A Likelihood-Free Particle Filter for Multi-Object Tracking

Fabian Sigges and Marcus Baum
Institute of Computer Science
University of Goettingen, Germany
Email: {fabian.sigges, marcus.baum}@cs.uni-goettingen.de

Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
Email: uwe.hanebeck@ieee.org

***Abstract*—We present a particle filter for multi-object tracking that is based on the ideas of the Approximate Bayesian Computation (ABC) paradigm. The main idea is to avoid the explicit computation of the likelihood function by means of simulation. For this purpose, a large amount of particles in the state space is simulated from the prior, transformed into measurement space, and then compared to the real measurement by using an appropriate distance function, i.e., the OSPA distance. By selecting the closest simulated measurements and their corresponding particles in state space, the posterior distribution is approximated. The algorithm is evaluated in a multi-object scenario with and without clutter and is compared to a global nearest neighbour Kalman filter.**

I. Introduction

Multi-object tracking (MOT) has been an ongoing research area since the 1950's. Typical MOT applications can be found in robotics, remote sensing, and surveillance. In general, the range of objects to track, the different sensors, and application areas are almost endless. Since the requirements are usually different for every scenario, there exists a huge amount of different MOT algorithms. A recent overview, including the joint probabilistic data association filter (JPDAF) and random finite set (RFS) approaches, can be found in [1].

An important class of algorithms for object tracking are particle filters [2]–[5]. Particle filters use a set of weighted particles to represent the posterior distribution. Usually, updating the weights from one time step to the next involves computation of the likelihood function. For MOT, this is a significant computational bottleneck, as the likelihood function considers all possible measurement-to-object associations. Already for a small number of objects, evaluation of the likelihood function takes a vast amount of time.

The problem of an intractable likelihood function is not only restricted to MOT. In biology, psychology, and economics, where Bayes' theorem is often used for parameter inference and model selection, the likelihood function often does not have an analytical expression or is computationally intractable. To overcome this issue, the likelihood-free class of Approximate Bayesian Computation (ABC) algorithms was invented [6]–[10]. In [11], an ABC rejection method was proposed for binary sensors, which is, however, a completely different likelihood from the classical MOT.

The contribution of this work are the following: Based on the ABC ideas, we develop a particle filter for MOT that neither requires the evaluation of the likelihood function nor the computation of particle weights. For this purpose, we employ the following two main aspects: (1) In the measurement space, a set distance is used (OSPA distance) in order to compare measurements. (2) High rejection rates are avoided by means of a k-nearest neighbour strategy for the measurement update. To the best of our knowledge, ABC ideas have not yet been applied to the classic MOT problem.

In the remainder of this work, we first briefly present the development of the ABC paradigm and its various algorithms. Then, the MOT problem is specified and some challenges are highlighted. Subsequently, our ABC particle filter tailored for MOT is introduced and evaluated with a global nearest neighbour Kalman filter (GNN) [12] as a comparison. The computational complexity of our algorithm is slightly higher than the GNN. Intuitively, for each particle, a GNN association is calculated.

II. Approximate Bayesian Computation

In this section, we give a short introduction to ABC and track the development of the different ABC algorithms over the years.

For the most part, ABC algorithms were designed in various fields of biology. Here, one of the main goals is to estimate parameters for certain models given some data. One powerful approach to parameter estimation is the application of Bayes' theorem. For simple models, where the likelihood function can be derived analytically or efficiently evaluated, parameters can be directly inferred given the data. In many interesting cases though, the likelihood function cannot be derived or is computationally intractable. In these cases, Bayes' theorem cannot be applied directly.

Although the idea of utilising simulations in order to circumvent the likelihood function originated in the 80's, lack of computing power inhibited a widespread use. One of the first papers to pick up the idea and deliver an algorithm was by Pritchard et al. in 1999 [6]. They developed an algorithm, today known as ABC rejection, that contains the key idea included in all subsequent ABC algorithms. Assume, we want the posterior distribution of a parameter x given some data or measurement z. The ABC rejection algorithm works as follows:

1) Draw a candidate value from the prior distribution $x^* \sim p(x)$.
2) Simulate an observation given the candidate value $z^* \sim p(z|x^*)$.
3) Compare simulated and real observation and accept, if $d(z, z^*) \leq \varepsilon$.
4) If not accepted, discard x^* and repeat with 1).

