5-16-2003

Comparison of Sampling-Based Algorithms for Multisensor Distributed Target Tracking

Trang Nguyen
University of New Orleans

Recommended Citation

Nguyen, Trang, "Comparison of Sampling-Based Algorithms for Multisensor Distributed Target Tracking" (2003). *University of New Orleans Theses and Dissertations*. 20.
https://scholarworks.uno.edu/td/20

Comparision of Sampling-Based Algorithms for Multisensor Distributed Target Tracking

A Thesis

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirement for the degree of

Master of Science
in
The Department of Electrical Engineering

by
Trang Minh Nguyen
B.S., University of New Orleans, 2000

May 2003

Acknowledgment

My supervisors University Research Professor X. Rong Li and Assistant Professor Vesselin P. Jilkov deserve my deepest gratitude. This thesis is to a large extent the result of their skillful guidance and encouraging support of my work. I would like to thank Dr. Li for having taught me everything. I have a special thank to Dr. Jilkov for his fruitful suggestions and spontaneous discussions we have had during this work.

I would like to thank Dr. Hiumin Chen for serving in the commitee and his useful comments to my thesis. I like to thank Kushu Zhang, an outstanding researcher, for her help.

New Orleans, April 2003
Trang Nguyen

Contents

Abstract

Nonlinear filtering is certainly very important in estimation since most real-world problems are nonlinear. Recently a considerable progress in the nonlinear filtering theory has been made in the area of the sampling-based methods, including both random (Monte Carlo) and deterministic (quasi-Monte Carlo) sampling, and their combination.

This work considers the problem of tracking a maneuvering target in a multisensor environment. A novel scheme for distributed tracking is employed that utilizes a nonlinear target model and estimates from local (sensor-based) estimators. The resulting estimation problem is highly nonlinear and thus quite challenging. In order to evaluate the performance capabilities of the architecture considered, advanced sampling-based nonlinear filters are implemented: particle filter (PF), unscented Kalman filter (UKF), and unscented particle filter (UPF). Results from extensive Monte Carlo simulations using different configurations of these algorithms are obtained to compare their effectiveness for solving the distributed target tracking problem.

Chapter 1

Introduction

1.1 Motivation for Using Simulation Based Methods

Estimation problem concerns with estimating unknown quantities based on obtained measurements. If the unknown quantities are modeled as nonrandom parameter, the measurements provide the likelihood function of the sought-after parameters; this is static estimation. On the other hand if the unknown quantities are modeled as random parameters, e.g. dynamic systems, there has to be some prior knowledge about the paramters available for estimation. This is referred to as state estimation. Bayesian estimation is a framework that solves estimation problems by utilizing both the likelihood function and the prior knowledge (in the form of a distribution). In this framework all inference about the parameters can be drawn from the posterior density function (PDF) conditioned on the observed data.

Many estimation problems obtain data sequentially in, e.g., time such as in tracking a target. Thus it is necessary to update the posterior PDF as new measurements arrive. Sequential Bayesian estimation does this in two steps: measuement update and time update. Under linear Gaussian assumption, Kalman filter is used to compute the sequence of the posterior distributions by calculating the mean and covariance of the posterior. This is perhaps the only perfect case (in the sense of mathematical tractability) that gives analytical solution to recursive Bayesian estimation. Kalman filter can only deal with a limited class of problems. In other words, Kalman filter is not a general method for estimation. Many problems encountered are nonlinear and typically the distributions of the model are known to be far from Gaussian, and we need a filter to handle these cases also.

An algorithm that is widely used to deal with nonlinear problem is the Extended Kalman filter (EKF). It basically linearizes the nonlinear function(s) at each time step and employs the Kalman filter (KF) to do filtering. The linearization and Gaussian assumption are the main sources of error in the EKF method. And if the error is large, the EKF could diverge.

Numerous approximate methods have been developed for nonlinear nonGaussian problems, and among them are the Gaussian sum filter and a method based on a piecewise linear approximation of the distribution. A more direct numerical method represents each distribution as a set of values on a grid. This is a general approach but it is impractical for high dimensional problem due to its suffering from the “curse of dimensionality”.

Another approach to nonlinear nonGaussian problems is by keeping track of the posterior density evolution using finite samples from that density, not by direct approximation. This makes sense since the posterior PDF is hard or there is no way to compute and since in most problems what one is interested in is just some features (usually the mean and covariance) of the PDF which can be deduced from the samples (or particles). Extracting these features in theory needs to evaluate integrals. Since we do not have access to the PDF, features can be approximated by integration with Monte Carlo method. Hence simualtion methods are also called Monte Carlo simulation methods. Simulation based technique is general and virtually can be applied to any Bayesian estimation problem. Though this implementation is machine intensive, in general the computation cost does not increase proportionally on the dimention of the problem at hand.

1.2 Simulation Monte Carlo Methods

Though Monte Carlo method has been in the literature for a long time, only recent years does it gain moments and made considerable progress following the advances in computer technolygy. Progress have been in both general methodolygies as well as specific applications. A survey of the progress can be found in [4]. The first functional Monte Carlo method for sequential estimation was only recently developed in [3]. Following this development were a number of proposals for improvements on the generic particle filter. Three prominent methods are random (Monte Carlo)[3], [4], [5] and deterministic sampling (quasi-Monte Carlo) [6], [7], [8] and their combinations [9].

In random sampling (*Particle filter* (PF)) the probability densisty functions (PDFs) of the state are represented by a finite sets of sample points (i.e., particles) which are propagated and updated by the filter to approximate the posterior density of the target state at every time step. Provided the number of particles is large enough, the accuracy of this approximation is high. In the deterministic sampling (e.g., *unscented transform* (UT)) based filtering the PDFs are represented by a small number of deterministic points (referred to as sigma points), specially designed to approximate the moments of states after nonlinear transformations by the sample moments of the transformed sigma points. This is based on the principle that it is easier to approximate a Gaussian density than to approximate nonlinear function (as in the *extended Kalman filter* (EKF)). The *unscented particle filter* (UKF) is a PF technique which employs unscented Kalman filters to obtain better proposal distributions and thus dramatically reduces the number of particles in the PF implementation.

1.3 Distributed Target Tracking

Nonlinear filtering is very important in estimation since most of real-world problems are nonlinear. A typical nonlinear problem in target tracking is that of a maneuvering target in which the target, e.g., an airplane, climbs up/down or makes a turn. It is common to track a target using measurements from several sources e.g., radars or sensors that may be located on the same platform or may be in geographicaly different locations like in case of radars. In either case state estimation of the target motion calls for data fusion. Data fusion concerns with how to best utilize the useful information contained in multiple sets of data for the purpose of estimating an unknown quantity–a parameter or process. There are two basic architectures for processing data from different sensors: centralized and decentralized (also called distributed) fusion depending on what form of data the fusion center receives: raw or processed data. Centralized fusion is the case in which data received from local sensors are raw or unprocessed observations; this is the traditional estimation in the literature. In the other architecture, distributed fusion, what the fusion center receives are local state estimates output by those local sensors.

Distributed data fusion poses its own challenges. Among those is especially the formulation of the crosscovariances between the local estimates. This is the case because local sensors measure the same motion, like in the case of tracking a single target, that is, present in the local observations is the same process noise. Models for data fusion and fusion rules are discussed in [11].

1.4 Thesis Outline

Sampling based methods are general algorithms. Successful applying of the methods to a specific problem requires solving other subproblems specific to the problem at hand. This work formulates a scheme for state estimation in nonlinear distributed target tracking using sampling based algorithms (PF, UKF, and UPF). Specifically, it uses nonlinear equations to model dynamic state and observation, employs these sampling based methods to filter at both local sensors and fusion center. Also the crosscovariance of the local estimates are computed using the sigma points sent from the local UKFs to the fusion center. This work also compares these algorithms using the root mean square error.

This thesis is a detailed version of [12]. The rest of this thesis is organized as follows. Chapter 2 briefly reviews Bayesian estimation in the literature. Chapter 3 contains a brief survey of approaches to recursive Bayesian filtering. General numerical integration, derivation and detailed implementations of the three prominent sampling based filters: PF, UKF, and UPF, are given in chapter 4. Chapter 5 consists of methodology of using sampling based method to distributed state estimation, simulation scenarior and results.

Chapter 2

Bayesian Estimation

Statistical estimation concerns with inferring knowledge about unknown parameters of a process indirectly through measurements. The parameters may be states of the process or system, and the system affects these parameters in a known manner described by a system model. Observations are measurements of these states at some point in time and are described by the measurement equation. In many practical problems the sought parameters have dynamic properties that make them change with, for example, time. Thus, the inferred knowledge need to be updated as new observatons arrive. This is referred to as recursive estimation. Recursive Bayesian estimation calls for the update of the statistical characteristics of the system, namely the posterior pdf, recursively.

As opposed to Bayesian point of view, unkown parameters are assumed to be unkown constants in nonBayesian approach, and these two phylosophies results in different estimators. Note that commonly nonBayesian approach is also referred to as parametric estimation. Since Monte Carlo method is used mainly to solve recursive Bayesian estimation problems, this chapter reviews only Bayesian estimation and serves as a theorectial basis for the discussion in the sequel. As a result, it does not provide a complete coverage of estimation which can be found in any text book on estimation such as [19] and [2] upon which this review is based.

2.1 Notation Convention and Terminology

In literature the term parameter is usually used to denote unknown deterministic entities, while the term state is used to denote random entities. In this thesis paramerter is used to refer any sought-after entities when it is unambiguous. When needed to distinguish random from nonrandom entities, parameter and state are used as in the literature.

Unless stated differently, x denote a generic n-dimentional state vector, the estimatee–the sought-after entity, and z a p-dimentional observation vector.

$$Z^k = \{z_1, z_2, ..., z_k\}$$

denotes the sequence of observations, and index k may be dropped when the context is unambiguous. Random variables are sometimes typed in bold face, $\mathbf{x}$, when it serves to clarify the underlying meaning. Generally $p(x)$ denotes the probability density function of the random variable, x. $p(z|x)$ is a likelihood function of x or probabilty density funciton of z given x. $\widehat{x}$ is estimate and the expression that produces $\widehat{x}$ is an estimator.

2.2 Bayesian Estimation

In Bayesian approach the parameters are treated as random variables; that is, x is a random vector. It also assumes that the prior probability density function of the parameter is known. This consists of all known and unkown information about the parameter prior to the experiment–the evolving of the system dynamics. The observation vector z is also assumed random. It is so to account for random noise that the sensor themselves have introduced in the measurement process and to account for the unmodeled effects in modeling. The statistical characteristics are described by the probability density function of the observation vector conditioned on the parameter vector x, $p(z|x)$, also called the likelihood function of the parameter vector x. Thus when the state vector is known the pdf of the z is completely known. After receiving measurement z the knowledge about the vector x is updated so that

$$p(x|Z) = \frac{p(Z|x)p(x)}{p(Z)} \tag{2.1}$$

where

$$p(Z) = \int_{R^n} p(Z|x)p(x)dx$$

Equation (2.1) gives a solution to the Bayesian estimation problem which is definded by

$$p(x, Z) = p(Z|x)\, p(x) \tag{2.2}$$

where $p(x|Z)$ is the posterior pdf of x. Given the data Z any characteristics of x can be deduced from this posterior pdf. Therefore the posterior pdf is regarded as the general solution because we may want to know certain features of this pdf.

