Numerical Studies on the $2R$ Ensemble Kalman Filter Inspired by the Langevinized Ensemble Kalman Filter

Wen-Hung Wang
Project Advisor: Dr. Faming Liang
Department of Statistics, Purdue University

November 29, 2024

Abstract

State-space models have been extensively used in various fields such as signal processing, economics, and weather forecasting due to their versatility in handling time series data. Among the techniques developed for filtering and state estimation, the Ensemble Kalman Filter (EnKF) has proven to be effective in addressing the challenges of non-linearity and high dimensionality, though it often suffers from issues such as covariance underestimation and filter divergence. Inspired by the Langevinized Ensemble Kalman Filter (LEnKF), this study explores the effects of modifying the Kalman gain computation by replacing the observation noise R_t with $2R_t$ in the EnKF. Through numerical studies on low-dimensional and high-dimensional systems—including the Ornstein-Uhlenbeck model, Van der Pol oscillator, and Lorenz-96 model, we demonstrate that the modified EnKF *achieves better uncertainty quantification without compromising state estimation accuracy.* These findings suggest the potential for further improvements in ensemble-based filtering methods, particularly in high-dimensional and non-linear systems.

1 Introduction

State-space models have achieved tremendous success over the past few decades. They have been widely applied in fields such as signal processing, economics, robotics, and weather forecasting, to name a few. Their mathematical formulation and flexibility enable them to address diverse types of problems, especially for time series data.

Following the notations in [6], consider a discrete-time nonlinear system with the latent dynamics process

$$x_{t+1} = f(x_t, u_t) + w_t, \tag{1}$$

and corrupted measurements

$$y_t = h(x_t) + v_t, \ t =, 1, 2, 3, ..., T \tag{2}$$

where y_t is the observed m-dimensional [1] data vector at time t, x_t is the n-dimensional unobserved state vector of interest, u_k is the control vector, representing the controlling input into control-input model, and the initial state x_0 measurement and state process error $v_t \in \mathbb{R}_t^m$ and $w_t \in \mathbb{R}^n$ are mutually and serially independent. w_t and v_t are typically assumed to be zero mean Gaussian with covariance matrices Q_t and R_t, respectively. Throughout this article, we assume that the system contains no unknown parameters, i.e., h, f, Q_t, and R_t are known[2].

The most important inference problem in state-space models is their filtering distribution since its mean provides an online optimal estimate of the states [9]. The filtering distribution π_t is defined as the posterior distribution of the state at time t given all the observations up to time t, i.e., $\pi(x_t \mid y_1, ..., y_t)$. Hence $E[x_t \mid y_1, ..., y_t]$ can be used as a state estimate at t and $Cov(x_t \mid y_1, ..., y_t)$ can be used in uncertainty quantification.

In the case where f and h are linear, one obtains the linear-Gaussian state-space model:

$$x_{t+1} = A_t x_t + B_t u_t + w_t, \;\; w_t \sim N(\mathbf{0}, Q_t), \tag{3}$$

$$y_t = C_t x_t + v_t, \;\; v_t \sim N(\mathbf{0}, R_t). \tag{4}$$

In this simple case, the optimal filter can thus be derived analytically, namely, the Kalman filter (KF) proposed by [7]. However, in many complex systems, f and h can be non-linear, e.g., estimating the states of the Van der Pol oscillator. The KF fails in this setting. Other approaches are necessary; for example, approximation-based approaches such as the extended Kalman filter (XKF) [2] [10]. On top of non-linearity, the KF suffers from a costly computational cost when the state x_t or the observation y_t is high-dimensional due to the storage and computation of the Kalman gain [8].

The ensemble Kalman filter (EnKF) proposed by [4] has been very successful in overcoming these two problems, non-linearity and high-dimensionality. The EnKF was originally proposed to address the computational issue, but it turned out to be useful in non-linear systems [6] [8] [13]. The EnKF is essentially a kind of particle filter algorithm, which evolves a set of particles to approximate the posterior distribution [8] [9] [13]. Additionally, the EnKF updates the particles by shifting but not reweighting or resampling, making it immune to the weight degeneracy issue suffered in most particle filters [8] [13].

Despite its success, the EnKF is by no means perfect. Since it approximates the covariances only by a set of ensemble members, it is suboptimal [6]. This finite-size ensemble approximation of covariances makes the EnKF systematically underestimate the analysis/posterior covariance, leading to incorrect uncertainty quantification, as shown in later numerical studies. In severe cases, the estimated gain goes to zero, severely biases the filter toward forecast values, and ignores the observations [2] [5] [6] [8].

Several techniques can be applied to address this convergence issue. The most common approaches include covariance inflation and localization [2]. Besides these two approaches, a new method called the Langevinized Ensemble Kalman filter (LEnKF) was proposed by [13]. The LEnKF combines the stochastic gradient

[1] It could be m_t but for simplicity we fix the observation as m.

[2] In the case containing unknown parameters, the state augmentation techniques are often used for parameter estimation [1].

Langevin dynamics (SGLD) [12] in the state prior and the analysis and forecast steps from the EnKF. The LEnKF is a scalable algorithm and is immune to weight degeneracy. Most importantly, it enjoys the correct convergence, i.e., it converges to the correct filtering distribution. This allows a correct uncertainty quantification.

The LEnKF inspires this article. The LEnKF provides a theoretically correct convergence to the filtering distribution. It is noticed that in Algorithm 4 of [13], the computation of the Kalman gain takes $2R_t$ [3], which differs from that of the EnKF. Therefore, we are interested in the question: *Would using* $2R_t$ *in the EnKF Kalman gain computation improve the performance?* We will illustrate more in section 3.

The rest of the article is structured as follows: we briefly review the KF, the EnKF, and the LEnKF in section 2. We comment on their ideas and potential issues. Due to limited space, we will focus on the LEnKF's idea and properties instead of stating the algorithms. The main idea of this article is explained in section 3. Section 4 provides numerical studies on three examples. We conclude our results and discuss them in section 5.

2 Review of the Relevant Methodologies

In this section, we briefly review some relevant concepts and refer the reader to related readings for a deeper understanding.

2.1 The Kalman Filter: The Optimal Linear Filter

Suppose the system is linear Gaussian as defined in equations (3) and (4). Let x_t^a denote the analysis values at time t, which is the posterior state estimates of the state x_t upon observing y_t, x_t^f be the forecast values updated via the numerical model, and P_t^a and P_t^f correspond to the analysis state (error) covariance and forecast state (error) covariance, respectively [4]. The optimal filter that minimizes the MSEs

$$\text{MSE} = E\left[(x_t^a - x_t)^T(x_t^a - x_t) \mid y_1, ..., y_t\right] \tag{5}$$

is called the Kalman filter (KF) [7]. This also implies that the best state estimate of the state is $x_t^a = E[x_t \mid y_1, ..., y_t]$, the posterior mean.

