Defence R&D Canada
External Literature
DRDC-RDDC-2016-P004
December 2015

Particle Weight Approximation with Clustering for Gossip-Based Distributed Particle Filters

Chon Wang Chao, Michael Rabbat
Electrical and Computer Engineering, McGill University
Montreal, QC, Canada
chonwang.chao@mail.mcgill.ca, michael.rabbat@mcgill.ca

Stephane Blouin
DRDC Atlantic Research Center
Dartmouth, NS, Canada
stephane.blouin@drdc-rddc.gc.ca

***Abstract*—Distributed particle filters are appealing for cooperative tracking in distributed systems. However, a non-negligible amount of communication overhead can be required to synchronize the particle weights between agents. This paper proposes a particle weight approximation method, based on clustering and a smoothness assumption on the particle cloud distribution, to reduce the communication overhead. The proposed algorithm is evaluated on both simulated data and data from an at-sea trial involving bearings-only tracking. The results demonstrate that the proposed approach achieves state-of-the-art accuracy, especially in cases with a limited communication budget.**

I. Introduction

Particle filters are very effective for tracking with non-linear, non-Gaussian dynamic and/or measurement models. In a distributed tracking system, some or all of the nodes run their own filters and gather measurements, and they collaborate to improve the overall tracking accuracy by exchanging messages.

Since exchanging messages involves communication, a major challenge of designing such a system is to achieve high tracking accuracy while reducing the communication overhead. Much previous work has focused on parametric approaches where a parametric model is fit to the particle cloud. Gu et al. [1] propose to model the particle cloud with a Gaussian distribution. Sheng et al. [2] and Gu [3] adopt a Gaussian Mixture Model to handle multi-modal particle cloud distributions. In these frameworks, instead of communicating about particle weights directly, the associated model parameters are exchanged instead. However, these approaches may lead to significant errors when the particle cloud is not well-represented by a Gaussian or mixture of Gaussians.

Hlinka et al. [4] propose to gossip on sufficient statistics of the joint likelihood function. If the likelihood can be expressed as a combination of a few basis functions (e.g., the first few terms of the Fourier expansion, or a low-degree polynomial), then this approach is effective. Farahmand et al. [5] suggest to adapt the proposal distribution by first running max- and min-consensus methods to determine a region of the state space containing a subset of particles with the highest joint likelihood. This approach is effective when the joint likelihood is peaky and uni-modal, in which case the bulk of the density is concentrated on the particles in a small, localized region. However, when the joint likelihood function does not have a single, concentrated peak, then this approach involves communicating about all or most of the particles.

In this paper, by assuming that the log-likelihood is sufficiently smooth over the state space, the particles are grouped into a set of disjoint clusters. The log-likelihoods of the clusters are then exchanged and the particle weights are reconstructed locally and independently at each node. Our experiments show that the proposed approach achieves competitive tracking accuracy while having a communication overhead that is significantly lower than other approaches.

II. Problem Formulation

A. Target Dynamics and Measurement Model

This paper considers the problem of tracking a state vector $\mathbf{x}_t \in \mathbb{R}^{n_x}$ over discrete time steps $t \geq 0$. The evolution of the state is assumed to follow the discrete-time dynamics:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \xi_t, \tag{1}$$

where $f : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ is a (possibly non-linear) mapping, and $\xi_t \in \mathbb{R}^{n_x}$ is time-varying process noise.

A collection of n_v sensors each gather noisy measurement of the target state. The set of sensors is denoted by $\mathcal{V} = \{v_1, ..., v_{n_v}\}$ and the measurement vector $\mathbf{z}_{t,v} \in \mathbb{R}^{n_{z,v}}$ collected by sensor $v \in \mathcal{V}$ at time t is modeled as:

$$\mathbf{z}_{t,v} = h_{t,v}(\mathbf{x}_t) + \zeta_{t,v}, \tag{2}$$

where $h_{t,v} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_{z,v}}$ is a node-dependent, possibly non-linear mapping, and $\zeta_{t,v}$ is the measurement noise at sensor v at time t.