Here, $d(\cdot, \cdot)$ is a suitable distance measure between the observations. Optimally, the information from the observations can be expressed using sufficient summary statistics to reduce the cost of the distance calculation. For a small ε, a large number of accepted x^* represents a good approximation of the correct posterior distribution $p(x|z)$. Although the method delivers independent samples of the wanted distribution, it suffers from the same problems as the standard accept-reject algorithm. If the prior and the posterior distribution differ substantially or a diffusive prior has to be assumed, the acceptance rate becomes very low, e.g. [6] reports acceptance rates between 10^{-3} to 10^{-6}.

In order to remedy this situation, Marjoram et al. (2003) integrated the idea of simulating and comparing observations into the well known framework of Markov Chain Monte Carlo (MCMC) [7]. Instead of using the likelihood to compute the acceptance probability, the drawn sample is used to simulate data, which is then compared to the real observation. If the distance between the observations is too large, the sample is discarded directly. The acceptance probability can be written as follows:

$$\alpha = \begin{cases} \frac{p(x^*)q(x_i|x^*)}{p(x_i)q(x^*|x_i)} & \text{if } d(z, z^*) \leq \varepsilon \\ 0 & \text{if } d(z, z^*) > \varepsilon \end{cases}$$

Here, $p(x)$ is the prior distribution and $q(\cdot|\cdot)$ is the transition kernel used to generate new candidates x^* from the current state x_i. Similar to the ABC rejection algorithm the MCMC ABC produces samples from the desired posterior distribution. For good convergence, special attention has to be paid to the prior distribution and the transition kernel. Since standard MCMC chains are already prone to get stuck in regions of low probability, the problem of getting stuck is even worse for the ABC variant due to the additional threshold for passing. But as long as the chain is in regions of high probability [8] reports acceptance rates of up to 5%.

In 2007, the notion of ABC was first included in Sequential Monte Carlo (SMC) samplers by Sisson et al. [8] and shortly after, multiple versions of ABC SMC samplers appeared [9], [10]. They all work in a mainly similar fashion. Consider a pool of N particles with associated weights, like in standard particle filters. Subsequently, particles are drawn according to their weight, perturbed with a chosen transition kernel, an observation is simulated for the drawn particle, and compared against the real observation. After N particles passed the inspection, the particles are weighted. The differences between the SMC methods lie in the weighting procedures. An overview of the weighting procedures can, for example, be found in [13].

By employing SMC methods, most of the issues of the previous methods can be circumvented. First, an SMC sampler cannot get stuck in regions of low probability. Second, a low acceptance rate can be avoided by using descending values $\varepsilon_0 > \varepsilon_1 > \cdots > \varepsilon_m$. While in the first step the largest ε and a diffuse prior is used, in the subsequent steps smaller ε are chosen. This yields a smooth transition from prior to posterior distribution.

III. Problem Formulation

This section is dedicated to a short introduction into multi-object tracking and highlights some major challenges associated with it. Additionally, we introduce some notation that will be used throughout the rest of the paper. Multi-object tracking can be formulated with a linear process model and a linear measurement equation

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1}, \qquad \boldsymbol{w}_{k-1} \sim \mathcal{N}(0, \boldsymbol{R}) \ , \quad (1)$$

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k, \qquad \boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{Q}) \ , \quad (2)$$

