Relaxation of Distributed Data Aggregation for Underwater Acoustic Sensor Networks

Fiscal Year 2014-2015

Prepared By:
M. Rabbat
M. Coates
McGill University
Montreal, QC, Canada
Project Manager: Michael Rabbat, 514-398-1847
Contract Report # AMBUSH.2.2

PWGSC Contract Number: W7707-145675
Contract Scientific Authority: Stephane Blouin, Defence Scientist

Contract Report
DRDC-RDDC-2015-C208
March 2015

Table of contents

List of figures

This page intentionally left blank.

1 Relaxation of Distributed Data Aggregation for Underwater Acoustic Sensor Networks

1.1 Introduction

This aspect of the project is concerned with coordinating the sensors in an underwater acoustic network to collaboratively track an acoustic source or sources. Measurements are taken at each sensor node, and in order to obtain the best accuracy, the measurements should be jointly processed or fused. This requires communication and coordination among the nodes. At the same time, underwater communication is notoriously challenging. Channel conditions change rapidly and high-datarate communications are generally not possible. Consequently, protocols and mechanisms must be used that can adapt to the time-varying and unreliable communication medium.

Tracking strategies that rely on transmitting all measurements to a central location are infeasible in this setting, both because of the communication overhead and for robustness concerns. We therefore focus on decentralized methods in which the microprocessor system attached to each sensor execute the tracking algorithms. These local tracking algorithms communicate to share information between neighbouring sensors, distilling the raw measurements into essential summary statistics to reduce the communication overhead. In this project we specifically consider the use of gossip algorithms for processing information over the network because of their robustness to unreliable communication media, and we consider particle filters and random finite set methods for the local target tracking algorithms executed at each node because of their ability to scale and handle well scenarios with significant clutter and time-varying numbers of targets.

In the second year of the project we have focused on developing and analyzing distributed algorithms for single- and multi-target tracking. Our work has followed four main thrusts. 1) We have developed an analytical framework for quantifying the worst-case performance of gossip-based distributed particle filters. 2) We have devised a cardinalized probability hypothesis density filter for tracking multiple targets using measurements from multiple sensors. 3) We have developed a communication-efficient method for tracking a single target using bearings-only measurements in a spherical coordinate system. 4) We have developed a graph-based compression method to reduce the communication overhead for general gossip-based implementations of the distributed particle filter. The details of these contributions are summarized below.

1.2 Problem Formulation and Background

Our goal is to track the state of one or more noise sources using measurements gathered from a distributed network of sensors. We denote the state of the noise sources at time t by $\mathbf{x}_t$. Typically the state of a single source is a four-dimensional vector, with two dimensions for the source position and two dimensions for the source velocity. If there are three sources present in the monitored region then the state $\mathbf{x}_t$ would be 12-dimensional. The state evolves according to a dynamic model,

$$\mathbf{x}_{t+1} = g(\mathbf{x}_t) + \mathbf{v}_t \ , \tag{1}$$

where $g(\cdot)$ is a (possibly non-linear) mapping describing the evolution of the state from one time step to the next, and $\mathbf{v}_t$ is a random vector of process noise. This model is very general and can be used to capture a wide range of noise source dynamics. This sort of first-order model (where $\mathbf{x}_{t+1}$ only depends on $\mathbf{x}_t$) is standard in the tracking and filtering literature. Examples of specific dynamic models were discussed in the report [1].

Measurements are obtained by a network of sensors. We assume there are n sensors, and the measurement obtained at sensor j at time t is given by

$$\mathbf{z}_t^{(j)} = h_{j,t}(\mathbf{x}_t) + \mathbf{w}_{j,t} \ , \tag{2}$$

where $h_{j,t}(\cdot)$ is the function describing the mapping from the current noise source state to the observation at node j at time t, and $\mathbf{w}_{j,t}$ is additive measurement noise at node j at time t. Again, the measurement model is general. Of particular interest in this project is the case where $h_{j,t}(\mathbf{x}_t)$ is the true bearing angle(s) from the sensors position at time t to the noise source position(s).