It is assumed that the dynamic model f and the distribution of the process noise ξ_t are known to all nodes $v \in \mathcal{V}$. Moreover, each node v knows its own measurement function $h_{t,v}$ and measurement noise distribution $\zeta_{t,v}$.

B. Network Model

In the distributed setting, to improve the tracking accuracy, neighboring nodes should communicate to share information about their observations. However, not all pairs of nodes may communicate directly due to reasons such as distance and shadowing effects. The connectivity between nodes where message exchange is possible is modeled using a graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ which is assumed to be connected and undirected. The vertices $\mathcal{V}$ represent the nodes, and there is an edge

$(v_i, v_j) \in \mathcal{E}$ between a pair of vertices if communication is possible between the corresponding nodes. To simplify the presentation in this conference paper, this graph is further assumed to be static over time, although this is not essential.

III. Background

A. Particle Filters

A particle filter performs a Monte-Carlo simulation-based approximation of the Bayes-optimal recursion. The posterior density $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ is approximated using a set of N weighted particles, $\mathcal{X}_t = \{\hat{\mathbf{x}}_t^{(i)}, w_t^{(i)}\}_{i=1}^N$, and the posterior approximation is given by:

$$\hat{p}(\mathbf{x}_t|\mathbf{z}_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta(\mathbf{x}_t - \hat{\mathbf{x}}_t^{(i)}) \ , \tag{3}$$

where δ is the delta function, each particle $\hat{\mathbf{x}}_t^{(i)}$ is a hypothesis of the current state value, and $w_t^{(i)}$ is the importance weight of particle i and an approximate value of the posterior density at $\hat{\mathbf{x}}_t^{(i)}$.

Given a prior distribution $p_0(x)$ on the initial target location, the particle cloud is initialized by sampling N locations from p_0 and setting the associated weights to $w_0^{(i)} = 1/N$ for all $i = 1, \ldots, N$. At each subsequent time step t when a new measurement is made, the particles are first propagated (prediction step) according to a proposal distribution $q(\mathbf{x}_{t|t-1}^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{z}_{1:t})$. Then (data update step) the weights are updated to:

$$w_{t|t-1}^{(i)} \propto w_{t-1}^{(i)} \frac{p(\mathbf{z}_t|\mathbf{x}_{t|t-1}^{(i)}) p(\mathbf{x}_{t|t-1}^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_{t|t-1}^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{z}_{1:t})} \ , \tag{4}$$

where the constant of proportionality is such that the resulting weights $w_{t|t-1}^{(i)}$ sum to 1.

This forms a set of intermediate particle and weights, $\mathcal{X}_{t|t-1} = \{\mathbf{x}_{t|t-1}^{(i)}, w_{t|t-1}^{(i)}\}_{i=1}^N$. A new set of particles $\mathcal{X}_t$ is usually generated by resampling with replacement from $\mathcal{X}_{t|t-1}$, where $\mathbf{x}_{t|t-1}^{(i)}$ is sampled with probability $w_{t|t-1}^{(i)}$, and then all weights are set to $w_{t|t}^{(i)} = 1/N$. Hence, the new particles will be generated based on the more likely hypotheses (i.e., particles with higher weights). If resampling is not required, then $\mathcal{X}_t = \mathcal{X}_{t|t-1}$.

We focus here on the particular case of the *bootstrap particle filter* (BPF), where the proposal density is taken to be the same as the density $p(x_{t|t-1}^{(i)}|x_{t-1}^{(i)})$ defined by the dynamic model presented in Eq. 1. The proposal density is then independent of the measurement and the weight update simplifies to:

$$w_{t|t-1}^{(i)} \propto w_{t-1}^{(i)} p(\mathbf{z}_t|\mathbf{x}_{t|t-1}^{(i)}) \ . \tag{5}$$

