WAC 2014 1569945011

# Noise-Estimate Particle PHD filter

Masanori Ishibashi, Yumi Iwashita, *and* Ryo Kurazume
Kyushu University
744, Motooka, Nishi-ku, Fokuoka, 819-0395, Japan
Email: ishibashi@irvs.ait.kyushu-u.ac.jp, yumi@ieee.org, kurazume@ait.kyushu-u.ac.jp

***Abstract*—This paper proposes a new radar tracking filter named Noise-estimate Particle PHD Filter (NP-PHDF). Kalman filter and particle filter are popular filtering techniques for target tracking. However, the tracking performance of the Kalman filter severely depends on the setting of several parameters such as system noise and observation noise. It is an open problem how to choose proper parameters for various scenarios, and they are often regulated in trial-and-error manner. To tackle this problem, Noise-estimate Particle Filter (NPF) has been proposed so far. The NPF estimates proper noise parameters of a Kalman filter on-line based on a scheme of particle filter. In this paper, we extend the NPF so that it enables to track multiple targets simultaneously by combining with Probability Hypothesis Density (PHD) filter, and propose a new Noise-estimate Particle PHD Filter (NP-PHDF). Simulation results show that the proposed filter has higher tracking performance in various scenarios than conventional Kalman filter, particle filter, and PHD filter for multiple-targets tracking.**

## I. Introduction

Target tracking is a fundamental technique in not only computer vision but also various application fields. For example, in air traffic control by a radar sensor, multiple high-speed flight vehicles must be tracked simultaneously without lost-tracking while estimating their velocities and heights. However, in general, radar data is severely corrupted by noise due to atmospheric conditions or radar reflection property of the target, and it is still challenging to track targets stably under severe noisy condition.

For target tracking, a time series filter is very effective to suppress noise in sensory data and track targets smoothly and stably. In a time series filter, the current target position, which is estimated based on past observation and a motion model, are merged with current observation and the optimum position is derived. Especially, the time series filter which removes the noise component in the radar data and estimates the current position and velocity of the target is called "tracking filter".

Kalman filter is the most popular and widely-used tracking filter and has been applied to various applications so far. Based on the assumptions on linear and Gaussian noise in sensory data, this filter estimates the statistically-optimized state of the target. Uniform linear motion is usually adopted as a motion model in simple Kalman filter. In case that target motion does not follow the uniform linear model, a system noise parameter which should be adjusted beforehand to absorb the system error affects the tracking performance severely. However, it is an open problem how to choose a proper value of the system noise parameter beforehand for various scenarios such as non-linear motion or rapid acceleration/deceleration. In addition, an observation error which represents the accuracy of sensor/radar and has to be adjusted beforehand affects the performance of the Kalman filter, too. The observation model is also difficult to be set appropriately for various targets with a variety of shapes, motion directions, heights, distances, etc. Currently, a target tracking system based on Kalman filter has to be designed with proper system parameters in a try-and-error manner to meet the desired performance [1].

In few decades, particle filter has been attracting much attention as a high-performance tracking filter. In particle filter, instead of estimating the probability distribution of the object state from the past observation and motion model in a parametric way, the probability distribution is represented by a set of particles. Each particle has a weight which represents the probability of the state of the particle. The particle is updated and re-sampled according to the Bayesian recursion equation.

Particle filter has been applied to various applications such as human tracking, state estimation, etc. [2], [3], [4], [5], [6], [7], [8]. Particle filter does not depend on the assumption of linear or Gaussian noise, and is able to be applied for various systems even with non-linear and non-Gaussian noise. However, the particle filter is less effective than the optimized Kalman filter for a target moving by a motion model, and the improvement of the tracking performance for various conditions is an open problem in particle filter.

We have proposed a tracking filter named noise-estimate particle filter (NPF) [9], which combines Kalman filter and particle filter. As mentioned above, Kalman filter is an optimum filter in case that the motion model of targets and observation model of sensor/radar are correctly provided. In NPF, instead of estimating the state of the target such as position or velocity directly, the system error and the observation error are estimated on-line by particle-filter based approach. More correctly, the state of the target is estimated based on Kalman filter, and its motion noise and observation noise are optimized by particle filter. Since the critical parameters of the Kalman filter are adjusted on-line according the observation, the proposed filter can be applied for a variety of target motions including not only uniform linear motion, but also sudden motion changes such as abrupt acceleration/deceleration or steep turn. However, the NPF is limited to a single target tracking and cannot be applied to multiple target tracking problem directly.

