Realistic sensor modelling for multi-target tracking

by

Haohan Yao
(Warwick ID No.2119251)

Department of Statistics

University of Warwick

Coventry CV4 7AL

United Kingdom

Email: u2119251@warwick.ac.uk

14 September 2022

Report submitted in partial fulfillment of the requirements for the degree of MSc in Statistics in the University of Warwick

Abstract

Multiple object tracking is a necessary technology especially for some advanced technical systems like autonomous cars, which can be roughly characterised as the challenge of estimating the quantity of objects as well as their states respectively (such as position or velocity) from the data received by sensors. This problem can be solved in a Bayesian way by, e.g., modelling the targets as a Poisson (spatial) point process in which case the mean of the point process, which is a function of space, provides details on the expected quantity of targets and associated potential locations.
The main objective of the project is to adapt a recent multi-target tracking algorithm HISP Filter to the two following cases:

- Multi-sensor environments, e.g. when data from two cameras or from both a camera and a radar is available. In this case the challenges are to handle the boundary effects around the field of view of each sensor as well as the combination of the measurements from two sensors when a target is within both their field of view at the same time. In practical situations, multi-sensor fusion is extremely important for situational awareness since it allows for combining the strength of sensors, e.g. cameras give accurate information regarding the direction in which a target can be seen but nothing about the distance whereas a radar measures the distance accurately but might be less precise regarding the direction. In my simulation, we test the influence of the update order of different sensors, which shows that HISP filter is a very robust algorithm in terms of update order.

- Scanning sensors, e.g. radars. These sensors continuously scan their surroundings instead of taking snapshots like a camera. There is therefore a challenge in mod-

elling the process by which these sensors observe the targets. Scanning sensors can be found in airports and on most ships where they are used for traffic control, navigation and surveillance. In my simulation, I assume the radar with each scan cover an angle of π and make it scan for 100 periods

Acknowledgements

Time goes so fast that finally I come to the end of Master's learning. Appreciate for all the time of hardworking and harvesting.

My deepest and sincerest gratitude for the dissertation supervisor, Doctor Jeremie Houssineau, whose patient and considerate guidance makes the experience of exploring a total new region of research so interesting and meaningful. It is my privilege to study under his encouragement and instruction, which gives me confidence to face the unknown problem.

Very thankful to my girl friend Janet, for her taking care of me warmly in covid-positive period and for all the lovely time we have been sharing. With deep appreciation to my parents for supporting and encouraging both mentally and physically, their love is always silent but strong.

Contents

Chapter 1

Introduction

Multi-target tracking is to find the subset of observations paths that corresponds to the true observation history of each target. Depending on the context we can assume the objects to be humans such as football players or pedestrians moving around or it could be surrounding boats or submarines around the sea. Multiple object tracking is a necessary technology especially for some advanced technical systems like autonomous cars. A modern example of where it is important to keep track of moving objects is airport surveillance, the control tower is responsible for the planes that take off and land and also for the planes that are on the ground, in addition there are many other types of objects that move around at an airport such as vehicles that fill the planes with fuel or bring luggage to the planes. To avoid collisions at the airport, it is important to keep track of all these different objects. Frankly speaking, the objective of multi-target tracking is to learn how many targets there are at each time step and which observation was generated by which target.

Chapter 2

Literature Review

Traditional multi-target tracking algorithms mainly include multiple hypothesis tracking (MHT)[1], joint probabilistic data association (JPDA)[2], multi-target particle filter and their variances[3], etc.[4]. Singer proposed the nearest neighbor (NN) algorithm based on data association in 1971[5], which is simple and efficient, but not suitable for complex environments with dense sets of clutter(spurious measurements not originating from any target). In 1975, Bar-Shalom and Tse proposed the classical Probabilistic Data Association (PDA) algorithm based on the nearest neighbour approach[6]. Later, Bar-Shalom extended the PDA algorithm to multi-target tracking scenarios and proposed the Joint Probabilistic Data Association (JPDA)[7]. In 1979, Reid proposed another data association algorithm: multiple hypothesis tracking (MHT) algorithm[8], which enumerates all possible track hypotheses at the level of the track according to a certain length of time window. Later, Streit et al. modified the MHT algorithm by introducing the idea of expectation maximisation (EM) into the MHT algorithm and proposed the Probability Multiple Hypothesis Tracking (PMHT) algorithm[9]. JPDA and MHT are optimal algorithms at the point estimation and track estimation levels respectively, where JPDA considers all possible assignments of the measured point track to the target state association, and MHT considers the assumptions of all track branches association. The common denominator of these algorithms is that when the number of targets and clutter increases, the computational complexity of the algorithm increases rapidly and the problem of association explosion is faced, which is a difficulty faced by many data association

algorithms.

The finite set algorithm provides a descriptive model for sensor observation space and multi-target state space establishment, and directly uses the measurement set to filter the target state set for estimation, which effectively circumvents the measurement correlation problem and keeps the algorithm computational effort under control, which has received much attention and research. The theory of finite sets started in 1975 when Matheron proposed the earliest theory of random sets by studying random geometry problems[10]. Mahler proposed Finite Sets Statistics (FISST) based on this theory and combined it with Bayesian theory to derive a multi-objective Bayesian filter for finite sets: Probability Hypothesis Density (PHD) filter. However, Mahler only gives a theoretical representation of PHD, but not an implementation of PHD. In 2005, Vo used the Sequential Monte Carlo (SMC) method to give a PHD implementation under the nonlinear non-Gaussian assumption and proposed the SMC-PHD filter[11]. In 2006, under the linear Gaussian assumption, Vo derived a closed solution of the PHD filter using the Gaussian Mixture (GM) strategy, namely the GM-PHD filter[12]. In 2007, Mahler proposed the Cardinalized Probability Hypothesis Density (CPHD) filter[13], which takes into account the cardinality distribution of multiple targets and brings the algorithm's estimate of the number of targets closer to the true value than the PHD filtering algorithm. In 2007, Mahler described the Bernoulli Filter (BF) as a stochastic system with on and off states to represent the appearance or disappearance of targets in the tracking region, and proposed another multi-target approximation estimation method: the Multi-target Multi-Bernoulli Filter (MeMBer)[14]. Vo et al. modified its update step to propose the Cardinality Balanced MeMBer (CB-MeMBer) filter [15]. In order to solve the drawback that the multi-target tracking algorithm cannot output the target track directly in a finite set, Vo et al. introduced the concept of label as an attribute of the set element and proposed the Delta Generalized Labeled Multi-Bernoulli(δ-GLMB). PHD, CPHD, and CBMeMBer are representative single-sensor multi-target tracking algorithms in the context of finite set theory. The three algorithms have different characteristics in terms of computational performance requirements and real-time performance.

A new filter based on stochastic populations has been developed with the concept of

partially-distinguishable populations and is termed as Distinguishable and Independent Stochastic Populations (DISP) filter[16], which has a high computational cost. Following the DISP filter, a low-complexity filter known as the Hypothesized and Independent Stochastic Population (HISP)[17] filter has been developed, which has a linear complexity. The benefits of point-process-based techniques like PHD filter and conventional tracking methods like MHT are both included in the HISP filter.

Chapter 3

Background

3.1 Single-target Estimating.

Single object tracking is a filtering problem: the sequential processing of noisy measurements to determine the state of objects: position, properties describe its motion and other characters of interest. In general, we assume the motion model of objects' state to be a Markov chain on space $\mathcal{X} \subseteq R^{n_x}$ with the Markov transition density:

$$p(x_k|x_{k-1}) = \pi_{k|k-1}(x_k|x_{k-1}). \tag{3.1}$$

Observations are collected by sensors at time k on space $\mathcal{Z} \subseteq R^{n_z}$, measurement model can be expressed by a likelihood function

$$p(z_k|x_k) = g_k(z_k|x_k). \tag{3.2}$$

Assume that until time k, a series of observation $\mathcal{Z}_k : z_1, \ldots, z_k$ have been collected. So here we express the posterior density of states at k given all observations as:

$$p_k(x_k|z_{1:k}).$$

With certain initial density $p_0(x)$ the Bayes recursion is formulated by Chapman-Kolmogorov equation and Bayesian update equation

$$p_{k|k-1}\left(x_k \mid z_{1:k-1}\right) = \int \pi_{k|k-1}\left(x_k \mid x\right) p_{k-1}\left(x \mid z_{1:k-1}\right) dx, \tag{3.3}$$

$$p_k\left(x_k \mid z_{1:k}\right) = \frac{g_k\left(z_k \mid x_k\right) p_{k|k-1}\left(x_k \mid z_{1:k-1}\right)}{\int g_k\left(z_k \mid x\right) p_{k|k-1}\left(x \mid z_{1:k-1}\right) dx}. \tag{3.4}$$

Generally it is necessary to compute an object estimate using the posterior density. Two common estimators are EAP (expected a posteriori) and MAP (maximum a posteriori) and use MSE(mean squared error) to evaluate the performance of estimators. Given a performance measure, we can find an estimator giving minimal error (MMSE). Bayes filtering does not have a closed form except when models are linear with Gaussian noise and initial prior and the closed form is given by Kalman Filter, which is also the MMSE estimator.

3.2 Kalman Filter

Kalman filter, also known as linear quadratic estimation (LQE) is the basic process in information infusion. Based on linear model, Kalman filter compute the kalman gain K recursively to adjust the influence of new information (bias between measure and prediction). Generally the Kalman filter contains prediction and update process which includes five calculation steps, introducing as following:

Briefly consider a Gaussian linear dynamic model satisfying Markov property: The state x_k and observation z_k follows the model of form:

$$\text{motion} : \boldsymbol{x}_k = F_k \boldsymbol{x}_{k-1} + \boldsymbol{u}_k,$$
$$\text{observation} : \boldsymbol{z}_k = H_k \boldsymbol{x}_k + \boldsymbol{v}_k.$$

where F_k and H_k is called transition and observation matrix. Assume the noises are modelled as Gaussian, i.e.:

$$\begin{aligned} p_{\boldsymbol{u}_k}(u) &= \mathcal{N}(0, Q_k), \\ p_{\boldsymbol{v}_k}(v) &= \mathcal{N}(0, R_k). \end{aligned}$$

Besides, assume the $m_{k|k-1}$ to be posterior mean of targets state distribution at time k and $P_{k-1|k-1}$ to be the posterior covariance of. Then Kalman filter could be described as: prediction:

- target states mean prediction: $m_{k|k-1} = F m_{k-1|k-1}$

- target states covariance prediction: $P_{k|k-1} = FP_{k-1|k-1}F + Q_k$

update:

- Kalman gain K_k: $K_k = P_{k|k-1}H^T \left(HP_{k|k-1}H + R_k\right)^{-1}$
- update of posterior mean: $m_{k|k} = m_{k|k-1} + K_k \cdot \left(z_k - H\hat{m}_{k|k-1}\right)$
- update of posterior covariance: $P_{k|k} = \left(I - K_k H\right) P_{k|k-1}$

The general prove of Kalman filter involves the following two formula combining with Bayesian theory[14]:

$$\int \mathcal{N}(x; F\eta, Q)\mathcal{N}(\eta; m, P)d\eta = \mathcal{N}\left(x; Fm, Q + FPF^T\right)$$

$$\mathcal{N}(z; Hx, R)\mathcal{N}(x; m, P) = q(z)\mathcal{N}(x; \tilde{m}, \tilde{P})$$

where:

$$\begin{aligned}
\tilde{m} &= m + K(z - Hm) \\
\tilde{P} &= (I - KH)P \\
K &= PH^T \left(HPH^T + R\right)^{-1}
\end{aligned}$$

3.3 Random Finite Set Formulation of Multi-target Filtering.

Random finite set (RFS) could be considered as a random variable whose potential outcomes are sets with a finite quantities of unique elements, and both the number of elements and elements themselves are random. The number of elements in random finite set is defined as cardinality. With the elements belong to space D, the random finite set itself takes value in $\mathcal{F}(D)$, which refers to the set of all finite subsets of D. The property of RFS is usually described by a discrete probability distribution which characterizes the cardinality, and a family of joint probability densities which characterizes the distribution of the elements.

Note that there are many important properties of random finite set for modeling the states and measurements:

- sets are invariant to order,
- easy to add/remove elements,
- the set of state vectors is our quantity of interest.

Considering the multiple target situation, we assume that at time step k, $M(k)$ and $N(k)$ to be the quantities of targets and observations respectively, $x_{k,1}, ..., x_{k,M(k)}$ to be targets states, $z_{k,1}, \ldots, z_{k,N(k)}$ to be observations. As for targets, old objects may disappear and new objects may appear randomly, so only survived objects will be able to generate observations in the next time step. As for sensor, it would receive clutters in the field of view while receiving measurements only when targets are detected, all of what we have observed is source indistinguishable, i.e. we don't know whether it is clutter or measurements generated by targets. What's more, it is important to know that data association between targets and measurements is also unclear. The collections of target states and observations at time k are not ordered, hence it is logical to describe them as finite sets:

$$X_k = \left\{x_{k,1}, \ldots, x_{k,M(k)}\right\} \in \mathcal{F}(\mathcal{X}),$$
$$z_k = \left\{z_{k,1}, \ldots, z_{k,N(k)}\right\} \in \mathcal{F}(\mathcal{Z}).$$

Similar to the way of defining uncertainty in single target, We define the uncertainty by modelling the multi-target state X_k and multi-target measurement z_k as a RFS[12].Then multi-target tracking could be viewed as a filtering problem involved with space $\mathcal{F}(\mathcal{X})$ and $\mathcal{F}(\mathcal{Z})$. The RFS model involved with target motion, birth, and death, is described below for the time evolution of the multi-target state. Considering the time $k-1$, every $x_{k-1} \in X_{k-1}$ will continue to exist in the next time step k with probability $P_{S,k}(x_{k-1})$, known as survival probability, which means that objects would disappear with probability $1 - P_{S,k}(x_{k-1})$. With the condition of still existing at time k, the probability of transition between states is described as $\pi_{k|k-1}(x_k|x_{k-1})$.
All of the above could be defined as random finite sets:

$$S_{k|k-1}\left(x_{k-1}\right).$$

In general modelling there are two components of the target birth model conditional on state X_{k-1}:

1. spontaneous births (new targets that are independent of current targets, with states represented as RFS Γ_k)
2. spawning (new targets generated from existing targets, with states represented as RFS $B_{k|k-1}(x_{k-1})$. But for simplification there is no implementation of spawning in our model.

