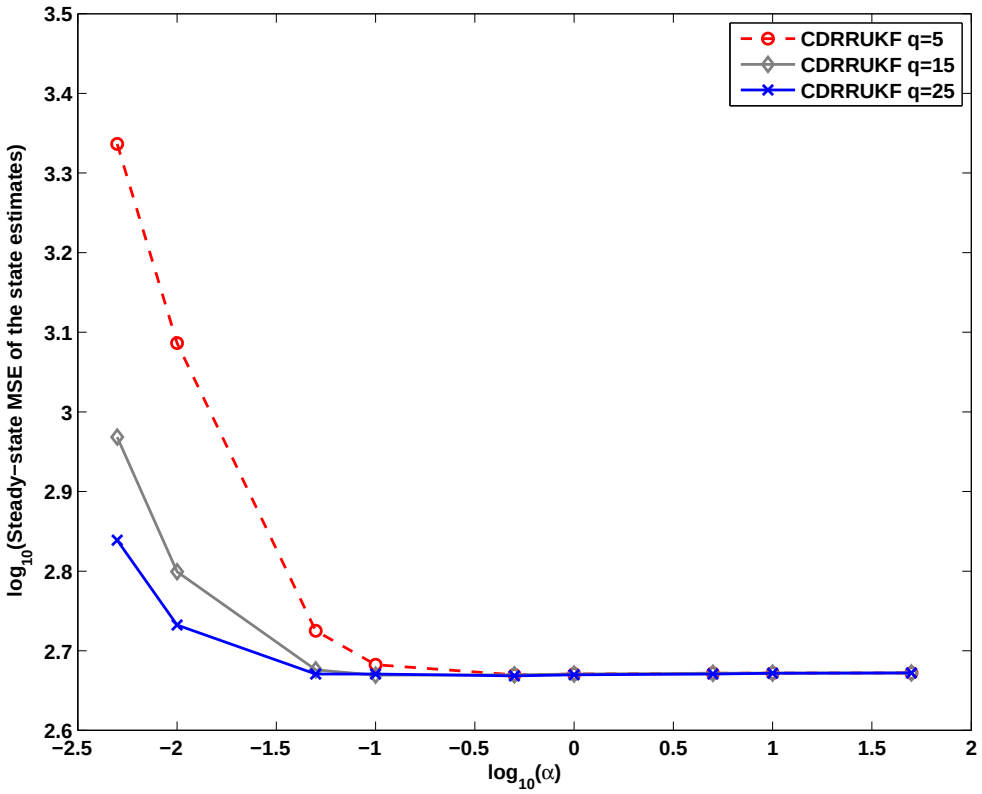


Figure 6.6: Steady-state performance of CDRRUKF with $q = 5, 15, 25$. In all three cases, we use a poor estimate of the process noise covariance for data assimilation, that is, we replace Q_k in (6.4.16) by $\hat{Q}_k$, where $\hat{Q}_k = \alpha I$ for all $k \geqslant 0$. In spite of the presence of an error in the process noise covariance, CDRRUKF is stable and thus robust to uncertainty in the process noise covariance. For a fixed level of uncertainty in the process noise covariance, the performance of CDRRUKF improves when the ensemble size increases. Moreover, for a specific choice of q, the performance improves as α increases. These results suggest that it is advantageous to overestimate the process noise covariance. The performance of SVDRRUKF is not shown since SVDRRUKF is unstable for all values of α and $q = 5, 15, 25$.

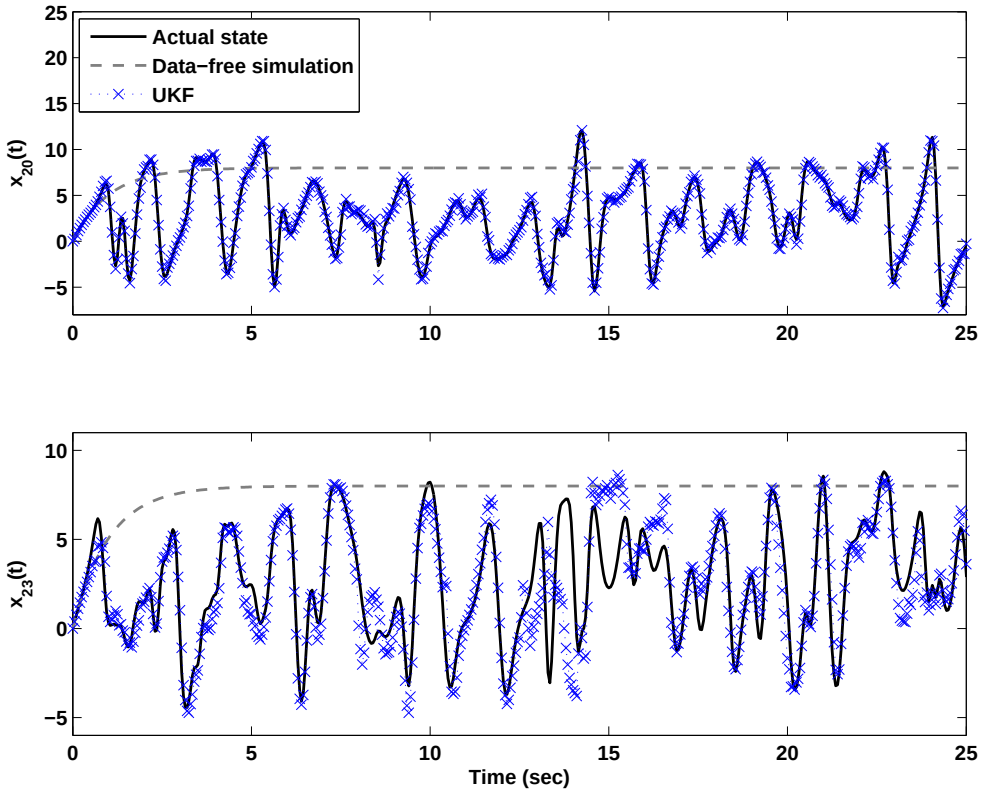


Figure 6.7: Estimates of $x_{20}(t)$ and $x_{23}(t)$ when measurements of $x_{20}(t)$ and $x_{21}(t)$ are used by UKF. The results of data-free simulation are shown for comparison. In both UKF and data-free simulation, all of the initial states are set to zero.

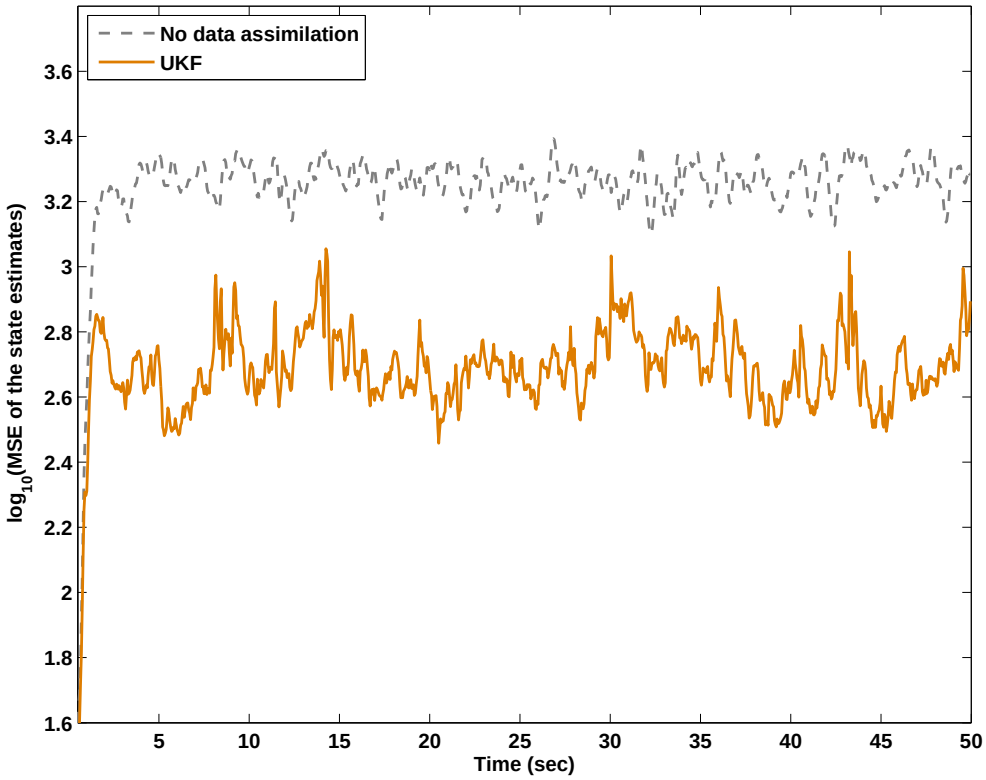


Figure 6.8: MSE of the state estimates obtained using UKF when the exact value of the process noise covariance is used. The MSE of the state estimates obtained from data-free simulation is also shown for comparison.

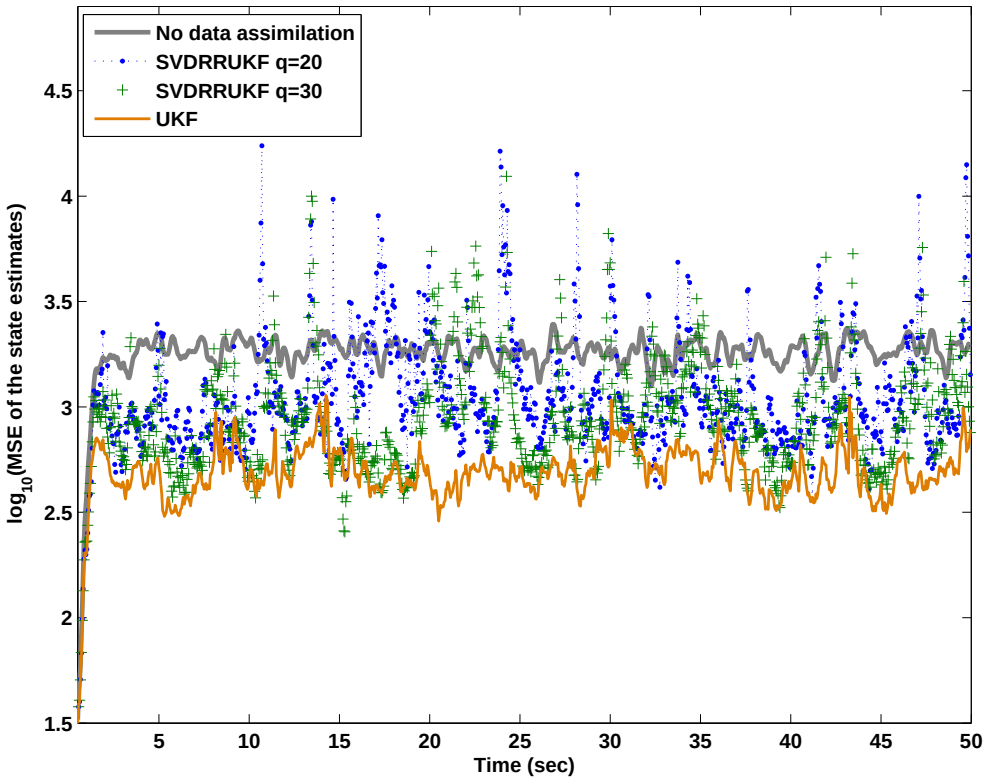


Figure 6.9: MSE of the state estimates obtained using SVDRRUKF with $q = 20, 30$. The error in state estimates when UKF is used and for data-free simulation is shown for comparison. The performance of SVDRRUKF with $q = 20$ and $q = 30$ is poor. In fact, SVDRRUKF with $q = 20$ and $q = 30$ sometimes yields estimates that are worse than estimates obtained from data-free simulation.

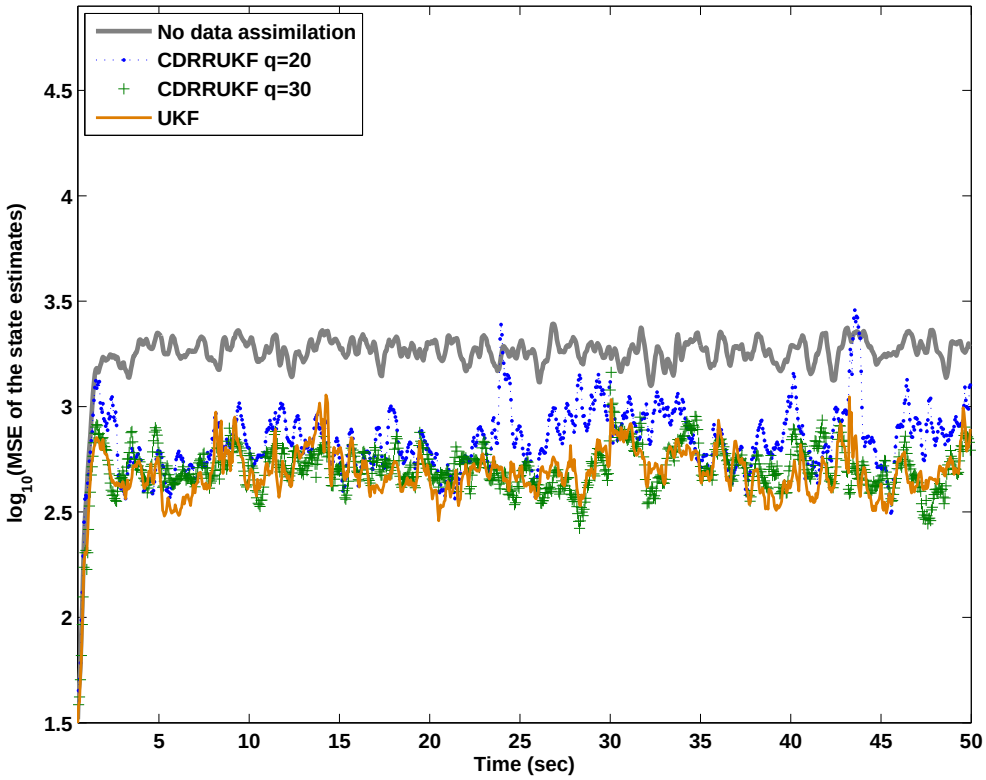


Figure 6.10: Performance of CDRRUKF with $n = 40$ and $q = 20, 30$. Note that the performance of CDRRUKF with $q = 20$ is better than the performance of SVDRRUKF with $q = 30$.

