Particle filtering with Lagrangian data in a point vortex model

by

Subhadeep Mitra

B.Tech., Mechanical Engineering, Indian Institute of Technology Madras
(2010)

Submitted to the School of Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Computation for Design and Optimization

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2012

Author ....
School of Engineering
May 24, 2012

Certified by ....
Youssef Marzouk
Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by ....
Nicolas Hadjiconstantinou
Director, Computation for Design and Optmization

Particle filtering with Lagrangian data in a point vortex model

by

Subhadeep Mitra

Submitted to the School of Engineering
on May 24, 2012, in partial fulfillment of the
requirements for the degree of
Master of Science in Computation for Design and Optimization

Abstract

Particle filtering is a technique used for state estimation from noisy measurements. In fluid dynamics, a popular problem called Lagrangian data assimilation (LaDA) uses Lagrangian measurements in the form of tracer positions to learn about the changing flow field. Particle filtering can be applied to LaDA to track the flow field over a period of time. As opposed to techniques like Extended Kalman Filter (EKF) and Ensemble Kalman Filter (EnKF), particle filtering does not rely on linearization of the forward model and can provide very accurate estimates of the state, as it represents the true Bayesian posterior distribution using a large number of weighted particles. In this work, we study the performance of various particle filters for LaDA using a two-dimensional point vortex model; this is a simplified fluid dynamics model wherein the positions of vortex singularities (point vortices) define the state. We consider various parameters associated with algorithm and examine their effect on filtering performance under several vortex configurations. Further, we study the effect of different tracer release positions on filtering performance. Finally, we relate the problem of optimal tracer deployment to the Lagrangian coherent structures (LCS) of point vortex system.

Thesis Supervisor: Youssef Marzouk
Title: Associate Professor of Aeronautics and Astronautics

Acknowledgments

This work has shaped my understanding of modern day uncertainty quantification. I would like to thank my family for their unconditional encouragement and support at all times. I can be absolutely certain that this thesis would not have been possible without the energy, motivation and insight provided by my research advisor, Dr. Youssef Marzouk. I would also like to thank the Department of Energy, Office of Advanced Scientific Computing Research, for funding this work (grant number DE-SC0003908).

Contents

List of Figures

Chapter 1

Introduction

Data assimilation into models have recently gained a lot of prominence in almost all areas of geophysics. The ability to run extensive simulations of ever-faster computers and ever-improving observational technologies have tremendously improved the feasibility of these data-assimilation problems. Data assimilation can be defined as the problem of estimating the optimal prediction that combines model output, with its attendant uncertainty, and observations, which will also contain errors [84]. Stuart et al [3] give a good description of data assimilation, from a statistical perspective, in the context of fluid problems. In data assimilation applications, the analysis and forecasts are best thought of as probability distributions. The analysis step is an application of the Bayes theorem and the overall assimilation procedure is an example of Recursive Bayesian estimation. A good overview of Bayesian data assimilation can be found in the following references [29, 2, 8, 97, 78]. Traditionally, techniques of data assimilation are based on control theory and optimization. Their adaptation to highly nonlinear fluid flows, as in atmospheric and oceanic dynamics, presents many mathematical challenges. Protas [74] considers a good example of a flow control problem arising out of data assimilation.

In this study, we focus on the statistical method of filtering based on measurements. In filtering, at every stage there is new data appearing in form of measurements, based on which one has to predict the estimates of the current states. The data that we use for our filtering problem is Lagrangian as opposed to Eulerian. Lagrangian observations refer to sequences of position measurements along trajectories of (Lagrangian) tracers. Such data are abundant in geophysical systems. A new approach for assimilating Lagrangian data into the fluid dynamical systems was introduced by Ide et al. [41]. The essential idea behind the approach was to augment the state space of the model by including tracer coordinates as additional model variables, thus the resulting model directly relates to the Lagrangian observations and the technique is called Lagrangian data assimilation (LaDA).

The filtering problem was first formulated in terms of the extended Kalman filter (EKF)[44, 45, 42], which computes the forecast error-covariance matrix using the linearized model of the system equations. The system model can vary in complexity. Several times in literature, a relatively low-dimensional, computationally tractable model called the point vortex model has been used [21, 4, 41]. Better models, such a vorticity-velocity formulation of the Navier-Stokes equation [22] and the DNS (Direct numerical simulation) model [77], have also been used. In our study, we use the point vortex model. The EKF was applied to the point vortex model by Ide et al. [41] and [50]. A better version of the EKF called the Ensemble Kalman filter (EnKF) [14] has also been applied to the Lagrangian data assimilation problems [90, 39, 66, 75, 49, 57]. Spiller et. al [84] showed that particle filter does better than both EKF and EnKF for a point vortex model. The formulation based on either the EKF or the EnKF approximates the model uncertainty up to second order using the error covariance matrix. When model nonlinearity is strong and the observation period is long, this approximation may become invalid and filter divergence can occur [50, 81].The particle filter (PF), in contrast, is an approach in which, a full distribution of the model uncertainty is attained by a large number of particles [13, 24]. In other words, the resulting distributions are not approximated by Gaussians in the process of particle filtering. Highly nonlinear effects in the flow can therefore be accommodated without causing breakdown of the EKF and EnKF.

In this study, we implement the particle filter with Lagrangian data assimilation into the point-vortex model and investigate the performance of the filter. We consider different vortex configurations, which were not considered by Spiller et al. [84]. Chorin et al. [63, 62, 19, 18] applied an implicit particle filtering approach to tackle the data assimilation problem. In this work, we use the traditional particle filtering algorithms like the Sequential Importance Resampling (SIR). Further, we extend our study to answer the question of "ideal" tracer deployment, which leads to best filtering performance. In particular, we consider Lagrangian coherent structure (LCS) [32, 30, 83] of the flow based on its initial conditions, and relate it to the tracer deployment problem. We divide our study into the following chapters.

In chapter 2, we discuss the particle filtering and related algorithms. First, we look at different state of the art particle filter algorithms like the Sequential importance re-sampling (SIR) and Auxiliary particle filtering (APF) algorithms. Next, we look at the problem of smoothing, where all the data is available at once, and we try to get the best estimates of the state variable. Finally, we look at simultaneous state and parameter estimation, where, we not only need to infer the states of the system, but also, the parameters associated with the system model.

In chapter 3, we derive the point vortex model from basic fluid dynamics principles and also discuss its validity for the filtering problem. We look at a few basic vortex configurations and discuss the numerical integration schemes we use to solve the stochastic version of the point vortex model numerically.

In chapter 4, we introduce the concept of finite-time lyapunov exponents (FTLE) and Lagrangian coherent structures (LCS). FTLE help us to understand the extent to which the flow is chaotic, while LCS are invariant structures arising out of the FTLE field. We discuss their relationship to our "ideal" tracer deployment problem.

In chapter 5, we discuss the numerical results of our study. We first investigate the performance of the filter under different vortex configurations by varying different parameters associated with the particle filtering algorithm. Next, we compare the performance of the SIR and the APF. Finally, we study the effect of tracer release positions on the filtering performance under different vortex configurations.

In chapter 6, we provide a summary of our study, followed by mentioning the key conclusions and scope for future work.

Chapter 2

Particle filtering and related Algorithms

2.1 Particle Filtering

Many scientific problems require the estimation of the unknown states of system that changes with time given a certain amount of noisy data. In order to analyze and make inference about such a dynamic model, mainly two models are required. First, a model describing the evolution of the state with time (called the "system" model) and, second, a model relating the noisy measurements to the hidden states. If these models are available in the probabilistic form, its sets the an ideal stage for a Bayesian based approach. Here, one attempts to construct posterior probability density functions for the states given the measurements or data. Further, there are two kinds of state estimation problems, namely, filtering and smoothing. In filtering, at every stage there is new data appearing in form of measurements, based on which one has to predict the estimates of the current states. Smoothing is a different problem in which the data for all times is available at the beginning of the inference process. Sequential Monte Carlo methods form the backbone of this kind of recursive estimation. Before heading into the filtering problem we discuss some basics which are required for Particle Filtering.

2.1.1 Theory behind particle filtering

Importance Sampling

Importance sampling is a general technique for estimating properties of a particular distribution, while only having samples generated from a different distribution called the proposal distribution rather than the distribution of interest. In any inference problem, we need to sample from the posterior of the variable we wish to estimate. However, in most cases, it is difficult to sample directly from the posterior, and we use the method of importance sampling. In the context of particle filtering we estimate the probability densities using particles with weights. These weights are associated with ratio of the true distribution to the proposal distribution. More formally, we can write the particle approximation of a density function as follows

$$p(x) \approx \sum_{i=1}^{N} w^i \delta(x - x^i), \quad where \quad w^i \propto \frac{p(x^i)}{q(x^i)} \tag{2.1}$$

Hidden Markov Models

Most of the inference problems on sequential state estimation falls under the category of Hidden Markov Models (HMM), where we have a set of dependent variables called $\{Y\}_{t=1}^{T}$ are available in form of measurements and one has to infer the hidden states $\{X\}_{t=1}^{T}$ from the measurements. The graphical model of a HMM is shown in fig. 2-1 below. The system model and observation model, in terms of probability distributions, associated with an HMM is detailed below.

Particle Filtering is usually done on HMMs. However, inference on problems having more complex graphical structure is also being pursued by some researchers and seems to be much more involved. The main issue in non-HMM models lies in the fact that particles from different branches of the graph need to be integrated and propagated forward while doing the inference.

$$X_n \mid X_{n-1} = x_{n-1} \sim f(x_n \mid x_{n-1})$$
$$Y_n \mid X_n = x_n \sim g(y_n \mid x_n)$$
$$X_1 \sim \mu(x_1)$$

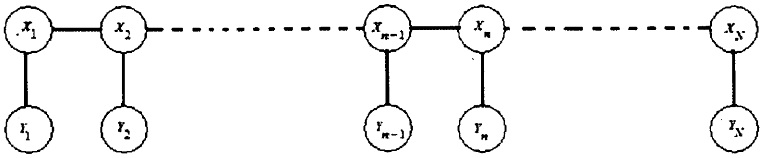


Figure 2-1: Standard Hidden Markov Model

Steps in Particle Filtering

Particle filtering on an HMM consists of two steps. Also, from now on particle filtering on HMMs shall be referred to as plain Particle Filtering. First, the predict step is used to generate samples of the new state given observations till the previous time step. Then these samples are updated in terms of weights through the update step by incorporating the likelihood of the predicted samples. These two steps are described mathematically as follows. Also, a pictorial description of the same is given in fig. 2-2.

Predict Step:

$$\textit{Given} \quad p_{t|t}(x_t|y_{1:t})$$

$$p_{t+1|t}(x_{t+1}|y_{1:t}) = \int_{x_t} p(x_{t+1}|x_t)p_{t|t}(x_t|y_{1:t})\mathrm{d}x_t \tag{2.2}$$

Update Step:

$$p_{t+1|t+1}(x_{t+1}|y_{1:t+1}) \propto p(y_{t+1}|x_{t+1})p_{t+1|t}(x_{t+1}|y_{1:t}) \tag{2.3}$$

As we mentioned before, it is often difficult to sample from the true posterior distributions in inference problems and hence, we use a proposal distribution $q_(x_t|y_{1:t})$. We sample

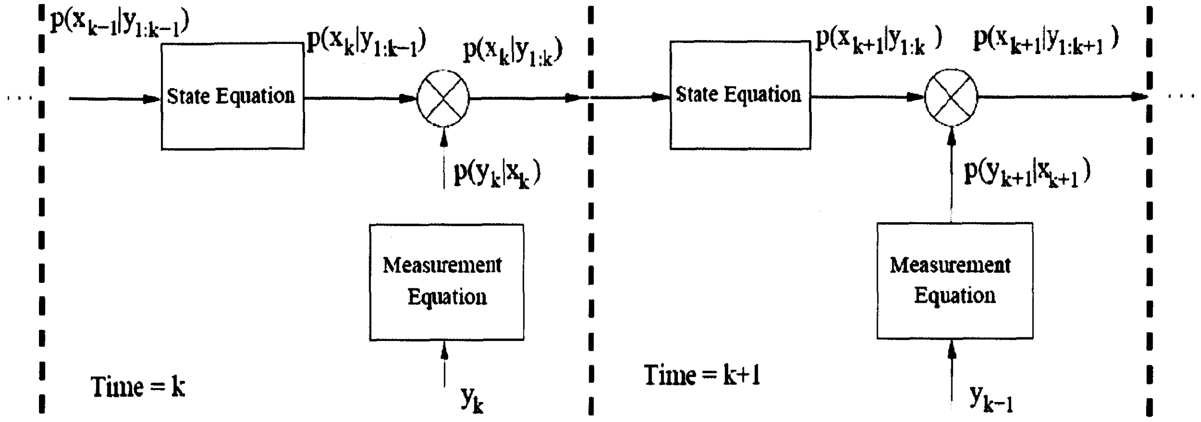


Figure 2-2: Steps in Particle Filtering [5]

values of x_t from the proposal distribution and associate weights w_t with each of these samples based on the ratio of the true distribution and the proposed distribution. Once this is done, we have particle approximation of the posterior of the states. Mathematically,

$$p(x_t|y_{1:t}) \approx \sum_{i=1}^{N} w_t^i \delta(x_t - x_t^i), \quad where \quad w_t^i = \frac{p(x_t^i|y_{1:t})}{q(x_t^i|y_{1:t})}, x_t^i \sim q(x_t^i|y_{1:t}) \tag{2.4}$$

We also assume the importance density to factorize as

$$q(x_t|y_{1:t}) = q(x_t|x_{t-1}, y_t)q(x_{t-1}|y_{1:t-1}) \tag{2.5}$$

Eq.s 2.2-2.3 above lead us to a simplified update for the weights given by,

$$\begin{aligned} w_t^i &\propto \frac{p(x_t^i|y_{1:t})}{q(x_t^i|y_{1:t})} \\ &\propto \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)p(x_{t-1}^i|y_{1:t-1})}{q(x_t^i|x_{t-1}^i, y_t)q(x_{t-1}^i|y_{1:t-1})} \\ &\propto w_{t-1}^i \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, y_t)} \end{aligned} \tag{2.6}$$

The above set of Eq.s 2.2-2.6 set up the most generic particle filtering algorithm, which we shall look into next. However, a major concern with the particle approximation and the

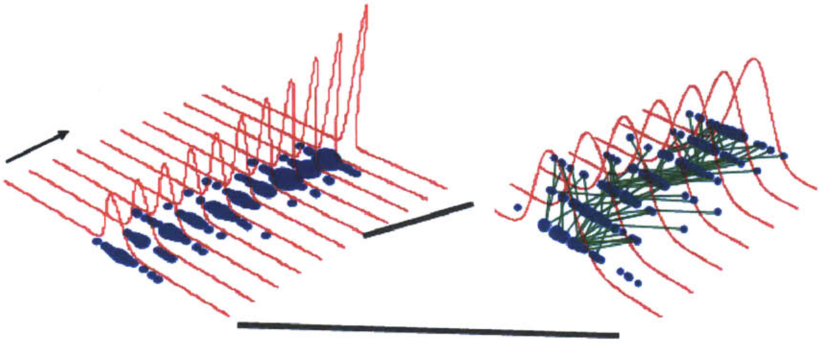


Figure 2-3: Qualitative effect of resampling on particle degeneracy [5]

update equation given by 2.6 is that over time the variance of the weights can only increase. This leads to a serious problem of particle degeneracy wherein only a handful of particle carry all the weights.

2.1.2 Degeneracy and Resampling

One of the main problems associated with particle filtering is degeneracy. As the filter moves in time the posterior distribution of the particles becomes skewed in certain directions, which is clearly seen in the left diagram of fig. 2-3. The reason behind this is that the particles with higher weights continue to get even higher weights in the next time step and as the result the distribution collapses. The phenomenon of all the weight being carried by a few particles and rest of the particles having negligible weights is called sample degeneracy. Apart from sample degeneracy an immense amount of computational power by updating weights for particles with close to zero weight. There have been several solutions proposed to this problem, two of them being Resampling and Good Choice of Importance Density. However, it is argued that resampling does not reduce the degeneracy, it merely reduces the computational effort of dealing with low weight particles.

We define effective particle size (ESS) as $\hat{N_{eff}} = \frac{1}{\sum_{i=1}^{N}(w_t^i)^2}$.Though this formula is not exact it gives a good approximation of the effective sample size. It is easy to see that in the extreme cases i.e. when one particle has all the weight, the ESS is reduced to 1 and when all particles have equal weights $(1/N)$ the ESS is N. Resampling is done when the ESS falls below a certain threshold (N_T). The basic idea of resampling is to eliminate all particles with low weights and to concentrate on particles with high weights. In the process, a large number of particles are generated in the high probability zone, each having equal weight, and thereby helping in throwing around the particles in the high probability zone. One can use the Multinomial Resampling Algorithm as detailed below.

Multinomial Resampling Algorithm

```
$[\{x_t^k, w_t^k\}_{k=1}^N]$=RESAMPLE$[\{x_t^i, w_t^i\}_{i=1}^N]$
Initialize the CDF: $c_1 = 0$
FOR $i = 1, ..., N$
    Construct CDF: $c_i = c_{i-1} + w_t^i$
END FOR
Start at the bottom of the CDF: $i = 1$
Draw a starting point: $u_1 \sim$ U$(0, N^{-1})$
FOR $k = 1, ..., N$
    Move along the CDF: $u_k = u_1 + N^{-1}(k-1)$
    WHILE $u_k > c_i$
            $i = i + 1$
    END WHILE
    Assign sample: $x_t^k = x_t^i$
    Assign weight: $w_t^k = w_t^i$
END FOR
```

The other way of decreasing sample degeneracy is by choosing a good importance density. This is definitely not an easy task. One can evaluate the optimal importance density according to a particular criteria, however, sampling from such an optimal density is the difficult part. For example, Doucet [25] showed that the importance density to minimize the variance of the weights and maximize the ESS is given by $q(x_t|x^i_{t-1}, y_t) = p(x_t|x^i_{t-1}, y_t)$. However, it is very difficult to sample from this distribution except in cases when $p(x_t|x^i_{t-1}, y_t)$ is gaussian.

2.1.3 Filtering Algorithms

Now, we have all the background to independently look at different filtering algorithms and evaluate their respective performances using the Lorenz model. The most generic particle filter algorithm is detailed below. As we shall see we can modify the generic particle filter using a particular set of assumptions leading us to different types of particle filtering algorithms.

Generic Particle Filter

```
$[\{x^i_t, w^i_t\}^N_{i=1}]$=PF$[\{x^i_{t-1}, w^i_{t-1}\}^N_{i=1}, y_t\}]$
FOR $i = 1, ..., N$
    Draw $x^i_t \sim q(x_t|x^i_{t-1}, y_i)$
    Assign the particle a weight, $w^i_t = w^i_{t-1}\frac{p(y_t|x^i_t)p(x^i_t|x^i_{t-1})}{q(x^i_t|x^i_{t-1},y_k)}$
END FOR
Normalize the weights $\{w^i_t\}^N_{i=1}$
Calculate $\hat{N_{eff}} = \frac{1}{\sum^N_{i=1}(w^i_t)^2}$
IF $\hat{N_{eff}} < N_T$
  Resample using the multinomial resampling algorithm: $[\{x^i_t, w^i_t\}^N_{k=1}]$=RESAMPLE$[\{x^i_t, w^i_t\}^N_{i=1}]$
END IF
```

Stochastic Lorenz Model

Before we jump into specific particle filtering algorithms we describe the Lorenz model which we use to compare the performance of various algorithms later. The stochastic Lorenz equations are a a set of 3-dimensional equations describing a chaotic system. The deterministic Lorenz model is given by the following equations.

$$\begin{aligned}
\dot{x} &= \sigma(y - x) \\
\dot{y} &= \rho x - y - xz \\
\dot{z} &= xy - \beta
\end{aligned}$$

We obtain the stochastic set of Lorenz equations by adding independent noise terms in each of the three equations above. Further, we discretize the stochastic system numerically using the Euler-Maruyama scheme to obtain the following set of equations on which we apply our particle filtering algorithms.

$$\begin{aligned}
x_{t+1} &= x_t + \sigma(y - x)dt + e.\mathrm{N}(0, dt) \\
y_{t+1} &= y_t + (\rho x - y - xz)dt + e.\mathrm{N}(0, dt) \\
z_{t+1} &= z_t + (xy - \beta)dt + e.\mathrm{N}(0, dt)
\end{aligned}$$

In literature, one uses the start values for the variables $(x, y, z) = (5.91652, 5.52332, 24.5723)$ as it gives rise to a chaotic Lorenz system. Also, the values of the parameters are given by $(\sigma, \rho, \beta, e) = (10, 28, 8/3, 0.5)$. Although the value of e can be altered to change the process noise or our "degree of belief" in the model, we keep it fixed for the purpose of this study.

Specific Filtering Algorithms

One of the most common filtering algorithms is the Sequential Importance Sampling (SIS) Algorithm. The assumptions behind the algorithm are very simple. The choice of importance density is the prior density, i.e. $q(x_t|x_{t-1}^i, y_{1:t}) = p(x_t|x_{t-1}^i)$ and resampling is done in a time step if the ESS falls below a threshold sample size. The choice of prior density as the importance density gives a simple update equation for the weights $w_t^i \propto w_{t-1}^i p(y_t|x_t^i)$, only involving the likelihood of the new observation. The Sequential Importance Resampling (SIR) algorithm is only a little different from the SIS algorithm. In SIR resampling is done at every time step unlike in SIS. The SIS is algorithm is formally described below.

SIS Particle Filter

```
$[\{x_t^i, w_t^i\}_{i=1}^N]$=SIR$[\{x_t^i, w_t^i\}_{i=1}^N, y_k\}]$
FOR $i = 1, ..., N$
    Draw $x_t^i \sim p(x_t|x_{t-1}^i)$
    Assign the particle a weight, $w_t^i = w_{t-1}^i p(y_t|x_t^i)$
END FOR
Normalize the weights $\{w_t^i\}_{i=1}^N$
Calculate $\hat{N_{eff}} = \frac{1}{\sum_{i=1}^N (w_t^i)^2}$
IF $\hat{N_{eff}} < N_T$
  Resample using the multinomial resampling algorithm: $[\{x_t^i, w_t^i\}_{k=1}^N]$=RESAMPLE$[\{x_t^i, w_t^i\}_{i=1}^N]$
END IF
```

Theoretically, the runtime of the SIS particle filter is $\mathcal{O}(NT)$, where N is the number of particles used and T is the total number of time steps used in the numerical integration scheme. The constant factor depends on several other parameters associated with algorithm, like observation frequency, resampling threshold etc. We now study the performance of this algorithm using the Lorenz model described above. We use a linear observation model as follows:

$$x_{obs} = x_{true} + \sigma_{obs}.\mathrm{N}(0,1)$$
$$y_{obs} = y_{true} + \sigma_{obs}.\mathrm{N}(0,1)$$
$$z_{obs} = z_{true} + \sigma_{obs}.\mathrm{N}(0,1)$$

First, we run the SIS filter, with an observation noise of $\sigma_{obs} = 1$, start values of states same as their true values, observations made every 0.2s and a sample size (number of particles) equal to 500. The results are shown in fig.s 2-4 and 2-5. We plotted the mean estimates of the states and their respective quantiles along with the true values and the noisy observations. The true values of the states here correspond to those which were used to generate the noisy observations. We observe that in all the three dimensions that the inferred estimates of the states are very close to their true values and also that the true value lies within the quantile estimates of the states. After looking at a particular case we now turn our attention to the effect of various parameters associated with the algorithm and look at the their effect on the performance of the algorithm. Specifically, we vary each of observation noise, observation frequency, sample size and start values (priors) and study their respective effect on the performance of the algorithm.

To compare the results from various algorithms, we need to account for the uncertainty in the system. We need to evaluate the filtered estimates over several realizations. A good estimate of the filtered states being the mean over the samples. In literature, Root Mean Squared Error (RMSE) has been extensively used to compare filtered state estimates. The"mean" in RMSE refers to the mean taken over several realizations of the filter. In other words, the filter is run several times to obtain estimates of the states (and parameters) to account for the uncertainty. More formally, we can write,

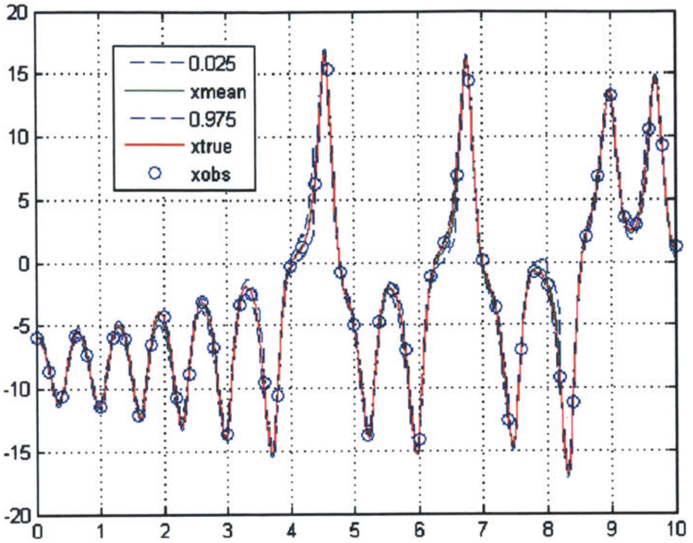
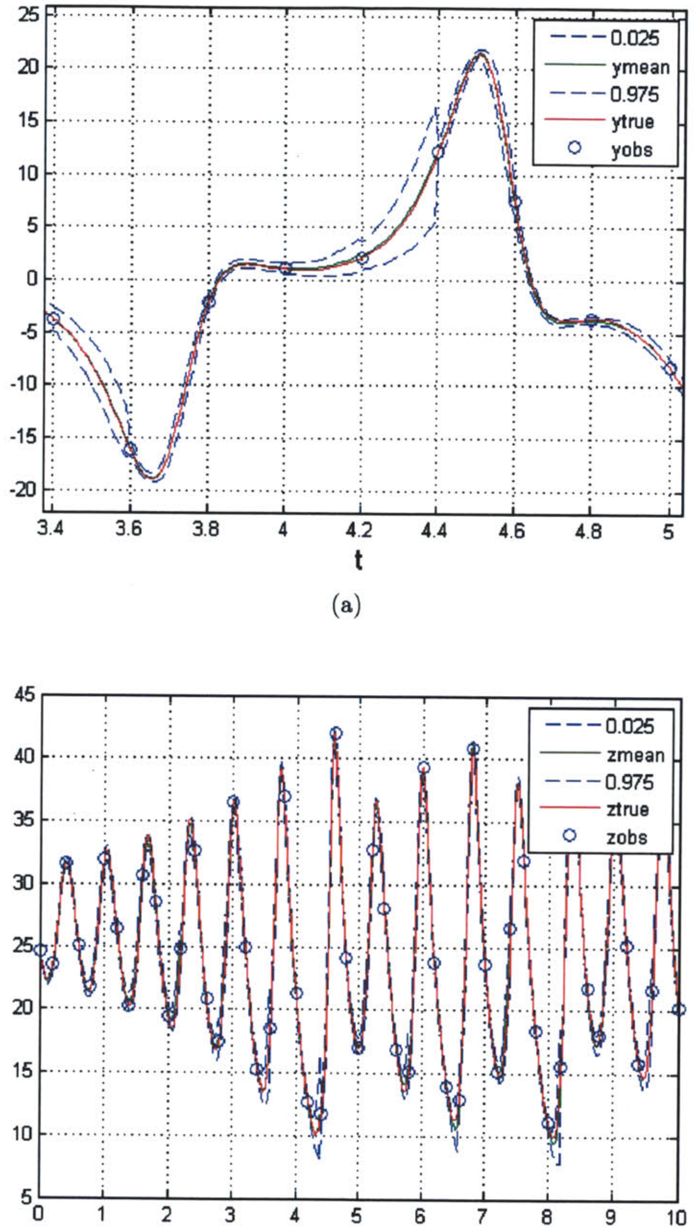


Figure 2-4: Estimates of state x compared to its true values

$$Root\ mean\ square\ error\ (RMSE)\ =\ \sqrt{\frac{1}{M}[\sum_{m=1}^{M}(\hat{x}_m - x_{true})^2 + Var(x)_m]}$$

where $\hat{x}_m$ and $Var(x)_m$ are the state estimate (mean) and variance of the m_{th} realization, x_{true} is the true state and M is the total number realizations used. Typically, around 10-100 realizations are used in simple sequential estimation problems.

First, we study the effect of the number of particles we use in filtering on the estimated states. Figure 2-6 shows the effect of the sample size on the RMSE of x. One can clearly infer that the performance of the algorithm improves with the number of particles used to carry out the filtering. For the case shown in fig. 2-6 we used an observation noise with $\sigma_{obs} = 1$, observation time (the time interval between successive observations) $ot = 0.2s$, threshold sample size $N_T = 0.2N$ and start values same as true values of states. We use a sample size of $N = 500$ for our remaining studies.



(a)

(b)

Figure 2-5: Estimates of state y and z compared to their true values

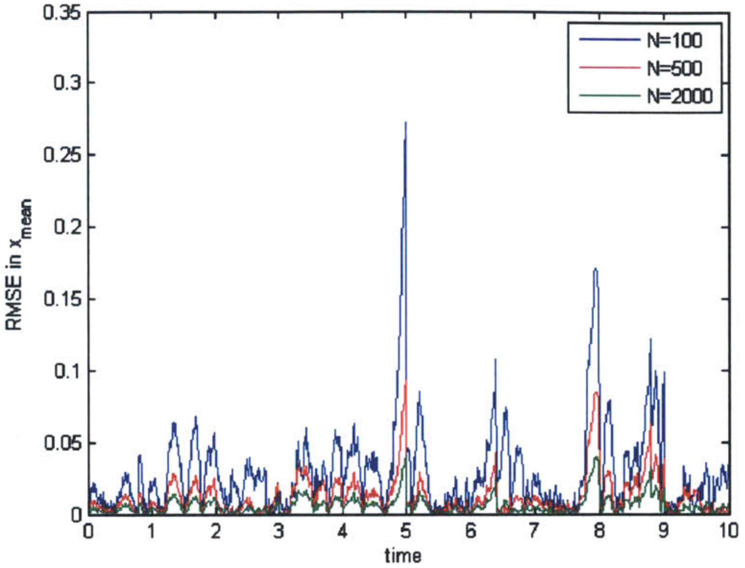
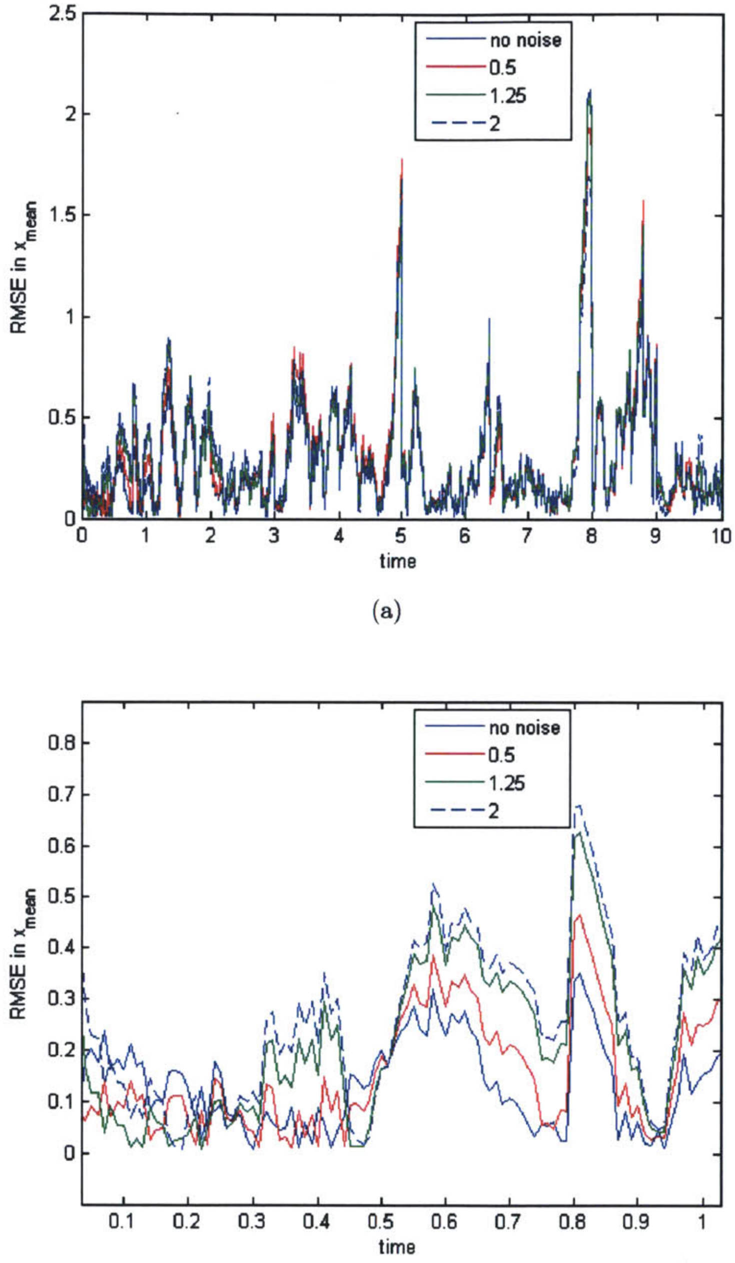


Figure 2-6: RMSE of x for different sample sizes (number of particles)

The start values or priors of the states can also determine the performance of the filtering algorithm. Figure 2-7 shows the RMSE of x for different normal priors on x with $\sigma_x = 0, 0.5, 1.25, 2$ respectively. In fig.2-7a we see that over the long run all priors do not really matter. It is only in the first few time iterations, as evident from fig. 2-7b that the priors on x effects the performance. This can be attributed to the fact that with increasing time, more observations come in and hence the posterior distributions of the states rely more on the likelihood and the influence of the initial prior decays.

The threshold sample size which decides whether to resample or not is a difficult parameter to set. In other words, it is difficult to say whether or not resampling is good at any time step. Figure 2-8 shows the results of the RMSE with time for filtered state x for various values of the threshold sample size used. Figure 2-8a tells us that the no resampling case is easily the worst among the lot. However, as we can see from fig. 2-8b there is no single threshold sample size which does the best at all times. It appears that this is a difficult parameter to set and depends on the system model as well.