2.2.1 Bayesian Estimators

The posterior pdf is general and complex solution. Each value of x gives a value of $p(x|y)$ reflecting the posterior probability of that parameter value. Fortunately it is often desired to obtain an estimate of x rather than the whole posterior pdf. An expression mapping $R^p \to R^n$ is called an estimator which gives estimate $\widehat{x}$ optimal in some sense. In Bayesian framework an estimator is chosen as the function of observations that minimizes the *conditional Bayes risk*, that is, the expected value $E[C(\widetilde{x})|z]$ of a positive (semi)definite cost function C($\widetilde{x}$) of the estimaton error $\widetilde{x}$. In general, $C(\widetilde{x})$ is defined in such a way that for the larger the difference $x - \widehat{x}$, the greater the cost. The optimal choice of $\widehat{x}$, using a cost defined by $C(\widetilde{x})$, is the one that minimizes the conditional Bayesian risk,

$$\widehat{x} = \arg\min_{\widehat{x}} E[C(\widetilde{x})|z] \tag{2.3}$$

Since

$$E\left[C\left(\widetilde{x}\right)\right]=E\left\{E\left[C\left(\widetilde{x}\right)|z\right]\right\} \tag{2.4}$$

the estimator (2.3) also minimizes the Bayes risk $E\left[C\left(\widetilde{x}\right)\right]$. In general each cost function gives a different estimator. Many important estimation methods using the Bayesian framework are special cases of the above Bayesian estimator.

Minimum Mean Square Error Estimator

Let the cost function be $C\left(\widetilde{x}\right)=\widetilde{x}'\widetilde{x}$, where $\widetilde{x}$ is vector-value estimation error. Note that this cost function is in a quadratic form. $E\left[C\left(\widetilde{x}\right)|z\right]$ is the mean square error. Thus using the Bayesian formula (2.3)

$$\widehat{x}^{MMSE}\triangleq\arg\min_{\widehat{x}}\int_{-\infty}^{\infty}\left(\widetilde{x}'\widetilde{x}\right)p\left(\widetilde{x}|z\right)d\widetilde{x}$$

Taking derivative, using Leibuiz' rule of differentiation of the integral, and setting it to zero yields

$$2\int_{-\infty}^{\infty}\widetilde{x}p\left(\widetilde{x}|z\right)d\widetilde{x}=2\int_{-\infty}^{\infty}\left(x-\widehat{x}\right)p\left(x|z\right)dx=0$$

$$\widehat{x}^{MMSE}=E\left[x|z\right]=\int_{-\infty}^{\infty}xp\left(x|z\right)dx \tag{2.5}$$

Thus $\widehat{x}^{MMSE}$ is the conditional mean of the posterior pdf $p\left(x|z\right)$.

Maximum a Posteriori Estimator

Now if the cost function is chosen to be the one that penallizes all errors equally (also called "hit-or-miss" cost function)

$$C\left(\widetilde{x}\right)=\left\{\begin{array}{cc}0 & |\widetilde{x}|<\frac{\Delta}{2}\\ 1 & |\widetilde{x}|>\frac{\Delta}{2}\end{array}\right\}=1\left(|\widetilde{x}|-\frac{\Delta}{2}\right)\qquad 1\left(\cdot\right):\text{ unit step function}$$

given some small $\Delta>0$, then the Bayesian estimator is the maximum a posteriori (MAP) estimator as below (scalar case). Let $\epsilon=\frac{\Delta}{2}$

$$\begin{aligned}E\left[C\left(\widetilde{x}\right)|z\right]&=\int_{-\infty}^{\infty}C\left(\widetilde{x}\right)f\left(\widetilde{x}|z\right)d\widetilde{x}\\&=\int_{-\infty}^{-\epsilon}1\cdot f\left(\widetilde{x}|z\right)d\widetilde{x}+\int_{\epsilon}^{\infty}1\cdot f\left(\widetilde{x}|z\right)d\widetilde{x}\\&=\left(\int_{-\infty}^{-\epsilon}+\int_{\epsilon}^{\infty}+\int_{-\epsilon}^{\epsilon}\right)f\left(\widetilde{x}|z\right)d\widetilde{x}-\int_{-\epsilon}^{\epsilon}f\left(\widetilde{x}|z\right)d\widetilde{x}\\&=1-\int_{-\epsilon}^{\epsilon}f\left(\widetilde{x}|z\right)d\widetilde{x}\\&=1-\int_{\widehat{x}-\epsilon}^{\widehat{x}+\epsilon}f\left(x|z\right)dx\end{aligned}$$

This is minimized if $\int_{\widehat{x}-\epsilon}^{\widehat{x}+\epsilon} f(x|z)\,dx$ is maximized, and this is the case when $\widehat{x}$ is chosen to be the maximum point (mode) of $f(x|z)$. Thus

$$\widehat{x}^{MAP} = \lim_{\epsilon\to 0} \widehat{x}^{Bayes}|_{C(\widetilde{x})=1(|\widetilde{x}|-\epsilon)}$$

Conditional Median Estimator

Yet another common choice of cost function is the absolute (for scalar) value of the estimation error $C(\widetilde{x}) = |\widetilde{x}|$.

$$\begin{aligned}
E[C(\widetilde{x})|z] &= E[|x-\widehat{x}|\,|z] \\
&= \int_{-\infty}^{\infty} |x-\widehat{x}|\, f(x|z)\,dx \\
&= \int_{-\infty}^{\widehat{x}} (\widehat{x}-x)\, f(x|z)\,dx + \int_{\widehat{x}}^{\infty} (x-\widehat{x})\, f(x|z)\,dx
\end{aligned}$$

Taking derivative and setting it to zero yields

$$\int_{-\infty}^{\widehat{x}} f(x|z)\,dx - \int_{\widehat{x}}^{\infty} f(x|z)\,dx = 0$$

$$\rightarrow P\{x \le \widehat{x}|z\} = P\{x \ge \widehat{x}|z\}$$

Or $\widehat{x}$ is the conditional median of the conditional PDF $f(x|z)$.

2.2.2 Estimation Error Covariance

Though the estimator $\widehat{x}(Z)$ is random, the estimate is fixed. The estimate does not reveal its relative goodness, but its associated estimmation error covariance does, used to quantify the variation about the estimatee. The estimation error conrresponding to an estimate $\widehat{x}(Z)$ is defined by

$$\widetilde{x} \triangleq x - \widehat{x}$$

The unconditional mean-square error (MSE) matrix of estimator $\widehat{x}(Z)$ is defined by

$$P = MSE[\widehat{x}(Z)] \triangleq E\left[(x-\widehat{x})(x-\widehat{x})^T\right], \tag{2.6}$$

If $E[(x-\widehat{x})] = 0$ this quantity also coincides with the estimation error covariance. After observing the event $\{\mathbf{z} = z\}$ and computing a suitable estimate $\widehat{x}(Z)$ the (conditional) MSE matrix can be computed from the posterior pdf as

$$MSE[\widehat{x}(Z)|Z] \triangleq \int_{-\infty}^{\infty} (x-\widehat{x})(x-\widehat{x})^T p(x|Z)\,dx. \tag{2.7}$$

Since the conditional mean squared estimate (2.5) satisfies $E\left[\left(x-\widehat{x}^{MMSE}\right)|Z\right]=0$, it has an error covariance matrix

$$P^{MMSE}=Cov\left(x-\widehat{x}|Z\right)=Cov\left(x|Z\right) \tag{2.8}$$

Hence, the first and second central moments of the posterior density function $p\left(x|Z\right)$ are the MMSE estimate and the estimation error covariance, repectively. Note that 2.8 gives the error covariance of the MMSE estimate based on the observation Z and it quantifies the estimation error of this estimate. The error covariance of the estimator can be found by computing expectation over z as well, and will quantify the overall performance of the estimator, prior to collecting any observations.

2.3 Recursive Bayesian Estimation

In estimation many applications deal with problems that requires determining state of the system on-line or in other words estimating as the data come in time. An example of sush a system is tracking a target in real time. It is so either because the systems have dynamics that make the state change with time or because the estimate of nonrandom parameters are needed on-line demanding the information about the parameter or state to be updated with each new measurement. Like in target tracking the position and velocity of the motion need to be updated frequently. Other examples of recursive estimation are all types of navigation applications ranging from global aircraft and ship navigation to autonomous robot navigation. Recursive estimation can also be applied to adaptive control in which the parameters of a dynamical system working in a closed loop are tracked using measurements of system input and output. One of the main reason for using recursive estimation is the advantage in data storage, thus one may also resort to recursive estimation in off-line application with large amount of data. In cases when the density $p\left(x|z\right)$ is very complicated due to the size of the measurement vector z, treating the problem recursively can yield a reasonable tradeoff between computational complexity and performance.

Let x_k denote the state at time k, where we always let the time index $k\in N$ independently of the acutal time evolved. T will be used for the fixed sampling interval.

Recursive estimation usually assumes the system is Markov process by which given the measurement observed at time k is conditionally independent of the previously observed measurements given the current state value. Hence, conditioned on the present state no additional information about the future states should be available in past observations.

A recursive estimation problem is an estimation problem where the state evolves in time according to a Markov process with an initial state $x_0\sim p\left(x_0\right)$ and transition kernel,

$$p\left(x_{k+1}|x_k\right) \qquad k=0,1,...$$

This transition kernel may explicitly depend on the time index.

The likelihood

$$p\left(z_k|x_k\right) \qquad k = 0, 1, ...$$

may explicitly depend on the time index k.

A recursive estimation problem is uniquely specified when the prior $p\left(x_0\right)$, the transition kernel $p\left(x_{k+1}|x_k\right)$ and the likelihood $p\left(z_k|x_k\right)$ are given. Note that this coincides with the problem described above since these entities define the joint density of all measurements and all states from time zero to time k.

2.3.1 Conceptual Solution

All recursive estimation problems that can be expressed in the form formulated above have a common conceptual solution consisting of a recursive propagation of the conditional density. The stacked vector of the complete measurement history at time k has length $(k+1)$p and is denoted by

$$Z^k = \left[z_0, z_1, ..., z_k\right]^T$$
$$Z_{s:k} = \left[z_s, z_{s+1}, ..., z_k\right]^T$$

for any $s < k$. This notation will be adopted generally for stacked vectors of random process. Applying Bayes' rule (2.1) to the last element of the measurement vector Z^k yields

$$p\left(x_k|Z^k\right) = \frac{p\left(z_k|x_k, Z^{k-1}\right)p\left(x_k|Z^{k-1}\right)}{p\left(z_k|Z^{k-1}\right)} \tag{2.9}$$

$$= \frac{p\left(z_k|x_k\right)p\left(x_k|Z^{k-1}\right)}{p\left(z_k|Z^{k-1}\right)} \tag{2.10}$$

since, by $p\left(z_k|x_k\right)$, we assume that the observation z$_k$ is conditionally independent of previous measurements given the state x$_k$. The expression (2.9) is referred to as the *measurement update* in the Bayesian recursion. As in (2.1) the denominator can be expressed through the law of total probability, i.e., by marginalization.