1.3 Methodology

In our proposed approach, each node runs its own local instance of the tracking algorithm. Rather than having each node operate in isolation, with node j updating its state estimate using only its own measurements, our aim is develop schemes under which sensors share information about their local measurements so that the tracking performance of the overall system is superior to that of any individual sensor. The main issues to be addressed are then:

- What information should be communicated?
- What tracking algorithm should be used at each node?

Below we describe our approach to these issues. With each node running a local tracking algorithm, communication plays the role of synchronizing the state estimates across all nodes in the network so that, ideally, all nodes have the same estimate that would be computed by a single tracking algorithm that had direct access to all of the measurements. We use gossip algorithms [2] to diffuse information across the network and drive the state estimates at each node to a consensus. Gossip algorithms generally refer to iterative distributed message passing algorithms. In one iteration, nodes communicate with their neighbors and then update local variables based on the information received. Iterations can be executed synchronously or asynchronously, and with different communication patterns (e.g., point-to-point, broadcast). As more iterations are performed, the variables at all nodes converge to a consensus on the value being computed. Hence, the accuracy of the computation improves with additional communication, and it is important to understand how errors in the distributed computations effect the performance of distributed tracking algorithms. For the local tracking algorithm used at each node we consider particle filters [3] and random finite set filters [4]. Both types of tracking methods fit the general form of approximate Bayesian filtering, where the state update is divided into *prediction* and *update* steps (analogous to the familiar Kalman filter). In the prediction step, the state is propagated based on the assumed target dynamic model, and in the update step the measurements taken at all nodes are used to correct the prediction to better match the evidence observed in the measurements. Hence, it is in the update step where communication is essential to share information about the measurements among the different sensors.

1.4 Results

1.4.1 Error Bounds for Distributed Particle Filters

Many methods for distributed particle filtering have been proposed in the past five years. Particle filters approximate the posterior distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ of the target state given the measurements from times 1 up to t using a set of m weighted particles, $\{\widehat{\mathbf{x}}_t^{(i)}, w_t^{(i)}\}_{i=1}^m$; specifically,

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \approx \hat{p}_m(\mathbf{x}_t|\mathbf{z}_{1:t}) \stackrel{\text{def}}{=} \sum_{i=1}^{m} w_t^{(i)} \delta(\mathbf{x}_t - \widehat{\mathbf{x}}_t^{(i)}) , \tag{3}$$

where the weights $w_t^{(i)}$ satisfy $\sum_{i=1}^m w_t^{(i)} = 1$ and where $\delta(\cdot)$ is the Dirac delta function.

In the distributed setting, each sensor gathers measurements which are assumed to be conditionally independent given the target state. The challenge is to simultaneously fuse the information contained in these measurements and share it with all nodes in the network in a communication-efficient manner.

Within the particle filter, this fusion boils down to evaluating (either exactly, or approximately) the joint log-likelihood function of the measurements $\mathbf{z}_t^v = (\mathbf{z}_t^{(1)}, \ldots, \mathbf{z}_t^{(n)})$ from all n sensors,

$$\log p(\mathbf{z}_t^V|\mathbf{x}_t) = \log \left(\prod_{j=1}^{n} p(\mathbf{z}_t^{(j)}|\widehat{\mathbf{x}}_t^{(i)}) \right) = \sum_{j=1}^{n} \log p(\mathbf{z}_t^{(j)}|\widehat{\mathbf{x}}_t^{(i)}) ,$$

at each particle value, $\widehat{\mathbf{x}}_t^{(i)}$, $i = 1,\dots,m$. Since the joint log-likelihood is a linear combination of the local log-likelihoods at each sensor, it can be easily calculated using gossip algorithms. However, when only a finite number of gossip iterations is used there will be some residual error introduced into the update equations. Previously it was not known to what extent this error may accumulate and affect the long-run performance of the distributed tracking algorithm.