(a)

(b)

Figure 2-7: RMSE of x for different normal priors on x with $\sigma_x = 0, 0.5, 1.25, 2$

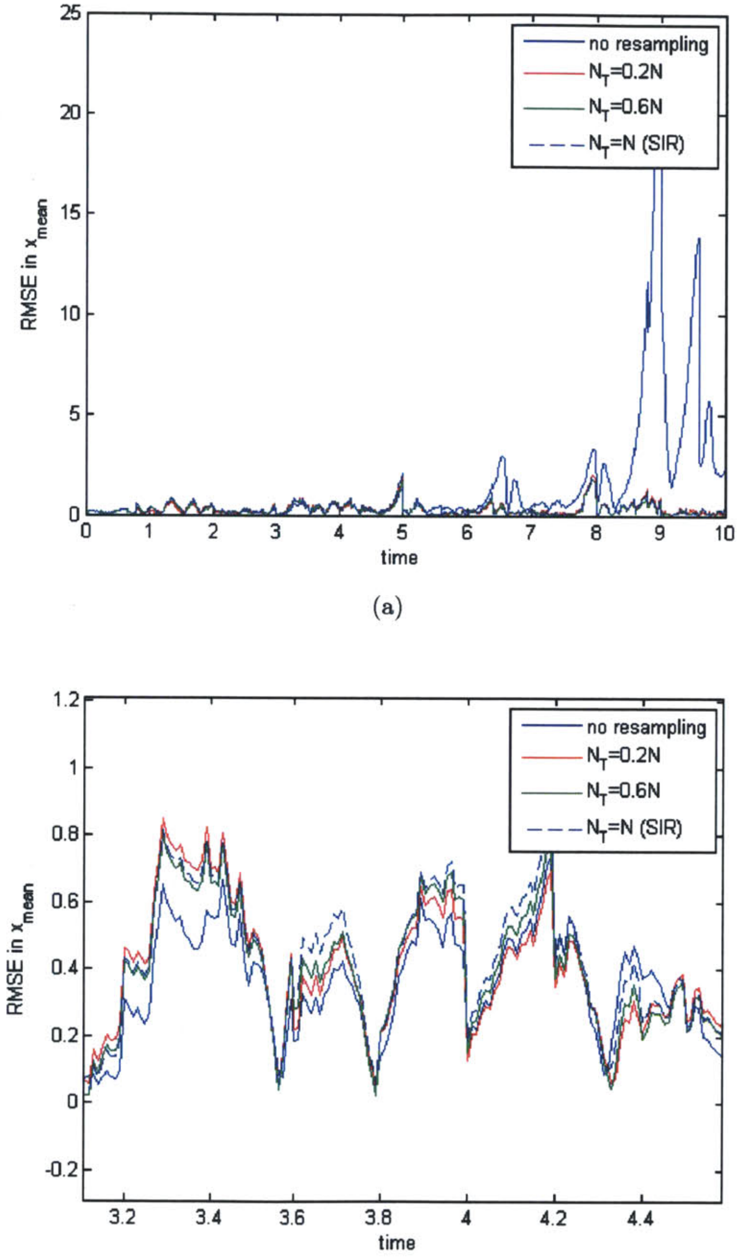
Next,we look at the effect of observation noise on the RMSE of x. The four cases considered are with a gaussian noises with $\sigma_{obs} = 0.1, 1, 2, 3$ respectively. We observe from fig. 2-9 that the performance goes down with increasing noise. This is further confirmed when we evaluate the integral mean of the RMSE values over time as $0.0923, 0.2213, 0.4564, 0.6941$ with increasing noise. We recommend an observation noise with $\sigma_{obs} = 1$, which is reasonable going by usual measurement standards.

Finally, we vary the observation frequency and look at its effect on the RMSE. We define the observation time (ot) as the time interval between two successive observations. Figure 2-10 shows that the the case of $ot = 0.5s$ is clearly the worst. We evaluate the integral mean RMSE as 0.0724, 0.1892 and 0.4321 for $ot = 0.1s$,$0.2s$ and $0.5s$ respectively. Hence, as expected the performance gets better with increasing observation frequency of more incoming data per unit time.

The other popularly used filtering algorithm is the auxiliary particle filter. The auxiliary particle filter (APF) was introduced by Pitt and Shephard [70] as a variant of the standard SIR filter. This filter can be derived from the SIR filter by introducing an importance density $q(x_t, i|y_{1:t})$, which samples from the pair $\{x_t^k, k\}_{k=1}^N$ where k refers to the index of the particle at $t-1$. By applying Bayes' rule, a proportionality can be derived for $p(x_t, i|y_{1:t})$ as

$$\begin{aligned} p(x_t, i|y_{1:t}) &\propto p(y_t|x_t)p(x_t, i|y_{1:t-1}) \\ &= p(y_t|x_t)p(x_t|i, y_{1:t-1})p(i|y_{1:t-1}) \\ &= p(y_t|x_t)p(x_t|x_{t-1}^i)w_{t-1}^i \end{aligned} \tag{2.7}$$

The APF operates by obtaining a sample from the joint density $p(x_t, i|y_{1:t})$ and then omitting the indices i in the pair (x_t, i) to produce a sample $\{x_t^k\}_{k=1}^N$ from the marginalized



(a)

(b)

Figure 2-8: RMSE of x for different threshold sample sizes

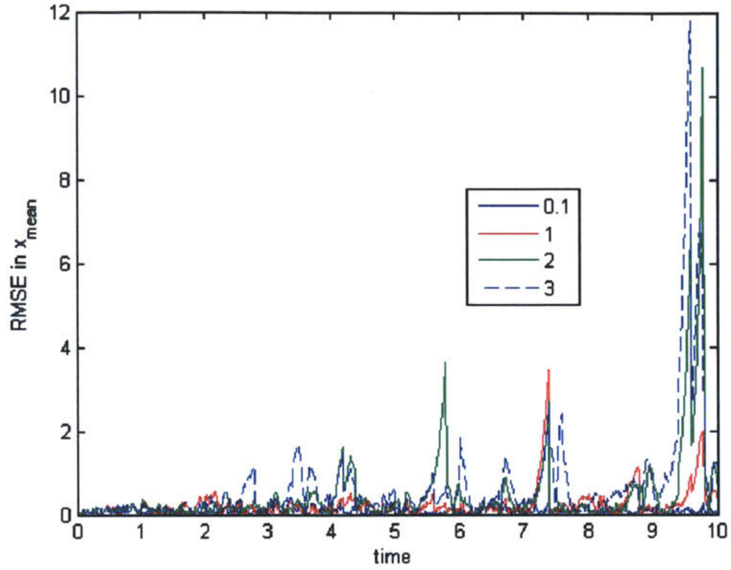


Figure 2-9: RMSE of x for different observation noises

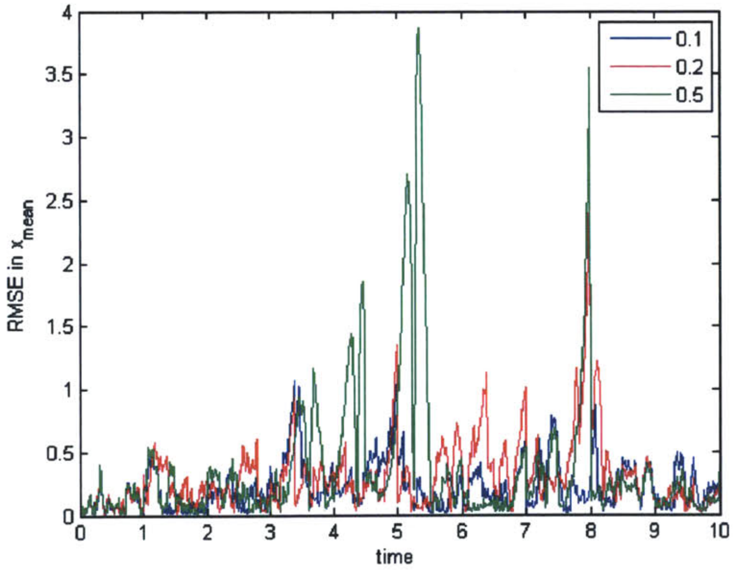


Figure 2-10: RMSE of x for different observation times

density $p(x_t|y_{1:t})$. The importance density used to draw the sample $\{x_t^k, k\}_{k=1}^N$ is defined to satisfy the proportionality

$$q(x_t, i|y_{1:t}) \propto p(y_t|\mu_t^i)p(x_t|x_{t-1}^i)w_{t-1}^i \tag{2.8}$$

where μ_t^i is some characterization of x_t, given x_{t-1}^i. This could be the mean, in which case, $\mu_t^i = \mathrm{E}(x_t|x_{t-1}^i)$. We can write

$$q(x_t, i|y_{1:t}) = q(i|y_{1:t})q(x_t|i, y_{1:t})$$

and defining $q(x_t|i, y_{1:t}) = p(x_t|x_{t-1}^i)$ it follows from Eq. 2.8 that

$$q(i, y_{1:t}) \propto p(y_t|\mu_t^i)w_{t-1}^i$$

The sample $\{x_t^k, k\}_{k=1}^N$ is then assigned a weight proportional to the ratio of the right hand side of Eq. 2.7 to Eq. 2.8.

$$w_t^k \propto w_{t-1}^k \frac{p(y_t|x_t^k)p(x_t^k|x_{t-1}^k)}{q(x_t^k, k|y_{1:t})} = \frac{p(y_t|x_t^k)}{p(y_t|\mu_t^k)}$$

Finally, we describe the algorithm below.

Auxiliary Particle Filter

```
$\{x_t^i, w_t^i\}_{i=1}^N$]=APF[$\{x_{t-1}^i, w_{t-1}^i\}_{i=1}^N, y_t\}$]
FOR $i = 1, ..., N$
    Calculate $\mu_t^i \sim \mathrm{E}(x_t|x_{t-1}^i)$
    Calculate $w_t^i = w_{t-1}^i p(y_t|\mu_t^i)$
END FOR
Resample using the multinomial resampling algorithm.
FOR $i = 1, ..., N$
    Draw $x_t^i \sim p(x_t|x_{t-1}^i)$
```

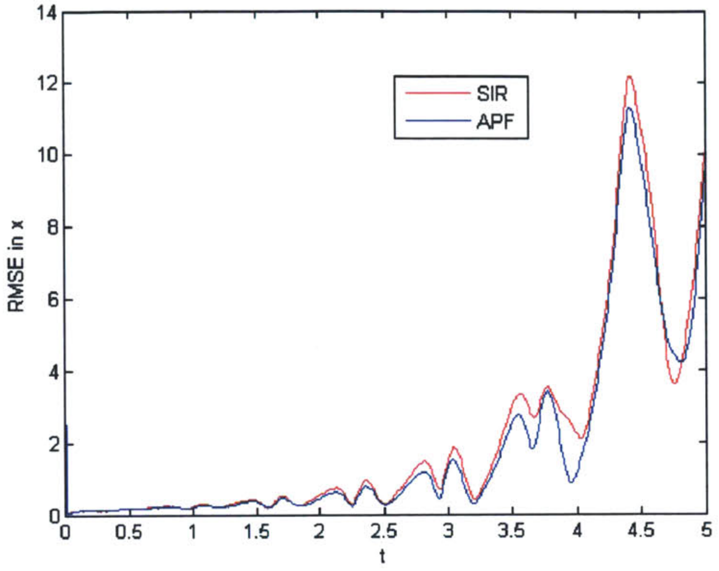


Figure 2-11: RMSE of x using the APF and SIR

```
    Assign the particle a weight, $w_t^i = \frac{p(y_t|x_t^i)}{p(y_t|\mu_t^i)}$
END FOR
Normalize the weights $\{w_t^i\}_{i=1}^N$
```

The auxiliary particle filter has an advantage over the SIR because it resamples the previous states according to the likelihood of an estimate of the present states. This ensures that the states generated have higher likelihoods. The advantage is clearly seen if the overlap of the likelihood and the prior is not good. The APF ensures that the previous states are sampled such that they give rise to states which have higher likelihood in terms of the new data available. Figure 2-11 compares the performance of the APF and the SIR for $\sigma_{obs} = 1$, $ot = 0.01s$ and $N = 100$ particles. It appears that both the algorithms have almost similar accuracy, although the APF is more computationally expensive than the SIR.

2.2 Particle Smoothing

A related idea to particle filtering is particle smoothing. Filtering corresponds to estimating the distribution of the current state of an HMM based upon the observations received up until the current time, whereas, smoothing corresponds to estimating the distribution of the state at a particular time given all of the observations up to some later time T. The trajectory of the estimates obtained by smoothing tend to be smoother than the one obtained through filtering. It is kind of intuitive to say that smoothing gives a better approximation of the hidden states of the HMM because it uses more data than filtering. The additional information incorporated in smoothing in the form of some extra information at later times makes it better. Hence there is a clear need to come up with smoothing algorithms which might be able to give better estimates of the hidden states.

Formally, assuming that we have access to data up to time T, we need to compute the marginal distributions $\{p(x_t|y_{1:T})\}_{t=1}^{T}$ or to sample from the joint distribution $p(x_{1:T}|y_{1:T})$. In principle the marginals could be obtained from this joint distribution by integrating out the additional variables $(x_{1:t-1}, x_{t+1:T})$. However, Doucet and Johansen [25] note that using particle methods to obtain the joint distribution and then integrating out the extra variable does not work, especially in cases when $t << T$. Instead the marginal filtering distribution $p(x_t|y_{1:T})$ plays a crucial role and needs to be used in this framework, as it is better characterized as a marginal distribution. Hence, to obtain smoothing algorithms we need to build recursions which rely on the marginal filtering distributions $p(x_t|y_{1:t})$.

2.2.1 Forward Filtering Backward Smoothing (FFBS)

The joint distribution $p(x_{1:T}|y_{1:T})$ can be factored using the fact that $\{X_t\}$ is a Markov process, conditional on $y_{1:T}$ as follows:

$$\begin{aligned} p(x_{1:T}|y_{1:T}) &= p(x_T|y_{1:T}) \prod_{t=1}^{T-1} p(x_t|x_{t+1}, y_{1:T}) \\ &= p(x_T|y_{1:T}) \prod_{t=1}^{T-1} p(x_t|x_{t+1}, y_{1:t}) \end{aligned} \tag{2.9}$$

Also, using Bayes' rule we can write,

$$p(x_t|x_{t+1}, y_{1:t}) = \frac{f(x_{t+1}|x_t)p(x_t|y_{1:t})}{p(x_{t+1}|y_{1:t})} \tag{2.10}$$

Eq.s 2.9 and 2.10 form the basis of the forward-backward recursion. First, we compute and store the marginal distributions $\{p(x_t|y_{1:t})\}_{t=1}^{T}$. Then sample $X_t \sim p(x_T|y_{1:T})$ and for $t = T-1, T-2, ..., 1$, sample $X_t \sim p(x_t|x_{t+1}, y_{1:t})$ which leads us to samples from from the joint distribution $p(x_{1:T}|y_{1:T})$, as suggested by Eq. 2.9. However, instead of having to sample from $p(x_t|x_{t+1}, y_{1:t})$, we integrate out $(x_{1:t-1}, x_{t+1:T})$ in Eq.2.9 and use Eq. 2.10 to obtain,

$$p(x_t|y_{1:T}) = p(x_t|y_{1:t}) \int \frac{f(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})} p(x_{t+1}|y_{1:T}) \mathrm{d}x_{t+1} \tag{2.11}$$

So, to compute $\{p(x_t|y_{1:T})\}_{t=1}^{T}$, we simply modify the backward pass and use Eq. 2.11. To obtain the SMC estimates of the smoothed marginal $p(x_t|y_{1:T})$, we use the following particle approximations,

$$p(x_t|y_{1:t}) = \sum_{i=1}^{N} w_t^i \delta(x_t - x_t^i) \tag{2.12}$$

$$p(x_t|y_{1:T}) = \sum_{i=1}^{N} w_{t|T}^i \delta(x_t - x_t^i) \tag{2.13}$$

We plug the above approximations from Eq.s 2.12 and 2.13 into Eq. 2.11 to obtain the final FFBS recursion using particle/SMC methods.

$$w^i_{t|T} = w^i_t \sum_{j=1}^{N} w^j_{t+1|T} \frac{f(x^j_{t+1}|x^i_t)}{\sum_{l=1}^{N} w^l_t f(x^j_{t+1}|x^l_t)} \tag{2.14}$$

We can infer from Eq. 2.14 that the recursion involves the marginal filtered distribution weights and the transition probability density of the process. It might be useful to note that in some cases the transition density might be difficult to obtain analytically. In such cases, we might have to some approximation for the transition density, or not be able to use the FFBS at all. Once, we have the marginal filtered distributions as particle approximations and the transition density between the hidden states, the formula is recursion is straight forward to apply. To develop an intuition for the recursion, we observe from the formula that the weights are updated using taking into considerations the following:

1) Marginal filtered weights of the present state
2) Sum of chances of making a transition to all "good" particles in the next state, where a "good" particle is one which has a large smoothed distribution weight.

The order of complexity of the FFBS is $\mathcal{O}(N^2T)$. This is clearly seen from Eq. 2.14 as we need to compute the transition probability between every pair particles in the consecutive time steps. Hence this method is computationally expensive and is used when sample degeneracy prevents usage of computationally cheaper methods.

2.2.2 Generalized Two-Filter Formula

The two filter formula is another recursion method to compute the smoothed marginals. It further attempts to decrease the computational complexity using certain approximations.

The two-filter formula is based on the following identity

$$p(x_t|y_{1:T}) = \frac{p(x_t|y_{1:t-1})p(y_{t:T}|x_t)}{p(y_{t:T}|y_{1:t-1})} \tag{2.15}$$

where the backward filter is initialized at $t = T$ by $p(y_t|x_t) = g(y_t|x_t)$ and satisfies the following recursion,

$$\begin{aligned} p(y_{t:T}|x_t) &= \int \prod_{k=t+1}^{T} f(x_k|x_{k-1}) \prod_{k=t}^{T} g(y_k|x_k) \mathrm{d}x_{t+1:T} \\ &= g(y_t|x_t) \int_f (x_{t+1}|x_t) p(y_{t+1:T}|x_{t+1}) \mathrm{d}x_{t+1} \end{aligned} \tag{2.16}$$

The backward information filter is not a probability density argument in x_t and it is even possible that $\int p(y_{t:T}|x_t)\mathrm{d}x_t = \infty$. Although when one can analytically find $p(y_{t:T}|x_t)$ particle approximation may not be a problem. To develop particle approximations of the backward information filter, we introduce a set of artificial distributions $\{\hat{p}_t(x_t)\}$ and the joint distributions

$$\hat{p}_t(x_{t:T}|y_{t:T}) \propto \hat{p}_t(x_t) \prod_{k=t+1}^{T} f(x_k|x_{k-1}) \prod_{k=t}^{T} g(y_k|x_k)$$

which are constructed such that their marginal distributions, $\hat{p}_t(x_t|y_{t:T}) \propto \hat{p}_t(x_t)p(y_{t:T}|x_t)$, are integrable version of the backward information filter. The generalized two-filter formula hence becomes

$$p(x_t|y_{1:T}) = \frac{p(x_t|y_{1:t-1})\hat{p}_t(x_t|y_{t:T})}{\hat{p}_t(x_t)} \tag{2.17}$$

whichever is valid whenever the support of $\hat{p}_t(x_t)$ includes support of the prior $p_t(x_t)$. The generalized two filter smoother proceeds as follows. Using the standard filtering methods, we can compute the marginal distributions $\{p(x_t|y_{1:t-1})\}_{t=1}^{T}$. Using a backward recursion in Eq. 2.16, we compute and store $\{\hat{p}_t(x_t|y_{t:T})\}_{t=1}^{T}$. Then we combine $p(x_t|y_{1:t-1})$ and $\hat{p}_t(x_t|y_{t:T})$ to obtain $p(x_t|y_{1:T})$ using Eq. 2.15. One main problem with this method is choosing appropriate $\hat{p}_t(x_t)$. We need to be able to compute it directly, hence having $\hat{p}_t(x_t) = p(x_t)$ is ruled out for most non-linear non-gaussian models. In literature it is suggested that we use a heavy tailed approximation of $p(x_t)$ for $\hat{p}_t(x_t)$ instead in such settings. A good overview of the two-filter smoothing can be found in [12]. Also, they mention methods using which the complexity of the smoothing algorithm can be improved to $\mathcal{O}(NT)$.

2.2.3 Practical Filter - Fixed Lag smoothing

The Practical Filter by Polson et al [73] is actually a method for filtering, since it provides samples from the marginal distribution $p(x_t|y_{1:t})$. However, the procedure they adopt to estimate the posterior of the states is based on the concept of fixed lag-smoothing. The lag-smoothing distribution is defined as the conditional distribution $p(x_{t-k+1:t}|x_{t-k}, y_{t-k+1:t})$ and the marginal lag-smoothing is defined as $p(x_{t-k}|y_{1:t})$, where k is the lag-length. One can sample from the lag-smoothing distributions using standard MCMC methods or the FFBS in case of conditionally linear models.

The state filtering distribution is given by $p(x_t|y_{1:t})$. As we observe, a new observation y_t we wish to update the filtering distribution $p(x_{t-1}|y_{1:t-1})$ to $p(x_t|y_{1:t})$. We start by writing the desired filtering distribution as the marginal

$$p(x_t|y_{1:t}) = \int p(x_{t-k+1}|y_{1:t})\mathrm{d}x_{t-k+1:t-1}$$

At the core of the algorithm is th representation of this lag smoothing distribution as a mixture with respect to the smoothing distribution on x_{t-1}.

$$\begin{aligned} p(x_{t-k+1}|y_{1:t}) &= \int p(x_{t-k+1:t}|x_{t-k}, y_{1:t})\mathrm{d}p(x_{t-k}|y_{1:t}) \\ &= \int p(x_{t-k+1:t}|x_{t-k}, y_{t-k+1:t})\mathrm{d}p(x_{t-k}|y_{1:t}) \end{aligned} \tag{2.18}$$

Now, we argue that the as the we are implementing the algorithm sequentially that samples from $p(x_{t-k}|y_{1:t})$ are approximately samples from $p(x_{t-k}|y_{1:t-1})$. The approximation is reasonable if the predictive distribution $p(y_t|y_{t-k+1:t-1}, x_{t-k})$ is conditionally independent of x_{t-k} as reflected in the identity below

$$p(x_{t-k}|y_{1:t}) = \frac{p(y_t|x_{t-k}, y_{t-k+1:t-1})}{p(y_t|y_{t-k+1:t-1})} p(x_{t-k}|y_{1:t-1})$$

Hence we can further simplify Eq. 2.18 as follows

$$p(x_{t-k+1}|y_{1:t}) = \int p(x_{t-k+1:t}|x_{t-k}, y_{t-k+1:t}) \mathrm{d}p(x_{t-k}|y_{1:t-1}) \tag{2.19}$$

The above identity provides the motivation of the simulation based Practical Filter. The algorithm is initialized with samples generated from the smoothing distribution $p(x_{1:k}|y_{1:k})$. Starting with $t = k + 1$ samples from the desired smoothing distributions $p(x_t|y_{1:t})$ are generated by following Eq. 2.19. At each time t we generate new samples $x_{t-k+1:t} \sim p(x_{t-k+1}|x_{t-k}, y_{t-k+1:t})$, using the imputed values for x_{t-k} saved from the previous iteration $t - 1$. We save the simulated x_{t-k+1} for used in the next iteration with $t + 1$. The algorithm reduces the filtering problem to a lag smoothing problem. We formally state the algorithm as follows.

```
Fixed-lag smoothing Algorithm:
Burn-In: FOR $t = 1, ..., k$:
    FOR $i = 1, ..., M$:
        Generate $x_{1:t} \sim p(x_{1:t}|y_{1:t})$ and save $\hat{x}_1^i = x_1$.
    END FOR
END FOR
Sequential Updating: FOR $t = k + 1, ..., T$:
    FOR $i = 1, ..., M$:
        Generate $x_{t-k+1:t} \sim p(x_{t-k+1:t}|\hat{x}_{t-k}^i, y_{t-k+1:t})$
        Set $\hat{x}_{t-k+1}^i = x_{t-k+1}$ and leave $\hat{x}_{t-k}^i$ unchanged.
    END FOR
END FOR
```

The algorithm replaces the filtering problem with $T - k$ smoothing problems for a time series of length k, which is typically easy to carry out. However, the computational complexity associated with this method is large when compared to usual filtering and even smoothing algorithms and goes as $\mathcal{O}(N^2kT)$, when the FFBS is used to sample from the lag-smoothing distributions. Also, before starting the problem, a preliminary study needs to be done to choose the appropriate lag-length k for the problem. Towards this, one can perform MCMC smoothing or FFBS to obtain the distributions $p(x_{1:t-1}|y_{1:t-1})$ and $p(x_{1:t}|y_{1:t})$ and compare the marginal distributions $p(x_{t-k}|y_{1:t-1})$ and $p(x_{t-k}|y_{1:t})$ for different values of k.

2.3 Parameter Learning using Particle Filtering Ideas

We now turn to the more involved problem of estimating the unknown system parameters and the states using particle filtering ideas. We assume, in this section that the both the state transition density and the conditional likelihood function depend not only upon the dynamic state x_t, but also on a parameter θ. In other words, the state transition density and the likelihood function can be represented as $f(x_t|x_{t-1}, \theta)$ and $g(y_t|x_t, \theta)$ respectively.

There are two kinds of parameter estimation in dynamic problems based on the application. If all the data $(y_{1:t})$ is available before the inference is to be carried out, the process of parameter estimation is called off-line estimation. In some cases, the parameters have to be estimated sequentially with continuously incoming data, which we call as online estimation. In this section, we concentrate on the online parameter estimation problem.

The online parameter estimation problem requires the joint estimation of the state and the parameter to be carried out at every stage of the process. The problem of joint estimation is to find out the joint probability distribution of the unknown parameters and states,

$p(x_t, \theta|y_{1:t})$, which usually has no easy analytical form. The most natural option consists of treating the unknown parameter θ as a component of the state with no dynamic evolution. Hence, we can directly use particle filtering methods to simulate the joint posterior distribution of the states and the parameters $p(x_t, \theta|y_{1:t})$. However, this method fails miserably as the absence of dynamic evolution of the parameter implies that the parameter space is not explored at any instant, or that the particles are not moved around in space but only re-weighted again and again. A practical solution to this problem is to make the parameter a dynamically evolving quantity by adding a random noise to it. With this as the motivation there have been four algorithms which have successfully tackled the online parameter estimation problem, namely, Storvik's Algorithm [86], Liu and West [55], Practical Filter [73] and the Online Expectation Maximization (EM) algorithm [1, 15]. We discuss the first three of these algorithms here.

2.3.1 Storvik's algorithm

Storvik's approach [86] is based on marginalizing the static parameters at every stage of the filter, such that only the state vector is considered. Real-time applications of this algorithm are only possible when the distribution of unknown parameters based given both the observations and the hidden state vector depends on some low dimensional sufficient statistics. For most models, such statistics exist. When such statistics are not available ,the algorithm can still be applied, but the computational complexity increases with time. In particular, for Gaussian based system processes combined with general observation distribution, sufficient statistics involve Kalman type updates.

Suppose that for given $x_{1:t}$ and $y_{1:t}$, the distribution of θ is analytically tractable. In particular, the distribution of θ is assumed to depend on some sufficient statistics $S_t = S(x_{1:t}, y_{1:t})$ where S_t can be recursively updated. The approach is based on the following result:

$$\begin{aligned} p(x_{1:t},\theta|y_{1:t}) &= Cp(x_{1:t},\theta,y_t|y_{1:t-1}) \qquad \textit{where} \quad C^{-1}=p(y_t|y_{1:t-1}) \\ &= Cp(x_{1:t-1}|y_{1:t-1})p(\theta|x_{i:t-1},y_{1:t-1})p(x_t|x_{1:t-1},y_{1:t-1},\theta)p(y_t|x_{1:t},y_{1:t-1},\theta) \\ &= Cp(x_{1:t-1}|y_{1:t-1})p(\theta|S_{t-1})p(x_t|x_{t-1},\theta)p(y_t|x_t,\theta) \end{aligned} \tag{2.20}$$

Simulation can be performed as before from the marginal $p(x_{1:t}|y_{i:t})$ as before , but with an additional step that θ also needs to be simulated. The simplest approach is to simulate $x_{1:t-1}$ from $p(x_{1:t-1}|y_{1:t-1}$, θ from $p(\theta|S_{t-1})$, x_t from $p(x_t|x_{t-1},\theta)$ and accept with probability proportional to $p(y_t|x_t,\theta)$. However, any simulation technique such as MCMC, rejection sampling can be applied. The most widely used approach is sequential importance resampling.

Storvik's algorithm

```
For $t = 1, 2, ..., T$
Importance Sampling: For $i = 1, 2, ..., N$
1. Sample $\theta \sim f_{t,1}(\theta|x^i_{1:t-1}, y_{1:t})$; (sample from prior for t=1)
2. Sample $\hat{x}^i_t \sim f_{t,2}(x_t|x^i_{1:t-1}, y_t, \theta)$; and define $\hat{x}^i_{1:t} \equiv (\hat{x}^i_t, x^i_{1:t-1})$; (sample from prior for t=1)
3. Evaluate the importance weights
$\hat{w}^i_t = \hat{w}^i_{t-1} \frac{p(\theta|S^i_{t-1})p(\hat{x}^i_t|x^i_{t-1},\theta)p(y_t|\hat{x}^i_t,\theta)}{f_{t,1}(\theta|x^i_{1:t-1},y_{1:t})f_{t,2}(\hat{x}^i_t|x^i_{1:t-1},y_t,\theta)}$
Resampling: For $i = 1, 2, ..., N$
4. Sample an index $k$ from $1, 2, ..., N$ with probabilities proportional to $\hat{w}^i_t$;
5. Put $x^i_{1:t} = \hat{x}^k_{1:t}, S^i_t = S(S^k_{t-1}, x^i_t, y_t)$, and $w^i_t = N^{-1}$.
end
```

Here $f_{t,1}(\theta|x^i_{1:t-1}, y_{1:t})$ and $f_{t,2}(x_t|x^i_{1:t-1}, y_t, \theta)$ are the proposal distributions for θ and x_t respectively. Typically, we choose $f_{t,1}(\theta|x^i_{1:t-1}, y_{1:t}) = p(\theta|S^i_{t-1})$ and $f_{t,2}(x_t|x^i_{1:t-1}, y_t, \theta) = p(\hat{x}^i_t|x^i_{t-1}, \theta)$, so that the weight update equation becomes $\hat{w}^i_t = \hat{w}^i_{t-1}p(y_t|\hat{x}^i_t, \theta)$. It is necessary to marginalize the joint distribution $p(x_t, \theta|y_{1:t})$ at the end of every time step to obtain the particles $\{x_t\}_{i=1}^{N}$ for the next iteration. Also, to find out the posterior distribution of theta $p(\theta|y_{1:t})$ at any instant, we must marginalize out the states.

We applied the Storvik's algorithm on the Lorenz model described in earlier. Sequential parameter learning algorithms are quantified in terms of the root mean square error (RMSE) between the true states (or parameters) and the estimates of the filtered states (or parameters). A good esmitate of the filtered states (or parameters) being the mean over the samples. The"mean" in RMSE refers to the mean taken over several realizations of the filtered values. In other words, the filter is run several times to obtain estimates of the states (and parameters) to account for the uncertainty in the system equations. More formally, we can write,

$$Root\ mean\ square\ error\ (RMSE) \;=\; \sqrt{\frac{1}{M}\sum_{m=1}^{M}(\hat{x}_m - x_{true})^2}$$

where $\hat{x}_m$ is the state or parameter estimate of the m_{th} realization, x_{true} is the true state/parameter value and M is the total number realizations used. Typically, around 100 realizations are used in sequential estimation problems.

In our first study, we try to estimate the states and parameter σ in the Lorenz equation with true values for the other parameters. The Stochastic Lorenz equation is discretized using the Euler-Maruyama scheme with a numerical discretization time of $t = 0.01s$ with an observation made at every instant (observations made less frequently results in the algorithm performing poorly). We use a prior of $N(0, 100)$ on the parameter σ. The best results are obtained for realistic observation noises between 0.05 and 0.3. We use an observation noise of 0.1 for all our parameter estimation studies henceforth. The sample size (number of particles used) of the particle filter plays a crucial role in estimation of the posteriors. Figures 2-12 and 2-13 shows the effect of number of particles used on the RMSE of mean estimates of σ and state variable x respectively. As expected the RMSE in parameter σ decreases with increase in number of particles, showing that the algorithm improves in terms of estimation of parameter estimate with an increase in number of particles. In terms of the RMSE of state x, it appears that the algorithm does very for small number of particle as well. This can been seen as a strength of the algorithm.

In our next study, we look at the evolution of the mean estimate σ over several realizations. Figure 2-14 shows the evolution of the mean estimate σ over the first few iterations in time. We notice that the posterior converges very quickly and its mean of the estimate is close to the true value of the parameter. Figure 2-15 shows the distribution of the mean estimate $\hat{\sigma}_m$ over $M = 100$ realization at time $t = 5s$.

Finally, we try simulating the posteriors for the case when all three parameters appearing in the Lorenz equations are unknowns. We use priors than the earlier case, $\sigma \sim N(10, 10), \rho \sim N(28, 10), b \sim N(8/3, 10)$. The mean estimates of the parameters after $5s$ are $\sigma = 10.58, \rho = 28.52, b = 2.74$ which are very close to their actual values. Figure 2-14 show the estimates of the state variable and parameters for the case when all three parameters are unknown. We can visually infer that the filtered state estimates and the true states overlap with each other. SD denotes the standard deviation of the variable in the figures.