The effect of a time step is obtained by observing that

$$\begin{aligned} p\left(x_{k+1}, x_k|Z^k\right) &= p\left(x_{k+1}|x_k, Z^k\right)p\left(x_k|Z^k\right) \\ &= p\left(x_{k+1}|x_k\right)p\left(x_k|Z^k\right) \end{aligned}$$

which follows from the assumption that the process $\{x_k\}$ is Markov, and that x_{k+1} is independent of Z^k when x_k is given. Intergrating both sides with respect to x_k yields the *time update* equation

$$p\left(x_{k+1}|Z^k\right) = \int_{R^n} p\left(x_{k+1}|x_k\right)p\left(x_k|Z^k\right)dx_k \tag{2.11}$$

After (2.11) has been evaluated, the time index can be increased and the effect of a new measurement incorporated as in (2.9). To summarize, a recursive propagation of the posterior filter density of the states given the measurements is obtained. (2.9) and (2.11) provides a mechanism to do recursion that is initialized by $p\left(x_0\right)$.

2.4 Conclusion

Bayesian framework (2.1) provides a mechanism for estimation problem that is assumed to have a prior density of the unknown and the likelihood function through observation. the posterior can be viewed as a general solution to estimation problem, but it is a rather complex and inconvenient solution since most problems encountered need to deduce only certain features of the posterior. Bayes risk or cost function is introduced to provide users with a way to specify optimality criterion for estimation. Recursive solution consists of 2 steps: time update and measurement update that facilitates updating the posterior and prior PDFs.

Chapter 3

Recursive Bayesian Filtering

This chapter gives a brief review of the approaches to recursive Bayesian filtering in the literature. First, consider the general system and observation models below

$$x_{k+1} = f(x_k, v_k) \qquad k = 1, 2, ... \tag{3.1}$$

$$z_k = h(x_k, w_k) \tag{3.2}$$

where x_k is the state vector and v_k and w_k are system and measurement noise, respectively.

When both f and h are linear functions, the problem at hand is a linear problem otherwise it is a nonlinear one. For a linear problem that has prior, process noise, and measurement noise being Gaussian, the problem is classified as a linear Gaussian problem. This kind of problem presents a very important classs that can be tractable in recursive Bayesian estimation framework and is also easily implementable by Kalman filter. For non-linear problem several classes of estimators have been developed. Exended Kalman filter (EKF) and its variants linearize the system function and/or observation funciton at each time step and use the Kalman filter to do the filtering. Another class deals with nonlinear non-Gaussian problems by approximating the posterior density of the state by mixture distributions. Yet another class deals with nonlinearity and non-Gaussianity by discrete approximation of the posterior density. These filter are called grid-based filters. They associate with each grid point a probability value. The last class is Monte Carlo methods. They do not approximate the posterior density directly, but sample it.

The chapter is organized as follows. Section 3.1 discusses linear recursive filtering, the Kalman filter, and gives the algorithm. Sections 3.2, 3.3, and 3.4 reviews the EKF, Gaussian-mixture filter, and grid-based filter, respectively. Monte Carlo Methods which are the main focus of this thesis are discussed in details in the next chapter.

3.1 Kalman Filter

Under linear Gaussian assumption, the initial state, process noise, and observation noise are Gaussian and mutually independent, and, as a result of this and Markov assumption of the state, the prior and the posterior desities are Gaussian themselves at each time step. The Kalman filter is an algorithm to track the evolution of these two densities by their first two moments: the conditional mean and the conditional covariance. Thus it is an optimal MMSE estimator. The following is the Kalman filter algorithm based on ([2], chapter 5) from which the derivation can be found.

With linearity assumption the state space model can be rewritten as

$$\begin{aligned} x_{k+1} &= F_k x_k + G_k u_k + \Gamma_k v_k \qquad k = 0, 1, ... \\ z_k &= H_k x_k + w_k \end{aligned}$$

where x is $n \times 1$ state vector, F $n_x \times n_x$ system matrix, G_k $n_x \times n_u$ input gain matrix, u_k $n_u \times 1$ input vector, Γ $n_x \times n_v$ gain matrix, z_k $n_z \times 1$ observation vector, H $n_z \times n_x$ observation matrix. the initial state is Gaussian with known mean and covariance. The process noise v_k is Gaussian with zero mean and covarince Q, and v_k and v_l are uncorrelated for $k \neq l$. Observation noise $n_z \times 1$ vector w_k is Gaussian with zero mean and covariance R, and w_k and w_k are uncorrelated for $k \neq l$.

$$\widehat{x}_{k+1|k} = F_k \widehat{x}_{k|k} + G_k u_k \tag{3.3}$$

$$\widehat{z}_{k+1|k} = H_{k+1} \widehat{x}_{k+1|k} \tag{3.4}$$

$$\widehat{x}_{k+1|k+1} = \widehat{x}_{k+1|k} + W_{k+1} \left(z_{k+1} - \widehat{z}_{k+1|k} \right) \tag{3.5}$$

The covariance computation is as

$$P_{k+1|k} = F_k P_{k|k} F_k' + \Gamma_k Q_k \Gamma_k' \tag{3.6}$$

$$S_{k+1} = R_{k+1} + H_{k+1} P_{k+1|k} H_{k+1}' \tag{3.7}$$

$$W_{k+1} = P_{k+1|k} H_{k+1}' S_{k+1}^{-1} \tag{3.8}$$

$$P_{k+1|k+1} = P_{k+1|k} - W_{k+1} S_{k+1} W_{k+1}' \tag{3.9}$$

Since the covariance calculation is independent of the state and measurements, it can be performed offline. If the noises are not Gaussian, Kalman filter is a *best linear state estimator.*

3.2 Extended Kalman filter

Extended Kalman filter is the most widely used filter to deal with nonlinear recursive filteing. Its assumptions about the state model are the same as that of the Kalman filter. EKF deals with nonlinearity by linearizing the nonlinear system and/or observation functions using Taylor's series expansion about the prediction state and about the observation prediction at every time step; that is, it calculates the jacobian matrix and sets

$$\begin{aligned} F_k &= \left. \frac{\partial f_k}{\partial x} \right|_{x=\widehat{x}_{k|k}} \\ H_{k+1} &= \left. \frac{\partial h_{k+1}}{\partial x} \right|_{x=\widehat{x}_{k+1|k}} \end{aligned} \tag{3.10}$$

These evaluations are inserted into the algorithm of the Kalman filter just before the calculation of the state prediction covariance. As can be seen in the expression above these evaluations of the Jacobians call for the predicted state and observation and thus they, and the covariance computation, must be computed on-line.

Two limitations of the EKF are the error from the linearization and the approximation of the pdfs by their mean and covariance. If the linearity is severe, the linearization error is significant and EKF could diverge. It is clearly a suboptimal state estimator. Despite of these drawbacks it is widely used.

3.3 Approximation by Mixture Density

As computing power has increased, there has been a trend to solve nonlinear recursive state estimation by approximating the prior and posterior distribution by mixture distributions, as opposed to estimating just first two moments as in the case of the EKF. Most filters of this class use Gaussian mixtures to approximate the densities. These would use either the Kalman filter or EKF as the main tool in updating the mixture distributions. The mostly used of these is the interacting multiple model (IMM) which is briefly reviewed below. A thorough discusion, derivation, and complete algorithm of IMM filter can be found in [16] upon which the review below is based.

3.3.1 Mixture Density

A mixture density is a weighted sum of, say, M distributions

$$p\left(x\right) = \sum_{i=1}^{M} \mu_i p_i\left(x\right) \qquad \text{with } \sum_{i=1}^{M} \mu_i = 1 \tag{3.11}$$

where μ_i is the probability of the event that model i being the true one. (3.11) can be rewritten as

$$p(x) = \sum_{i=1}^{M} p\left(x|A_i\right) p(A_i)$$

The mean of the mixture, by the total expectation theorm, is

$$\overline{x} = \sum_{i=1}^{M} \mu_i \overline{x}_i$$

And the covariance of the mixture is

$$E\left[\left(x-\overline{x}\right)\left(x-\overline{x}\right)'\right] = \sum_{i=1}^{M} P_i \mu_i + \widetilde{P}$$

$$\widetilde{P} \triangleq \sum_{i=1}^{M} \left(\overline{x}_i - \overline{x}\right)\left(\overline{x}_i - \overline{x}\right)' \mu_i = \sum_{i=1}^{M} \overline{x}_i \overline{x}_j \mu_i - \overline{x}\overline{x}'$$

A mixture density, for example Gaussian mixture, can be approximated by, i.e., a single Gaussian distribution.

3.3.2 Interacting Multiple Model Filter

Different from those filters described previously which are intended for systems that can be modeled by one mathematical model, interacting multiple model filter (IMM) is usually used for hybrid system which has both discrete state and continuous state. An example of this kind is a maneuveing target whose motion can be modeled not by a single model but by a group of models. In the IMM framework a number of elemental filters, usually KF or EKF, run in parallel each corresponding to a mode of the motion. Since the optimal multiple model estimator is not implementable due to the large number of model-path combinations to keep track, IMM is designed to limit the number of model-path histories finite and thus is a suboptimal estimator.

IMM has a soft decision mechanism by which the estimate is a joint estimate by all filters. The feature of IMM that makes it efficient is the model-conditional reinitialization which is done by mixing all latest model-conditional estimates whose weights are the mixing weights:

$$\mu^j_{k|k-1} \triangleq P\left\{m^i_{k-1} | m^j_k, z^{k-1}\right\} = \frac{\pi_{ij}}{\widehat{\mu}^j_{k|k-1}} \mu^i_{k-1}$$

where π_{ij} is the transition probability and $\widehat{\mu}^j_{k|k-1}$ is the predicted mode j probability:

$$\widehat{\mu}^j_{k|k-1} \triangleq P\left\{m^j_k | z^{k-1}\right\} = \sum_{i=1}^{M} \pi_{ij} \mu^i_{k-1}$$

The mode probabilities for estimate combination are, for $j = 1, 2, ..., M$

$$\mu^j_k \triangleq P\left\{m^j_k | z^k\right\} = \frac{L^j_k \widehat{\mu}^j_{k|k-1}}{\sum_{j=1}^{M} L^j_k \widehat{\mu}^j_{k|k-1}}$$

where L^j is the likelihood function of model j.

3.4 Grid Based Filters

Grid based approach is a numerical method that represents the distributions as numbers on a fixed grid. The grid is simply a large series of points and to each grid-point is associated a number which is the density of the distribution in the grid-point. Suppose that we want to represent the distribution $p(x)$ of the $(p \times 1)$ vector x in this way. Then $p(x)$ is lying in some p-dimetional space and if we use n points in each direction of the space we need n^p grid-points. Even for a modest value of n the number of grid-points becomes enormous if the dimention p is just less then 10. Typically p is much greater than this, so the method becomes useless in many practical applications, even when one is equipped with the fastest computers. If we use a fixed grid it must cover the whole area where the distribution may be. The number of points in each direction of the p-directional space must then be so large that the method is limited to problems where p is even smaller than 5. Progress in the field of nonlinear state estimation has up to now been stucked by this drawback, known as the "curse of dimentionality". It is obvious that using of a fixed grid we keep calculating the density in a large number of points where the density is zero for all practical purposes. In fact, the set of points of interest where the density is nonzero is often very small compared to the set of points where the density is zero. So, much work is spent on calcualtions of no interest.