We have developed a novel general bounding framework to characterize the effect of approximate errors introduced in the context of distributed particle filtering, and we have specialized the bound to the case where gossip algorithms are used for distributed computation. Let $\widehat{\eta}_t^N$ denote the approximate posterior distribution after t time steps when N particles are used, and let η_t denote the exact posterior distribution calculated using the Bayes optimal update rule. Note that computing η_t is not tractable in general and would require having the exact measurements available from all sensors at a single location, but we use it here as a theoretical point of comparison for characterizing the error introduced by using the particle approximation and by using gossip algorithms for approximate joint likelihood calculations. Now, suppose that the absolute error between the exact and approximate joint likelihood function satisfies

$$\frac{\left|\log p(\mathbf{z}_t^V|\mathbf{x}_t) - \log \widehat{p}(\mathbf{z}_t^V|\mathbf{x}_t)\right|}{\left|\log p(\mathbf{z}_t^V|\mathbf{x}_t)\right|} \le \delta \qquad (4)$$

for all particles $\widehat{\mathbf{x}}_t^{(j)}$, $j = 1,\dots,n$. We have shown that for any $p \ge 1$, it holds that

$$\sup_{t\ge 0} \mathbb{E}\left[\left|[\widehat{\eta}_t^N - \eta_t](h_t(\mathbf{x}_t))\right|^p\right]^{1/p} \le \frac{C\varepsilon^{-2\delta}}{\sqrt{N}} ,$$

where C is a constant depending on the target dynamics and likelihood function, $\varepsilon > 0$ is a constant that only depends on characteristics of the likelihood function, and $h_t(\cdot)$ is a test function (e.g., ℓ_2 error or ℓ_1 error).

First, note that the right-hand side gives a bound on the error which holds regardless of the time t, indicating that the distributed particle filter remains stable; i.e., error does not diverge over time, despite the approximate likelihood calculation. Here, the approximate method is being compared with the performance of the Bayes-optimal filter which has access to all measurements and unbounded computational resources. This result also directly quantifies how the approximate log-likelihood accuracy, measured in terms of δ, and the number of particles N, affect the long-run error. The dependence on the number of particles is clear. When a gossip algorithm is used to directly compute the particle filter weights, the dependence on the constant δ can be controlled precisely. Specifically, we have also shown that (4) holds for a given value of δ if

$$\text{\# gossip iterations} \ge \frac{(3/2)\log(n) + \log(\frac{n+1}{\delta})}{\log(1/\rho)} ,$$

where n is the number of nodes and ρ is a parameter related to the connectivity of the network over which gossiping takes place.

The details of the results just discussed are presented in our paper [5] which is currently under review for publication in a peer-reviewed journal.

1.4.2 Generalized Cardinalized Probability Hypothesis Density Filter

The *probability hypothesis density* (PHD) and *cardinalized PHD* (CPHD) filters give a principled and effective approach to multi-target tracking in non-linear and/or non-Gaussian problem settings [4]. They are especially useful in the presence of clutter and when the number of targets may vary over time. In the PHD filter, rather than tracking each target state individually and associating individual measurements to specific targets, the filter tracks the density of targets over the state space. Instead of the dimension of the state $\mathbf{x}_t$ being proportional to the number of targets present, it remains constant (e.g., equal to 4 for 2-d position and 2-d velocity) regardless of the number of targets. This simplification leads to improved tracking performance.

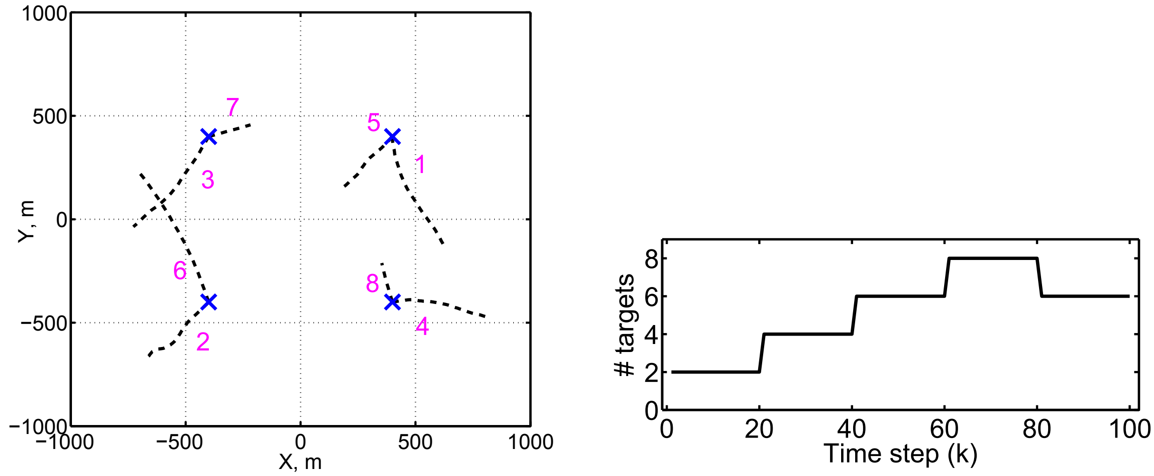
The CPHD filter improves upon the PHD filter by explicitly tracking a distribution on the number of targets over time. The CPHD filter generally has superior tracking accuracy while being more computationally intensive.