2.3.2 Liu and West Algorithm

In the general model with fixed parameters θ, extend the sample-based framework as follows. At each time t, we have combined samples

$$\{x_t^i, \theta_t^i \colon i = 1, 2, ..., N\}$$

and associated weights

$$\{w_t^i \colon i = 1, 2, ..., N\}$$

for an importance sampling/resampling approximation to the time posterior $p(x_t, \theta)$ for both parameter and state. Notice that the t suffix on the θ samples indicate that they are from the time t posterior, *not* that θ is time-varying. As time evolves to $t + 1$, we observe y_{t+1} and want to generate samples from

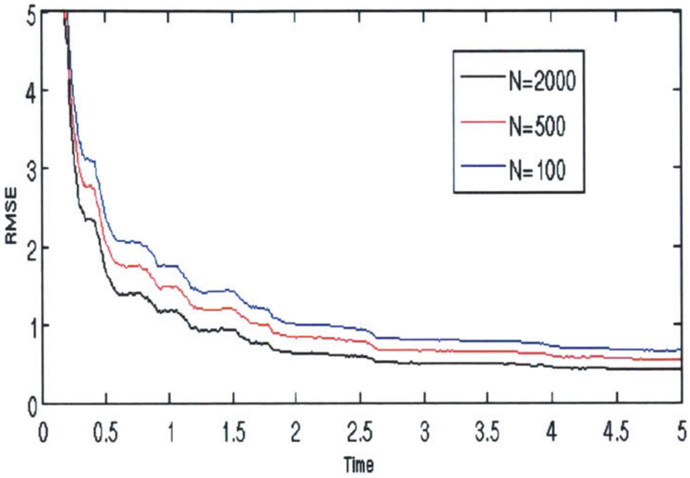


Figure 2-12: Variation of RMSE of parameter σ with time for different number of particle N used in the filter.

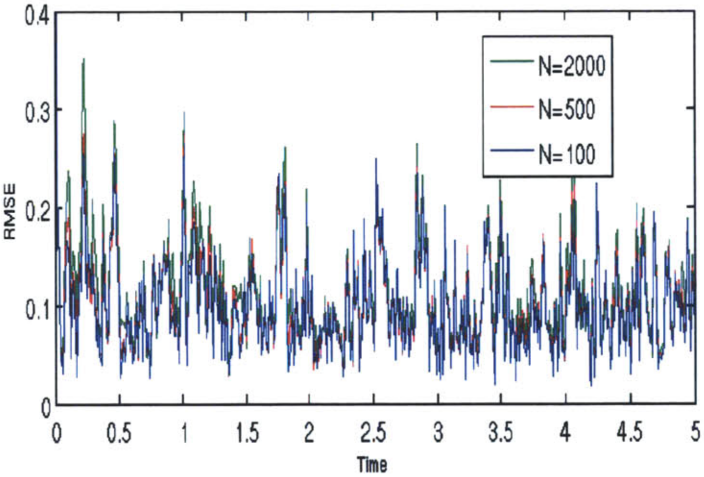


Figure 2-13: Variation of RMSE of state variable x with time for different number of particle N used in the filter.

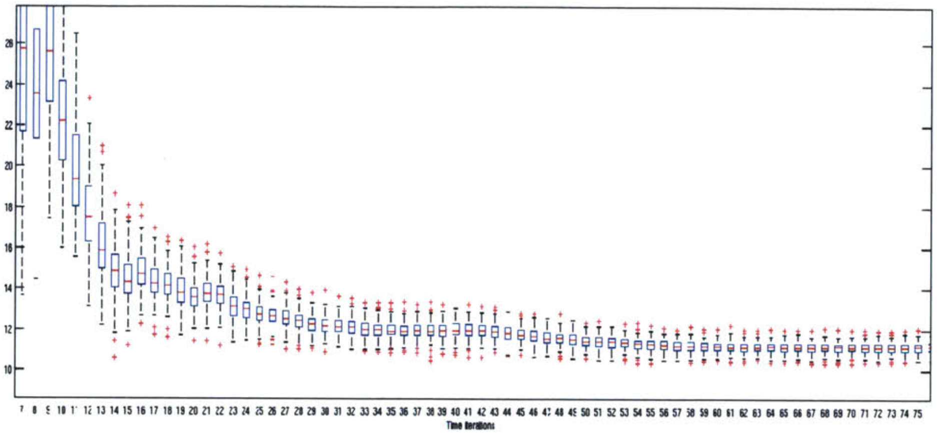


Figure 2-14: Boxplot of the estimate $\hat{\sigma}_m$ over several realizations m with time iterations

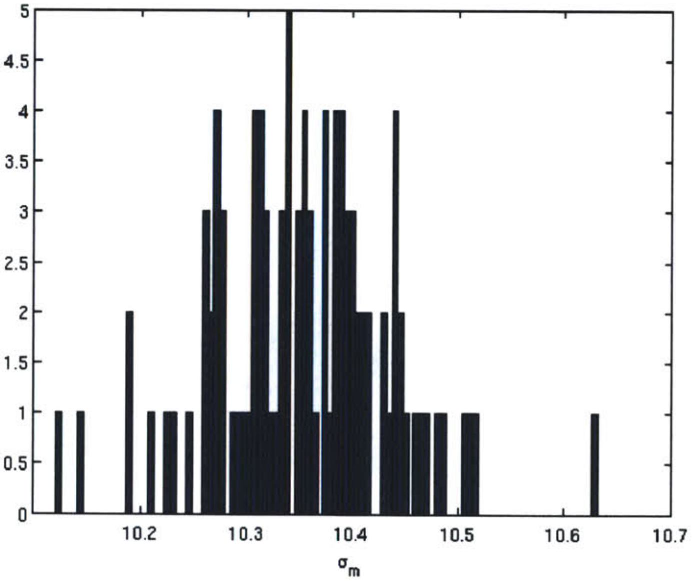
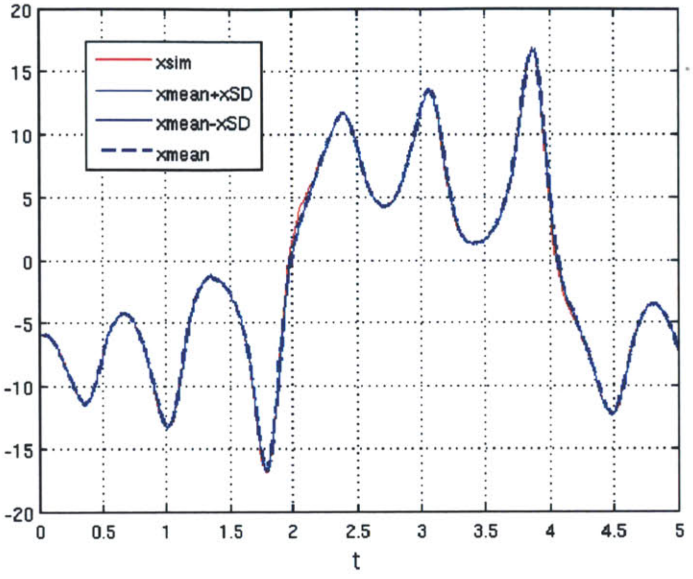
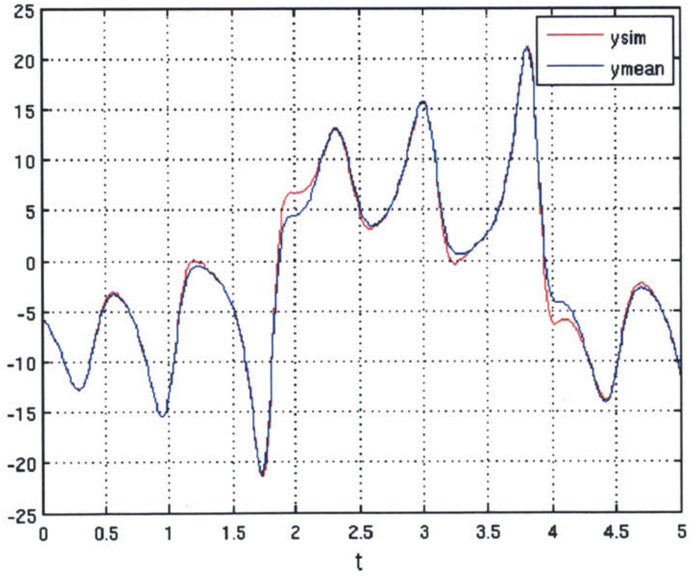


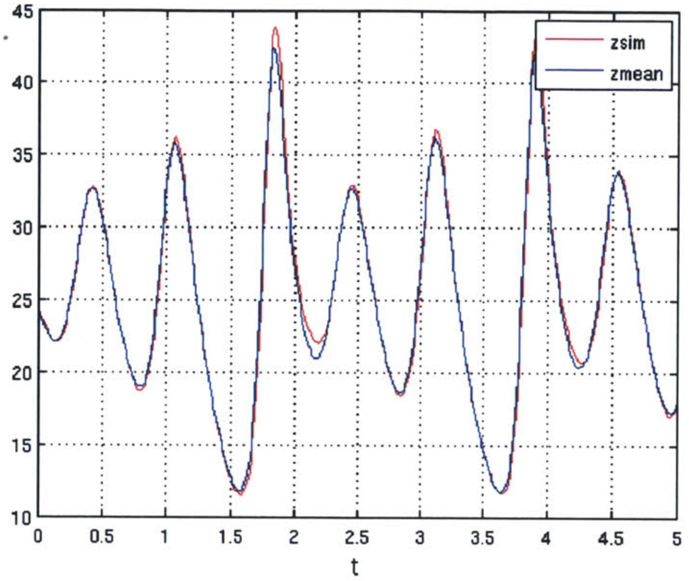
Figure 2-15: Mean estimate $\hat{\sigma}_m$ distribution over several realizations m at time $t = 5s$


xsim
xmean+xSD
xmean-xSD
xmean
20
15
10
5
0
-5
-10
-15
-20
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
t

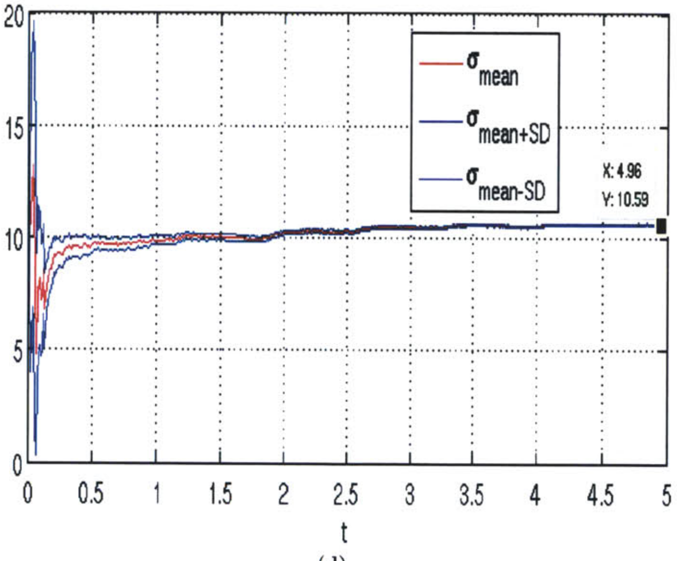
(a)


ysim
ymean
25
20
15
10
5
0
-5
-10
-15
-20
-25
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
t

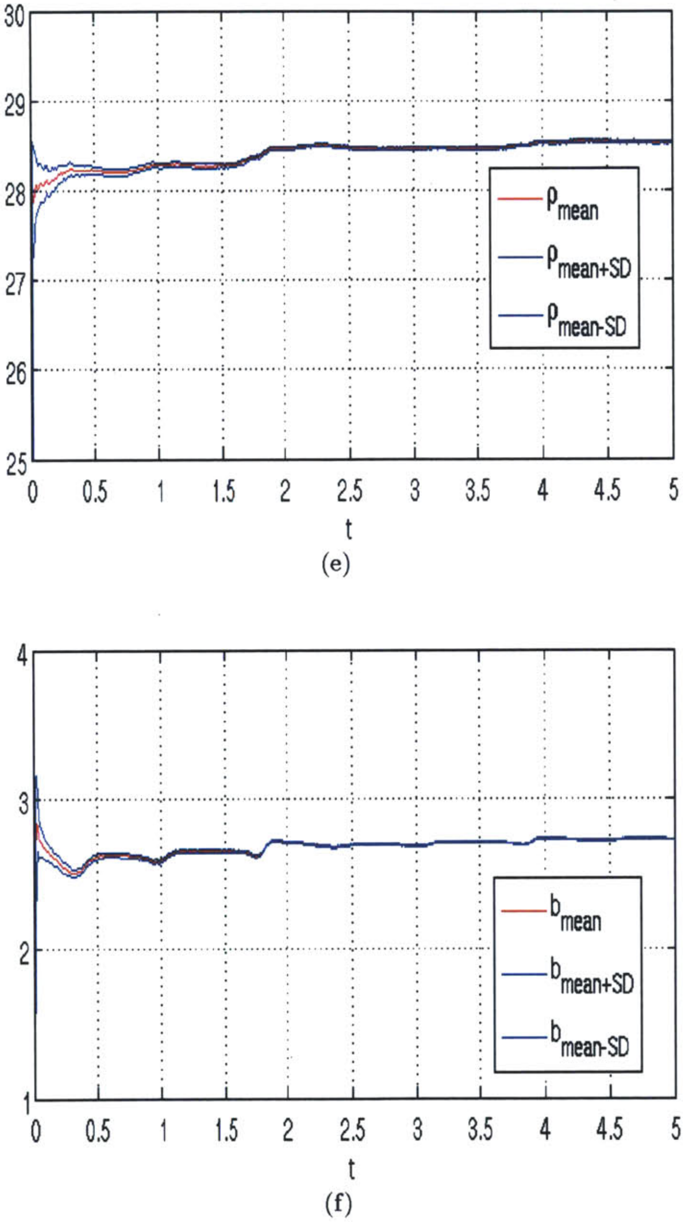
(b)


zsim
zmean
45
40
35
30
25
20
15
10
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
t

(c)


σ_{mean}
$\sigma_{mean+SD}$
$\sigma_{mean-SD}$
X: 4.96
Y: 10.59
20
15
10
5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
t

(d)



(e)

(f)

Figure 2-14: Estimates of various states and parameters against their true values

$$p(x_{t+1}, \theta | y_{1:t}) \sim p(y_{t+1}|x_{t+1}, \theta) p(x_{t+1}|\theta, y_{1:t}) p(\theta|y_{1:t})$$

which makes it explicit that the estimation of the density function $p(\theta|y_{1:t})$ is crucial for the joint parameter and state estimation. The two important ingredients involved in the Liu and West Algorithm are the artificial evolution and kernel smoothing of parameters.

The idea of adding small random disturbances to the states at every time step to reduce sample degeneracy was proposed Gordon et al [28]. Liu and West apply the same to static parameters by considering a different model in which the model parameters θ are replaced by θ_t and it is simply included in the an augmented state vector.

$$\theta_{t+1} = \theta_t + \zeta_{t+1} \text{ where } \zeta_{t+1} \sim N(0, W_{t+1})$$

for some specified variance matrix W_{t+1} and θ_t and ζ_{t+1} independent given $y_{1:t+1}$. The motivating idea for evolving parameters is that at each time step in the simulation new parameter values are generated resulting in re-weighting of the samples and lower degeneracy. However, a drawback is that artificial evolution creates "loss of information" as the parameters do not evolve in reality, leading to posteriors of the parameters which are more diffused compared to their true posteriors. Liu ánd West propose the kernel smoothing of parameters to modify the artificial evolution to avoid "information loss".

At a time t, the current posterior parameter samples $\{\theta_t\}_{i=1}^{N}$ and weights $\{w_t\}_{i=1}^{N}$ provide a Monte Carlo approximation to $p(\theta|y_{1:t})$. Suppose $\hat{\theta}_t$ and V_t denote the Monte Carlo mean and variance matrices of $p(\theta|y_{1:t})$, computed from the Monte Carlo sample θ_t^i with weight w_t^i, then the smooth kernel density form of West [95], [96] is given by

$$p(\theta|y_{1:t}) \approx \sum_{i=1}^{i} N(\theta|m_t^i, h^2 V_t) \tag{2.21}$$

where m_t^i corresponds to the i_{th} kernel location and h is a scaling parameter for the variance. h is assumed as a decreasing function of N, so that the kernel components are more concentrated about their locations m_t^i for larger N. Standard kernel methods assume $m_t^i = \theta_t^i$

so that the kernels are located about the existing sample values. However, this results in a kernel density function that is over-dispersed relative to the posterior sample, as the variance of the resulting mixture of normals is $(1 + h^2)V_t$, always larger than V_t. This is a big issue in sequential simulation as an over-dispersed approximation to $p(\theta|y_{1:t})$ will result in an over-dispersed approximation to $p(\theta|y_{1:t+1})$, and the consequent "loss of information" accumulates over time as the operation is repeated. West [95], [96] proposed shrunk kernel locations taking

$$m_t^i = a\theta_t^i + (1 - a)\hat{\theta}_t \tag{2.22}$$

where a is the shrinkage factor. This shrinkage pushes samples θ_t^i towards their mean θ_t before adding a small amount of noise implied by the normal kernel. Liu and West [55] suggest that the artificial evolution method should be modified by introducing correlations between θ_t and the random noise ζ_{t+1} and obtain $a = \sqrt{1 - h^2}$.The following is the general form of the Liu and West Algorithm, based on the above derivations and the concept of auxiliary particle filtering as described in an earlier section.

Liu and West Algorithm:

1. For each $i = 1, 2, ..., N$ identify the prior point estimates of (x_t, θ) given by (μ_{t+1}^i, m_t^i) where

$$\mu_{t+1}^i = E(x_{t+1}|x_t^i, \theta_t^i)$$

may be computed from the state evolution density and

$$m_t^i = a\theta_t^i + (1 - a)\hat{\theta}_t^i$$

is the i^{th} kernel location from equation from (2.22). 2. Sample an auxiliary integer variable k from the set $\{1, 2, ..., N\}$ with probabilities proportional to

$$g_{t+1}^i \propto w_t^i p(y_{t+1}|\mu_{t+1}^i, m_t^i)$$

3. Sample a new parameter vector θ_{t+1}^k from the k^{th} normal component of the kernel density, namely

$$\theta_{t+1}^k \sim N(\cdot|m_t^k, h^2V_t)$$

4. Sample a value of the current state vector x_{t+1}^k from the system equation

$$x_{t+1}^k \sim p(\cdot|x_t^k, \theta_{t+1}^k)$$

5. Evaluate the corresponding weight

$$w_{t+1}^k \sim \frac{p(y_{t+1}|x_{t+1}^k, \theta_{t+1}^k)}{p(y_{t+1}|\mu_{t+1}^k, \theta_{t+1}^k)}$$

6. Repeat step (2)-(5) a number of times (say G) to produce a final posterior approximation $(x_{t+1}^k, \theta_{t+1}^k)$ with weights w_{t+1}^k as required.

The Liu and West Algorithm has two additional parameters a and G, whose effect on the posterior approximations of the states and parameters needs to be studied. We use the Lorenz model to as an example. The Stochastic Lorenz equation is discretized using the Euler-Maruyama scheme with a numerical discretization time of $t = 0.01s$ with an observation made at every instant (observations made less frequently results in the algorithm performing poorly). The best results are obtained for realistic observation noises between 0.05 and 0.3. We use an observation noise of 0.1 for all our parameter estimation studies henceforth.

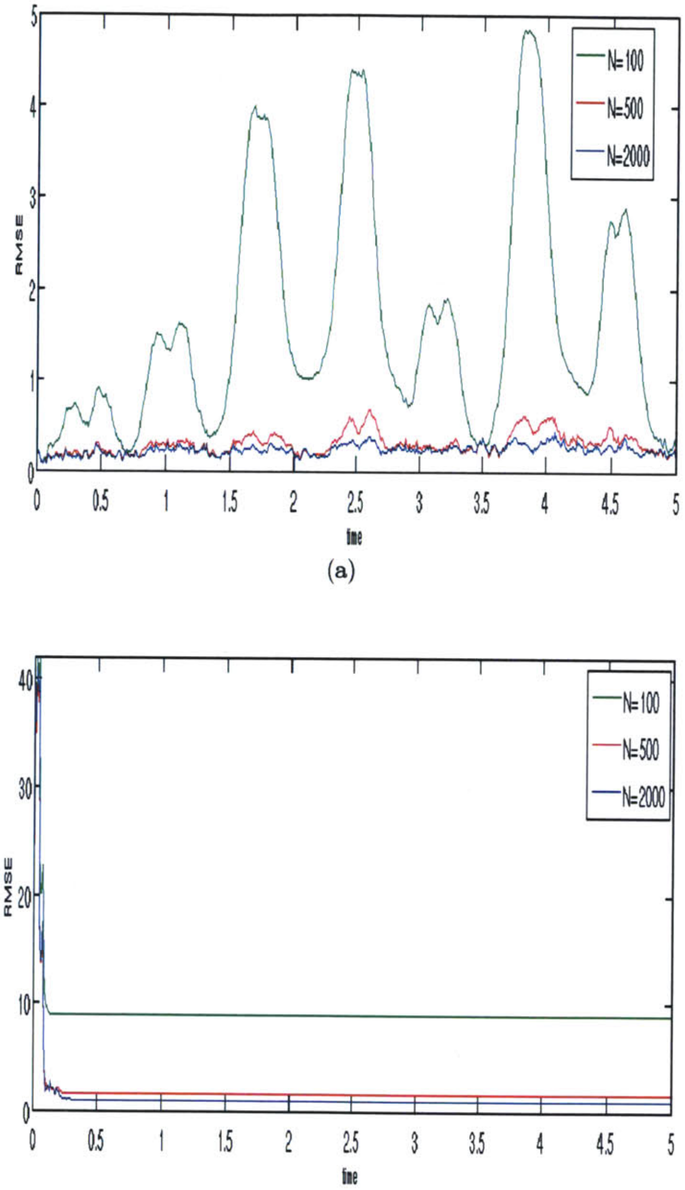
First, we study the single parameter estimation problem, using a prior of $N(0, 100)$ on the parameter σ. The parameter a associated with the algorithm basically describes the shrinkage factor i.e a smaller value of a suggest that after every time iteration the samples of the parameters take values closer to their mean. We study the effect of the parameter a on the algorithm performance taking $G = 1$ and 500 particles. Figure 2-15 shows the effect of a on the RMSE of state variable x and parameter σ. We can observe from the RMSE values that as we decrease the value of a the algorithm performance improves. However, decreasing the value of a too much collapses all the kernel approximation. In case of the Lorenz model, a value of $a = 0.5$ can be said to optimum. We now look at the effect of the number of particles on the algorithm's performance as in figure 2-16. We notice that for $N = 100$ particles the algorithm fails miserably. It performs well for sample sizes of 500 and 2000. Next, we explore the effect of the parameter G on the algorithms performance for reasonable values of $a = 0.5, N = 500$. Figure 2-17 shows that G does not effect the output of the result. Even $G = 1$ suffices to give good results from the algorithm in the Lorenz case. This can be attributed to the sequential nature of the problem. The sequential nature of the problem does not require many iterations at each time step to predict the joint posterior of states and parameter $p(x_t, \theta_t | y_{1:t})$ because with every time step the marginal posterior of the parameters $p(\theta_t | y_{1:t})$ gets closer to the true posterior.

As in the Storvik's Algorithm, we also look at the box plot of the parameter estimate $\hat{\sigma}_m$ over several realizations with time in figure 2-18. The parameter estimate moves quickly towards the true value of the parameter in the first few time iterations and then evolves slowly. Figure 2-19 gives the distribution of the parameter estimate over several realizations at time $t = 5s$.

Finally, we try simulate the posteriors for a single realization in the case when all three parameters appearing in the Lorenz equations are unknowns. We use priors $\sigma \sim N(10, 10), \rho \sim N(28, 10), b \sim N(8/3, 10)$. The mean estimates of the parameters after $5s$ are $\sigma = 10.00, \rho = 27.91, b = 2.89$ which are very close to their actual values. Figure 2-18 show the estimates of the state variable and parameters for the case when all three parameters are unknown. We can visually infer that the filtered state estimates and the true states overlap with each other. SD denotes the standard deviation of the variable in the figures.

2.3.3 Practical Filter

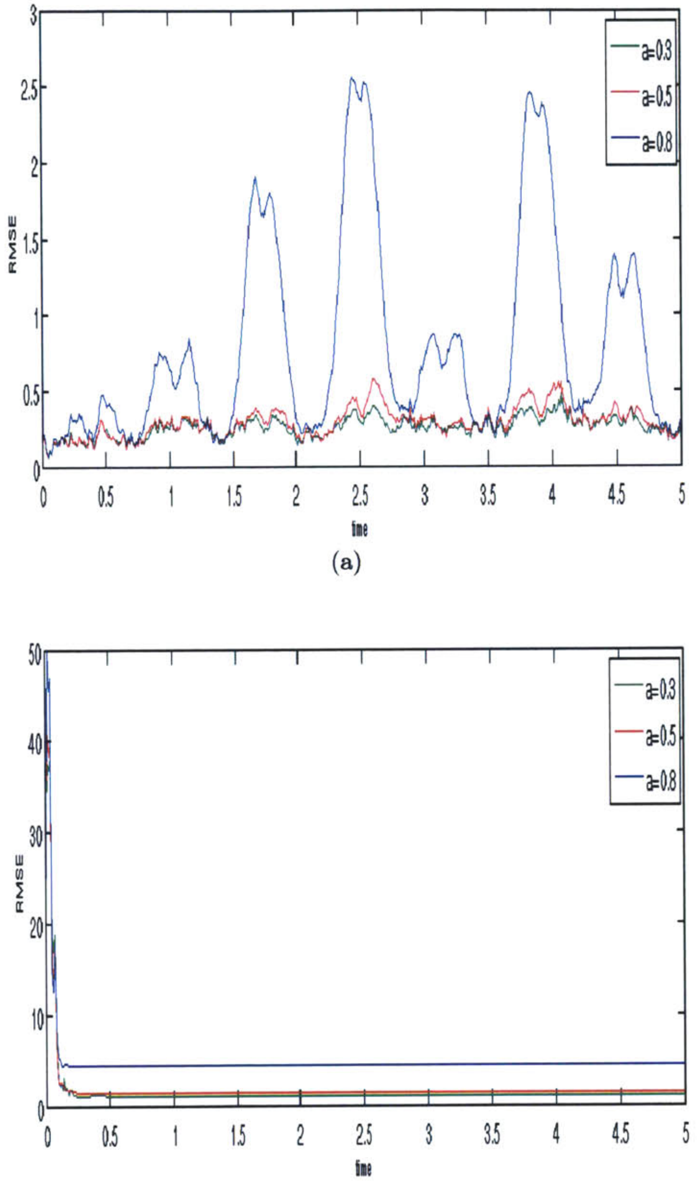
The Practical Filter proposed by Polson et al [73] is a simulation based approach to filtering and sequential parameter learning which uses a key idea of expressing filtering distribution as a mixture of lag-smoothing distributions. It is based on the concept of fixed-lag smoothing



(a)

(b)

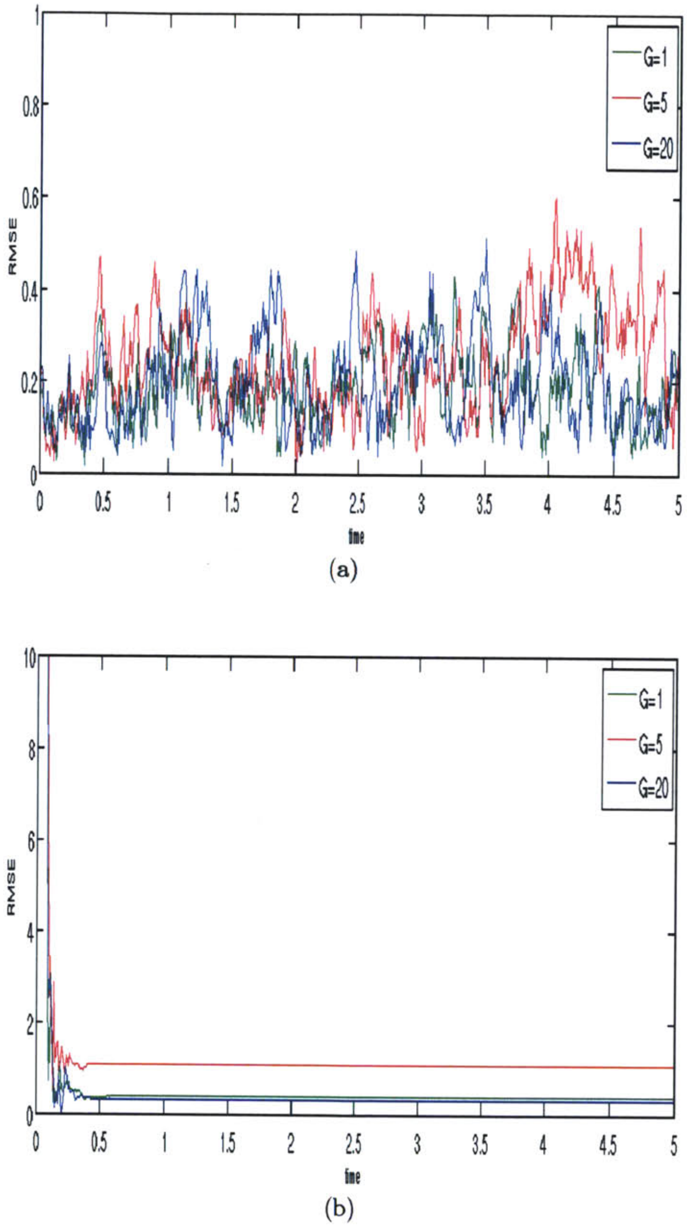
Figure 2-15: Variation of RMSE of state variable x and parameter σ with time for different values of algorithm parameter a having $N = 500$ and $G = 1$



(a)

(b)

Figure 2-16: Variation of RMSE of state variable x and parameter σ with time for different values number of particles used N having $a = 0.5$ and $G = 1$



(a)

(b)

Figure 2-17: Variation of RMSE of state variable x and parameter σ with time for different values of algorithm parameter G having $N = 500$ and $a = 0.5$

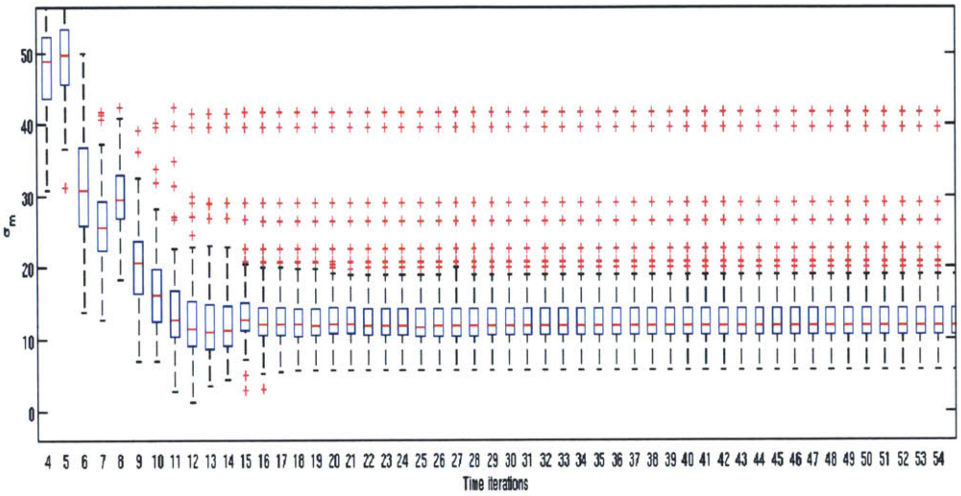


Figure 2-18: Boxplot of the estimate $\hat{\sigma}_m$ over several realizations m with time iterations

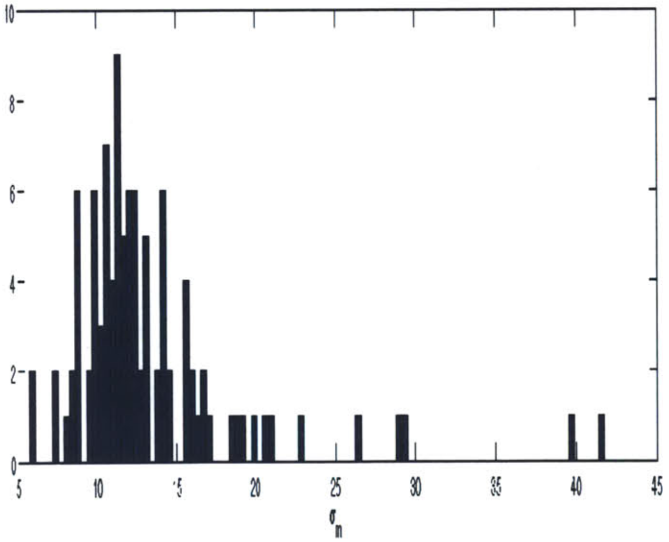
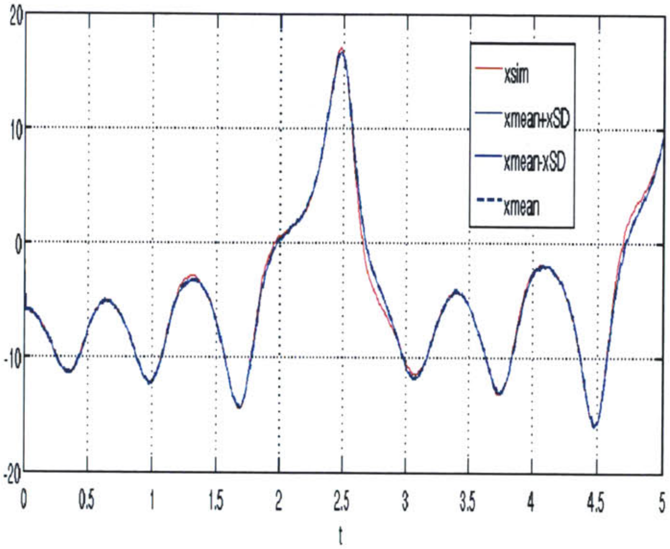
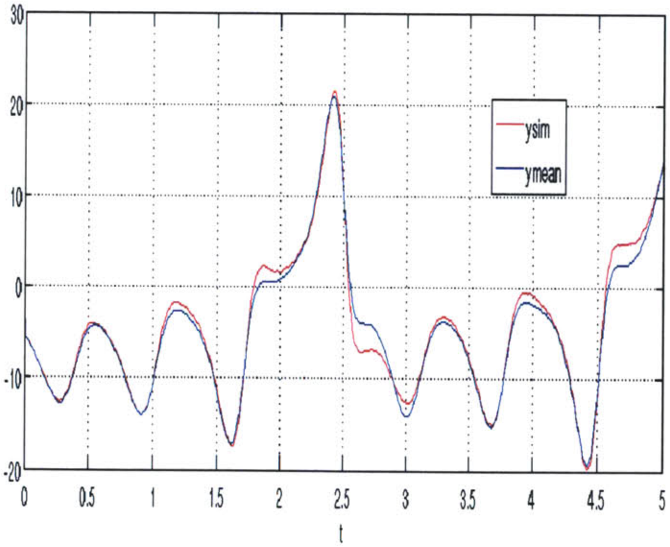


