An Improved Resampling Approach for Particle Filters in Tracking

Abdullahi Daniyan, Yu Gong, and Sangarapillai Lambotharan
Signal Processing and Networks Research Group
Wolfson School of Mechanical, Electrical and Manufacturing Engineering
Loughborough University, U.K.
Email: {a.daniyan, y.gong, s.lambotharan}@lboro.ac.uk

***Abstract*—Resampling is an essential step in particle filtering (PF) methods in order to avoid degeneracy. Systematic resampling is one of a number of resampling techniques commonly used due to some of its desirable properties such as ease of implementation and low computational complexity. However, it has a tendency of resampling very low weight particles especially when a large number of resampled particles are required which may affect state estimation. In this paper, we propose an improved version of the systematic resampling technique which addresses this problem and demonstrate performance improvement.**

***Index Terms*—Resampling, particle filters, sequential Monte Carlo (SMC) methods, systematic resampling.**

I. Introduction

Sequential Monte Carlo (SMC) or particle filtering (PF) methods have proven useful within the past couple of decades in handling target tracking, especially when the observations are represented by nonlinear state-space models with non-Gaussian noise models. The PF method have been applied in a diverse range of disciplines including control, wireless communications, surveillance, defence, space applications, oceanography, finance, autonomous vehicles, robotics, remote sensing, computer vision and biomedical research, see for example [1]–[13].

When PF methods are used, for example in tracking applications, the goal is to track and estimate various distributions that emerge in the dynamic state-space models [14]. To this end, randomly generated samples (*particles*) are used to explore the states of the space. The generated samples along with associated weights are then used to approximate the distributions of interest [14].

Most PF methods generally involve a process for generating and propagating particles, weight assignment and computation and resampling of particles [15]. The resampling process replaces certain set of particles with another (usually depending on the particle weights) [15]. The resampling process is crucial in PF methods in order to avoid a situation where a few set of particles dominate other particles with their weights; a process known as degeneracy [3], [15], [16]. Having a degenerate set of particles is undesired as this will cause large variances in the obtained state estimates.

Several methods of resampling in PF methods have been proposed in the literature. These include residual resampling [17], [18], multinomial resampling [19], stratified resampling [20], systematic resampling (SR) [15], [20], [21], branching corrections [22], resampling with rejection control [23]. For more details on surveys and review of resampling methods, the reader is refered to [14], [15], [24]–[26]. The resampling methods listed are sequential algorithms and the most common are the multinomial, residual, systematic and startified resampling techniques [14]. In these algorithms, resampling is performed from the approximating distributions utilizing the latest weights [14]. Among the most common algorithms listed, the SR method is often more desired due to its ease of implementation, less computational complexity and less random number generation [14]. The stratified resampling has the same order of complexity as the SR (in the order of number of resampled particles required N) but requires N number of random number generation during implementation while the SR requires only one random number generation.

In PF methods, very low weight particles are less likely to contribute to the estimates of an approximating distribution. Therefore, allowing very low weights to contribute-during resampling-to the approximating distribution estimates could add to estimation variance leading to poor state estimates especially when the required number of resampled particles is large. The SR method, despite its desirable properties, has a tendency to resample very low weight particles especially when the required number of resampled particles is large. This phenomenon was hinted in [27].

In this paper, we describe the reason for the occurrence of this phenomenon in the SR algorithm and propose a way to address the problem. We named this as improved systematic resampling (ISR) method. In subsequent sections, we present how ISR addresses the phenomenon affecting the SR and demonstrate the improvement offered by our method.

The rest of the paper is organized as follows. Section II gives a brief background on initial stages preceding the resampling stage. In Section III we present the systematic resampling algorithm. In section IV, the proposed improvement of the systematic resampling technique is presented. Simulation results and discussion are presented in Section V. Finally, conclusions are drawn in Section VI.

II. Problem Formulation

We mentioned earlier that PF methods usually involve particle propagation, weight computation followed by resampling. In this section, we give a brief overview of the PF stages preceding the resampling stage in order to set a scene for presenting our algorithm.