Previous work on the PHD and CPHD filters has mainly focused on the setting where the measurements are made by a single sensor monitoring the entire region of interest (e.g., radar). While there has been some prior work proposing heuristics or approximate implementations of the multi-sensor PHD or CPHD filter, these methods are either sensitive to the order in which targets are processed or they are extremely computationally intensive [6, 7, 8, 9, 10, 11, 12].

In our work [13, 14] we derive exact expressions for the multi-sensor CPHD filter. An exact evaluation of these expressions is computationally intractable unless the number of sensors and the number of observations-per-sensor are small. In our report we also propose a computationally-efficient method for approximately evaluating the multi-sensor PHD and CPHD updates. Our approximation approximates the PHD (the density of targets over state-space) using a mixture of Gaussians, and the approximate update involves a greedy lattice search akin to the Baum-Welch algorithm.

Experiments demonstrate that our approach provides superior tracking performance without a significant increase in computational overhead. Figure 1 shows the trajectories of eight simulated targets as well as the number of targets present over time.

Figure 2(a) shows the performance of our generalized PHD and CPHD filter implementations as well as the performance of the iterated-corrector (IC) PHD and CPHD filters [6],



(a) (b)

Figure 1: *(a) Tracks of eight simulated targets for a multi-sensor multi-target tracking experiment. (b) The number of targets present over time.*

as a function of the probability P_d that each sensor detects a target at each time step. Performance is measured in terms of the OSPA error metric which accounts for the accuracy of the estimated track of each target as well as the estimated number of targets [15]. The lower errors of the general PHD and CPHD implementations indicate that they provide more accurate tracks and the generalized CPHD filter also provides a more accurate estimate of the number of targets present. Figure 2(b) illustrates how the average OSPA error decreases as the number of sensors increases. Figure 2(c) illustrates that the computational complexity of the proposed implementation scales gracefully as more sensors are added.

1.4.3 Communication-Efficient Distributed Bearings-Only Tracking in Spherical Coordinates

We have developed a communication-efficient distributed particle filtering method for tracking a target using bearings-only measurements when the target state-space is the surface of a sphere (such as Earth). As mentioned in Section 1.4.1, the key challenge in distributed particle filtering methods is to evaluate the joint log-likelihood function of the sensors' measurements at each particle location. A naïve approach is to directly gossip on individual particle weights, but then the communication overhead scales linearly with the number of particles used. Typically, for reasonable tracking accuracy, one may use between a few 100's to a few 1,000's of particles, and direct communication of particle values clearly is not reasonable with contemporary underwater communication technologies.

The *constraint sufficient statistics distributed particle filter* (CSSDPF) is an efficient method for tracking a single target using bearings-only measurements in the presence of additive Gaussian noise [16]. In particular, Mohammadi and Asif [16] show that the joint log-