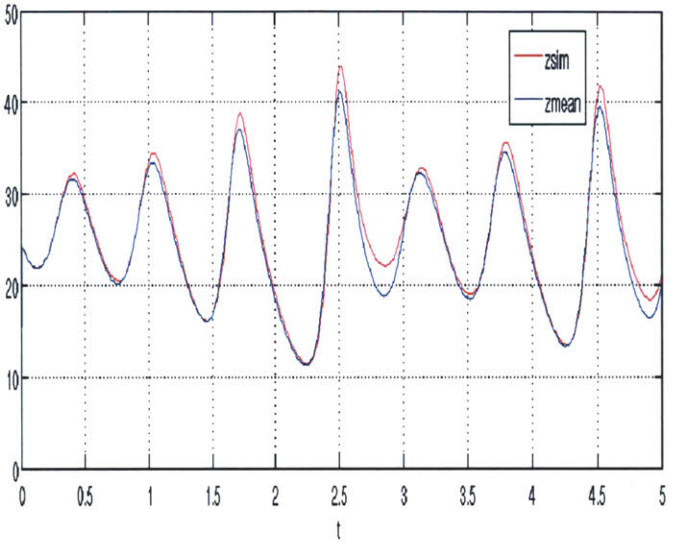
Figure 2-19: Mean estimate $\hat{\sigma}_m$ distribution over several realizations m at time $t = 5s$


xsim
xmean+xSD
xmean-xSD
xmean
20
10
0
-10
-20
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
t

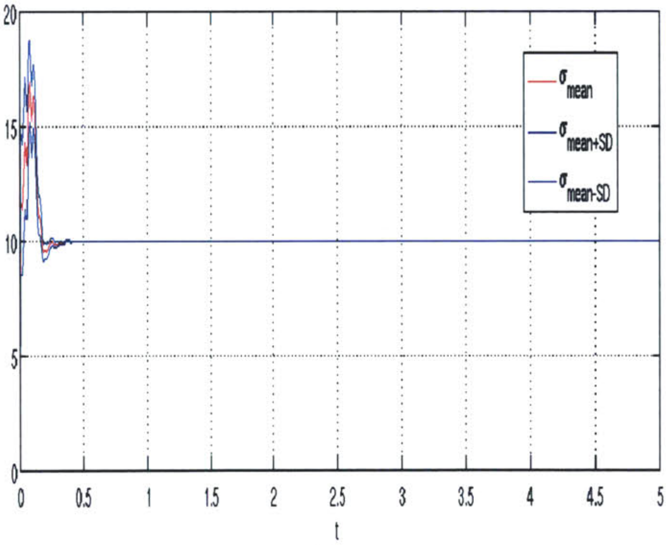
(a)


ysim
ymean
30
20
10
0
-10
-20
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
t

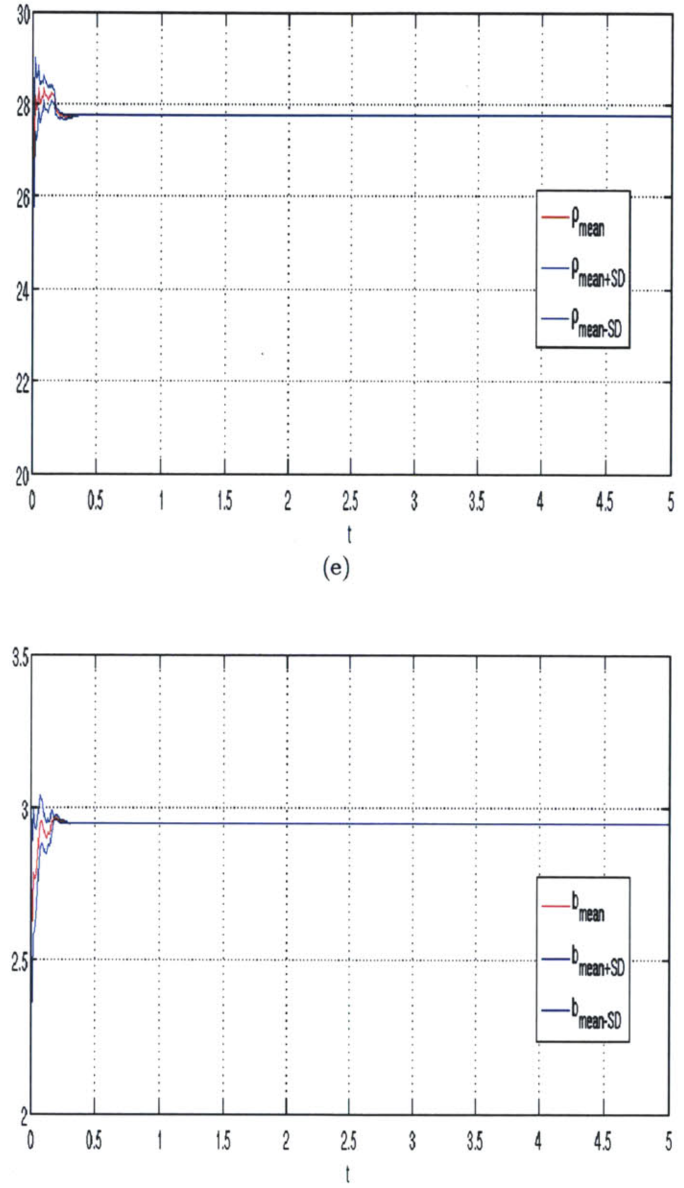
(b)


zsim
zmean
0
10
20
30
40
50
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
t

(c)


σ_{mean}
$\sigma_{mean+SD}$
$\sigma_{mean-SD}$
0
5
10
15
20
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
t

(d)



(e)

(f)

Figure 2-18: Estimates of various states and parameters against their true values

as described before. In this section we discuss the application of the same idea to sequential parameter learning. The most importance difference between the Practical Filter and the earlier approaches is that it uses an MCMC scheme and not importance sampling to sample the particles.

As earlier, the lag-k smoothing distribution, now augmented with the static parameter θ is given by:

$$\begin{aligned} p(x_{t-k+1:t}, \theta | y_{1:t}) &= \int p(x_{t-k+1:t}, \theta | x_{1:t-k}, y_{1:t}) \mathrm{d}p(x_{1:t-k} | y_{1:t}) \\ &\approx \int p(x_{t-k+1:t}, \theta | x_{1:t-k}, y_{1:t}) \mathrm{d}p(x_{1:t-k} | y_{1:t}) \end{aligned}$$

Again, we use the approximation that draws from $p(x_{1:t-k}|y_{1:t-1})$ approximate draws from $p(x_{1:t-k}|y_{1:t})$. Like in the plain filtering case, we cannot use the Markov property of the dynamic model to simplify the above equation further, as the Markov property now holds only conditional on the parameter θ. Instead, we proceed via a Gibbs sampling scheme as follows. First assume that $\hat{x}_{1:t-k}|y_{1:t})$ are samples from $p(\hat{x}_{1:t-k}|y_{1:t-1})$ which are approximate samples from $p(\hat{x}_{1:t-k}|y_{1:t})$. Next, we generate from $p(x_{t-k+1:t}, \theta|\hat{x}_{1:t-k}, y_{1:t})$ by repeated simulation from

$$p(x_{t-k+1:t}|\hat{x}^i_{t-k}, \theta, y_{t-k+1:t}) \text{ and } p(\theta|\hat{x}^i_{1:t-k}, x_{t-k+1:t}, y_{1:t})$$

We have exploited the conditional independence of $x_{t-k+1:t}$ and $(x_{1:t-k-1}, y_{1:t-k})$ given (x_{t-k}, θ), which is the Markov property of the dynamic model. Repeated Gibbs sampling from $p(x_{t-k+1:t}, \theta|\hat{x}_{1:t-k}, y_{1:t})$ followed by saving the last imputed value of x_{t-k+1} as $\hat{x}^i_{t-k+1}$.

Algorithm:

```
Initialization: Set $\theta^i = \theta_0, i = 1, 2, ..., M$. Use, for example, the prior mean.
Burn-In: For $t = 1, 2, ..., k :$
         For $i = 1, 2, ..., M$ : initialize $\theta = \theta^i$.
            For $g = 1, ..., G$ :
               Generate $x_{1:t} \sim p(x_{1:t}|\theta, y_{1:t})$
            Generate $\theta \sim p(\theta|x_{1:t}, y_{1:t})$
        Set $(\hat{x}_1^i, \theta^i)$ equal to the last imputed $(x_1, \theta)$.
Sequential updating: For $t = k+1, ..., T$ :
      For $i = 1, 2, ..., M$ : initialize $\theta = \theta^i$.
        For $g = 1, ..., G$ :
            Generate $x_{t-k+1:t} \sim p(x_{t-k+1:t}|\hat{x}_{t-k}^i, \theta, y_{t-k+1:t})$
            Generate $\theta \sim p(\theta|\hat{x}_{1:t-k}, x_{t-k+1:t}, y_{1:t})$
        Set $(\hat{x}_{t-k+1}, \theta^i)$ equal to the last imputed $(x_{t-k+1}, \theta)$ and leave $\hat{x}_{1:t-k}^i$ unchanged.
```

The performance of the algorithm depends on three parameters (M, G, k). The parameter corresponds to the number of samples drawn from the joint distribution or the number of particles involved in the simulation. If the Metropolis Hastings scheme is employed ,then one can use a small number of samples. However, one should remember that, there is additional computational complexity involved due to Metropolis iterations inside each Gibbs iteration. The parameter G denoted the number of MCMC Gibbs iterations involved to simulate from $p(x_{t-k+1:t}, \theta|\hat{x}_{1:t-k}, y_{1:t})$. In most cases, we can choose $G = 1$ and use FFBS as earlier. However, if the parameter can change values and not entirely static, it is recommended to use a small number of MCMC Gibbs iterations. However, even then, the value of G chosen can be reasonably small because the sequential nature of the problem ensures that the initial starting values will be already close to the stationary distributions. Next, if θ enters the model only through the evolution equation then the choice of appropriate lag k could be highly dependent on the true posterior of θ. Polson et al [73] mention a few heuristics to choose an appropriate k.

Finally one may be able to update θ using a sufficient statistic. Polson et al [73] uses examples in which the conditional dependence of the parameter is known to be gaussian, i.e. $\theta|x_{1:t}, y_{1:t} \sim N(\hat{\theta}_t, V_t)$ where $\hat{\theta}_t$ and V_t are the mean and variance of the gaussian. The update equations for these sufficient statistics are given by $V_t^{-}1 = V_{t-k}^{-}1 + H'H$ and $\hat{\theta}_t = V_t^{-}1(V_{t-k}\hat{\theta}_{t-k} + H'x_{t-k+1})$ where the evolution equation is defined as $x_t = H\theta_t + \eta_t$,

η is the stochastic term.Polson et al [73] compare the Practical Filter with the Storvik's Algorithm in several aspects. It turns out that for the static parameter learning case, the Practical filter is more computationally expensive and gives similar results and hence has no advantage. The real advantage of the practical filter comes into notice is the parameter changes suddenly, or is not entirely static. The results from the practical filter are much better than Storvik's Algorithm. This is mainly attributed to the fixed-lag smoothing and repeated Gibbs iterations involved in the algorithm.

Chapter 3

Point Vortex Model

The estimation problem we discuss here is associated with fluid flow. The idea is to estimate the flow field (comprising the position and velocity fields) using measurements from passive tracers released in the fluid. These tracers go around the fluid as governed by the flow field and do not interact with the fluid at all. We restrict ourselves to a two-dimensional problem, where the tracers are released such that they float on the surface of the fluid. Measurements are obtained in form of position of the tracer in the fluid at regular intervals of time. These measurements, along with the system model for fluid flow is used to estimate the flow fields. As mentioned earlier, this problem finds a direct application in ocean modeling, where the position data can be obtained in the form of satellite images of the ocean.

The system or process model in the most general case would be the Navier-Stokes equations. It is known that these equations are highly non-linear. Also, the resulting flow can be extremely chaotic which can make the inference process even harder. Kalman filtering and its modified versions like the Extended Kalman Filter (EKF) are expected to fail in such cases. Spiller et al [84] used a highly simplified version of the process model (non-linear but not chaotic) namely, the point vortex model and concluded that the EKF gives a very bad approximation of the states when compared to particle filtering. In our study, we use the particle filtering approach to infer the states. Further, we use a point vortex model

[17, 21, 84, 41]which is a vorticity based approximation of the Euler equations.

3.1 Point Vortex Model

The Navier-Stokes equations can be approximated by the Euler equations in case of a inviscid and incompressible flow. The Euler equations are as follows

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0 \tag{3.1}$$

$$\frac{\partial \rho u}{\partial t} + u \cdot \nabla(\rho u) = -\nabla P \tag{3.2}$$

The momentum conservation Euler equation can be further simplified by changing it into the vorticity form. Vorticity field is given by $\omega = \nabla \times u$. Taking the the curl on either side of eq. 3.2 helps in getting rid of the pressure term and we get the following equation in terms of the vorticity and velocity by setting the density to unity.

$$\frac{\partial \omega}{\partial t} + \nabla \cdot (\omega \times u) = 0 \tag{3.3}$$

Continuing further with the assumption that the our model is two dimensional and using eq. 3.1, we can simplify eq. 3.3 to the following form

$$\frac{\partial \omega}{\partial t} + u \cdot \nabla \omega = 0 \tag{3.4}$$

Equation 3.4 means that the material derivative of vorticity is zero i.e. $\frac{D\omega}{Dt} = 0$. So, for a two dimensional flow the vorticity field does not change with time in the Lagrangian sense. In other words, if a particle in the fluid has a vorticity ω at a particular time, it will have the same vorticity ω at all subsequent times. This fact provide the main motivation of the point vortex model. In the point vortex model, it is assumed that the fluid consists of finite number of points called point vorticities having non-zero vorticities. The point vortices move in the fluid as dictated by the other point vortices. Every other point is assumed to have no vorticity and are propagated forward in time only due to the effect of the point vortices. This framework simplifies the Euler equations to set of coupled of ordinary differential equations in terms of the positions of the point vorticities given by eq. 3.5 below. The vorticity field at any point $(x, y) \in \mathbb{R}^2$ in the system is approximated as a sum of delta functions as shown in eq. 3.6. Once the positions of these point vortices are known the positions of all the other particles in the fluid can be found out using eq. 3.7 below.

$$\begin{aligned} \frac{d\alpha^i}{dt} &= \sum_{j=1,j\neq i}^{V} \frac{\Gamma^j}{2\pi} \frac{(\beta^i - \beta^j)}{(\alpha^i - \alpha^j)^2 + (\beta^i - \beta^j)^2} \\ \frac{d\beta^i}{dt} &= \sum_{j=1,j\neq i}^{V} \frac{\Gamma^j}{2\pi} \frac{(\alpha^j - \alpha^i)}{(\alpha^i - \alpha^j)^2 + (\beta^i - \beta^j)^2} \end{aligned} \tag{3.5}$$

$$\omega(x, y) = \sum_{j=1}^{V} \Gamma^j \delta_{(\alpha^j,\beta^j)}(x, y) \tag{3.6}$$

$$\begin{aligned} \frac{da}{dt} &= \sum_{j=1}^{V} \frac{\Gamma^j}{2\pi} \frac{(b - \beta^j)}{(a - \alpha^j)^2 + (b - \beta^j)^2} \\ \frac{db}{dt} &= \sum_{j=1}^{V} \frac{\Gamma^j}{2\pi} \frac{(\alpha^j - a)}{(a - \alpha^j)^2 + (b - \beta^j)^2} \end{aligned} \tag{3.7}$$

Here (a, b) and $(\alpha^i, \beta^i), i = 1, 2, .., V$ denote the positions of the tracer and vortices respectively, V being the total number of point vortices. Also, Γ^i denotes the circulation associated with the i^{th} point vortex. We note here that, for the point vortex model, since the vorticity is assumed to concentrated at a singular point (zero area) its vorticity is a infinite. However, the circulation, which is the integral of the vorticity over the area is finite, even though the vorticity is delta function.

The point vortex model largely simplifies the inference problem. Instead of having to track the state vector of the entire system (every point in the fluid), we now only need to track the states of vortices as they move around in the fluid. The states of any other point in the system can be determined using the states of these handful point vortices. Hence the most important advantage of the point vortex is that we have changed a very high-dimensional filtering problem into a relatively low dimensional problem. Additionally, the point vortex model in two dimensions only involves ordinary differential equations making it the problem more easier.

3.2 Vortex Configurations

To make the inference problem interesting we investigate the flow using different number of point vortices and their relative positions. We try cases with 3 and 4 vortices and varying configurations. In all cases, the flow is nonlinear as given by the above equation. However, depending on the relative configuration of the vortices the flow can be dramatically different.

3.2.1 3 and 4 vortex systems

Aref and Pomphrey [4] showed that the three-vortex problem in two-dimensions is integrable, whereas the motion of four vortices is not. A sequence of canonical transformations can be obtained that reduces the V-degree-of-freedom Hamiltonian, which describes the interaction of V identical vortices, to one with V-2 degrees of freedom. Hence, for V=3 a reduction to single degree of freedom is obtained and the problem is integrable. As a consequence of Aref and Pomphrey's result, the 3 vortex is at most quasi-periodic as it is always integrable. On the other hand, the 4 vortex system, cannot be integrated and most 4 vortex configurations lead to chaotic system. Precisely, Aref and Pomphrey established that for $V = 4$ vortices there are three cases that arise. If the four vortices are at the corner of a square, the vortex motion is periodic. If there is inversion symmetry among the two pairs of vortices, we observe a quasi-periodic vortex motion. Finally, if the vortices are in neither of the earlier configurations to start with , then the vortex motion is chaotic. In noisy systems the 4 vortex configuration always leads to chaotic systems. Also, because of the chaotic nature the movement of the vortices become highly dependent on their initial positions. We consider the following cases for our study. All of them consist of vortices which start at the edges of a unit circle.

1. Symmetric 3 vortex system. Vortices start at (0,1), ($\sqrt{3}/2$,-0.5),(-$\sqrt{3}/2$,-0.5)
2. Asymmetric 3 vortex system. Vortices start at (ϵ,$\sqrt{1-\epsilon^2}$), ($\sqrt{3}/2$,-0.5),(-$\sqrt{3}/2$,-0.5), where ϵ denotes the asymmetricity of the system i.e. higher its value, the more asymmetric is the system.
3. Square 4 vortex system. Vortices start at (1,0), (0,1), (-1,0), (0,-1)
4. Inversion symmetry preserving 4 vortex system. Vortices start at (1,0), (-1,0,), (1/$\sqrt{2}$,-1/$\sqrt{2}$), (-1/$\sqrt{2}$,1/$\sqrt{2}$)
5. Asymmetric 4 vortex system. Vortices start at (1,0), (0,1), (-1/$\sqrt{2}$,1/$\sqrt{2}$), (-1/$\sqrt{2}$,-1/$\sqrt{2}$)

3.2.2 Chaos and Chaotic flows

In common usage, chaos refers to a state of disorder. There is no universally accepted mathematical definition of chaos, a commonly used definition says that, for a dynamical system to be classified as chaotic, it must have the following properties [33]:

1. It must be sensitive to initial conditions; Sensitivity to initial conditions means that each point in such a system is arbitrarily closely approximated by other points with significantly different future trajectories. Thus, an arbitrarily small perturbation of the current trajectory may lead to significantly different future behavior [92].
2. It must be topologically mixing; Topological mixing (or topological transitivity) means that the system will evolve over time so that any given region or open set of its phase space will eventually overlap with any other given region [91].

3. Its periodic orbits must be dense; Density of periodic orbits means that every point in the space is approached arbitrarily closely by periodic orbits [23].

As mentioned before, having a chaotic flow also makes the problem more challenging. Chaos is inherently a form of uncertainty. One important property of a chaotic flow is that particles in the fluid can move away or towards each other at an exponential rate in time. This also means that the positions of vortices are very sensitive to their initial positions. It becomes increasing difficult to track them as time goes on, because even small errors in estimation at a particular time can result in the estimated state totally losing track of the true state at subsequent times. Other filtering algorithms fail completely to track chaotic systems. Although particle filtering can handle such problems better, a huge number of particles are required to track the states, otherwise, the filter loses track of the vortices. In the later chapters we shall look at how well the particle filtering techniques can be used to track chaotic 4-vortex system.

3.3 Numerical Integration Schemes

The point vortex model involves solving a system of coupled ordinary differential equations. There are various numerical integration schemes of variable accuracy that one can use for this. A good overview of the different stochastic numerical integration schemes can be found in [47, 46, 72, 48, 98, 35, 79, 10, 64, 40, 71, 34]. The simplest and most well known scheme is the Euler-Maruyama scheme [68, 99, 36]. However, it is only one-half order accurate in time and can lead to large numerical errors. Lamba et al [51] provide a good description of an adative version of the Euler-Maruyama scheme. Another frequently used scheme, when the diffusion coefficient is not a constant, is the Milstein scheme [47, 72, 68, 43]. The Klauder-Petersen scheme, which is first order accurate in time, has been used several times in the context of particle filtering by Chorin et al [6, 18]. In this study, we choose to use the 4-order Runge-Kutta scheme (only second order accurate in time), which is better known as RK4.

We show the implementation of the 4th order stochastic Runge Kutta scheme for a scalar variable as follow:

$$\mathrm{dy} = \mathrm{f}(y)\mathrm{dt} + \sigma \mathrm{dW} \tag{3.8}$$

Equation 3.8 is ordinary stochastic differential equation in a scalar variable y

The stochastic RK4 method for this problem is given by the following equations:

$$\begin{aligned} y_{n+1} &= y_n + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4) \\ t_{n+1} &= t_n + h \end{aligned}$$

where y_{n+1} is the RK-4 approximation of $y(t_{n+1})$, and

$$\begin{aligned} k_1 &= f(y_n) \\ k_2 &= f(y_n + \frac{1}{2}hk_1 + \frac{1}{2}\sigma \mathrm{N}(0,h)) \\ k_3 &= f(y_n + \frac{1}{2}hk_2 + \frac{1}{2}\sigma \mathrm{N}(0,h)) \\ k_4 &= f(y_n + k_3 + \sigma \mathrm{N}(0,h)) \end{aligned} \tag{3.9}$$

Thus, the next value y_{n+1} is determined by the present value y_n plus the weighted average of four increments which are specified by eqs. 3.9. A detailed overview of the different stochastic Runge Kutta schemes can be found in [37, 38]. The deterministic RK-4 scheme is 4-order accurate in time. However, in our estimation problem, the system model is stochastic, i.e. we add gaussian noises to the discretized versions of the ordinary differential equations 3.5 and 3.7. As a result, the stochastic version of the Runge-Kutta 4 like scheme is only second order accurate in time. This is mainly because the variance (proportional to integration error squared) of the Weiners process scales linearly with time. We discuss the convergence and the implication of using this integration scheme in later chapters.

3.4 Estimation Problem Setup

The final process model is given by adding noises to the eqs. 3.5 and 3.7. They can be represented as follows

$$\begin{aligned}\frac{d\alpha^i}{dt} &= \sum_{j=1,j\neq i}^{V} \frac{\Gamma^j}{2\pi}\frac{(\beta^i-\beta^j)}{(\alpha^i-\alpha^j)^2+(\beta^i-\beta^j)^2}+\eta^i_\alpha \\ \frac{d\beta^i}{dt} &= \sum_{j=1,j\neq i}^{V} \frac{\Gamma^j}{2\pi}\frac{(\alpha^j-\alpha^i)}{(\alpha^i-\alpha^j)^2+(\beta^i-\beta^j)^2}+\eta^i_\beta \end{aligned} \qquad (3.10)$$

$$\begin{aligned}\frac{da}{dt} &= \sum_{j=1}^{V} \frac{\Gamma^j}{2\pi}\frac{(b-\beta^j)}{(a-\alpha^j)^2+(b-\beta^j)^2}+\eta_a \\ \frac{db}{dt} &= \sum_{j=1}^{V} \frac{\Gamma^j}{2\pi}\frac{(\alpha^j-a)}{(a-\alpha^j)^2+(b-\beta^j)^2}+\eta_b \end{aligned} \qquad (3.11)$$

Here the state vector to be estimated obtained by augmenting the positions of the vortices with the positions of the tracers and is given by $(\{\alpha^i\}_{i=1}^V, \{\beta^i\}_{i=1}^V, a, b)$. Also, all the process noises in eq.s 3.10 and 3.11 are assumed to be uncorrelated gaussians with variances of 0.02 in accordance with Salman [80]. As mentioned above, we use a fourth order stochastic Runge-Kutta scheme, which is second order accurate in time, to solve eq.s 3.10 and 3.11.

The observation is assumed to be available in terms of the discrete tracer positions over regular time intervals given by

$$\begin{aligned} a_o &= a + \eta_{a_o} \\ b_o &= b + \eta_{b_o} \end{aligned} \qquad (3.12)$$

Here, the observations are given by tracer noisy tracer positions (a_o, b_o) and the observation noises are assumed to be uncorrelated gaussians with variance 0.05 in accordance with Salman [80]. Also, the value of the circulations are assumed to be known and equal to 2π

for all vortices. An advantage of having all the circulations having the same sign is that the point vortices never collapse together.

Chapter 4

Lyapunov exponents and Lagrangian coherent structures

4.1 Finite time Lyapunov exponents

In the last chapter we learnt that chaotic flows are extremely sensitive to initial conditions. Lyapunov exponents can help us ascertain the extent to which the system is sensitive to initial conditions. The Lyapunov exponent or Lyapunov characteristic exponent of a dynamical system is a quantity that characterizes the rate of separation of infinitesimally close trajectories [9, 56, 26, 58]. Some very interesting applications of Lyapunov exponents to dynamical systems outside the area of fluid dynamics can be found in [94, 93, 60, 87, 88]. In fluid dynamics, we can define Laypunov exponents as follows: Two trajectories in phase space with initial separation $\delta \mathbf{x}_0$ diverge (provided that the divergence can be treated within the linearized approximation) at a rate given by

$$|\delta \mathbf{x}(t)| \approx e^{\lambda t} |\delta \mathbf{x}_0| \tag{4.1}$$

Here λ is the Lyapunov exponent.

The rate of separation can be different for different orientations of initial separation vector. Thus, there is a spectrum of Lyapunov exponents equal in number to the dimensionality of the phase space. It is common to refer to the largest one as the maximal Lyapunov exponent (MLE), because it determines a notion of predictability for a dynamical system. A positive MLE is usually taken as an indication that the system is chaotic (provided some other conditions are met, e.g., phase space compactness). It makes sense to consider the MLE because any arbitrary initial separation vector will typically contain some component in the direction associated with the MLE, and because of the exponential growth rate, the effect of the other exponents will be obliterated over time.

Mathematically, the maximal Lyapunov exponent is defined as follows:

$$\lambda = \lim_{t\to\infty} \lim_{\delta\mathbf{x}_0 \to 0} \frac{1}{t} \ln \frac{|\delta\mathbf{x}(t)|}{|\delta\mathbf{x}_0|}. \tag{4.2}$$

The limit $\delta\mathbf{x}_0 \to 0$ ensures the validity of the linear approximation at any time [16].

Evaluating the Lyapunov exponents of the particles in the fluid lead to Lagrangian coherent structures, which are stable/unstable material structures in the flow. There are two kinds of coherent structures; attracting and repelling. We discuss these structures more in the next section. However, as the calculation of Lyapunov exponents involve a limit to infinite time, in practice, we use the finite time Lyapunov exponents (FTLE) instead. Speicifically, in the context of ocean modeling and data assimilation, FTLE has been used in [83, 85, 89]

Suppose the flow map is denoted by $\mathbf{x}(t, t_0, \mathbf{x}_0)$, i.e. the trajectory followed by a particle at point $\mathbf{x}_0$ at time t_0. Let F be the deformation tensor given by,

$$\mathbf{F}^T = \nabla_{\mathbf{x}_0}\mathbf{x}(t, t_0, \mathbf{x}_0) \tag{4.3}$$

Now, F describes the deformation at $(\mathbf{x}, t)$ with respect to the initial conditions. For $t > t_0$, the largest singular value of $\mathbf{F}$, $\sigma_1(\mathbf{F})$, gives the maximum relative growth of an infinitesimal vector at $\mathbf{x}_0$ advanced by the flow map from t_0 to t. Hence we have,

$$\sigma_1(\mathbf{F}) = \sqrt{\lambda_{max}(\mathbf{F}^T\mathbf{F})} \tag{4.4}$$

where $\mathbf{F}^T\mathbf{F}$ is known as the Cauchy-Green tensor [67]. The largest finite time Lyapunov exponent (FTLE) is thus given by [30],

$$\Lambda(t, t_0, \mathbf{x}_0) = \frac{1}{2(t - t_0)} \ln\left(\lambda_{max}(\mathbf{F}^T\mathbf{F})\right) \tag{4.5}$$

We calculate Λ in forward time ($t > t_0$) using a dense initial grid of particles. The derivatives in $\mathbf{F}$ are approximated with central differences. The local maxima of the resulting FTLE field $\Lambda_{t_0}^t(\mathbf{x}_0)$ are repelling coherent structures. These repelling structures signify regions in the flow from where particles diverge away as time increases from t to t_0. We investigate these structure in the next section. An alternate and interesting way to compute the FTLE field has been proposed by Okushima [65]. This method is very helpful for evaluating the FTLE field for multidimensional dynamical systems with degenerate spectra. Beyn et al proposed another hybrid method to evaluate FTLE fields [11].

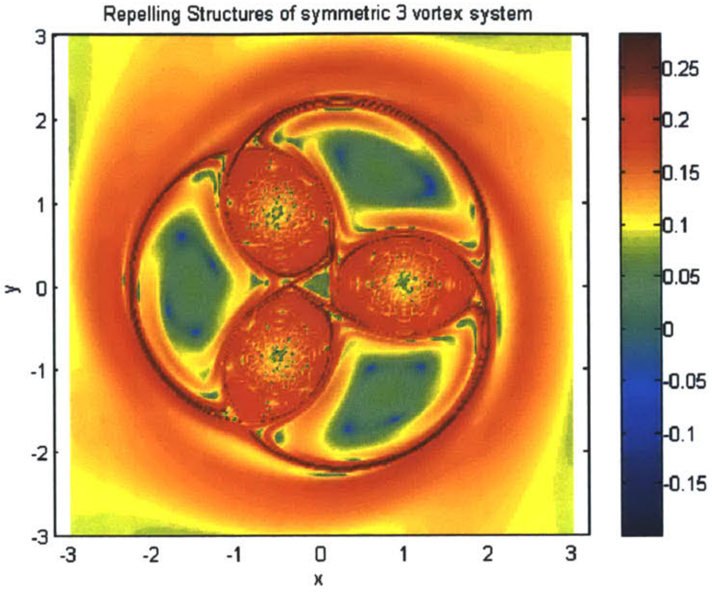
4.2 Lagrangian coherent structures

In this section, we explore Lagrangian coherent structures (LCS), which are defined as ridges of finite-time Lyapunov exponent (FTLE) fields. These ridges can be seen as finite-time mixing templates. In dynamical systems theory, LCS are stable and unstable manifolds of fixed points and periodic orbits.

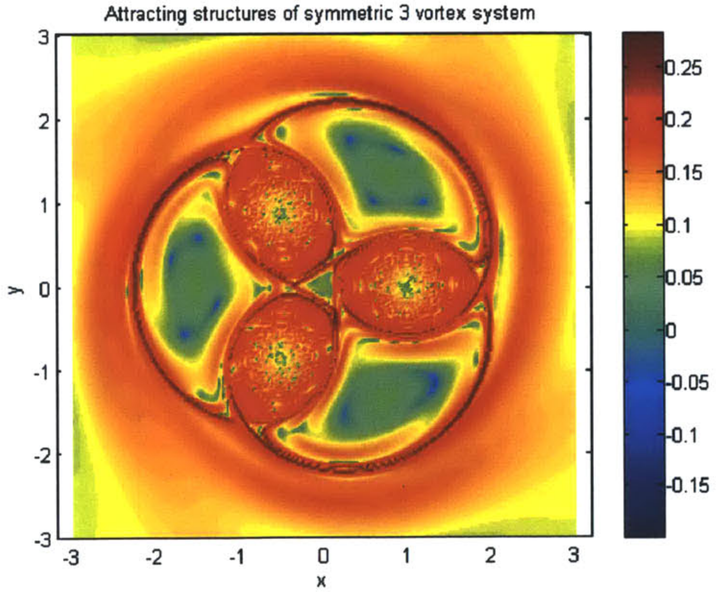
It has long been recognized that flows with general time dependence admit emergent patterns which influence the transport of tracers; these structures are often generically referred to as coherent structures. When coherent structures are studied in terms of quantities derived from fluid trajectories they are often named Lagrangian coherent structures (LCS). For example, in the special case of time-independent systems, one might classify the stable and unstable manifolds of hyperbolic fixed points as coherent structures, but typically one uses the notion of coherent structures in the context of more general flows. There is a vast body of literature on coherent structures in fluid mechanics that we will not attempt to overview here. One can find a good overview of LCS their properties in [50, 32, 59, 83, 27, 61, 52, 54, 31, 53, 76, 7, 20, 82, 69].