Then the multi-target state X_k at time k is determined by the union of the above three parts [12]:

$$X_k = \left[\bigcup_{\eta \in X_{k-1}} S_{k|k-1}(\eta)\right] \cup \left[\bigcup_{\eta \in X_{k-1}} B_{k|k-1}(\eta)\right] \cup \Gamma_k \tag{3.5}$$

As for the measurement model, it's necessary to take clutter and detection uncertainty into consideration. A given target $x_{k-1} \in X_{k-1}$ is detected with probability $P_{D,k}(x_k)$ and conditional on the detected fact, likelihood is given by $g_k(z_k|x_k)$. Hence at time k, each state $x_{k-1} \in X_{k-1}$ yeilds an RFS:

$$\theta_k(x_k) = \begin{cases} \emptyset & \text{if } x_k \text{ undetected} \\ z_k & \text{if } x_k \text{ detected} \end{cases}$$

Besides, the clutter or false alarm received by the sensor is defined as a RFS κ_k, so the multi-target observation z_k received at the sensor is formed by the union:

$$z_k = K_k \cup \left[\bigcup_{x \in X_k} \Theta_k(x)\right] \tag{3.6}$$

Similar to the single target situation, the transition density $\pi_{k|k-1}(\cdot|\cdot)$ and likelihood $g_k(\cdot|\cdot)$ respectively represent the randomness in the motion and observation in multi-target case. The form could be derived using Finite Set Statistics (FISST) and actual model of targets and sensors.

Analogue to Bayes filter in single target scene, the multi-target Bayes recursion based

on RFS including two step as well:

$$p_{k|k-1}\left(X_k \mid z_{1:k-1}\right) = \int \pi_{k|k-1}\left(X_k \mid X\right) p_{k-1}\left(X \mid z_{1:k-1}\right) \mu_s(dX) \tag{3.7}$$

$$p_k\left(X_k \mid z_{1:k}\right) = \frac{g_k\left(z_k \mid X_k\right) p_{k|k-1}\left(X_k \mid z_{1:k-1}\right)}{\int g_k\left(z_k \mid X\right) p_{k|k-1}\left(X \mid z_{1:k-1}\right) \mu_s(dX)} \tag{3.8}$$

In the equation, μ_s is defined as appropriate measure on $\mathcal{F}(\mathcal{X})$. The recursion above are intractable for computation as it involves multiple integrals on the space $\mathcal{F}(\mathcal{X})$. The optimal multi-target Bayes filter are now never practically implemented because of the complex multi-dimension integration and no computational solution. But direct approximations of it performs well when there are not many targets.

3.4 The PHD Filter

As multiple integrals in multi-object Bayesian filtering has a large computational complexity, there is a need for efficient estimation of multi-object Bayesian filters. Moment based approximation multi-object filters have been widely implemented, including PHD and CPHD filters. The PHD filter propagates the first order statistical moment defined as posterior intensity, instead of the posterior probability density. For a RFS X on space $\mathcal{X}$ with probability distribution P, the intensity v also known as first order moment, satisfies that for each region $S \subseteq \mathcal{X}$

$$\widehat{N} = \int |X \cap S| \; P(dX) = \int_S v(x)dx \tag{3.9}$$

which shows that the integral $\widehat{N}$ represent the expected number of the elements of X that are in S as well. Before characterising the PHD recursion it's necessary to make some assumptions and define the poisson RFS which is an important class of random finite sets: A random finite set X is *Poisson* if

- the distribution of cardinality $Prob(|X| = n)$ is Poisson with mean $\widehat{N}$.
- conditional on a given cardinality, the elements in X should be independently and identically distributed as $v(\cdot)/\widehat{N}$

based on following assumptions:

- The motion and observation between targets are independent.
- Clutter RFS follows Poisson and independent of measurement generated by target.
- The predicted multi-target RFS governed by $p_{k|k-1}$ is Poisson.

Refer to equation 3.5 and 3.6 it can be shown that the intensity of RFS could be calculated recursively through PHD recursion:

$$v_{k|k-1}(x) = \int p_{S,k}(\eta)\pi_{k|k-1}(x \mid \eta)v_{k-1}(\eta)d\eta + \gamma_k(x) \tag{3.10}$$

$$v_k(x) = [1 - p_{D,k}(x)]\, v_{k|k-1}(x) + \sum_{o \in z_k} \frac{p_{D,k}(x)g_k(z \mid x)v_{k|k-1}(x)}{\kappa_k(z) + \int p_{D,k}(\xi)g_k(z \mid \xi)v_{k|k-1}(\xi)d\xi} \tag{3.11}$$

where:

- γ_k intensity of birth RFS
- κ_k intensity of clutter RFS
- $P_{S,k}$ $P_{D,k}$ probability of survival and detection
- $\pi_{k|k-1}(\dots)$ and $g_k(\dots)$ are transition and measurement function.

Note that PHD recursion does not lead to an analytic solution, it is usually implemented using Gaussian Mixture (GM) or sequential Monte Carlo (SMC). Here the Gaussian Mixture PHD filter is one of which we adopt in the model.

Chapter 4

PHD filter on multiple sensor

4.1 The PHD recursion for linear Gaussian models

In the PHD filter, the Gaussian mixture distribution can be used to describe the posterior density in order to get closed form solution. Assume targets to be characterized as lineal Gaussian multi-target models:

- Each target's movement is governed by a linear Gaussian dynamic model, and the sensor's observation is also governed by a linear Gaussian.

$$\pi_{k|k-1}(x \mid \eta) = \mathcal{N}\left(x; F_{k-1}\eta, Q_{k-1}\right),$$
$$g_k(z \mid x) = \mathcal{N}\left(z; H_k x, R_k\right).$$

- the probabilities of survival and detection are constant and independent from state (things would be different when implement scanning sensors.)

$$P_{S,k}(x) = p_{S,k},$$
$$P_{D,k}(x) = p_{D,k}.$$

- the intensity of birth is Gaussian Mixture:

$$\gamma_k(x) = \sum_{n=1}^{N_{\gamma,k}} w_{\gamma,k}^{(n)} \mathcal{N}\left(x; m_{\gamma,k}^{(n)}, P_{\gamma,k}^{(n)}\right). \tag{4.1}$$

where $N_{\gamma,k}, w_{\gamma,k}^{(n)}, m_{\gamma,k}^{(n)}, P_{\gamma,k}^{(n)}, n = 1, \ldots, N_{\gamma,k}$, are given model parameters that determine the birth intensity. Adopting a linear Gaussian model and making clear assumptions, the posterior PHD at time $k-1$ is of form:

$$v_{k-1}(x) = \sum_{n=1}^{N_{k-1}} w_{k-1}^{(n)} \mathcal{N}(x; m_{\gamma,k}^{(n)}, P_{\gamma,k}^{(n)}). \tag{4.2}$$

4.1.1 GM-PHD prediction process

Refer to equation 3.10 at time k the predicted intensity would be a Gaussian Mixture as well:

$$v_{k|k-1}(x) = v_{S,k|k-1}(x) + \gamma_k(x) \tag{4.3}$$

where the form of $\gamma_k(x)$ is given above, and the form of $v_{S,k|k-1}(x)$ is

$$v_{S,k|k-1}(x) = p_{S,k} \sum_{n=1}^{N_{k-1}} w_{k-1}^{(n)} \mathcal{N}\left(x; m_{S,k|k-1}^{(n)}, P_{S,k|k-1}^{(n)}\right) \tag{4.4}$$

where according to the motion model it can be shown:

$$\begin{aligned} m_{S,k|k-1}^{(n)} &= F_{k-1} m_{k-1}^{(n)}, \\ P_{S,k|k-1}^{(n)} &= Q_{k-1} + F_{k-1} P_{k-1}^{(n)} F_{k-1}^T. \end{aligned}$$

Hence the prection intensity at time k is:

$$v_{k|k-1}(x) = \sum_{n=1}^{N_{k|k-1}} w_{k|k-1}^{(n)} N\left(x; m_{k|k-1}^{(n)}, P_{k|k-1}^{(n)}\right) \tag{4.5}$$

4.1.2 GM-PHD update process

Then following we show that posterior intensity would be a Gaussian Mixture as well. Refer to equation 3.11 we decide the form of $v_k(x)$ as:

$$v_k(x) = (1 - p_{D,k})\, v_{k|k-1}(x) + \sum_{z \in Z_k} v_{D,k}(x; z). \tag{4.6}$$

where

$$v_{D,k}(x; z) = \sum_{i=1}^{N_{k|k-1}} w_k^{(n)}(z) \mathcal{N}\left(x; m_{k|k}^{(n)}(z), P_{k|k}^{(n)}\right), \tag{4.7}$$

$$w_k^{(n)}(z) = \frac{p_{D,k} w_{k|k-1}^{(n)} q_k^{(n)}(z)}{\kappa_k(z) + p_{D,k} \sum_{\ell=1}^{N_{k|k-1}} w_{k|k-1}^{(\ell)} q_k^{(\ell)}(z)}. \tag{4.8}$$

in which:

$$q_k^{(n)}(z) = \mathcal{N}\left(z; H_k m_{k|k-1}^{(n)}, R_k + H_k P_{k|k-1}^{(n)} H_k^T\right) \tag{4.9}$$

$$\begin{cases} m_{k|k}^{(n)}(z) = m_{k|k-1}^{(n)} + K_k^{(n)}\left(z - H_k m_{k|k-1}^{(n)}\right), \\ P_{k|k}^{(n)} = \left[I - K_k^{(n)} H_k\right] P_{k|k-1}^{(n)}, \\ K_k^{(n)} = P_{k|k-1}^{(n)} H_k^\top \left(H_k P_{k|k-1}^{(n)} H_k^\top + R_k\right)^{-1}. \end{cases} \tag{4.10}$$

Here the equations 4.10 is the form of Kalman filter and $\kappa_k(z)$ in 4.8 represent the intensity of clutters.

From above it's obvious that the posterior propagation process is a recursion from v_{k-1} to $v_{k|k-1}$ and then to v_k, where the propagation in GM-PHD is actually the mean, covariance and weights of Guassians, which include whole information of posterior density. $(1 - P_{D,k})v_{k|k-1}$ refer to the mis-detection term and $v_{D,k}(\cdot|z)$ refer to detection term for $z \in \mathcal{Z}$. And the update of the mean and covariance of $v_{D,k}(\cdot|z)$ is Kalman updates, whereas recursions for the means and covariances of $v_{S,k|k-1}$ turn out to be Kalman predictions. The whole processes are built on the Gaussian mixture terms.

The property of Gaussian Mixture turns out that the expected number of targets could be calculated from the sum of the appropriate weights[12]. The expected number of the predicted targets is:

$$\hat{N}_{k|k-1} = \hat{N}_{k-1} p_{S,k} + \sum_{n=1}^{N_{\gamma,k}} w_{\gamma,k}^{(n)}. \tag{4.11}$$

The expected number of the updated targets is:

$$\hat{N}_k = \hat{N}_{k|k-1}\left(1 - p_{D,k}\right) + \sum_{z \in Z_k} \sum_{n=1}^{N_{k|k-1}} w_k^{(n)}(z). \tag{4.12}$$

4.2 Management of Mixture Components

As the increasing number of Gaussian terms during PHD recursions, the computation volume increase without bound. It's necessary to cut out some Gaussian terms.

- **Merge** After calculating the weight of each term, we can reduce the components with small weights which could be controlled by the threshold set at first. Or we can keep a certain number of components the largest weights. Sometimes when the Gaussian terms are close so that they could be approximately merged into one Gaussian.

- **Prune** When we got the posterior intensity v_k, the means of Gaussian components could be extracted as the estimates of multi-target state under the condition that distribution between Gaussian term is distinguishable because the mean is actually the local maxima of v_k. A general better way of selecting the means of Gaussians is to adopt means with weights greater than set threshold e.g. 0.5.