B. Gossip-based Distributed Particle Filters

Assume there is a common source of randomness (e.g., all nodes initialize their pseudo-random number generators using the same seed). The particle cloud $\mathcal{X}_t$ and $\mathcal{X}_{t|t-1}$ for the nodes can be synchronized at time t so that the particles are identical at all nodes. To achieve a common network-wide estimate of the posterior at every node, the joint likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$ of the measurements $\mathbf{z}_t = \{z_{t,1}, ..., z_{t,n_v}\}$ or an approximation thereof, needs to be computed by all nodes. Under the standard assumption that the measurements at each node are conditionally independent given the state, the joint log-likelihood factorizes as the sum of the local log-likelihoods, and so it can be computed using gossip algorithms [6]. Specifically, each node initializes a gossip algorithm with value $\mathbf{y}_{t,v} = n_v \ln(p(\mathbf{z}_{t,v}|\hat{\mathbf{x}}_t))$, and then the gossip algorithm for distributed averaging converges asymptotically to

$$\bar{\gamma}_t = \frac{1}{n_v} \sum_{v \in \mathcal{V}} \mathbf{y}_{t,v} = \sum_{v \in \mathcal{V}} \ln(p(\mathbf{z}_{t,v}|\hat{\mathbf{x}}_t)) = \ln(p(\mathbf{z}_t|\hat{\mathbf{x}}_t)) \ . \tag{6}$$

Since convergence is only asymptotic, in practice a finite number of averaging gossip iterations are executed, and then a max-consensus algorithm [7] is run to ensure all nodes use exactly the same value for the approximate joint log-likelihood. The weight of particle i is then calculated as

$$w_{t|t-1}^{(i)} = w_{t-1}^{(i)} \frac{\exp(\bar{\gamma}_t^{(i)})}{\sum_{j=1}^{N} w_{t-1}^{(j)} \exp(\bar{\gamma}_t^{(j)})} \ . \tag{7}$$

IV. Particle Clustering Approximation

To achieve reasonable estimation accuracy, a few thousand particles are usually needed. Communicating individually to compute the joint log-likelihood of each individual particle brings considerable communication overhead. In this section, a particle weight approximation method based on clustering is proposed to reduce the communication cost.

A. Particle Clustering

Recall that, after performing the prediction step, the predicted particle locations $\mathbf{x}_{t|t-1}^{(i)}$ are identical at all nodes. To reduce communication overhead while adapting to the current particle distribution, we cluster the predicted particles based on their locations. In this paper, we use the k-means algorithm for clustering (typically $k \ll N$) with the Euclidean distance in the state space as the distance measure. Let C be a $k \times N$ cluster assignment matrix, with $C_{i,j} = 1$ if particle j is in cluster i, and $C_{i,j} = 0$ otherwise. Rather than gossip on the N-dimensional vector $\mathbf{y}_{t,v} = n_v \ln(p(\mathbf{z}_{t,v}|\hat{\mathbf{x}}_t))$ of particle log-likelihoods, the nodes gossip on the k-dimensional vectors $\mathbf{y}_{t,v}^c = C\mathbf{y}_{t,v}$, where the value associated with a cluster is the sum of the log-likelihoods of particles in that cluster.

After gossiping, all nodes have reached agreement on a k-dimensional vector of approximate joint log-likelihoods $\tilde{\gamma}_t^c$ associated with each cluster. To project these values back to weights for each of the N predicted particle locations, one could simply spread the cluster weight evenly over all particles. However, we have found that this typically works poorly since it creates sharp changes in the likelihood at cluster boundaries, whereas log-likelihood functions usually vary in a smooth manner over the state space. Instead, we propose a graph-based smoothing technique described next.