One of the most important used of the LCS is that they can reveal an underlying flow structure in time-dependent systems that is typically not evident from the Eulerian field. In addition, these structures divide dynamically distinct regions in the flow, which allow for the easy analysis of transport. Formally, we can define a Lagrangian coherent Structure (LCS) as the second-derivative of the scalar FTLE field $\Lambda(t, t_0, \mathbf{x}_0)$ at a time instant t_0 [83]. We also note here that the flux across these coherent structures, while not necessarily zero, is usually negligible, and therefore these structures are effectively invariant manifolds and hence act as transport barriers. We plan to use these Lagrangian coherent structures evaluate from finite time lyapunov exponents to effectively study the problem of ideal tracer placement to carry out effective filtering in the later chapters.

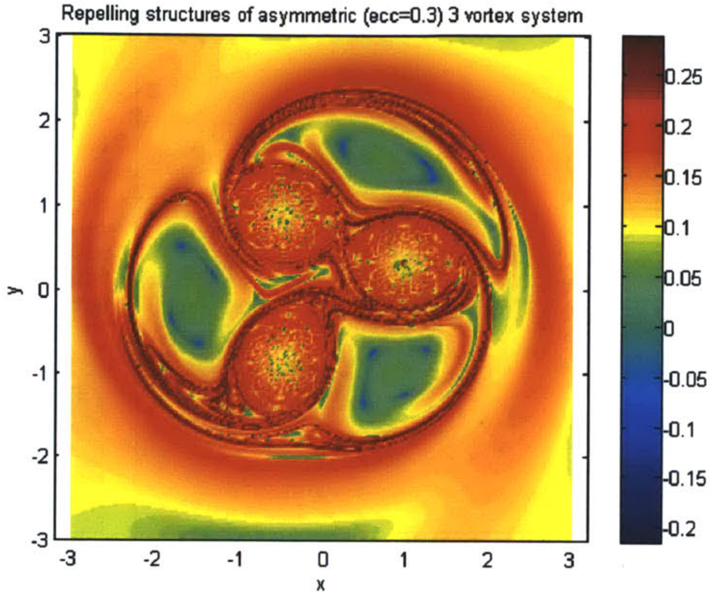
Finally, in figs. 4–2 and 4–2, we show the Attracting and Repelling LCS for simple point vortex systems. We evaluate the repelling LCS by integrating the system in forward time and then evaluating the FTLE field of the space, while we use backward-time integration for locating attracting LCS. This makes sense because repelling structures are signify regions from which the fluids particles diverge away, while attracting structures refer to boundaries which attract fluid particles towards themselves. Figure 4–2 show the LCS for different 3 vortex systems with increasing asymmetricity. We can see that the difference between the attracting and repelling structures of the system increases as we increase the asymmetricity of the system. Also, we can identify regions with high value of FTLE, which have more irregular flow than other regions with fairly low and uniform values of FTLE. We explore more about these structures when we discuss the numerical results. Figure 4–2 shows the LCS for 4 vortex system. We can clearly see that for chaotic systems like an asymmetric 4 vortex systems the LCS can get very complicated with several regions of high FTLE field.


Repelling Structures of symmetric 3 vortex system
3
2
1
y 0
-1
-2
-3
-3
-2
-1
0
1
2
3
x
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15

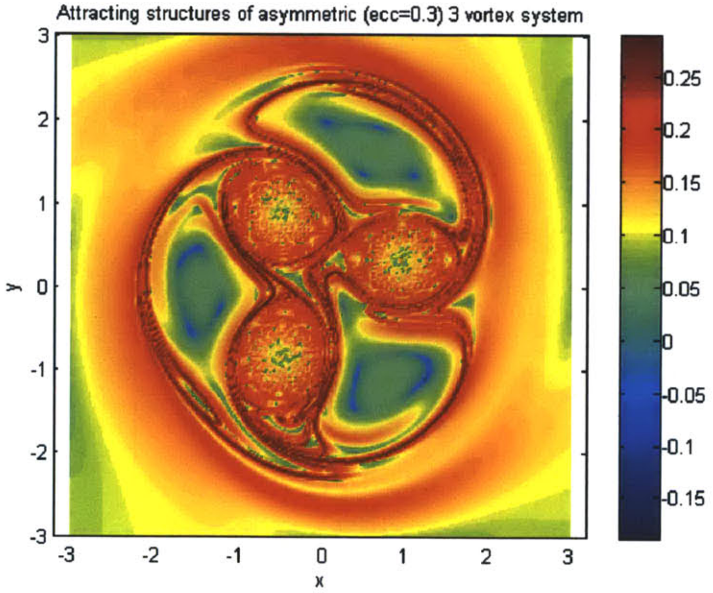
(a)


Attracting structures of symmetric 3 vortex system
3
2
1
y 0
-1
-2
-3
-3
-2
-1
0
1
2
3
x
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15

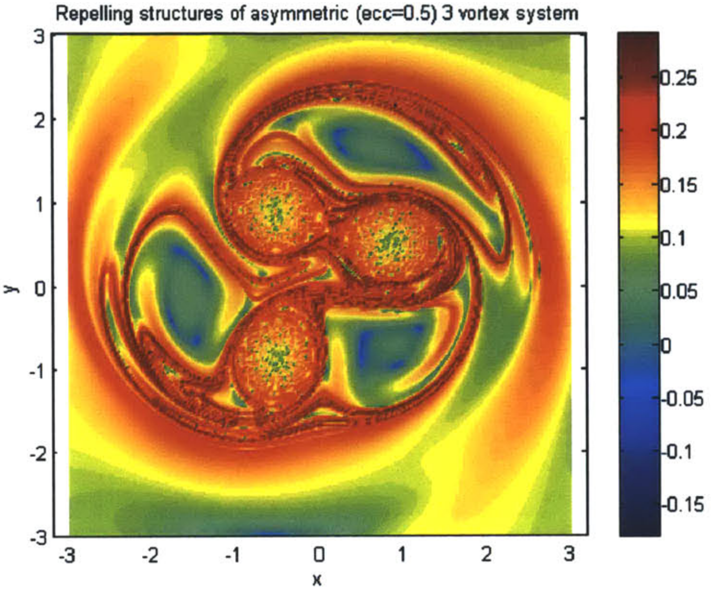
(b)


Repelling structures of asymmetric (ecc=0.3) 3 vortex system
3
2
1
y 0
-1
-2
-3
-3
-2
-1
0
1
2
3
x
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2

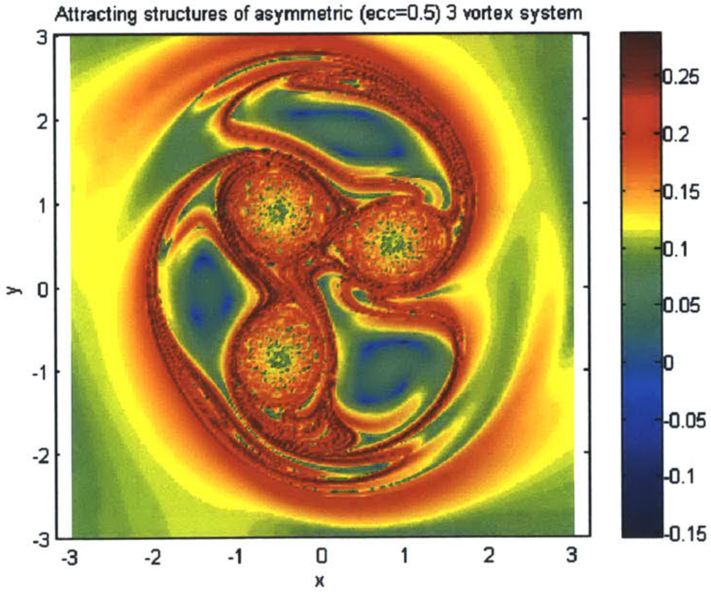
(c)


Attracting structures of asymmetric (ecc=0.3) 3 vortex system
3
2
1
y 0
-1
-2
-3
-3
-2
-1
0
1
2
3
x
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15

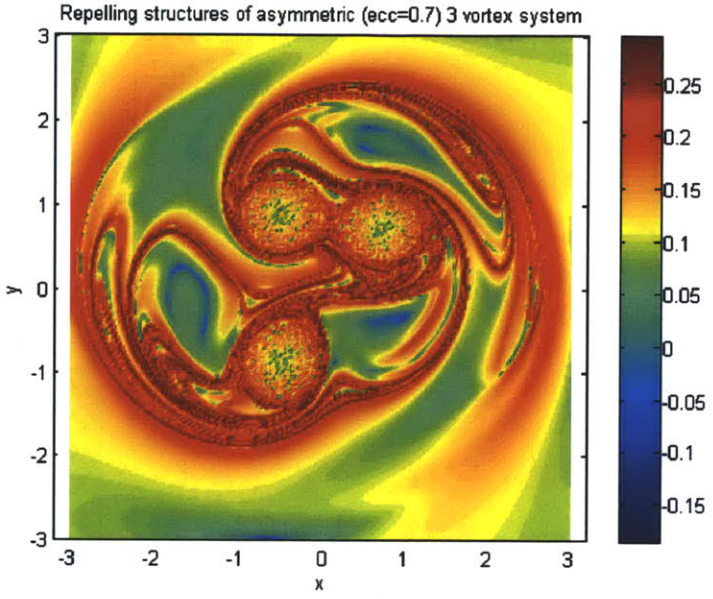
(d)


Repelling structures of asymmetric (ecc=0.5) 3 vortex system
3
2
1
y 0
-1
-2
-3
-3
-2
-1
0
1
2
3
x
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15

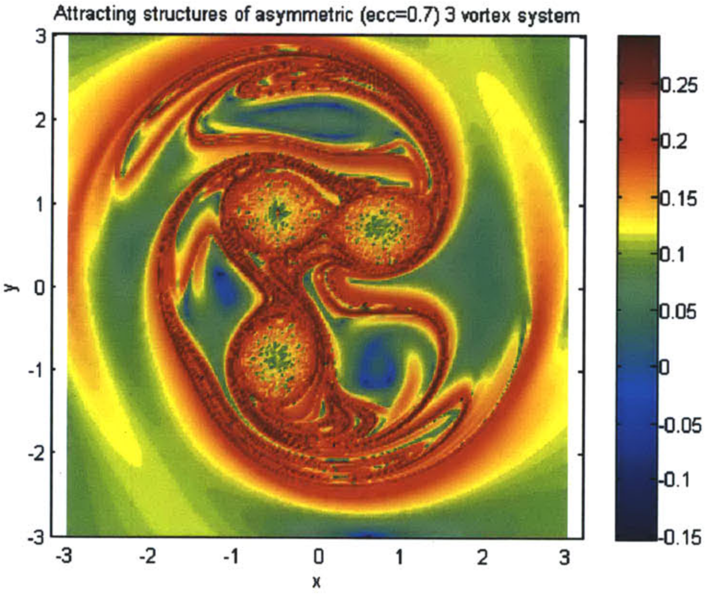
(e)


Attracting structures of asymmetric (ecc=0.5) 3 vortex system
3
2
1
y 0
-1
-2
-3
-3
-2
-1
0
1
2
3
x
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15

(f)

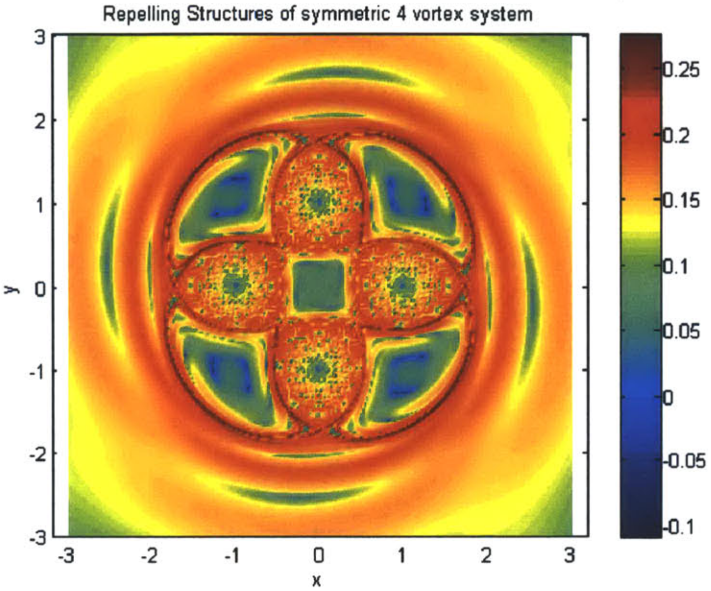


(g)

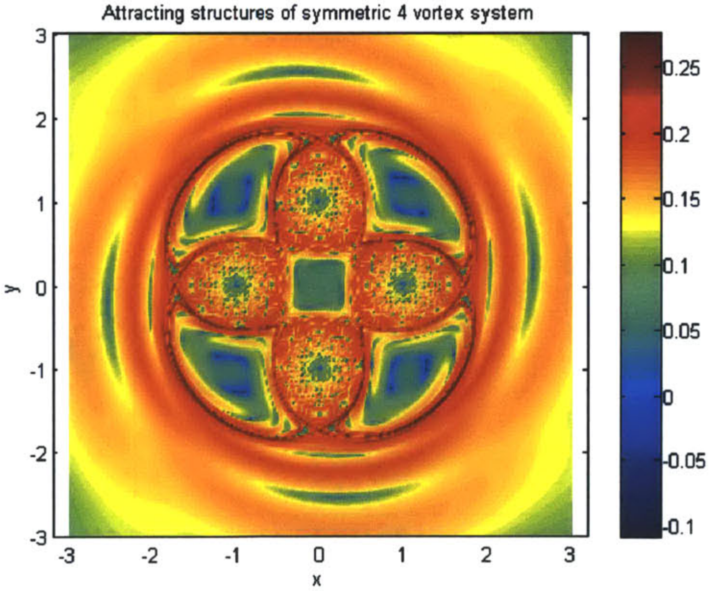


(h)

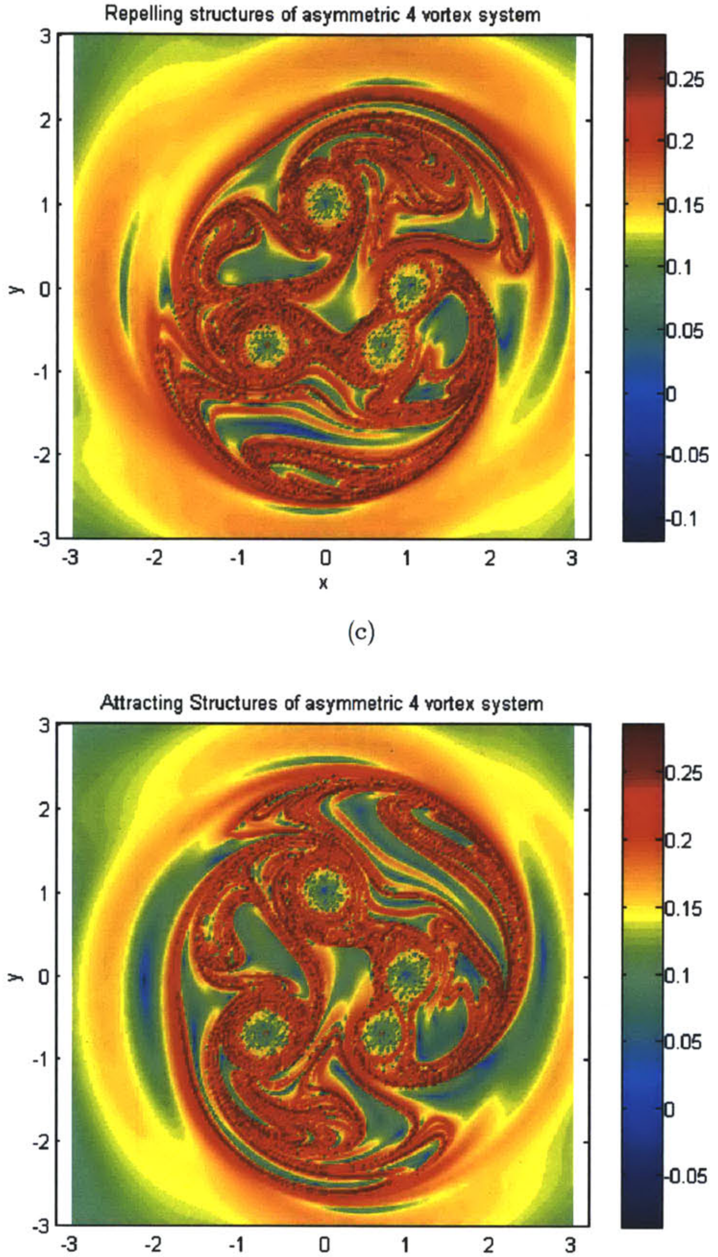
Figure 4–2: Lagrangian coherent structures of 3 vortex systems


Repelling Structures of symmetric 4 vortex system
3
2
1
y 0
-1
-2
-3
-3
-2
-1
0
1
2
3
x
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1

(a)


Attracting structures of symmetric 4 vortex system
3
2
1
y 0
-1
-2
-3
-3
-2
-1
0
1
2
3
x
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1

(b)



(c)

(d)

Figure 4–2: Lagrangian coherent structures of 4 vortex systems

Chapter 5

Numerical Results

5.1 Error measures

We use two different error measures to quantify the errors associated with the filtering process. In our vortex problem, these error measures show the effect of different initial tracer placement positions on the process of inference of the flow states through the particle filtering algorithm. Suppose that our state vector X_t has K components $\{x_t^k\}_{k=1}^K$. We note that for our problem the state vector $X = (\{\alpha\}_{i=1}^V, \{\beta\}_{i=1}^V, \{a\}_{j=1}^P, \{b\}_{j=1}^P)$, where (α, β) and (a, b) denote the two-dimensional coordinates of the V vortices and P tracers respectively. So, specifically we have $K = 2(V + P)$.

5.1.1 Median mean square Error

Firstly, from a filtering run we obtain the posterior distribution of the states $p(X_t|Y_{1:t})$ or $\{X_t^i, w_t^i\}_{n=1}^N$ at every time t. We evaluate the mean squared error (MSE) of this run as follows;

$$\begin{aligned}\text{MSE}_{t,1} =& \frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N}(x_{t,n,1}^k - x_{t,true,1}^k)^2 w_{t,n,1}\\ =& \frac{1}{K}\sum_{k=1}^{K}((x_{t,mean,1}^k - x_{t,true,1}^k)^2 + \sum_{n=1}^{N}(x_{t,n,1}^k - x_{t,mean,1}^k)^2 w_{t,n,1}) \qquad (5.1)\end{aligned}$$

Here $x^k_{t,true,1}$ is the true value of the k^{th} state component and $x^k_{t,mean,1}$ is the mean of the posterior of the k^{th} state for the first filtering run or the first realization of the filter. The subscript on x^k and w denotes time, particle number and realization number respectively. Now we repeat this process for several filtering runs; $m = 1, 2, .., M$ to obtain $\{\text{MSE}_{t,m}\}_{m=1}^{M}$. We now look to quantify the uncertainty in the simulated data (truth) and the observations using a median of the MSE over all the M runs of the filter. We choose to use median instead of mean after verifying that the median gives more robust results for our problem. The median mean squared error (MMSE) is given by the following formula.

$$\text{MMSE}_t = \text{median}_{m=1}^{M} \left(\frac{1}{K} \sum_{k=1}^{K} \sum_{n=1}^{N} (x^k_{t,n,m} - x^k_{t,true,m})^2 w_{t,n,m}\right) \tag{5.2}$$

Here $x^k_{t,true,m}$ is the true value of the k^{th} state component for the m^{th} filtering run or the m^{th} realization of the filter.

5.1.2 Root square error estimator - Chorin's Measure

In order to form this measure, we first consider a root squared error estimator (RSEE) defined for a single filter run as follows [18];

$$\text{RSEE}_{t,1} = \sqrt{\frac{1}{K} \sum_{k=1}^{K} (x^k_{t,mean,1} - x^k_{t,true,1})^2} \tag{5.3}$$

The RSEE for one run basically indicates the deviation of the mean of the posterior of the state from the truth at each time. Now we repeat this process for several runs of the filter $m = 1, 2, .., M$ to obtain independent samples of the error estimator $\{RSEE_{t,m}\}_{m=1}^{M}$. We now look at the mean and variance of the estimator to give us a sense of how "good" this estimator is on average and also with what certainty it achieves this level of "goodness".

$$\text{mean}\,(\text{RSEE}_{t,m})_t = \frac{1}{M} \sum_{m=1}^{M} \sqrt{\frac{1}{K} \sum_{k=1}^{K} (x^k_{t,mean,m} - x^k_{t,true,m})^2} \tag{5.4}$$

$$\text{variance}\,(\text{RSEE}_{t,m})_t = \frac{1}{M}\sum_{m=1}^{M}(\text{RSEE}_{t,m} - \text{mean}\,(\text{RSEE}_{t,m})_t)^2 \tag{5.5}$$

Note: For both these measures we need not use all the state component i.e. $k = 1, 2, .., K$ to find out the error measures. For example, in our problem we use two kinds of error estimates, one for the vortices and one for the tracers. However, it makes more sense to evaluate the error estimates of the vortices because the tracer positions are functions of the vortex positions and the vortex model is defined by the positions of the vortices.

5.2 Performance of the Sequential Importance Resampling (SIR) Algorithm

Now, we study the performance of the Sequential Importance Resampling (SIR) filtering algorithm in tracking the two-dimensional flow. There are several parameters which effect the performance of the algorithm. We carry out the following studies in particular :

1. Effect of starting priors of the vortex positions on the estimation errors.
2. Effect of the number of particles used on the estimation errors.
3. Effect of the tracer release position on the estimation errors.

5.2.1 Starting priors on vortices

We explore the effect of starting priors on the estimation in case of an asymmetric three vortex system with tracer released at (0,0) and 100 particles. We consider gaussian priors with mean at true values and varying variances.

We use 30 filter runs for our RMSE calculations. From Figure 5-1 we conclude that a prior with variance of 0.01 is reasonable for the purpose of estimation. We use this prior

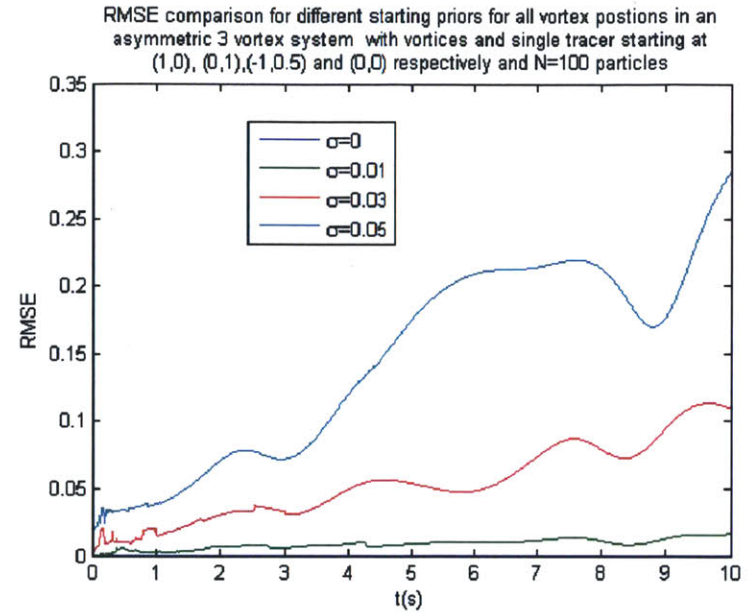


Figure 5-1: Effect of starting prior on vortices on the estimation process

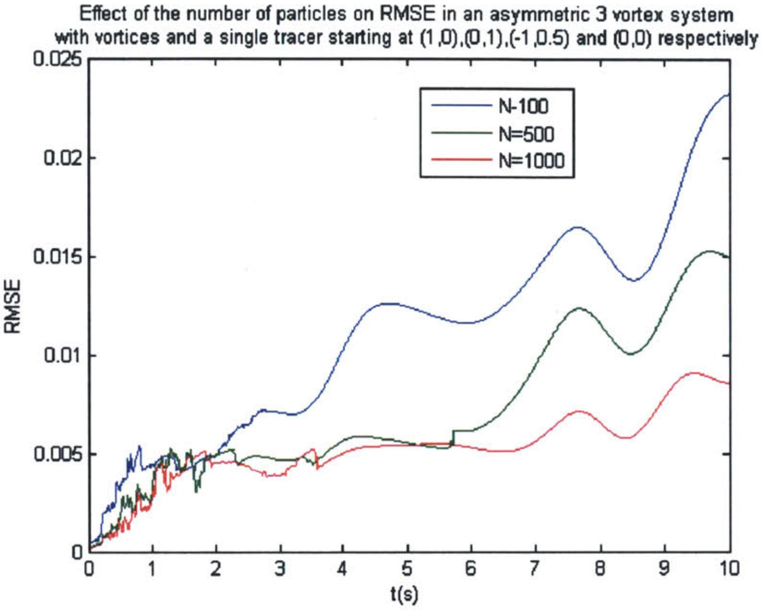
variance for our further studies.

5.2.2 Number of particles used in filtering

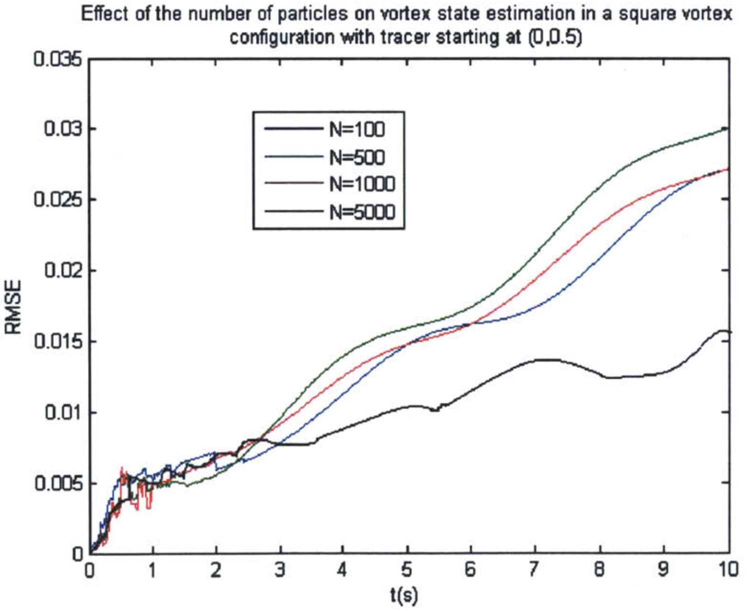
Figure 5-1 shows the effect of number of particles used on the estimation performance. We clearly see that as we increase the number of particles the estimation quality improves. This is an inherent problem with particle filtering. For this problem in particular, to get a very good estimate of the states, one has to use particle sizes of the order of 10000 particles. We use 5000 particles for most of our studies in the later sections.

5.2.3 Tracer release position

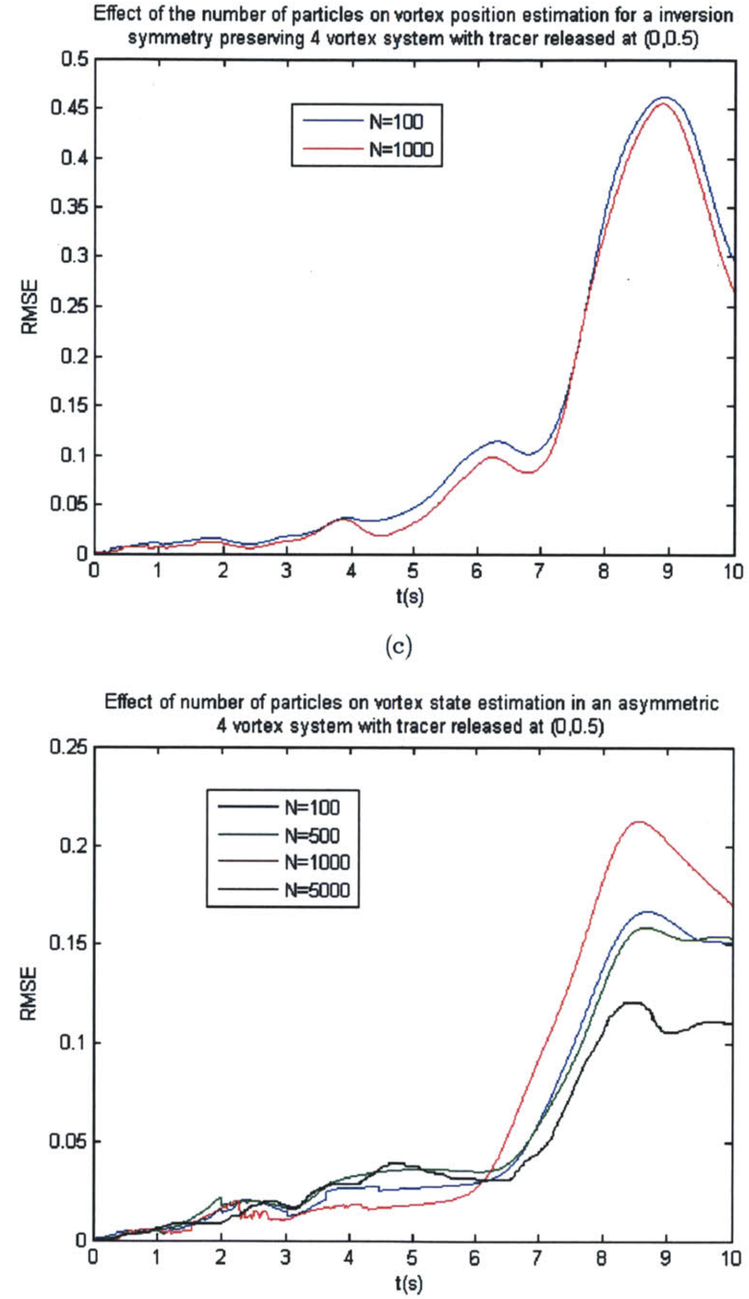
Figure 5-1 shows the effect of tracer release position on the estimation performance. We release the tracers at two different positions in the two cases, one very close to the vortex


Effect of the number of particles on RMSE in an asymmetric 3 vortex system
with vortices and a single tracer starting at (1,0),(0,1),(-1,0.5) and (0,0) respectively
N-100
N=500
N=1000
RMSE
0.025
0.02
0.015
0.01
0.005
0
0
1
2
3
4
5
6
7
8
9
10
t(s)

(a)


Effect of the number of particles on vortex state estimation in a square vortex
configuration with tracer starting at (0,0.5)
N=100
N=500
N=1000
N=5000
RMSE
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
1
2
3
4
5
6
7
8
9
10
t(s)

(b)



(c)

(d)

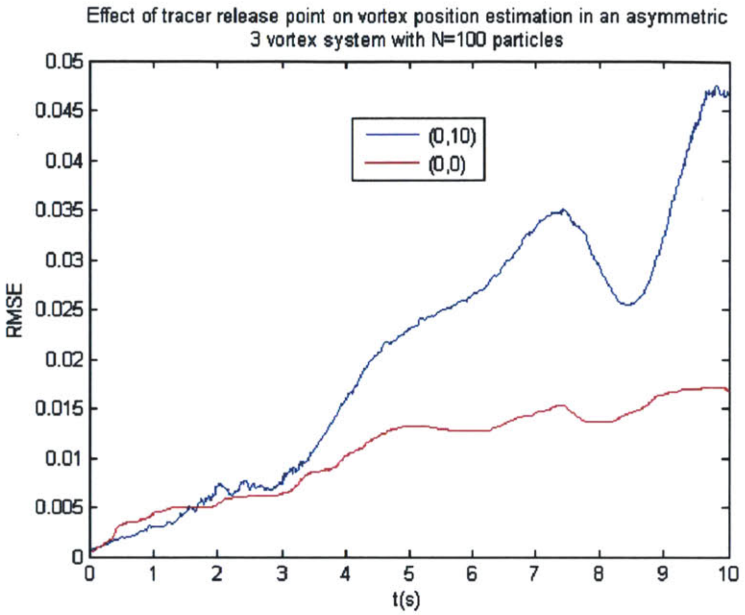
Figure 5-1: Effect of the number of particles used on the estimation errors for different configurations

locations at (0,0.5) and the other very far away from the vortices at (0,10). We expect the data from the closely placed tracer to be more relevant as seen clearly in all the cases we have considered. However, as the flow is chaotic as well, a more extensively study needs to be done to find out the locations where the tracer placement gives the best results. This strongly motivates the us towards experimental design in terms of optimal tracer placement. We look at this in detail in a later section. In this study, we have been using just a single tracer. We show the vortex tracking in case of an asymmetric four vortex system in fig. 5-2. We can clearly see that the estimated state loses track of the true state after some time. This clearly motivates us to look at the tracking of chaotic systems in detail.

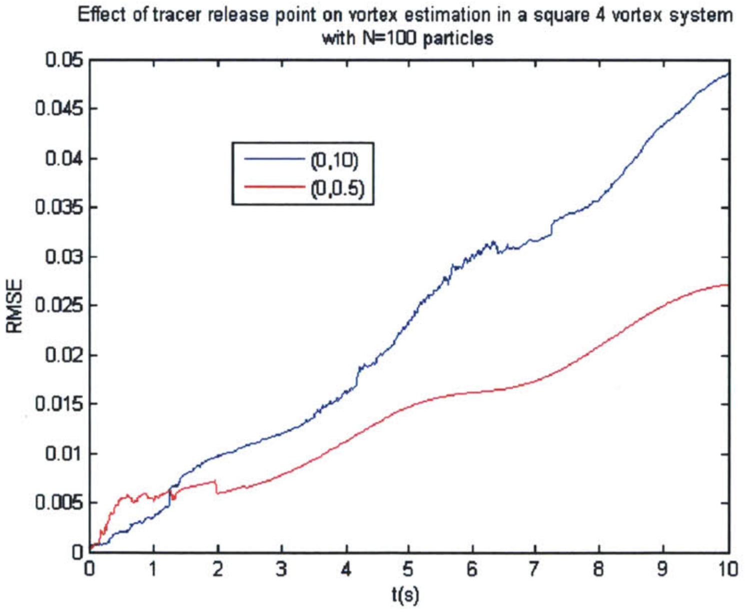
5.2.4 Tracking Chaotic Flow

We can possibly think of two reasons as to why the estimated state loses track of the true state in the chaotic 4 vortex system. Firstly, upon observing the propagation of the particles of the tracer over time, we notice that after a certain period of time all the particles cluster around each other and are sufficiently not thrown apart. This happens because the process noise is much lesser in comparison to the observed noise. We step up the process noise to a value comparable to the observation noise (only while filtering and not in data simulation). Figure 5-3 shows the tracking of both vortex and tracer with the adjusted process noise. We again see that the estimated vortex state loses track of the true vortex position, although the filtered estimate of the tracer is still able to track the true position of the tracer.