Besides, **Gating** could be used to reduce the number of measurements that the filter has to process by only retaining the measurements that are close to the predicted measurements.[18]

4.2.1 Simulation of GMPHD

Here we briefly introduce the simulation of target tracking based on Gaussian mixture PHD. The following simulations in this article would be similar.

parameters

Assume our field of view is a 1000×1000 square and the true tracks of 12 targets is recorded as in Figure 4.1, in which every target starts from red circle and end at blue triangle. The state $x_k = [p_{x,k}, p_{y,k}, \dot{p}_{x,k}, \dot{p}_{y,k}]^T$ consists of position p and velocity $\dot{p}$ in x and y directions at time k. In the ground truth we specially set targets appear from 4 possible locations, so here set Poisson random finite sets Γ_k with intensity:

$$\gamma_k(x) = 0.03\mathcal{N}\left(x; m_\gamma^{(1)}, P_\gamma\right) + 0.03\mathcal{N}\left(x; m_\gamma^{(2)}, P_\gamma\right) + 0.03\mathcal{N}\left(x; m_\gamma^{(3)}, P_\gamma\right) + 0.03\mathcal{N}\left(x; m_\gamma^{(4)}, P_\gamma\right) \tag{4.13}$$

where:

$$
\begin{aligned}
m_{\gamma}^{(1)} &= [0, 0, 0, 0]^T \\
m_{\gamma}^{(2)} &= [-200, 800, 0, 0]^T \\
m_{\gamma}^{(3)} &= [400, -600, 0, 0]^T \\
m_{\gamma}^{(4)} &= [-800, -200, 0, 0]^T \\
P_{\gamma} &= \text{diag}\left([10, 10, 10, 10]^T\right)
\end{aligned}
$$

Table 4.1: Parameter of Simulation

P_S	P_D	κ_c	σ_v	σ_x	σ_y
0.99	0.95	5	5	10	10

[1] κ_c : intensity of clutters

[2] σ_v : std. deviation of process noise

[3] σ_x, σ_y: std. deviation of observation noise

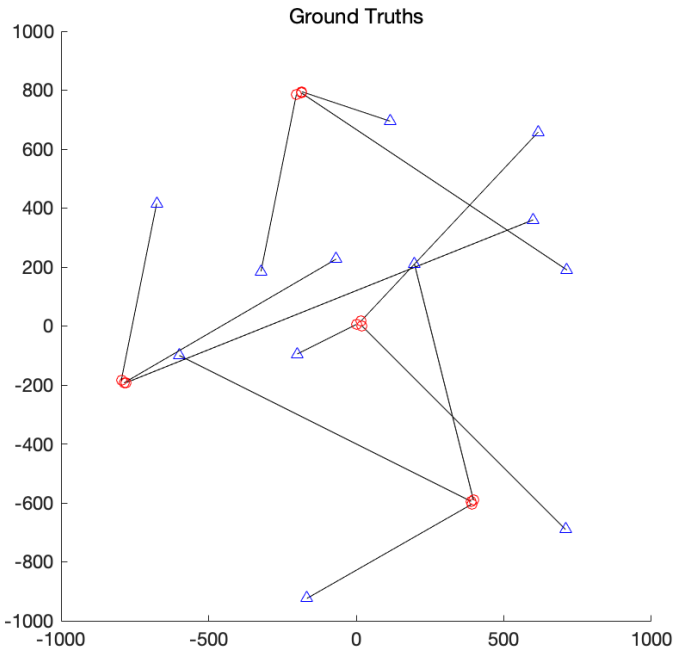

Figure 4.1: True track of targets

OSPA

Here adopting OSPA(Optimal Sub-Pattern Assignment) distance as the performance evaluation indicators. Each time we compute the distance that tells us how far it is from the truth which is much important when comparing algorithms. Following is some basic theoretical introduce about OSPA distance.

The OSPA metric is a distance between two sets of points that takes both the differences in the numbers and values of the points in two sets into consideration, which is often used in multi-target filtering.[19] The OSPA distance between set $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_n\}$ is defined as

$$\bar{d}_{c,p}^{(c)}(\mathbf{X}, \mathbf{Y}) = \left(\frac{1}{n} \left(\min_{\boldsymbol{a} \in \boldsymbol{\Pi}_n} \sum_{i=1}^{m} d^{(c)} \left(\mathbf{x}_i, \mathbf{y}_{\alpha(i)} \right)^p + c^p (n - m) \right) \right)^{\frac{1}{p}}. \tag{4.14}$$

The function $d_p^{(c)}$ is the OSPA metric of order p with cut-off c, which determines the importance of the accuracy of the number estimate compared to the accuracy of the target state estimate.$\boldsymbol{\Pi}_n$ denotes an arbitrary arrangement of the set $\{1, 2...n\}$, $d^{(c)} \left(\mathbf{x}_i, \mathbf{y}_{\alpha(i)} \right)$ is the measure of cutoff:

$$d^{(c)} \left(\mathbf{x}_i, \mathbf{y}_{\alpha(i)} \right) = \min \left(c, d \left(\mathbf{x}_i, \mathbf{y}_{\alpha(i)} \right) \right). \tag{4.15}$$

When using OSPA to evaluate multi-target tracking performance, we can adjust the p and c to vary the importance of multi-target state error and multi-target number error in OSPA. The figure 4.2 show the evaluation of a GM-PHD multi-target tracking using OSPA which contains three part: OSPA distance, state estimate error and cardinality estimate error.

simulation

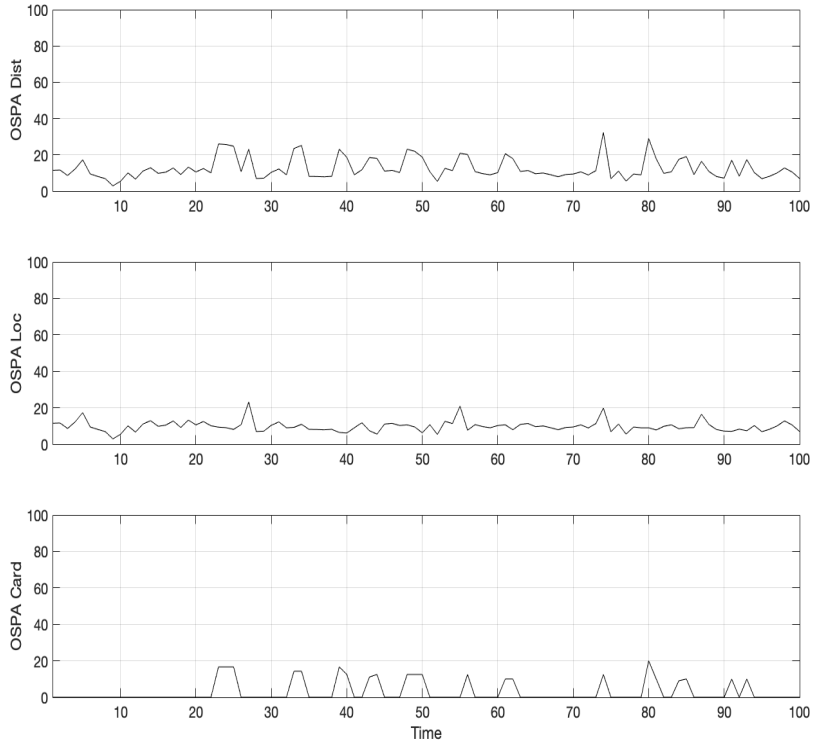

Figure 4.2: GM-PHD OSPA

Figure 4.3 and Figure 4.4 show the estimate and truth of position states and cardinality of targets cahnges through time in which we can clearly see where the algorithm perform well or bad. This is pretty helpful when improving our algorithm or adjusting the parameters of the simulation process.

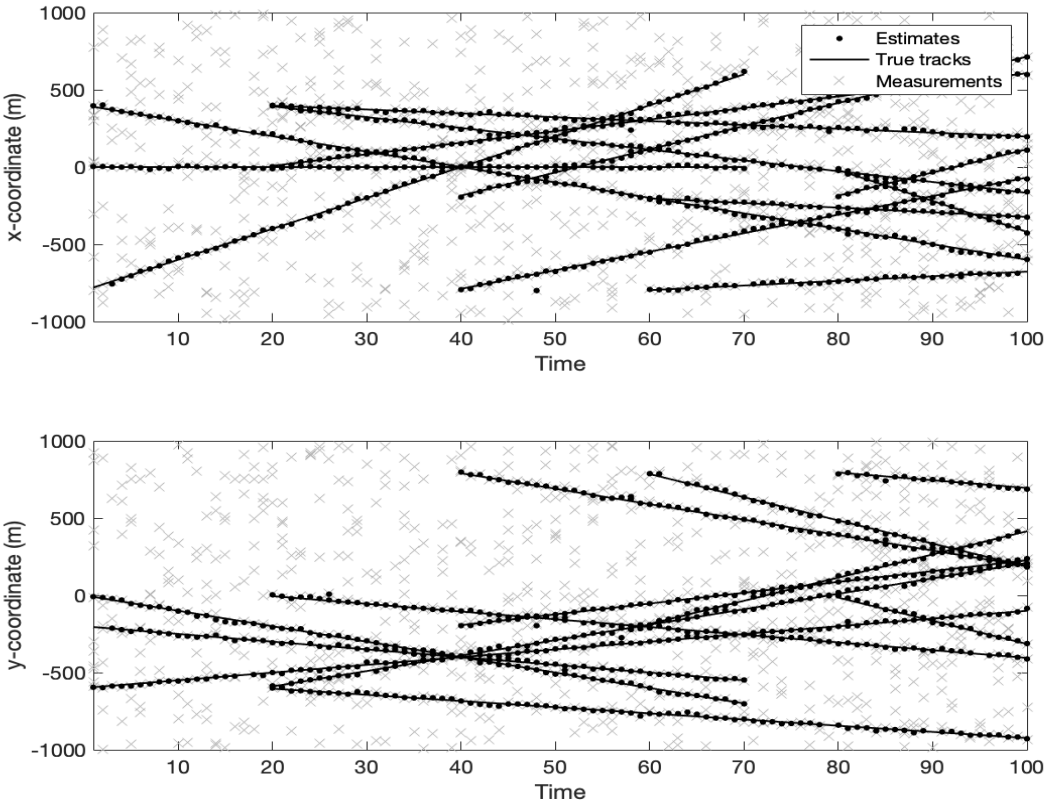

Figure 4.3: GM-PHD Estimate

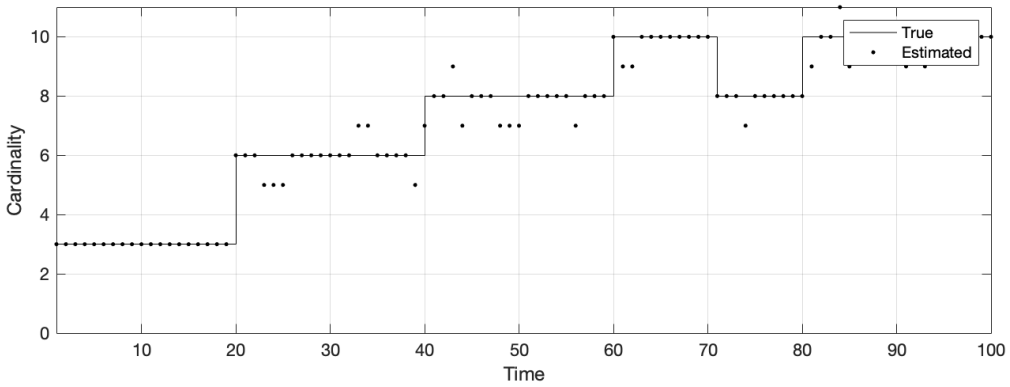

Figure 4.4: GM-PHD Cardinality

4.3 Extension to Nonlinear scenario: Taylor Series Approximation

In the scenario when the motion or observation models are nonlinear there are generally two ways of analytic approximations of the nonlinear Bayes filter including the extended Kalman (EK) filter[20],[21] and the unscented Kalman (UK) filter[22],[23]. The extended kalman filter propagates Gaussian by applying local linearization to kalman recursion, Whereas the unscented kalman filter computes the Gaussian approximation at the next time step using the unscented transform. Here in polar system only focus on the use of EKF(Extend Kalman Filter) derived by Taylor Series Approximation.
Let the motion model be $x_k = f_k\left(x_{k-1}\right)+\nu_{k-1}$ and measurement model be $z_k = h_k\left(x_k\right)+\varepsilon_k$ where f_k and h_k are non-linear models, ν_{k-1} and ε_k are random noises with covariance Q_{k-1} and R_k. First we could begin from the estimate of target states at time k without lose of generality. Apply Taylor expansion on f_k around $x_{k|k}$ to first order, characterized as:

$$f_k(x) - f_k\left(x_{k|k}\right) \cong F_k\left(x - x_{k|k}\right)$$

then transition matrix F_k would be approximated as

$$F_k x \triangleq \frac{\partial f_k}{\partial x}\left(x_{k|k}\right),$$

hence we could describe prediction as before:

$$x_{k+1|k} = f_k\left(x_{k|k}\right)$$
$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_k$$

based on predicted estimate $x_{k+1|k}$. Apply Taylor expansion on h_{k+1} around $x_{k+1|k}$ to first order, characterized as:

$$h_{k+1}(x) - h_{k+1}\left(x_{k+1|k}\right) \cong H_{k+1}\left(x - x_{k+1|k}\right),$$

then observation matrix would be approximated as:

$$H_{k+1} x \triangleq \frac{\partial h_{k+1}}{\partial x}\left(x_{k+1|k}\right).$$

hence adjusted observation would be:

$$\tilde{z}_{k+1} \triangleq z_{k+1} + Hx_{k+1|k} - h_{k+1}\left(x_{k+1|k}\right).$$

and update equations would be describe as before:

$$\begin{aligned}
x_{k+1|k+1} &= x_{k+1|k} + K_{k+1}\left(\tilde{z}_{k+1} - H_{k+1}x_{k+1|k}\right), \\
&= x_{k+1|k} + K_{k+1}\left(z_{k+1} - h_{k+1}\left(x_{k+1|k}\right)\right), \\
P^{-1}_{k+1|k+1} &= \left(I - K_{k+1}H_{k+1}\right)P_{k+1|k}, \\
K_{k+1} &= P_{k+1|k}H^T_{k+1}\left(H_{k+1}P_{k+1|k}H^T_{k+1} + R_{k+1}\right)^{-1}.
\end{aligned}$$