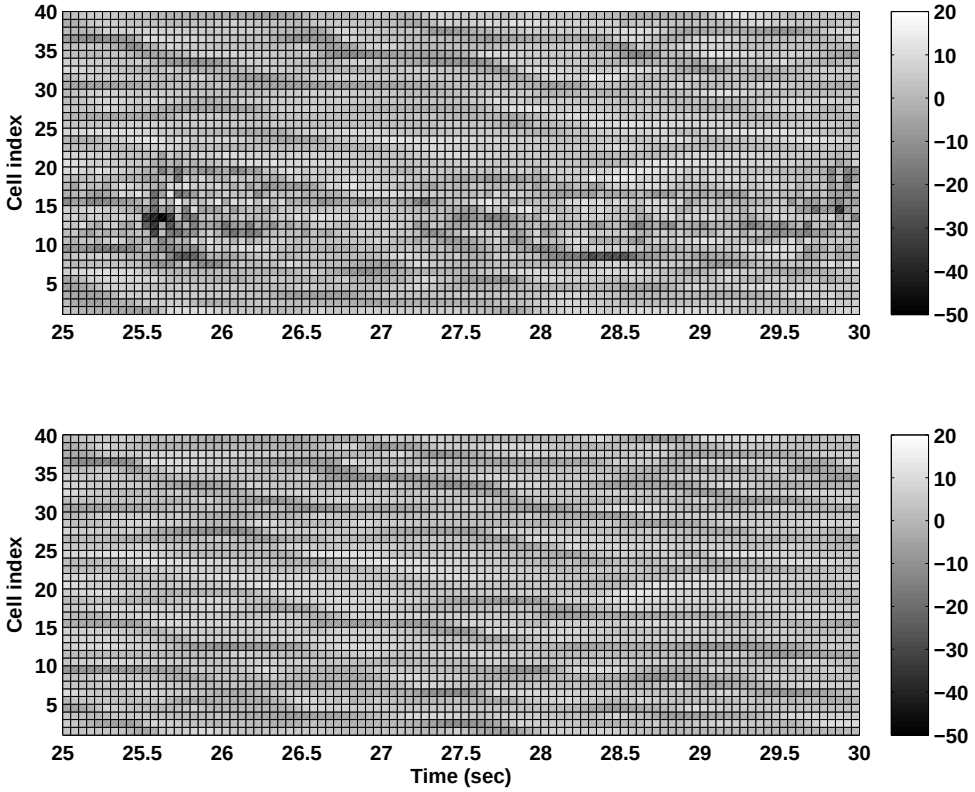


Figure 6.11: Difference in the MSE of state estimates between data-free simulation and SVDRRUKF and CDRRUKF. We use measurements from cells 20 and 21 for data assimilation. For both SVDRRUKF and CDRRUKF, we choose $q = 10$ so that the ensemble size is 21. Regions with positive values indicate the cells and time instants at which the estimates from the reduced-rank filters are better than the estimates obtained when data assimilation is not performed. Alternatively, negative values indicate time instants at which the estimates from SVDRRUKF and CDRRUKF are worse than the estimates obtained from data-free simulation. Note that CDRRUKF with 21 ensembles improves the estimates in most of the cells. However, the estimates from SVDRRUKF are extremely poor in certain cells, for example, in cells $10, \ldots, 15$ between 25.5 sec and 26 sec.

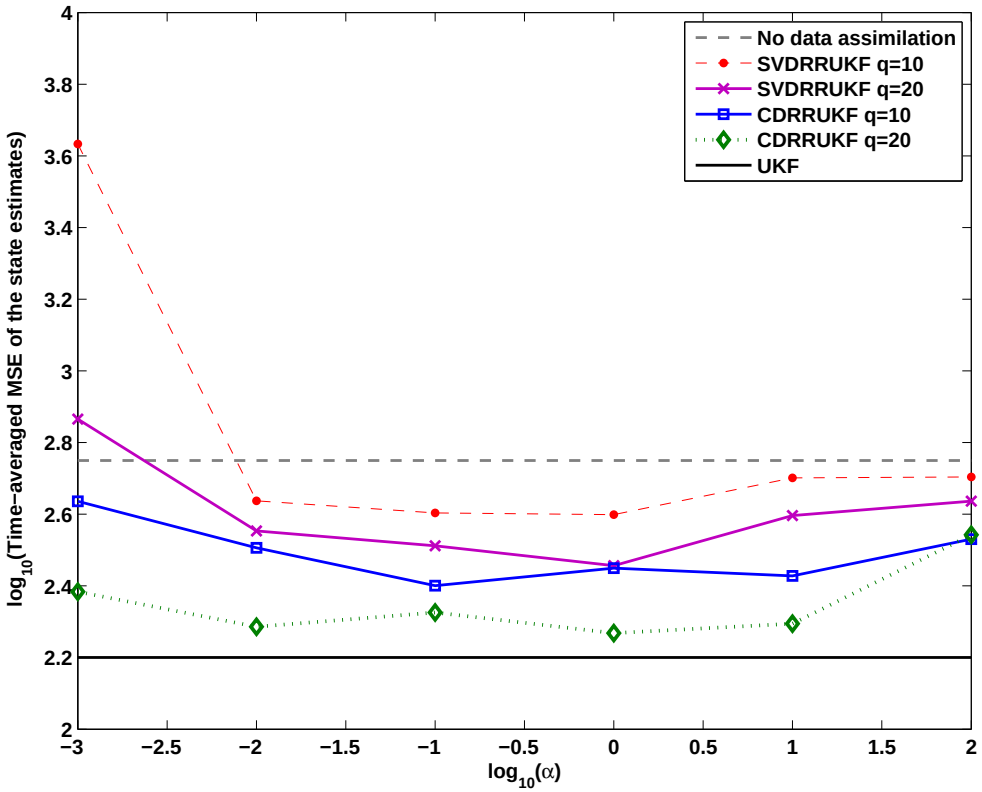


Figure 6.12: Time-averaged MSE of state estimates between 35 sec and 50 sec. The state estimates are obtained using SVDRRUKF and CDRRUKF with $q = 10$ and $q = 20$, and a poor estimate of the process noise covariance. Specifically, we replace Q_k in (6.3.12) and (6.4.16) by $\hat{Q}_k$, where $\hat{Q}_k = \alpha I$ for all $k \geqslant 0$. The error in the state estimates from data-free simulation and UKF is shown for comparison. For all values of α, the performance of CDRRUKF is better than the performance of SVDRRUKF. Furthermore, CDRRUKF is more robust to uncertainties in the estimate of the process noise covariance.

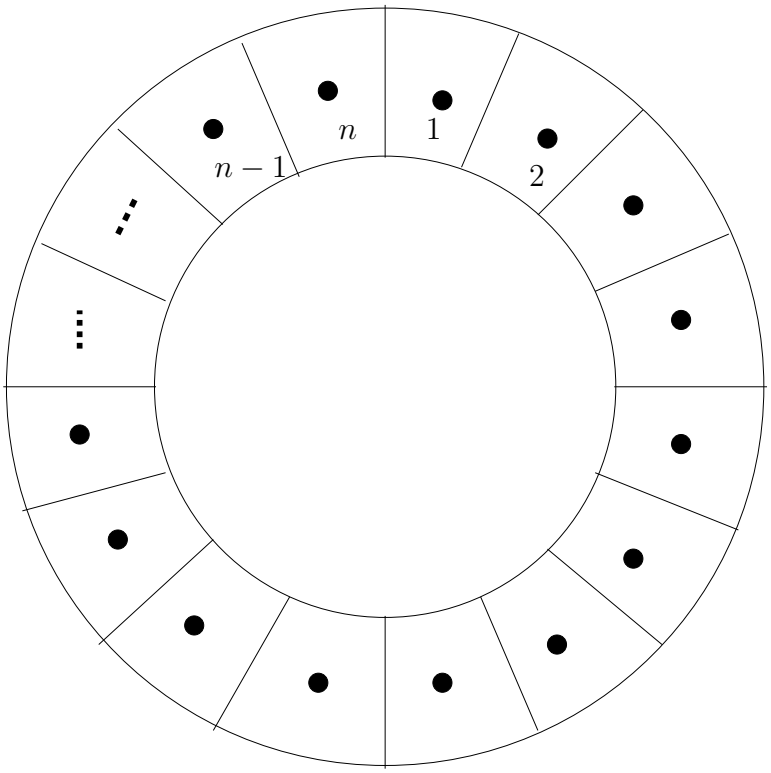


Figure 6.13: One-dimensional circular grid used in the finite volume scheme

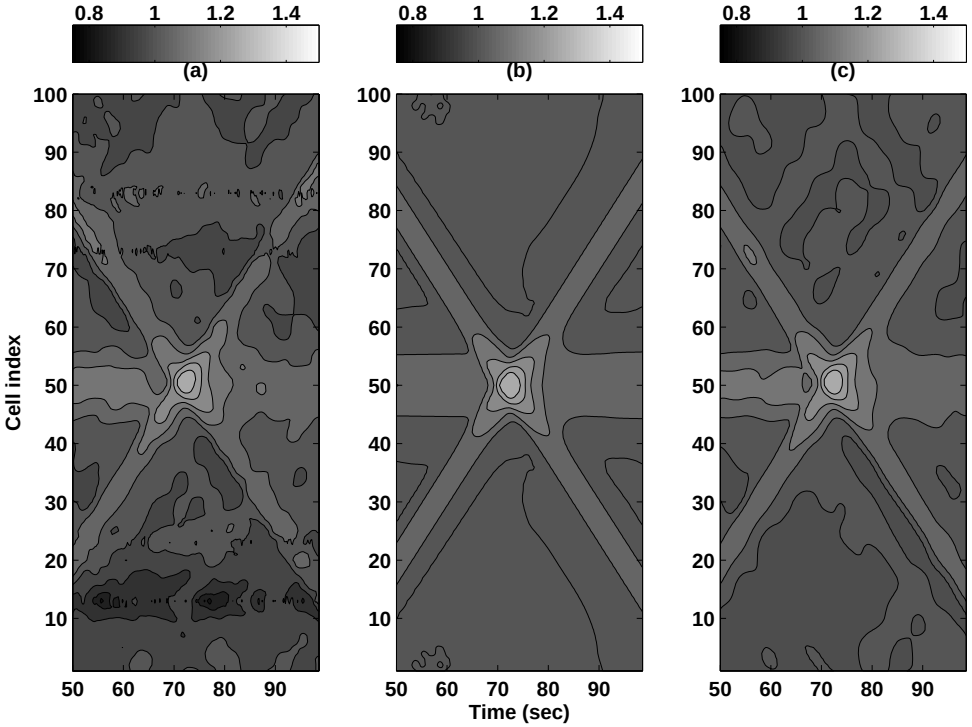


Figure 6.14: Evolution of density between 50 sec and 100 sec. The estimates from (b) data-free simulation and (c) UKF are also shown.

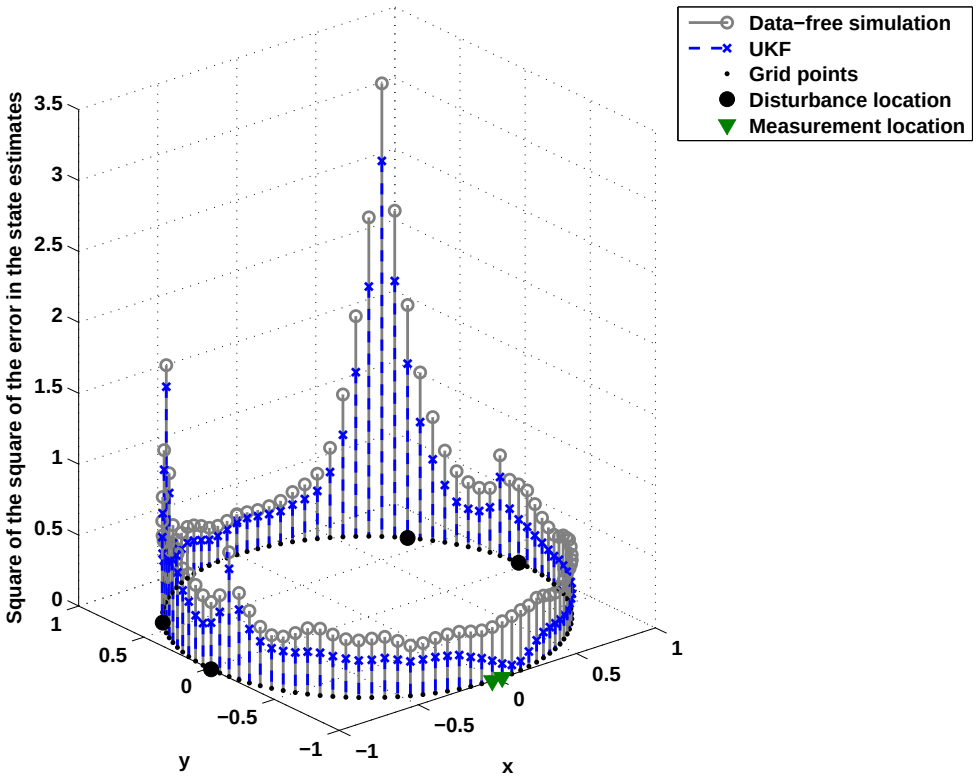


Figure 6.15: Total MSE of the state estimates between 0 sec and 100 sec in a one-dimensional circular channel with periodic boundary conditions obtained using UKF.