The KF iterates between two steps: analysis and forecast steps. In the forecast step, the filter propagates the analysis values from the previous step x_{t-1}^a and P_{t-1}^a to the forecast values of the next step x_t^f and P_t^f via the numerical model (3) without the use of data y_t. Hence $x_t^f = E[x_t \mid y_1, ..., y_{t-1}]$ is the predictive mean and $P_t^f = Cov(x_t \mid y_1, ..., y_{t-1})$ is the predictive covariance. These values are sometimes referred to as prior values. The analysis step then integrates the values from the forecast step and the data y_t to obtain estimates of the posterior mean and variance of the filtering distribution, i.e., x_t^a and P_t^a.

Let a randomly generated initial state estimate be x_0^a and covariance estimate be P_0^a (independent of the errors). Let the estimates from the previous time $t-1$ are x_{t-1}^a and P_{t-1}^a which satisfy $x_{t-1} \mid y_{1:t-1} \sim N(x_{t-1}^a, P_{t-1}^a)$. Then the Kalman updates give the following steps [6].

[3] Assume that all data is used in the update step

[4] Sometimes they are called state error covariances because $\text{tr}(P_t^f) = \text{tr}E[(x_t^f - x_t)^T(x_t^f - x_t) \mid y_1, ..., y_{t-1}] = E[(x_t^f - x_t)^T(x_t^f - x_t) \mid y_1, ..., y_{t-1}]$ is the forecast state MSE; the same reason for P_t^a. Note that we sometimes ignore $\cdot \mid y_1, ...y_{t-1}$ and $\cdot \mid y_1, ...y_t$ for notational simplicity.

- Forecast step:

$$x_t^f = A_t x_{t-1}^a + B_t u_{t-1}, \tag{6}$$
$$P_t^f = A_t P_{t-1}^a A_t^T + Q_t. \tag{7}$$

- Analysis step:

$$K_t = P_{xy_t}^f \left(P_{yy_t}^f\right)^{-1} = P_t^f C_t^T (C_t P_t^f C_t^T + R_t)^{-1}, \tag{8}$$
$$P_t^a = (I - K_t C_t)\, P_t^f, \tag{9}$$
$$x_t^a = x_t^f + \underbrace{K_t}_{\text{Kalman gain}} \underbrace{\left(y_t - C_t x_t^f\right)}_{\text{Innovation}}, \tag{10}$$

where $P_t^f \in \mathbb{R}^{n\times n}$ is called the forecast state error covariance, and

$$P_{xy_t}^f = E\left[e_t^f \left(y_t - y_t^f\right)^T\right] = P_t^f C_t^T, \tag{11}$$
$$P_{yy_t}^f = E\left[\left(y_t - y_t^f\right)\left(y_t - y_t^f\right)^T\right] = C_t P_t^f C_t^T + R_t, \tag{12}$$

where $y_t^f = C_t x_t^f$, $e_t^f = x_t - x_t^f$, $P_{xy_t}^f \in \mathbb{R}^{n\times m}$ and $P_{yy_t}^f \in \mathbb{R}^{m\times m}$.

K_t is called the Kalman gain, which is usually written explicitly as $P_t^f C_t^T (C_t P_t^f C_t^T + R_t)^{-1}$. This is straightforward to show that through these two steps, $N(x_t^a, P_t^a) \overset{d}{=} \pi(x_t \mid y_1, ..., y_t)$. $y_t - C_t x_t^f$ is often referred to as innovation in the KF. Rewriting equation (10) as $K_t y_t + (I - K_t C_t)\, x_t^f$, we see that the analysis step integrates y_t and x_t^f by taking linear combination [8]. For the derivation, we refer the reader to [11] where the linear update (10) is first defined, and then the optimal K_t is thus found via minimization, and [8] where they utilize the properties of Gaussian distribution.

The KF suffers from two major problems: high dimensionality and non-linearity. The computational cost for one iteration is generally $O(n^3 + mn^2 + m^3)$, and the memory storage cost for P_t^f is $O(n^2)$. The update could be computationally expensive when n or m is high. Several works have given efficient methods to overcome this challenge, such as the EnKF, which leverages $q < n, m$ ensemble members to reduce the storage cost [8] as well as the LEnKF, which uses both q ensemble members and a subsampling technique in the spirit of SGLD to overcome large m problem [13]. As for the non-linearity, if h and/or f are non-linear, the propagation above does not yield the desired result. It is generally hard to propagate the error covariances correctly in the non-linear systems largely because expectation operates only linearly. Some well-known methods were proposed to bypass the non-linearity. For example, the XKF applies a first-order Taylor expansion on f and h, bringing a non-linear problem back to a linear system via the Jacobian matrices [2] [6] [10]. However, the XKF fails to incorporate the non-linearity fully.

2.2 The Ensemble Kalman Filter: An Approximated Version of the Kalman Filter

The ensemble Kalman filter (EnKF) was first proposed by [4]. The original goal was to resolve the computational cost issue in high-dimensional systems. It turned out

that it is very successful in non-linear problems [6] [8] [13]. In fact, the EnKF is a kind of particle filtering algorithm. Particle filtering algorithms have been successful in non-linear, non-Gaussian systems [3]. The basic idea of the EnKF is to represent the posterior distribution $\pi(x_t \mid y_1, ...y_t)$ by a set of q analysis ensemble $\{x_t^{a,1}, ..., x_t^{a,q}\}$ and to propagate these ensemble members/particles over time.

Besides particle filtering, the EnKF can be considered an approximated version of the KF. The EnKF does not propagate the error covariances; instead, it approximates them by the ensemble, e.g., the analysis covariance is estimated by

$$\hat{P}_t^a = \frac{1}{q-1} E_t^a (E_t^a)^T \in \mathbb{R}^{n \times n}, \tag{13}$$

$$E_t^a = [x_t^{a,1} - \bar{x}_t^a \cdots x_t^{a,q} - \bar{x}_t^a] \in \mathbb{R}^{n \times q}, \tag{14}$$

which is the sample covariance of the analysis ensemble $\{x_t^{a,1}, ..., x_t^{a,q}\}$. Therefore, there is no need for storing the $n \times n$ matrices P_t^a and P_t^f, reducing the computational cost when x_t is high-dimensional. Further reduction can be achieved by serial updating [8].