3.5 Conclusion

Kalman filter is nothing but a linear MMSE estimator. It calculates first two moments of the prior and posterior PDFs. If the problem at hand is linear Gassian, KF gives a optimal solution (MMSE) otherwise the solution is best in the linear class. EKF uses linearization of the nonlinear function(s) and then utilizes Kalman filter to do filtering; it could diverge if the linearity is severe. Density mixture approach use a group of filter to approxiamte nonlinear and nonGaussian PDFs. Grid based method is infeasible for most practical problems due to its "curse of dimensionality".

Chapter 4

Sampling Based Methods

Sampling based (or Monte Carlo) methods are based on numerical intergration. They sample (particles) from the posterior. This comes from the fact that since the whole posterior approximation is hard or infeasible such as suffering from the "curse of dimentionality" in the case of grid based methods, in estimation one needs only some features, usually mean and covariance, of the posterior that can be deduced form random variables. Simulation based techniques were introduced early in the literature, but only recently have they been practically applied to problems of statistical inference. Particle filter (or bootstrap filter) was the first functional filter of this class and was introduced in the early 1990 [3].

This chapter provides a review of general numerical intergration in section 5.1, two sampling techniques in section 5.2, the derivation of particle filter (PF) and improvements on particle filter are considered in section 5.3. In section 5.4, unscented transform (UT) followed by the algorithm of unscented Kalman filter (UKF) are presented. Finally unscented particle filter (UPF) algorithm is in section 5.5. PF, UKF, and UPF are those that will be compared in the next chapter. The algorithms of this chapter are gathered largely from [3] and [9], and the section on numerical integration and *importance sampling* methods is based on chapter 6 of [18].

4.1 Numerical Integration

Numerical integration deals with the problem of numerically evaluating general integrals,

$$I = \int_{R^n} g\left(x\right) dx. \tag{4.1}$$

Monte Carlo methods for numerical integration regard problems of the form

$$I = \int_{R^n} f\left(x\right) \pi\left(x\right) dx, \tag{4.2}$$

where

$$\pi\left(x\right) \geqslant 0, \qquad \int_{R^n} \pi\left(x\right) dx = 1.$$

Conceptually (4.1) , which may be difficult, if not impossible, to evaluate, can be decomposed or transformed into an integral for Monte Carlo evaluation through a suitable factorization of the integrand $g(x) = f(x)\pi(x)$. The assumptions on the factor $\pi(x)$ impose a natural interpretation of $\pi(x)$ as a probability density function. To interprete (4.2) in Bayesian estimation context, this density can be the filtering PDF (posterior density) of the parameters given the observed data, i.e., $\pi(x) = p(x|z)$, and considering, e.g., the MMSE estimator, we identify $f(x) = x$, and $I = \widehat{x}^{MMSE}$.

The Monte Carlo methods rely on the assumption that it is possible to draw $N \gg 1$ samples $\{x^i\}_{i=1}^{N}$ distributed according to the probability density $\pi(x)$. Two sampling techniques are presented in section 4.2. The Monte Carlo estimate of the integral (4.2) is formed by taking the average over the set of samples

$$f_N = \frac{1}{N}\sum_{i=1}^{N} f\left(x^i\right) \tag{4.3}$$

where N is assumed to be large. In essence Monte Carlo simulation uses a set of weighted particles to map integrals to sums. the *law of large numbers* guarantees convergence

$$\lim_{N\to\infty} f_N = I. \tag{4.4}$$

The convergence result above is asymptotic. This means that as $N \to \infty$ the error of the approximation will tend to zero. With support from this asymptotic result we usually assume that a large but finite N will lead to a small error. In practical applications the number of samples might have to be very large for a given error bound. Advantages of Monte Carlo sumulation methods is that while grid based methods suffer from the "curse of dimentionality" the expression (4.4) yields [18] that the error $\varepsilon = f_N - I$ of the Monte Carlo estimate is of the order

$$\varepsilon = O\left(N^{-1/2}\right), \tag{4.5}$$

and in general does not increase proportionaly with x. Moreover, while the numerical intergration methods require the user to define a grid over the integration area that naturally is dependent of the integrand, the estimate (4.3) is obtained using the same technique for any function $f(x)$.

4.2 Sampling Techniques

The Monte Carlo integration in the previous section is quite general. In fact what recursive Bayesian estimation faces is difficulties in sampling posterior density directly. Therefore to apply Monte Carlo simulation to recursive Bayesian estimation, there must be sampling schemes; that is, it boils down to how actually the sampling can be implemented. in this section two general sampling techniques are presented: rejection sampling and importance sampling.

4.2.1 Rejection Sampling

This sampling assumes that an upper bound on the range of the generic density $\pi(x)$ exist and is known and that it is possible to evaluate $\pi(x)$ everywhere up to a normalizing constant. Assume that the upper bound is $M < \infty$ such that $\pi(x) < Mq(x)$ where $q(x)$ is a proposal ditribution, rejection sampling is as follows.

Rejection Sampling Algorithm (algorithm 6.1 of **??**)

1. Draw $x^* \sim q(x)$ and $u \sim U(0,1)$.
2. If $u < \frac{\pi(x^*)}{Mq(x^*)}$ accept $x = x^*$ otherwise goto step 1.

x drawn from $\pi(x)$ since

$$\begin{aligned}\Pr(x^* \leq t, x^* \text{ is accepted}) &= \int_{-\infty}^{t} \frac{\pi(x)}{Mq(x)} q(x)\,dx \\ &= \frac{1}{M}\int_{-\infty}^{t} \pi(x)\,dx\end{aligned}$$

The acceptance probability of a generic x^* is

$$\begin{aligned}\Pr(x^* \text{ is accepted}) &= \frac{1}{M}\int_{-\infty}^{\infty} \pi(x)\,dx \\ &= \frac{1}{M}\end{aligned}$$

Therefore,

$$\Pr(x^* \leq t | x^* \text{ is accepted}) = \int_{-\infty}^{t} \pi(x)\,dx$$

which shows that the accepted x^* is sampled from $\pi(x)$.

The drawback with this technique is efficiency: there is no way to determine how long this process takes, in fact it could be infinite.

4.2.2 Importance Sampling

Importance sampling assumes that there exists a proposal distribution $q(x)$ which is easy to draw a samples from, and it also assumes that the support set of $q(x)$ covers the support of $\pi(x)$, i.e., that $\pi(x) > 0 \rightarrow q(x) > 0$ for all $x \in R^n$. Under this assumption, any integral of the form (4.2) can be rewritten

$$I = \int_{R^n} f(x)\pi(x)\,dx = \int_{R^n} f(x)\frac{\pi(x)}{q(x)}q(x)\,dx$$

A Monte Carlo estimate of I is computed by generating $N \gg 1$ independent samples from $q\left(x\right)$, and forming the weighted sum

$$f_N = \frac{1}{N}\sum_{i=1}^{N} f\left(x_i\right) w\left(x_i\right), \quad \text{where} \qquad w\left(x_i\right) = \frac{\pi\left(x_i\right)}{q\left(x_i\right)} \tag{4.6}$$

are the *importance weights.* It is straightforward to verify that (4.4) is satisfied for the importance sampling Monte Carlo estimate (4.6) .

If the normalizing factor of the target density $\pi\left(x\right)$ is unknown, the importance weights in (4.6) can only be evaluated up to a normalizing factor. Then, the weights can be formed using a function propotional to the target density and then normalized afterward, forming the estimate

$$f_N = \frac{\sum_{i=1}^{N} f\left(x_i\right) w\left(x_i\right)}{\sum_{j=1}^{N} w\left(x_j\right)} \qquad \text{where} \qquad w\left(x_i\right) \propto \frac{\pi\left(x_i\right)}{q\left(x_i\right)} \tag{4.7}$$

The estimate f_N is biased for finite N, but asymptotically both a law of large numbers and a central limit theorem hold.

4.3 Particle Filter

Particle filter (PF) is an algorithm that applies Monte Carlo numerical integration to recursive Bayesian estimation. It uses importance sampling technique to obtain particles' importance weights. Equation (4.6) can be rewritten as

$$\begin{aligned} f_N &= \frac{1}{N}\sum_{i=1}^{N} f\left(x_i\right) w\left(x_i\right), \quad \text{where} \qquad w\left(x_i\right) = \frac{\pi\left(x_i\right)}{q\left(x_i\right)} \\ &= \frac{1}{N}\sum_{i=1}^{N} f\left(x_i\right)\frac{\pi\left(x_i\right)}{q\left(x_i\right)} \\ &= \frac{1}{N}\sum_{i=1}^{N} f\left(x_i\right)\frac{p\left(z|x_i\right)p\left(x_i\right)}{p\left(z\right)}\frac{1}{q\left(x_i\right)}, \qquad \pi\left(x\right) = p\left(x|z\right) \\ &= \frac{1}{N}\sum_{i=1}^{N} f\left(x_i\right)\frac{p\left(z|x_i\right)p\left(x_i\right)}{\frac{1}{N}\sum_{j=i}^{N} p\left(z|x_j\right)}\frac{1}{q\left(x_i\right)} \end{aligned}$$

If the prior density is the proposal density $p\left(x\right) = p\left(x\right)$, then we have

$$\begin{aligned} f_N &= \frac{1}{N}\sum_{i=1}^{N} f\left(x_i\right)\frac{p\left(z|x_i\right)}{\frac{1}{N}\sum_{j=i}^{N} p\left(z|x_j\right)} \\ &= \sum_{i=1}^{N} f\left(x_i\right) w_i \end{aligned}$$

where $w_i = \frac{p(z|x_i)}{\sum_{j=i}^{N} p(z|x_j)}$. In recursive Bayesian estimation, $f(x_i)$ is the sample x_i drawn from the pdf of the state x, andthe mean of the posterior pdf is

$$\widehat{x} = \sum_{i=1}^{N} w_i x_i$$

Particle filter incorporates resampling step for generating approximately independent samples drawn from $\pi(x)$ after calculating the weights, as in the algorithm below.