Consider a tracking context (either single or multiple target tracking) where at time k we have a single target state and measurement model respectively given by:

$$x_k = f(x_{k-1}, n_k), \tag{1}$$

$$z_k = g(x_k, v_k), \tag{2}$$

where $f(\cdot)$ and $g(\cdot)$ are nonlinear functions; $x_k \in \mathbb{X}$ is the state of the model and $z_k \in \mathbb{Z}$ is the observation with state and observation space $\mathbb{X}$ and $\mathbb{Z}$ respectively; n_k and v_k are independent and identically distributed white noises. Assume that an alternate representation of the state, (1) is the probability distribution, $p(x_k|x_{k-1})$ and that of the observation, (2) is the distribution, $p(z_k|x_k)$. We aim to sequentially estimate the filtering distribution $p(x_k|z_{1:k})$ in a recursive manner by computing

$$p(x_k|z_{1:k}) \propto \int p(z_k|x_k)p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1}. \tag{3}$$

Generally, the above cannot be solved analytically therefore approximations (such as PF methods) are required.

In PF, the distribution $p(x_{k-1}|z_{1:k-1})$ is approximated by a set of particles with assigned weights $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^M$ where M is the number of particles, such that:

$$p(x_{k-1}|z_{1:k-1}) \approx \sum\nolimits_{i=1}^{M} w_{k-1}^i \delta(x_{k-1} - x_{k-1}^i), \tag{4}$$

where $\delta(\cdot)$ is the Dirac delta operator. The weights are normalized such that they all sum up to one. This approximation makes it possible to solve (3), so that

$$p(x_k|z_{1:k}) \propto p(z_k|x_k) \sum_{i=1}^{M} w_{k-1}^i p(x_k|x_{k-1}^i). \tag{5}$$

The expression of (5) shows how the approximating distribution of $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^M$ can be obtained. This is the particle propagation stage as the particle x_{k-1}^i is propagated in time to give x_k^i through importance sampling (see e.g. [3] for more details).

As for weight computation, we draw equally weighted particles from $p(x_k|z_{1:k})$. However, since this is not possible in most cases, we resort to sampling from an alternate distribution called the *proposal/importance* distribution, $q(x_k)$ [3], [16], [28]. An example of such distribution is $p(x_k|x_{k-1})$ [16], [28]. Since $q(x_k)$ is different from $p(x_k|z_{1:k})$, the particles drawn from $q(x_k)$ need to be weighted in order to have a correct inference [16], [28]. This can be acheived by recursively computing

$$w_k^i \propto \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i)} w_{k-1}^i. \tag{6}$$

Eq. (6) is usually followed by a normalization to ensure all weights sum to one. A recursive progression of this expression can lead to degeneracy. This is an undesired situation where one or few particles have large weights and others have negligible weights. This in turn causes an increase in weight variances as observations are processed. This will lead to a very poor approximation of the filtering distribution $p(x_k|z_{1:k})$ [14]. This is why the resampling stage is needed in PF methods. Following particle propagation and weight computation for the filtering distribution, we now have the approximating distribution at time k given by $\{x_k^i, w_k^i\}_{i=1}^M$.

III. Systematic Resampling

In this section, the SR algorithm is described. We also explain why the phenomenon described earlier exists.

SR and sequential resampling methods in general require particles from an approximating distribution with associated weights. These particles are resampled (usually depending on the weights) to give an estimate of the approximate distribution.

Assume we have an approximating distribution represented by a set of particles and associated weights $\{x_k^i, w_k^i\}_{i=1}^M$. We aim to resample N particles from these set of particles such that the outcome is the set $\{x_k^j, w_k^j\}_{j=1}^N$. N can be greater than the number of propagated particles M but for most applications, it is kept constant, i.e., $N = M$ [14]. The SR method achieves this in what is described next.