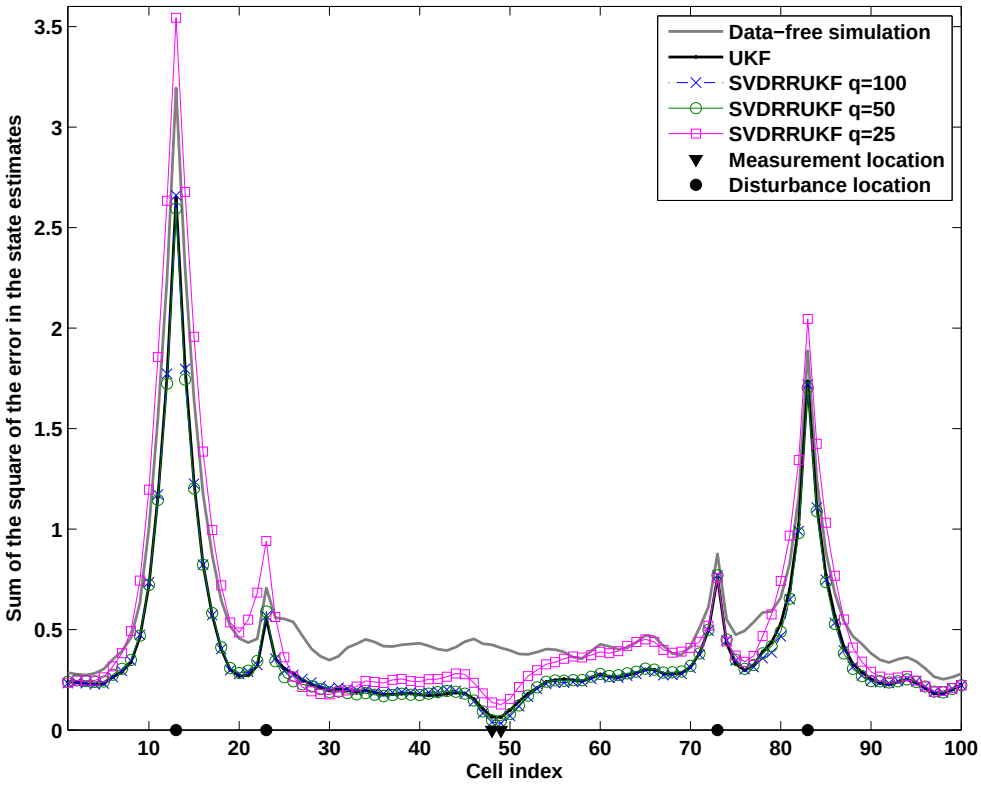


Figure 6.16: Total MSE of the state estimates between 0 sec and 100 sec in a one-dimensional circular channel with periodic boundary conditions. The state estimates are obtained using SVDRRUKF with $q = 100, 50, 25$. The error in the state-estimates in each cell when no data assimilation is performed is also shown as for comparison. The performance of SVDRRUKF with $q = 100$ and $q = 50$ is close to that of UKF. However, the accuracy of the estimates from SVDRRUKF with $q = 25$ is poor in certain cells and in some cases worse than the estimates obtained from data-free simulation.

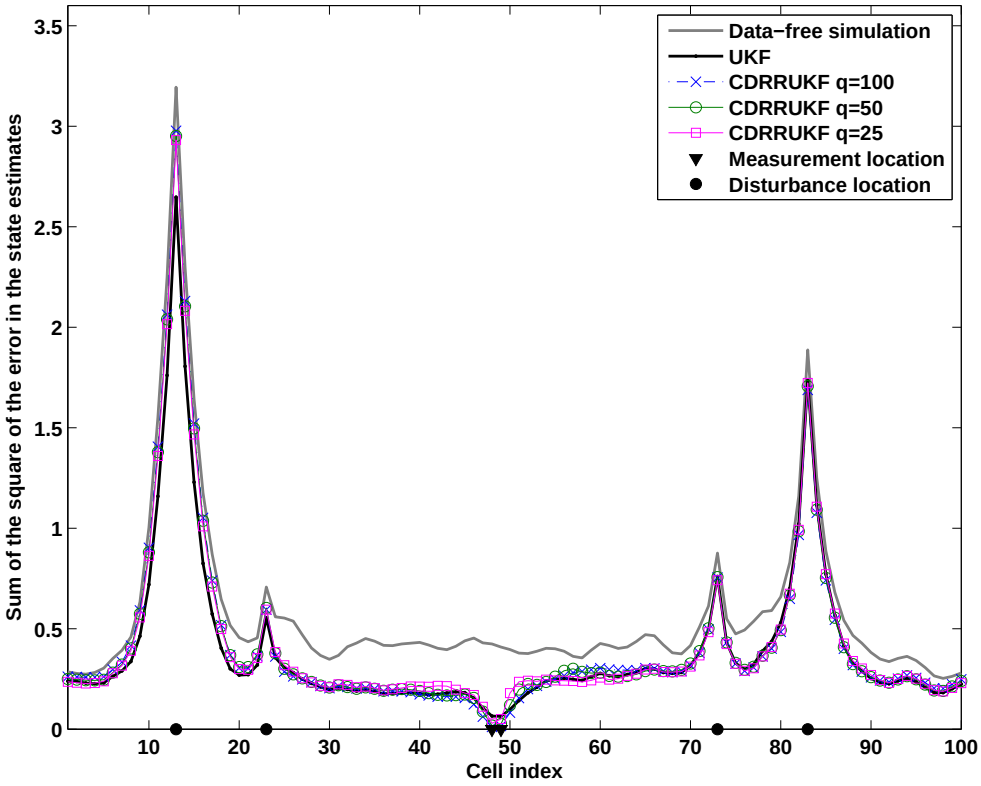


Figure 6.17: Total MSE of the state estimates between 0 sec and 100 sec in a one-dimensional circular channel with periodic boundary conditions. The state estimates are obtained using CDRRUKF with $q = 100, 50, 25$. The error in the state-estimates in each cell when no data assimilation is performed is also shown as for comparison. The performance of CDRRUKF with $q = 100$, $q = 50$, and $q = 25$ is close to that of UKF. Note that the performance of CDRRUKF with $q = 25$ is much better than that of SVDRRUKF with $q = 25$.

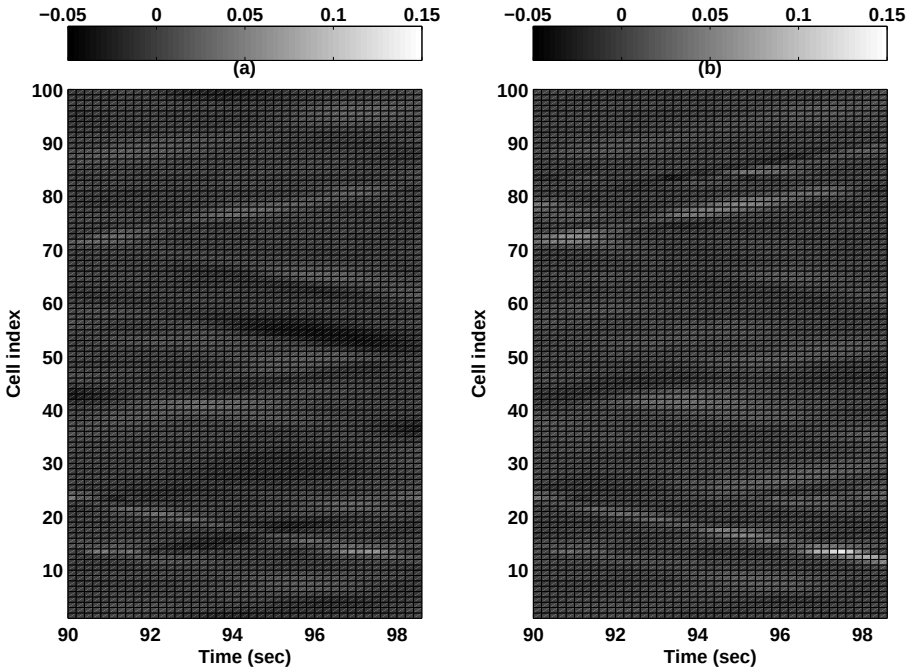


Figure 6.18: Difference in the MSE of state estimates between data-free simulation and SVDRRUKF and CDRRUKF. We use measurements from cells 50 and 51 for data assimilation. For both SVDRRUKF and CDRRUKF, we choose $q = 15$ so that the ensemble size is 31. Regions with positive values indicate the cells and time instants at which the estimates from the reduced-rank filters are better than the estimates obtained when data assimilation is not performed. Alternatively, negative values indicate time instants at which the estimates from SVDRRUKF and CDRRUKF are worse than the estimates obtained from data-free simulation. Note that CDRRUKF with 31 ensembles improves the estimates in most of the cells. However, the estimates from SVDRRUKF are extremely poor in certain cells.

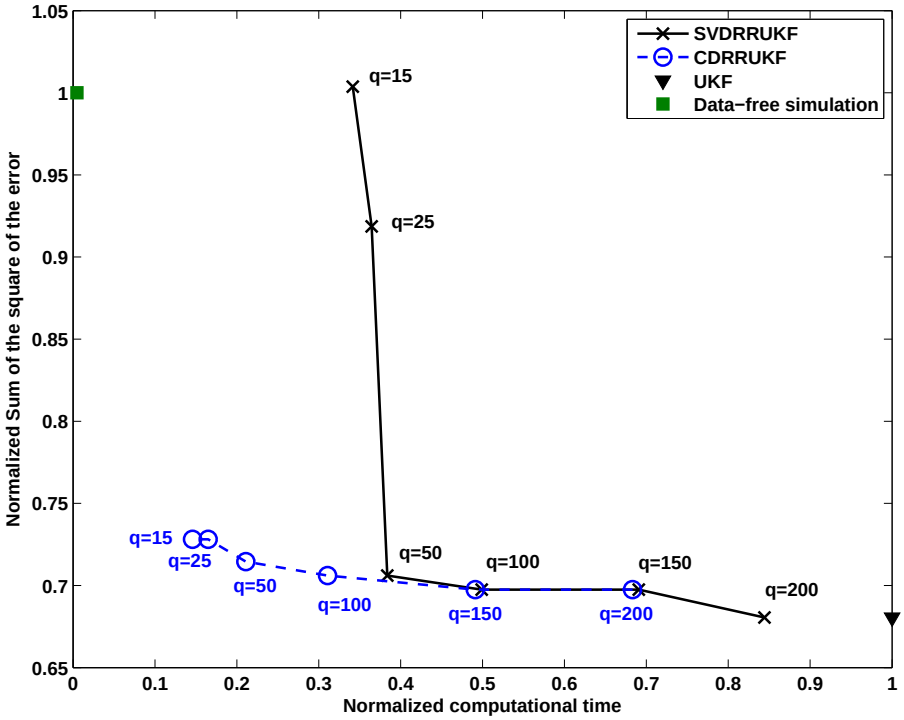


Figure 6.19: Normalized computational time and normalized sum of the square of the error in the state estimates obtained from the various reduced-rank filters. The computational time of UKF is normalized to 1 and the error in the state estimates from data-free simulation is normalized to 1. The performance of CDRRUKF with $q = 25$ is close to UKF, and the computational effort of CDRRUKF with $q = 25$ is only a fraction of that of UKF. For all ensemble sizes, the superiority of CDRRUKF over SVDRRUKF in terms of estimation accuracy and computational effort is clearly seen.

CHAPTER VII

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

In the previous chapter, we reduced the number of ensembles of the unscented Kalman filter by propagating a low-rank approximation of the error covariance. In this chapter, we consider yet another approach to reduce the ensemble size. We consider an estimation algorithm that uses the full-order model for propagating the state estimates, but uses a reduced-order model to propagate the error covariance, thus reducing the size of the error covariance matrix used for data assimilation. Specifically, multiple copies of only a specific subset of the state estimate are used to calculate the reduced-order error covariance. Since only a reduced-order pseudo-error covariance is calculated, we compensate for the neglected correlations by using a static estimator gain based on steady-state correlations that can be determined offline. We use this estimation algorithm to perform data assimilation of one-dimensional compressible flow and two-dimensional magnetohydrodynamic flow models. The results in this chapter have been published in [81].

7.1 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 [82, 83]. Although the literature on reduced-order filtering extends back several decades [8, 25], 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 [20, 27] based on balanced reduction, and this algorithm is compared to several alternative reduced-order error-covariance propagation algorithms in [9]. 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 this chapter we extend the approaches considered in [9] to nonlinear systems by using the unscented Kalman filter [19]. 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 [9] 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. Extended Kalman filter and state-dependent Riccati equation techniques were applied to these problems in [50, 57, 85]. Finally, we consider a two-dimensional finite-volume magnetohydrodynamic (MHD) simulation using the BATSRUS MHD code developed in [84].

7.2 Localized Unscented Kalman Filter (LUKF)

Consider the discrete-time nonlinear system with dynamics

$$x_{k+1} = f(x_k, u_k, k) + w_k \tag{7.2.1}$$

and measurements

$$y_k = h(x_k, k) + v_k, \tag{7.2.2}$$

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.