In this paper, we extend NPF so that it enables to track multiple targets simultaneously, and propose a new tracking filter named Noise-estimate Particle PHD Filter (NP-PHDF). Probability Hypothesis Density (PHD) filter is a multiple-target tracking filter proposed by Vo et al.[10],[11]. In this filter, multiple state spaces for multiple targets are combined to a single state space, and their probability distributions are expressed by a set of particles. Based on PHD filter, NPF is extended so that it can track multiple targets stably by estimating optimal parameters of Kalman filter for individual target, and a new Noise-estimate Particle PHD Filter (NP-PHDF) is introduced in this paper. The performance of the proposed NP-PHDF is verified through computer simulation for multiple-target tracking problems.

Some authors proposed combination filters of particle filter and Kalman filter[12], [13], [14], [15]. Rao-Blackwellized Particle Filters [14] is the most popular technique in SLAM (Simultaneous Localization and Mapping). Localization and mapping procedures are separated in this filter and implemented using particle filter and Kalman filter, individually. Marginalized Particle Filter proposed Schon et al. [15] separates linear and nonlinear parts in the control system and assigns Kalman filter and particle filter separately. However, to our best knowledge, the proposed filter which utilizes particle filter for estimating the parameters of optimum Kalman filter has not been proposed so far. Satoh et al. [13] proposed a color-based tracking technique using Kalman Particle Filter [12]. As the noise-estimate particle filter proposed in the paper, the state of each particle is updated by Kalman filter. In their method, however, the observation noise is determined according to the weight of the particle and the state noise is determined previously.

## II. Noise-estimate Particle Filter (NPF)

In the noise-estimate particle filter (NPF)[9], a number of Kalman filters with a variety of sets of parameters runs simultaneously in a parallel way. As conventional particle filter, particles are re-sampled according to the errors between the estimated and observed states. By evaluating the performance of a number of Kalman filters with a variety of motion and observation parameters estimated adaptively, optimum target tracking which has similar high performance as optimized Kalman filter for a fixed condition is achieved for various motion patterns including uniform linear motion, abrupt acceleration/deceleration or steep turn.

### A. Model definition

In NPF, each particle executes a similar process as Kalman filter individually. Firstly, a state space model for a target system is defined in each particle.

$$\mathbf{x}_{k+1}^i = \mathbf{\Phi}_k \mathbf{x}_k^i + \mathbf{w}_k^i \quad (1)$$
$$\mathbf{z}_k^i = \mathbf{H}\mathbf{x}_k^i + \mathbf{v}_k^i \quad (2)$$

Eq.(1) is a motion model which represents a state transition and Eq.(2) is an observation model which shows the relation of estimated and observed states. $\mathbf{x}_k^i$ is a state vector which contains position and velocity terms. $\mathbf{\Phi}_k$ is a state transition matrix and, in this paper, the uniform linear motion is assumed in all particles as follows.

$$\mathbf{\Phi}_k = \begin{pmatrix} \mathbf{I} & \Delta t\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \quad (3)$$

where $\Delta t$ is a sampling interval, $\mathbf{I}$ is an identity matrix, and $\mathbf{H} = (\mathbf{I}\ \ \mathbf{0})$ is an observation matrix. $\mathbf{w}_k^i$ is a vector of system noise with an average $\mathbf{0}$ and an error covariance matrix $\mathbf{Q}_k^i$, respectively. $\mathbf{v}_k^i$ is also a vector of observation noise with an average $\mathbf{0}$ and an error covariance matrix $\mathbf{R}_k^i$.

### B. Tracking process

A set of particles at time $t_k$ is defined as $\mathbf{X}_k = \left\{\mathbf{x}_k^i, w_k^i, q_k^i, r_k^i\right\}_{i=1}^N$. Here, $\mathbf{x}_k^i$ is a hypnosis for a state vector of position and velocity, $w_k^i$ is a weight of each particle, and $q_k^i$ and $r_k^i$ are system and observation noises at each particle, respectively.