4.3.1 Simulation

In our simulation we assume the scene is in polar system so what we've observed is information about angle and distant, So the measurment model would be non-linear whereas the motion model is still linear. The state $x_k = [p_{x,k}, p_{y,k}, \dot{p}_{x,k}, \dot{p}_{y,k}]^T$ consists of position p and velocity $\dot{p}$ in x and y directions at time k, while the transition and covariance matrixes are:

$$F_k = \begin{bmatrix} I_2 & \Delta I_2 \\ 0_2 & I_2 \end{bmatrix}; \quad Q_k = \sigma^2_\nu \begin{bmatrix} \frac{\Delta^4}{4}I_2 & \frac{\Delta^3}{2}I_2 \\ \frac{\Delta^3}{2}I_2 & \Delta^2 I_2 \end{bmatrix}$$

assume the sampling period $\Delta = 1s$, and standard deviation of transition noise as $\sigma_\nu = 1\,(\ \mathrm{m/s^2})$, the survival probability of every targets is $P_{S,k} = 0.99$. Each target is detected with probability $P_{D,k} = 0.9$. the form of measurements and observation function would be:

$$z_k = \begin{bmatrix} \rho_k \\ \theta_k \end{bmatrix}; \quad h(x_k) = \begin{bmatrix} \sqrt{p^2_{x,k} + p^2_{y,k}} \\ \arctan\left(\frac{p_{y,k}}{p_{x,k}}\right) \end{bmatrix} \tag{4.16}$$

The H_k is Jacobian Matrix caculated as:

$$H_k = \begin{bmatrix} \frac{p_{x,k}}{\sqrt{p^2_{x,k}+p^2_{y,k}}} & \frac{p_{y,k}}{\sqrt{p^2_{x,k}+p^2_{y,k}}} & 0 & 0 \\ -\frac{p_{y,k}}{p^2_{x,k}+p^2_{y,k}} & \frac{p_{x,k}}{p^2_{x,k}+p^2_{y,k}} & 0 & 0 \end{bmatrix} \tag{4.17}$$

Here we assume the same motion scenario of targets as GM-PHD, But our observation field has changed into $[-\pi, \pi] * [0, 1000]$. Note that its important here to define the clutter field within the field of observation. And as the observation measure has changed, the covariance of the noise of observation need to be changed as well.

Table 4.2: Parameter of Simulation

P_S	P_D	κ_c	σ_v	σ_ρ	σ_θ
0.99	0.9	10	1	10	$\pi/180$

[1] σ_v : std. deviation of transation noise

[2] $\sigma_\rho, \sigma_\theta$: std. deviation of observation noise

The results of simulation using EKF is shown in Figure 4.5 4.6 4.7

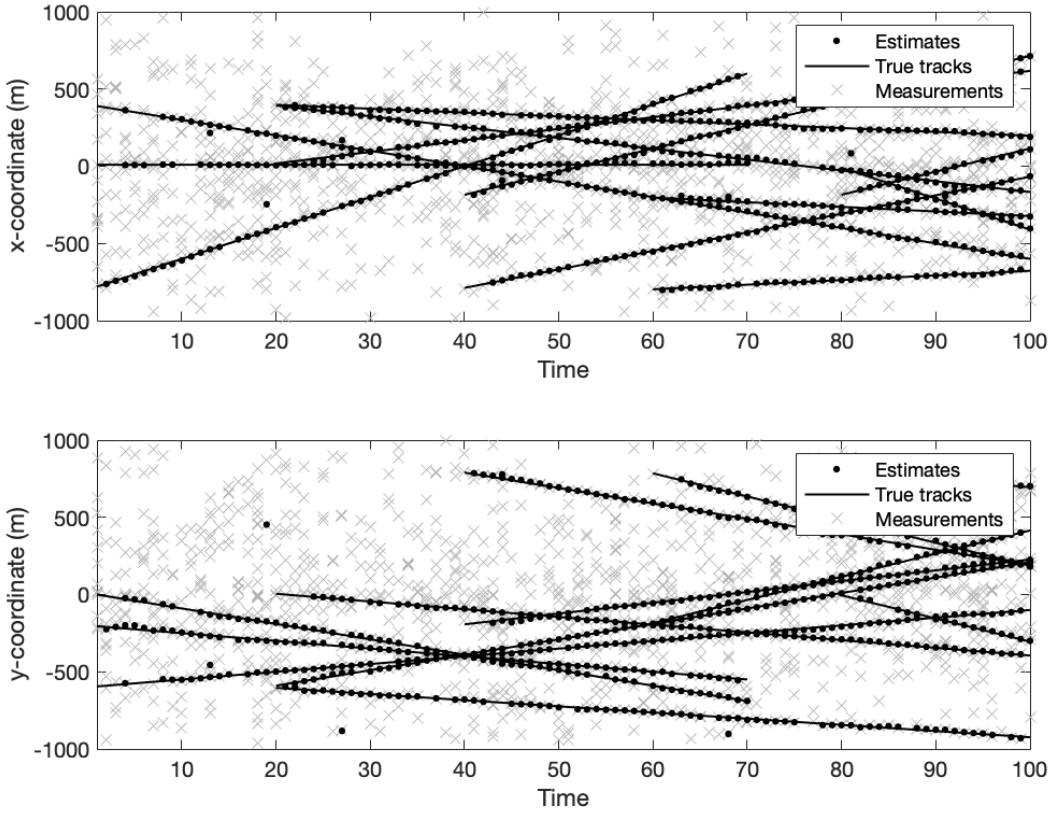

Figure 4.5: Estimate of States in Polar system

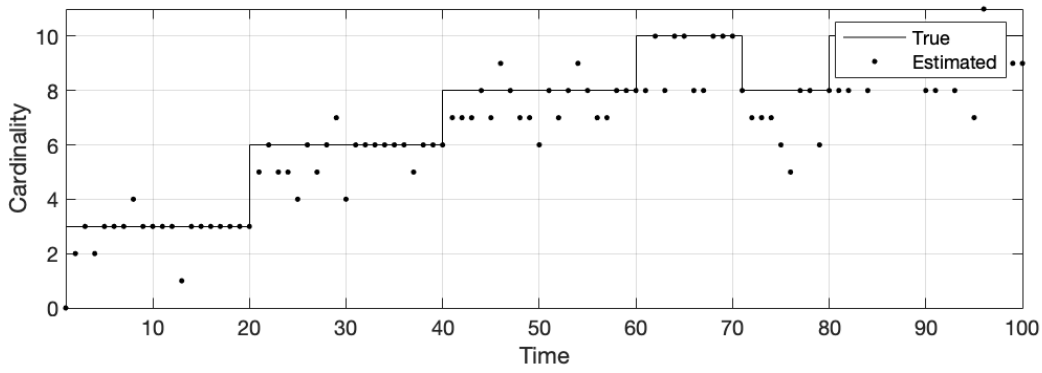

Figure 4.6: Estimate of cardinality in Polar system

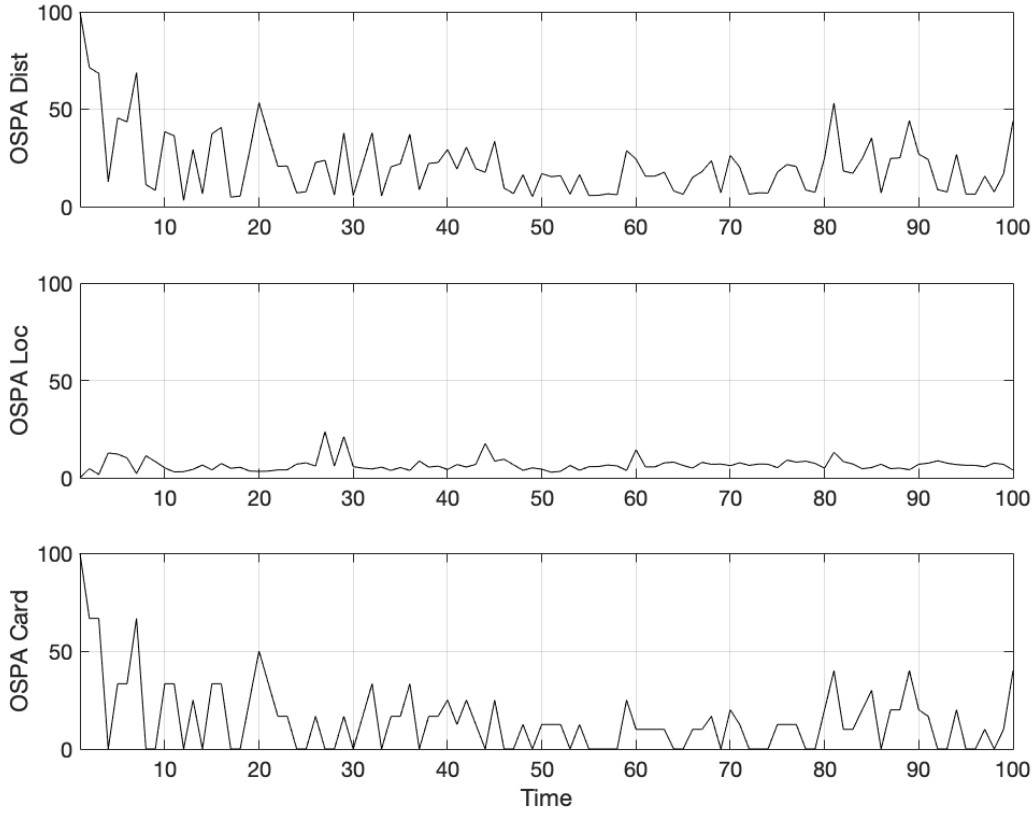

Figure 4.7: OSPA distance of EKF

It's obvious there should be noise in the OSPA figure which may influence our analysis, an intuitive method is to do algorithm for many times and take the average of

OSPA distance, which is shown in figure 4.8.

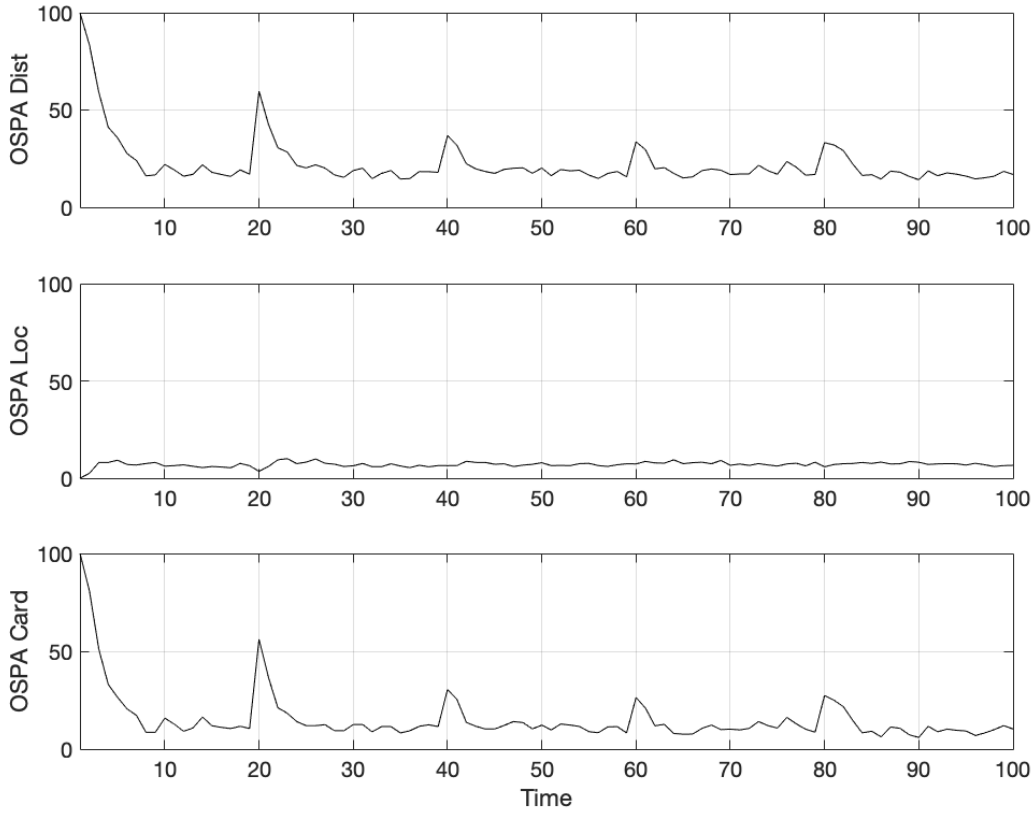

Figure 4.8: average OSPA distance of EKF

It's obvious that there are several peaks in the trends of OSPA distance, which corresponding to the states and cardinality change of targets, eg. birth or death of targets. There are births of Targets at time $20, 40, 60, 80$ where the peaks show the measure of difference in information between estimation and truth of Targets. This is because that the multi-targets tracking algorithm could not previously know what would happen, the filtering process has time lags as we update only after we have observed. And comparing to linear case 4.2, we may find EKF do not perform that good.