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 (see Appendix A). 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. Assume that the state $x_k \in \mathbb{R}^n$ has components

$$x_k = \begin{bmatrix} x_{\mathrm{L},k}^{\mathrm{T}} & x_{\mathrm{E},k}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{7.2.3}$$

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

$$y_k = h(x_{\mathrm{L},k}, k) + v_k. \tag{7.2.4}$$

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

$$Q_k = \begin{bmatrix} Q_{\mathrm{L},k} & Q_{\mathrm{LE},k} \\ Q_{\mathrm{LE},k}^{\mathrm{T}} & Q_{\mathrm{E},k} \end{bmatrix}, P_0^{\mathrm{f}} = \begin{bmatrix} P_{\mathrm{L},0}^{\mathrm{f}} & P_{\mathrm{LE},0}^{\mathrm{f}} \\ (P_{\mathrm{LE},0}^{\mathrm{f}})^{\mathrm{T}} & P_{\mathrm{E},0}^{\mathrm{f}} \end{bmatrix}. \tag{7.2.5}$$

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

$$P_k^{\mathrm{f}} = \begin{bmatrix} P_{\mathrm{L},k}^{\mathrm{f}} & 0 \\ 0 & 0 \end{bmatrix}, \tag{7.2.6}$$

where $P^{\mathrm{f}}_{\mathrm{L},k} \in \mathbb{R}^{n_{\mathrm{L}} \times n_{\mathrm{L}}}$ represents the covariance of error corresponding to the state $x_{\mathrm{L},k}$. Hence, it follows from (5.5.3) and (7.2.6) that if $X^{\mathrm{f}}_k = \Psi(x^{\mathrm{f}}_k, P^{\mathrm{f}}_k, \alpha)$ then for $i = n_{\mathrm{L}} + 1, \ldots, n, n + n_{\mathrm{L}} + 1, \ldots, 2n$,

$$X^{\mathrm{f}}_{i,k} = X^{\mathrm{f}}_{1,k} = x^{\mathrm{f}}_k. \tag{7.2.7}$$

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 (7.2.6) 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

$$x^{\mathrm{da}}_{\mathrm{L},k} = x^{\mathrm{f}}_{\mathrm{L},k} + K_{\mathrm{L},k}(y_k - y^{\mathrm{f}}_k), \tag{7.2.8}$$

$$x^{\mathrm{da}}_{\mathrm{E},k} = x^{\mathrm{f}}_{\mathrm{E},k} \tag{7.2.9}$$

$$y^{\mathrm{f}}_k = h(x^{\mathrm{f}}_{\mathrm{L},k}, k), \tag{7.2.10}$$

$$X^{\mathrm{da}}_{\mathrm{L},k} = \Psi(x^{\mathrm{da}}_{\mathrm{L},k}, P^{\mathrm{da}}_{\mathrm{L},k}, \alpha), \tag{7.2.11}$$

$$P^{\mathrm{da}}_{\mathrm{L},k} = P^{\mathrm{f}}_{\mathrm{L},k} - K_{\mathrm{L},k} P_{yy,k} K^{\mathrm{T}}_{\mathrm{L},k}, \tag{7.2.12}$$

where

$$K_{\mathrm{L},k} = P_{x_{\mathrm{L}}y,k} P^{-1}_{yy,k}, \tag{7.2.13}$$

$$P_{x_{\mathrm{L}}y,k} = \sum_{i=0}^{2n_{\mathrm{L}}} \gamma_i (X^{\mathrm{f}}_{\mathrm{L},i,k} - x^{\mathrm{f}}_{\mathrm{L},k})(Y^{\mathrm{f}}_{i,k} - y^{\mathrm{f}}_k)^{\mathrm{T}}, \tag{7.2.14}$$

$$P_{yy,k} = \sum_{i=0}^{2n_{\mathrm{L}}} \gamma_i (Y^{\mathrm{f}}_{i,k} - y^{\mathrm{f}}_k)(Y^{\mathrm{f}}_{i,k} - y^{\mathrm{f}}_k)^{\mathrm{T}} + R_k, \tag{7.2.15}$$

$$Y^{\mathrm{f}}_{i,k} = h(X^{\mathrm{f}}_{\mathrm{L},i,k}, k), \tag{7.2.16}$$

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 (7.2.8)-(7.2.9) 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

$$X^{\mathrm{da}}_{i,k} \triangleq \begin{bmatrix} X^{\mathrm{da}}_{\mathrm{L},i,k} \\ x^{\mathrm{da}}_{\mathrm{E},k} \end{bmatrix}, \tag{7.2.17}$$

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 (7.2.6) 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 (7.2.17) 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

$$X^{\mathrm{f}}_{i,k+1} = f(X^{\mathrm{da}}_{i,k}, u_k, k). \tag{7.2.18}$$

The forecast estimate of the state x_k is obtained by

$$x^{\mathrm{f}}_{k+1} = \sum_{i=0}^{2n_{\mathrm{L}}} \gamma_i X^{\mathrm{f}}_{i,k+1}. \tag{7.2.19}$$

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

$$X^{\mathrm{f}}_{i,k+1} = \begin{bmatrix} X^{\mathrm{f}}_{\mathrm{L},i,k+1} \\ X^{\mathrm{f}}_{\mathrm{E},i,k+1} \end{bmatrix} \tag{7.2.20}$$

with $X^{\mathrm{f}}_{\mathrm{L},i,k+1} \in \mathbb{R}^{n_{\mathrm{L}}}$ and $X^{\mathrm{f}}_{\mathrm{E},i,k+1} \in \mathbb{R}^{n_{\mathrm{E}}}$. 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

$$P^{\mathrm{f}}_{\mathrm{L},k+1} = \sum_{i=0}^{2n} \gamma_i (X^{\mathrm{f}}_{\mathrm{L},i,k+1} - x^{\mathrm{f}}_{\mathrm{L},k+1})(X^{\mathrm{f}}_{\mathrm{L},i,k+1} - x^{\mathrm{f}}_{\mathrm{L},k+1})^{\mathrm{T}} + Q_{\mathrm{L},k}. \tag{7.2.21}$$

Although (7.2.9) implies that data is not directly injected into the state estimates corresponding to the subspace $x_{\mathrm{E},k}$, it follows from (7.2.17)-(7.2.19) 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 $(2n_{\mathrm{L}}+1)n^2$. Hence, if $n_{\mathrm{L}} \ll n$, then LUKF is computationally efficient compared to UKF.

7.3 Complementary Steady-State Correlation

Although LUKF provides estimates of all of the states, (7.2.9) 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 (7.2.1) and (7.2.2) 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.

7.3.1 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

$$P_{xx} = AP_{xx}A^{\mathrm{T}} + Q. \tag{7.3.1}$$

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

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 (7.2.1)-(7.2.2) so that for $i = 1, \ldots, N$,

$$\begin{aligned} \tilde{x}_{i,k+1} &= f(\tilde{x}_{i,k}, u_k, k) + \tilde{w}_{i,k}, \\ \tilde{y}_{i,k} &= h(\tilde{x}_{i,k}, k) + \tilde{v}_{i,k}, \end{aligned} \tag{7.3.2}$$

where $\tilde{x}_{i,0}$ is a random variable with the specified mean x_0 and variance P_0^{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 $P_{OL,xy}$ and $P_{OL,yy}$ by

$$P_{\mathrm{OL},xy} \triangleq \lim_{k\to\infty} \frac{1}{N-1} \sum_{i=1}^{N} (\tilde{x}_{i,k} - \overline{x}_k)(\tilde{y}_{i,k} - \overline{y}_k)^{\mathrm{T}}, \tag{7.3.3}$$

$$P_{\mathrm{OL},yy} \triangleq \lim_{k\to\infty} \frac{1}{N-1} \sum_{i=1}^{N} (\tilde{y}_{i,k} - \overline{y}_k)(\tilde{y}_{i,k} - \overline{y}_k)^{\mathrm{T}}, \tag{7.3.4}$$

where

$$\overline{x}_k \triangleq \frac{1}{N} \sum_{i=1}^{N} \tilde{x}_{i,k}, \quad \overline{y}_k \triangleq \frac{1}{N} \sum_{i=1}^{N} \tilde{y}_{i,k}. \tag{7.3.5}$$

Alternatively, the unscented transformation can also be used to approximate the steady state open-loop state covariance. Note that the state covariance of (7.2.1) 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 (5.5.6)-(5.5.17) with $K_k = 0$ for all $k \geqslant 0$, and define $P_{\mathrm{OL},xy}$ and $P_{\mathrm{OL},yy}$ by

$$P_{\mathrm{OL},xy} \triangleq \lim_{k\to\infty} P_{xy,k}, \quad P_{\mathrm{OL},yy} \triangleq \lim_{k\to\infty} P_{yy,k}. \tag{7.3.6}$$

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

$$K_{\mathrm{OL}} \triangleq P_{\mathrm{OL},xy} P_{\mathrm{OL},yy}^{-1}. \tag{7.3.7}$$

and let K_{OL} have entries

$$K_{\mathrm{OL}} = \begin{bmatrix} K_{\mathrm{OL,L}} \\ \\ K_{\mathrm{OL,E}} \end{bmatrix}, \tag{7.3.8}$$

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 (7.2.17) - (7.2.21). The analysis step of the LUKFCOLC is given by

$$x_{\mathrm{L},k}^{\mathrm{da}} = x_{\mathrm{L},k}^{\mathrm{f}} + K_{\mathrm{L},k}(y_k - y_k^{\mathrm{f}}), \tag{7.3.9}$$

$$x_{\mathrm{E},k}^{\mathrm{da}} = x_{\mathrm{E},k}^{\mathrm{f}} + K_{\mathrm{OL,E}}(y_k - y_k^{\mathrm{f}}), \tag{7.3.10}$$

$$y_k^{\mathrm{f}} = h(x_{\mathrm{L},k}^{\mathrm{f}}, k), \tag{7.3.11}$$

$$X_{\mathrm{L},k}^{\mathrm{da}} = \Psi(x_{\mathrm{L},k}^{\mathrm{da}}, P_{\mathrm{L},k}^{\mathrm{da}}, \alpha), \tag{7.3.12}$$

$$P_{\mathrm{L},k}^{\mathrm{da}} = P_{\mathrm{L},k}^{\mathrm{f}} - K_{\mathrm{L},k} P_{yy,k} K_{\mathrm{L},k}^{\mathrm{T}}, \tag{7.3.13}$$

where $K_{\mathrm{L},k}$ and $P_{yy,k}$ are defined in (7.2.13) and (7.2.15).

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, (7.3.10) 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.

7.3.2 LUKF with Complementary Closed-Loop Correlation (LUKFC-CLC)

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 (7.3.2) 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 $\tilde{x}_{i,k}^{\mathrm{f}}$ be the estimate of $\tilde{x}_{i,k}$ provided by the ith simulation of LUKF. We approximate the steady-state closed-

loop correlations by

$$P_{\mathrm{CL},xy} \triangleq \lim_{k\to\infty} \frac{1}{N-1}\sum_{i=1}^{N}\left[\tilde{x}_{i,k}-\tilde{x}^{\mathrm{f}}_{i,k}\right]\left[\tilde{y}_{i,k}-h(\tilde{x}^{\mathrm{f}}_{i,k})\right]^{\mathrm{T}}, \tag{7.3.14}$$

$$P_{\mathrm{CL},yy} \triangleq \lim_{k\to\infty} \frac{1}{N-1}\sum_{i=1}^{N}\left[\tilde{y}_{i,k}-h(\tilde{x}^{\mathrm{f}}_{i,k})\right]\left[\tilde{y}_{i,k}-h(\tilde{x}^{\mathrm{f}}_{i,k})\right]^{\mathrm{T}}. \tag{7.3.15}$$

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 (7.3.14) and (7.3.15), 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

$$K_{\mathrm{L}} \triangleq \lim_{k\to\infty} K_{\mathrm{L},k}, \tag{7.3.16}$$

where $K_{\mathrm{L},k}$ is the estimator gain given by (7.2.13) 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 (5.5.6)-(5.5.17) with

$$K_k = \begin{bmatrix} K_{\mathrm{L}} \\ 0 \end{bmatrix}, \tag{7.3.17}$$

for all $k \geqslant 0$, and view the correlations $P_{xy,k}$ and $P_{yy,k}$ in (5.5.11) and (5.5.12) 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

$$P_{\mathrm{CL},xy} = \lim_{k\to\infty} P_{xy,k}, \quad P_{\mathrm{CL},yy} = \lim_{k\to\infty} P_{yy,k}. \tag{7.3.18}$$

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

$$K_{\mathrm{CL}} = P_{\mathrm{CL},xy} P_{\mathrm{CL},yy}^{-1} \tag{7.3.19}$$

with entries

$$K_{\mathrm{CL}} = \begin{bmatrix} K_{\mathrm{CL,L}} \\ \\ K_{\mathrm{CL,E}} \end{bmatrix}, \tag{7.3.20}$$

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 (7.2.17) - (7.2.21), and the analysis step of LUKFCCLC is given by (7.3.9)-(7.3.13) with $K_{\mathrm{OL,E}}$ replaced by $K_{\mathrm{CL,E}}$ in (7.3.10).

Next, we compare the performance of UKF, LUKF, LUKFCOLC, and LUKFCCLC on three different finite volume models.

7.4 One-Dimensional Hydrodynamics

First, we consider state estimation of one-dimensional hydrodynamic flow based on a finite volume model. The flow of an inviscid, compressible fluid along a one-dimensional channel is governed by Euler's equations

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= -\frac{\partial}{\partial x} \varrho v, \\ \frac{d}{dt}\left(\frac{p}{\rho^{\gamma}}\right) &= 0, \\ \rho \frac{\partial v}{\partial t} &= -\rho v \frac{\partial v}{\partial x} - \frac{\partial p}{\partial x}, \end{aligned} \tag{7.4.1}$$

where $\rho \in \mathbb{R}$ is the density, $v \in \mathbb{R}$ is the velocity, $p \in \mathbb{R}$ is the pressure of the fluid, and $\gamma = \frac{5}{3}$ is the ratio of specific heats of the fluid. A discrete-time model of hydrodynamic flow can be obtained by using a finite-volume based spatial and temporal discretization.