1) Produce $N$ initial particles $\mathbf{X}_0 = \left\{\mathbf{x}_0^i, w_0^i, q_0^i, r_0^i\right\}_{i=1}^N$. The position, velocity, and system and observation noises are set with random numbers in particular ranges.
2) Execute step (a) to step (f) at time $t_k$ ($k = 1, \cdots, T$)
(a) Estimation
Execute prediction procedure in each particle by Kalman filter and estimate the current state from the previous state and the motion model.

$$\mathbf{x}_{k|k-1}^i = \mathbf{\Phi}_k \mathbf{x}_{k-1|k-1}^i \quad (4)$$
$$\mathbf{P}_{k|k-1}^i = \mathbf{\Phi}_k \mathbf{P}_{k-1|k-1}^i \mathbf{\Phi}_k^T + \mathbf{Q}_k^i \quad (5)$$

where $\mathbf{x}_{k|k-1}^i$ is an estimated state in each particle and $\mathbf{x}_{k-1|k-1}^i$ is a previous state at time $t_{k-1}$. $\mathbf{P}_{k|k-1}^i$ is an error covariance matrix in each particle and $\mathbf{P}_{k-1|k-1}^i$ is an previous error covariance matrix at time $t_{k-1}$.
An error covariance matrix of system noise $\mathbf{Q}_k^i$ at time $t_k$ is obtained as follows.

$$\mathbf{Q}_k^i = \mathrm{diag}\left\{{q_{k,x}^i}^2, {q_{k,y}^i}^2, {q_{k,z}^i}^2\right\} \quad (6)$$

(b) Smoothing
To fit the estimated state with the current observation and estimate more accurate state, smoothing process in Kalman filter is applied in each particle.

$$\mathbf{x}_{k|k}^i = \mathbf{x}_{k|k-1}^i + \mathbf{K}_k^i[\mathbf{z}_k - \mathbf{H}\mathbf{x}_{k|k-1}^i] \quad (7)$$
$$\mathbf{P}_{k|k}^i = (\mathbf{I} - \mathbf{K}_k^i\mathbf{H})\mathbf{P}_{k|k-1}^i \quad (8)$$
$$\mathbf{K}_k^i = \mathbf{P}_{k|k-1}^i\mathbf{H}^T[\mathbf{H}\mathbf{P}_{k|k-1}^i\mathbf{H}^T + \mathbf{R}_k^i]^{-1} \quad (9)$$

where $\mathbf{x}_{k|k}^i$ is the estimated state of the particle at time $t_k$, $\mathbf{P}_{k|k}^i$ is the estimated error covariance matrix in each particle, and $\mathbf{K}_k^i$ is a gain matrix.
The error covariance matrix of observation noise $\mathbf{R}_k^i$ at time $t_k$ is obtained as follows.

$$\mathbf{R}_k^i = \mathrm{diag}\{{r_{k,x}^i}^2, {r_{k,y}^i}^2, {r_{k,z}^i}^2\} \quad (10)$$

(c) Likelihood calculation
The likelihood $p(\mathbf{z}_k|\mathbf{x}_{k|k}^i)$ at each particle is calculated as follows.

$$p(\mathbf{z}_k|\mathbf{x}_{k|k}^i) = \frac{1}{\sqrt{2\pi\sigma_s^2}}\exp\left(\frac{-d_i^2}{2\sigma_s^2}\right) \quad (11)$$

where $\sigma_s$ is a parameter to evaluate the accuracy of the hypnosis, $d_i$ is an Euclidean distance between the position component in $\mathbf{x}_{k|k}^i$ and the observed position $\mathbf{z}_k$. In the following experiments, we set $\sigma_s = 300$.
Next, the weight of each particle is updated according to the obtained likelihood as follows.

$$w_k^i = w_{k-1}^i p(\mathbf{z}_k|\mathbf{x}_{k|k}^i) \quad (12)$$

In addition, the sum of the weight of all particles is calculated by $W_k = \Sigma_{i=1}^{N} w_k^i$ and the weight of each particle is normalized as $w_k^i = w_k^i / W_k$.
(d) State estimation
Estimated state $\hat{\mathbf{x}}_k$ at time $t_k$ is calculated by the weighted mean of $N$ particles.