The full scheme of the EnKF in a general state-space model is given below [6]. Given an initial ensemble $\{x_0^{a,1}, ..., x_0^{a,q}\}$ and assume $\{x_{t-1}^{a,1}, ..., x_{t-1}^{a,q}\} \sim \pi(x_{t-1} \mid y_1, ..., y_{t-1})$ approximately, the EnkF iterates between the following two steps like the KF

- Forecast step:

$$x_t^{f,i} = f\left(x_{t-1}^{a,i}, u_{t-1}\right) + \underbrace{w_{t-1}^i}_{\text{Additional noise}}, \tag{15}$$

$$\bar{x}_t^f = \frac{1}{q} \sum_{i=1}^{q} x_t^{f,i}. \tag{16}$$

- Analysis step:

$$\hat{K}_t = \hat{P}_{xy_t}^f \left(\hat{P}_{yy_t}^f\right)^{-1}, \tag{17}$$

$$x_t^{a,i} = x_t^{f,i} + \underbrace{\hat{K}_t}_{\text{Approximated by ensembles}} \left(y_t + v_t^i - h\left(x_t^{f,i}\right)\right) = x_t^{f,i} + \hat{K}_t \left(y_t - y_t^{f,i}\right), \tag{18}$$

for all ensemble $i = 1, ..., q$ and

$$\bar{x}_t^a = \frac{1}{q} \sum_{i=1}^{q} x_t^{a,i}, \tag{19}$$

$$y_t^{f,i} = h\left(x_t^{f,i}\right) + \underbrace{v_t^i}_{\text{Additional noise}}, \tag{20}$$

$$\hat{P}_{xy_t}^f = \frac{1}{q-1} E_t^f \left(E_{y_t}^f\right)^T, \ \hat{P}_{yy_t}^f = \frac{1}{q-1} E_{y_t}^f \left(E_{y_t}^f\right)^T, \tag{21}$$

$$E_t^f = \left[x_t^{f,1} - \bar{x}_t^f \cdots x_t^{f,q} - \bar{x}_t^f\right], \ E_{y_t}^f = \left[y_t^{f,1} - \bar{y}_t^f \cdots y_t^{f,q} - \bar{y}_t^f\right]. \tag{22}$$

where $y_t^{f,i} = h(x_t^{f,i}) + v_t^i$ is called perturbed observations, $w_t^i \sim N(\mathbf{0}, Q_t)$ and $v_t^i \sim N(\mathbf{0}, R_t)$. $\bar{x}_t^a$ can thus be used as the state estimate and $\hat{P}_t^a$ can be used as the state covariance estimate.

In the linear measurement case where $h(\cdot) = C_t\cdot$, the estimated Kalman gain $\hat{K}_t$ can be simplified as $\hat{K}_t = \hat{P}_t^f C_t^T (C_t \hat{P}_t^f C_t^T + R_t)^{-1}$ where $\hat{P}_t^f = E_t^f (E_t^f)^T/(q-1)$ and the perturbed observations are of the form $y_t^{f,i} = C_t x_t^{f_i} + v_t^i$. From now on, we only focus on cases where $h(\cdot) = C_t\cdot$ unless further notification. In such cases, the EnKF is the KF with an added noise v_t^i in the innovation term, an estimated Kalman gain $\hat{K}_t$, and a non-linear noisy function projecting $x_{t-1}^{a,i}$ to $x_t^{f,i}$ for all $i = 1, ..., q$. We use under braces to denote the difference between the EnKF and the KF.

The EnKF is essentially a particle filtering algorithm, but it enjoys some advantages over conventional particle filters. The first merit is that the EnKF evolves the particles through a linear shifting by the analysis and forecast steps. This makes the EnKF immune from weight degeneracy, while conventional methods using reweighting or resampling often suffer from this issue [8].

The second advantage is computational efficiency. As pointed out in [13], traditional particle filters often require LU decomposition, which has a cost of $O(n^3)$. In contrast, the cost for the EnKF is $O\left(\max\left\{n^2 m, m^3\right\} + qnm\right)$, meaning that if m and q is less than n, the cost of the EnKF is less than that of the conventional particle filters. Additionally, if m and q are less than n, the cost of the KF is roughly $O(n^3)$, which is larger than that of the EnKF. In summary, if m and q are less than n, the EnKF achieves dimension reduction [8]. An ensemble size of 50 to 100 is typically enough for thousands of states [6].

The EnKF is suboptimal since it uses Monte Carlo samples to approximate the error statistics. Furthermore, it has two significant issues: the *convergence problem* and *filter divergence* [2] [5] [13]. Due to a finite q, the representation of the posterior distribution using the ensemble is generally poor. This leads to a systematic *underestimation* of the posterior covariance P_t^a. In severe cases, this results in filter divergence, where the filter overly relies on the forecast values and ignores the observations. This can be seen by rewriting $\hat{K}_t$ as

$$\hat{K}_t = \left(C_t^T R_t^{-1} C_t + (\hat{P}_t^f)^{-1}\right)^{-1} C_t^T R_t^{-1}$$

by using the Woodbury identity twice [13]. When $\hat{P}_t^f$ is small, $\hat{K}_t$ becomes small as well. Common solutions to this problem include covariance inflation [5] and localization [2] [5] [8].

Another significant issue is the convergence. It is well-known that *the EnKF only converges to a mean-field filter, meaning that it only provides the correct state estimate (the mean of the filtering distribution) but not the correct covariance when $q \to \infty$, except in the linear Gaussian case* [13]. This hinders the ability to quantify uncertainty in the EnKF. The resulting confidence intervals often fall short due to the *underestimation* of $\hat{P}_t^f$, e.g., the $1-\alpha$ confidence interval for the $d-$th dimension of the state $\bar{x}_{t,d}^a \pm Z_{1-\alpha/2}\sqrt{S_{t,d}}$ (for the Gaussian noise cases) where S is the sample variance of $\{x_{t,d}^{a,1}, ..., x_{t,d}^{a,q}\}$ for $d = 1, ..., n$ such that $x_t = (x_{t,1}, ...x_{t,d}, ..., x_{t,n})$, often captures less than $1-\alpha$ of the cases as will be shown in numerical experiments. These drawbacks motivate the LEnKF algorithm.

[5] Although we do not report the results of naively chosen inflation factor EnKF. In our experiment, the naively inflated EnKF performs poorly. This shows the importance of tuning the inflation factor.

2.3 The Langevinized Ensemble Kalman Filter: A Scalable Ensemble Kalman Filter for Big Data

To address the issues of convergence and scalability of the EnKF, [13] proposed the LEnKF. The basic idea of the LEnKF is to combine the subsampling technique deployed in the SGLD and the forecast-analysis type update in the EnKF. The former makes the algorithm scalable with respect to the number of observations and the latter makes it efficient and immune to weight degeneracy. Here, since we only focus on the linear measurement case. We only discuss Section 3.3. in [13]. Readers are referred to the paper to learn more about how the LEnKF can be applied in the inverse problems and non-linear measurement data assimilation problems.

The LEnKF for data assimilation is motivated by the following fact of the filtering distribution $\pi(x_t \mid y_{1:t})$:

$$\begin{aligned} \pi(x_t \mid y_{1:t}) &\propto \pi(x_t, y_{1:t-1}, y_t) && (23)\\ &\propto f(y_t \mid x_t, y_{1:t-1})\pi(x_t \mid y_{1:t-1}) && (24)\\ &= \underbrace{f(y_t \mid x_t)}_{\text{Likelihood}} \underbrace{\pi(x_t \mid y_{1:t-1})}_{\text{Prior}}, && (25) \end{aligned}$$

which suggests using the predictive distribution $\pi(x_t \mid y_{1:t-1})$ as the prior. In order to us the SGLD in the prior, we need an unbiased estimate of the prior gradient $\nabla \log \pi(x_t \mid y_{1:t-1})$. The estimate can be found using a Bayesian version of Fisher's identity. Let D be data, β be the parameter of interest, and γ be a latent variable, then the following identity holds

$$\nabla_\beta \log \pi(\beta \mid D) = \int \nabla_\beta \log \pi(\beta \mid \gamma, D)\pi(\gamma \mid \beta, D)\, d\gamma. \tag{26}$$