SIS/R Algorithm

```
Initialization $(k = 0)$
    For $i = 1, ..., N$
        set $w_0^{(i)} = 1/N$ and draw $x_0^{(i)} \sim q(x_0)$.
    For $k = 1, 2, ...$
      * Importance Sampling step
          For$i = 1, ..., N$
              - Draw a sample $x_k^i \sim q\left(x_k|x_{k-1}^{(i)}, z^k\right)$
              - Evaluate (unnormalized) importance weights
                    $\widetilde{w}_k^i = w_{k-1}^i \frac{p\left(z_k|x_k^{(i)}\right) p\left(x_k^{(i)}|x_{k-1}^{(i)}\right)}{q\left(x_k^{(i)}|x_{k-1}^{(i)}, z^{k-1}\right)}$
          For $i = 1, ..., N$
              - Normalize the importance weights
                    $w_k^i = \frac{\widetilde{w}_k^i}{\sum_{j=1}^{N} \widetilde{w}_k^j}$
      * Resampling step
          Effective sample size estimation
                    $\widehat{N}_{eff} = \frac{1}{\sum_{j=1}^{N} \left(w_k^j\right)^2}$
          If $\widehat{N}_{eff} < N_{th}$
              - Obtain new samples $\left\{x_k^{(j_i)}\right\}_{i=1}^{N}$ by resampling $N$ times with replacement
                from $\left\{x_k^{(j_i)}\right\}_{i=1}^{N}$ such that $\Pr\left\{x_k^{(j_i)} = x_k^{(j)}\right\} = w_k^j$
              - Reset $w_k^i = 1/N$
      * Output (optional)
                    $\widehat{x}_{k|k} = \sum_{i=1}^{N} w_k^i x_k^{(i)}$
```

The above SIS/R filtering scheme is quite general. The importance distribution q is a "design parameter" and it permits quite a lot of choices to influence the performance of any specific application. Another important feature of the above scheme is the on-line dectection of degeneracy and the use of resampling for its mitigation. In the particular case of choosing q to be the prior $p\left(x_k|x_{k-1}^i\right)$ and with resampling at every k the above generic particle filter reduces to the original SIR (Bayesian bootstrap) algorithm of [3].

One cycle of the algorithm is shown in Figure 4.3.

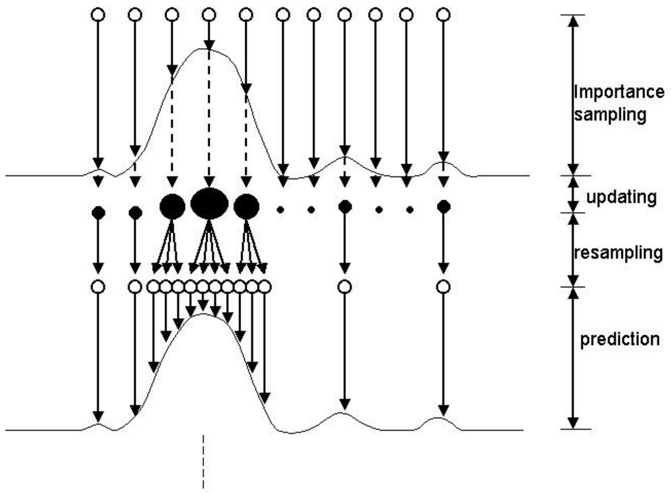

Figure 1: One cycle of the SIS/R algorithm

Resampling on the other hand usually leads to another problem, known as sample impoverishment, which, simply put, means replicating high-probability samples and thus losing the low-probability samples. An optional technique to deversify the samples from the posterior is to use the MCMC *move* (MH-step) after an eventual resampling in the above SIS/R scheme [15].

4.3.1 Improvements on Particle Filter

The problem with particle filter is particle depletion [9] in which resampling causes high-probability particles to be replicated thus losing the low-probability particles. This problem can be so severe that only few recursions one particle dominates the particle cloud. Also it is important to have a good proposal distribution from which to sample particles.

A number of algorithms have been proposed which can be lumped into two categories: MCMC move and better proposal function design. Among these better proposal function design by localization [9] gives improving results, and unscented particle filter is the result of this effort which will be discussed in section 4.5.

4.4 Unscented Kalman Filter

While the extended Kalman filter (EKF) linearizes nonlinear function(s) using Taylor's series expansion, the unscented Kalman filter (UKF) uses the nonlinear function(s) and approximates the state distribution by a Gaussian distribution. It is based on the principle [9] that it is easier to approximate a Gaussian distribution than to approximate nonlinear function.

Unlike the PF which approximates the entire PDFs by a large number of randomly sampled points, UKF is based on approximating the first and second moments of the PDFs by (a small number of) deterministic sample points[1], referred to as sigma points. These points are designed by using *unscented transformation* (UT). They are designed in a way that the mean and covariance of a nonlinearly transformed random variable can be approximated by the sample mean and sample covariance of the transformed sigma points for any nonlinearity [6].

4.4.1 Unscented Transformation

Unscented transformation is a method for calculating the statistics (first two moments) of a random variable which undergoes a nonlinear transformation. It computes (or deterministically samples) $(2n_x + 1)$ sigma points and their associated weights using the formulars below, where n_x is the dimention of state vector.

In the UKF algorithm the weights associated with sigma points are given by
$x^0 = \overline{x}$ $\qquad W_0 = \lambda / (n_x + \lambda) + (1 - \alpha^2 + \beta)$
$x^i = \overline{x} \pm \left[\sqrt{(n_x + \lambda) P} \right]_i$ $\qquad W_i = 1/(2(n_x + \lambda))$ $\qquad i = 1, 2, ..., 2n_x$
with $\lambda = \alpha^2 (n_x + \kappa) - n_x$, where α, β, and κ are design parameters.

These sigma points then propagate through the nonlinear function $\varsigma^i = g(\chi^i)$, and the sample mean and covariance can be computed as

$$\overline{\varsigma} = \sum_{i=0}^{2n_x} W^i \varsigma^i$$

$$P_\varsigma = \sum_{i=0}^{2n_x} W^i \left(\varsigma^i - \varsigma \right) \left(\varsigma^i - \varsigma \right)'$$

4.4.2 Unscented Kalman Filter Algorithm

UKF basically applies unscented transformation to recursive Bayesian estimation.

Algorithm UKF

[1] In this regard UKF is considered as a sampling based (quasi-Monte Carlo) approximation approach.

```
Initialization $(k=0)$
        $\widehat{x}_0 = E[x_0], \quad P_0 = E\left[(x_0-\widehat{x}_0)(x_0-\widehat{x}_0)'\right]$
For $k = 1, 2, ...$

    1 Calculate $2n_x+1$ sigma points[2]

        For $i = 0, \pm 1, \pm 2, ..., \pm n_x$
        $x_{k-1}^0 = \widehat{x}_{k-1|k-1}$
        $\chi_{k-1}^i = \widehat{x}_{k-1|k-1} \pm \left[\sqrt{(n_x+\lambda) P_{k-1|k-1}}\right]_i$

    2 Prediction

        2.1 State
        $\chi_{k|k-1}^i = f\left(\chi_{k-1}^i\right), \quad i = 0, \pm 1, \pm 2, ..., \pm n_x$
        $\widehat{x}_{k|k-1} = \sum_{i=-n_x}^{n_x} W_i \chi_{k|k-1}^i$
        $P_{k|k-1} = \sum_{i=-n_x}^{n_x} W_i \left[\chi_{k|k-1}^i - \widehat{x}_{k|k-1}\right]\left[\chi_{k|k-1}^i - \widehat{x}_{k|k-1}\right]' + \Gamma_k Q_k \Gamma_k'$
        2.2 Measurement
        $\varsigma_{k|k-1}^i = h\left(\chi_{k|k-1}^i\right), \quad i = 0, \pm 1, \pm 2, ..., \pm n_x$
        $\widehat{z}_{k|k-1} = \sum_{i=-n_x}^{n_x} W_i \varsigma_{k|k-1}^i$

    3 Update

        $C_{\widetilde{z}_k,\widetilde{z}_k} = \sum_{i=-n_x}^{n_x} W_i \left[\varsigma_{k|k-1}^i - \widehat{z}_{k|k-1}\right]\left[\varsigma_{k|k-1}^i - \widehat{z}_{k|k-1}\right]' + R_k$
        $C_{\widetilde{x}_k,\widetilde{z}_k} = \sum_{i=-n_x}^{n_x} W_i \left[\chi_{k|k-1}^i - \widehat{x}_{k|k-1}\right]\left[\varsigma_{k|k-1}^i - \widehat{z}_{k|k-1}\right]'$
        $K_k = C_{\widetilde{x}_k,\widetilde{z}_k} C_{\widetilde{z}_k,\widetilde{z}_k}^{-1}$
        $\widehat{x}_{k|k} = \widehat{x}_{k|k-1} + K_k\left(z_k - \widehat{z}_{k|k-1}\right)$
        $P_{k|k} = P_{k|k-1} - K_k C_{\widetilde{z}_k,\widetilde{z}_k}^{-1} K_k'$
```

4.5 Unscented Particle Filter

Producing a good proposal distribution is critical for the performance of the PF. The UPF of [9] utilizes UKFs to produce better proposal densities $q\left(x_k|x_{k-1}^{(i)}, z^k\right)$ within the framework of the PF. Specifically, each particle is updated by a UKF and the output mean and covariance are used to sample new particles.

Algorithm UPF

```
Initialization $(k=0)$

    For $i = 1, ..., N$
```

[2] n_x denotes the dimention of the state vector, and $[A]_i$ – the ith row (column) of $A^{1/2}$

```
        set $w_0^{(i)} = 1/N$ and draw $x_0^{(i)} = p\left(x_0\right).$
    For $k = 1, 2, ...$

        * IS step

            For $i = 1, ..., N$
                - On each $\mathrm{x}_{k-1}^i$ apply a UKF to obtain $\overline{x}_k^i$, $P_k^i$
                - Draw $\widehat{x} \sim q\left(x_k|x_{k-1}^{(i)}\right) = \mathrm{N}\left(\overline{x}_k^i, P_k^i\right)$
                - Evaluate (unnormalized) importance weights
                    $\widetilde{w}_k^i = w_{k-1}^i p\left(z_k|x_k^{(i)}\right)$
            For $i = 1, ..., N$
                - Normalize the importance weights
                    $w_k^i = \frac{\widetilde{w}_k^i}{\sum_{j=1}^N \widetilde{w}_k^j}$

        * R step

            - Obtain a new set of samples $\left\{x_k^{(i)}\right\}_{i=1}^N$ by resampling N times with
              replacement from $\left\{x_k^{(i)}\right\}_{i=1}^N$ such that $\Pr\left\{x_k^{(i)} = \widehat{x}_k^{(j)}\right\} = w_k^j$
            - reset $w_k^i = 1/N$.

        * Output (optional)

                $\widehat{x}_{k|k} = \sum_{i=1}^N w_k^i x_k^{(i)}$
```

4.6 Conclusion

Particle filter is truely a Monte Carlo simulation method. It samples random variables from, theoretically, posterior PDF. PF has its own weakness: particle depletion. On the other hand, unscented Kalman filter uses sigma points to capture the first two moments of the posterior, and computes these first two moments after nonlinear transformation using the designed sigma points. The UPF is in essence a particle filter with better proposal distribution: using one UKF for each particle.

Chapter 5

Multisensor Distributed Target Tracking

In estimation there are situations that need to use multiple sensors to gather (measure) data. Those observations are then sent to a fusion center to fuse (estimate) unknown states. These sensors may be located on the same platform, e.g., sensors on an airplane for navigation, or they can be located remotely, such as tracking an airplane using several geographically dispersed radars.

There are two basic architectures for fusion: centralized and distributed (also called decentralized) fusion depending on what type of data are sent to the fusion center. If raw data or measurements are collected and sent to the fusion center, it is the centralized fusion architecture. In distributed fusion, local sensors collect observations, filter them locally, and finally send the local estimates to the fusion center.