Defence R&D Canada
External Literature
DRDC-RDDC-2016-P004
December 2015

B. Particle Graph and Laplacian-Regularized Reconstruction

To interpolate the cluster weights $\tilde{\gamma}_t^c$ over all particles, we construct a κ-nearest neighbor graph $\mathcal{G}_p = \{\mathcal{V}_p, \mathcal{E}_p\}$ over the particles, where there is one vertex $v_{pi} \in \mathcal{V}_p$ for each particle $\mathbf{x}_t^{(i)}$, and there is an edge between v_{pi} and v_{pj} if either one of the two particles is one of the κ nearest neighbors of the other; the Euclidean distance in the state space is used as the distance measure. Let A denote the (unweighted) adjacency matrix associated with this graph, let D denote the diagonal degree matrix, with $D_{i,i} = \sum_{j=1}^{N} A_{i,j}$, and let $L = D - A$ denote the particle graph Laplacian.

To reconstruct the particle likelihoods, we seek log-likelihood values for each particle which are smooth with respect to particle proximity, as encoded in $\mathcal{G}_p$, and which also respect the sum-weight with respect to the joint log-likelihoods of each cluster, as computed via gossip. The particle likelihood reconstruction with smoothness constraint is formulated as an equality constrained quadratic program

$$\begin{aligned} \underset{\tilde{\gamma}_t}{\text{minimize}} \quad & \tilde{\gamma}_t^T L \tilde{\gamma}_t \\ \text{subject to} \quad & C\tilde{\gamma}_t = \tilde{\gamma}_t^c. \end{aligned} \tag{8}$$

Since L is always positive semi-definite, this problem is convex and can be solved using well-known methods.

V. Numerical Evaluation

This section compares the proposed approach with existing approaches in the literature on two bearings-only tracking scenarios, one using data from an at-sea trial and one simulated.

We compare the proposed distributed *bootstrap particle filter using clustering approximation* (BPFC) with two other methods: a naive distributed BPF that gossips on all particles, and the *set membership constrained* (SetM) distributed particle filter [5]. Recall that the SetM first computes a sub-region of the state space containing particles supposed to have the highest joint log-likelihood, and then it only gossips on particle weights within that sub-region. We also compare with a centralized BPF that has direct access to all measurements, as a benchmark for tracking accuracy. With sufficiently many gossip iterations, the distributed BPF should perform identically to the benchmark.

Evaluation metrics. To quantify tracking accuracy we report the *time-averaged root mean square position error* (ARMSE) and the percentage of lost tracks. Both scenarios involve targets spread roughly over a 10km-by-10km region. A target is declared as lost if the instantaneous position error exceeds 2 km. All the results reported are the average of 100 Monte Carlo trials and the ARMSE is calculated based on the trials not deemed lost. To quantify the communication overhead we report the number of scalar values transmitted per node per time step.

A. At-Sea Trial

This dataset is collected from an experiment conducted in October 2012 in the Emerald Basin area of the Scotian Shelf, off the coast of Nova Scotia, Canada. Five hydrophones were used to track an acoustic source. See [8] for more details on the experiment setup. The acoustic source is tracked over 178 time steps, each 1 minute in duration. The data is pre-processed to produce a sequence of noisy bearing measurements from each sensor, with a bearing error typically between ± 5 degrees of the true bearing to the target.

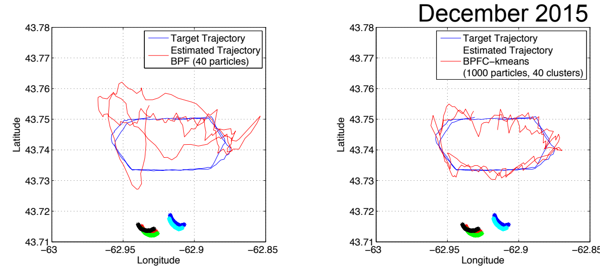

Fig. 1. Sample tracking performance of BPF (left) and BPFC (right) on target 1. The sensors are denoted by the colored trajectories at the bottom of the figures. With the same communication overhead, the clustering approach demonstrated better tracking performance.