4.3.2 Extend to Multi-sensor scenario

Multi-sensor environments, e.g. when data from two cameras or from both a camera and a radar is available. In this case the challenges are to handle the boundary effects around the field of view of each sensor as well as the combination of the measurements from, say, two sensors when a target is within both their field of view at the same time. In practical situations, multi-sensor fusion is extremely important for situational awareness since it allows for combining the strength of sensors.

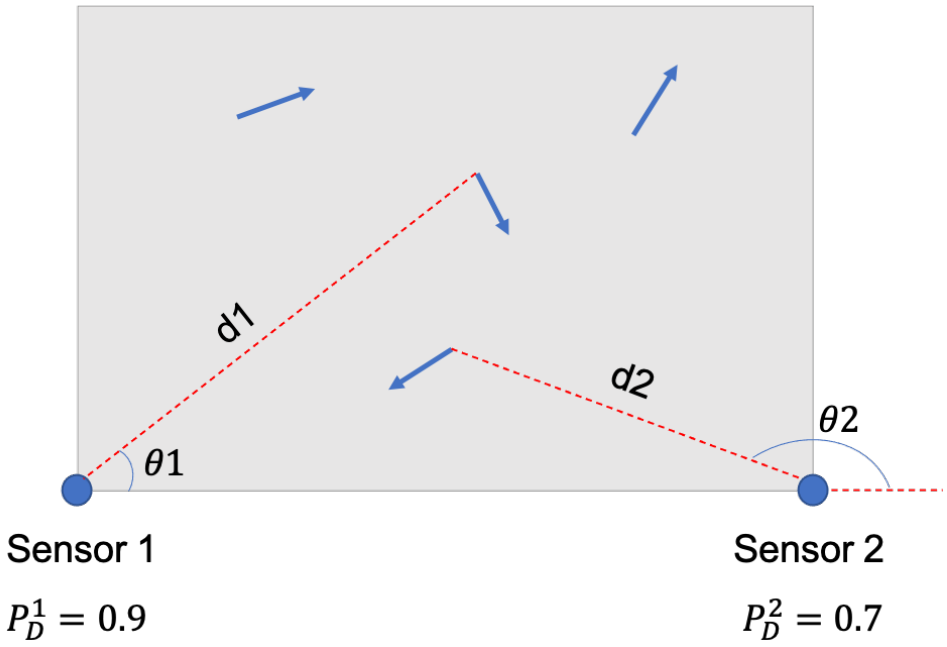

Figure 4.9: multi-sensor scenario

Comparing to single sensor situation, iterating the updated PHD of each sensor is an approximate solution to multi-sensor case, which is called Iterated-corrector PHD filter[4]. Iterate filter may to some extent reduce the miss detection and false alarm.[24] The Iterated-PHD filter is appealing because of its simple implementation and low computational complexity even if it lacks a sound mathematical foundation.

Table 4.3: Parameter of Simulation

sensors	P_S	P_D	κ_c	σ_v	σ_ρ	σ_θ
sensor1	0.99	0.9	10	1	10	$\pi/180$
sensor2	0.99	0.7	10	1	10	$\pi/180$

Note that we would update these two sensor every time step in different order to compare

Figure 4.10: Average OSPA Compare of different order

The figure 4.10 shows that even updated with same sensors, the order of update could have remarkable influence on the performance of target tracking. Between sensors there is relatively good and bad based their detection ability, observation noise, clutter intensity. Intuitively thinking, if the observations of bad sensor were updated at first, it will still be corrected by later observations from good sensors.[24] On the contrast,

if the bad sensor is last updated, the high noise would not be corrected by any precise observation, which cause a higher information lost showing in OSPA distance.

Chapter 5

HISP filter

At each time step $t \in \mathbb{N}$, we receive a set Z_t of observations in the space $\mathbb{R}^{d'}$ and we introduce ϕ as an isolated point corresponding to a false negative. At a given time t, we consider the set $\mathcal{O}_t$ of observation paths, i.e. of sequences of past observations, defined as

$$\mathcal{O}_t = (Z_1 \cup \{\phi\}) \times \cdots \times (Z_t \cup \{\phi\}) \setminus \boldsymbol{\phi}_t,$$

with $\boldsymbol{\phi}_t = (\phi, \ldots, \phi)$, the vector of size t with each component equal to ϕ. Remind the standard assumptions are:

1. The population of target is subject to a birth-death process
2. A given target can generate no more than one observation at each time step
3. An observation can originate from no more than one target
4. A random number of observations originate from background noise

Although some of these assumptions simplify the problem, multi-target tracking remains challenging in terms of computational complexity and usually requires approximations even in the simplest cases because of the absence of information regarding data association, i.e. the origin of observations at each time step. Target hypotheses are characterised by their index $\boldsymbol{i} = (T, \boldsymbol{o})$ where $\boldsymbol{o} \in \mathcal{O}_t$ is their observation path and T is the current time t of their last time of existence if the target has not survived to the current time. If $T = t$ then the distribution of the state $X_t^{\boldsymbol{i}}$ of the target hypothesis $\boldsymbol{i}$ on the state space $\mathbb{R}^d$ can be deduced from the observation path $\boldsymbol{o}$ by a suitable inference algorithm and

is denoted $p_t^{\boldsymbol{i}}$. Instead, if it holds that $t > T$ then $p_t^{\boldsymbol{i}} = \delta_\psi$ where ψ is a cemetery state isolated from $\mathbb{R}^d$. The probability for an hypothesis $\boldsymbol{i}$ to be valid is denoted by $w_t^{\boldsymbol{i}}$ and is referred to as its *weight*. The set of possible target hypotheses at time t is defined as $\mathcal{I}_t = \{1, \dots, t\} \times \mathcal{O}_{t-1}$ after the prediction and as $\hat{\mathcal{I}}_t = \{1, \dots, t\} \times \mathcal{O}_t$ after the update.

5.1 Prediction

We denote by $q_t(\cdot|x)$ the Markov transition from state x from time $t-1$ to time t and by $p_t^{\mathrm{s}}(x)$ the probability of survival at x from time $t-1$ to time t. Setting the current time to $t-1$, we consider an hypothesis $\boldsymbol{i} = (T, \boldsymbol{o})$; if $T = t-1$ then $\boldsymbol{i}$ gives rise to two distinct hypotheses at time t: one with index $\boldsymbol{j} = (t, \boldsymbol{o})$ corresponding to the survival of the target and one with index $\boldsymbol{k} = (t-1, \boldsymbol{o})$ corresponding to the case where the target has not survived. The associated weights and probability distributions are

$$w_t^{\boldsymbol{j}} = w_{t-1}^{\boldsymbol{i}} \int p_t^{\mathrm{s}}(x) p_{t-1}^{\boldsymbol{i}}(x) \mathrm{d}x \tag{5.1}$$

$$p_t^{\boldsymbol{j}}(x) \propto \int p_t^{\mathrm{s}}(x') q_t(x|x') p_{t-1}^{\boldsymbol{i}}(x') \mathrm{d}x', \tag{5.2}$$

and

$$w_t^{\boldsymbol{k}} = w_{t-1}^{\boldsymbol{i}} \int (1 - p_t^{\mathrm{s}}(x)) p_{t-1}^{\boldsymbol{i}}(x) \mathrm{d}x$$

$$p_t^{\boldsymbol{k}}(x) = \delta_\psi(x).$$

The hierarchical nature of the problem already appears in these prediction equations, and especially in the ones for $\boldsymbol{j}$, with the normalising constant in (5.2) appearing as the likelihood for the weight of the hypothesis in (5.1).

5.2 Update

We denote by $\ell_t(\cdot|x)$ the likelihood at time t for any $x \in \mathbb{R}^d \cup \{\psi\}$, the inclusion of ψ allowing for treating both $T = t$ and $T < t$ with the same update equations. Similarly, we define the probability of detection for any $x \in \mathbb{R}^d \cup \{\psi\}$ as $p_t^{\mathrm{d}}(x)$ and assume that

$p_t^{\mathrm{d}}(\psi) = 0$ so that the definition of $\ell_t(\cdot|\psi)$ is irrelevant. The likelihood $\ell_t(\cdot|x)$ is not required to be a probability distribution and we will actually consider

$$\ell_t(z|x) = \exp\Big(-\frac{1}{2}(H(x) - z)^T R^{-1}(H(x) - z)\Big)$$

where R is a positive-definite matrix and H is the observation function. This type of likelihood can seen as behaving like an indicator function of an event such as a target being detected within a group of pixels. Because of the hierarchical nature of the problem, the absence of normalising constant is important when evaluating the underlying data association.

The probability for an observation $z \in \mathbb{R}^{d'}$ to be a false alarm (or false positive) is denoted $w_t^{\mathrm{fa}}(z)$. We adopt an observation-driven birth mechanism and denote by $w_t^{\mathrm{b}}(z)$ the probability for the observation z to be originating from a new target and by $p_t^{\mathrm{b}}(\cdot|z)$ the corresponding probability distribution on $\mathbb{R}^d$.

We now consider an hypothesis $\boldsymbol{k} = (T, \boldsymbol{o} \times z)$ at time t, with $\boldsymbol{o} \times z$ being a sequence $\boldsymbol{o} \in \mathcal{O}_{t-1} \cup \{\boldsymbol{\phi}_{t-1}\}$ concatenated with a new observation $z \in Z_t \cup \{\phi\}$ and with $\boldsymbol{o} \times z \neq \boldsymbol{\phi}_t$. Denoting $\boldsymbol{i} = (T, \boldsymbol{o})$, the hypothesis $\boldsymbol{k}$ is associated with the probability distribution

$$p_t^{\boldsymbol{k}}(x) \propto \begin{cases} p_t^{\mathrm{d}}(x)\ell_t(z|x)p_t^{\boldsymbol{i}}(x) & \text{if } z \in Z_t \text{ and } \boldsymbol{o} \neq \boldsymbol{\phi}_t \\ (1 - p_t^{\mathrm{d}}(x))p_t^{\boldsymbol{i}}(x) & \text{if } z = \phi \text{ and } \boldsymbol{o} \neq \boldsymbol{\phi}_t, \\ p_t^{\mathrm{b}}(x|z) & \text{if } \boldsymbol{o} = \boldsymbol{\phi}_t, \end{cases}$$

which follows from Bayes' rule. The weight $w_t^{\boldsymbol{k}}$ can be defined in two different ways as

$$w_t^{\boldsymbol{k}} = \frac{\breve{w}_t(\boldsymbol{i}, z)\ell_t(z|\boldsymbol{i})w_t^{\boldsymbol{i}}}{1 - w_t^{\boldsymbol{i}} + \breve{w}_t(\boldsymbol{i}, \phi)\ell_t(\phi|\boldsymbol{i})w_t^{\boldsymbol{i}} + \sum_{z' \in Z_t} \breve{w}_t(\boldsymbol{i}, z')\ell_t(z'|\boldsymbol{i})w_t^{\boldsymbol{i}}} \tag{5.3}$$

when $\boldsymbol{o} \neq \boldsymbol{\phi}_{t-1}$, or as

$$w_t^{\boldsymbol{k}} = \frac{\breve{w}_t(\boldsymbol{i}, z)\ell_t(z|\boldsymbol{i})w_t^{\boldsymbol{i}}}{\breve{w}_t(\mathrm{fa}, z)w_t^{\mathrm{fa}}(z) + \breve{w}_t(\mathrm{b}, z)w_t^{\mathrm{b}}(z) + \sum_{\boldsymbol{j} \in \mathcal{I}_t} \breve{w}_t(\boldsymbol{j}, z)\ell_t(z|\boldsymbol{j})w_t^{\boldsymbol{j}}}, \tag{5.4}$$

when $z \neq \phi$, where $\ell_t(z|\boldsymbol{i})$ can be seen as the marginal likelihood of the observation $z \in Z_t \cup \{\phi\}$ for the target hypothesis $\boldsymbol{i}$, defined as

$$\ell_t(z|\boldsymbol{i}) = \begin{cases} \int p_t^{\mathrm{d}}(x)\ell_t(z|x)p_t^{\boldsymbol{i}}(x)\mathrm{d}x & \text{if } z \in Z_t \\ \int (1 - p_t^{\mathrm{d}}(x))p_t^{\boldsymbol{i}}(x)\mathrm{d}x & \text{if } z = \phi \end{cases}$$

and where $\breve{w}_t(\boldsymbol{i}, z)$ is the probability for the observations in $Z_t \setminus \{z\}$ to be originating from the hypotheses in $\mathcal{I}_t \setminus \boldsymbol{i}$ (or $\mathcal{I}_t$ when $\boldsymbol{i}$ is equal to "b" or "fa"). The update equations for the weight (5.3) and (5.4) are equivalent for the hypotheses for which they both apply, and correspond to enumerating all possibilities with respect to hypotheses or to observations.