This chapter considers the scenario of tracking a target in a multisensor environment with distributed fusion architecture. First the mathematical model of target motion is presented, second the fusion scheme at the fusion center is formulated, third details how to calcualte the cross covariances between local estimates at the fusion center is discussed, and finally the performance of PF, UKF, and UPF is compared.

5.1 Multisensor Tracking Scenario

The scenario considered is a maneuvering target tracked by multiple distributed sensors. Since sampling based methods are mainly used for nonlinear estimation, the target motion is modeled by nonlinear function of *nearly constant turn*, and the measurements are measured in polar coordinates thus the measurement function is also nonlinear.

5.1.1 Target Model

The target motion is described by the nonlinear discrete-time nearly constant turn (CT) model [10]

$$\mathbf{x}_k = f\left(\mathbf{x}_{k-1}\right) + \Gamma_k \mathbf{w}_k, \qquad k = 1, 2, ... \tag{5.1}$$

where the target state vector $x_k = \left[x, \dot{x}, y, \dot{y}, \omega\right]'$ consists of the position and velocity components and the turn rate ω, $w_k \sim N\left(0, Q_k\right)$ is white process noise, and

$$f(\mathbf{x}_{k-1}) = \begin{bmatrix} 1 & \frac{\sin\omega(k)T}{\omega(k)} & 0 & -\frac{1-\cos\omega(k)T}{\omega(k)} & 0 \\ 0 & \cos\omega(k)T & 0 & -\sin\omega(k)T & 0 \\ 0 & \frac{1-\cos\omega(k)T}{\omega(k)} & 1 & \frac{\sin\omega(k)T}{\omega(k)} & 0 \\ 0 & \sin\omega(k)T & 0 & \cos\omega(k)T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1},$$

$$\Gamma_k = \begin{bmatrix} \frac{1}{2}T^2 & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{1}{2}T^2 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix}$$

Note that if $\omega = 0$ the above model describes a nearly constant velocity motion.

5.1.2 Measurement Model

The measurement model is also a nonlinear function that measures the range and the bearing of the motion. There are N$_s$ sensors each taking its own mearsurement according to equation

$$\mathbf{z}_k^i = h(\mathbf{x}_k) + \mathbf{v}_k^i, \qquad i = 1, 2, ..., N_s$$

where

$$h_i(\mathbf{x}_k) = \begin{bmatrix} r_i^i \\ b_k^i \end{bmatrix} = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \tan^{-1}\left(\frac{y}{x}\right) \end{bmatrix}$$

and

$$\mathbf{v}_k \sim N(0, \mathbf{R}_k^i), \quad \mathbf{R}_k^i = \begin{bmatrix} \sigma_{r_k}^2 & 0 \\ 0 & \sigma_{b_k}^2 \end{bmatrix}$$

The index i in $h_i(\mathbf{x}_k)$ signifies that each sensor takes measurement in its own coordinate system.

5.1.3 Scheme for Distributed Fusion

Estimation fustion has wide-spread applications since many practical problems involve data from multiple sources. One of the most important applications is target tracking using muliple sensors which is the scenario being simulated. Note that the measurements are assumed available without data association issue.

Let superscript i denotes quantities pertaining to sensor i. It is assumed that at every sensor a corresponding local tracker operates and provides to the central processor (fusion center) estimates $\widehat{x}^i_{k|k}$ of the target state (with the corresponding mean square error matrix $P^i_{k|k}$) for $i = 1, 2, ..., N_s$, respectively. The fusion center treats these local estimates as of target measuements according to the equation

$$z_k^* = \begin{bmatrix} \widehat{x}^1_{k|k} \\ \vdots \\ \widehat{x}^{N_s}_{k|k} \end{bmatrix} = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} x + \underbrace{\begin{bmatrix} \widehat{x}^1_{k|k} - x \\ \vdots \\ \widehat{x}^{N_s}_{k|k} - x \end{bmatrix}}_{v_k^*} \tag{5.2}$$

This model was introduced in [11] as a universal model for standard distributed fusion.

Then the target trackig problem at the fusion center is cast as an estimation problem for the state x_k subject to the model (5.1) based on the sequence of "measurements" $z^{*k} = \{z_1^*, z_2^*, ..., z_k^*\}$. This is a nonlinear estimation problem and its complete solution is given by the posterior PDF $p\left(x_k | z^{*k}\right)$.

5.2 Error Covariance of Pseudo-Measuement

In order to make the estimation problem well defined, the covariance of the "measurement error" v_k^* at the fusion center needs to be determined.

Denote

$$\mathbf{R}_k^{i*} = cov(v_k^*) = \begin{bmatrix} \Sigma_k^{11} & \cdots & \Sigma_k^{1n} \\ \vdots & \ddots & \vdots \\ \Sigma_k^{n1} & \cdots & \Sigma_k^{nn} \end{bmatrix}$$

Then clearly $\Sigma_k^{ii} = P_k^i$, $i = 1, 2, ..., N_s$ where P_k^i are provided by the local estimators along with the corresponding local state estimates $\widehat{x}^i_{k|k}$.

The crosscovariance Σ_k^{ij} for $i \neq j$ can be derived (approximately in our nonlinear case) following the methodology proposed in [20]:

Assume that the local filters are *best linear unbiased estimators* (BLUE). In this contest K_k^i denotes the ith estimator's gain, and F_k^i, H_k^i are the Jacobians evaluated by linearization of the system and measurement equations, respectively. Then subsequently

$$\begin{aligned} \widetilde{z}^i_{k|k-1} &\triangleq z_k^i - \widehat{z}^i_{k|k-1} \\ &= H_k^i x_k + v_k^i - H_k^i \widehat{x}^i_{k|k-1} = H_k^i \widetilde{x}^i_{k|k-1} + v_k^i \end{aligned}$$

$$\begin{aligned}
\Sigma^{ij}_{k|k} &= cov\left(x_k - \widehat{x}^i_{k|k}, x_k - \widehat{x}^j_{k|k}\right) \\
&= cov\left(x_k - \widehat{x}^i_{k|k}, x_k\right) - cov\left(x_k - \widehat{x}^i_{k|k}, \widehat{x}^j_{k|k}\right) \\
&= \Sigma^{ii}_{k|k} - cov\left(x_k - \widehat{x}^i_{k|k-1} - K^i_k \widetilde{z}^i_k, \widehat{x}^j_{k|k-1} + K^j_k \widetilde{z}^j_k\right) \\
&= \Sigma^{ii}_{k|k} - cov\left(x_k - \widehat{x}^i_{k|k-1}, \widehat{x}^j_{k|k-1}\right) - cov\left(x_k - \widehat{x}^i_{k|k-1}, K^j_k \widetilde{z}^j_k\right) + \\
&\qquad cov\left(K^i_k \widetilde{z}^i_k, \widehat{x}^j_{k|k-1}\right) + cov\left(K^i_k \widetilde{z}^i_k, K^j_k \widetilde{z}^j_k\right) \\
&= \Sigma^{ii}_{k|k} - cov\left(\widetilde{x}^i_{k|k-1}, \widetilde{x}^j_{k|k-1}\right) - cov\left(\widetilde{x}^i_{k|k-1}, \widetilde{z}^j_{k|k-1}\right)\left(K^j_k\right)' + \\
&\qquad K^i_k cov\left(\widetilde{z}^i_{k|k-1}, \widehat{x}^j_{k|k-1}\right) + K^i_k cov\left(\widetilde{z}^i_{k|k-1}, \widetilde{z}^j_{k|k-1}\right)\left(K^j_k\right)' \\
&= \Sigma^{ii}_{k|k} - \Sigma^{ij}_{k|k-1} - cov\left(\widetilde{x}^i_{k|k-1}, H^j_k \widetilde{x}^j_{k|k-1} + v^j_k\right)\left(K^j_k\right)' + \\
&\quad K^i_k cov\left(H^i_k \widetilde{x}^i_{k|k-1} + v^i_k, \widetilde{x}^j_{k|k-1}\right) + K^i_k cov\left(H^i_k \widetilde{x}^i_{k|k-1} + v^i_k, H^j_k \widetilde{x}^j_{k|k-1} + v^j_k\right)\left(K^j_k\right)'
\end{aligned}$$

Then, under the assumption that $\widetilde{x}^i_{k|k-1} \perp w^j_k, i \pm j$ and $v^i_k \perp v^j_k, i \neq j$ finally

$$\Sigma^{ij}_{k|k} = \Sigma^{ii}_{k|k} - \Sigma^{ij}_{k|k-1} - \Sigma^{ij}_{k|k-1}\left(H^j_k\right)'\left(K^j_k\right)' + K^i_k H^i_k \Sigma^{ij}_{k|k-1} + K^i_k H^i_k \Sigma^{ij}_{k|k-1}\left(H^j_k\right)'\left(K^j_k\right)' \quad (5.3)$$

Similarly

$$\begin{aligned}
\Sigma^{ij}_{k|k-1} &= cov\left(x_k - \widehat{x}^i_{k|k-1}, x_k - \widehat{x}^j_{k|k-1}\right) && (5.4) \\
&= cov\left(x_k - \widehat{x}^i_{k|k-1}, x_k\right) - cov\left(x_k - \widehat{x}^i_{k|k-1}, \widehat{x}^j_{k|k-1}\right) \\
&= \Sigma^{ii}_{k|k-1} - cov\left(F^i_{k-1} x_{k-1} + w_{k-1} - F^i_{k-1} \widehat{x}^{i)}_{k-1|k-1}, \widehat{x}^j_{k|k-1}\right) \\
&= \Sigma^{ii}_{k|k-1} - cov\left(F^i_{k-1} x_{k-1} + w_{k-1} - F^i_{k-1} \widehat{x}^i_{k-1|k-1}, F^j_{k-1} \widehat{x}^j_{k-1|k-1}\right) \\
&= \Sigma^{ii}_{k|k-1} - cov\left(F^i_{k-1} \widetilde{x}^i_{k-1|k-1} + w_{k-1}, F^j_{k-1} \widehat{x}^j_{k-1|k-1}\right) \\
&= \Sigma^{ii}_{k|k-1} - F^i_{k-1} \Sigma^{ij}_{k-1|k-1}\left(F^j_{k-1}\right) && (5.5)
\end{aligned}$$

Equations (5.3) and (5.4) provide a recursive algorithm for computation of R^*_k necessary for a state estimation at the fusion center.

An alternative method for evaluating R^*_k, tailored to the specific application of an UKF, is presented in the next subsection.

5.2.1 Crosscovariance

The equations (5.3) and (5.4) for the determination of the local estimates' crosscovariances require evaluation of the matrices F^j_{k-1}, H^j_k and filter gain K^j_k as well. These matrices could be evaluated via linearization of the equations. In this work only UKFs are used as local filters. As such, a natural and more accuate alternative is to send to the fusion center the local sigma points from which the crosscovariances are directly computed at the center. This is implemented in the algorithm. The drawback of this approach is that it could in general increases the communication from the sensors to the center. However, based on the fact that the covariance does not change considerably (as in our simulations), we can reduce the frequency of the communication on the sigma points without noticeable loss of accuracy.