Equation (26) provides a way for using a Monte Carlo estimator to unbiasedly estimate the log gradient. [13] applies the identity to get the estimate; that is

$$\begin{aligned} \nabla \log \pi(x_t \mid y_{1:t-1}) &= \int \nabla_{x_t} \log \pi(x_t \mid x_{t-1}, y_{1:t-1})\pi(x_{t-1} \mid x_t, y_{1:t-1})\, dx_{t-1}\\ &= \int \nabla_{x_t} \log \pi(x_t \mid x_{t-1})\pi(x_{t-1} \mid y_{1:t-1})w(x_t \mid x_{t-1})\, dx_{t-1}, \end{aligned}$$

where $w(x_t \mid x_{t-1}) = \pi(x_{t-1} \mid x_t, y_{1:t-1})/\pi(x_{t-1} \mid y_{1:t-1}) = \pi(x_t \mid x_{t-1})/\pi(x_t \mid y_{1:t-1}) \propto \pi(x_t \mid x_{t-1})$ [6]. This means sampling from $\pi(x_{t-1} \mid x_t, y_{1:t-1})$ is the same as resampling from $\mathcal{X}_{t-1} = \{x_{t-1,1}, ..., x_{t-1,q}\}$ with weight $w(x_t \mid x_{t-1})$ given that $\mathcal{X}_{t-1} \sim \pi(x_{t-1} \mid y_{1:t-1})$. w is easy to compute since it is just a density of multivariate Gaussian and $\nabla_{x_t} \log \pi(x_t \mid x_{t-1})$ is easy to evaluate.

Therefore, if $\tilde{x}_{t-1}$ is an importance sample from $\mathcal{X}_{t-1}$ with weight $w \propto \pi(x_t \mid x_{t-1})$, then a self-normalizing estimator $\nabla_{x_t} \log \pi(x_t \mid \tilde{x}_{t-1})$, which is asymptotically unbiased [7]. However, in practice the number of importance samples k cannot go to infinity. The authors dealt with this case in the paper and the convergence was well established. Please refer to Algorithm 4 and remarks in the paper [13].

[6] Since we are evaluating the gradient at x_t given data, x_t and data are given as constant.

[7] This is why in the paper we need the third loop k.

3 The Main Idea

It is noticed that in Algorithm 4 of [13], when updating using the whole data set, the Kalman gain is $K_{t,k} = Q_{t,k}H_{t,k}^T \left(H_{t,k}Q_{t,k}H_{t,k}^T + 2R_t\right)^{-1}$ where $Q_{t,k} = \epsilon_{t,k}I_p$. This is crucial for the convergence. Inspired by this observation, in this study, we are interested in the question: *Would the EnKF converge to the correct distribution if we use* $\hat{K}_t = \hat{P}_t^f C_t^T (C_t \hat{P}_t^f C_t^T + 2R_t)^{-1}$ *as the Kalman gain and* $y_t^{f,i} = h(x_t^{f,i}) + v_t^i$ *where* $v_t^i \sim N(\mathbf{0}, 2R_t)$*?* We study this question through three numerical examples in the next section.

The new algorithm mentioned above is denoted as the $2R$ EnKF in the following studies. It is known that, in practice, researchers use covariance inflation $\alpha^2 \hat{P}_t^a$ where $\alpha \geq 1$ or localization $\tilde{P}_t^f = \hat{P}_t^f \circ \mathcal{T}_t$ where $\circ$ denotes the Hadamard (entrywise) product, and $\mathcal{T}_t$ is a sparse positive definite correlation matrix, to fix the problem [2]. We found in our experiment (not shown here) that a wrong inflation factor could make inference worse than a naively chosen inflation factor makes the results worse, hinting that tuning α is nontrivial.

Before we present any numerical results. We provide some simple analytical study. In the $2R_t$ algorithm, given the estimates we have

$$x_t^{a,i} = x_t^{f,i} + \tilde{K}_t \left(y_t + v_t^i - h\left(x_t^{f,i}\right)\right) = x_t^{f,i} + \tilde{K}_t \left(y_t - y_t^{f,i}\right),$$
$$\tilde{K}_t = \hat{P}_t^f C_t^T \left(C_t \hat{P}_t^f C_t^T + 2R_t\right)^{-1},$$

where $y_t^{f,i} = C_t x_t^{f,i} + v_t^i$ is called perturbed observations and $v_t^i \sim N(0, 2R_t)$. The variance from the innovation term inflates but the $2R_t$ term in the Kalman deflates the variance. Hence it is hard to see immediately how would this affect the EnKF when q is large.

4 Numerical Studies

This section presents three numerical studies. We confine ourselves to only the linear measurement cases; however, the state process h could be non-linear. To better cover different cases, the three cases include a low-dimensional linear model, a low-dimensional non-linear model, and a high-dimensional linear model. We focus on state estimation accuracy regarding MSEs and empirical coverage probabilities.

$$\text{MSE} = \frac{1}{T}\sum_{t=1}^{T}(x_t^* - \bar{x}_t^a)^T(x_t^* - \bar{x}_t^a), \tag{27}$$

$$\text{Coverage} = \frac{1}{mT}\sum_{d=1}^{m}\sum_{t=1}^{T} I(x_{t,d}^* \in \bar{x}_{t,d}^a \pm Z_{1-\alpha/2}\sqrt{S_{t,d}}), \tag{28}$$

where the subscript d denotes the dimension of a vector. In the second example, the coverage is not computed over all the dimensions; instead, we directly report the result for each dimension, i.e., $\text{Coverage} = \sum_{t=1}^{T} I(x_{t,d}^* \in \bar{x}_{t,d}^a \pm Z_{1-\alpha/2}\sqrt{S_{t,d}})/T$ and $\text{MSE}_d = \sum_{t=1}^{T}(x_{t,d}^* - \bar{x}_{t,d}^a)/T$ for $d = 1, 2$. $*$ denotes the true values. We study the convergence behavior of the $2R$ EnKF numerically, compared to that of the plain EnKF.

4.1 Ornstein-Uhlenbeck Model

The Ornstein-Uhlenbeck model is fundamental and useful in many applications such as interest rate modeling in finance due to its mean-reverting property. In this example, we study its discrete-time equivalence. The equivalent discretization of the Ornstein-Uhlenbeck model is given by [10]:

$$x\left(t_{k+1}\right) = a_k x\left(t_k\right) + q_k, \tag{29}$$

$$y_k = x\left(t_k\right) + r_k, \tag{30}$$

where $q_k \sim \mathrm{N}\left(0, \Sigma_k\right), r_k \sim \mathrm{N}(0, R), a_k = \exp\left(-\lambda \Delta t_k\right)$, and $\Sigma_k = q\lambda[1 - \exp\left(-2\lambda\Delta t_k\right)]/2\lambda$.