If computed exactly, this update would not lead to any reduction in complexity; the principle of this approach is to compute $\breve{w}_t(\boldsymbol{i}, z)$ approximately in order to reduce the complexity to linear in the number of observations and in the number of target hypotheses. One such approximation can be found when assuming that $\ell_t(z'|\boldsymbol{j}) w_t^{\boldsymbol{j}} \ell_t(z'|\boldsymbol{j}') w_t^{\boldsymbol{j}'} \approx 0$ for any $z' \in Z_t$ and any $\boldsymbol{j}, \boldsymbol{j}' \in \mathcal{I}_t$ such that $\boldsymbol{j} \neq \boldsymbol{j}'$, which yields

$$\breve{w}_t(\boldsymbol{i}, z) \propto \frac{1}{A_t(\boldsymbol{i}, z)} \prod_{\boldsymbol{j} \in \mathcal{I}_t \setminus \{\boldsymbol{i}\}} \left[1 - w_t^{\boldsymbol{i}} + \ell_t(\phi|\boldsymbol{i}) + \sum_{z' \in Z_t \setminus \{z\}} \frac{\ell_t(z'|\boldsymbol{i})}{C_t(z')} \right] \tag{5.5}$$

where

$$C_t(z') = \frac{w_t^{\mathrm{b}}(z')}{1 - w_t^{\mathrm{b}}(z')} + \frac{w_t^{\mathrm{fa}}(z')}{1 - w_t^{\mathrm{fa}}(z')}$$

and

$$A_t(\boldsymbol{i}, z) = \begin{cases} 1 - w_t^{\mathrm{fa}}(z) & \text{if } \boldsymbol{i} = \mathrm{fa} \\ 1 - w_t^{\mathrm{b}}(z) & \text{if } \boldsymbol{i} = \mathrm{b} \\ C_t(z) & \text{if } z \neq \phi. \end{cases}$$

The targeted computational complexity can be achieved with this approximation of $\breve{w}_t(\boldsymbol{i}, z)$ because of the similarities in its expression for all hypotheses and all observations, e.g. the product in (5.5) can be computed once over $\mathcal{I}_t$ and then divided by the suitable term to obtain the value of the product over a given $\mathcal{I}_t \setminus \{\boldsymbol{i}\}$. The expression of $\breve{w}_t(\boldsymbol{i}, z)$ in (5.5) is expressed up to a multiplicative constant both for notational and computational simplicity; for instance, the probability $\breve{w}_t(\boldsymbol{i}, z)$ would depend on the number of potential target births at time t, but this terms cancels out in the computation of the weights $w_t^{\boldsymbol{i}}$ and is therefore irrelevant. From a computational viewpoint, the actual value of $\breve{w}_t(\boldsymbol{i}, z)$ tends to be very small and cancelling out terms allows to minimise issues with numerical accuracy.

Although hypotheses corresponding to targets that have not survived to the current time step can only be updated through their weight, this is an important aspect since it often take several time steps to understand that a target has not survived and the

formulation above allows to retain and update that information. This will also be crucial when performing track extraction.

5.3 Simulation

Similar as PHD filter, at first to simulate the single sensor case and then try different order of multi-sensors to see whether HISP filter would be easily influenced by the order of sensors. Here without lose of generation, we would like to put our simulation in a simpler scenario with truth described as figure5.1. Note that here for comparing we still put it in nonlinear case, which means there is EKF in update process.

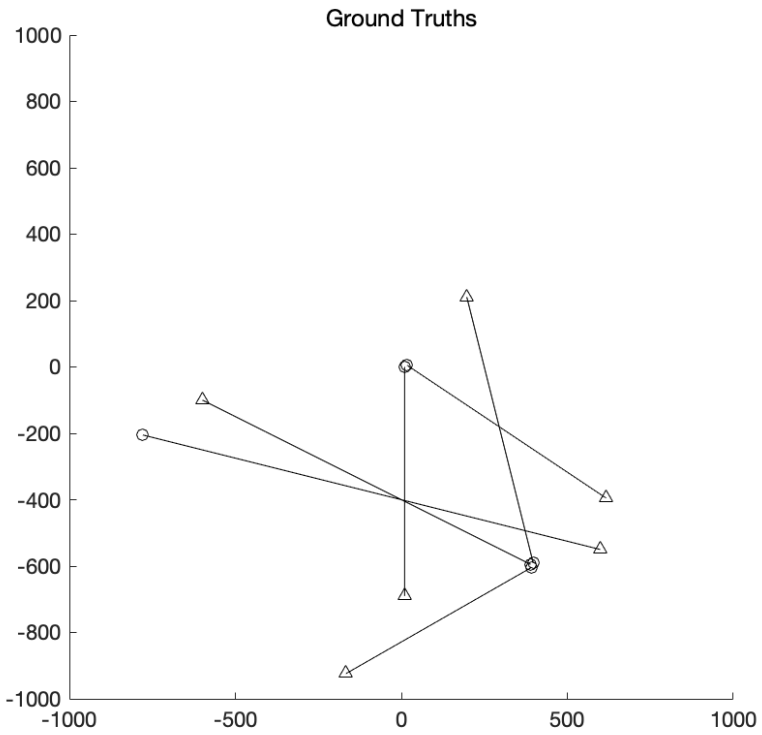

Figure 5.1: Targets truth in HISP

Table 5.1: Parameter of Simulation

P_S	P_D	κ_c	σ_v	σ_ρ	σ_θ	λ_b
0.99	0.9	10	1	10	$\pi/180$	0.1

[1] κ_c : intensity of clutters

[2] σ_v : std. deviation of process noise

[3] λ_b : poisson average rate of birth

[4] $\sigma_\rho, \sigma_\theta$: std. deviation of observation noise

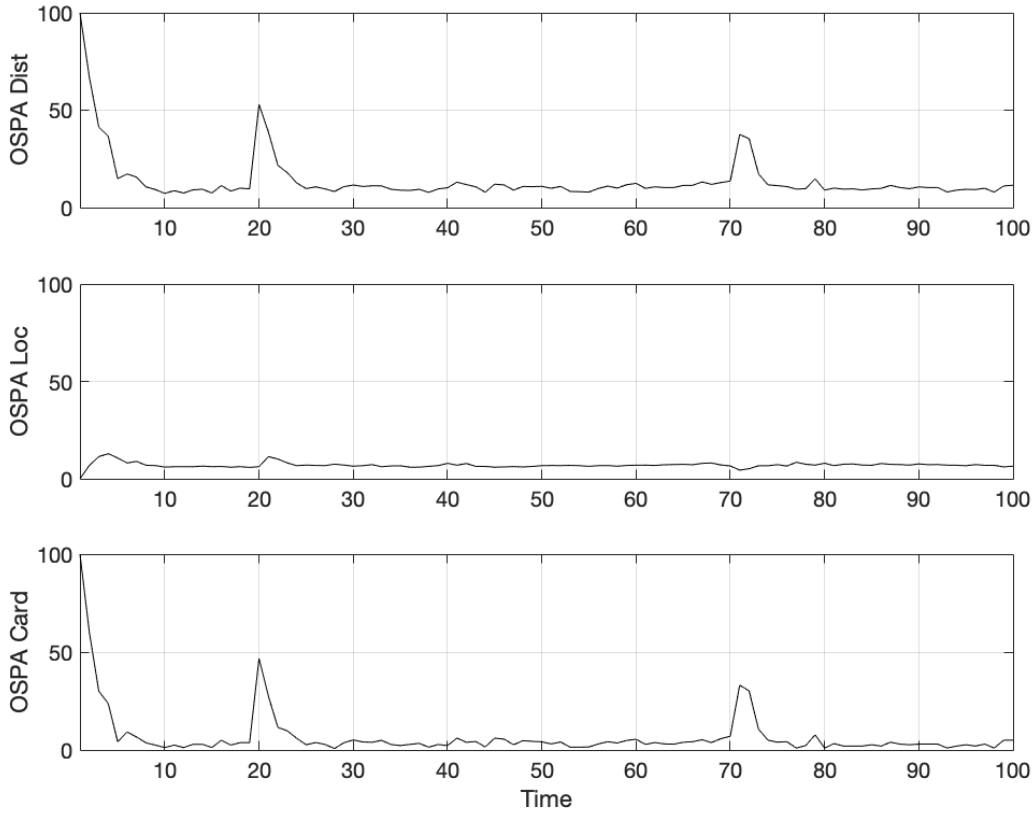

Figure 5.2: Average OSPA in HISP

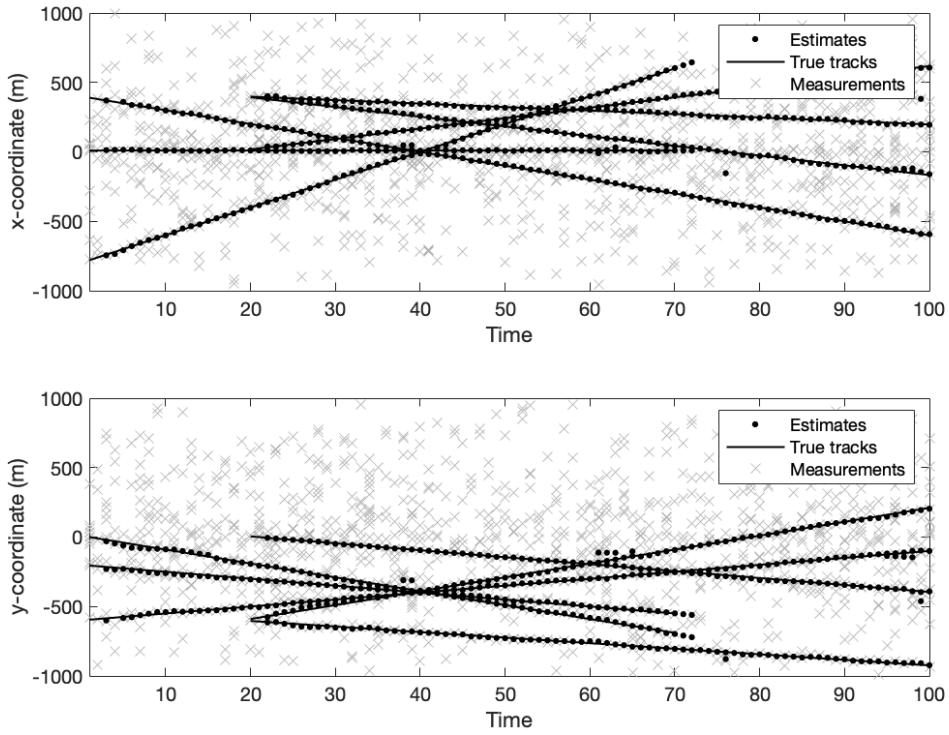

Figure 5.3: Estimation in HISP

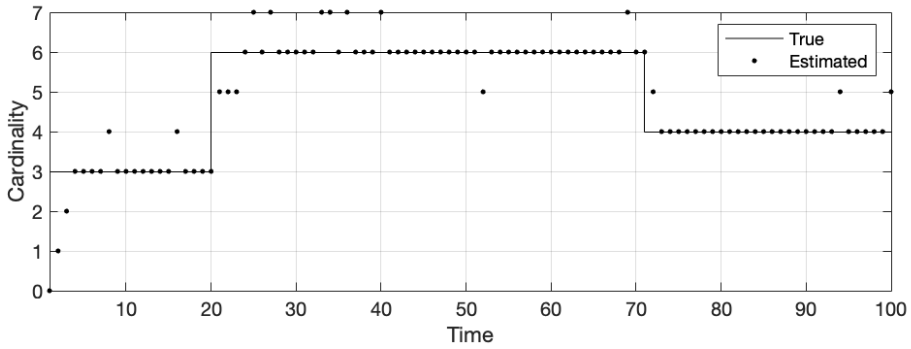

Figure 5.4: Cardinality estimation in HISP

Generally speaking, HISP filter is average better that the performance of PHD filter. Although it takes a little longer time to run HISP in the same condition as PHD filter, both the PHD and HISP filters have a linear complexity.[17] In the cardinality estimation

figure 5.4 there is a time lag between time 20 and 30 which show a increase process of Numbers of targets as here we set 3 births of new targets.