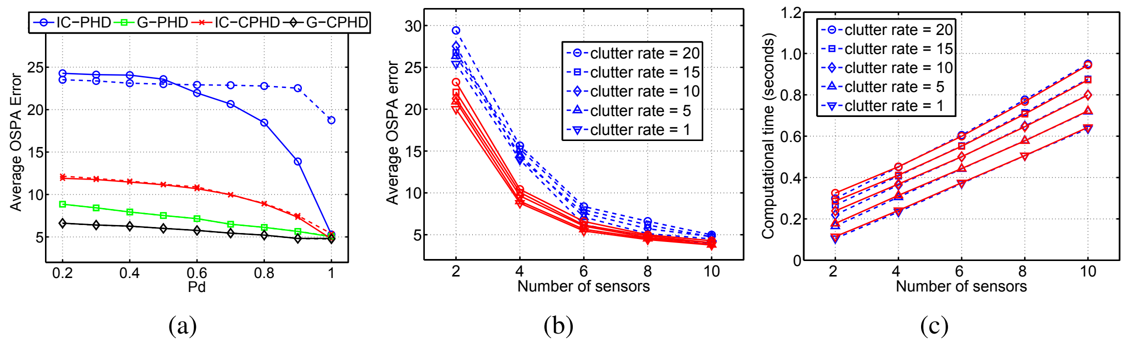


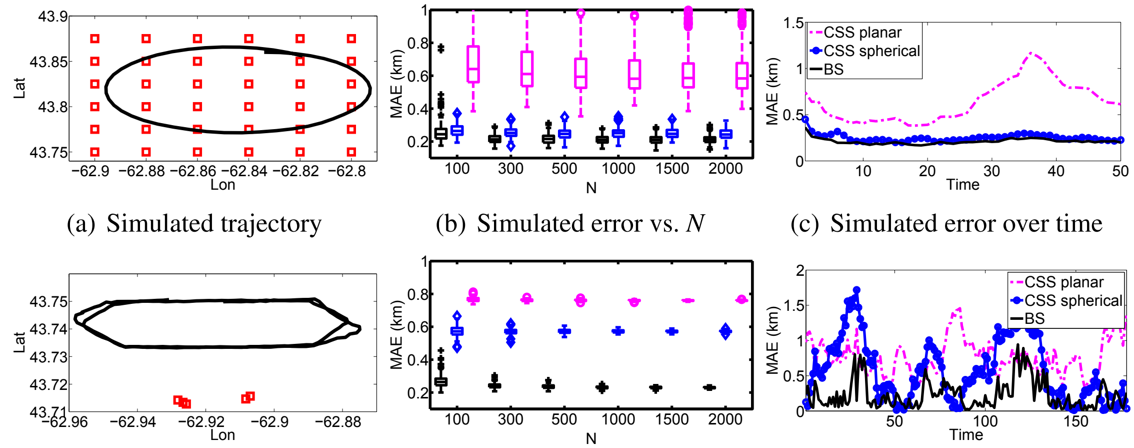
Figure 2: *(a) Performance of the generalized PHD and CPHD filters, as measured in OSPA error, compared to the iterated-corrector PHD and CPHD filters in a two-sensor scenario where one sensor is more accurate than the other. The dashed lines correspond to when the less accurate sensor is processed second, and the solid lines correspond to when the less accurate sensor is processed first. Clearly the performance of the IC-PHD filter depends strongly on the order in which they are processed. (b) OSPA error decreases as the number of sensors increases. (c) Computational complexity of the proposed implementations of the multi-sensor PHD and CPHD filters increases linearly with the number of sensors.*

likelihood calculation can be expressed in terms of six sufficient statistics, each of which is a linear combination of local sufficient statistics that are calculated at each node. Thus, once nodes gossip to compute these six sufficient statistics they can evaluate the joint log-likelihood at any particle location, and so the communication overhead is independent of the number of particles used.

The derivations in [16] explicitly assume that bearings are calculated for nodes with coordinates in the plane. When nodes lie on the surface of a sphere, such as Earth, this planar approximation breaks down [17], especially when the sensors and targets are further from the equator. We have re-derived the CSSDPF accounting for this planar approximation.

Properly accounting for the spherical surface significantly improves the performance for tracking applications considered within the purview of this project. Figure 3(a) and (d) shows trajectories and sensor positions for a simulated scenario and for the sea-trial data set. The remaining panels in Figure 3 illustrate the performance of the CSS approximation as a function of the number of particles used and over time. The three schemes compared in this example are a standard (centralized) bootstrap particle filter (BS), the CSSDPF derived for the planar bearing equation, and the CSSDPF derived for spherical coordinate systems.

The details of these results are presented in our conference paper [18] and in the report [19] which has been submitted for publication.