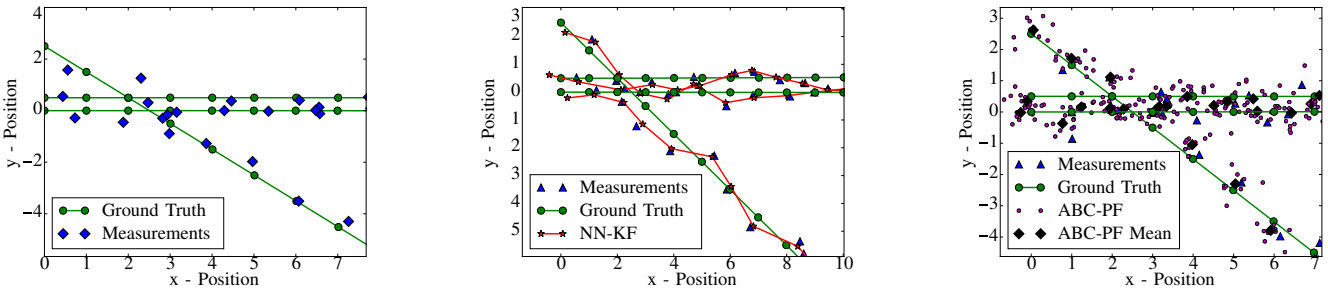
where $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ denote the process and measurement noise in a suitable dimension. For a number M of multiple objects, the state vector $\boldsymbol{x}_k = [x_{k,1}^T, \ldots, x_{k,M}^T]^T$ consists of the stacked individual states. Depending on the applied process model, the individual states might consist of the position and velocity. For the sake of simplicity, we assume that the detection probability is one, i.e., there are no missed detections. But there might be false detections, i.e., clutter measurements. All measurements can be stacked into an overall measurement vector $\boldsymbol{z}_k = [z_{k,1}^T, \ldots, z_{k,M_c}^T]^T$, where M_c is the number of measurements including clutter. In multi-object tracking, the measurements are unlabeled, i.e., can be regarded as a set of individual measurements without any specific order. Hence, the measurement equation becomes

$$\underbrace{\begin{bmatrix} z_{k,\pi(1)} \\ \vdots \\ z_{k,\pi(M_c)} \end{bmatrix}}_{=\boldsymbol{z}_k} = \underbrace{\begin{bmatrix} \boldsymbol{H}_1 & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{H}_M \\ & \boldsymbol{0}^{c\times M} & \end{bmatrix}}_{=\boldsymbol{H}} \cdot \underbrace{\begin{bmatrix} x_{k,1} \\ \vdots \\ x_{k,M} \end{bmatrix}}_{=\boldsymbol{x}_k} + \underbrace{\begin{bmatrix} v_{k,1} \\ \vdots \\ v_{k,M} \\ \tilde{v}_{k,1} \\ \vdots \\ \tilde{v}_{k,c} \end{bmatrix}}_{=\boldsymbol{v}_k} . \quad (3)$$

Here, $\pi \in \Pi_{M_c}$ denotes an *unknown* permutation, i.e., an element from the group of permutations of order M_c. Furthermore, c denotes the number of false detections that can vary between the timesteps. The $\tilde{v}_{k,i}$ $i = 1, \ldots, c$ are uniformly distributed over the area of computation. The corresponding likelihood for the measurement equation (3) incorporates the enumeration of all measurement-to-object associations. Simply put, this is not possible for a larger number of objects, since the number of possible associations for N objects is $M_c!/c!$.

IV. ABC Particle Filter for MOT

In this section, we want to discuss some ideas how we put the principles of ABC into the multi-object tracking



(a) Ground truth. (b) Kalman filter. (c) ABC filter for $N = 10$.

Fig. 1: Ground truth and example estimates of the Kalman filter and the ABC filter.

framework. We will only focus on the SMC sampler, since it provides the best results for parameter estimation [13] and is also frequently applied in tracking scenarios. Let us consider the same conditions as mentioned in the previous section. Similar to other particle filters, the state is represented by a number of particles N, each of which consists of the full state $\boldsymbol{x}_k^i = [(x_{k,1}^i)^T, \dots, (x_{k,M}^i)^T]^T$ for $i = 1, \dots, N$. Suppose we already have a population of particles at time step $k-1$, $\{\boldsymbol{x}_{k-1}^i\}_{i=1}^N$, and want to transition to time step k. In the fashion of ABC we draw large number $N_{\text{pred}} \gg N$ of proposal particles and predict their location in the next step by applying the process model

$$\widehat{\boldsymbol{x}}_k^p = \boldsymbol{A}\tilde{\boldsymbol{x}}_{k-1}^p + \boldsymbol{w}_{k-1}^p, \quad p = 1, \dots, N_{\text{pred}} \ . \tag{4}$$