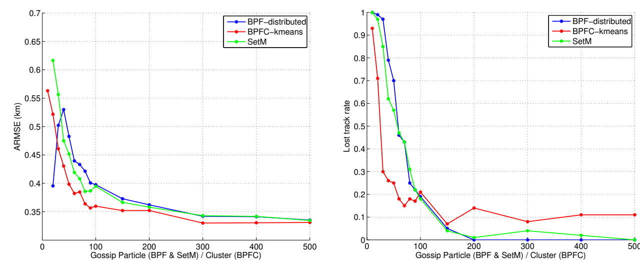

Fig. 2. Tracking performance of BPF, BPFC and SetM on the target with varying communication overhead. There are 1000 particles initially for the BPFC. Notice the low ARMSE of the distributed BPF with 10 particles are due to the high lost track rate.

We focus on a single target in this dataset. The target dynamics are assumed to follow a constant velocity plus clockwise coordinated turn model [9]. The parameters of the dynamic model are the angular acceleration a and the probability of not turning p, and these are fit in advance to be $a = -0.0009$ rad/s^2 and $p = 0.77$. The process noise is modeled as zero-mean Gaussian with a standard deviation of 2×10^{-3} degrees (roughly 222 m).

In this experiment, we focus on understanding the effect of the error introduced by the different particle approximation methods. The five sensors drift over time, and we assume that each sensor knows its location (e.g., via GPS). Since the five sensors are relatively close at all times (see Fig. 1), the communication topology is a complete graph, in which case gossip converges exactly after each node transmits once.

By assuming that the communication overhead to transmit the log likelihood of a particle and a cluster is the same, a performance comparison of the algorithms with varying communication overhead is shown in Fig. 2. The BPFC uses 1000 particles which are grouped to form k clusters, where k varies between 10 and 500. The performance of all of the algorithms degrades quickly when too few particles or clusters are used. However, with as few as 30 clusters, the lost track rate of BPFC drops to approximately 0.25 for BPFC, which is significantly less than the distributed BPF with 30 particles. Moreover, the mean ARMSE is also lower than that of the BPF and SetM. The performance of SetM and BPF are bad on this problem because the likelihood is generally spread over many of the particles.

Although the mean ARMSE is consistently lower for the BPFC approach, the associated lost track rate is higher if the number of clusters exceeds 150. This is likely due to the approximation error of clustering. However, for very low communication overhead (k = 30,...,100), the BPFC has a lower lost track rate and lower ARMSE.

The results can be explained by the advantage of having more particles initially. The performance of a particle filter is strongly tied to the number of particles. With too few particles, the posterior approximation is bad, and thus it degrades the tracking performance. However, with the clustering approach, although the same number of scalars is transmitted, each cluster's log-likelihood encapsulates the information of several particles. At the expense of some reconstruction error and computational overhead, the posterior is better approximated than the BPF with the same communication overhead.

B. Simulated Data

In the at-sea trial dataset, the effect of gossip is disregarded since there are only 5 sensors in close proximity. In the following simulated dataset, we introduce more sensors to study how the performance of the algorithm depends on the number of gossip iterations.

The simulated scenario consists of two targets and their dynamics are coupled. The first target is the leader which has the same dynamic model as mentioned above with $a = 0.0003$ rad/s^2 and $p = 0.5$. The second target is a follower and it moves towards the first target with a constant velocity of 1.8×10^{-3} rad/s at every time step. The process noise is modeled as a zero-mean Gaussian with standard deviation of 1×10^{-4} degrees, and the measurement noise is also modeled as zero-mean Gaussian with a standard deviation of 5 degrees. Sixteen sensors are arranged in a grid within the region. Since the dynamics of the two targets are coupled, the state is an 8-dimensional vector with 4 dimensions (position and velocity) for each target. The filters use $N = 2000$ particles.

Fig. 3 shows the performance of the algorithms while varying the number of gossip iterations. The errors reported are the sum of the two targets' position errors. If either or both of the two targets have an error greater than 2 km, the track is declared to be lost. When more than 100 clusters are used, the error is relatively insensitive to the number of clusters and all the algorithms achieve similar accuracy. The lost track rate is significant if less than 30 gossip iterations are used.