(a) Simulated trajectory (b) Simulated error vs. N (c) Simulated error over time

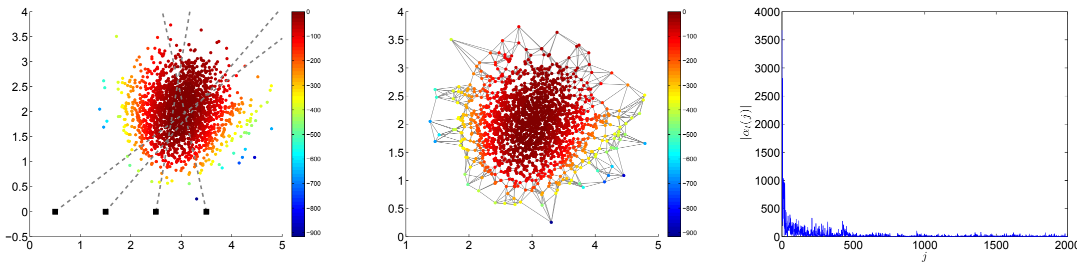
(d) Sea trial trajectory (e) Sea trial error vs. N (f) Sea trial error over time

Figure 3: *Results for experiments with the CSSDPF on (a) a simulated trajectory and (d) data from an at-sea trial. Red squares indicate sensor locations. Panels (b) and (e) show box plots of the mean absolute position error for each experiment as a function of the number of particles for 200 Monte Carlo trials. For each value of* N*, the group of three box plots correspond to (from left to right) the bootstrap particle filter (black* $+$*), the CSSDPF for spherical coordinates (blue* $\diamond$*), and the CSSDPF for planar coordinates (magenta* $\circ$*). Panels (c) and (f) show the average absolute position error over time for* $N = 1000$ *particles.*

1.4.4 Graph Laplacian Distributed Particle Filter

The CSSDPF is appealing because it makes the overall communication overhead independent of the number of particles used. However, the sufficient statistics discussed in the previous section are very specific to the form of the likelihood function for a single target with bearings-only measurements corrupted by additive Gaussian noise. In more general settings (e.g., multiple targets, with clutter, or non-Gaussian noise) it is not clear how to obtain such a simplified factorization of the joint log-likelihood model, and general-purpose schemes are required to carry out distributed joint log-likelihood evaluations in a communication-efficient manner.

We have developed a compression scheme tailored to distributed particle filtering problems. Our compression scheme aims to obtain an approximate joint log-likelihood at each node while exploiting the fact that the nodes are (approximately) synchronized since they are all tracking the same targets. As above, each node executes a local particle filter and gathers local measurements. The nodes communicate with nearby neighbours using gossip algorithms to keep their filters synchronized by fusing their measurements. Consequently, before performing the next update step, the particle locations $\{\widehat{\mathbf{x}}_t^{(i)}\}_{i=1}^m$ at all nodes are synchronized, and only the particle weights $\{w_t^{(i,j)}\}_{i=1}^m$ are different from node-to-node because they have been calculated using only the local measurements at that node.



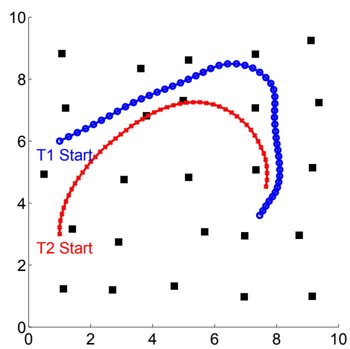
(a) (b) (c)

Figure 4: *(a) An example scenario where four sensors (black squares) obtain bearings measurements to a target (not shown). The colour of each of the 2000 particles corresponds to its log-likelihood value. (b) We form a graph over the particles by connecting each one to its nearest neighbours. (c) The magnitude of the graph Laplacian transform coefficients* $\alpha_t(j)$ *for* $j = 1, \ldots, 2000$*. We can interpret* $\alpha_t(j)$ *as a Fourier coefficient of the signal shown in panel (b), where* $j = 0$ *corresponds to the DC component, and larger* j *correspond to higher frequency components (varying more rapidly over the structure of the graph). Clearly most of the energy of the signal is captured in a few of the low-frequency coefficients.*