Assume that the channel consists of n identical cells (see Figure 7.4). For all $i = 1, \ldots, n$, let $\rho^{[i]}$, $v^{[i]}$, and $p^{[i]}$ be the density, velocity, and pressure in the ith cell, and define $U^{[i]} \in \mathbb{R}^3$ by

$$U^{[i]} = \begin{bmatrix} \rho^{[i]} & m^{[i]} & \mathcal{E}^{[i]} \end{bmatrix}^{\mathrm{T}}, \tag{7.4.2}$$

where the momentum $m^{[i]}$ and energy $\mathcal{E}^{[i]}$ in the ith cell are given by

$$m^{[i]} = \rho^{[i]} v^{[i]}, \;\; \mathcal{E}^{[i]} = \frac{1}{2}\rho^{[i]}(v^{[i]})^2 + \frac{p^{[i]}}{\gamma - 1}. \tag{7.4.3}$$

We use a second-order Rusanov scheme [66] to discretize (7.4.1)-(7.4.1) and obtain a discrete-time model that enables us to update the flow variables at the center of each cell.

The discrete-time state update equation [66] is given by

$$U_{k+1}^{[i]} = U_k^{[i]} - \frac{t_{\mathrm{s}}}{\Delta x}\left[\overline{F}_{\mathrm{Rus},k}^{[i]} - \overline{F}_{\mathrm{Rus},k}^{[i-1]}\right], \tag{7.4.4}$$

where $t_{\mathrm{s}} > 0$ is the sampling time and Δx is the width of each cell, and $\overline{F}_{\mathrm{Rus},k}^{[i]}$ depends on $U_k^{[i-1]}, \ldots, U_k^{[i+2]}$. Hence, $U_{k+1}^{[i]}$ depends on $U_k^{[i-2]}, \ldots, U_k^{[i+2]}$, as expected for a second-order scheme.

Next, define the state vector $x \in \mathbb{R}^{3(n-4)}$ by

$$x \stackrel{\triangle}{=} \begin{bmatrix} (U_k^{[3]})^{\mathrm{T}} & \cdots & (U_k^{[n-2]})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}. \tag{7.4.5}$$

Furthermore, we assume Neumann boundary conditions at cells with indices 1, 2, $n-1$ and n so that, for all $k \geqslant 0$,

$$U_k^{[1]} = U_k^{[2]} = U_k^{[3]}, \;\; U_k^{[n]} = U_k^{[n-1]} = U_k^{[n-2]}. \tag{7.4.6}$$

Let $n = 54$ so that $x \in \mathbb{R}^{150}$. It follows from (7.4.4) that the second-order Rusanov scheme yields a nonlinear discrete-time update model of the form

$$x_{k+1} = f(x_k) + w_k, \tag{7.4.7}$$

where $w_k \in \mathbb{R}^{3(n-4)}$ represents unmodeled drivers and is assumed to be zero-mean white Gaussian process noise with covariance matrix $Q \in \mathbb{R}^{3(n-4)\times 3(n-4)}$, so that the flow variables in only the 5th, 15th, 25th, 35th, and 45th cell are directly affected by w_k. Next, for $i = 3, \ldots, n-2$, define $C^{[i]} \in \mathbb{R}^{3\times 3(n-4)}$

$$C^{[i]} \triangleq \left[\begin{array}{ccc} 0_{3\times 3(n-4-i)} & I_{3\times 3} & 0_{3\times 3(i-1)} \end{array} \right] \tag{7.4.8}$$

so that the measurement $y_k \in \mathbb{R}^6$ of density, momentum and energy at cells with indices 24 and 26 is given by

$$y_k = Cx_k + v_k, \tag{7.4.9}$$

where $C = \left[\begin{array}{cc} (C^{[24]})^{\mathrm{T}} & (C^{[26]})^{\mathrm{T}} \end{array} \right]^{\mathrm{T}}$ and v_k is zero-mean white Gaussian noise with covariance matrix $R = 0.01 I_{6\times 6}$.

We simulate the truth model (7.4.7) with the initial condition $\varrho_0^{[i]} = 1$, $v_0^{[i]} = 0$, and $p_0^{[i]} = 1$ for $i = 1, \ldots, n$ and obtain measurements y_k from (7.4.9). The objective is to estimate the density, momentum, and energy at the cells where measurements of flow variables are unavailable using UKF, LUKF, LUKFCOLC, and LUKFCCLC.

The square root of the sum of the square of the error in the estimates of the energy at cells $1, \ldots, 50$, when measurements y_k are used in the UKF is shown in Figure 7.2. The error in energy estimates when no data assimilation is performed is also shown in the same figure for comparison. Note that the performance of UKF degrades as the distance from the measurement cells 24 and 26 increases. Next, we compare the performance of LUKF for various local grid sizes, that is, we set

$$x_{\mathrm{L}} \triangleq \left[\begin{array}{ccc} (U_k^{[L_1]})^{\mathrm{T}} & \cdots & (U_k^{[L_n]})^{\mathrm{T}} \end{array} \right]^{\mathrm{T}}, \tag{7.4.10}$$

where $(L_1, L_n) \in \{(20, 30), (16, 34), (12, 38)\}$. We choose the subset $x_{\mathrm{L}} \in \mathbb{R}^{3(L_n - L_1 + 1)}$ of x so that x_{L} spans the cells where measurements are available. The square root

of the sum of the square of the error in energy estimates of LUKF is shown in Figure 7.2 for the three different local grid sizes. It can be seen that the performance of the LUKF improves as the size of the local grid where direct data injection is performed increases. Furthermore, even though data is injected directly into only the estimates of the states corresponding to the local grid, LUKF improves the estimates of the states outside this region as well. However, for all three local grid sizes, the performance of UKF is much better than the performance of LUKF because LUKF ignores correlations between the measurement and the states that are outside the local region.

Finally, we obtain the steady-state open-loop and closed-loop error correlations defined in (7.3.6) and (7.3.18), respectively, by using the unscented transformation method. Note that the computational effort of determining the steady-state correlations using the unscented transformation is equivalent to the computational effort of using UKF. However, once the steady-state correlations are determined offline, the computational effort of LUKFCOLC and LUKFCCLC while performing the actual data assimilation is similar to that of LUKF which is significantly lower than the computational effort of UKF.

The square root of the sum of the square of the error in energy estimates when LUKFCOLC and LUKFCCLC are used to perform data assimilation is shown in Figure 7.3. The performance of UKF and LUKF is also shown for comparison. We choose $(L_1, L_n) = (20, 30)$ for LUKF, LUKFCOLC, and LUKFCCLC. It can be seen that using a static gain based on the steady-state correlations improves the performance. Moreover, the performance of LUKFCCLC is better than the performance of LUKFCOLC because LUKFCCLC accounts for the change in the measurement-error correlation when data is injected during estimation.

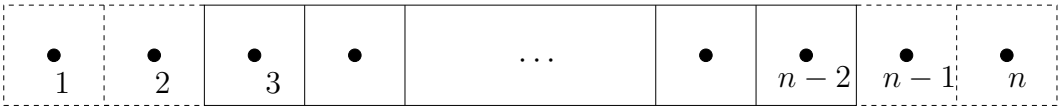


Figure 7.1: One-dimensional grid used in the finite volume scheme

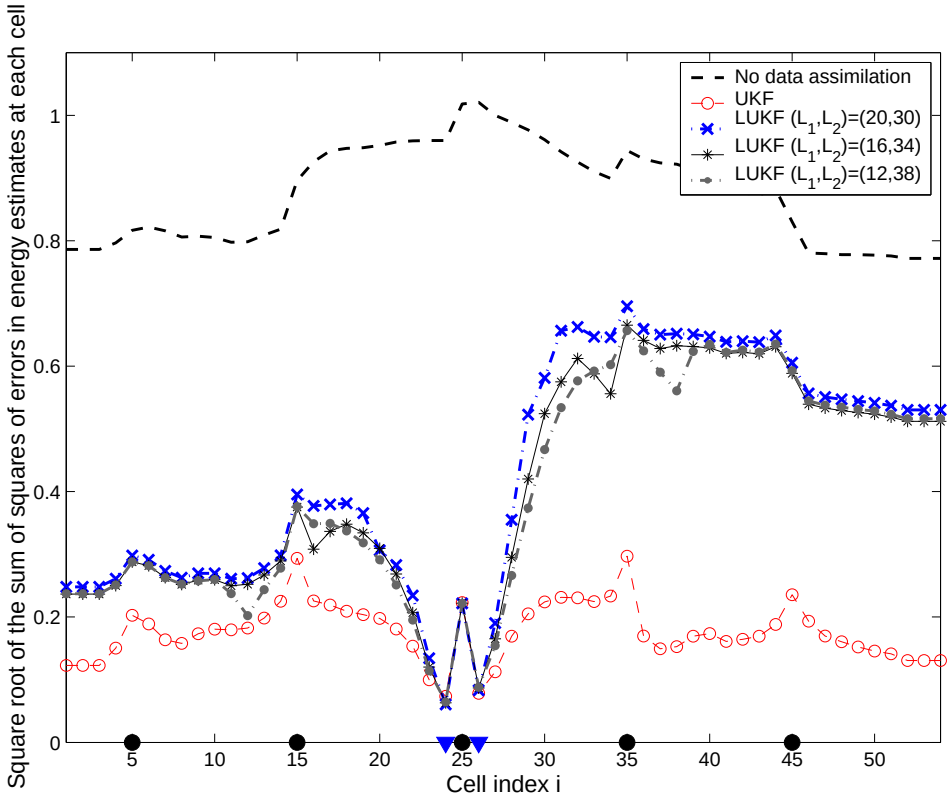


Figure 7.2: Square root of the sum of the square of the error in energy estimates at the various cells using UKF and LUKF with 3 different local grid sizes. Although the local grid size where data is directly injected increases, the performance of LUKF shows only a minor improvement. The cells where disturbance enters the system are indicated by '●' and the cells where measurements are available are indicated by '▼'.

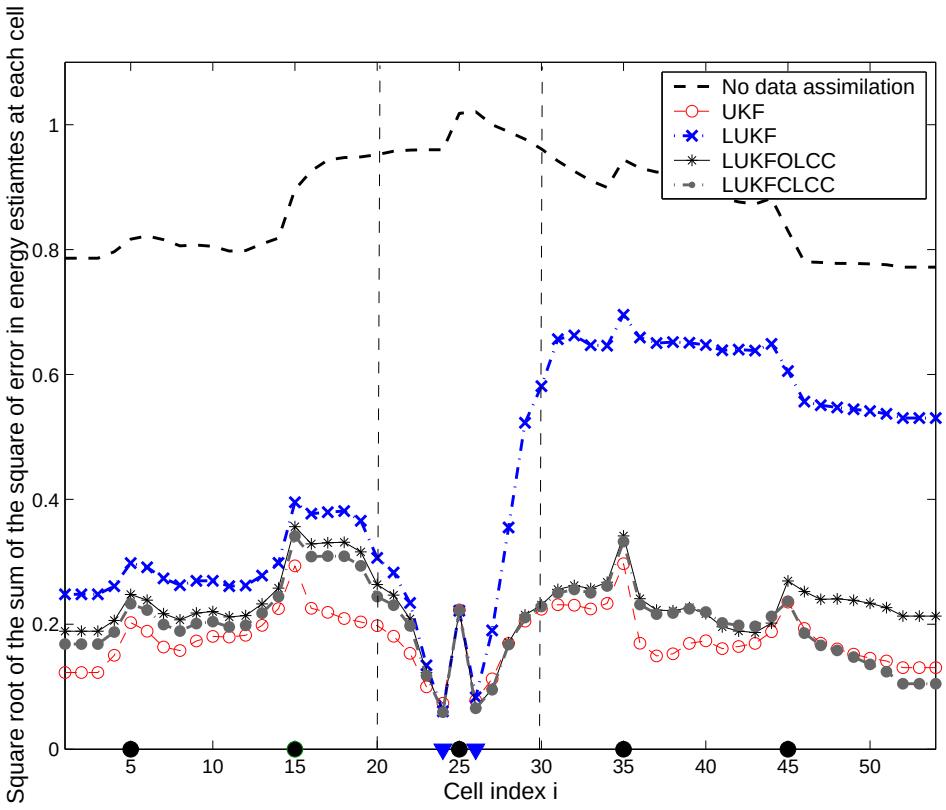


Figure 7.3: Square root of the sum of the square of the error in energy estimates from LUKF, LUKFCOLC, and LUKFCCLC. All three estimators use a time varying estimator gain to inject data into the cells with index between 20 and 30. The error in energy estimates from UKF is performed is also plotted for comparison. The performance of LUKFCCLC is close to that of UKF.

7.5 Two Dimensional Magnetohydrodynamics Using BATSRUS

BATSRUS (Block Adaptive-Tree Solar-wind Roe-type Upwind Scheme) [84] is a finite volume scheme used to model the interactions between the magnetic field of various planets with the solar wind. The dynamics of the flow variables is governed by Euler's equations and Maxwell's electromagnetic equation. BATSRUS divides the three-dimensional spatial domain into cubes of various sizes and a finite volume discretization technique similar to the one mentioned in the previous section is used to model the dynamics of the flow variables density, momentum, pressure, and magnetic field. BATSRUS has the ability to change the resolution of the grids adaptively so that enhance resolution can be obtained in regions of interest. However, we do not use this feature in our simulations. Instead, we use BATSRUS to test the data assimilation techniques on a simple 2-D magnetohydrodynamic bowshock model.