5.4 Update order in HISP

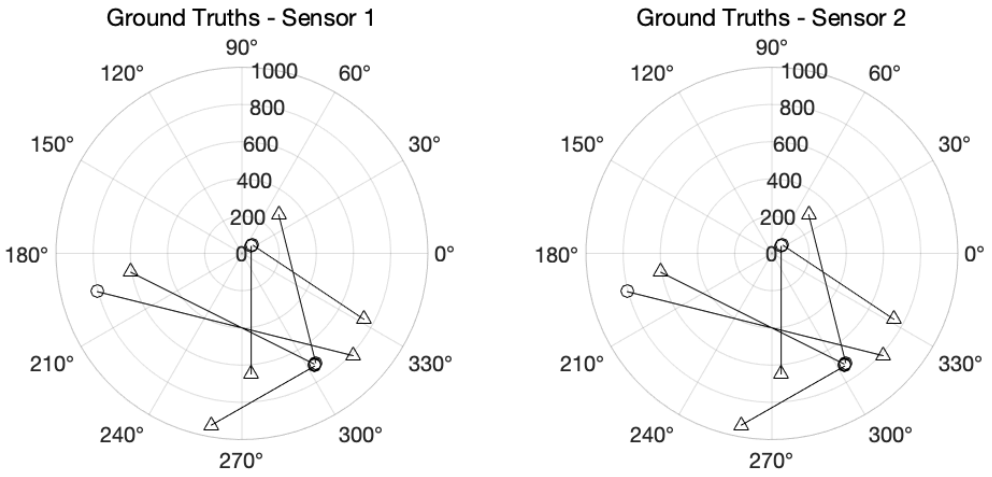

Figure 5.5: truth in two sensors

Table 5.2: Parameter of Simulation

sensors	P_S	P_D	κ_c	σ_v	σ_ρ	σ_θ	λ_b
sensor.1	0.99	0.9	10	1	10	$\pi/180$	0.1
sensor.2	0.99	0.7	10	1	10	$\pi/180$	0.1

[1] κ_c : intensity of clutters

[2] λ_b : poisson average rate of birth

[3] σ_v : std. deviation of process noise

[4] $\sigma_\rho, \sigma_\theta$: std. deviation of observation noise

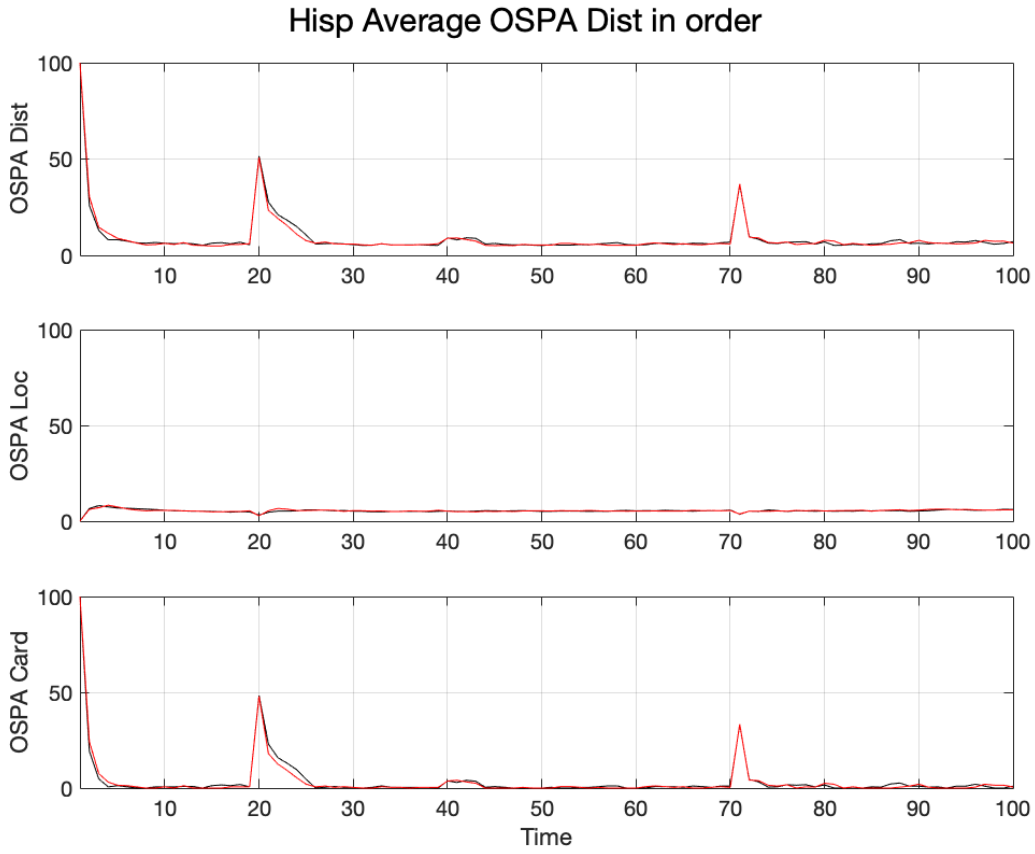

Figure 5.6: Average OSPA Compare of different order in HISP

The OSPA comparison in figure 5.6 shows a pretty good results as they HISP Filter literally perform the same no matter what the orders of sensors are, which prove that HISP filter has a good robust property so that we can always ignore the update order of

different sensors. One of the reason that we could expect the better performance of HISP filter is that the approximations HISP makes is not as strong as PHD filter does.

5.5 Scanning sensor

One of the standard assumptions in multi-target tracking is that set of observations received at each time step corresponds to the observations of all targets at once. However, many real sensors do not capture such a snapshot of the scene and instead scan different locations at different times. We consider the case of a rotating sensor such as many radars. More specifically, we assume that the sensor measures the distance from the sensor to the target, referred to as the *range*, as well as the angular position of target from the viewpoint of the sensor. Each set of observations now corresponds to an angular sector $S_t \subseteq [0, 2\pi)$ of the sensor and it takes several time steps for a detection to be attempted on all targets. The probability of detection in this case can be assumed to be of the form

$$p_t^{\mathrm{d}}(x) = \begin{cases} c_{\mathrm{d}} & \text{if } H_\theta(x) \in S_t \\ \exp\Big(-\frac{1}{2\sigma_{\mathrm{d}}^2} d(H_\theta(x), S_t)^2 \Big) & \text{otherwise,} \end{cases}$$

where $c_{\mathrm{d}} \in [0, 1]$ is the constant probability of detection within the sector, H_θ is the component of the observation function H related to the angular observation, σ_{d} is the standard deviation of the angular uncertainty in the observation process and $d(\theta, S_t)$ is the distance between the point θ and the set S_t. Most scanning Sensors would also have built-in bounds for the range to allow for the next phase of the observation to start but this is not modelled here for the sake of simplicity. As the p_d are not constant anymore the posterior are no longer normal distributed so instead of EKF it would be better to use UKF here, which would be more simply than calculating the Jacobian Matrix.

Simulation

In practical simulation assume there is a radar constantly scan in circles, then we can set a period t_s which means that with t_s time steps we scan a circle 2π. So each time step we scan a sector of $2\pi/t_s$ in which we set the probability of detection as c_d.

Each time step with the distance become larger from the sector, the probability fade as exponential way. So in our simulation we can set a sector function of time and space indicate the probability of detection for certain measurement. As for clutter, we still consider a uniform way of clutter generation in the field of view but divide the intensity into sectors. In HISP, when we consider local hypothesis, we would connect the recent N_h time steps, but when we do scanning it would takes $N_h \times t_s$ to connect the same length of hypothesis.

At first try a simple case, set the scanning period to 2 which means each time the scan would cover a range of π. Assume the target truth set as Figure **??** with truth and estimate cardinality in Figure **??**.

Table 5.3: Parameter of Simulation

t_s	c_{d}	σ_d	P_S	$\kappa_{c,s}$	σ_v	σ_ρ	σ_θ	$\lambda_{b,s}$
2	0.98	$\pi/16$	0.99	2.5	0.1	10	$\pi/180$	0.05

[1] c_{d}, σ_d : parameters in equation $p_t^{\mathrm{d}}(x)$

[2] t_s : time steps in one scanning period

[3] $\kappa_{c,s} = \kappa_c / t_s$: poisson average rate of clutters in one sector

[4] $\lambda_{b,s} = \lambda_b / t_s$: possion average rate of birth in each sector

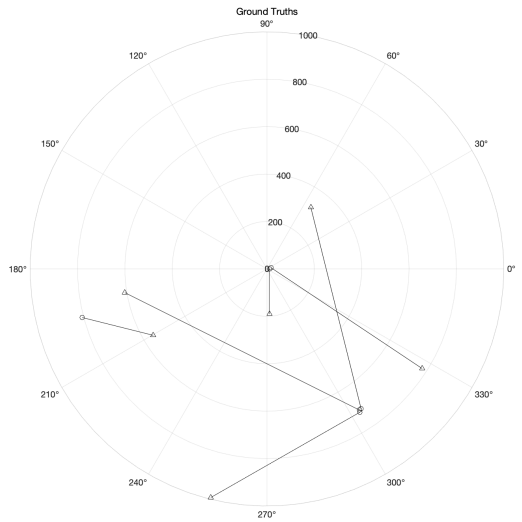

Figure 5.7: truth for scanning sensor

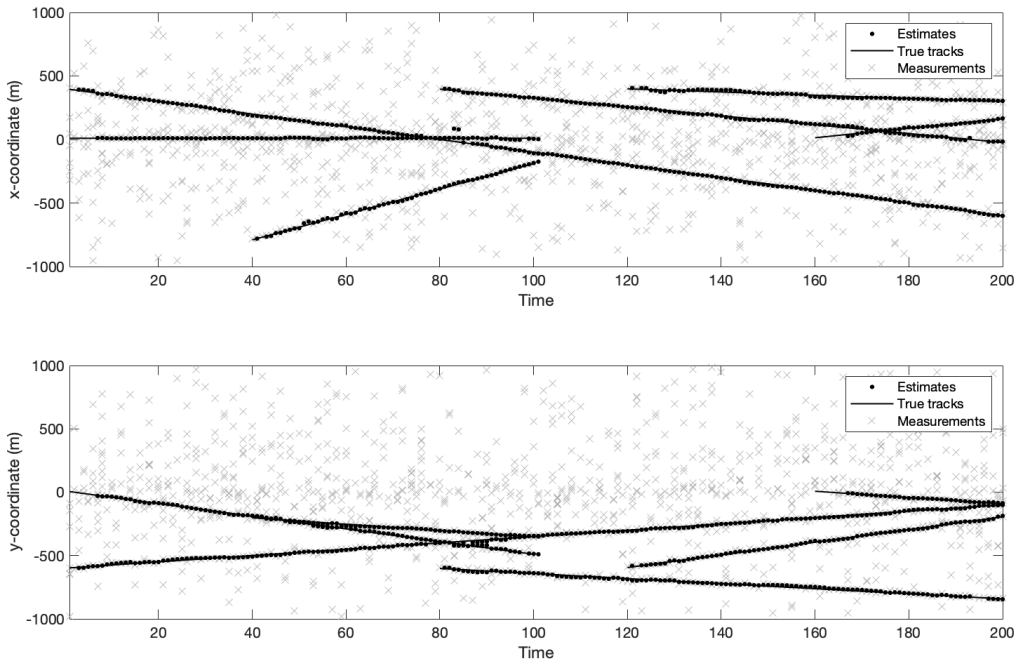

Figure 5.8: estimate of scanning sensors

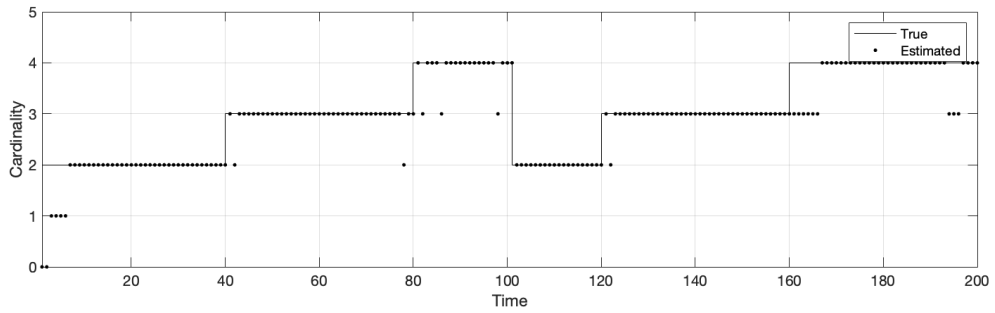

Figure 5.9: cardinality estimates

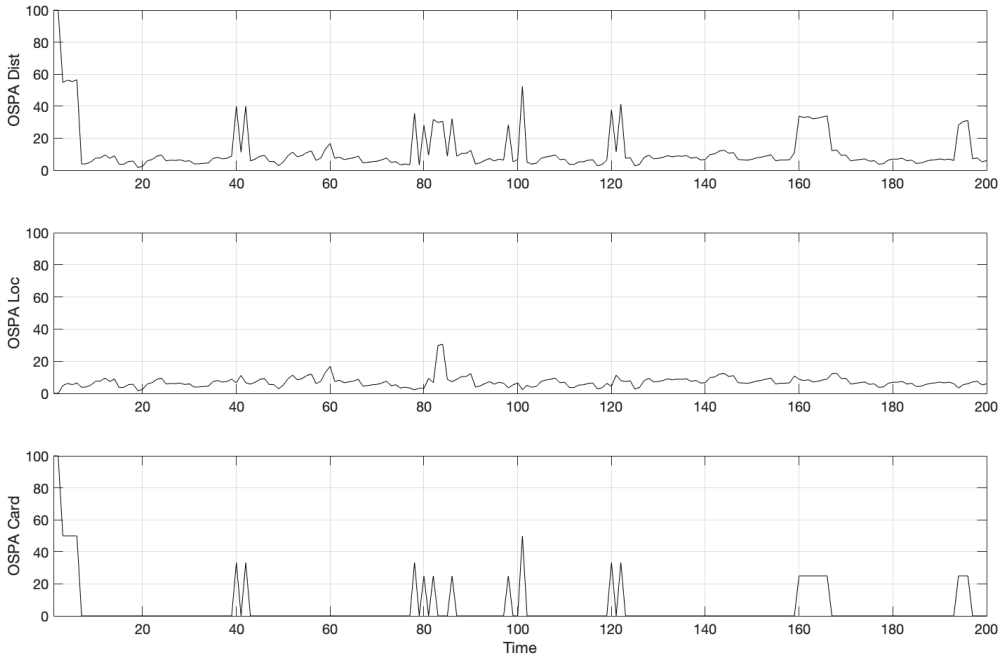

Figure 5.10: OSPA of estimate

In the simulation of scanning sensor, the choice of the σ_{d} is important as it decides how fast $p_t^{\mathrm{d}}(x)$ vanishes with the angular distance. At the same time, we need to consider the comparative angular size in each sector so that we would not let the probability of detection change little through the whole scan, for example if the sector angle was set to π and $c_{\mathrm{d}} = \pi/2$ then the $p_t^{\mathrm{d}}(x)$ would be similar around all states x, which means that it equals to a whole scan each time step.