The SR method [15], [20], [21] first computes the cumulative sum of the weights

$$\begin{aligned} Q^1 &= w_k^1, \\ Q^i &= Q^{i-1} + w_k^i, \quad i = 2, \cdots, M. \end{aligned} \tag{7}$$

The whole particle set is divided into subpartitions called *strata*. The first *strata* is a random number, U^1 generated from the uniform distribution $\mathcal{U}[0, \frac{1}{N}]$. The rest are updated by $U^n = U^{n-1} + \frac{1}{N}$ for $n = 1, \cdots, N$. SR compares the cumulative sum Q^i with the updated uniform number U^n. A possible implementation of the SR method is shown in Algorithm 1. The number of times the i-th particle is replicated (resampled) depends on how many times the updated uniform number U^n falls within the range of $(Q^{i-1}, Q^i]$. For a very low weight, its contribution to the cumulative sum Q will be small and if the required number of resampled particles N is small, the increment to the uniform number update $\frac{1}{N}$ will be large.

```
Data: $(\{x_k^i, w_k^i\}_{i=1}^M, N)$
Result: $\{x_k^j, w_k^j\}_{j=1}^N$
Normalize weight;
Generate random number $U \sim \mathcal{U}[0, \frac{1}{N}]$;
Compute cumulative sum of weights $Q$;
for $i = 1 : M$ do
    $t = 0$;
    while $Q^i > U$ do
        $t = t + 1$;
        $U = U + \frac{1}{N}$;
    end
    $N(i) = t$
end
```

Algorithm 1: A sample SR algorithm

Hence the probability of the i-th particle being resampled is very low if the i-th particle is very small. Similarly, if the weight of the i-th particle is very small but N is large, the increment term $\frac{1}{N}$ will be very small. Hence the probability of the uniform number U^n falling within the range $(Q^{i-1}, Q^i]$ increases. This presents a high tendency of the very low i-th particle being resampled. Allowing very low weight particles to contribute to the approximating distribution estimates could add to estimation variance and this could lead to poor state estimates.

IV. The Improved Systematic Resampling

In the previous section, we introduced the SR method. We also described a phenomenon that causes it to yield poor state estimates of the approximating distribution particularly when large number of resampled particles is required. In this section, we present the proposed improvement to the SR algorithm.

Given that in PF methods, a particle having very low weight is less likely to contribute (improvement wise) to the estimate of an approximating distribution; we then propose that, for a very low weight w_k^i, we want to be able to reduce the possibility of the updated uniform number U^n falling within the range $(Q^{i-1}, Q^i]$ given the increment term $\frac{1}{N}$ for a large N.

To this end, we perform a sort of *weight-relowering* technique where we identify very low weights $\tilde{w}_k \subset w_k$ and reassign them a much lower value, ρ such that $0 < \rho \ll 1$. This is so that for a very low weight $\tilde{w}_k^i$, its contribution to the cumulative sum of weights Q of (7) will be very small. A weight is classed as being very low if the condition

$$w_k^i < \tau, \quad \forall i \tag{8}$$

is satisfied, where $i = 1, \cdots, M$. The threshold τ is chosen such that $\Pr(w_k^i > \tau) = 99\%$. A possible implementation of the ISR method is shown in Algorithm 2.

```
Data: $(\{x_k^i, w_k^i\}_{i=1}^M, N, \rho, \tau)$
Result: $\{x_k^j, w_k^j\}_{j=1}^N$
Check that (8) is satisfied in order to identify $\tilde{w}_k$;
Apply weight-relowering to $\tilde{w}_k$ by assigning then the
value $\rho$;
Normalize weight;
Generate random number $U \sim \mathcal{U}[0, \frac{1}{N}]$;
Compute cumulative sum of weights $Q$;
for $i = 1 : M$ do
    $t = 0$;
    while $Q^i > U$ do
        $t = t + 1$;
        $U = U + \frac{1}{N}$;
    end
    $N(i) = t$
end
```

Algorithm 2: A sample ISR algorithm

We further illustrate our proposed method and contrast it with the SR method as shown in Fig. 1 for the same set of weights as input. The SR and ISR method are depicted in Fig. 1a and Fig. 1b respectively. From Fig. 1a, the weight marked with '$*$' is the very low weight. So depending on number of resampled particles required, N, the increment term, $\frac{1}{N}$ can cause the updated uniform number U^n to fall in the range $(Q^1, Q^2]$. This become even more likely especially when N is large. In Fig. 1b, after the *weight-relowering* technique is applied, we see that the height of the very low weight indicated by '$*$' has reduced, hence the updated uniform number U^n, given the increment term, $\frac{1}{N}$ is much less likely to fall in the range $(Q^1, Q^2]$ even for large N.