Consider a 2D spatial grid comprising of 4800 square cells with index (i, j) for $i = 1, \ldots, n_x = 40$ and $j = 1, \ldots, n_y = 120$, that covers a rectangular region spanning the coordinates $-10 \leqslant x_{\mathrm{c}} \leqslant 10$ and $-30 \leqslant y_{\mathrm{c}} \leqslant 30$. We use BATSRUS to model the dynamics of the flow variables density (ρ), momentum (m_x, m_y), pressure (p) and magnetic field (B_x, B_y) in each cell. The flow variables at the edges are determined by the boundary conditions and the flow variables at the interior cells are updated using the second-order Rusanov scheme. We choose initial flow conditions so that the flow is supersonic. We assume floating boundary conditions for all cells along the edges, except for two cells at locations indicated by '▶' in Figure 7.4 that are assigned reflective boundary conditions so that a bow-shock is created.

Let $U^{[i,j]} \in \mathbb{R}^6$ denote the flow variable at the center of (i, j) cell. Next, define

the state vector $x \in \mathbb{R}^{6(n_x-4)(n_y-4)}$ by

$$x \triangleq \left[\begin{array}{ccc} U_k^{[3,3]} & \cdots & U_k^{[n_x-2,n_y-2]} \end{array} \right] \tag{7.5.1}$$

so that the system dynamics are given by (7.2.1). We assume that w_k in (7.2.1) is zero-mean white Gaussian process noise with covariance Q so that only the cells with coordinates indicated by '•' in Figure 7.4 are directly affected by w_k. We simulate the truth model for 1 minute with a sampling time of $t_\mathrm{s} = 0.01$ s. We assume that noisy measurements y_k of the flow variables ρ, m_x, m_y, B_x, B_y and $\mathcal{E}$ at cells within the bow-shock region with coordinates indicated by '■' in Figure 7.4 are available so that y_k is given by (7.2.2), where $h(x_k, k) = Cx_k$ and C depends on the coordinates of the cells where measurements are available.

The density and magnetic filed lines at $t = 1$ minute are shown in Figure 7.4. The bow-shock is the semi-circular region where the density is higher than the density of inflow at the boundary cells. Note that the magnetic filed lines tend to curve around the bow-shock region. Next, we perform data assimilation using LUKF, LUKFCOLC, and LUKFCCLC. Figure 7.5 shows a plot of the difference in square root of the sum of the squares of error in energy estimates between the no data assimilation case and LUKF, LUKFCOLC, and LUKFCCLC. Hence, positive values indicate a significant improvement in the estimates. Note that the state dimension $n = 25056$ and since UKF requires $2n + 1 = 50113$ ensembles, we do not use UKF to obtain the state estimates. Also, we use Monte Carlo methods to determine the steady-state correlations used in LUKFCOLC and LUKFCCLC. The local region used in LUKF, LUKFCOLC and LUKFCCLC is shown in Figure 7.4 by the solid lines and x_L contains the state variables in this region.

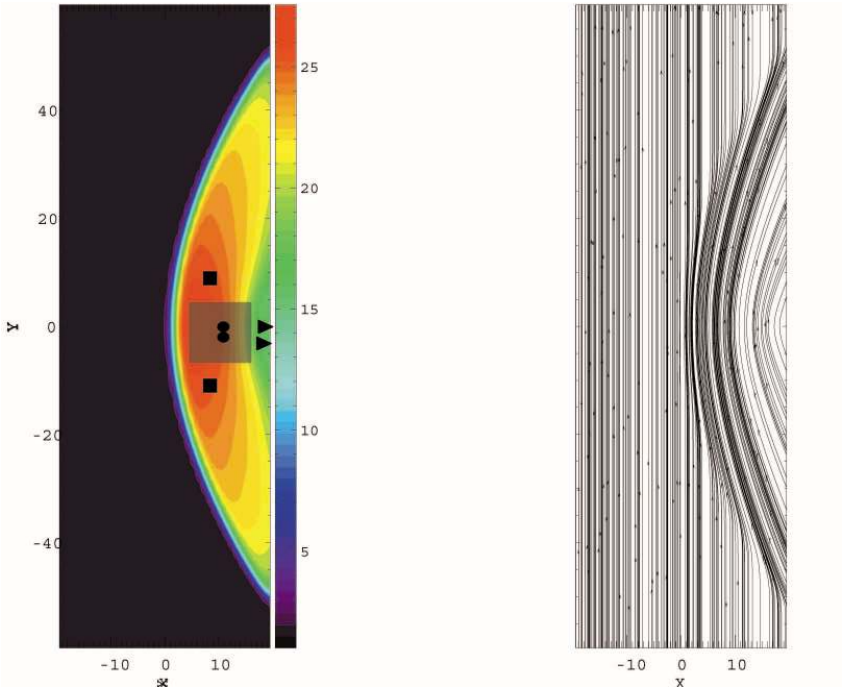


Figure 7.4: A bowshock is formed when supersonic flow from the left edge interacts with a stationary object ('▶'). The cells where disturbance enters the system is indiacted by '■' and the cells where measurements are available are indicated by '●'. The local region corresponding to the state x_{L} is indicated by the shaded rectangular region around the measurement cells.

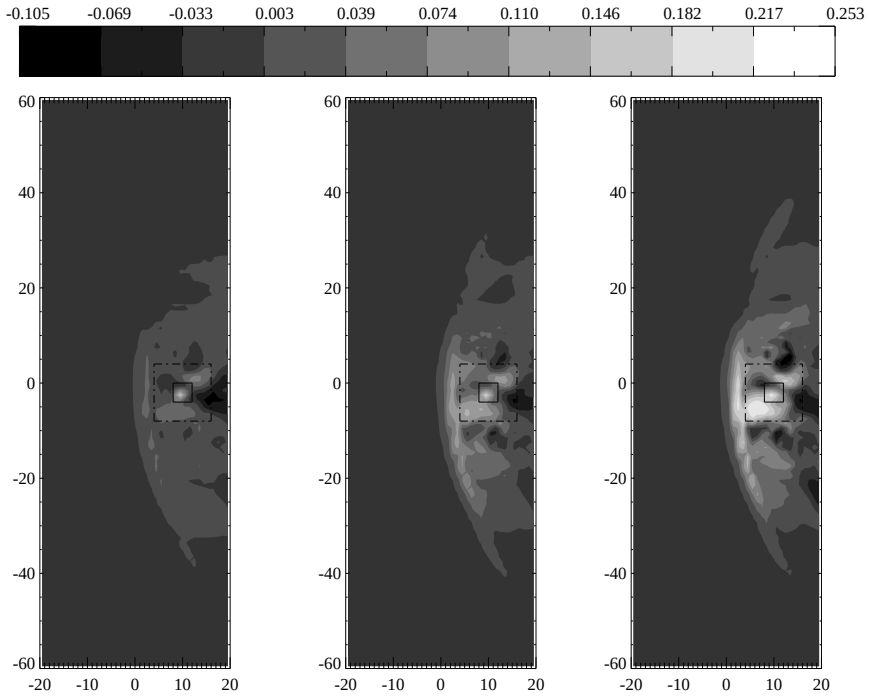


Figure 7.5: The difference in the error in the square root of the sum of the square of error in pressure estimates between the no data assimilation case and LUKF (left), LUKFCOLC (middle), and LUKFCCLC (right). The horizontal and vertical axis denote the x and y spatial coordinates. Positive values indicate regions where the estimators improve the estimates of the state compared to the no data assimilation case.

7.6 Conclusion

We presented extensions of the the unscented Kalman filter that propagates 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 VIII

Conclusions and Future Work

This dissertation presented reduced-complexity algorithms for data assimilation of large-scale linear and nonlinear discrete-time systems. Chapters II-IV dealt with linear systems and presented new estimation algorithms that are variations of the Kalman filter. Chapters V-VII presented variations of the unscented Kalman filter for data assimilation of nonlinear systems and dealt with reducing the ensemble size of the unscented Kalman filter.

The main contribution presented in Chapter II is an estimator that injects data into only a specific subset of the state. Unlike the Kalman filter, the estimator presented in Chapter II depends on the weighting on the error in the state estimates. Thus, a possible extension is to develop methods that determine the exact subspace of the state estimate that has to be injected with data in order to get a better estimate of a specific subset of the state. Another possible extension would be to obtain rigorous conditions that guarantee the stability of the spatially constrained estimator when used for linear systems.

In Chapter III, we obtained a reduced-order estimator using a finite-horizon cost-minimization technique. Although this estimator used a reduced-order dynamics to propagate the estimator state, the full-order covariance had to be propagated. Future

research may include developing square-root versions of the reduced-order estimator so that the rank reduction techniques used in Chapter III can be used to reduce the computational cost of propagating the full order error covariance.

Chapter IV introduced a reduced-rank square-root estimator that propagates a low-rank approximation of the error covariance by performing a Cholesky decomposition of the error covariance at every time step. Although this estimator provides better estimates than the analogous filter based on the singular value decomposition in many examples, future work could determine rigorous conditions that guarantees better estimates from the Cholesky-based estimator. The performance of the Cholesky-based estimator improves when a certain basis for the state is used during estimation. Hence, yet another extension would be to determine the basis transformation that yields the best performance for time-invariant systems.

Chapter V marks the transition from estimation of linear systems to estimation of nonlinear systems. Comparisons of the extended Kalman filter and unscented Kalman filter indicate that the unscented Kalman filter provides significantly better estimates compared to the extended Kalman filter when the nonlinearities in the system dynamics become severe. Moreover, since the Jacobian of the dynamics is not necessary, the unscented Kalman filter serves as a convenient algorithm for state estimation of complex large-scale systems like hydrodynamic and magnetohydrodynamic flow that are modeled using finite volume schemes. Future work could involve determining methods to ensure that the ensemble members that are reinitialized at every time step satisfy physical constraints, for example, the value of density in all of the ensemble members should be positive at every time step.

Chapter VI combines the unscented Kalman filter introduced in Chapter V and the reduced-rank square-root estimator introduced in Chapter IV. The resulting

variation of the unscented Kalman filter uses a reduced ensemble that is constructed using the columns of the Cholesky factor of the pseudo-error covariance. In the examples in Chapter VI, we use a basis transformation that is inspired by the observability matrix of banded linear systems. Future work could consider extensions to the case when the measurements are nonlinear functions of the state. Another possible extension could be to determine the basis transformation of the state vector that yields the best performance.

Finally, Chapter VII dealt with an estimator that uses a static estimator gain based on steady-state correlations to compensate for the neglected correlations in localized data assimilation schemes. Thus, data injection could be performed on a larger subset of the state estimate without additional online computational effort. Future extensions could consider comparisons between this estimation algorithm and the estimator in Chapter VI.

APPENDICES

APPENDIX A

Correlation Bounds for Discrete-time Systems with Banded Dynamics

We consider the steady-state error covariance for a discrete-time system with banded dynamics. Such systems frequently arise from the spatial and temporal discretization of partial differential equations. In such systems, the magnitudes of the entries of the steady-state covariance matrix typically decrease as the distance from the diagonal increases. We obtain a bound on the entries of the covariance matrix beyond a given distance from the diagonal. The results here have been published in [86].

A.1 Banded Matrices

Let $A \in \mathbb{R}^{n\times n}$ and assume that the nonzero entries of A are restricted to a banded region around the main diagonal. We define the *semi-width* $\omega(A)$ of A to be

$$\omega(A) \triangleq \min\{l : A_{i,j} = 0 \ \text{ for all } i, j \text{ such that } |i-j| > l\}. \tag{A.1}$$

For example, if A is diagonal, then $\omega(A) = 0$; if A is tridiagonal, then $\omega(A) = 1$; and if A is pentadiagonal, then $\omega(A) = 2$. Clearly, $\omega(A) \leqslant n-1$. It is easy to see that $\omega(AB) \leqslant \omega(A) + \omega(B)$. More generally, we have the following observation.