$$\hat{\mathbf{x}}_k \approx \sum_{i=1}^{N} w_k^i \mathbf{x}_{k|k}^i \tag{13}$$

(e) Resampling
Particles are re-sampled according the the probability proportional to the weight value $w_k^i$. As a result, particles with lower weight are removed and ones with higher weight are increased.
(f) Update
In contrast to updating position and/or velocity in a conventional particle filter, random offset values obtained with a normal distribution are added to the system noise $q_k^i$ and the observation noise $r_k^i$, respectively.

$$q_{k,s}^i = q_{k,s}^i + \Delta q_{k,s}^i \tag{14}$$
$$r_{k,s}^i = r_{k,s}^i + \Delta r_{k,s}^i \tag{15}$$

where $s \in \{x, y, z\}$, and $\Delta q_{k,s}^i$ and $\Delta r_{k,s}^i$ are determined according to the normal random number with an average of 0 and a variances of $\sigma_{q,s}$ and $\sigma_{r,s}$, respectively. $\sigma_{q,s}$ and $\sigma_{r,s}$ are predetermined parameters, and in the following experiments, we set $\sigma_{q,s} = 0.1$ and $\sigma_{r,s} = 5$ for all $s \in \{x, y, z\}$.

## III. Noise-estimate Particle PHD Filter (NP-PHDF)

Probability Hypothesis Density (PHD) filter is a multiple-target tracking filter proposed by Vo et al.[10],[11]. In this section, we introduce a basic procedure of PHD filter and propose a new noise-estimate particle PHD filter which combines PHD filter and NPF mentioned above.

### A. Probability Hypothesis Density (PHD) filter

Let a set of particle at time $t_k$ be $\mathbf{X}_k = \{\mathbf{x}_k^i, w_k^i, j_k^i\}_{i=1}^{L_k}$. Here, $\mathbf{x}_k^i$ is a hypnosis for a state vector of position and velocity, $w_k^i$ is a weight of each particle, and $j_k^i$ is the label indicating which trajectory the particle is assigned to at previous time.

At time $t_k (k > 0)$,

1) Estimation: Particles at current time $t_k$ are updated by

$$\mathbf{x}_k^i = \mathbf{\Phi}_k \mathbf{x}_{k-1}^i, i = 1, 2, \cdots, L_{k-1} \tag{16}$$

Where $L_{k-1}$. is the number of particles at the previous time. In addition, new $J_k$ particles are produced at each time so that it can track new targets. $J_k$ is determined as

$$J_k = \alpha \int P_B(x) dx \tag{17}$$

$\alpha$ is a number of particles assigned for single target and $P_B$ is the probability of occurrence of a new target. In addition, the weight $w_k^i$ and the label $j_k$ is set as follows.

$$w_k^i = \frac{1}{\alpha} \tag{18}$$

$$j_k^i = 0 \tag{19}$$

for $i = L_{k-1} + 1, \cdots, L_{k-1} + J_k$.

2) Likelihood calculation: The weight of each particle is calculated according to the following equation.

$$w_k^i = w_{k-1}^i \big(1 - P_D(\mathbf{x}_k^i) + \sum_{m=1}^{M_k} \frac{P_D(\mathbf{x}_k^i) L(z_k^m \mid \mathbf{x}_k^i)}{\sum_{n=1}^{L_{k-1}+J_k} P_D(\mathbf{x}_k^n) L(z_k^m \mid \mathbf{x}_k^n) w_{k-1}^n}\big) \tag{20}$$

Where $w_k^i$ is the weight of the particle $i$ at time $t_k$ and $z_k^m$ is an observed value of $m^{th}$ target ($m = 1 \sim M_k$). $P_D(x_k^i)$ is the probability for a particle $x_k^i$ to detect the target.