In this example, the KF can be served as the benchmark since it is a linear Gaussian model. The coverage of the KF is about 95%, which is correct. We ran 100 independent experiments and recorded the coverage probabilities and MSEs for three types of filters. Table 1 reports the results from one run. Figures 1d, and 1a show the results of 5 ensemble members; while figures 1c and 1b show those of 30 ensemble members.

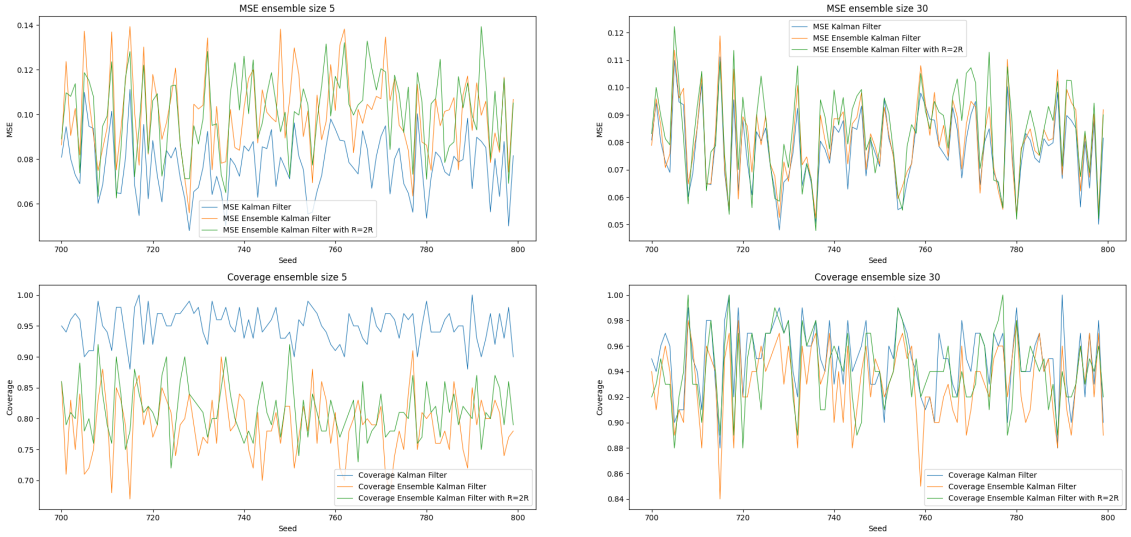
A larger ensemble size does not seem to improve accuracy a lot, but it improves uncertainty quantification. As pointed out previously, a small ensemble systematically underestimates the posterior variance. This is present in the "short" coverage in table 1, figure 1b, and figure 1a. In the $q = 5$ case (the plain EnKF), none of the coverage probabilities is within $95\% \pm 3\%$; while in the $q = 30$, 75 out of 100 cases, the coverage probabilities are within $95\% \pm 3\%$.

In low-dimensional cases, using an ensemble size of 30 is generally enough as the size used in high-dimensional cases is usually between 50 and 100 [8]. In this example, when $q = 30$, $2R$ EnKF and EnKF have roughly the same MSEs; however, the coverage of $2R$ EnKF is better than that of EnKF in 62% of the cases as shown in figure 1b [8]. Besides, increasing the ensemble size from 5 to 30 significantly improves the MSEs and coverage for the $2R$ EnKF and the EnKF. The result of the large ensemble size is not shown here since in linear Gaussian case, it is well-known that the EnKF correctly converges to the filtering distribution.

Table 1: Ornstein-Uhlenbeck Model. Comparison of Kalman Filter (KF) and Ensemble Kalman Filter (EnKF) with Different Ensemble Sizes (Random Seed 701).

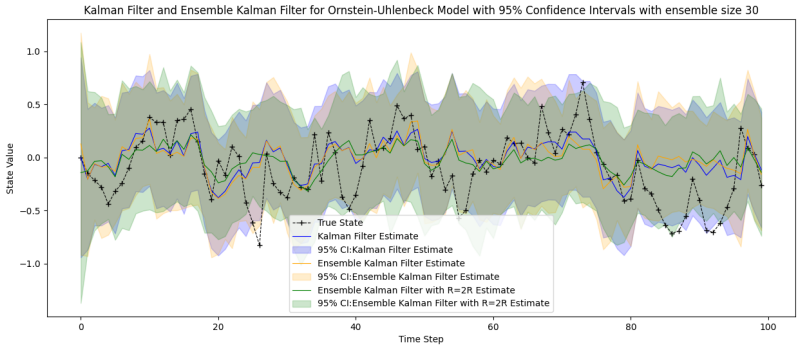
Method	**Ensemble Size 5**		**Ensemble Size 30**	
	Coverage Probability	**MSE**	**Coverage Probability**	**MSE**
KF	0.94	0.095	0.94	0.095
EnKF	0.71	0.120	0.91	0.096
EnKF with R=2R	0.79	0.110	0.93	0.100

[8] In our numerical experiments, we also tried a naive variance inflation approach with $\alpha = 1.5$. We report that the MSEs are roughly the same; however, the coverage is way shorter than those of $2R$ EnKF and EnKF. The choice of α is challenging and highly problem-dependent.

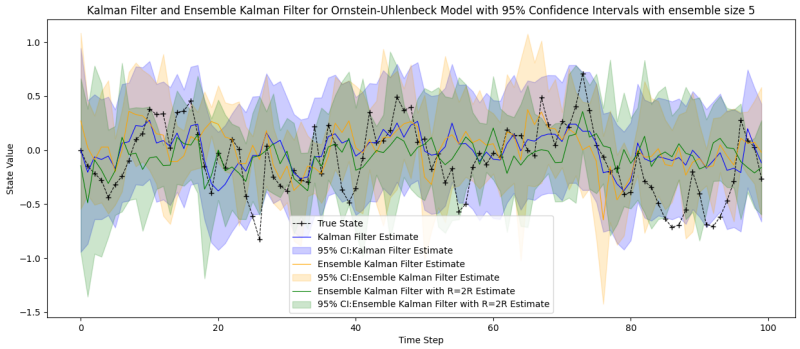


(a) MSE Coverage: Ensemble size 5.

(b) MSE Coverage: Ensemble size 30.



(c) Kalman vs. EnKF: Ensemble size 30.



(d) Kalman vs. EnKF: Ensemble size 5.

Figure 1: Ornstein-Uhlenbeck Model. (a) MSE Coverage: Ensemble size 5. (b) MSE Coverage: Ensemble size 30. (c) Kalman vs. EnKF: Ensemble size 30. (d)Kalman vs. EnKF: Ensemble size 5.