The probability for a particle to be transitioned to the next step is equally likely, so that each particle will be father to approximately N_{pred}/N proposal particles. For each proposal particle, a simulated measurement is drawn according to the measurement equation. This step can also be regarded as a transformation from the state space into the measurement space

$$\widehat{\boldsymbol{z}}_k^p = \boldsymbol{H}\widehat{\boldsymbol{x}}_k^p + \boldsymbol{v}_k^p, \quad p = 1, \dots, N_{\text{pred}} \ . \tag{5}$$

Now, the simulated measurements have to be compared to the real measurements. In the case of unclear associations between object and measurement and additional false detections, the distance function has to be chosen carefully. The standard Euclidean distance used in ABC methods would produce wrong results as the associations do not fit.

We propose to use a distance metric on sets, i.e., the OSPA distance [14], which allows us to deal with different cardinalities (e.g., for clutter). As we consider the number of tracks to be known and fixed over the time span, the OSPA distance can be simplified. Let $\boldsymbol{z}_k$ be the vector of true measurements with the additional false detections, then the OSPA-like distance function looks as follows

$$\boldsymbol{d}(\widehat{\boldsymbol{z}}_k^p, \boldsymbol{z}_k) = \min_{\pi \in \Pi_{M_c}} \sum_{m=1}^{M} ||z_{k,m} - \widehat{z}_{k,\pi(m)}^p||_2 \ . \tag{6}$$

Here, M_c is the number of true measurements plus the number of false detections in that time step. In the definition of the OSPA distance, a generic norm is used to assign the optimal sub-pattern. Since our measurements consist of only the position, we chose the Euclidean distance as a norm. If other values are measured, such as angles, the choice of norm should be revisited. At this point, our algorithm differs from the usual ABC paradigm. Instead of choosing a threshold ε to discard particles that differ too much from the measurement, i.e., $\boldsymbol{d}(\widehat{\boldsymbol{z}}_k^p, \boldsymbol{z}_k) > \varepsilon$, we select the N particles with the smallest distance $\boldsymbol{d}(\widehat{\boldsymbol{z}}_k^p, \boldsymbol{z}_k)$. This is mainly to ensure that the algorithm finishes in a limited, predictable time span. If a borderline situation occurs, where the measurement happens to be far away from the predictions waiting for particles that actually pass the threshold might take an unreasonable amount of time. A similar procedure for the ABC rejection sampler, together with some analytical results, has been presented by Biau et. al [15].

With that in mind, let us have a quick look at the computational complexity. The main computational burden is obviously hidden in (6), but instead of having to search through all $M_c!$ possible permutations, the Hungarian algorithm solves this optimization problem in $\mathcal{O}(M_c^3)$ [16]. Since this computation has to be done for every proposal particle we arrive at a complexity of $\mathcal{O}(N_{\text{pred}} \cdot M_c^3)$ for one time step.

The algorithm for one time step can be found in Algorithm 1.

V. Evaluation

The algorithm is evaluated in a scenario with three moving objects. Two objects move in parallel from left to right, while a third object crosses their trajectories. An example of the trajectories can be found in Fig. 1a. As a process model a nearly constant velocity model (NCV) is used, a common choice for slowly maneuvering objects. In one dimension the model looks as follows

$$\begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta T^2 \\ \Delta T \end{bmatrix} w_{k-1} \ , \tag{7}$$

where ΔT is the difference between two time steps. Since the measurements consist of only the position, the measurement equation only extracts the current position from the state vector, i.e., $\boldsymbol{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$.