3) Resampling: Particles are re-sampled according the the probability proportional to the weight value $w_k^i$. In PHD filter, the sum of the weights in a certain area indicates the expectation of the number of targets $N_k$ in this area. With this expectation, the number of particles $L_k$ at the next period is determined as follows.

$$N_k = \sum_{n=1}^{L_{k-1}+J_k} w_t^n \tag{21}$$

$$L_k = \alpha N_k \tag{22}$$

4) Update: Noises are added to the position and the velocity in each particle.

5) Clustering: Apply clustering algorithm and determine the index of the cluster $c_k^n$ which the particle belongs to.

$$[\{\mathbf{x}_k^{i_1}, c_k^{i_1}\}_{i_1=1}^{L_k}\}] = Clustring(\mathbf{x}_k^{i_1}) \tag{23}$$

6) Labeling of clusters: Calculate the correlation with the existing clusters of trajectories.

$$Id(i_1) = \arg_{i_2} \max\{SumWgt(i_1, i_2)\} \tag{24}$$

Where $i_2 = 1, \cdots, r_{max}$ and $SumWgt(i_1, i_2)$ is a sum of likelihoods in particles of $j = i_2$ among the whole particles belonging to the cluster $i_1$.

7) Correlation calculation: For all clusters,
$if \ \ Id(i_1) = 0$
  New trajectory
  $r = r_{max} + 1$
  $r_{max} = r_{max} + 1$
  $k_s = k$
  $Id(i_1) = r$
$else$
  Copy $Id(i_1)$ corresponding to $\hat{\mathbf{x}}_k^{i_1}$ from $\aleph_{k-1}$ to $\aleph_k$. $\aleph_k$ is a set of observed trajectories.

8) Delete trajectories: The trajectory which existed at $\aleph_{k-1}$ but not observed at time $t_k$ is deleted.

### B. *Noise-Estimate Particle PHD filter*

We introduce a new noise-estimate particle PHD filter which combines a noise-estimate particle filter and PHD filter. In this filter, a particle holds state $\mathbf{s}_k^i = \{\mathbf{x}_k^i, w_k^i, j_k^i, \mathbf{q}_k^i, \mathbf{r}_k^i\}$ at time $t_k$. $j_k^i$ is a label indicating the trajectory which the particle is assigned to. $\mathbf{q}_k^i$ and $\mathbf{r}_k^i$ are the system noise and the observation noise, respectively.

In each particle, the position and the velocity of a target are estimated through a same procedure as the noise-estimate particle filter. At the same time, the correlation to the trajectories is also estimated in a framework of the PHD filter. The flowchart of the proposed noise-estimate particle filter and PHD filter is shown in Fig. 1.

At time $t_k (k > 0)$,

1) Estimation: Particles are updated according to Eqs. (4) and (5) by the noise-estimate particle filter. At the same time, new particles are produced according to Eqs.(17) $\sim$ (19) by PHD filter.
2) Smoothing: Smoothing process in Kalman filter is applied in each particle according to (7) $\sim$ (9).
3) Likelihood calculation and resampling: The weight calculation and resampling are performed according to Eqs.(20) $\sim$ (22)
4) Update: Random noise is added to the system and observation noises in each particle as the noise-estimate particle filter.
5) Clustering, labeling and correlation calculation: Each procedure in PHD filter is applied.
6) Repeat 1) $\sim$ 5).

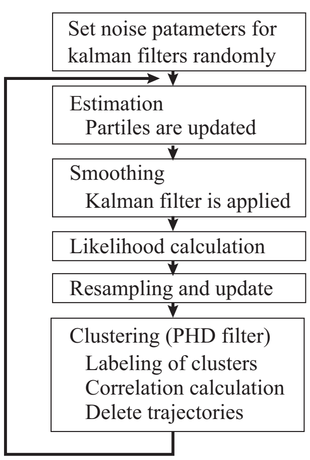



Fig. 1. Flowchart of Noise-Estimate Particle PHD filter

## IV. Computer Simulation

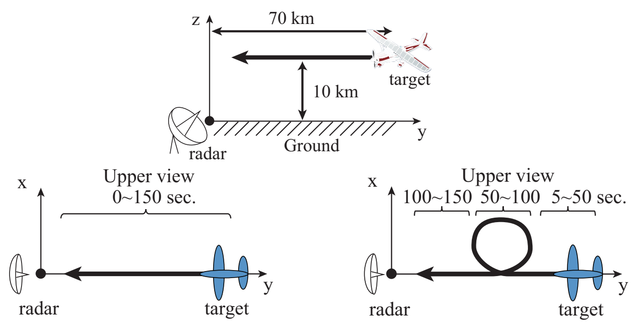
We conduct several computer simulations to compare the performance of the proposed noise-estimate particle PHD filter (NP-PHDF), noise-estimate particle filter (NPF)[9] Kalman filter, and conventional particle filters optimized for straight and curved paths.