4.2 Van der Pol Oscillator

Now we study to a more complicated case: a low-dimensional non-linear case: The Van der Pol oscillator. The Van der Pol oscillator is widely applied in biology, engineering, physics, etc. Assume that the Van der Pol oscillator is driven by $w_t \sim$

$N(\mathbf{0}, Q)$. The state process is given by [6]:

$$x_{t+1} = f(x_t) + w_t, \tag{31}$$

$$f(x_t) = \begin{bmatrix} x_{1,t} + hx_{2,t} \\ x_{2,t} + h\left(\alpha\left(1 - x_{1,t}^2\right)x_{2,t} - x_{1,t}\right) \end{bmatrix}, \tag{32}$$

where $x_t = (x_{1,t}\ x_{2,t})^T$ and h is the step size. We assume that only one of the dimensions $x_{1,t}$ or $x_{2,t}$ at a time can be observed with corruption for all $t \geq 0$, i.e.,

$$y_t = Cx_t + v_t, \tag{33}$$

where $v_t \sim N(\mathbf{0}, R)$ and C selects $x_{1,t}$ or $x_{2,t}$. In our experiments, we set $\alpha = 1$ and the step size $h = 0.1$ to stabilize the discrete-time system. The noises are set to be $Q = \text{diag}(0.0262, 0.008)$ and $R = 0.5$ [9], respectively. The total time steps is 500 and C selects $x_{2,t}$ for all t.

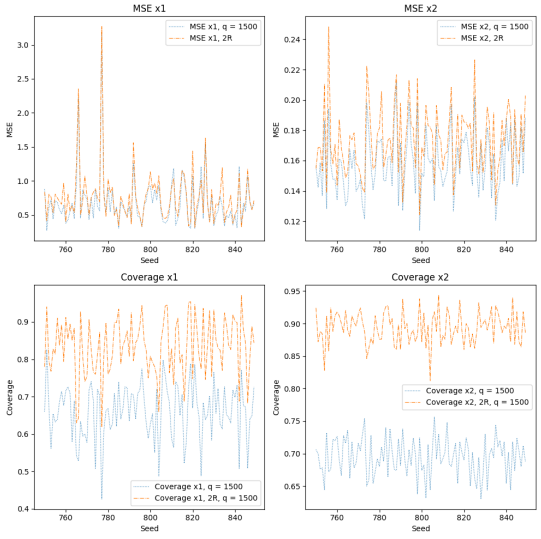
Figure 3 indicates that a larger ensemble improves the estimation. Figure 2 shows the resulting MSEs, coverage probability, and state estimations for two ensemble sizes 1500 and 30 over 100 data sets. While table 2 reports the result from one data set. The 1500 ensemble size case is studied since we are interested in the convergence of the $2R$ EnKF. The ensemble size of 30 is generally enough for state estimation when x_t is low-dimensional as implied from figures 2a and 2d: increasing the ensemble size from 30 to 1500 does not improve the accuracy significantly. The coverage, however, improves significantly, especially in $x_{2,t}$.

Furthermore, the MSEs of the two methods are roughly the same; but the coverage of the $2R$ EnKF is way better than that of the EnKF. Although in this case 1500 might still not be enough to achieve 95% coverage.

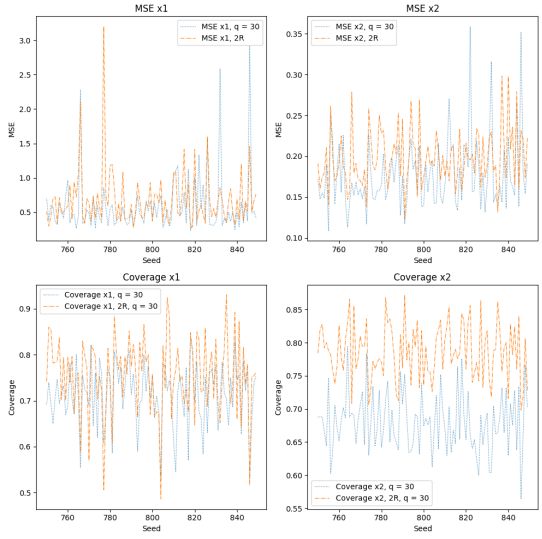
Table 2: Van der Pol Oscillator. Comparison of Ensemble Kalman Filter (EnKF) and $2R$ EnKF with Different Ensemble Sizes (Random Seed 801).

Method	**Ensemble Size 30**		**Ensemble Size 1500**	
	Coverage Probability	**MSE**	**Coverage Probability**	**MSE**
EnKF $x_{1,t}$	0.67	0.63	0.65	0.67
EnKF $x_{2,t}$	0.69	0.19	0.68	0.15
EnKF with R=2R $x_{1,t}$	0.66	0.76	0.79	0.88
EnKF with R=2R $x_{2,t}$	0.79	0.19	0.92	0.16

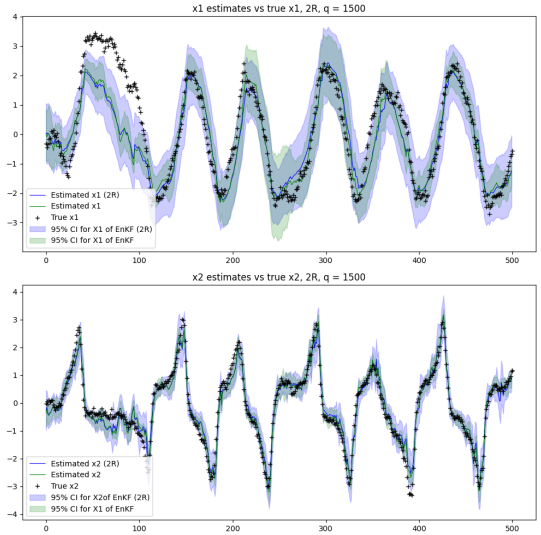
[9] $R = 0.5$ results a larger observation variation in $x_{2,t}$ than [6]. Other parameters are the same



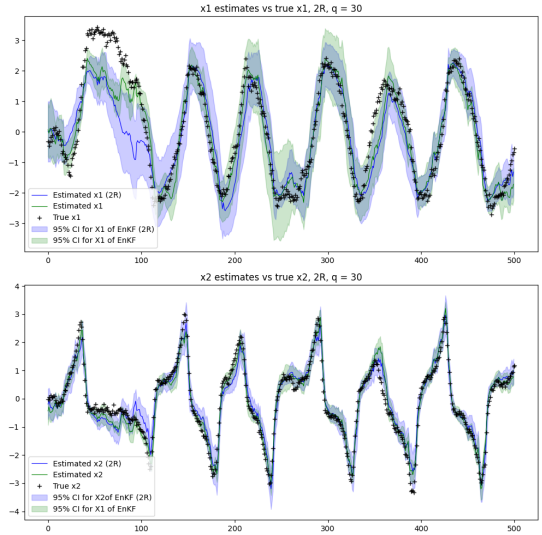
(a) MSE Coverage: Ensemble size 1500.



(b) MSE Coverage: Ensemble size 30.



(c) $2R$ EnKF vs. EnKF: Ensemble size 1000.



(d) $2R$ EnKF vs. EnKF: Ensemble size 30.

Figure 2: Van der Pol Oscillator. (a) MSE Coverage: Ensemble size 30. (b) MSE Coverage: Ensemble size 1500. (c) $2R$ EnKF vs. EnKF: Ensemble size 30. (d)$2R$ EnKF vs. EnKF: Ensemble size 1500.