V. Simulation Results

In this section, we demonstrate the performance of the proposed ISR method against the SR resampling method. We consider a 2-D multiple target tracking (MTT) scenario where a total of four targets are tracked using a nonlinear observation model. The targets were observed for 100 discrete time steps i.e., $k = 1, \cdots, 100$. The true trajectories of the targets are shown in Fig. 3. The start and end positions are indicated by a triangle and a square respectively. The dynamics of the of the targets is described using a nonlinear model with nearly constant turn rate given by:

$$\mathbf{x}_k = \begin{bmatrix} 1 & 0 & \frac{\sin\omega_k\Delta t}{\omega_k} & -\frac{1-\cos\omega_k\Delta t}{\omega_k} \\ 0 & 1 & \frac{1-\cos\omega_k\Delta t}{\omega_k} & \frac{\sin\omega_k\Delta t}{\omega_k} \\ 0 & 0 & \cos\omega_k\Delta t & -\sin\omega_k\Delta t \\ 0 & 0 & \sin\omega_k\Delta t & \cos\omega_k\Delta t \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \mathbf{n}_k \tag{9}$$

where $\omega_k = \omega_{k-1} + \Delta t u_{k-1}$, Δt denotes the sample period which is assumed to be $1s$. The target state vector $\boldsymbol{x}_k = [\mathbf{x}_k, \omega_k]^{\bar{T}}$ comprises of planar positions and velocities are given as the last two elements of $\mathbf{x}_k = [x_k, y_k, \dot{x}_x, \dot{y}_k]^{\bar{T}}$ along with turn rate ω_k. The variables (x_k, y_k) represent the position of the target; $\mathbf{n}_k = \mathcal{N}(\cdot, 0, \sigma_n^2 I)$ and $u_{k-1} = \mathcal{N}(\cdot, 0, \sigma_u^2 I)$ with $\sigma_n = 10$ m/s^2 and $\sigma_u = \pi/180$ rad/s.

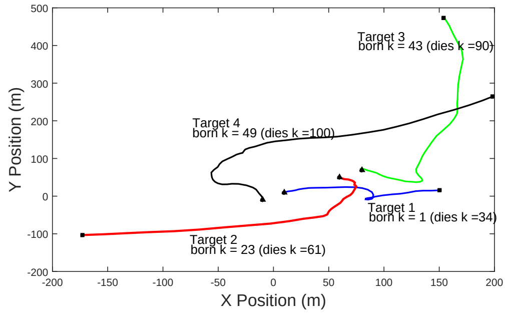
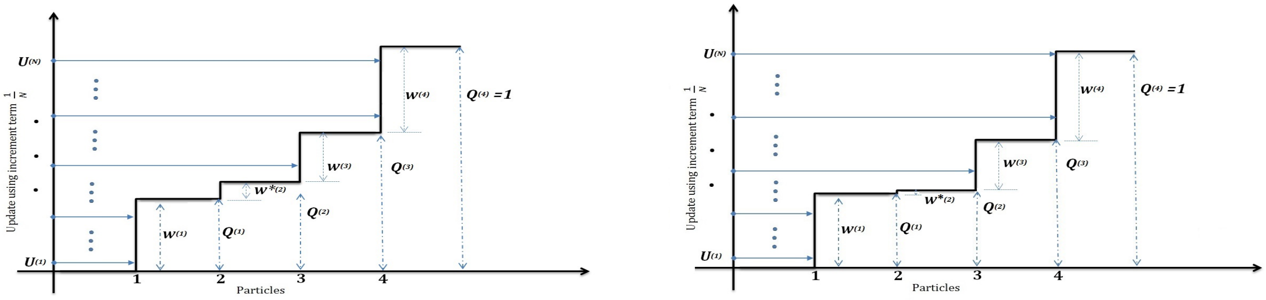


Figure 3: Ground truth for target trajectories of four tracks superimposed on the xy plane over 100 time steps

We assume a sensor $[x_s, y_s]^{\bar{T}}$ located at the origin of the x-y Cartesian coordinate which generates noisy range-bearing measurements of the targets with false alarms. The target-originated measurements are given by the nonlinear model