Proposition A.1.1 *Let $A_1, \ldots, A_p \in \mathbb{R}^{n\times n}$. Then,*

$$\omega\left(A_1 \cdots A_p\right) \leqslant \min\left\{n-1, \sum_{i=1}^{p} \omega(A_i)\right\}. \tag{A.2}$$

A.2 Correlation Bounds

Consider the linear time-invariant discrete-time system

$$x_{k+1} = Ax_k + w_k, \tag{A.1}$$

where $x_k, w_k \in \mathbb{R}^n$ and w_k is zero-mean white noise with covariance Q. Furthermore, we assume that A is asymptotically stable, that is,

$$\operatorname{sprad}(A) < 1, \tag{A.2}$$

where for all $A \in \mathbb{R}^{n\times n}$, the spectral radius of A is defined by

$$\operatorname{sprad}(A) \triangleq \max\{|\lambda| : \lambda \in \operatorname{spec}(A)\}. \tag{A.3}$$

The positive-semidefinite state covariance $P_k \triangleq \mathcal{E}[x_k x_k^{\mathrm{T}}]$, where $\mathcal{E}[\cdot]$ denotes the expected value, is updated using

$$P_{k+1} = AP_kA^{\mathrm{T}} + Q. \tag{A.4}$$

Since A is asymptotically stable and Q is positive semidefinite, $P \triangleq \lim_{k\to\infty} P_k$ exists and satisfies the discrete-time Lyapunov equation

$$P = APA^{\mathrm{T}} + Q. \tag{A.5}$$

Furthermore,

$$P = \sum_{i=0}^{\infty} A^i Q A^{i\mathrm{T}}. \tag{A.6}$$

Let $\varepsilon > 0$ satisfy

$$\operatorname{sprad}(A) < \varepsilon < 1, \tag{A.7}$$

so that

$$\operatorname{sprad}(\frac{1}{\varepsilon}A) = \frac{1}{\varepsilon}\operatorname{sprad}(A) < 1. \tag{A.8}$$

It thus follows from (A.6) that

$$P = \sum_{i=0}^{\infty} \varepsilon^{2i} Q_i, \tag{A.9}$$

where $Q_0 = Q$ and, for all $i = 1, 2, \ldots$, Q_i is defined by

$$Q_i \triangleq \left(\frac{A}{\varepsilon}\right)^i Q \left(\frac{A^{\mathrm{T}}}{\varepsilon}\right)^i. \tag{A.10}$$

Since $\omega(\varepsilon A) = \omega(A) = \omega(A^{\mathrm{T}})$, it follows from (A.2) that, for all $i = 0, 1, \ldots$,

$$\omega(Q_i) \leqslant \min\bigl\{n-1, 2i\omega(A) + \omega(Q)\bigr\} \tag{A.11}$$

Next, for $i = 0, \ldots, n-1$, define $H_i \in \mathbb{R}^{n\times n}$ by

$$H_i \triangleq \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & & \ddots & \ddots & \vdots \\ 1 & & \ddots & & \ddots & 0 \\ 0 & \ddots & & \ddots & & 1 \\ \vdots & \ddots & \ddots & & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & \cdots & 1 \end{bmatrix}, \tag{A.12}$$

where the semi-width of the band of ones is chosen such that

$$\omega(H_i) = i. \tag{A.13}$$

Now, for $i = 0, \ldots, n-1$, define P_i by

$$P_i \triangleq H_i \circ P, \tag{A.14}$$

where $\circ$ denotes the Schur product. Then the (k,l) entry of P_i is given by

$$(P_i)_{k,l} = \begin{cases} P_{k,l}, & \text{if } |k-l| \leqslant i, \\ 0, & \text{else.} \end{cases} \tag{A.15}$$

For all $j = 0, 1, \ldots$ and $i = 0, \ldots, n-1$, if $\omega(Q_j) \leqslant \omega(H_i)$, then $(1_n - H_i) \circ Q_j = 0$, where 1_n is the $n \times n$ matrix whose entries are all equal to 1. Therefore, for $i = 0, \ldots, n-1$, taking the Schur product of (A.9) with $1_n - H_i$ and using $(1_n - H_i) \circ P = P - P_i$ yields

$$P - P_i = \sum_{j=L(i)}^{\infty} \varepsilon^{2j} (1_n - H_i) \circ Q_j, \tag{A.16}$$

where $L : \mathbb{N} \to \mathbb{N}$ is defined by

$$L(i) \triangleq \max\left\{0, \text{floor}\left(\frac{i - \omega(Q)}{2\omega(A)}\right) + 1\right\}. \tag{A.17}$$

Proposition A.2.1 *Assume that $A \in \mathbb{R}^{n\times n}$ satisfies* (A.2) *and let $\varepsilon > 0$ satisfy* $\text{sprad}(A) < \varepsilon < 1$. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^{n\times n}$. Then,*

$$\sigma_A \triangleq \max_{i\in\mathbb{N}} \frac{1}{\varepsilon^i} \|A^i\| \tag{A.18}$$

exists.

Proof. It follows from (A.8) that $\lim_{i\to\infty} \frac{1}{\varepsilon^i} A^i = 0$. Hence, σ_A exists. □

Proposition A.2.2 *Assume that $A \in \mathbb{R}^{n\times n}$ satisfies* (A.2) *and let $\varepsilon > 0$ satisfy* $\text{sprad}(A) < \varepsilon < 1$. *Let $\|\cdot\|$ be a monotonic submultiplicative norm on $\mathbb{R}^{n\times n}$. Then, for $i = 0, \ldots, n-1$,*

$$\|P - P_i\| \leqslant \frac{\varepsilon^{2L(i)}}{1 - \varepsilon^2} \sigma_A^2 \|Q\|. \tag{A.19}$$

Proof. Since $\|\cdot\|$ is monotonic, it follows that, for all $i = 0, \ldots, n-1$ and $j = 0, 1, \ldots,$

$$\|(1_n - H_i) \circ Q_j\| \leqslant \|Q_j\|. \tag{A.20}$$

Furthermore, since $\|\cdot\|$ is submultiplicative, it follows that, for all $j = 0, 1, \ldots,$

$$\|Q_j\| \leqslant \|Q\|\|\frac{1}{\varepsilon^j} A^j\|^2. \tag{A.21}$$

Hence, it follows from Proposition 3.1 that, for all $j = 0, 1, \ldots,$

$$\|Q_j\| \leqslant \|Q\|\sigma_A^2. \tag{A.22}$$

Taking the norm of $P - P_i$ in (A.16) and using (A.20) yields

$$\|P - P_i\| \leqslant \varepsilon^{2L(i)}\|Q_{L(i)}\| + \varepsilon^{2L(i)+2}\|Q_{L(i)+1}\| + \cdots. \tag{A.23}$$

It then follows from (A.22) that

$$\|P - P_i\| \leqslant \sigma_A^2\|Q\|(\varepsilon^{2L(i)} + \varepsilon^{2L(i)+2} + \cdots). \tag{A.24}$$

Since $0 < \varepsilon < 1$,

$$\sum_{j=L(i)}^{\infty} \varepsilon^{2j} = \frac{\varepsilon^{2L(i)}}{1 - \varepsilon^2}. \tag{A.25}$$

Therefore, (A.24) and (A.25) imply (A.19). □

A.3 Compartmental Model Example

We consider a system comprised of n compartments or subsystems that exchange energy through mutual interaction [49]. Applying conservation of energy yields, for $i = 1, \ldots, n$,

$$x_i(k+1) = x_i(k) - \beta x_i(k) - \alpha\left(x_{i+1}(k) - x_i(k)\right) - \alpha\left(x_i(k) - x_{i-1}(k)\right), \tag{A.1}$$

where $0 < \beta < 1$ is the loss coefficient and $0 < \alpha < 1$ is the flow coefficient. It follows from (A.1) that

$$x(k+1) = Ax(k), \tag{A.2}$$

where

$$x \triangleq \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^{\mathrm{T}} \tag{A.3}$$

and $A \in \mathbb{R}^{n \times n}$ is defined by

$$A \triangleq \begin{bmatrix} 1-\beta-\alpha & \alpha & 0 & 0 & \cdots & 0 \\ \alpha & 1-\beta-2\alpha & \alpha & 0 & \cdots & 0 \\ 0 & \alpha & 1-\beta-2\alpha & \alpha & \cdots & 0 \\ \vdots & & \ddots & & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \alpha & 1-\beta-\alpha \end{bmatrix}. \tag{A.4}$$

Since A is tridiagonal, $\omega(A) = 1$. We choose $n = 20$ and evaluate P using (A.5) with $Q = I_n$ for $(\alpha, \beta) = (0.1, 0.8)$. The spectral radius of A, and the chosen value of ε are shown in Table 1. We choose $\|\cdot\|$ to be the Frobenius norm $\|\cdot\|_{\mathrm{F}}$.

α	β	sprad(A)	ε
0.1	0.8	0.2	0.4, 0.3, 0.21

Table A.1: Parameters used in the compartmental model example.

Note that for $(\alpha, \beta) = (0.1, 0.8)$, sprad$(A) < 1$ and hence, σ_A defined in (A.18) exists and is determined numerically. Next, for $i = 0, \ldots, 9$, we plot $\frac{\varepsilon^{2L(i)}}{1-\varepsilon^2}\sigma_A^2\|Q\|_{\mathrm{F}}$ and $\|P - P_i\|_{\mathrm{F}}$ with $(\alpha, \beta) = (0.1, 0.8)$ in Figure 1. Note that $\|Q\|_{\mathrm{F}} = \sqrt{20}$. The magnitudes of the entries of the steady-state covariance P for $(\alpha, \beta) = (0.1, 0.8)$ are plotted in Figure 2. It can be seen that the magnitude of the entries of the covariance decrease as the distance from the diagonal increases.

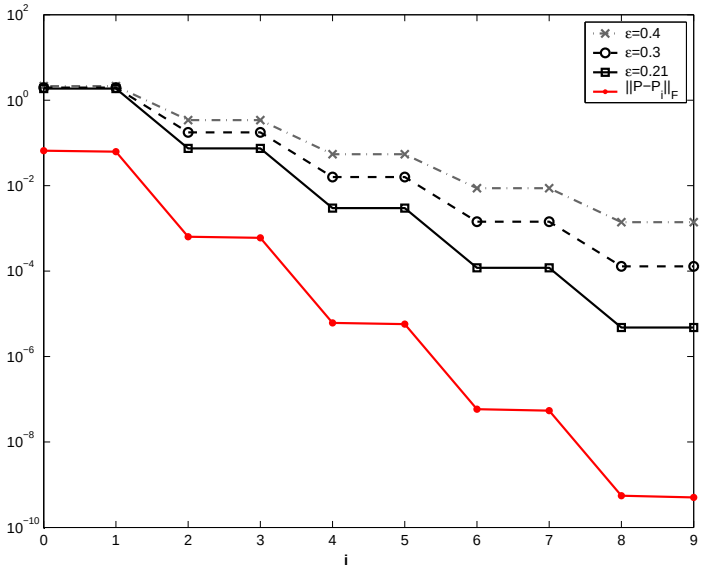


Figure A.1: $\|P - P_i\|_{\mathrm{F}}$ and bound (A.19) for $\alpha = 0.1$ and $\beta = 0.8$ and various values of ε.

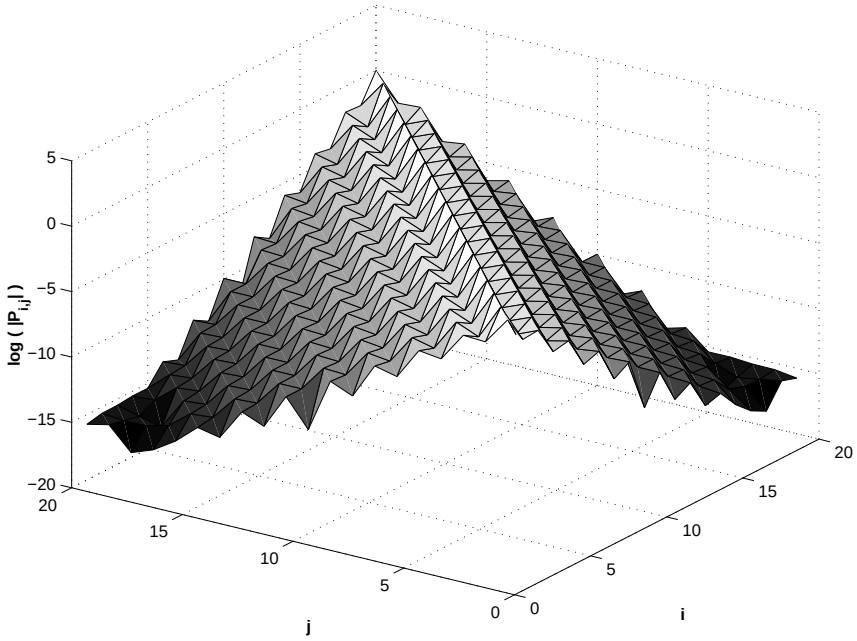


Figure A.2: Surface plot of $\log(|P_{i,j}|)$ for $\alpha = 0.1$ and $\beta = 0.8$.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] G. Fisher, "Applications of Least Squares in Econometrics," *The Canadian Journal of Economics*, vol. 29, , pp. S548-S550, 1996.

[2] B. H. Bransden and C. J. Noble, "Application of Least-Squares Methods to Atomic Rearrangement Collisions," *J. Phys,* vol. 32, pp. 1305-1314, 1999.

[3] M. Cirrincione, M. Pucci, G. Cirrincione, and G. A. Capolino, "A New Experimental Application of Least-Squares Techniques for the Estimation of the Induction Motor Parameters," *Industry Applications Conference,* vol. 2, pp. 1171 - 1180, 2002.

[4] S. Haykin, *Adaptive filter theory,* Prentice-Hall, 1996

[5] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Trans, ASME–J. of Basic Eng.*, vol. 82, pp. 35–45, 1960.

[6] B. D. O. Anderson and J. B. Moore, *Optimal Filtering,* Dover Publications Inc., Mineola, NY, 1979.

[7] D. S. Bernstein and D. C. Hyland, "The Optimal Projection Equations for Reduced-Order State Estimation", *IEEE Trans. Autom. Contr.*, Vol. AC-30, pp. 583-585, 1985.

[8] W. M. Haddad and D. S. Bernstein, "Optimal Reduced-Order Observer-Estimators", *AIAA J. Guid. Dyn. Contr.,* Vol. 13, pp. 1126-1135, 1990.

[9] 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. Amer. Contr. Conf.*, New York, June 2007.

[10] F. E. Daum, "Exact Finite-Dimensional Nonlinear Filters," *IEEE Trans. Auto. Contr.,* vol. AC-31, pp. 616-622, 1986.

[11] M. Athans, R. P. Wishner and A. Bertolini, "Suboptimal State Estimation for Continous-Time Nonlinear Systems from Discrete Noisy Measurements," *IEEE Trans. Auto. Contr.,* vol. AC-13, pp. 504-514, 1968.

[12] V. E. Benes, "Exact Finite-Dimensional Filters For Certain Diffusions with Nonlinear Drift," *Stochastics*, vol. 5, pp. 65-92, 1981.

[13] A. J. Krener and A. Duarte, "A Hybrid Computational Approach to Nonlinear Estimation," *Proc. Conf. Dec. Contr.,* Kobe, Japan, pp. 1815-1819, December 2004.