5.3 Simulation and Results

We consider a tracking scenario in which the position of a maneuvering target is sampled every $T = 2$s. The target was making a turn in a plane at nearly constant turn rate of $3^\circ/\sec$, starting at $k = 1$ and ending at $k = 100$.

The initial condition of the target, with the units km for position and radian for bearing, was

$$x = \begin{bmatrix} 7 & 0 & 45 & -0.12 & 0 \end{bmatrix}'$$

$$Q = \begin{bmatrix} 0.6 \times 10^{-4} & 0 & 0 \\ 0 & 0.6 \times 10^{-4} & 0 \\ 0 & 0 & 0.13963 \times 10^{-4} \end{bmatrix}$$

Two local sensors were utilized, each with measurement noise covariance

$$R = \begin{bmatrix} 0.080^2 & \\ & 0.0087^2 \end{bmatrix}$$

The filter configurations implemented include: UKF for the two local (radar) trackers; PF, UKF, and UPF respectively for the fusion center with the local estimates treated as measurements in the distributed estimation scheme. For the purpose of comparison centralized (uses raw measurements directly) fusion at the center using UKF is also implemented. Figure 1 below shows a typical true state trajectory considered and measurements by one of the sensors.

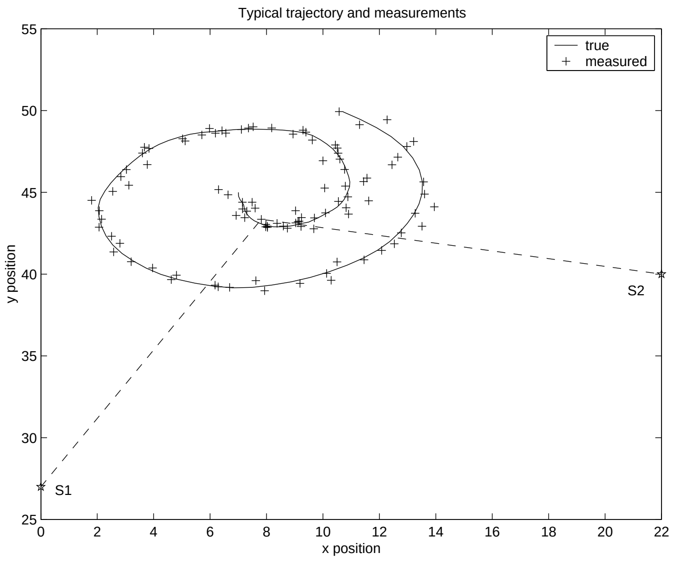

Figure 2: Typical trajectory considered

RMS estimation error is used to evaluate the algorithms' accuracy. In figure 2 (and subsequent figures) the measurement errors in Cartesian coordinates were obtained by the approximate polar-to-Cartesian conversion [16]

$$R = cov(w) \simeq \frac{\sigma_r^2 - r^2\sigma_\theta^2}{2} \left[\begin{array}{cc} b + \cos 2\theta & \sin 2\theta \\ \sin 2\theta & b - \cos 2\theta \end{array} \right]$$

with $b = \frac{\sigma_r^2 + r^2\sigma_\theta^2}{\sigma_r^2 - r^2\sigma_\theta^2}$.

The state estimation RMSEs by the fusion center running a PF are shown in Figure 3. The number of particles was 3000. It can be seen that for this problem of a 5-dimensional state vector the use of 3000 particles is not sufficient: the fusion results over 100 runs are not better than the local estimates from each single sensor. The computational burden is shown in Table 1.

Figure 4 shows the results over 100 runs by the center running a UKF. The fused estimates are substantially improved in accuracy as compared to the local estimates, both for position and speed. Similar results using a UPF center are given in Figure 5.

The UKF and the UPF (with 20 particles) centers are compared and the results are shown in Figure 6. It is seen that UPF gives smaller position RMSEs than UKF but their speed RMSEs are comparable.

Figure 7 shows performance of UPFs using different number of particles: 2, 10, 20, and 50. The RMSEs using 1 and 10 particles still have room for improvement, and 20 and beyond are very close to the RMSE of the centralized UKF filter.

The same UPF center was run which received local estimates every sampling period but sigma points at every 10 and 100 sampling periods, respectively. The RMSEs are shown in Figures 8. Clearly, for this scenario the crosscovariance of the local estimates varies slowly and its recomputation is not necessary at every time step of the central filter.

The CPU time (in sec) required by each run for the three filters is given in table below .

PF (3000 particles)	UKF	UPF(20 particles)
573.922	2.344	13.031

5.4 Conclusion

The particle filter is expensive in computation, and thus it needs to improve the efficiency. Though in this thesis we used Gaussian noise model, in principle PF can be used for any problem: linear/nonlinear and Gaussian/nonGaussian where UKF finds its limitations. The UPF is shown to be superior to UKF at the cost of more computation but it can provide flexible accuracy by using more or less particles. As expected the covariance of the measurement error of the fusion center changes slightly. The RMSE by skipping 9 sampling periods is almost the same as by skipping 99 sampling periods. So the state estimates are degraded little by reducing substantially the communication from local filters to the fusion center.

In general, the simulation results demonstrate that the proposed scheme for nonlinear distributed estimation by using sampling based filters is effective for multisensor tracking of maneuvering targets. Further research will elaborate this solution in a multiple model estimation framework.