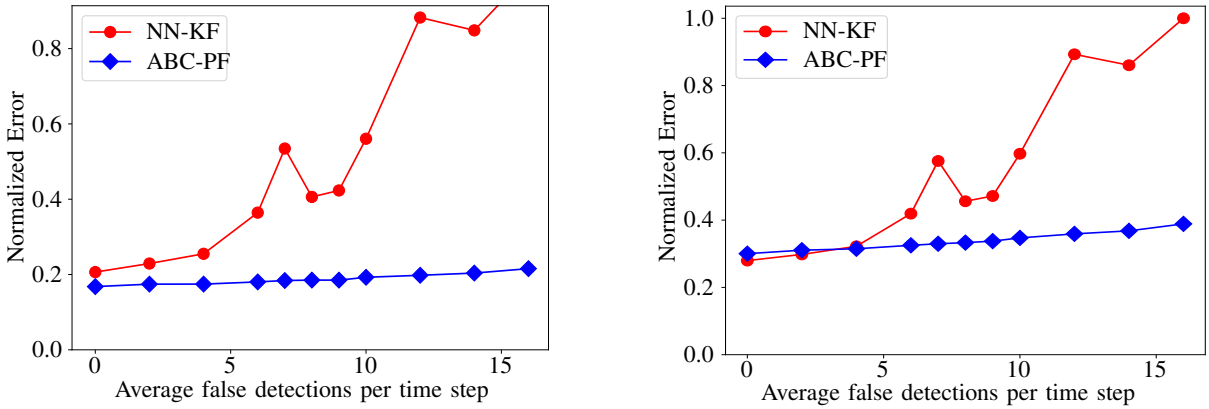


Fig. 2: Errors for different values of λ, averaged over the whole trajectory and 100 runs. Left: OSPA of means, see (8). Right: MMOSPA between ground truth and particles, see (9).

Algorithm 1 ABC Particle Filter

Input: N particles $\{\boldsymbol{x}_{k-1}^i\}_{i=1}^N$ that represent the state at time step $k-1$ and measurements $\boldsymbol{z}_k$ for step k

Output: N particles $\{\boldsymbol{x}_k^i\}_{i=1}^N$ that represent the state at time step k

1: **for** $p = 1$ to N_{pred} **do**
2: Draw $\tilde{\boldsymbol{x}}_{k-1}^p$ uniformly from $\{\boldsymbol{x}_{k-1}^i\}_{i=1}^N$
3: Predict the next state for the particle: $\widehat{\boldsymbol{x}}_k^p = \boldsymbol{A}\tilde{\boldsymbol{x}}_{k-1}^p + \boldsymbol{w}_{k-1}^p$
4: Simulate the measurement: $\widehat{\boldsymbol{z}}_k^p = \boldsymbol{H}\widehat{\boldsymbol{x}}_k^p + \boldsymbol{v}_k^p$
5: Calculate the distance to the real measurement: $\boldsymbol{d}(\widehat{\boldsymbol{z}}_k^p, \boldsymbol{z}_k) = \min_{\pi\in\Pi_{M_c}} \sum_{m=1}^M ||z_{k,m} - \widehat{z}_{k,\pi(m)}^p||_2$
6: **end for**
7: Choose N particles $\{\boldsymbol{x}_k^i\}_{i=1}^N \subset \{\widehat{\boldsymbol{x}}_k^p\}_{p=1}^{N_{\text{pred}}}$ with lowest $\boldsymbol{d}(\widehat{\boldsymbol{z}}_k^p, \boldsymbol{z}_k)$ as new state for time step k

The ABC filter is compared to the global nearest neighbour Kalman filter (GNN). GNN keeps a Kalman filter for each object and uses the measurement-to-object association that minimises the global distance, similar to (6), but compares the prediction to the measurement instead of a simulated measurement. As the GNN uses a Kalman filter for prediction and update, the solution is optimal as long as the associations are correct. In order to create a more difficult and realistic scenario, each time step a different number, provided by a draw from a Poisson distribution with mean λ, of clutter points is simulated. The clutter is uniformly distributed over the whole area the objects are tracked in. Both filters are assumed to know the correct starting point and velocity, but we initialised the ABC filter as a point cloud with the correct vector as the mean. The covariance matrices $\boldsymbol{R}$ and $\boldsymbol{Q}$ from the dynamic model are both chosen to be diagonal with 0.3 as only values. One of the initial vectors for $k = 0$ is $x_0 = \begin{bmatrix}0, & 1, & 2.5, & -1\end{bmatrix}^T$. The other initial vectors have similar starting velocities.