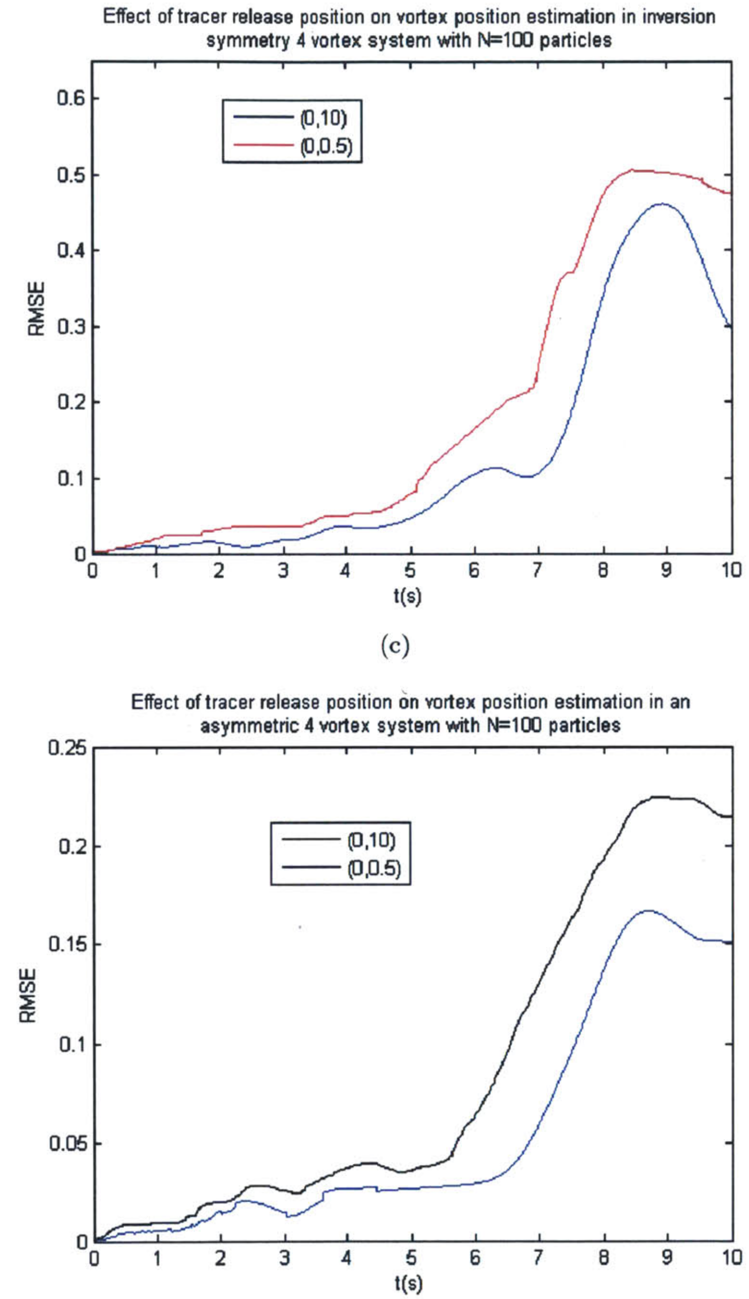
This clearly shows that using just a single tracer we may not be able to extract sufficient information about the dynamics (which is chaotic in this case) of the flow field and hence the position of the vortices. In fig. 5-4 we plot the tracer state estimation corresponding to fig. 5-2. We clearly see that the though the tracer is continuously tracked the vortex isn't. So, to track chaotic systems, we need to introduce a large number of tracers and also introduce them at appropriate positions.


Effect of tracer release point on vortex position estimation in an asymmetric 3 vortex system with N=100 particles
(0,10)
(0,0)
RMSE
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.045
0.05
0
1
2
3
4
5
6
7
8
9
10
t(s)

(a)


Effect of tracer release point on vortex estimation in a square 4 vortex system with N=100 particles
(0,10)
(0,0.5)
RMSE
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.045
0.05
0
1
2
3
4
5
6
7
8
9
10
t(s)

(b)



(c)

(d)

Figure 5-1: Effect of the tracer release position on the estimation errors

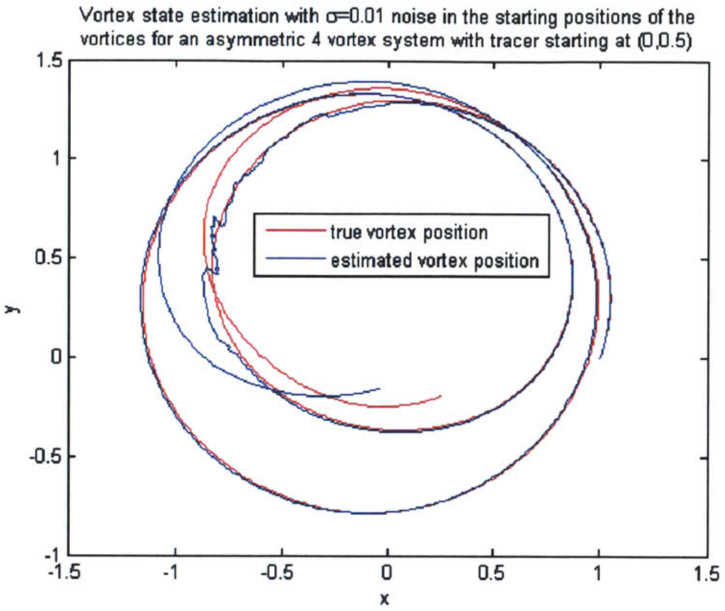
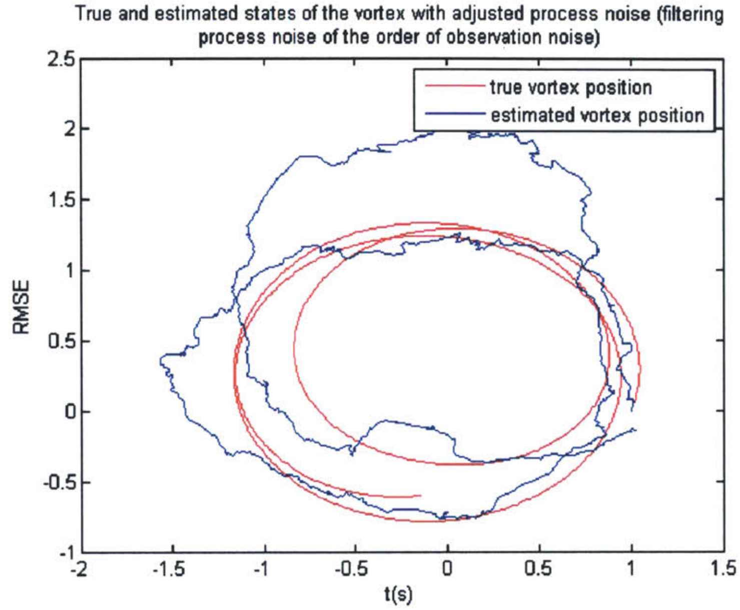


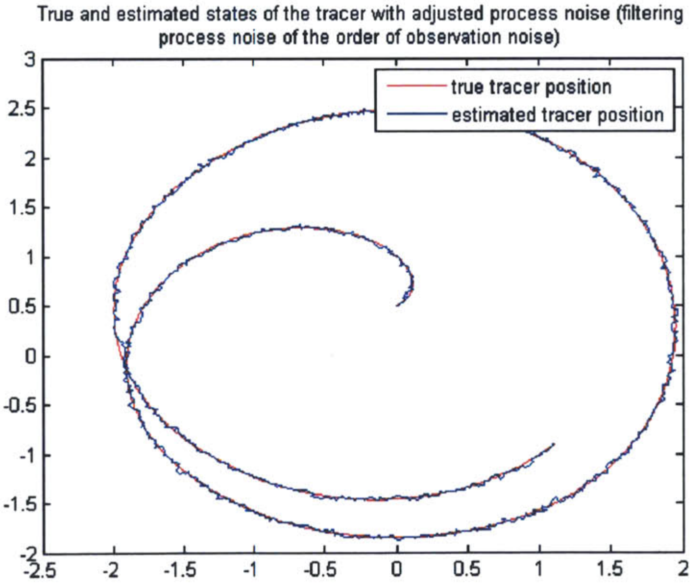
Figure 5-2: Vortex state estimation for an asymmetric 4 vortex system

5.3 Comparison of auxiliary particle filter (APF) and sequential importance resampling (SIR)

Here, we compare the performance of the APF and the SIR for an asymmetric four vortex system with tracer released at (0,0.5) and using 1000 particles in fig. 5-5. The APF performs better than the SIR and hence is a very good alternative to the SIR, especially in problems where the data is more valuable than the prior. The reason why the APF performs better than the SIR is that the APF samples the particles high probability regions in the posterior. Hence, we should consider APF to be a good alternative to the SIR. However, as the APF resamples at every step, it can lead to sample impoverishment quicker than the SIR. This can especially be a problem if the estimated state loses track of the true state as in chaotic systems.



(a)



(b)

Figure 5-3: Vortex and tracer tracking with adjusted process noise

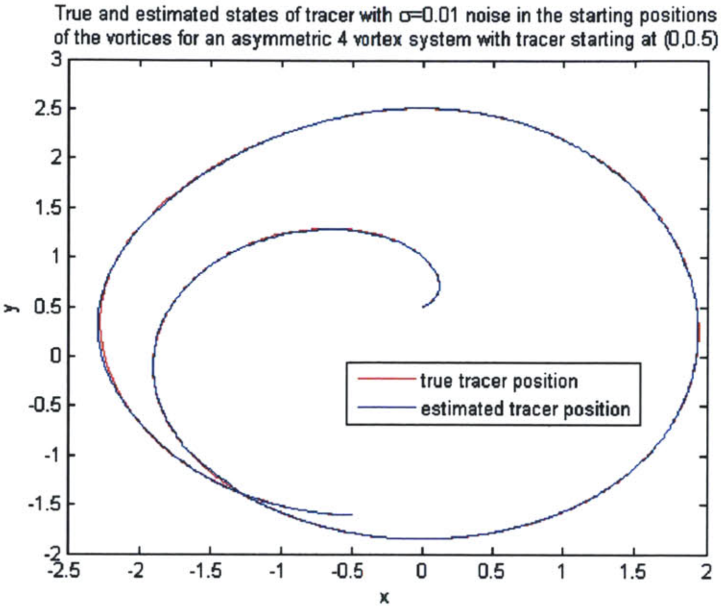


Figure 5-4: Tracer state estimation for asymmetric 4 vortex system

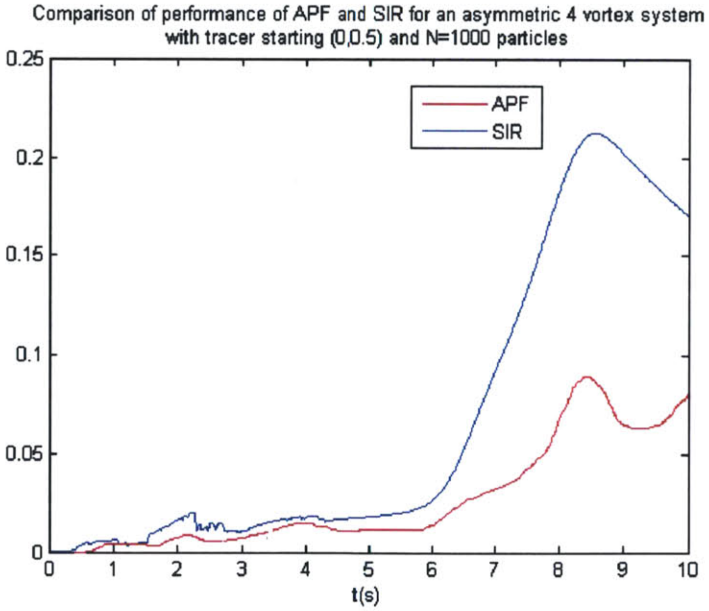


Figure 5-5: Comparison of the APF and SIR for asymmetric 4 vortex system

5.4 Effect of Tracer release positions on the estimation process

We now move our attention to the effect of tracer release positions on the estimation process. In our preliminary study about this in the last section, we figured that tracers released close to the vortex positions lead to better estimation than ones released far away. An intuitive explanation for this is that when tracers are placed close to the vortices, they directly come under their influence of the fields generated by the vortices, and their motion is thereby influenced by these fields. This, in turn, results in them capturing better information about the movement of the vortices.

We also learned earlier that some of the systems that we are looking at are chaotic and irregular. Also, finite time lyapunov exponent plots can be effectively used to study chaotic systems. This motivates us to build up a link between the tracer placement problem and the corresponding FTLE plots of the system. We divide our study into several cases arising from different vortex configurations.

5.4.1 Symmetric 3 vortex system

First we look at the symmetric 3 vortex system. Figures 4–2 a,b represent the attracting and repelling structures of the symmetric 3 vortex system. As the system is symmetric, the attracting and repelling features turn out to be very similar. Based on these coherent structures, we define regions 1, 2a, 2b, 2c, 3a, 3b, 3c and 4, as shown in fig. 5-6, where we intend to test out tracer placement problem.

We consider the following tracer release positions:

1. Single tracer release positions: 1 $(-4,0)$, 2a $(-0.5,0.5\sqrt{3})$, 2c $(1,0)$, 3a $(-1.5,0)$, 3b $(0.75,0.75\sqrt{3})$, 4 $(0,0)$.

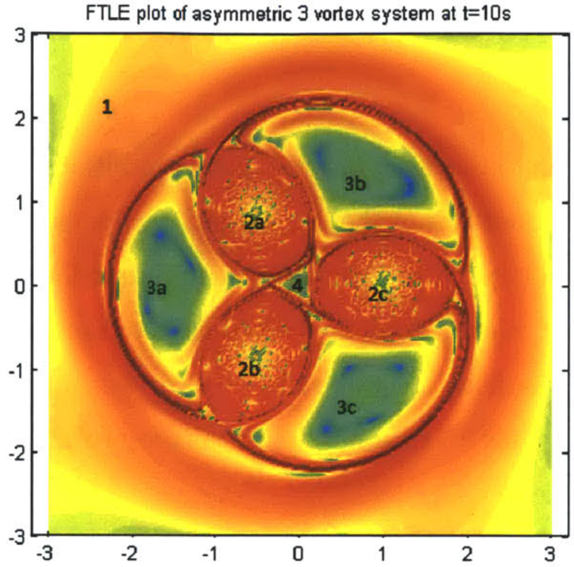


Figure 5-6: Different regions in the symmetric 3 vortex system based on Lagrangian coherent structures

2. Double tracer release positions: 2a-2a1, 2a-2c, 2a-3a, 2a-3b, 2c-3a, 3a-3a1, 3a-3b.

We note that the different tracer release positions in region 2 and 3 are symmetric about the origin. We also note that 2a1 and 3a1 are tracers released very close to 2a and 3a and their coordinates are given by $(-0.6, 0.5\sqrt{3})$ and $(-1.6, 0)$. We now apply the SIR filter with $N = 5000$ particles to carry out inference on the symmetric 3 vortex system. Based on the two different error measures we had developed, i.e. MMSE and RSEE, we make the following observations.

We make the following observations from plots of the Median Mean Square Error (MMSE) of the vortices for the various tracer release positions.

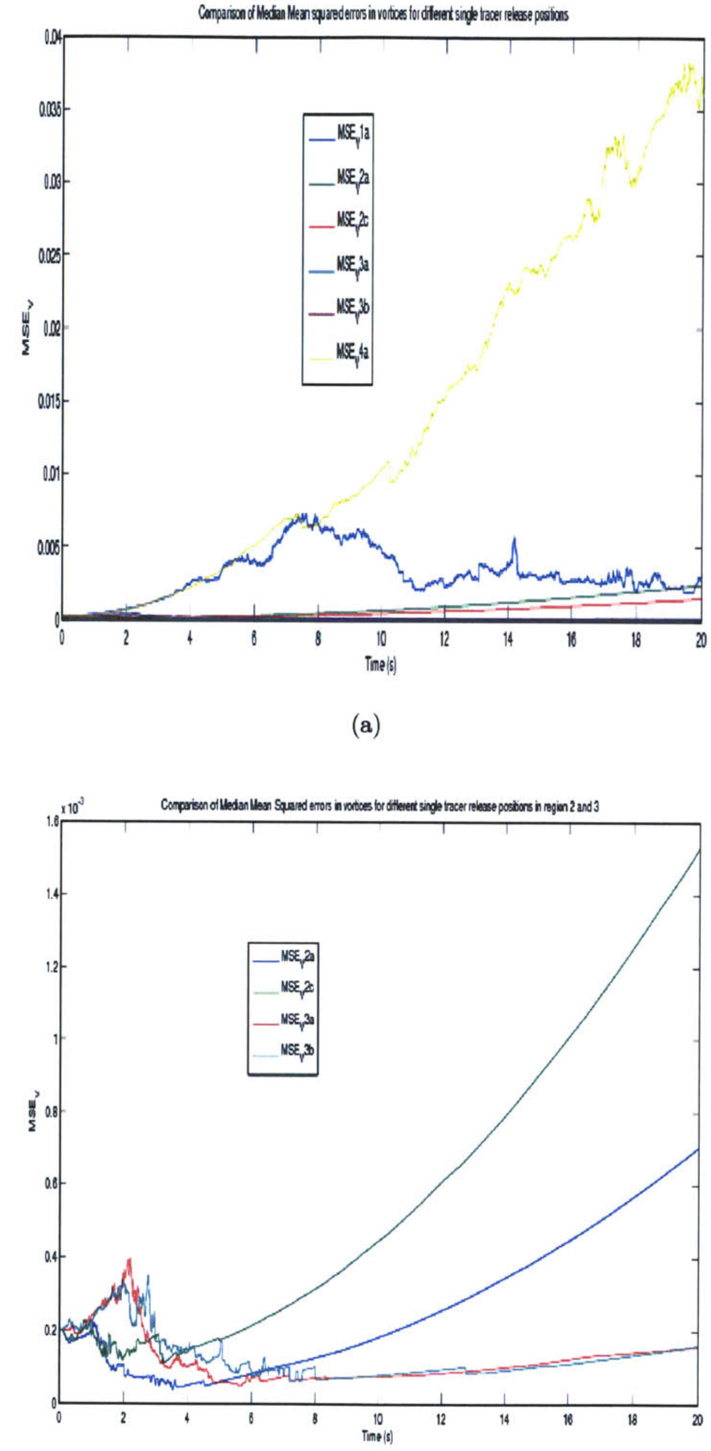
1. Single tracers released in regions 1 and 4 perform significantly worse than region 2 and 3. This can be clearly observed in fig. 5-5a. The repelling structures separating region 1 from region 2 and 3 indicated that the particles on either side of the structure do not interact. This is in fact confirmed by a physical simulation of the system. In a similar

way, we think that region 4 does not interact much with region 2 and 3. However, there are few particles from region 4 which escape into region 2 and 3 as we mover forward in time. Both these suggest that more complicated flow of the vortices can be captured better by placing tracers in regions 2 and 3 than by tracers released in region 1 or 4 as tracers in region 1 and 4 do not interact much with regions 2 or 3 and hence cannot infer the movement of vortices in region 2 well.

2. Single tracer in region 2 performs worse than region 3. This is evident from fig. 5-5b. We observe that the MMSE of vortices with a tracer released in region 2 is an order of magnitude higher than when it is released in region 3.Also, the magnitude of FTLE in region 2 is on an average more than that in region 3. Further, the boundaries (repelling structures) of region 2 are far more diffused than those of region 4. These suggest that the flow in region 2 is more irregular than in region 3 and hence tracers from region 2 are less useful in inferring about the locations of vortices in the system.

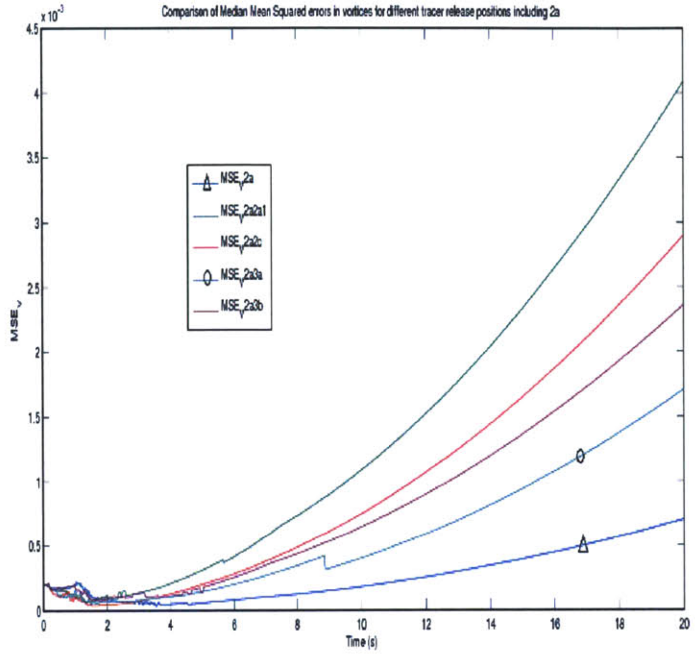
3. Figure 5-5b shows that symmetrically placed tracers in region 3 perform almost the same. As the 3 vortex system we study is symmetric about the origin and so are the tracers released in different parts of region 3, we can expect them to lead to similar observations and also similar inference results. Further, as the MMSE of the vortices is of significantly low order of magnitude, the numerical integration error does not seem to affect the results for the symmetrically placed tracer release positions in region 3 to be different.

4. Symmetrically placed tracers in region 2 do not perform the same. This is observed in fig. 5-5b, where symmetrically placed tracers in regions 2a and 2c lead to very different errors. Like in the previous case, we would expect tracers released symmetrically located positions in region 2 to lead to similar inference results. However, this is not something that we observe. We think that there are two factors that work in conjunction to give rise to this result. Firstly, the inference results are very sensitive to tracer release positions. This result if further justified by one of the later observations.

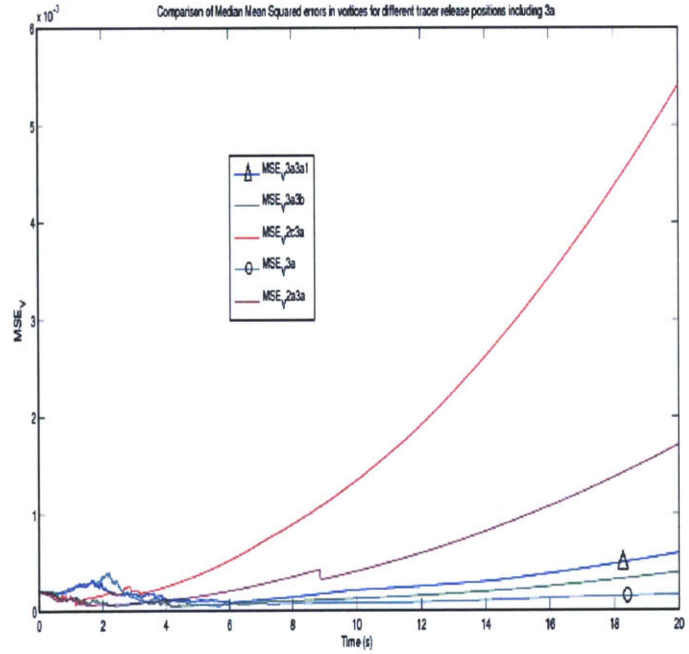

Comparison of Median Mean squared errors in vortices for different single tracer release positions
MSE$_v$
Time (s)
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0
2
4
6
8
10
12
14
16
18
20
MSE$_v$1a
MSE$_v$2a
MSE$_v$2c
MSE$_v$3a
MSE$_v$3b
MSE$_v$4a
Comparison of Median Mean Squared errors in vortices for different single tracer release positions in region 2 and 3
x 10^{-3}
MSE$_v$
Time (s)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
0
2
4
6
8
10
12
14
16
18
20
MSE$_v$2a
MSE$_v$2c
MSE$_v$3a
MSE$_v$3b

(a)

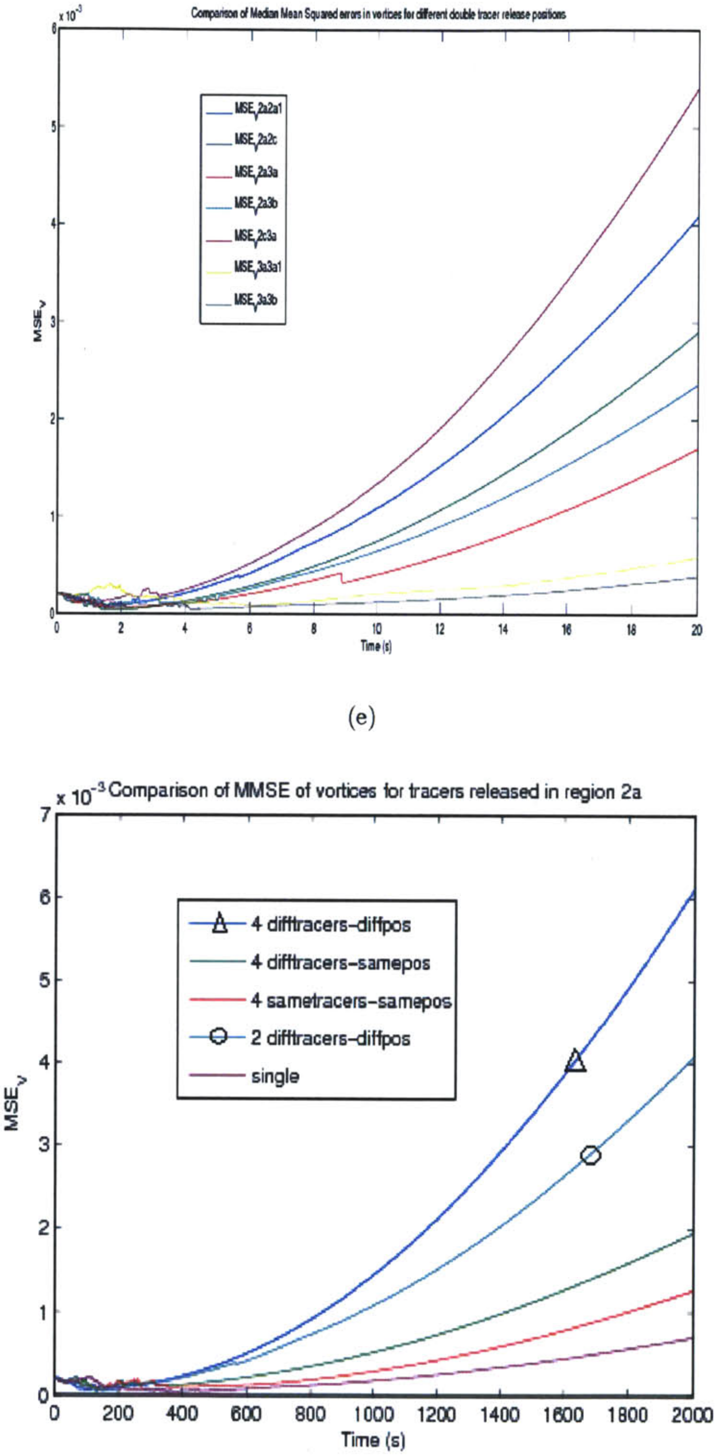
(b)


Comparison of Median Mean Squared errors in vortices for different tracer release positions including 2a
x 10^{-3}
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
MSE_V
MSE_V2a
MSE_V2a2a1
MSE_V2a2c
MSE_V2a3a
MSE_V2a3b
0
2
4
6
8
10
12
14
16
18
20
Time (s)

(c)


Comparison of Median Mean Squared errors in vortices for different tracer release positions including 3a
x 10^{-3}
6
5
4
3
2
1
0
MSE_V
MSE_V3a3a1
MSE_V3a3b
MSE_V2c3a
MSE_V3a
MSE_V2a3a
0
2
4
6
8
10
12
14
16
18
20
Time (s)

(d)



(e)

(f)

Figure 5-5: Comparison of Median Mean Squared Error (MMSE) for different tracer release position in a symmetric 3 vortex system

Secondly, due to the more irregular flow in region 2 when compared to region 3, the error due to numerical integration has more pronounced effects in region 2. This fact is corroborated by a separate study of numerical integration error on tracers in region 2 and 3. We make a short digression here to talk about the performance and errors associated with the stochastic RK-4 integration scheme that we have to using in our problem.

Stochastic RK-4 Numerical Integration Scheme

We learnt earlier that the stochastic RK-4 scheme is second order accurate in time. We prove this by looking at the convergence plot for the scheme. Figure 5-6 shows a loglog convergence plot of the absolute integral average error over time and over all the three vortices in a symmetric 3 vortex model. We consider a time step of $\Delta t = 10^{-4}$s to be the true state, while we evaluate time steps from $\Delta t = 10^{-3}$s to $\Delta t = 10^{-2}$s in the plot. We clearly see that the slope of the curve is close to 2 which corroborates the fact that the stochastic RK-4 scheme is second order accurate in time.

Figure 5-7 shows the effect of numerical integration error on tracers in region 2 and 3 for two different stochastic integration schemes labeled as "good" ($\Delta t = 10^{-3}$s) and "bad" ($\Delta t = 10^{-2}$s) schemes. The benchmark used is an even stronger scheme with $\Delta t = 10^{-4}$s. This proves that numerical integration error is more for tracers in region 2, making inference using tracers in region 2 harder.

5. Figure 5-5c shows that two closely spaced tracers in region 2 does worse than single in region 2. This is by far the most perplexing result. As we add new observations from an additional tracer at every instant in time, we should hope to get better state estimates than those obtained by using a single tracers. One way to rationalize this peculiar behavior would be to claim that additional tracers can lead to conflicting observations which in turn can lead to bad inference results. By conflicting, we mean that when observations from one tracer leads to a high likelihood, then the observation from the other leads to a low likelihood. Also, these additional observations can turn

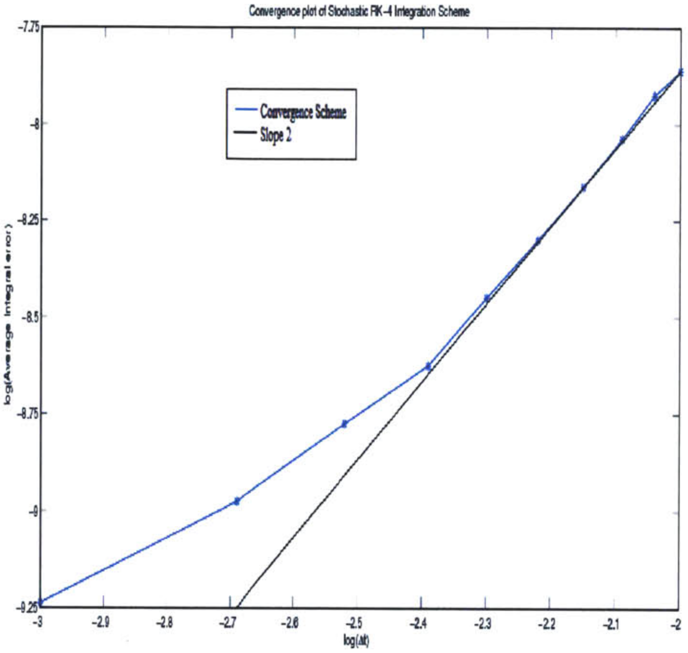
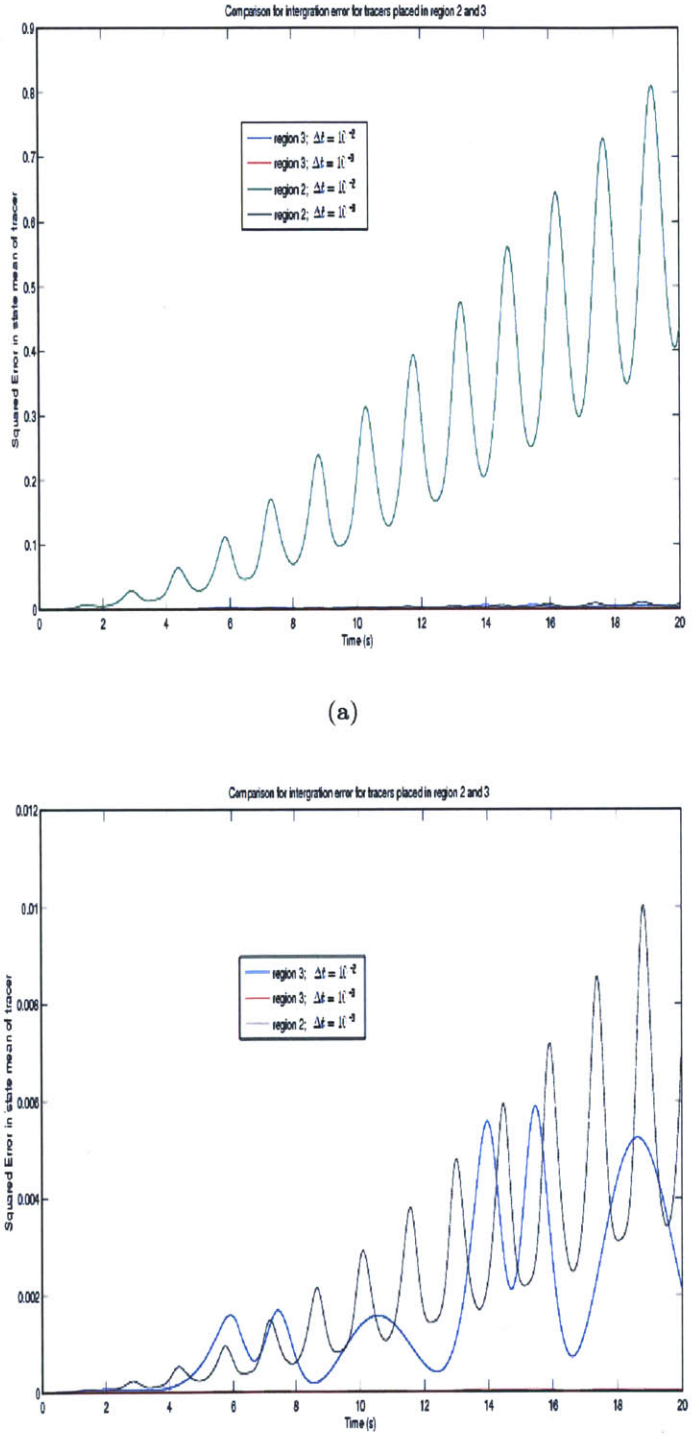


Figure 5-6: Convergence plot of Stochastic RK-4 Numerical Integration scheme when applied to symmetric 3 vortex system



(a)

(b)

Figure 5-7: Comparison of Numerical Integration Error on tracers in region 2 and 3

out to be conflicting as inference results using a single tracer is highly dependent on the start position. However, we are not entirely satisfied with this result and find it an interesting idea to be explored in further studies.