4.3 Lorenz-96 Model

The last example is the Lorenz-96 model used in [13]. This model possesses a chaotic behavior as shown in figures 5 and 6 is widely applied in weather forecasting. In this example, the state process is highly non-linear and the state dimension and observation are a bit higher. Specifically, we work on a 40-dimensional Lorenze-96 model with the measurement dimension being 20. The model is given by

$$\frac{dx^i}{dt} = \left(x^{i+1} - x^{i-2}\right) x^{i-1} - x^i + F, \quad i = 1, 2, ..., p \tag{34}$$

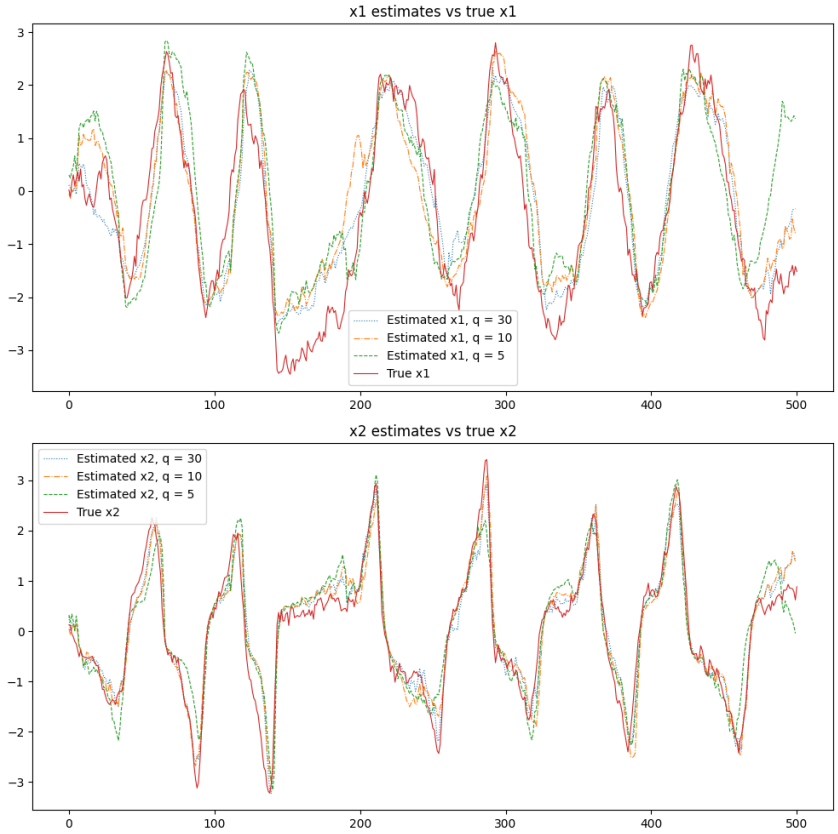


Figure 3: EnKF with different ensemble sizes.

where the forcing constant is $F = 8$, which is known to cause the chaotic behavior, and $p = 40$. Additional conditions include: $x^{-1} = x^{p-1}, x^0 = x^p$, and $x^{p+1} = x^1$.

We generate data according to [13]. x_0 was set as 20 for every element but $x_{0,20} = 20.01$. The differential equation was solved using the fourth-order Runge-Kutta (RK4) numerical method with a $\Delta t = 0.01$ time interval. and for each i and t. The generated states were added to $x_{t,i}$ a random noise generated from $N(0, 1)$. The observation model is

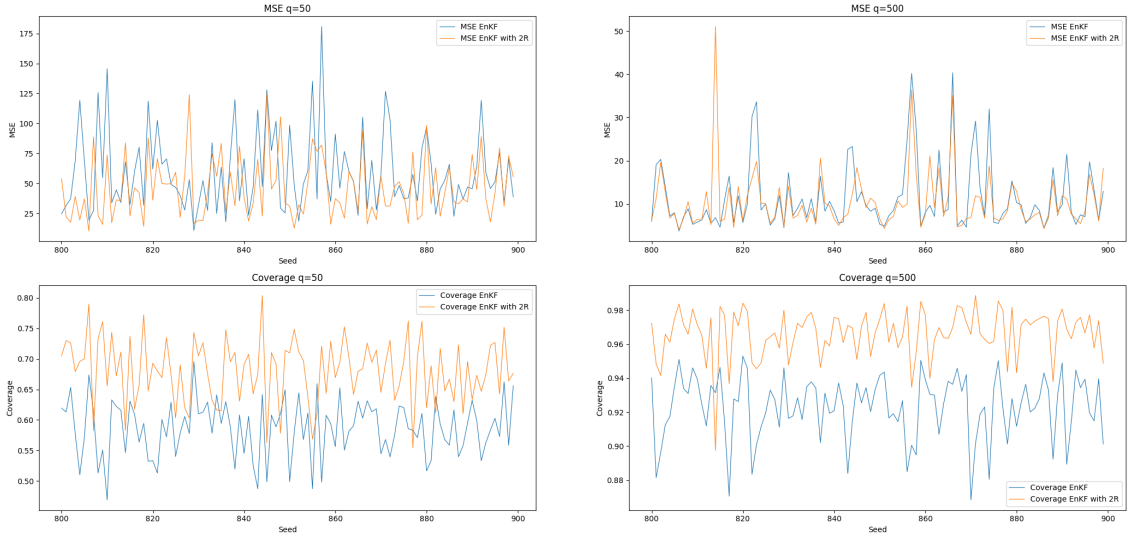
$$y_t = C_t x_t + \epsilon_t, \quad t = 1, 2, \ldots, T, \tag{35}$$

where $\epsilon_t \sim N\left(0, I_{p/2}\right)$, and C_t is a random selection matrix that selects 20 dimensions from x_t. T was set as 100.

Table 3 demonstrates the result from one data set. Figures 4a and 4b show the resulting MSEs and coverage of $q = 50$ and $q = 500$ for 100 data sets, respectively. The $q = 500$ case is included for the convergence study of the $2R$ EnKF. As in previous examples, the $2R$ EnKF has better coverage in both cases and both methods have similar MSEs. In figure 4b, we see that the $2R$ EnKF has coverage around 95% but that of the EnKF still falls short.

Table 3: Lorenz-96 model. Comparison of Ensemble Kalman Filter (EnKF) and $2R$ EnKF with Different Ensemble Sizes (Random Seed 801).

Method	**Ensemble Size 50**		**Ensemble Size 500**	
	Coverage Probability	**MSE**	**Coverage Probability**	**MSE**
EnKF	0.61	32	0.88	22
EnKF with R=2R	0. 73	22	0.95	11



(a) MSE Coverage: Ensemble size 50.

(b) MSE Coverage: Ensemble size 500.

Figure 4: Lorenz-96 Model. (a) MSE Coverage: Ensemble size 50. (b) MSE Coverage: Ensemble size 500.

5 Discussion

In this study, we study three cases comparing the plain EnKF and the $2R$ EnKF, which is inspired by the LEnKF. In most cases, the MSEs from both methods are roughly the same. *However, the coverage of the* $2R$ *EnKF is consistently better than that of the plain EnKF. In certain instances, the coverage approaches 95%.* This highlights the impact of using $2R_t$. Specifically, the inflation term in the innovation outweighs the deflation term in the Kalman gain. Here, we also want to point out that $2R$ EnKF is superior than covariance inflation since it does not require careful tuning, which sometimes might lead to worse results.