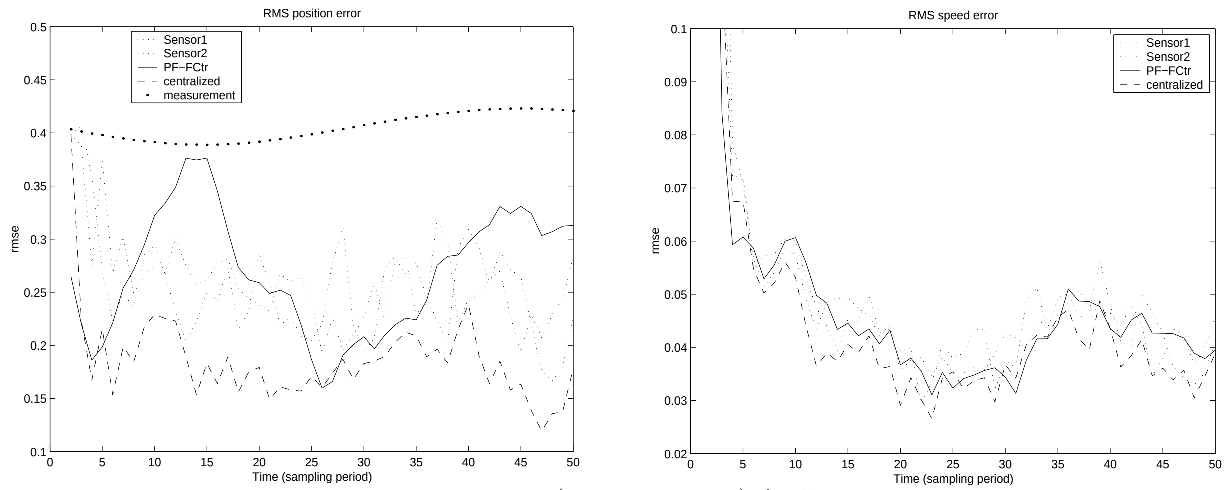

Figure 3: UKF-UKF-PF (3000 particles) & Centralized UKF

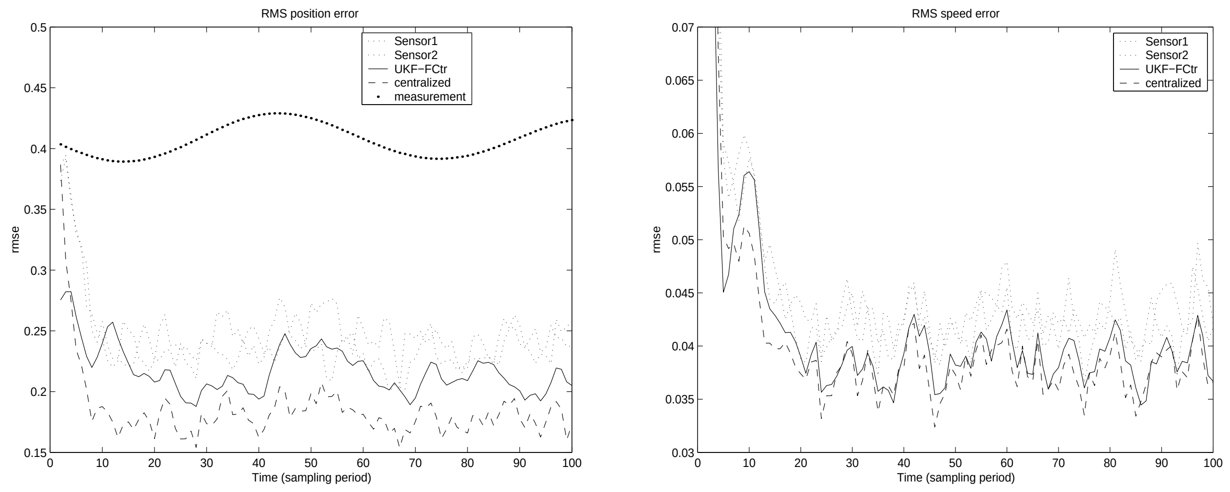

Figure 4: UKF-UKF-UKF & Centralized UKF

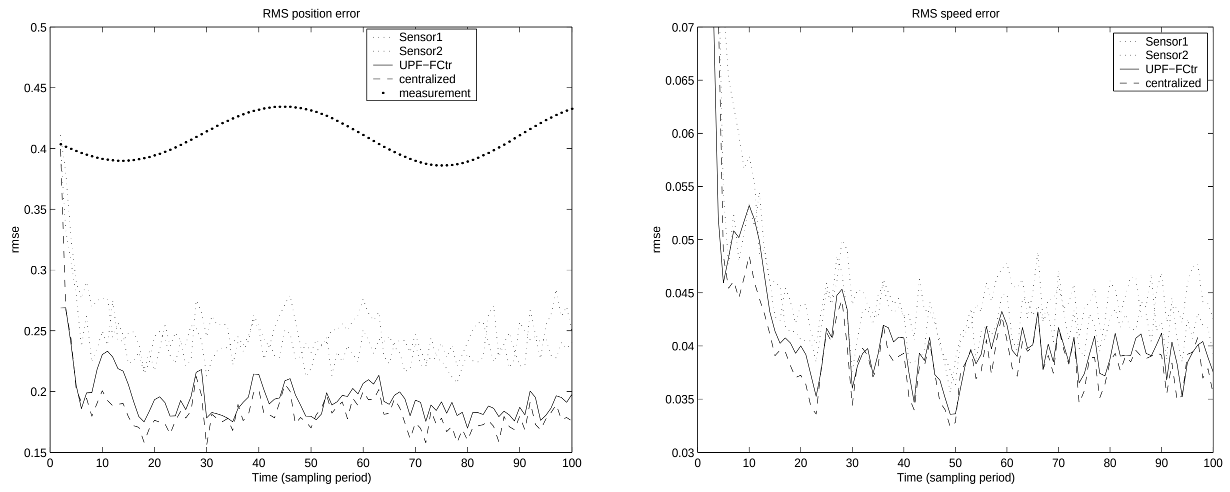

Figure 5: UKF-UKF-UPF (20 particles) & Centralized UKF

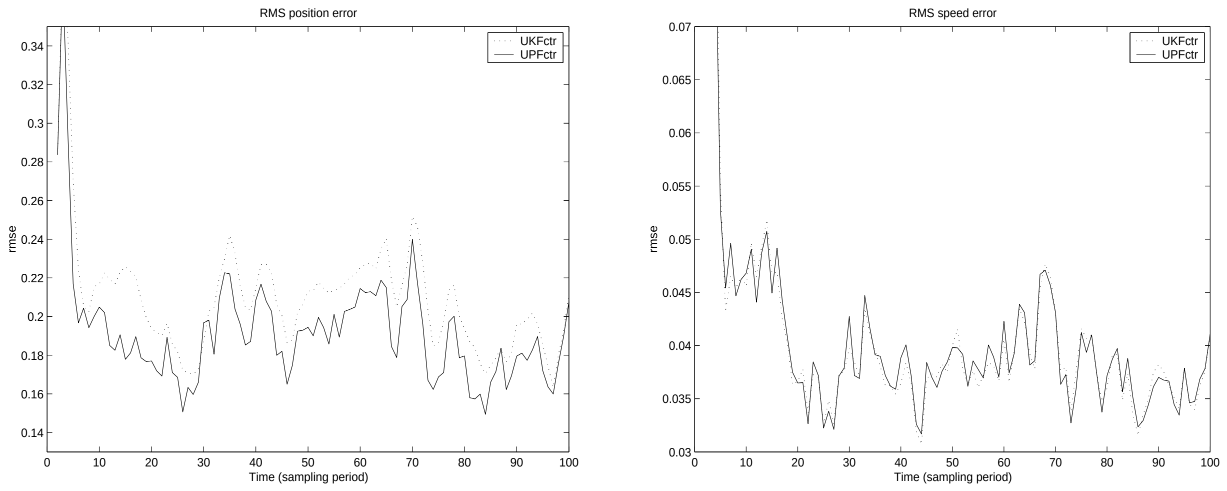

Figure 6: UKF vs. UPF (20 particles) (50 runs)

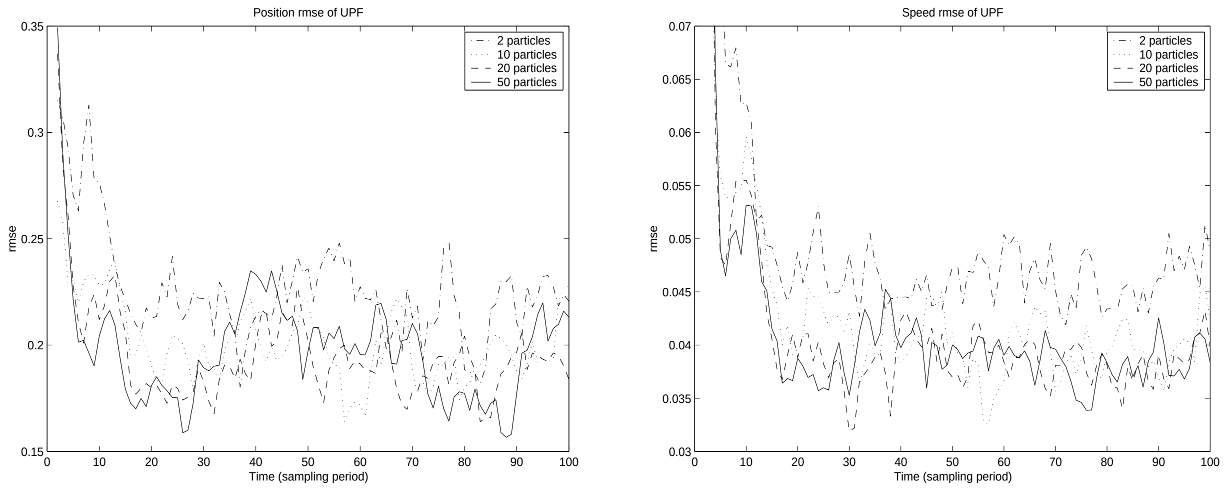

Figure 7: UPF center with different number of particles

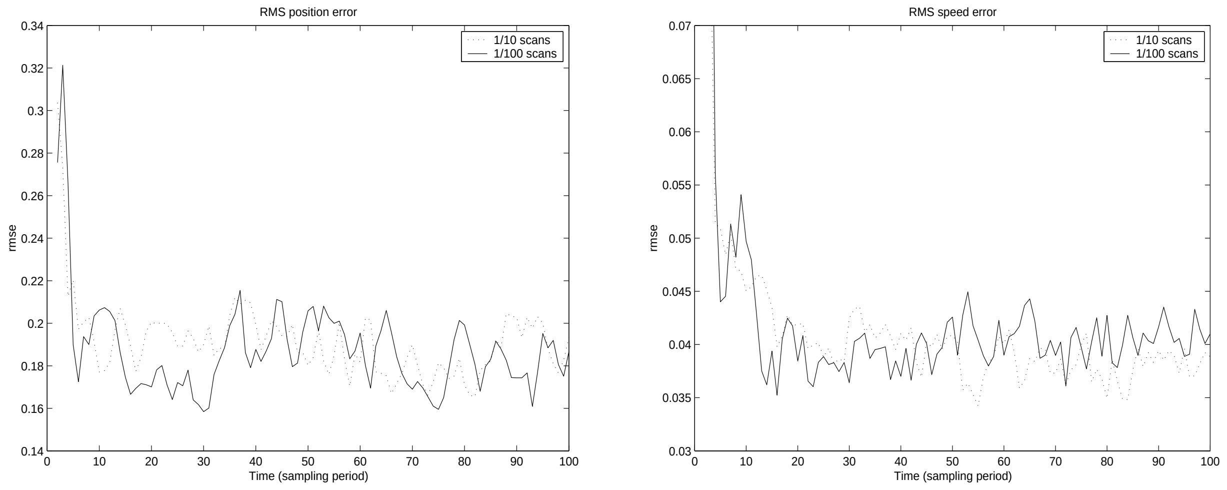

Figure 8: UPF center with different frequency of receiving sigma points

Bibliography

[1] X. R. Li and Y. Bar-Shalom. Design of an Interacting Multiple Model Algorithm for Air Traffic Control Tracking. *IEEE Trans. Control Systems Technology*, 1(3):186-194, Sept. 1993. Special issue on Air Traffic Control.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software.* Wiley, New York, 2001.

[3] N. Gordon, D. Salmond, and A. Smith. Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEE Proceeding* -F.,140(2):107-113, April 1993.

[4] A. Doucet, N. de Frietas, and N. Gordon, editors. it *Sequential Monte Carlo Methods in Practice.* Statistics for Engineering and Information Science. Springer-Verlag, New York, 2001.

[5] Special Issue on Monte Carlo Methods for Statistical Signal Processing. *IEEE Trans. Signal Processing,* 50(2), Feb. 2002.

[6] S. J. Julier and J. K. Uhlmann. A New Approach for Filtering Nonlinear Systems. In *Proceedings of the 1995 American Control Conference,* pages 1628-1632, Seattle, WA, 1995.

[7] S. Julier, J. Uhlmann, and H. F. Durrant-whyte. A New Method for Nonlinear Transformation of Means adn Covariances in Filters and Estimators. *IEEE Trans. Automatic Control,* AC-45(3):477-482, Mar. 2000.

[8] E. A. Wan and R. Van der Merwe. The Unscented Kalman Filter. In *Kalman Filtering and Newral Networks,* chapter 7. Wiley, 2001.

[9] R. V. Merwe, A. Doucet, N. De Freitas, and E. Wan. The Unscented Particle Filter. Technical Report CUED/F - INPENG/TR 380, Cambridge University Engineering Department, 2000. Also in: *Adv. Neural Inform. Process. Syst.,* Dec. 2000.

[10] X. R. Li and V. P. Jikov. A Survey of Maneuvering Target Tracking: Dynamic Models. In *Proc. 2000 SPIE Conf. Signal and Data Processing of Small Targets,* vol. 4048, pages 212-236, Orlando, Florida, USA, April 2000.

[11] X. R. Li, Y. M. Zhu, and C. Z. Han. Unified Optimal Linear Estimation Fusion–Part I: Unified Models and Fusion Rules. In *2000,* pages MoC2.10-MoC2.17, Paris, France, July 2000.

[12] T. M. Nguyen, V. P. Jilkov, and X. R. Li. "*Comparison of Sampling-Based Algorithms for Multisensor Distributed Target Tracking*". IF 2003 paper (number 194). Cairns, Australia. July 2003.

[13] A. Doucet, S. Godsil, and C. Andrieu. On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. *Statistics and Computing,* 10(3):197-208.

[14] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Trans. Signal Processing,* 50(2):174-188, Feb. 2002.

[15] W. R. Gilks and C. Berzuini. Following a Moving Target–Monte Carlo Inference for Dynamic Bayesian Models. *J. Royal Stat. Soc. B,* 63:127-146, 2001.

[16] Y. Bar-Shalom and X. R. Li. *Estimation and Tracking: Principles, Techniques, and Software.* Artech House, Boston, MA, 1993. (Reprinted by YBS Publishing, 1998).

[17] H. L. Van Trees. *Detection, Estimation and Modulation Theory.* Wiley, New York, 1968

[18] N. Bergman. "*Recursive Bayesian Estimation Navigation and Tracking Applications*" PhD thesis, Deparment of Electrical Engineering, Linkoping, Sweden, 1999.

[19] X. Rong Li. *Applied Estimation and Filtering.* Class notes, 2002.

[20] X. R. Li and P. Zhang. Optimal Linear Estimation Fusion–Part III: Cross-Correlation of Local Estimation Errors. In *2001*, pages WeB1.11-WeB1.18, Montreal, QC, Canada, Aug. 2001.

Vita

Born in Saigon,Vietnam, in 1965. He spent his first twenty eight years in Saigon, the biggest city of Vietnam. He enrolled in University of New Orleans in 1996 and earned his Bachelor of Science degree in Electrical Engineering in 2000. He then continued his education at the same school in the Master program in Electrical Engineering. His area of research has been on nonlinear filtering.

During his graduate study he also worked as teaching assistant and research assistant. He has been accepted to the PhD program in Electrical Engineering Department of University of New Orleans.

THESIS EXAMINATION REPORT

CANDIDATE: Trang Minh Nguyen

MAJOR FIELD: Electrical Engineering

TITLE OF THESIS: "Comparison of Sampling-Based Algorithms for Multisensor Distributed Target Tracking"

APPROVED:

[illegible]
Major Professor Chair – Dr. Xiao-Rong Li

Robert C. Cashner
Dean of the Graduate School

EXAMINING COMMITTEE:

[illegible]
Dr. Vesselin P. Jilkov

[illegible]
Dr. Humin Chen

____

____

DATE OF EXAMINATION: May 2, 2003