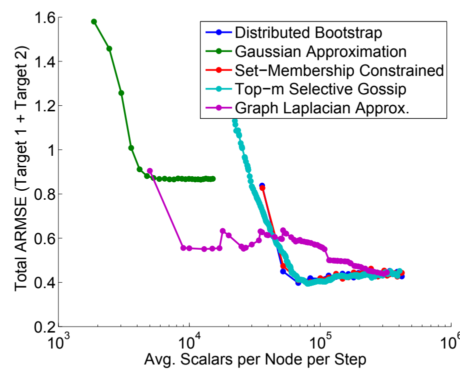
Our compression scheme is based on the principle of transform coding, where the transform used is adapted to the particle distribution over the state space. Specifically, we form a graph over the particle locations. Then we compute the Laplacian matrix of this graph (which encodes the connectivity). The eigenvectors of the Laplacian matrix can be interpreted as a Fourier-like basis adapted to the graph structure. Since likelihood functions are typically smooth (i.e., slowly-varying) as a function of the state space, we expect that the particle weights (which are proportional to the log-likelihood) vary slowly as we move through this "particle graph". Consequently, the joint log-likelihood function, viewed as a signal over the graph, can be well-approximated using only a few "low frequency" eigenvectors of the Laplacian matrix. This process is illustrated in Figure 4.

Now, based on this transform, rather than gossiping on the individual particle weights individually, or on all of the graph Laplacian transform coefficients, nodes can gossip only on the low-frequency coefficients (e.g., $\alpha_t(j)$ for $j = 0, \ldots, 500$. This reduces the communication overhead while incurring additional approximation error. We have studied this tradeoff initially for a simulated scenario shown in Figure 5(a). The results in Figure 5(b) illustrate that, in comparison to other distributed particle filtering approaches, the graph Laplacian approach makes it possible to significantly reduce the communication overhead (measured in number of value transmitted per node per time step) without a significant decrease in tracking accuracy.

Our work on the graph Laplacian distributed particle filter is discussed in the report [20].



(a)



(b)

Figure 5: *(a) A simulated scenario with 25 sensors scattered roughly in a grid and two targets moving in a correlated/coordinated manner. (b) A comparison of the average tracking error as a function of the communication overhead. The proposed graph Laplacian approximation scheme allows one to significantly reduce the communication overhead (number of values transmitted per node per time step) while only suffering a modest increase in average tracking error.*

1.5 Conclusion and Ongoing Work

We continue to investigate the problem of distributed target tracking in a network where each node gathers local measurements and runs a local tracking algorithm, and the nodes communicate to fuse information from their measurements and remain synchronized.

The multi-sensor PHD and CPHD filters we have developed show great promise for tracking multiple targets in challenging scenarios. Presently, these filters assume that all of the measurements are available at a single location for processing. In the next year of the project we will develop distributed versions of these filters which are suited to the gossip-based distributed tracking framework considered within this project.

Our present analysis of error bounds for gossip-based distributed particle filters is based on synchronous gossip iterations which may not be practical for implementation in underwater systems. We will extend these bounds to consider asynchronous broadcast gossip-type communications, modelling unreliable communications where nodes may not always successfully decode messages from their neighbours.

Finally, the graph Laplacian approach to reducing communication overhead is promising. However, implementing this approach requires each sensor to periodically perform an eigenvalue decomposition of a matrix of the same dimension as the number of particles, N. For large values of N this may be a significant computational burden for battery-powered nodes. We are developing compression methods for distributed particle filters which offer the same reduction in communication overhead with significantly less computational overhead by using clustering and approximate eigendecomoposition routines.

This page intentionally left blank.

References

[1] Rabbat, M. and Coates, M. (2015), Relaxation of Distributed Data Aggregation for Underwater Acoustic Sensor Networks – Phase 1 Report, (Technical Report) Contract Report #AMBUSH.1.2 (DRDC-RDDC-2015-C022), Contract #W7707-145675, McGill University (Montreal, QC, Canada).

[2] Dimakis, A. G., Kar, S., Moura, J. M., Rabbat, M. G., and Scaglione, A. (2010), Gossip Algorithms for Distributed Signal Processing, *Proceedings of the IEEE*, 98(11), 1847 – 1864.