6. Two closely spaced tracers in region 3 does worse than single in region 3.This is clearly seen in fig. 5-5d and be justified by the same reasons as the case of region 2.

7. In fig. 5-5e we see that four tracers starting from same location in region 2 performs almost the same as single tracer starting at that location. Further, it performs much better than 4 tracers starting from closely spaced points. This observation suggests that the inference results are highly dependent of the start location of the tracer as using 4 tracer observations starting at the same point leads to much better results than 4 tracer observations starting at closely spaced points. Also, using this result we verify that adding additional tracers at the same point leads to almost same inference results as single tracer released at that point. Although, one might expect the results to get a little better but definitely nor worse.

8. Again, in fig. 5-5f we notice that four tracers starting from same location in region 3 performs almost the same as single tracer starting at that location. Further, it performs much better than 4 tracers starting from closely spaced points. This observation can be explained just as in case of region 2. Additionally, in this case adding 4 tracer observations starting from the same point actually improves the inference results. This totally agrees with our intuition that additional observations should improve an inference process.

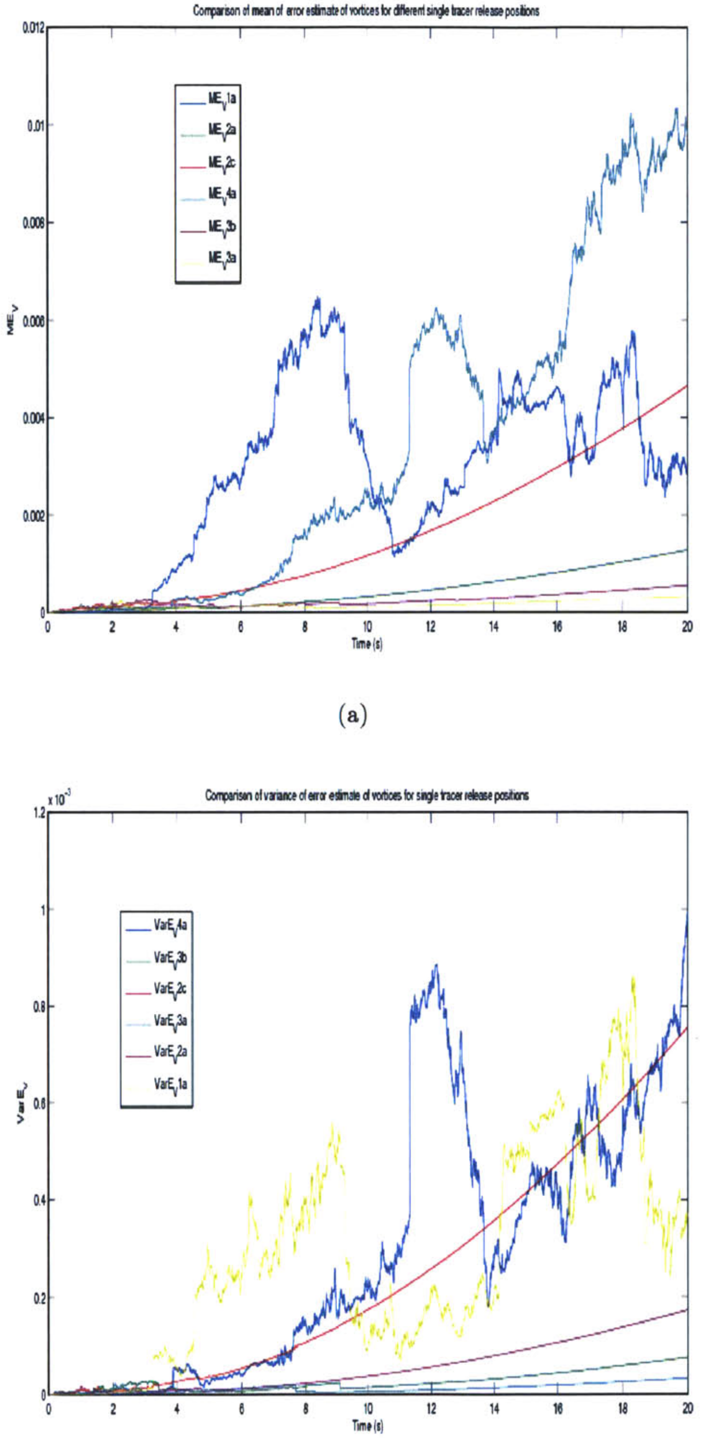
We now study plots of mean and variance of the Root Square Error Estimator (RSEE) of the vortices for the various tracer release positions. Figures 5-6 a,b shows that single tracers in regions 1 and 4 perform significantly worse than region 2 and 3. Single tracer in region 2 performs worse than region 3. While symmetrically placed tracers in region 3 perform almost the same, symmetrically placed tracers in region 2 do not. Figures 5-6 c,d show that two closely spaced tracers in region 2 does worse than single in region 2. Infact single tracer in region 2 performs better than any combination of that tracer with any other tracer in region 2 or 3. In figs. 5-6 e,f we see that two closely spaced tracers in region 3 does worse than single in region 3. All these observations can be explained in the same way as in the case of the Median Mean Squared Error. Additional, we note that the variance of the RSEE gives us additional information. The variance of RSEE for tracers placed in region 2 is much more than tracers in region 3, i.e. not only is the error of estimation lower in region 3 than in region 2, but also that this error is lower on an average in region 3 than in region 2. That being said, there can be cases when observations collected from a tracer released in region 2 can lead to better results than a tracer released in region 3. Hence, in a practical situation we should ideally release multiple tracers in every region, even if we know that releasing tracers in one region can lead to better inference.

5.4.2 Asymmetric 3 vortex system

We now study the performance of the SIR filtering algorithms under different tracer release positions for a asymmetric 3 vortex system. In particular, we consider a system with $ecc = 0.3$, which can be considered to be moderately asymmetric. Figures 4–2 c,d represent the attracting and repelling structures of the symmetric 3 vortex system. Based on the common feature in the attracting and repelling coherent structures, we define the following regions 1, 2a, 2b, 2c, 3a, 3b and 3c as shown in fig. 5-7, where we intend to test out tracer placement problem.

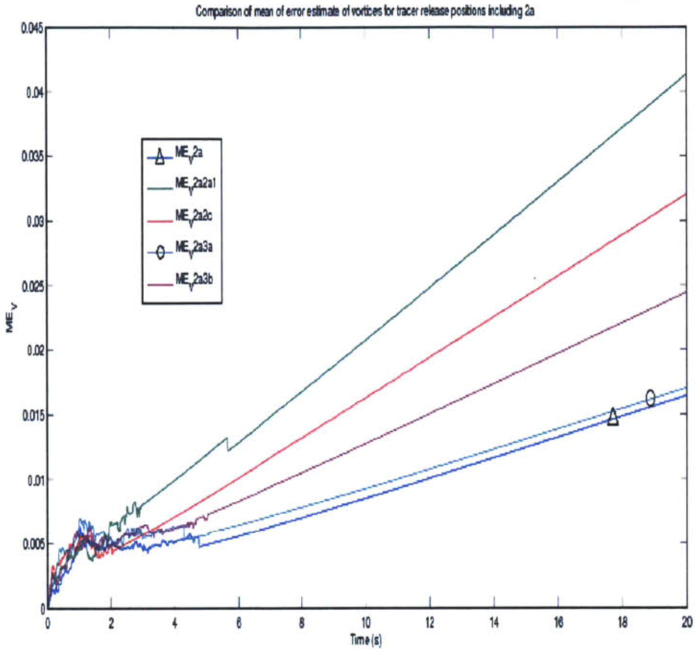
We consider the following tracer release positions:

1. Single tracer release positions: 1 $(-4, 0)$, 2a $(-0.5, 1)$, 2b $(-0.5, -1)$, 2c $(1, 0.3)$, 3a $(-1.5, 0)$, 3b $(1, 1.5)$, 4 $(0.8, -1)$.
2. Double tracer release positions: 2a-2a1, 2a-2b, 2a-2c, 2a-3a, 2a-3b, 2a-3c, 2b-2c, 3a-3a1, 3a-3b, 3a-3c, 3a-2b, 3a-2c, 3b-3c.

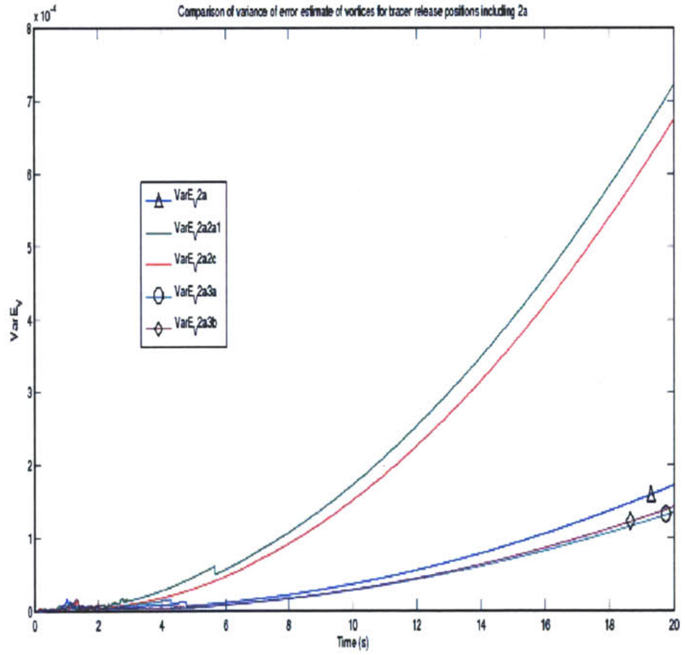

Comparison of mean of error estimate of vortices for different single tracer release positions
ME_V
Time (s)
0
0.002
0.004
0.006
0.008
0.01
0.012
0
2
4
6
8
10
12
14
16
18
20
$ME_V 1a$
$ME_V 2a$
$ME_V 2c$
$ME_V 4a$
$ME_V 3b$
$ME_V 3a$
Comparison of variance of error estimate of vortices for single tracer release positions
$\times 10^{-3}$
Var E_V
Time (s)
0
0.2
0.4
0.6
0.8
1
1.2
0
2
4
6
8
10
12
14
16
18
20
$VarE_V 4a$
$VarE_V 3b$
$VarE_V 2c$
$VarE_V 3a$
$VarE_V 2a$
$VarE_V 1a$

(a)

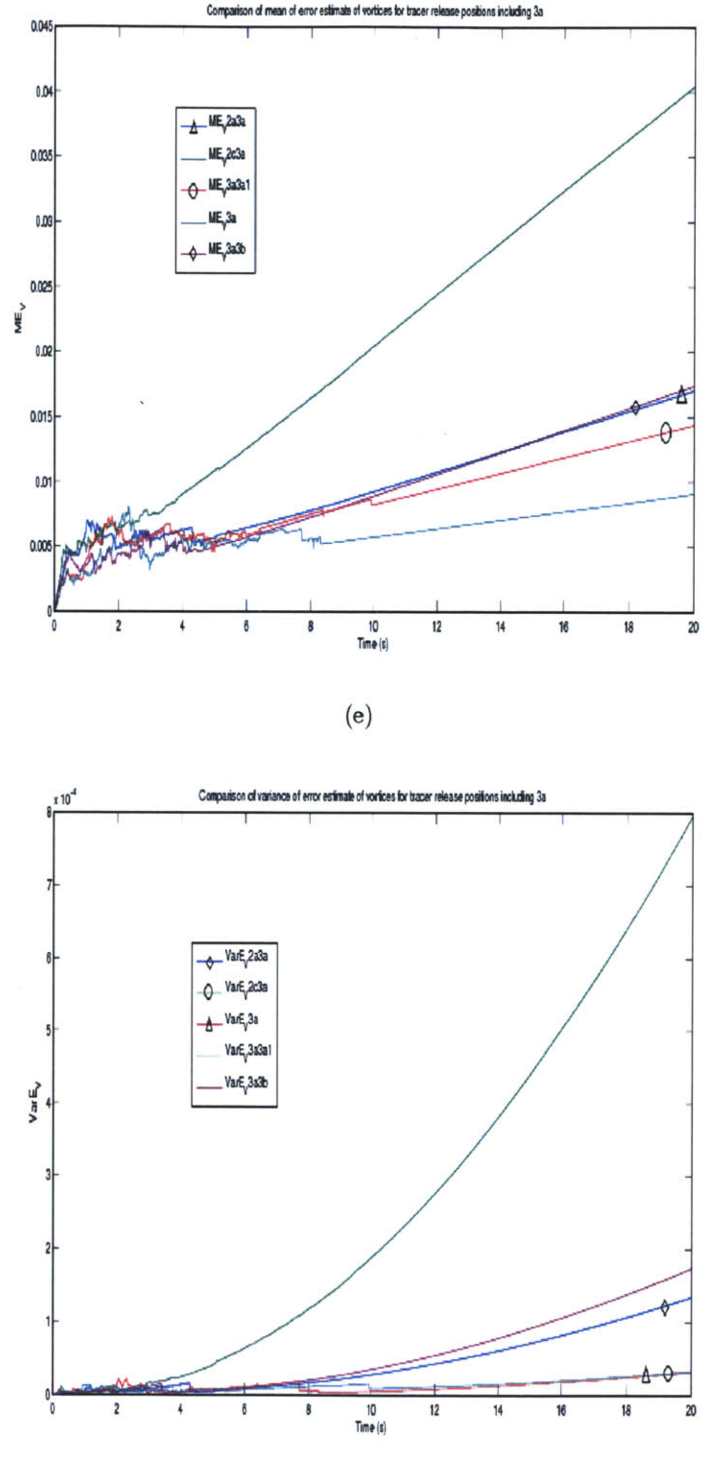
(b)


Comparison of mean of error estimate of vortices for tracer release positions including 2a
ME_V
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.045
Time (s)
0
2
4
6
8
10
12
14
16
18
20
ME_V2a
ME_V2a2a1
ME_V2a2c
ME_V2a3a
ME_V2a3b

(c)


Comparison of variance of error estimate of vortices for tracer release positions including 2a
x 10^-4
VarE_V
0
1
2
3
4
5
6
7
8
Time (s)
0
2
4
6
8
10
12
14
16
18
20
VarE_V2a
VarE_V2a2a1
VarE_V2a2c
VarE_V2a3a
VarE_V2a3b

(d)



(e)

(f)

Figure 5-6: Comparison of mean and variance of RSEE for different tracer release position in a symmetric 3 vortex system

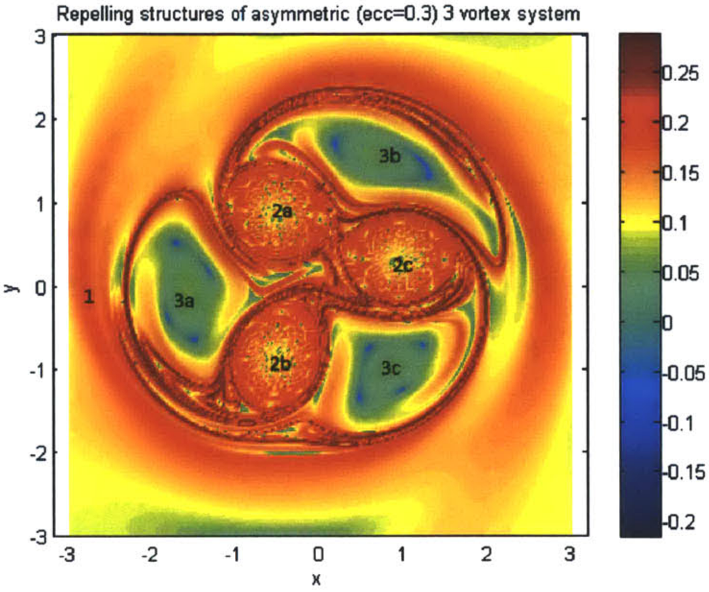


Figure 5-7: Different regions in the asymmetric 3 vortex system based on Lagrangian coherent structures

Here the different tracer release positions in region 2 and 3 are not symmetric about the origin. We also note that 2a1 and 3a1 are tracers released very close to 2a and 3a and their coordinates are given by $(-0.45, 1)$ and $(-1.55, 0)$. We now apply the SIR filter with $N = 5000$ particles to carry out inference on the asymmetric 3 vortex system.

We study the MMSE plots for these different tracer release positions and combinations. Clearly, from fig. 5-8a, we can say that tracers released in region 1 are not valuable to the inference process. The reason for this is that tracers in region 1 do not interact sufficiently well with the fields created by the vortices and are hence cannot capture valuable information about the positions of the vortices. In fig. 5-8b, we notice that the MMSE of tracers released in region 2 and 3 are fairly different from that in the symmetric 3 vortex case. Tracers in 2b and 2c perform significantly worse than the other tracers. However, tracer in region 2a perform very well. In all likelihood, the tracer released in 2a looks like an outlier. As we mentioned earlier, the estimation process is very sensitive to initial positions, and the variance of RSEE in region 2a is much higher than in region 3, it is very much possible that we got lucky in terms of tracer 2a. That being said, it is important to note that the difference in errors between tracers released in region 2 and region in the asymmetric 3 vortex system is much lesser than in the case of symmetric 3 vortex systems. We can further hypothesize that as we increase the asymmetricity of the 3 vortex system, the difference between region 2 and 3 becomes lesser as region 3 tends to have more irregular flow. Figures 5-8 c,d show that single tracers in region 2 and 3 performs significantly better than combinations of multiple tracers respectively.

5.4.3 Symmetric 4 vortex system

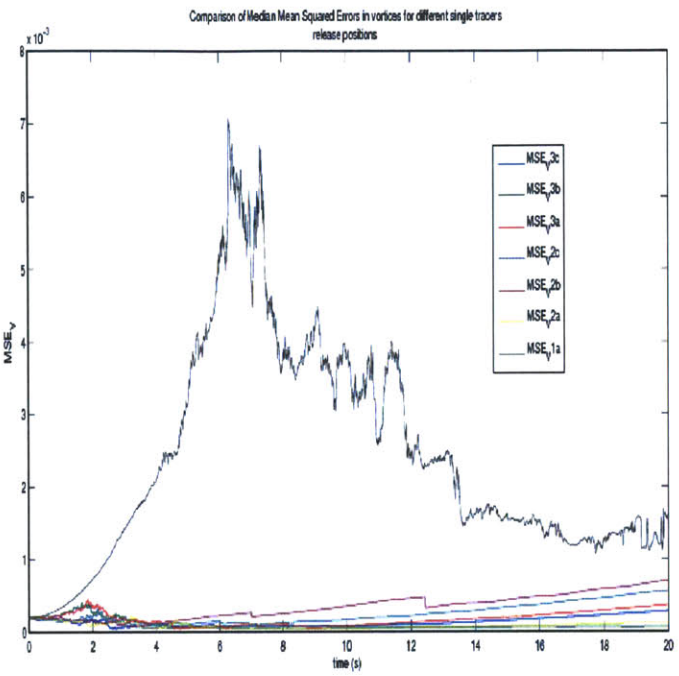
We now study the performance of the SIR filtering algorithms under different tracer release positions for a symmetric 4 vortex system. Figures 4–2 a,b represent the attracting and repelling structures of the symmetric 4 vortex system. Both the attracting and repelling coherent structures are very similar and based on them, we define the following regions 1, 2a, 2b, 2c, 3a, 3b and 3c as shown in fig. 5-9, where we intend to test out tracer placement problem.

We consider the following tracer release positions:

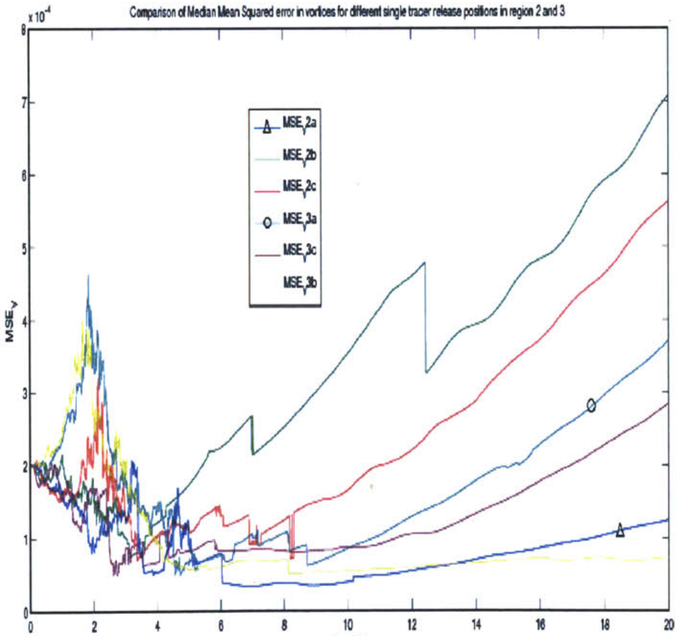
1. Single tracer release positions: 1 $(-4,0)$, 2a $(-1.05,0)$, 2b $(0,1.05)$, 2c $(1.05,0)$, 2d $(0,-1.05)$, 3a $(-1.05,1.05)$, 3b $(1.05,1.05)$, 3c $(1.05,-1.05)$, 3d $(-1.05,-1.05)$, 4 $(0,0)$.

2. Double tracer release positions: 2a-2a1, 2a-2b, 2a-2c, 2a-3a, 2a-3b, 3a-3a1, 3a-3b, 3a-3c, 2a-2asame, 3a-3asame.

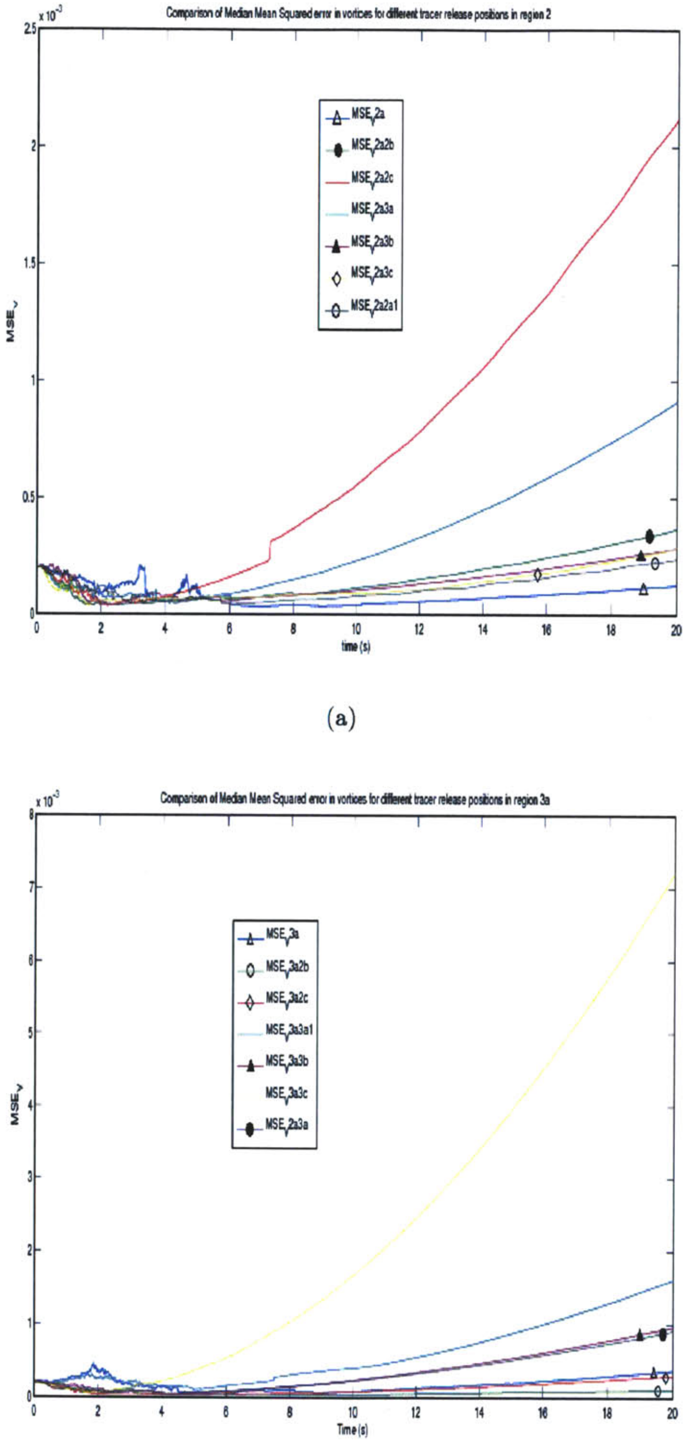
Here the different tracer release positions in region 2 and 3 are not symmetric about the origin. We also note that 2a1 and 3a1 are tracers released very close to 2a and 3a and their coordinates are given by $(-1.1,0)$ and $(-1.1,1.05)$. Also, 2a-2asame and 3a-3asame denote two tracers released from the same position in region 2a and 3a respectively. We now apply


Comparison of Median Mean Squared Errors in vortices for different single tracers
release positions
x 10^-3
8
7
6
5
4
3
2
1
0
MSE_V
0
2
4
6
8
10
12
14
16
18
20
time (s)
MSE_V3c
MSE_V3b
MSE_V3a
MSE_V2c
MSE_V2b
MSE_V2a
MSE_V1a

(a)


Comparison of Median Mean Squared error in vortices for different single tracer release positions in region 2 and 3
x 10^-4
8
7
6
5
4
3
2
1
0
MSE_V
0
2
4
6
8
10
12
14
16
18
20
time (s)
MSE_V2a
MSE_V2b
MSE_V2c
MSE_V3a
MSE_V3c
MSE_V3b

(b)



(a)

(b)

Figure 5-8: Comparison of Median Mean Squared Error (MMSE) for different tracer release position in an asymmetric 3 vortex system

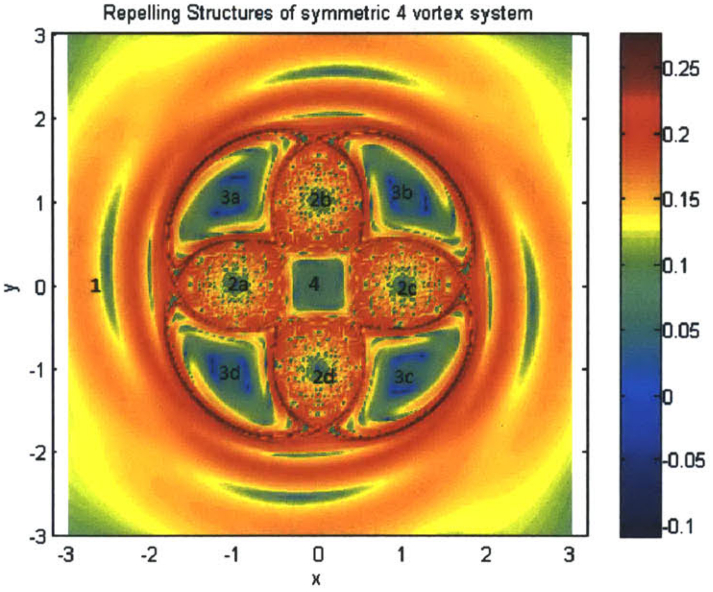


Figure 5-9: Different regions in the symmetric 4 vortex system based on Lagrangian coherent structures

the SIR filter with $N = 5000$ particles to carry out inference on the symmetric 4 vortex system.

We study the MMSE plots for these different tracer release positions and combinations. Clearly, from fig. 5-10a, we can say that tracers released in region 1 and 4 are not valuable to the inference process. The reason for this is that tracers in region 1 and 4 do not interact sufficiently well with the fields created by the vortices and are hence cannot capture valuable information about the positions of the vortices. In fig. 5-10b, we notice that the MMSE of tracers released in region 2 and 3 compare the same way as in the symmetric 3 vortex case. Tracers in region 2 perform significantly worse than the tracers in region 3 as their errors are an order of magnitude higher. As we had mentioned in the symmetric 3 vortex case, this mainly happens because the flow in region 2 is much more irregular and chaotic than in region 3, making them more prone to errors (including numerical integration error). Also, fig 5-10b shows that symmetrically placed tracers in region 3 perform the same, while symmetrically placed tracers in region 2 do not perform the same. Figure 5-10c shows the comparison of different tracers released in combination with tracer 2a. We observe that the single tracer in region 2a actually performs worse than most combination of double tracers. In fact, we can say for sure that single tracer in region 2a performs better than any combination of tracer 2a with another tracer from region 3. This is a promising result which agrees with our intuition that by increasing the number of tracers in right way one can improve the performance of the estimation process. Figure 5-10d shows that the performance of the single tracer released at 3a is equivalent to a lot of other double tracer combinations with 3a.

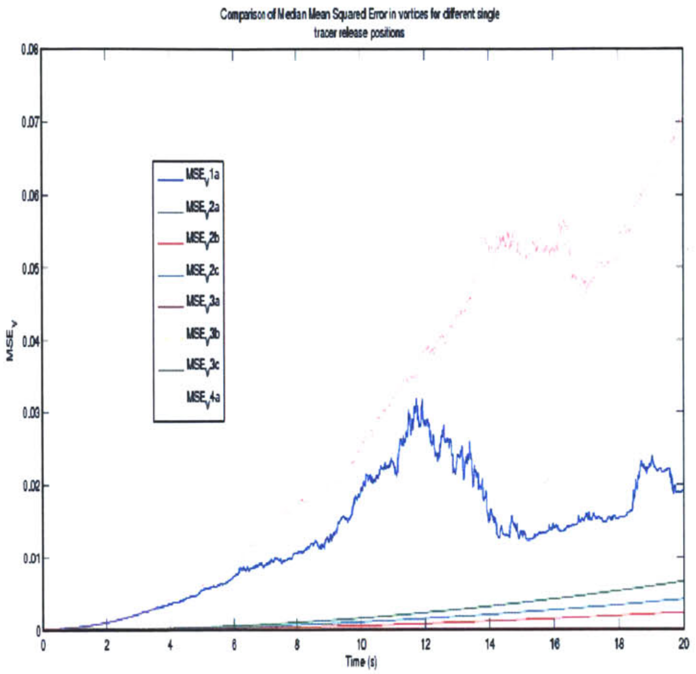
5.4.4 Asymmetric 4 vortex system

We now study the performance of the SIR filtering algorithms under different tracer release positions for an asymmetric 4 vortex system. Figures 4–2 c,d represent the attracting and repelling structures of the asymmetric 4 vortex system. Both the attracting and repelling coherent structures are very similar and based on them, we define the following regions 1, 2a, 2b, 2c, 2d, 3a, 3b and 4 as shown in fig. 5-11, where we intend to test out tracer placement problem.

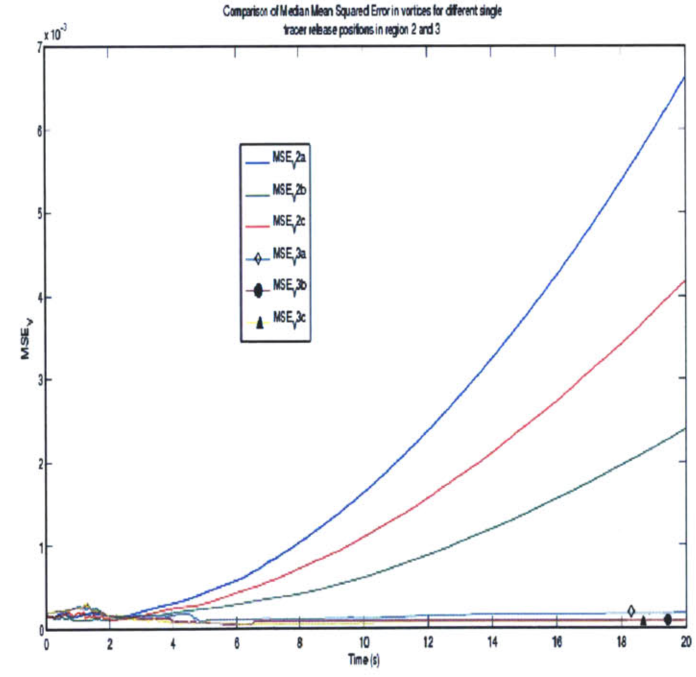
We consider the following single tracer release positions: 1 $(-4, 0)$, 2a $(0, 1.02)$, 2b $(1.02, 0)$, 2c $(-1/\sqrt{2} + 0.02, -1/\sqrt{2})$, 2d $(1/\sqrt{2}, -1/\sqrt{2})$, 3a $(-0.6, 0.12)$, 3b $(-0.6, 0.25)$, 4 $(0, 0)$.

Each of the tracers named as 2 define the regions surrounding eyes of the vortices that we observes in the FTLE plot (fig. 5-11). The tracer 3a is located exactly on a attracting/repelling structure, while tracer 3b is located in a region which is engulfed between attracting/repelling structures. We only study single tracer release positions here as double tracer combinations can get fairly complicated. We apply the SIR filter with $N = 5000$ particles to carry out inference on the asymmetric 4 vortex system.