### A. *Performance of single target tracking using noise-estimate particle filter (NPF)*

We assume a radar sensor is placed at the origin of the coordinate and set the sampling interval as 0.5 [s], the accuracy of the radar as 30 [m] in distance and 0.2 [deg.] in azimuth and elevation angles.

In this simulation, the target starts to move from 70 [km] apart from the radar toward the radar with a height of 10[km] and the velocity of 306 [m/s]. Two types of trajectories including straight and curved paths are tested as shown in Fig.2.

For a scenario of the straight path, the target moves by the uniform linear motion between 0 to 150 [sec.]. For a scenario of the curved path, the target moves by the uniform linear motion between 0 to 50 [sec.] and 100 to 150 [sec.], and with the turning motion from 50 to 100 [sec.] during level flight. In this scenario, the target turns 1 or 4 times on horizontal plane.




(a) Scenario of straight path (b) Scenario of curved path

Fig. 2. Scenario for computer simulatoin

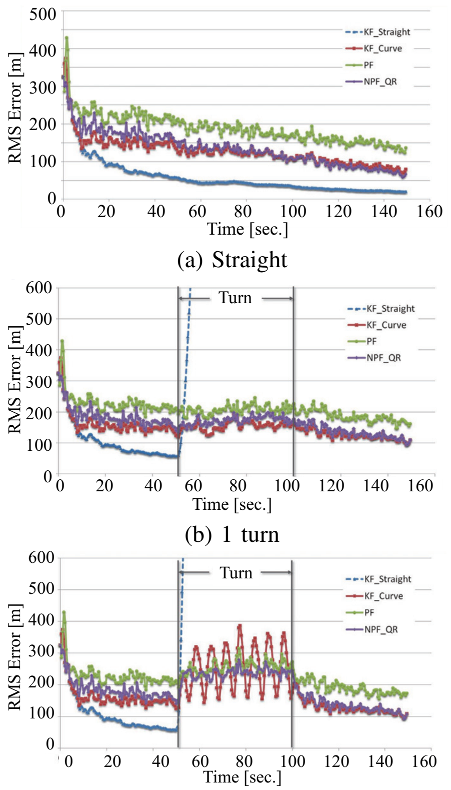
We run Monte-Carlo simulation in 50 times and evaluates the RMS (Root Mean Square) error. Four types of tracking filters are compared

1) Kalman filters which are adjusted to provide the best performance at straight path
2) Kalman filters which are adjusted to provide the best performance at curved path
3) Conventional particle filter
4) Noise-estimated particle filter (NPF)

The number of particles are 200 for conventional particle filter and 100 for noise-estimated particle filter, which are determined experimentally.

Tracking results for the proposed noise-estimated particle filter which estimates system and observation noises are shown in Fig.3. In Fig.3, KF_Straight is the Kalman filter for straight trajectory, KF_Curve is the Kalman filter for curved trajectory, PF is the conventional particle filter, and NPF_QR is the proposed noise-estimated particle filter. In addition, the estimated system and observation noises for 4-turns trajectory are shown in Fig.4.