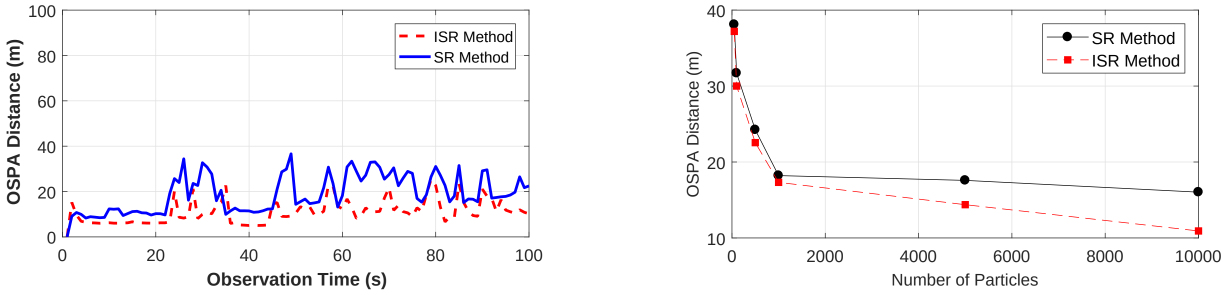
$$\boldsymbol{z}_k = \begin{bmatrix} r_k \\ \theta_k \end{bmatrix} + \mathbf{v}_k \tag{10}$$



(a) SR

(b) ISR

Figure 1: An illustration of the SR and ISR method. There are four particles with respective weights w. The very low weight is indicated as $w^{*(2)}$.



(a)

(b)

Figure 2: (a) Performance in terms of OSPA distance against observation time for $N = M$. (b) Performance in terms of OSPA measure against increasing number of particles for a constant $M = 1000$ and varying N from 50 to 10,000.

with

$$r_k = \left\| \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}_k - \begin{bmatrix} x_s \\ y_s \end{bmatrix} \right\|, \tag{11}$$

and

$$\theta_k = \arctan\left(\frac{[0\ 1\ 0\ 0]\mathbf{x}_k + y_s}{[1\ 0\ 0\ 0]\mathbf{x}_k + x_s} \right) \tag{12}$$

where the measurement noise, $\mathbf{v}_k$ is a zero-mean Gaussian white noise vector with covariance matrix $\mathbf{R}$ =diag($[\sigma_r^2, \sigma_\theta^2]$) with $\sigma_r = 9m$ and $\sigma_\theta = 0.45$ rad.

We use the particle probability hypothesis density filter [29], [30] to perform the multi-target tracking. So that at each time k we can obtain the multiple target approximating distributions and we apply both the SR and ISR to perform resampling. After resampling, we perform state estimation and compute the estimation error then move to the next time $k + 1$.

We use the optimal subpattern assignment (OSPA) [31] to evaluate the error in the estimation of the approximating distribution for using the ISR and the SR methods. A high OSPA measure translates to high estimation error while a lower OSPA measure means lower estimation error, hence higher accuracy. We also observe the OSPA for each of the methods for various number of resampled particles required N.

Fig. 2a shows OSPA measures versus observation time. The results were averaged over 100 Monte Carlo (MC) runs of the tracking filter. The number of particles used for the tracking filter is 10,000 particles per existing target. Both ISR and SR methods were implemented concurrently during each MC run. This means that the same approximating distributions were fed into to both resampling methods at each time k before state estimates were extracted and OSPA computed. From Fig. 2a, it is seen that the ISR method outperforms the SR method by giving a lower OSPA measure compared to the SR throughout the observation time. We attribute the poor state estimation for when the SR method is used to very low weights being resampled. Furthermore, with the *weight-relowering* technique, the state estimation accuracy is improved for when the ISR resampling method was used.

Fig. 2b shows OSPA measures versus number of resampled particles required. The results were averaged over 100 MC runs of the tracking filter. The result shows that both techniques give similar level of performance for when the number of resampled particles required, N is around 1000 or less. But for larger values of N, the proposed method performs better than the SR method by having a lower OSPA measure when compared to the SR method. The result further confirms our hypothesis that when the SR method is used, the state estimation error is likely to increase especially for large values of N.

VI. Conclusion

In this paper, we described a phenomenon in the systematic resampling method which causes a high tendency of very low weights to be replicated during the resampling stage of a PF method. We proposed an improvement of the systematic resampling method in order to solve this problem. Results demonstrate that the proposed resampling method outperformes the standard systematic resampling method particularly when large number of resampled particles are required.