Furthermore, we observe that as q increases, the MSEs remain approximately the same compared to cases with lower q for both methods, empirically proving the EnKF's ability to correctly estimate the state mean. Nevertheless, the poor coverage ability of the EnKF has been proven as well.

This preliminary study offers promising evidence for the convergence properties of using $2R_t$. However, its effects on the original EnKF remain unclear. A deeper investigation into the $2R_t$ algorithm and its asymptotic behavior would be an interesting direction for future research.

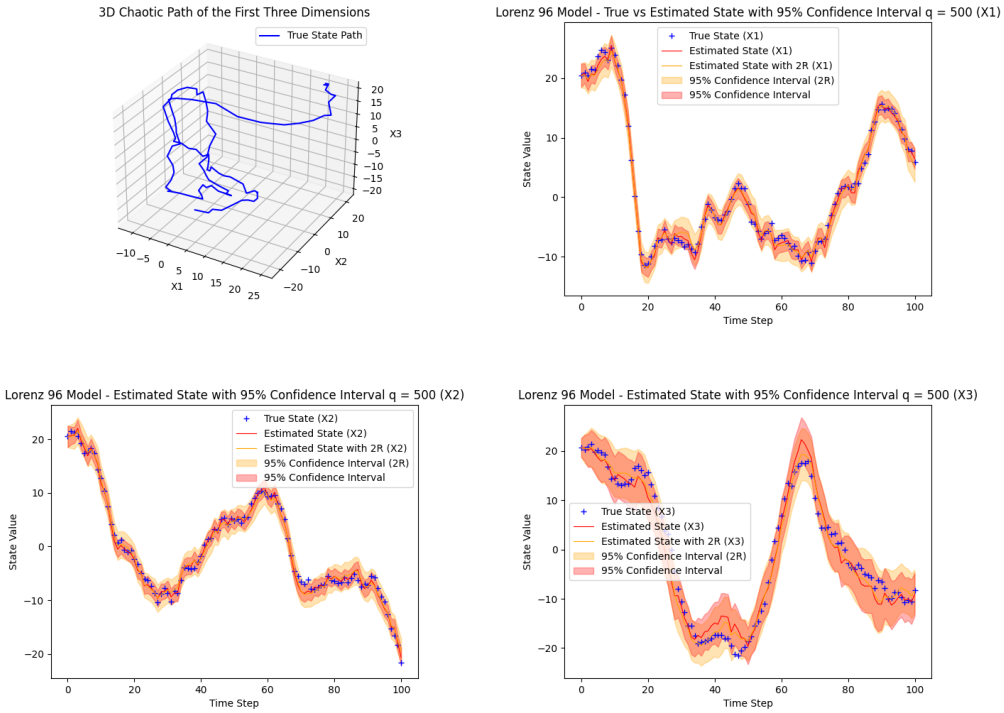


Figure 5: Results for the first three dimensions: Ensemble size 500

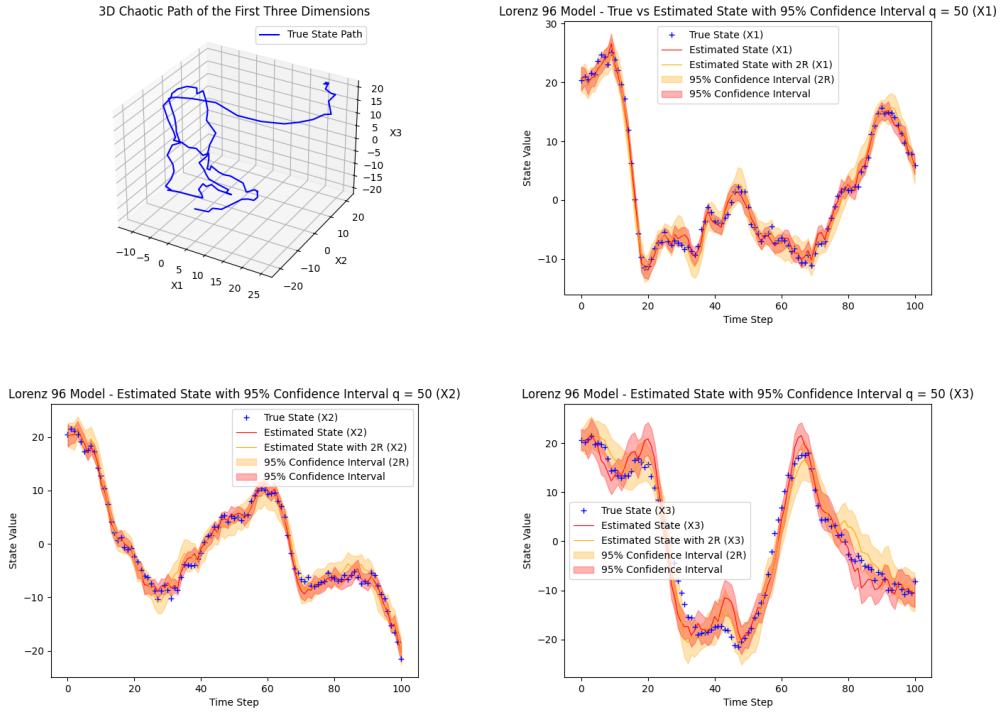


Figure 6: Results for the first three dimensions: Ensemble size 50

References

[1] Jeffrey L Anderson. An ensemble adjustment kalman filter for data assimilation. *Monthly weather review*, 129(12):2884–2903, 2001.

[2] Eviatar Bach, Ricardo Baptista, Daniel Sanz-Alonso, and Andrew Stuart. Inverse problems and data assimilation: A machine learning approach. *arXiv preprint arXiv:2410.10523*, 2024.

[3] Arnaud Doucet, Adam M Johansen, et al. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

[4] Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994.

[5] Geir Evensen, Femke C Vossepoel, and Peter Jan Van Leeuwen. *Data assimilation fundamentals: A unified formulation of the state and parameter estimation problem*. Springer Nature, 2022.

[6] Steven Gillijns, O Barrero Mendoza, Jaganath Chandrasekar, BLR De Moor, Dennis S Bernstein, and A Ridley. What is the ensemble kalman filter and how well does it work? In *2006 American control conference*, pages 6–pp. IEEE, 2006.

[7] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[8] Matthias Katzfuss, Jonathan R Stroud, and Christopher K Wikle. Understanding the ensemble kalman filter. *The American Statistician*, 70(4):350–357, 2016.

[9] Jun S Liu and Jun S Liu. *Monte Carlo strategies in scientific computing*, volume 10. Springer, 2001.

[10] Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

[11] Neil Thacker and A Lacey. Tutorial: The kalman filter. *Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester*, 61, 1998.

[12] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

[13] Peiyi Zhang, Qifan Song, and Faming Liang. A langevinized ensemble kalman filter for large-scale static and dynamic learning. *arXiv preprint arXiv:2105.05363*, 2021.