The smoothing performance of the proposed noise-estimated particle filter (NPF_QR) outperforms the conventional particle filter (PF), and is similar to the Kalman filter optimized for curved trajectory (KF_Curve). Though the Kalman filter optimized for straight trajectory (KF_Straight) shows the best performance in straight path, it cannot track the target while turning at all. In addition, though the tracking delay is occurred for the Kalman filter for curved trajectory (KF_Curve) at 4-turns scenario, the accuracy of the proposed




(a) Straight

(b) 1 turn

(c) 4 turns

Fig. 3. Tracking error (system and observation noises are estimated)

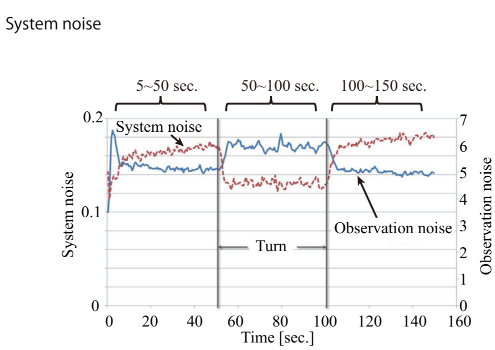



Fig. 4. Estimated system and observation noises (4 turns)

filter is similar to the conventional particle filter (PF). Consequently, it is confirmed that the proposed filter can be applied to various scenarios more adaptively than other conventional filters.

*B. Performance of multiple target tracking using noise-estimate particle PHD filter (NP-PHDF)*

In this simulation, multiple targets are tracked simultaneously using the proposed noise-estimate particle PHD filter.

We evaluate the performance by the success rate of multiple target tracking and RMS error between the actual and estimated trajectries through Monte-Carlo simulation in 50 times. To calculate the success rate, we define the tracking is suceeded if the same target is tracked successively until the end.

In this simulation, two targets $A$ and $B$ are tracked simultanously. The target $A$ starts at the velocity of 306 [m/s] and makes turns at the period from 50 [s] to 100 [s]. The target $B$ appears at 25 [s] after the target $A$ starts the motion and makes turns at the period from 50 [s] to 100 [s]. Both targets are crossed at the position of 50 [km] apart from the radar with the different of height of 300 [m], 600 [m], and 900 [m] as shown in Fig.5.

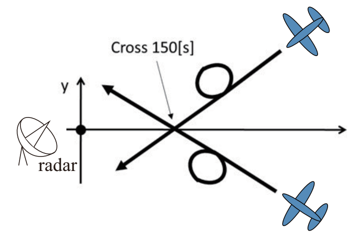


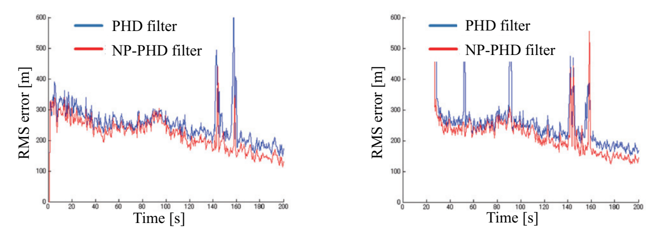

Fig. 5. Trajectries of targets

Table I shows the success rates of the conventional PHD filter and the noise-estimate particle PHD filter (NP-PHDF). We can see that the proposed noise-estimate particle PHD filter outperforms the conventional PHD filter in all senarios.

TABLE I. SUCCESS RATE

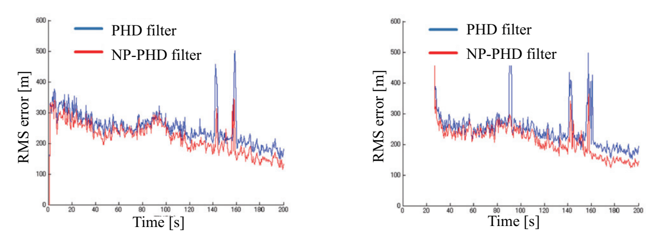
| Diff. height | PHD filter [%] | NP-PHDF [%] |
|---|---|---|
| 300 [m] | 54 | 78 |
| 600 [m] | 96 | 98 |
| 900 [m] | 100 | 100 |

Figs.6 $\sim$ 8 show the RMS errors in cases that the differences of height are 900[m], 600[m], and 300[m], respectively.