[14] A. Gelb, *Applied Optimal Estimation*, The M.I.T Press, 1974.

[15] C. P. Mracek, J. R. Cloutier, and C. A. D'Souza, "A New Technique for Nonlinear Estimation," in *Proc. Int. Conf. Contr. App.*, Dearborn, MI, June 1996, pp. 338–343.

[16] P. M. Djuric et. al., "Particle Filtering," *IEEE Signal Processing Magazine,* vol. 20, pp. 19 - 38, 2003.

[17] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter*, Springer, 2006.

[18] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter*, Artech House, 2004.

[19] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Trans. Autom. Contr.*, vol. 45, pp. 477–482, 2000.

[20] I. Szunyogh et. al., "Assessing a Local Ensemble Kalman Filter: Perfect Model Experiments with the National Centers for Environmental Prediction Global Model," *Tellus*, vol. 57A, pp. 528-545, 2005.

[21] M. Verlaan and A. W. Heemink, "Tidal Flow Forecasting Using Reduced-Rank Square-Root Filters," *Stochastics Hydrology and Hydraulics*, vol. 11, pp. 349–368, 1997.

[22] M. K. Tippett, J. L. Anderson, C. H. Bishop, T. M. Hamill, and J. S. Whitaker, "Ensemble Square Root Filters," *Mon. Wea. Rev.*, vol. 131, pp. 1485–1490, 2003.

[23] J. Chandrasekar, D. S. Bernstein, O. Barrero, and B. De Moor, "Kalman Filtering with Constrained Output Injection," *Int. J. Contr.*, 2007.

[24] J. Chandrasekar and D. S. Bernstein, "Spatially Constrained Output Injection for State Estimation with Banded Closed-Loop Dynamics," *Proc. Amer. Contr. Conf.*, Minneapolis, MN, June 2006, pp. 4454–4459.

[25] P. Hippe and C. Wurmthaler, "Optimal Reduced-Order Estimators in the Frequency Domain: The Discrete-Time Case", *Int. J. Contr.*, Vol. 52, pp. 1051-1064, 1990.

[26] C.-S. Hsieh, "The Unified Structure of Unbiased Minimum-Variance Reduced-Order Filters", *Proc. Contr. Dec. Conf.*, pp. 4871-4876, Maui, HI, December 2003.

[27] B. F. Farrell and P. J. Ioannou, "State Estimation Using a Reduced-Order Kalman Filter", *J. Atmos. Sci.*, Vol. 58, pp. 3666-3680, 2001.

[28] A. W. Heemink, M. Verlaan, and A. J. Segers, "Variance Reduced Ensemble Kalman Filtering", *Mon. Wea. Rev.*, Vol. 129, pp. 1718-1728, 2001.

[29] 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*, Vol. 14, pp. 1720-1737, 2001.

[30] P. Fieguth, D. Menemenlis, and I. Fukumori, "Mapping and Pseudo-Inverse Algorithms for Data Assimilation", *Proc. Int. Geoscience Remote Sensing Symp.*, pp. 3221-3223, 2002.

[31] O. Barrero, D. S. Bernstein, B. L. R. De Moor, "Spatially Localized Kalman Filtering for Data Assimilation", *Proc. Amer. Contr. Conf.*, Portland, pp. 3468-3473, 2005.

[32] D. I. Lawrie, P. J. Fleming, G. W. Irwin, and S. R. Jones, "Kalman Filtering: A Survey of Parallel Processing Alternatives", *Proc. IFAC Workshop on Algorithms and Architectures for Real-Time Control*, pp. 49-56, 1992, Pergamon.

[33] M. Morf and T. Kailath, "Square-Root Algorithms for Least-Squares Estimation", *IEEE Trans. Autom. Contr.*, Vol. AC-20, pp. 487–497, 1975.

[34] F. L. Lewis, *Optimal Estimation*, John Wiley and Sons, 1986.

[35] 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*, Vol. 39-RS1S04, 2004.

[36] D. S. Bernstein, *Matrix Mathematics*, Princeton University Press, 2005.

[37] J. Chandrasekar, I. S. Kim, and D. S. Bernstein, "Reduced-Order Kalman Filtering for Time-Varying Systems," *Conf. Dec. Contr.*, 2007, submitted.

[38] M. Lewis, S. Lakshmivarahan, and S. Dhall, *Dynamic Data Assimilation: A Least Squares Approach*, Cambridge, 2006.

[39] D. S. Bernstein, L. D. Davis, and D. C. Hyland, "The Optimal Projection Equations for Reduced-Order, Discrete-Time Modelling, Estimation and Control," *AIAA J. Guid. Contr. Dyn.*, vol. 9, pp. 288–293, 1986.

[40] W. M. Haddad, D. S. Bernstein, and V. Kapila, "Reduced-Order Multirate Estimation," *AIAA J. Guid. Contr. Dyn.*, vol. 17, pp. 712–721, 1994.

[41] A. Johnson, "Discrete and Sampled-Data Stachastic Control Problems with Complete and Incomplete State Information," *Applied Mathematics and Optimization*, vol. 24, pp.289-316, 1991.

[42] J. Chandrasekar, I. S. Kim, and D. S. Bernstein, "Cholesky-Based Reduced-Rank Kalman Filtering," 2007, submitted.

[43] S. Gillijns, D. S. Bernstein, and B. D. Moor, "The Reduced Rank Transform Square Root Filter for Data Assimilation," *Proc. 14th IFAC Symposium on System Identification (SYSID2006), Newcastle, Australia*, vol. 11, pp. 349–368, 2006.

[44] D. Treebushny and H. Madsen, "A New Reduced Rank Square Root Kalman Filter for Data Assimilation in Mathematical Models," *Lecture Notes in Computer Science*, vol. 2657, pp. 482–491, 2003.

[45] J. L. Anderson, "An Ensemble Adjustment Kalman Filter for Data Assimilation," *Mon. Wea. Rev.*, vol. 129, pp. 2884–2903, 2001.

[46] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, reprinted by Dover, 2006.

[47] M. Morf and T. Kailath, "Square-Root Algorithms for Least-Squares Estimation," *IEEE Trans. Autom. Contr.*, vol. AC-20, pp. 487–497, 1975.

[48] G. W. Stewart, "Matrix Algorithms Volume 1: Basic Decompositions," *SIAM*, 1998.

[49] D. S. Bernstein and D. C. Hyland, "Compartmental Modeling and Second-Moment Analysis of State Space Systems," *SIAM J. Matrix Anal. Appl.*, vol. 14, no. 3, pp. 880–901, 1993.

[50] J. Chandrasekar, A. J. Ridley, and D. S. Bernstein, "A Comparison of the Extended and Unscented Kalman Filters for Discrete-Time Systems with Non-differentiable Dynamics," *Proc. Amer. Contr. Conf.*, New York, NY, 2007.

[51] M. Athans, R. P. Wishner, and A. Bertolini, "Suboptimal State Estimation for Continuous-Time Nonlinear Systems from Discrete Noisy Measurements," *IEEE Trans. Autom. Contr.*, vol. 13, pp. 504–514, 1968.

[52] K. Ito and K. Xiong, "Gaussian Filters for Nonlinear Filtering Problems," *IEEE Trans. Autom. Contr.*, vol. 45, pp. 910–927, 2000.

[53] A. J. Krener and W. Kang, "Locally Convergent Nonlinear Observers," *SIAM J. Contr. Optim.*, vol. 42, pp. 155–177, 2003.

[54] M. Verlaan and A. W. Heemink, "Nonlinearity in Data Assimilation Applications: A Practical Method for Analysis," *Mon. Wea. Rev.*, vol. 129, pp. 1578–1589, 2001.

[55] R. Todling and S. E. Cohn, "Suboptimal Schemes for Atmospheric Data Assimilation Based on the Kalman Filter," *Mon. Wea. Rev.*, vol. 122, pp. 2530–2557, 1994.

[56] A. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, 1970.

[57] J. Chandrasekar, A. J. Ridley, and D. S. Bernstein, "A SDRE-Based Asymptotic Observer for Nonlinear Discrete-Time Systems," in *Proc. Amer. Contr. Conf.*, Portland, OR, June 2005, pp. 3630 – 3635.

[58] W. Sun, K. M. Nagpal, and P. P. Khargonekar, "H_∞ Control and Filtering for Sampled-Data Systems," *IEEE Trans. Autom. Contr.*, vol. 38, pp. 1162–1175, 1993.

[59] E. G. Collins Jr. and T. Song, "Robust H_∞ Estimation and Fault Detection of Uncertain Dynamic Systems," *Int. J. Guid. Cont. Dyn.,* vol. 23, no. 5, pp. 857–864, 2000.

[60] R. V. der Merwe and E. A. Wan, "The Square-root Unscented Kalman Filter for State and Parameter-Estimation," in *Proc. Int. Conf. Acou. Speech Sig. Process.*, May 2001, pp. 3461 – 3464.

[61] G. Evensen, "The Ensemble Kalman Filter: Theoretical Formulation and Practical Implementaion," *Ocean Dynamics*, vol. 53, pp. 343–367, 2003.

[62] J. S. Whitaker and T. M. Hamill, "Ensemble Data Assimilation without Perturbed Observations," *Mon. Wea. Rev.*, vol. 130, pp. 1913–1924, 2002.

[63] S. Gillijns, O. B. 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.*, Minneapolis, MN, June 2006.

[64] C. Groth, D. D. Zeeuw, T. Gombosi, and K. Powell, "Global 3D MHD Simulation of a Space Weather Event: CME Formation, Interplanetary Propagation, and Interaction with the Magnetosphere," *J. Geophys. Res.*, vol. 105, pp. 25 053–25 078, 2000.

[65] K. Powell, P. Roe, T. Linde, T. Gombosi, and D. D. Zeeuw, "A Solution-Adaptive Upwind Scheme for Ideal Magnetohydrodynamics," *J. Comp. Phys.*, vol. 154, p. 284, 1999.

[66] R. J. Leveque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, 2002.

[67] C. Hirsch, *Numerical Computation of Internal and External Flows*, John Wiley and Sons, 1990.

[68] C. E. Aull, "The First Symmetric Derivative," *Amer. Math. Mon.*, vol. 74, pp. 708–711, 1967.

[69] L. Larson, "The Symmetric Derivative," *Trans. Amer. Math. Soc.*, vol. 277, pp. 589–599, 1983.

[70] S. Julier, "The Scaled Unscented Transformation," in *Proc. Amer. Contr. Conf.*, Anchorage, May 2002, pp. 4555 – 4559.

[71] E. A. Wan and R. van der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," in *Proc. of IEEE Symp. (AS-SPCC)*, Alberta, Canada.

[72] J. J. LaViola, "A Comparison of Unscented and Extended Kalman Filtering for Estimating Quaternion Motion," in *Proc. Amer. Contr. Conf.*, June 2003, pp. 2435 – 2440.

[73] J. Chandrasekar, I. S. Kim, A. J. Ridley, and D. S. Bernstein, "Reduced-Order Covariance-Based Unscented Kalman Filtering with Complementary Steady-State Correlation," submitted, 2007.

[74] X. Wang and C. H. Bishop, "A Comparison of Breeding and Ensemble Transform Kalman Filter Ensemble Forecast Schemes," *J. Atmos. Sci.*, vol. 60, pp. 1140-1158, 2003.

[75] C. H. Bishop, B. J. Etherton, and S. J. Majumdar, "Adative Sampling with the Ensemble Transform Kalman Filter. Part I: Theoretical Aspects," *Mon. Wea. Rev.,* vol. 129, pp. 420–436, 2001.

[76] H. L. Mitchell and P. L. Houtekamer, "An Adaptive Ensemble Kalman Filter," *Mon. Wea. Rev.* , vol. 128, pp. 416–433, 2000.

[77] X. Wang, C. H. Bishop, and S. J. Julier, "Which is Better, an Ensemble of Positive-Negative Pairs or a Centered Spherical Simplex Ensemble ?," *Mon. Wea. Rev.*, vol. 132, pp. 1590–1605, 2004.

[78] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter*, Springer, 2006.

[79] E. N. Lorenz, "Predictability - A Problem Partly Solved," *Predictability of Weather and Climate*, Cambridge University Press, 2006.

[80] J. P. Hespanha et. al, "Nonlinear Norm-Observability Notions and Stability of Switched Systems," *IEEE Trans. Autom. Contr.*, vol. 50, no. 2, pp. 154 - 168, 2005.

[81] J. Chandrasekar, I. S. Kim, A. J. Ridley, and D. S. Bernstein, "Reduced-Order Covariance-Based Unscented Kalman Filtering with Complementary Steady-State Correlation," *Proc. Amer. Contr. Conf.,* New York, NY, 2007.

[82] Y. K. Sasaki and J. S. Goerss, "Satellite Data Assimilation Using NASA Data Systems Test 6 Observations," *Mon. Wea. Rev.*, vol. 110, pp. 16351644, 1982.

[83] 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,* vol. 30, pp. 294309, 1999.

[84] G. Toth, A. Ridley, K. Powell, and T. Gambosi, "A High-Performance Framework for Sun-to-Earth Space Weather Modeling,", *Parallel and Distributed Processing Symposium*, 2005.

[85] I. S. Kim, J. Chandrasekar, A. Ridley, and D. S. Bernstein, "Data Assimilation Using the Global Ionosphere-Thermosphere Model," *Proc. ICCS,* pp. 489-496, Reading, UK, May 2006.

[86] J. Chandrasekar and D. S. Bernstein, "Correlation Bounds for Discrete-Time Systems with Banded Dynamics," *Sys. Contr. Lett.*, vol. 56, pp. 83-86, 2007.