Acknowledgement

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/K014307/1, the MOD University Defence Research Collaboration (UDRC) in Signal Processing, UK and the Petroleum Technology Development Fund (PTDF), Nigeria.

References

[1] A. Doucet and A. M. Johansen, "A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later," *Handbook of nonlinear filtering*, vol. 12, no. 656-704, p. 3, 2009.

[2] A. Doucet, S. Godsill, and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[3] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[4] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle Filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, Sep 2003.

[5] O. Cappé, S. J. Godsill, and E. Moulines, "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[6] Y. Bar-Shalom, P. K. Willett, and X. Tian, "Tracking and Data Fusion," *A Handbook of Algorithms.*, 2011.

[7] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[8] S. Challa, *Fundamentals of Object Tracking*. Cambridge University Press, 2011.

[9] W. Koch, "Tracking and Sensor Data Fusion," *English, Mathematical Engineering. Springer Berlin Heidelberg*, pp. 157–185, 2014.

[10] R. P. Mahler, *Statistical Multisource-multitarget Information Fusion*. Artech House, Inc., 2007.

[11] ——, *Advances in Statistical Multisource-multitarget Information Fusion*. Artech House, 2014.

[12] M. Mallick, V. Krishnamurthy, and B.-N. Vo, *Integrated tracking, classification, and sensor management: theory and applications*. John Wiley & Sons, 2012.

[13] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell, *Bayesian multiple Target Tracking*. Artech House, 2013.

[14] T. Li, M. Bolic, and P. M. Djuric, "Resampling Methods for Particle Filtering: Classification, Implementation, and Strategies," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 70–86, May 2015.

[15] A. Doucet, A. Smith, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, ser. Information Science and Statistics. Springer New York, 2001.

[16] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[17] J. S. Liu and R. Chen, "Sequential Monte Carlo Methods for Dynamic Systems," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[18] E. R. Beadle and P. M. Djuric, "A Fast-weighted Bayesian Bootstrap Filter for Nonlinear Model State Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 1, pp. 338–343, 1997.

[19] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *IEE Proceedings F - Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, April 1993.

[20] G. Kitagawa, "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[21] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved Particle Filter for Nonlinear Problems," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 146, no. 1, pp. 2–7, 1999.

[22] D. Crisan, P. Del Moral, and T. Lyons, *Discrete Filtering Using Branching and Interacting Particle Systems*. Université de Toulouse. Laboratoire de Statistique et Probabilités [LSP], 1998.

[23] J. S. Liu, R. Chen, and W. H. Wong, "Rejection Control and Sequential Importance Sampling," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1022–1031, 1998.

[24] M. Bolić, P. M. Djurić, and S. Hong, "Resampling algorithms for particle filters: A computational complexity perspective," *EURASIP Journal on Applied Signal Processing*, vol. 2004, pp. 2267–2277, 2004.

[25] R. Douc and O. Cappé, "Comparison of Resampling Schemes for Particle Filtering," in *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, ISPA*. IEEE, 2005, pp. 64–69.

[26] J. D. Hol, T. B. Schon, and F. Gustafsson, "On Resampling Algorithms for Particle Filters," in *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*. IEEE, 2006, pp. 79–82.

[27] A. Daniyan, Y. Gong, P. Feng, J. Chambers, and S. Lambotharan, "Kalman-Gain Aided Particle PHD Filter for Multi-target Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[28] D. B. Rubin, "The Calculation of Posterior Distributions by Data Augmentation: Comment: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR Algorithm," *Journal of the American Statistical Association*, vol. 82, no. 398, pp. 543–546, 1987.

[29] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking," in *Information Fusion, 2003. Proceedings of the Sixth International Conference of*, vol. 2, July 2003, pp. 792–799.

[30] ——, "Sequential Monte Carlo Methods for Multitarget Filtering with Random Finite Sets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1224–1245, Oct 2005.

[31] D. Schuhmacher, B. T. Vo, and B. N. Vo, "A Consistent Metric for Performance Evaluation of Multi-Object Filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, Aug 2008.