To compare the performance between the algorithms while keeping the communication overhead fixed, the case with 200 particles/clusters is studied and the results are shown in Fig. 4. Conclusions similar to the at-sea trail dataset can be drawn for this two-target case. SetM again delivers similar results as the distributed BPF. The clustering approach outperforms the others for all gossip iterations under the condition of same communication overhead, and in this setup the lost track rate of BPFC is also consistently lower than the other approaches.

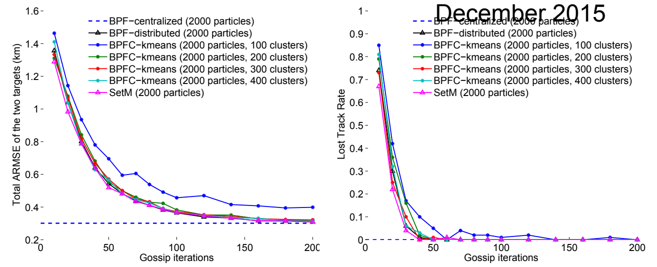

Fig. 3. Tracking performance of the centralized and distributed BPF, SetM and BPFC with different number of cluster versus gossip iterations.

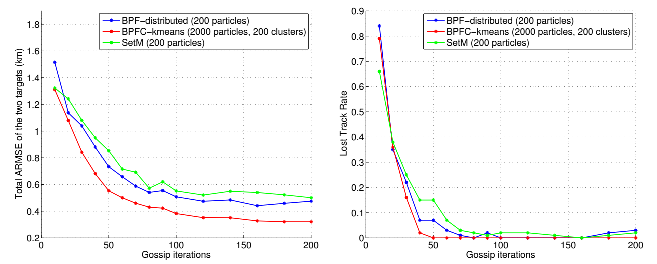

Fig. 4. Tracking performance of the distributed BPF, BPFC and SetM with the same number of particle/cluster (200) and varying gossip iterations.

VI. Conclusion

In this paper, an approximation method based on clustering is proposed. It outperforms the state-of-the-art with the same communication overhead. Moreover, the performance under very low communication overhead is considerably better; this nice property makes it particularly suitable for applications that have stringent communication budget.

References

[1] D. Gu, J. Sun, Z. Hu, and H. Li, "Consensus based distributed particle filter in sensor networks," in *International Conf. on Information and Automation*, June 2008, pp. 302–307.
[2] X. Sheng, Y.-H. Hu, and P. Ramanathan, "Distributed particle filter with gmm approximation for multiple targets localization and tracking in wireless sensor network," in *International Sym. on Information Processing in Sensor Networks*, April 2005, pp. 181–188.
[3] D. Gu, "Distributed particle filter for target tracking," in *IEEE International Conf. on Robotics and Automation*, April 2007, pp. 3856–3861.
[4] O. Hlinka, O. Slučiak, F. Hlawatsch, P. Djuric, and M. Rupp, "Likelihood consensus and its application to distributed particle filtering," *IEEE Trans. on Signal Processing*, vol. 60, no. 8, pp. 4334–4349, Aug 2012.
[5] S. Farahmand, S. Roumeliotis, and G. Giannakis, "Set-membership constrained particle filter: Distributed adaptation for sensor networks," *IEEE Trans. on Signal Processing*, vol. 59, no. 9, pp. 4122–4138, Sept 2011.
[6] A. Dimakis, S. Kar, J. Moura, M. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov 2010.
[7] F. Iutzeler, P. Ciblat, and J. Jakubowicz, "Analysis of max-consensus algorithms in wireless channels," *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 6103–6107, Nov 2012.
[8] J.-Y. Yu, D. Üstebay, S. Blouin, M. Rabbat, and M. Coates, "Distributed underwater acoustic source localization and tracking," in *Proc. Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2013.
[9] B. Ristic, S. Arulampalam, and N. Gordon, *"Beyond the Kalman filter: Particle filters for tracking applications"*. Artech House, 2004.