Then try a more practical case that the whole scan of 2π is seperated to 12 sectors with each scan cover $\pi/6$ of view. And decrease the variance of observation relatively. From the simulation results, it's obvious that there is time lags in Figure 5.12 with respect to the peaks in OSPA figure 5.13. The lags is longer than general case cause the scanning sensor would experience more time step to run a whole circle so it takes more time step to find new births or deaths.

Table 5.4: Parameter of Simulation

t_s	c_{d}	σ_d	P_S	$\kappa_{c,s}$	σ_v	σ_ρ	σ_θ	$\lambda_{c,s}$
12	0.98	$\pi/30$	0.99	5/12	0.1	10	$\pi/180$	0.01

[1] κ_c : c_{d}, σ_d parameters in equation $p_t^{\mathrm{d}}(x)$

[2] t_s : time steps in one scanning period

[3] $\kappa_{c,s} = \kappa_c / t_s$: poisson average rate of clutters in one sector

[4] $\lambda_{b,s} = \lambda_b / t_s$: possion average rate of birth in each sector

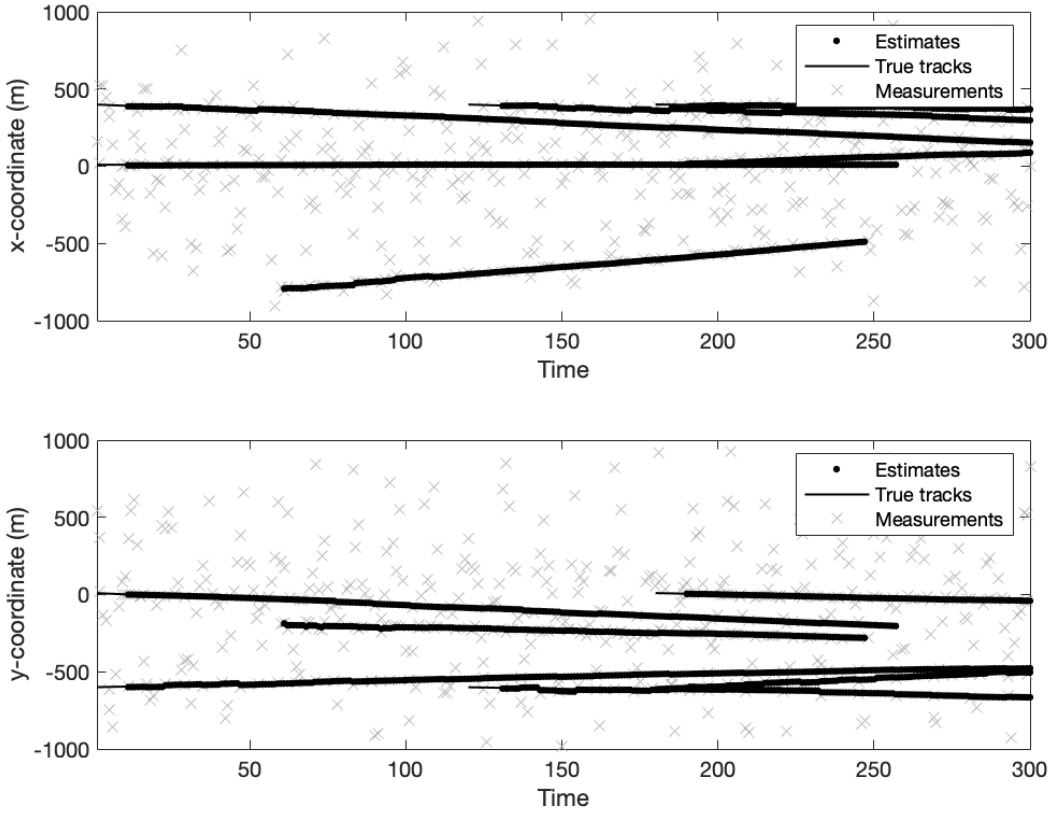

Figure 5.11: estimate of scanning sensors

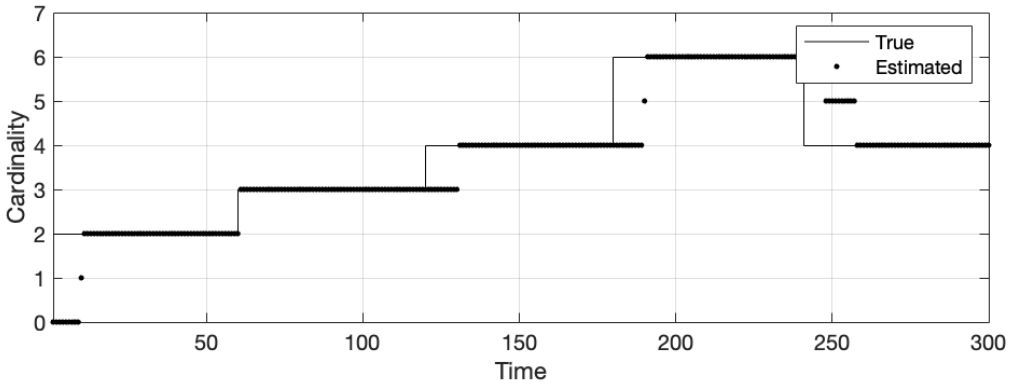

Figure 5.12: cardinality estimates

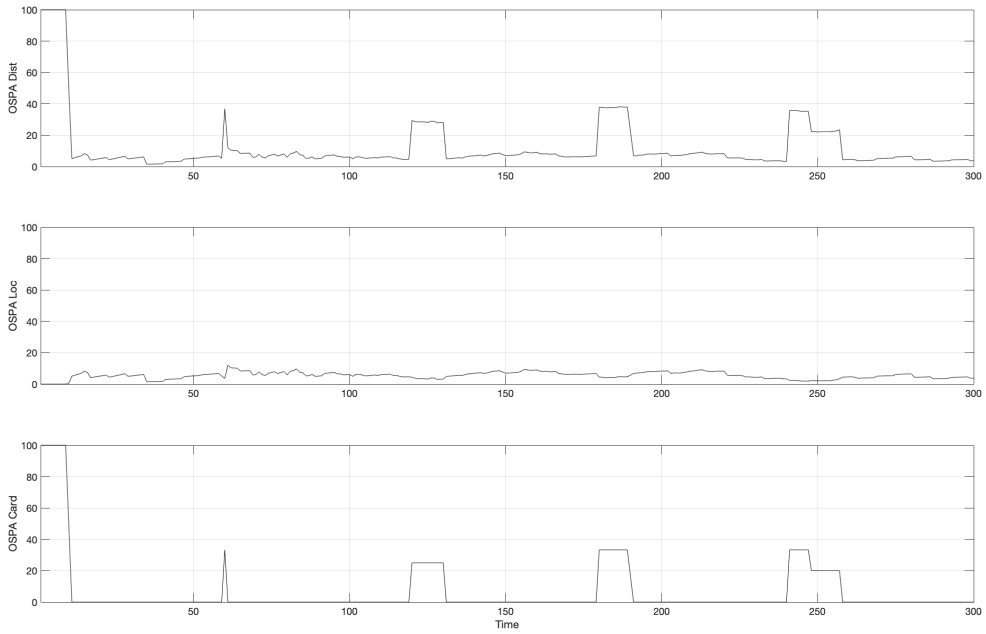

Figure 5.13: OSPA of estimate

Chapter 6

CONCLUSION

In this article I explore the region of Non-linear scenario of PHD filter, influence of update order in PHD and HISP filter and scanning sensor scenario in HISP. And get conclusion as following:

- In Non-linear scenario, I tried local linearization as the solution in PHD filter by using the Taylor expansion and calculating Jacobian matrix. In the simulation, what the figure and performance indicator shows is that extend Kalman filter actually works in non-linear scenario although it do not perform as well as linear case.
- In terms of PHD Filter, it is obvious that the order of sensors for updating does have remarkable influence on the performance of tracking algorithm: If we ended updating with observations from relative bad sensors, the performance tend to be bad.
- In terms of HISP Filter, there tends to be no influence of the update order on the performance of tracking. This proves that HISP filter is more robust in practical implementation.
- In terms of scanning sensor, the choice of the σ_{d} is pretty important as it decides how fast $p_t^{\mathrm{d}}(x)$ would faded with angular distance. And there is more time lags showing in estimate if we decrease the scanning field each time.

Bibliography

[1] Samuel S Blackman. "Multiple hypothesis tracking for multiple target tracking". In: *IEEE Aerospace and Electronic Systems Magazine* 19.1 (2004), pp. 5–18.

[2] Thomas Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. "Sonar tracking of multiple targets using joint probabilistic data association". In: *IEEE journal of Oceanic Engineering* 8.3 (1983), pp. 173–184.

[3] Hedvig Sidenbladh. "Multi-target particle filtering for the probability hypothesis density". In: *arXiv preprint cs/0303018* (2003).

[4] Ronald P.S. Mahler. "Multitarget Bayes Filtering via First-Order Multitarget Moments". In: *IEEE Transactions on Aerospace and Electronic Systems* 39.4 (2003). Cited by: 2003, pp. 1152–1178. DOI: 10.1109/TAES.2003.1261119. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-1642401353&doi=10.1109%2fTAES.2003.1261119&partnerID=40&md5=6a78ccd4649ec45005ee7ea49960fa06.

[5] Robert A Singer and John J Stein. "An optimal tracking filter for processing sensor data of imprecisely determined origin in surveillance systems". In: *1971 IEEE Conference on Decision and Control*. IEEE. 1971, pp. 171–175.

[6] Yaakov Bar-Shalom and Edison Tse. "Tracking in a cluttered environment with probabilistic data association". In: *Automatica* 11.5 (1975), pp. 451–460.

[7] Yaakov Bar-Shalom, Thomas E Fortmann, and Peter G Cable. *Tracking and data association*. 1990.

[8] Donald Reid. "An algorithm for tracking multiple targets". In: *IEEE transactions on Automatic Control* 24.6 (1979), pp. 843–854.

[9] Yanhua Ruan, Peter Willett, and Roy Streit. "The PMHT for maneuvering targets". In: *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207)*. Vol. 4. IEEE. 1998, pp. 2432–2433.

[10] JFC Kingman. "G. Matheron, Random sets and integral geometry". In: *Bulletin of the American Mathematical Society* 81.5 (1975), pp. 844–847.

[11] B-N Vo and W-K Ma. "A closed-form solution for the probability hypothesis density filter". In: *2005 7th International Conference on Information Fusion*. Vol. 2. IEEE. 2005, 8–pp.

[12] B-N Vo and W-K Ma. "The Gaussian mixture probability hypothesis density filter". In: *IEEE Transactions on signal processing* 54.11 (2006), pp. 4091–4104.

[13] Ronald Mahler. "PHD filters of higher order in target number". In: *IEEE Transactions on Aerospace and Electronic systems* 43.4 (2007), pp. 1523–1543.

[14] Ronald PS Mahler. *Statistical multisource-multitarget information fusion*. Vol. 685. Artech House Norwood, MA, USA, 2007.

[15] Ba-Tuong Vo, Ba-Ngu Vo, and Antonio Cantoni. "The cardinality balanced multi-target multi-Bernoulli filter and its implementations". In: *IEEE Transactions on Signal Processing* 57.2 (2008), pp. 409–423.

[16] Nathanael L Baisa et al. "Online Multi-target Visual Tracking using a HISP Filter." In: *VISIGRAPP (5: VISAPP)*. 2018, pp. 429–438.

[17] Jeremie Houssineau and Daniel E Clark. "Multitarget filtering with linearized complexity". In: *IEEE Transactions on Signal Processing* 66.18 (2018), pp. 4957–4970.

[18] Ba-Tuong Vo, Ba-Ngu Vo, and Antonio Cantoni. "Bayesian filtering with random finite set observations". In: *IEEE Transactions on signal processing* 56.4 (2008), pp. 1313–1326.

[19] Michael Beard, Ba Tuong Vo, and Ba-Ngu Vo. "OSPA(2): Using the OSPA metric to evaluate multi-target tracking performance". In: *2017 International Conference on Control, Automation and Information Sciences (ICCAIS)*. 2017, pp. 86–91. DOI: 10.1109/ICCAIS.2017.8217598.

[20] Andrew H Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.

[21] Brian DO Anderson and John B Moore. *Optimal filtering*. Courier Corporation, 2012.

[22] Simon J Julier and Jeffrey K Uhlmann. "New extension of the Kalman filter to nonlinear systems". In: *Signal processing, sensor fusion, and target recognition VI*. Vol. 3068. Spie. 1997, pp. 182–193.

[23] Simon J Julier and Jeffrey K Uhlmann. "Unscented filtering and nonlinear estimation". In: *Proceedings of the IEEE* 92.3 (2004), pp. 401–422.

[24] Long Liu, Hongbing Ji, and Zhenhua Fan. "Improved Iterated-corrector PHD with Gaussian mixture implementation". In: *Signal Processing* 114 (2015), pp. 89–99.