The evaluation in terms of errors is not obvious, since both algorithms suffer from labeling uncertainties [17]–[20]. Especially for the two parallel tracks, labels are switching regularly. To account for this, we used two different distance measures that calculate the global minimum distance. The first one calculates OSPA distance between the mean $\bar{x}$ of the particles and the ground truth x^*, see (8)

$$E_{\text{point}}(k) = \left(\frac{1}{M}\min_{\pi\in\Pi_M}\sum_{m=1}^{M}||x_{k,m}^* - \bar{x}_{k,\pi(m)}||_2^2\right)^{\frac{1}{2}}. \quad (8)$$

Due to the aforementioned labeling errors, a simple calculation of the particle mean yields bad results. To remedy the situation, the objects states for each particle are relabeled to provide the Minimum Mean OSPA estimate.

The second measure (9) is the Mean OSPA distance [21] between the ground truth and the particles

$$E_{\text{cov}}(k) = \left(\frac{1}{M}\frac{1}{N}\sum_{n=1}^{N}\min_{\pi\in\Pi_M}\sum_{m=1}^{M}||x_{k,m}^* - x_{k,\pi(m)}^n||_2^2\right)^{\frac{1}{2}}. \quad (9)$$

As the GNN works with Gaussians, we sample a sufficiently large number of points from a normal distribution with the mean and covariance from the GNN, which are then used as particles in the MOSPA calculation. Thereby, the spread of the particles is compared to the estimated covariance of the GNN.

In Fig. 2 the results of some simulations can be seen. The figures depict the average normalized error for different values of λ. The simulations are carried out with $N = 50$ and $N_{\text{pred}} = 5000$. For a low number of false detections the two filters produce similar results in both the points estimates as well as the particle-wise error calculation. As soon as the number of false detections rises above 5, the errors for the GNN increase considerably. On the contrary, the ABC particle filter is able to almost keep a constant error rate even in highly cluttered environments. This is mainly due to the implicit resampling in the prediction step. Particles that are chosen because of a

false detection, will very likely not produce proposal particles close to a new measurement and then be discarded.

VI. Conclusion and Future Work

In this paper, we gave a short introduction into the algorithm class of Approximate Bayesian Computation and how the paradigm developed over the last years. We adapted the ABC particle filters in such a way that estimation of dynamic processes is also possible. While the framework is very general and allows for estimation in various fields, we focused on multi-object tracking. Therefore, we frequently used a simplified OSPA distance and the Hungarian algorithm to solve the arising linear assignment problem.

For future work, there are many open areas where the algorithm can be further improved and tested. First of all, since the algorithm requires only forward computation of the process model and measurement equation, it is optimally suited for applications with highly nonlinear models. Depending on the scenario, the only part that might have to be changed is the distance function.

Furthermore, a substantial difference between the ABC particle filter and usual particle filters is that we currently do not make use of any weighting of the particles. Literature suggest various weighting schemes for ABC particle filters. It is planned to investigate these and adapt them to the area of multi-object tracking.

In [15] and [22], some analytical results regarding the consistency of ABC, summary statistics, and the ratio of accepted particles are shown. Future work should be based on these results with a discussion on appropriate summary statistics and different distance measures for sets, such as the kernel distance [23]–[25].

In order to have full target tracking capabilities, we intend to add missed detections by employing the full OSPA distance. For the case of an unkown and changing number of targets our algorithm does not offer a simple incorporation, and hence, a seperate birth-death process would be necessary.

VII. Acknowledgements

This work was supported by the Simulation Science Center Clausthal / Göttingen (SWZ) and the German Research Foundation (DFG) under grand BA 5160/1-1.

References

[1] B.-N. Vo, M. Mallick, Y. Bar-Shalom, S. Coraluppi, R. Osborne, R. Mahler, B.-t. Vo, and J. G. Webster, *Multitarget Tracking*. John Wiley & Sons, Inc., 2015.

[2] O. Cappe, 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, May 2007.

[3] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle Filters for Positioning, Navigation, and Tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, Feb 2002.

[4] C. Hue, J.-P. Le Cadre, and P. Perez, "Tracking Multiple Objects with Particle Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 791–812, 2002.

[5] J. Vermaak, S. Godsill, and P. Perez, "Monte Carlo Filtering for Multi Target Tracking and Data Association," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 1, pp. 309–332, Jan. 2005.