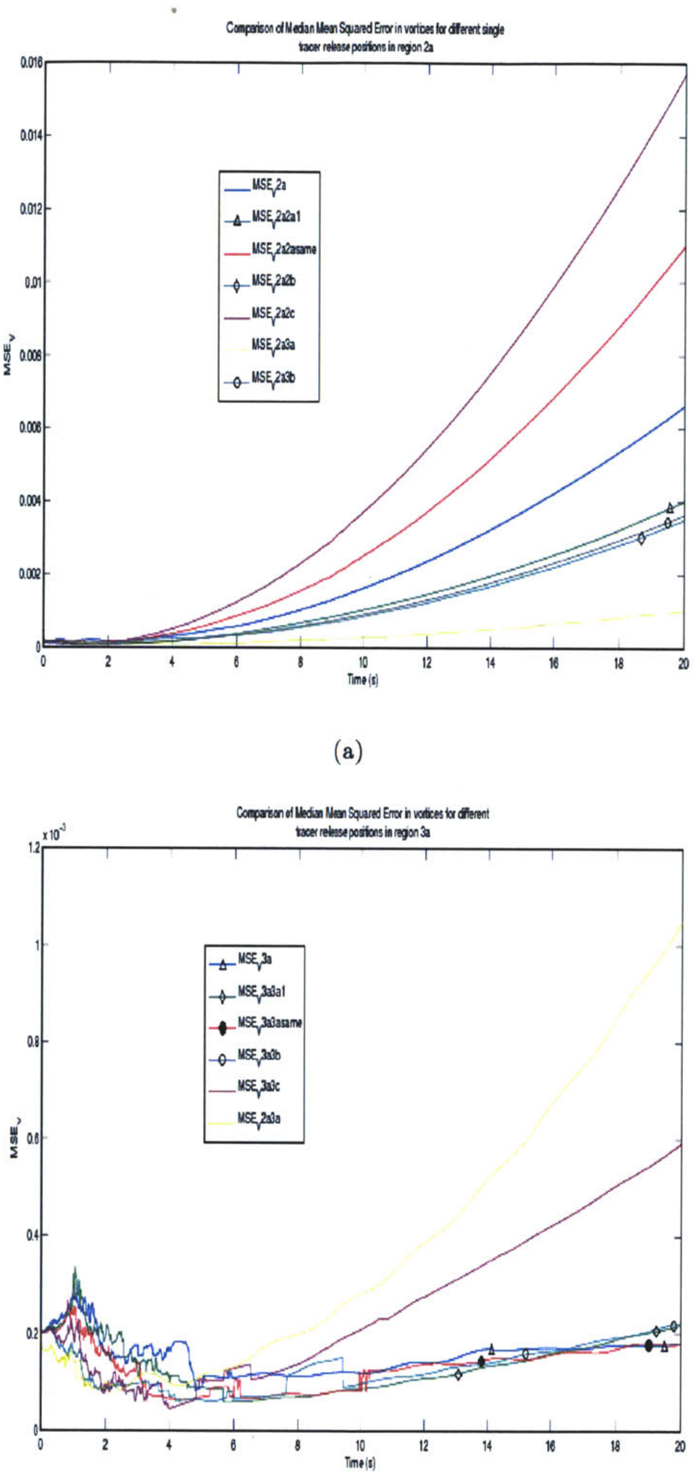
Figure 5-12 shows that the errors due to different placements are very similar and follow


Comparison of Median Mean Squared Error in vortices for different single tracer release positions
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
MSE_V
MSE_V1a
MSE_V2a
MSE_V2b
MSE_V2c
MSE_V3a
MSE_V3b
MSE_V3c
MSE_V4a
0
2
4
6
8
10
12
14
16
18
20
Time (s)

(a)


Comparison of Median Mean Squared Error in vortices for different single tracer release positions in region 2 and 3
$x\,10^{-3}$
7
6
5
4
3
2
1
0
MSE_V
MSE_V2a
MSE_V2b
MSE_V2c
MSE_V3a
MSE_V3b
MSE_V3c
0
2
4
6
8
10
12
14
16
18
20
Time (s)

(b)



(a)

(b)

Figure 5-10: Comparison of Median Mean Squared Error (MMSE) for different tracer release position in a Symmetric 4 vortex system

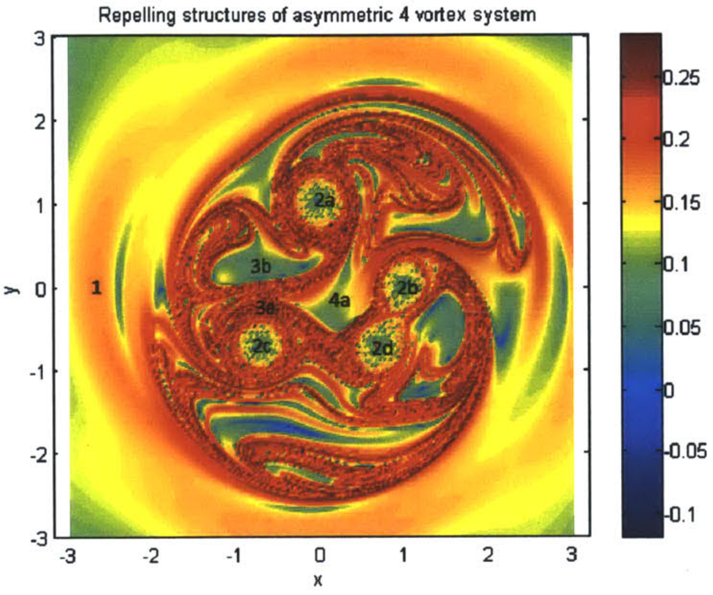


Figure 5-11: Different regions in the asymmetric 4 vortex system based on Lagrangian coherent structures

a similar pattern which seems to the evolution of the flow over time. In chaotic systems, filtering is a very hard process. At certain instances the chaotic nature of flow decreases and the filter can track the vortices efficiently for these short periods of time. This seems to happen between t=11-14s in fig. 5-12. In terms of tracer placement, strangely the tracer placed at region 1 which is outside the chaotic region does the best in terms of tracking. Although, we should not hypothesize anything from this observation, mainly because, the average MMSE in the asymmetric 4 vortex case is much larger and the filter loses track of the states of the vortices. Hence the problem lies in the fact that the current filter is not good enough to track this system. We should use particle filters with larger particle sizes and better algorithms like the APF to track these chaotic systems.

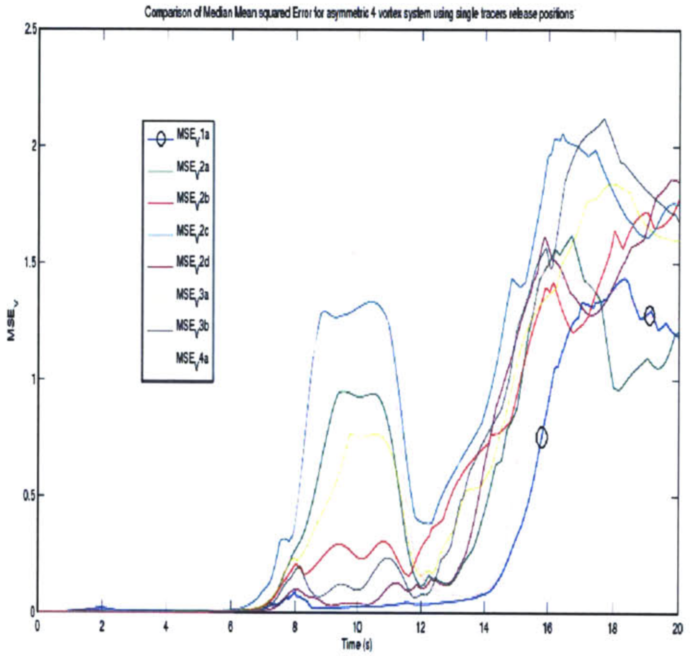


Figure 5-12: Comparison of Median Mean Squared Error (MMSE) for different single tracer release position in a asymmetric 4 vortex system

Chapter 6

Conclusion

We set out with the objective of using Particle filtering techniques in tracking fluid flow. The motivation behind the study was to explore the applications in ocean modeling, where tracers/buoys can be released in the flow to collect data in the form of position (say through satellite images) or velocity measurements over time. This sequentially available data could then be used to predict or track the flow field of the entire system (a larger part of the ocean). Our study provided several insights into the problem which can be used in the larger application of fluid tracking.

6.1 Filtering on point vortex models

6.1.1 Summary and key conclusions

We first explored the different variants of the particle filtering algorithms that are available for state estimation in Hidden Markov Models. Particle filtering is based on the idea of importance sampling and approximating probability distribution using a bunch of particles with varying weights. Particle filtering algorithms consist of two main steps: the predict step, where the particles are propagated from one state to the next and, the update step, where the weights of these particles are updated. The most commonly used algorithm is the Sequential Importance Resampling (SIR) algorithm, while we found out that variants like Auxiliary Particle filtering (APF) can do better under certain conditions. We realized that the biggest problem with most of these algorithms is that of particle degeneracy. In most of these algorithms, the variance of the posterior reduces with time naturally and we are left with only a handful of particles carrying all the weight. Resampling is an effective way of dealing with this; however, we clearly understood that this is the one of the main disadvantages of particle filtering algorithms. We then moved on to the application part of this study, i.e. tracking fluid flows using these algorithms and data.

In order to learn about the fluid flow, we needed to first consider a good system model which would define how the fluid particles move in time. Using either Navier-Stokes equations or Euler equations, would force us to learn about infinite dimensional state vectors. In order to make the forward model simpler and the state vector low dimensional, we considered the Point Vortex Model. The point vortex model is a simplification of the Euler equations, which assumes that all the vorticity in the system is concentrated at a certain number of particles. The assumption gives rise to a set of coupled ordinary differential equations in two-dimensions of space. Further, the state vector is also reduced to a small number equal to the number of vortices in the system. Any other particle in the fluid could be propagated if the positions of the vortices of the vortices are known. The point vortex model is a simple analytical model which makes the state vector relatively low-dimensional and in turn making the filtering problem more tractable.

We looked at few deterministic examples of the point vortex model. Specifically, we looked at N=3 and 4 vortices. We learned that both the cases were interesting in their own ways. The 3 vortex system was always non-chaotic, making it a good example for our study. On the other hand, 4 vortex systems are chaotic under most cases. We know that it can be very difficult to track chaotic systems, as they have an inherent amount of uncertainty associated with them. In setting up our filtering problem, we consider a stochastic version of the point vortex model, with added noises, to account for the uncertainty associated with the model (as it is different from a natural model). Further, we consider noisy observations from tracers released in the system.

We looked into different aspects of the SIR algorithm in terms of its performance for the point vortex model. We first figured out a good starting prior on the vortex positions, which is very essential especially for chaotic system like the 4 vortex model. Next, we studied the effect of increasing the number of particles used to characterize the posterior distribution of the states. As expected, the algorithm performance improved with increase in the number of particles used. We set a particle size of 5000 for the rest of our study, which we considered to be adequate. However, we also note here that one can improve the performance of the filtering process, especially in case of chaotic systems with even larger number of particles. Although, then the problem may start to become extremely computationally expensive.

The final study was focused on studying chaotic 4 vortex systems. As mentioned earlier, tracking chaotic systems can be a very difficult task. Chaotic flows have particles diverging away at exponential and hence even a small error in estimation can lead to the filtered state totally losing track of the true state. From our studies, we learn that we lose track of the chaotic system after a certain time. By increasing the number of particles, the system can be tracked for a longer time. However, eventually, the filtered state always loses track of the true state. We conclude that it is very difficult to track chaotic systems for prolonged periods of time using the generic versions of the particle filtering algorithms.

6.1.2 Improvements and open challenges

In terms of improvements, one can have a system model better than the point vortex model. For example, a model with vortex patches is a better description of chaotic fluid flow than a point vortex model. However, we note that a more elaborate can make the filtering process more difficult. As the number of dimensions of the state increases, the filtering problem becomes more difficult to solve and also becomes more computationally expensive.

Secondly, we assumed that all the parameters associated with the models (eg: circulation, process noise) are known. Otherwise, we need to use algorithms which carry out simultaneous state and parameter estimation. This can also be area of further study specifically in the context of the vortex model.

It can definitely be an area of future research to investigate whether or not chaotic systems can be effectively tracked for longer periods of time. The fact that chaotic systems cannot be tracked for long periods of time clearly stood out as a challenging task from our study. One can suggest using better versions of the particle filtering algorithms, involving better sampling algorithms, like Chorins filter, to track these chaotic systems. Although, as a side study we observed that the Chorins filter in particular does not help for this problem.

6.2 Tracer placement using LCS

6.2.1 Summary and key conclusion

As most flows are chaotic, it is imperative to understand the extent to which they were chaotic. In other words, we needed a measure which could quantify how chaotic a system really was. Lyapunov exponents have been used in literature to study chaotic systems. These exponents define how sensitive the system is to the initial conditions, which is a very important variable to be considered in the context of our filtering problem. As chaotic flows can have particles moving away from each other at exponential rates, it is of utmost importance to know their initial positions. We then explored a particle version of Lyapunov Exponents called finite time lyapunov exponents (FTLE) can be learnt from the initial conditions of the system. Further, we plotted the scalar FTLE field, which revealed certain coherent structures in the system. Coherent structures are invariant structures, which define boundaries of the flow of the system. They are mainly of two kinds: attracting and repelling. Attracting structures are ones which attract fluid particles towards themselves, while repelling structures repel them away. In other words, attracting structures are regions to which particles converge to over time, while repelling structures are regions from where particles diverge away. The attracting and repelling structures of the 3 and 4 vortex systems clearly helped us identify regions where one could possibly place tracers to start with to gain better information out of the system. It also motivated us to consider the problem for experimental design in terms of ideal tracer placement besides our original problem of tracking the system in a general case.

We studied the effect of tracer release positions on the state estimation process under different vortex configurations. In the symmetric 3 vortex case, we identified regions based on the FTLE plots where single tracer placements lead to better estimations. These regions were those labeled 3 and had relatively low values of FTLE but were still under the influence of the vortices. As opposed to a tracer released in region 3, the one released in region 2 did not perform well, mainly because the flow in this region was more irregular. Also, single tracers released in region 2, which were symmetrically about the origin, did not lead to similar estimation errors. Again, this suggested that the flow in region in 2 was irregular and more sensitive to initial conditions. We also tried to learn the estimation performance when two tracers were released in the flow. Strangely, it turned out that placing two tracers very close to each other in region 2 and region 3 did not lead to better estimation performance than placing a single tracer in region 2 and region 3 respectively. This was a perplexing result. Next, we moved on to an asymmetric 3 vortex case, where we learnt that the differences between filtering performance in regions 2 and 3 were less contrasting than in the symmetric case. This mainly happens because region 3 has more irregular flow than in the symmetric case. However, one clearly benefits by placing a tracers in region 3 than in region 2. In the 4 vortex symmetric case, we obtain similar results, where regions corresponding to region 3 in the 3 vortex system tend to lead better estimation performance.

The above studies on tracer placement lead us to the following conclusions. Firstly, we need to evaluate the attracting and repelling structures of the system based on FTLE field. Based, on these structures we should identify regions which do not interact much among themselves. The tracers should be placed in regions like region 3, which have the advantage of both being close to the vortices as well as low FTLE fields meaning that the flow in the region is less irregular or sensitive to initial conditions. When placing multiple tracers, they should be placed different regions like region 3 (eg: 3a and 3b), as they lead to best performance.

6.2.2 Future Work

We think that the most important contribution of this work was to set up the connection between FTLE based coherent structures and tracer release position. It clearly showed that we can exploit use the initial conditions of the system to ideally release tracers which lead to good estimation performance. This study forms the groundwork for more applied studies where we can release a large number of tracers in the fluid and try to learn the flow and in turn improve our tracer releases.

The fact that placing two tracers very close to each other in region 2 and region 3 did not lead to better estimation performance than placing a single tracer in region 2 and region 3 respectively was perplexing. One way to explain is that the data provided the tracers are conflicting. In other words, when one tracer gives rise to a high likelihood the other does not and vice versa. Also, we notice that when we start with multiple tracers from same point in region 3 this anomaly tends to go away. However, we definitely think that this result is an open question to be answered and can be the scope of future work based on this study.

The question of ideal tracer deployment is even more difficult for chaotic systems as the FTLE plots are far more messy and do not have well-defined structures. The results of the single tracer study in this regard can be considered to be inconclusive. In all, tracking and ideal tracer deployment in case of chaotic system can be a much more complex problem and should be a challenging study to pursue.

Bibliography

[1] Christophe Andrieu and Arnaud Doucet. Online expectation-maximization type algorithms for parameter estimation in general state space models. pages 69–72, 2003. 43

[2] A Apte, C K R T Jones, and A M Stuart. A Bayesian approach to Lagrangian data assimilation. *Tellus - Series A: Dynamic Meteorology and Oceanography*, 60(2):336–347, 2008. 13

[3] A Apte, C K R T Jones, A M Stuart, and J Voss. Data assimilation: Mathematical and statistical perspectives. *International Journal for Numerical Methods in Fluids*, 56(8):1033–1046, 2008. 13

[4] Hassan Aref and Neil Pomphrey. Integrable and Chaotic Motions of Four Vortices I . The Case of Identical Vortices. *Physics Letters*, 380(1779):359–387, 1982. 14, 70

[5] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A Tutorial on Particle Filters for Online Nonlinear / Non-Gaussian Bayesian Tracking. 50(2):174–188, 2002. 9, 20, 21

[6] Ethan Atkins, Matthias Morzfeld, and Alexandre Chorin. Implicit Particle Filters : a Numerical Case Study Using the Lorenz Attractor. *Geophysical Research Abstracts*, 13:2011–2011, 2011. 72

[7] S Barakat, C Garth, and X Tricoche. Interactive Computation and Visualization of Lagrangian Coherent Structures. *IEEE Trans Visual Comput Graph*, submitted, 2009. 80

[8] R L Barillec. *Bayesian data assimilation.* PhD thesis, 2008. 13

[9] M Bask. Testing chaotic dynamics via Lyapunov exponents. *Physica D: Nonlinear Phenomena*, 114(1-2):1–2, 1998. 77

[10] Richard F Bass. Stochastic differential equations with jumps. *Probability Surveys*, 2(1):107–144, 2003. 72

[11] Wolf-Jürgen Beyn and Alexander Lust. A hybrid method for computing Lyapunov exponents. *Numerische Mathematik*, 113(3):357–375, 2009. 79

[12] Mark Briers, Arnaud Doucet, and Simon Maskell. Smoothing algorithms for statespace models. *Annals of the Institute of Statistical Mathematics*, 62(1):61–89, June 2009. 40

[13] Amarjit Budhiraja, Lingji Chen, and Chihoon Lee. A survey of numerical methods for nonlinear filtering problems. *Physica D: Nonlinear Phenomena*, 230(1-2):27–36, 2007. 14

[14] Gerrit Burgers, Peter Jan Van Leeuwen, and Geir Evensen. Analysis Scheme in the Ensemble Kalman Filter. *Monthly Weather Review*, 126(6):1719–1724, 1998. 14

[15] Olivier Cappé. Online EM Algorithm for Hidden Markov Models Online EM in the Independent Case. pages 1–20, 2008. 43

[16] M Cencini, F Cecconi, and A Vulpiani. *Chaos: from simple models to complex systems*, volume 17. World Scientific Pub Co Inc, 2009. 78

[17] A J Chorin, J E Marsden, and A Leonard. A Mathematical Introduction to Fluid Mechanics. *Journal of Applied Mechanics*, 47(2):460, 1980. 68

[18] Alexandre Chorin, Matthias Morzfeld, and Xuemin Tu. Implicit particle filters for data assimilation. *Communications in Applied Mathematics and Computational Science*, 5(2):221–240, November 2010. 15, 72, 88

[19] Alexandre J Chorin and Xuemin Tu. Implicit sampling for particle filters. *Proceedings of the National Academy of Sciences of the United States of America*, 106(41):17249–17254, 2009. 15

[20] A Chrisohoides and F Sotiropoulos. Experimental visualization of Lagrangian coherent structures in aperiodic flows. *Physics of Fluids*, 15(3):L25, 2003. 80

[21] J P Christiansen. Numerical Simulation of Hydrodynamics by Method of Point Vortices. 379:363–379, 1973. 14, 68

[22] Anne Cuzol and Etienne Mémin. A stochastic filtering technique for fluid flow velocity fields tracking. *IEEE transactions on pattern analysis and machine intelligence*, 31(7):1278–93, July 2009. 14

[23] Robert L Devaney. *An Introduction to Chaotic Dynamical Systems*, volume 13 of *Studies in Nonlinearity*. 1989. 72

[24] A Doucet, N De Freitas, and N Gordon. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, 2001. 14

[25] Arnaud Doucet and Adam M Johansen. A Tutorial on Particle Filtering and Smoothing : Fifteen years later. (December):4–6, 2008. 23, 36

[26] Bruno Eckhardt and Demin Yao. Local Lyapunov Exponents in Chaotic Systems. *Physica D*, 65:100–108, 1993. 77

[27] AS Fokas and PM Santini. Coherent structures in multidimensions. *Physical Review Letters*, 63(13):1329–1333, 1989. 80

[28] N J Gordon, D J Salmond, and A F M Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing IEE Proceedings F*, 140(2):107–113, 1993. 52

[29] M Hairer, A M Stuart, and J Voss. A Bayesian Approach to Data Assimilation. *Physica D*, (May):1–20, 2005. 13

[30] G Haller. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D: Nonlinear Phenomena*, 149(4):248–277, 2001. 15, 79

[31] G Haller. Lagrangian coherent structures from approximate velocity data. *Physics of Fluids*, 14(6):1851, 2002. 80

[32] G. Haller and G. Yuan. Lagrangian coherent structures and mixing in two-dimensional turbulence. *Physica D: Nonlinear Phenomena*, 147(3-4):352–370, December 2000. 15, 80

[33] Boris Hasselblatt and Anatole Katok. A First Course in Dynamics: With a Panorama of Recent Developments., 2003. 71

[34] D J Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review*, 43(3):525–546, 2001. 72

[35] D J Higham and Peter E Kloeden. Numerical methods for nonlinear stochastic differential equations with jumps. *Numerische Mathematik*, 101(1):101–119, 2005. 72

[36] D J Higham, X Mao, and A M Stuart. Exponential mean square stability of numerical solutions to stochastic differential equations. *London Math Soc J Comput Math*, 6(1):297–313, 2003. 72

[37] Rebecca Honeycutt. Stochastic Runge-Kutta algorithms. I. White noise. *Physical Review A*, 45(2):600–603, 1992. 73

[38] Rebecca L Honeycutt. Stochastic Runge-Kutta algorithms. II. Colored noise. *Physical Review A*, 45(2):604–610, 1992. 73

[39] P L Houtekamer and Herschel L Mitchell. Data Assimilation Using an Ensemble Kalman Filter Technique. *Monthly Weather Review*, 126(3):796–811, 1998. 14

[40] S M Iacus. Numerical Methods for SDE. *Methods*, 1, 2008. 72

[41] Kayo Ide and Michael Ghil. Extended Kalman filtering for vortex systems . Part I : Methodology and point vortices. 27:301–332, 1997. 14, 68

[42] SJ Julier and JK Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004. 14

[43] Christian Kahl and Henri Schurz. Balanced Milstein Methods for Ordinary SDEs. *Monte Carlo Methods and Applications*, 12(2):143–170, 2006. 72

[44] R E Kalman. A new approach to linear filtering and prediction problems. *Journal Of Basic Engineering*, 82(Series D):35–45, 1960. 14

[45] R E Kalman and R S Bucy. New results in linear filtering and prediction theory. *Journal Of Basic Engineering*, 83(1):95–108, 1961. 14

[46] P E Kloeden and E Platen. A survey of numerical methods for stochastic differential equations. *Stochastic Hydrology and Hydraulics*, 3(3):155–178, 1989. 72

[47] Peter E Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*, volume 23 of *Applications of Mathematics*. Springer, 1992. 72

[48] Peter E Kloeden, Eckhard Platen, and Henri Schurz. *Numerical Solution of SDE Through Computer Experiments*, volume 1 of *Universitext*. Springer, 1994. 72

[49] J Koller, R H W Friedel, G D Reeves, and Y Chen. Radiation Belt Data Assimilation with an Ensemble Kalman Filter. In *4th Space Weather Symposium*, page 3.5. The 87th AMS Annual Meeting, 2007. 14

[50] L Kuznetsov and K Ide. A method for assimilation of Lagrangian data. *Monthly Weather Review*, pages 2247–2260, 2003. 14, 80

[51] H Lamba, J C Mattingly, and A M Stuart. An Adaptive Euler-Maruyama Scheme For SDEs: Convergence and Stability. *IMA Journal of Numerical Analysis*, 27(3):479–506, 2006. 72

[52] F Lekien, S C Shadden, and J E Marsden. N-dimensional Lagrangian Coherent Structures: Definition and Properties. *Physica D*, 2006. 80

[53] Francois Lekien and Naomi Leonard. Dynamically Consistent Lagrangian Coherent Structures. *Aip Conference Proceedings*, 742:132–139, 2004. 80

[54] Francois Lekien, Shawn C Shadden, and Jerrold E Marsden. Lagrangian coherent structures in n-dimensional systems. *Journal of Mathematical Physics*, 48(6):065404, 2007. 80

[55] J Liu and M West. Combined parameter and state estimation in simulation-based filtering. *Sequential Monte Carlo Methods in Practice*, pages 197–223, 2001. 43, 53

[56] L Longa, S Dias, and E Curado. Lyapunov exponents and coalescence of chaotic trajectories. *Physical Review E*, 56(1):259–263, 1997. 77

[57] H Madsen and R Cañizares. Comparison of extended and ensemble Kalman filters for data assimilation in coastal area modelling. *International Journal for Numerical Methods in Fluids*, 31(6):961–981, 1999. 14

[58] Olivier Martin. Lyapunov Exponents of Chaotic Systems. *System*, 41(1-2):249–261, 1985. 77

[59] Igor Mezic and Stephen Wiggins. A method for visualization of invariant sets of dynamical systems based on the ergodic partition. *Chaos (Woodbury, N.Y.)*, 9(1):213–218, March 1999. 80

[60] Katsuaki Mishima, Asuka Sugii, Tomohiro Yamada, Tatsushi Matsumura, and Norifumi Moritani. Lyapunov exponents. *ReCALL*, 63(3):129–133, 2010. 77

[61] A S Monin. The definition of coherent structures. *Sov Phys Dokl*, 36(6), 1991. 80

[62] Matthias Morzfeld, Ethan Atkins, and Alexandre Chorin. Geomagnetic data assimilation with an implicit particle filter into a one-dimensional sparsely observed MHD system. *Geophysical Research Abstracts*, 13:2011–2011, 2011. 15

[63] Matthias Morzfeld, Xuemin Tu, Ethan Atkins, and Alexandre J Chorin. A random map implementation of implicit filters. (1):1–34, 2011. 15

[64] S Ogawa. On a deterministic approach to the numerical solution of the SDE. *Mathematics and Computers in Simulation*, 55(1-3):209–214, 2001. 72

[65] T Okushima. New method for computing finite-time Lyapunov exponents. *Physical Review Letters*, 91(25):254101, 2003. 79

[66] Edward Ott, Brian R Hunt, Istvan Szunyogh, Aleksey V Zimin, Eric J Kostelich, Matteo Corazza, Eugenia Kalnay, D J Patil, and James A Yorke. A Local Ensemble Kalman Filter for Atmospheric Data Assimilation. *Tellus - Series A: Dynamic Meteorology and Oceanography*, 56(5):66, 2002. 14

[67] J M Ottino. The Kinematics of Mixing: Stretching, Chaos, and Transport. VonM. M. Ottino. Cambridge University Press, Cambridge - New York - New Rochelle 1989. XIV, 364 S., zahlr. Abb. U. Tab., geb., 50,-. *Cambridge University Press*, 62(1):74–74, 1990. 79

[68] Jan Palczewski. Milstein scheme and convergence Euler-Maruyama scheme Convergence of schemes for SDEs Convergence of Euler-Maruyama scheme Strong error for Euler-Maruyama. *Order A Journal On The Theory Of Ordered Sets And Its Applications*, 2009. 72

[69] Chong Pan, JinJun Wang, and Cao Zhang. Identification of Lagrangian Coherent Structures in a Turbulent Boundary Layer. *Spectrum*, 52(2):248–257, 2011. 80

[70] Michael K Pitt and Neil Shephard. Filtering via Simulation : Auxiliary Particle Filters. 94(446):590–599, 2007. 31

[71] E Platen. Derivative free numerical methods for stochastic differential equations. In *Stochastic Differential Systems*, volume 96/1987 of *Lecture Notes in Control and Information Sciences*, pages 187–193. Springer, 1987. 72

[72] Eckhard Platen. An introduction to numerical methods for stochastic differential equations. *Acta Numerica*, 8(1999):197, 2008. 72

[73] Nicholas G. Polson, Jonathan R. Stroud, and Peter Müller. Practical filtering with sequential parameter learning. *Journal of the Royal Statistical Society - Series B: Statistical Methodology*, 70(2):413–428, 2008. 40, 43, 55, 64, 65

[74] B Protas. Vortex dynamics models in flow control problems. *Nonlinearity*, 21(9):R203–R250, September 2008. 13

[75] Rolf H Reichle, Jeffrey P Walker, Randal D Koster, and Paul R Houser. Extended versus Ensemble Kalman Filtering for Land Data Assimilation. *Journal of Hydrometeorology*, 3(6):728–740, 2002. 14

[76] Erico L Rempel, Abraham C L Chian, and Axel Brandenburg. Lagrangian coherent structures in nonlinear dynamos. *The Astrophysical Journal*, 735(1):L9, 2010. 80

[77] Natalie Ross, Elizabeth Bradley, and Jean Hertzberg. Dynamics-Informed Data Assimilation in a Point-Vortex Model. pages 1–55. 14

[78] Yoram Rubin, Xingyuan Chen, Haruko Murakami, and Melanie Hahn. A Bayesian approach for inverse modeling, data assimilation, and conditional simulation of spatial random fields. *Water Resources Research*, 46(10):W10523, 2010. 13

[79] Yoshihiro Saito and Taketomo Mitsui. Stability Analysis of Numerical Schemes for Stochastic Differential Equations. *SIAM Journal on Numerical Analysis*, 33(6):2254, 1996. 72

[80] H Salman. A hybrid grid / particle filter for Lagrangian data assimilation . II : Application to a model vortex flow. 1565(August):1551–1565, 2008. 74

[81] H Salman, L Kuznetsov, C K R T Jones, and K Ide. A Method for Assimilating Lagrangian Data into a Shallow-Water-Equation Ocean Model. *Monthly Weather Review*, 134(4):1081–1101, 2006. 14

[82] Benjamin Schindler, Ronald Peikert, Raphael Fuchs, and Holger Theisel. Ridge Concepts for the Visualization of Lagrangian Coherent Structures. *submitted to TopoInVis*, pages 1–14, 2011. 80

[83] Shawn C. Shadden, Francois Lekien, and Jerrold E. Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodicflows. *Physica D: Nonlinear Phenomena*, 212(3-4):271–304, December 2005. 15, 78, 80

[84] E T Spiller, A Budhiraja, K Ide, and C.K.R.T. Jones. Modified particle filter methods for assimilating Lagrangian data into a point-vortex model. *Physica D: Nonlinear Phenomena*, 237(10-12):1498–1506, 2008. 13, 14, 15, 67, 68

[85] K Stefaski, K Buszko, and K Piecyk. Transient chaos measurements using finite-time Lyapunov exponents. *Chaos Woodbury Ny*, 20(3):033117, 2010. 78

[86] G Storvik. Particle filters for state-space models with the presence of unknown static parameters. IEEE Transaction on. *Signal Processing*, 50:281–289, 2002. 43

[87] Twiki Tags, Home Phys, View Edit, Jump Lyapunov Exponents, and Chaotic Pendulum. Lyapunov Exponents for the Chaotic Pendulum. *Search*, pages 1–6, 2012. 77

[88] Pierre Tisseur. Cellular automata and Lyapunov exponents. *October*, 13:1–21, 2003. 77

[89] Olivier Titaud, Jean-Michel Brankart, and Jacques Verron. On the use of Finite-Time Lyapunov Exponents and Vectors for direct assimilation of tracer images into ocean models. *Tellus - Series A: Dynamic Meteorology and Oceanography*, 63(5):1038–1051, 2011. 78

[90] Olli-pekka Tossavainen, Julie Percelay, Andrew Tinka, Qingfang Wu, and Alexandre M Bayen. Ensemble Kalman filter based state estimation in 2D shallow water equations using Lagrangian sensing and state augmentation. pages 1783–1790, 2008. 14

[91] Nikica Uglešić. Topology and its Applications. *Glasnik matematički*, 35(55)(2):245–259, 2002. 71

[92] H Waelbroeck and F Zertuche. Discrete Chaos. *American Journal of Mathematics*, 65(2):279, 1996. 71

[93] N Wessel, C Ziehmann, J Kurths, U Meyerfeldt, A Schirdewan, and A Voss. Short-term forecasting of life-threatening cardiac arrhythmias based on symbolic dynamics and finite-time growth rates. *Physical Review E Statistical Physics Plasmas Fluids And Related Interdisciplinary Topics*, 61(1):733–739, 2000. 77

[94] Niels Wessel, Udo Meyerfeldt, Alexander Schirdewan, Jurgen Kurths, and Andreas Voss. Short-term Forecasting of Life-threatening Arrhythmias with Finite Time Lyapunov Exponents. *Engineering In Medicine And Biology*, 20(1):326–329, 1998. 77

[95] Mike West. Approximating posterior distributions by mixtures. *Journal of the Royal Statistical Society Series B Methodological*, 55(2):409–422, 1993. 52, 53

[96] Mike West. Mixture Models, Monte Carlo, Bayesian Updating and Dynamic Models. *Computing Science and Statistics*, 24(325-333):325–325, 1993. 52, 53

[97] Christopher K Wikle and L Mark Berliner. A Bayesian tutorial for data assimilation. *Physica D*, 2006. 13

[98] Joshua Wilkie. Numerical methods for stochastic differential equations. *Physical Review E - Statistical, Nonlinear and Soft Matter Physics*, 70(1 Pt 2):017701, 2004. 72

[99] X Zhang. Euler Maruyama approximations for SDEs with non-Lipschitz coefficients and applications. *Journal of Mathematical Analysis and Applications*, 316(2):447–458, 2006. 72