(a) object A (b) object B

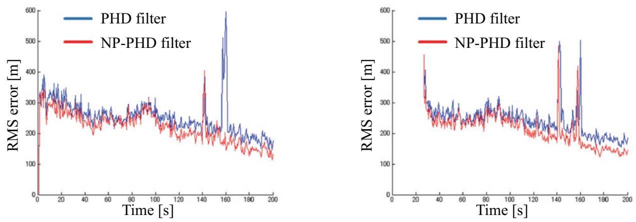
Fig. 6. RMS errors in case that difference of height is 900[m]




(a) object A (b) object B

Fig. 7. RMS errors in case that difference of height is 600[m]

The RMS error of the proposed noise-estimate particle PHD filter is almost same as the conventional PHD filter in a curved trajectry, and smaller than the PHD filter in a straight trajectory. From this simulation, it is verified that the




(a) object A (b) object B

Fig. 8. RMS errors in case that difference of height is 300[m]

proposed noise-estimate particle PHD filter has higher tracking performance than the conventional PHD filter.

## V. Conclusion

This paper proposed the new noise-estimated PHD particle filter for a multiple target tracking system. The proposed filter estimates the system and the observation noises in Kalman filter by using the particle filter and adapts to the abrupt changes of the target motion characteristics. In addition, multiple targets are tracked simultaneously in a framework of PHD filter.

We examined the tracking performance of the proposed noise-estimated PHD particle filter through computer simulations for the radar tracking system, and confirmed that the proposed filter has high smoothing performance for the tracking error and high stability performance for sudden changes of the target motions.

This paper focused on the radar-based target tracking, however, the applications of the proposed noise-estimated particle filter is not limited to radar tracking and we can apply the proposed filter for a variety of vision-based tracking systems. We are going to apply the proposed filter for, for example, pedestrian tracking using distributed cameras and laser range finders in near future.

## Acknowledgment

The present study was supported in part by a Grant-in-Aid for Scientific Research (A) (26249029)

## References

[1] Y. Kosuge, "Tracking performance improvement of $\alpha-\beta$ filter in terms of steady state velocity," in *WSANE2009*, 2009, pp. 107–112.

[2] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[3] J. Vermaak, A. Doucet, and P. Perez, "Maintaining multi-modality through mixture tracking," in *Proc. the International Conference on Computer Vision*, 2003, pp. 1110–1116.

[4] C. Yang, R. Duraiswami, and L. Davis, "Fast multiple object tracking via a hierarchical particle filter," in *In: International Conference on Computer Vision*, 2005, pp. 212–219.

[5] R. Kurazume, H. Yamada, K. Murakami, Y. Iwashita, and T. Hasegawa, "Target tracking using sir and mcmc particle filters by multiple cameras and laser range finders," in *Proc. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS08)*, 2008, pp. 3838–3844.

[6] J. Cui, H. Zha, H. Zhao, and R. Shibasaki, "Multi-modal tracking of people using laser scanners and video camera," *Image and Vision Computing*, vol. 26, no. 2, pp. 240–252, 2008.

[7] B. Lau, K. O. Arras, and W. Burgard, "Tracking groups of people with a multi-model hypothesis tracker," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3180–3185.

[8] J. H. Kotecha and P. M. Djuric, "Gaussian particle filtering," *IEEE Transaction on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, 2003.

[9] M. Ishibashi, Y. Iwashita, and R. Kurazume, "Adaptive target tracking by noise-estimated particle filter," in *20th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV2014)*, 2014.

[10] B. Vo, S. Singh, and A. Doucet, "Sequential monte carlo implementation of the phd filter for multi-target tracking," in *Proc. of Fusion 2003*, 2003, pp. 792–799.

[11] K. Panta, B. Vo, and S. Singh, "Improved probability hypothesis density (phd) filter for multitarget tracking," in *Proc. of Third International Conference on Intelligent Sensing and Information Processing (ICISIP 2005)*, 2005, pp. 213–218.

[12] P. Li and T. Zhang, "Visual contour tracking based on particle filters," *Image and Vision Computing*, vol. 21, pp. 111–123, 2002.

[13] Y. Satoh, T. Okatani, and K. Deguchi, "A color-based tracking by kalman particle filter," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3, 2004, pp. 502–505 Vol.3.

[14] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *2003 IEEE/RSJ International Conference on Intelligent Robots and System (IROS 2003)*, vol. 1. IEEE, 2003, pp. 206–211.

[15] T. Schon, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, 2005.