[6] J. K. Pritchard, M. T. Seielstad, A. Perez-Lezaun, and M. W. Feldman, "Population growth of human Y chromosomes: a study of Y chromosome microsatellites," *Molecular Biology and Evolution*, vol. 16, no. 12, 1999.

[7] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavar, "Markov Chain Monte Carlo without likelihoods," *Proceedings of the National Academy of Sciences*, vol. 100, no. 26, pp. 15 324–15 328, 2003.

[8] S. A. Sisson, Y. Fan, and M. M. Tanaka, "Sequential Monte Carlo without likelihoods," *Proceedings of the National Academy of Sciences*, vol. 104, no. 6, pp. 1760–1765, 2007.

[9] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. Stumpf, "Approximate Bayesian Computation scheme for parameter inference and model selection in dynamical systems," *Journal of The Royal Society Interface*, vol. 6, no. 31, pp. 187–202, 2009.

[10] M. A. Beaumont, J.-M. Cornuet, J.-M. Marin, and C. P. Robert, "Adaptive Approximate Bayesian Computation," *Biometrika*, vol. 96, no. 4, pp. 983–990, 2009.

[11] A. Ickowicz, "Approximate Bayesian algorithms for multiple target tracking with binary sensors," *CoRR*, vol. abs/1410.4262, 2014. [Online]. Available: http://arxiv.org/abs/1410.4262

[12] S. S. Blackman and R. F. Popoli, *Design and Analysis of Modern Tracking Systems*. Arthech House, 1999.

[13] B. M. Turner and T. Van Zandt, "A Tutorial on Approximate Bayesian Computation," *Journal of Mathematical Psychology*, vol. 56, no. 2, pp. 69–85, April 2012.

[14] 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.

[15] G. Biau, F. Crou, and A. Guyader, "New insights into Approximate Bayesian Computation," *Annales de l'Institut Henri Poincar, Probabilits et Statistiques*, vol. 51, no. 1, pp. 376–403, 2015.

[16] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Dover Publications, 1998.

[17] C. M. Leon, H. Driessen, and P. K. Mandal, "Efficient Characterization of Labeling Uncertainty in Closely-Spaced Targets Tracking," in *Proceedings of the 19th International Conference on Information Fusion (Fusion 2016)*, 2016, pp. 449–456.

[18] R. Georgescu, P. Willett, L. Svensson, and M. Morelande, "Two Linear Complexity Particle Filters Capable of Maintaining Target Label Probabilities for Targets in Close Proximity," in *15th International Conference on Information Fusion (FUSION)*, 2012, pp. 2370–2377.

[19] Y. Boers, E. Sviestins, and H. Driessen, "Mixed Labelling in Multitarget Particle Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 2, pp. 792–802, 2010.

[20] E. H. Aoki, P. Mandal, L. Svensson, Y. Boers, and A. Bagchi, "Labeling uncertainty in multitarget tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 3, pp. 1006–1020, June 2016.

[21] M. Guerriero, L. Svensson, D. Svensson, and P. Willett, "Shooting Two Birds with Two Bullets: How to Find Minimum Mean OSPA Estimates," *Proceedings of the 13th International Conference on Information Fusion (Fusion 2010)*, 2010.

[22] D. T. Frazier, G. M. Martin, C. P. Robert, and J. Rousseau, "Asymptotic Properties of Approximate Bayesian Computation," *ArXiv e-prints*, Jul. 2016.

[23] M. Baum and U. D. Hanebeck, "The Kernel-SME Filter for Multiple Target Tracking," in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

[24] J. M. Phillips and S. Venkatasubramanian, "A Gentle Introduction to the Kernel Distance," *CoRR*, vol. abs/1103.1625, 2011.

[25] U. D. Hanebeck and V. Klumpp, "Localized Cumulative Distributions and a Multivariate Generalization of the Cramér-von Mises Distance," in *Proceedings of the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)*, Seoul, Republic of Korea, August 2008, pp. 33–39.

[26] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*. YBS Publishing, 2011.

[27] M. G. B. Blum, "Approximate Bayesian Computation: A Nonparametric Perspective," *Journal of the American Statistical Association*, vol. 105, no. 491, pp. 1178–1187, 2010.