[3] Doucet, A. and Johansen, A. M. (2009), A tutorial on particle filtering and smoothing: Fifteen years later, In *Handbook of Nonlinear Filtering*, Ch. 12, pp. 656–704, Oxford, UK: Oxford University Press.

[4] Mahler, R. (2014), Advances in statistical multisource-multitarget information fusion, Artech House, Boston.

[5] Datta Gupta, S., Coates, M., and Rabbat, M. (2015), Error propagation in gossip-based distributed particle filters. McGill University Technical Report, submitted for publication.

[6] Mahler, R. (2009), The multisensor PHD filter: I. General solution via multitarget calculus, In *Proc. SPIE Int. Conf. Sig. Proc., Sensor Fusion, Target Recog.*, Orlando, FL, U.S.A.

[7] Mahler, R. (2009), The multisensor PHD filter: II. Erroneous solution via Poisson magic, In *Proc. SPIE Int. Conf. Sig. Proc., Sensor Fusion, Target Recog.*, Orlando, FL, U.S.A.

[8] Delande, E., Duflos, E., Heurguier, D., and Vanheeghe, P. (2010), Multi-target PHD filtering: proposition of extensions to the multi-sensor case, *Research Report RR-7337, INRIA*.

[9] Delande, E., Duflos, E., Vanheeghe, P., and Heurguier, D. (2011), Multi-sensor PHD: Construction and implementation by space partitioning, In *Proc. Int. Conf. Acoustics, Speech and Signal Proc.*, Prague, Czech Republic.

[10] Delande, E., Duflos, E., Vanheeghe, P., and Heurguier, D. (2011), Multi-sensor PHD by space partitioning: computation of a true reference density within the PHD framework, In *Proc. Stat. Signal Proc. Workshop*, Nice, France.

[11] Jian, X., Huang, F.-M., and Huang, Z.-L. (2013), The multi-sensor PHD filter: Analytic implementation via Gaussian mixture and effective binary partition, In *Proc. Int. Conf. Inf. Fusion*, Istanbul, Turkey.

[12] Nagappa, S. and Clark, D. E. (2011), On the ordering of the sensors in the iterated-corrector probability hypothesis density (PHD) filter, In *Proc. SPIE Int. Conf. Sig. Proc., Sensor Fusion, Target Recog.*, Orlando, FL, U.S.A.

[13] Nannuru, S., Coates, M., Rabbat, M., and Blouin, S. (2015), General solution and approximate implementation of the multisensor multitarget CPHD filter, In *Proc. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Brisbane, Australia.

[14] Nannuru, S., Blouin, S., Coates, M., and Rabbat, M. (2015), Multisensor CPHD filter. McGill University Technical Report.

[15] Schuhmacher, D., Vo, B.-T., and Vo, B.-N. (2008), A Consistent Metric for Performance Evaluation of Multi-Object Filters, *IEEE Trans. Signal Proc.*, 56(8), 3447–3457.

[16] Mohammadi, A. and Asif, A. (2012), A constraint sufficient statistics based distributed particle filter for bearing only tracking, In *IEEE Intl. Conf. on Communications*, pp. 3670–3675.

[17] Peters, D. (2005), Flatlanders in Space: Three-Dimensional Transformations of Two-Dimensional Data, (Technical Report 2005-259) Defense Research and Development Canada, Dartmouth, NS.

[18] Yu, J.-Y., Rabbat, M., Coates, M., and Blouin, S. (2015), Performance investigation on constraint sufficient statistics distributed particle filter, In *Proc. IEEE Canadian Conference on Electrical and Computer Engineering*, Halifax, NS, Canada.

[19] Yu, J.-Y., Coates, M., Rabbat, M., and Blouin, S. (2014), A Distributed Particle Filter for Bearings-Only Tracking on Spherical Surfaces. McGill University Technical Report, submitted for publication.

[20] Rabbat, M., Coates, M., Yu, J.-Y., Üstebay, D., and Blouin, S. (2015), Approximation methods for gossip-based distributed particle filters. McGill University Technical Report, in preparation.