DISTRIBUTED FUSION IN UNDERWATER SENSOR NETWORKS: FUSING BEARING INFORMATION

Roald Otnes[a], Per Zetterberg[b], Stephane Blouin[c], Magnus Lundberg Nordenvaad[b], Håvard Austad[a], Elin Dombestein[a]

[a]Norwegian Defence Research Establishment (FFI), Horten, Norway
[b]Swedish Defence Research Agency (FOI), Kista, Sweden
[c]Defence Research and Development Canada (DRDC) Atlantic, Dartmouth, NS, Canada

Roald Otnes, Norwegian Defence Research Establishment (FFI), PO box 115, NO-3191 Horten, Norway, email: roald.otnes@ffi.no

Abstract: *In underwater sensor networks, distributed data fusion may be more efficient than centralized fusion because the limited data transmission capacity can make it difficult to collect all required sensor information at a centralized fusion centre. In this paper, we investigate three distributed fusion techniques applied to a network of passive acoustic underwater sensor nodes. We focus on the process of having a node combining its own bearing-to-target information with bearing-to-target information received from another node. In one of the techniques, we approximate the uncertainty in crossfixes in Cartesian coordinates by a Gaussian distribution with their second-order statistics derived from an exact distribution. The bearings and covariance matrixes are fed into a Kalman filter for tracking. The other methods are a particle filter using an exact distribution, and a distributed particle filter using an approximate likelihood representation. The performance of the methods is investigated on simulated data as well as on real-world data collected by seafloor sensor nodes during a Stockholm Archipelago sea trial in the trilateral collaborative project DUSN (Distributed Underwater Sensor Networks) between Canada, Norway, and Sweden.*

Keywords: *Underwater sensor networks, data fusion, tracking*

1. INTRODUCTION

Distributed fusion in underwater sensor networks is a main topic in the trilateral collaborative project DUSN (Distributed Underwater Sensor Networks) between Canada, Norway, and Sweden. Distributed data fusion may be more efficient than centralized fusion because the limited data transmission capacity can make it difficult to collect all required sensor information at a centralized fusion centre.

Small and easily deployable underwater sensor nodes equipped with passive acoustic sensors provide bearing-only estimates, with limited accuracy due to small array sizes. In this paper we focus on one step of distributed fusion, where a sensor node has received bearing-to-target information deemed worthy of reporting from another node, and combines it with its own bearing-to-target information. We investigate the problem through a case study using real-world data, and also using idealized synthetic data for the same scenario.

2. CASE STUDIES

2.1. Real-world data

At a DUSN sea trial in September 2018, underwater sensor nodes from Canada, Norway, and Sweden were deployed in an area in the Stockholm archipelago. We here use data from the Norwegian NILUS (Networked Intelligent Underwater Sensors) sensor nodes [1][2], which are equipped with a tetrahedral hydrophone array and local signal processing [3], which enables them to estimate bearing to target(s) and also to produce an estimate of the uncertainty in the bearing estimates.

Fig. 1 (left) shows the case study geometry. In a littoral area, three NILUS nodes were deployed on the sea floor in the marked positions. A surface vessel passed through the sensor field from North to South along the indicated trajectory.

Each sensor node estimated at least one bearing and bearing uncertainty every 5 s, as shown in Fig. 1 (right). There was also an algorithm running in the sensor nodes determining which bearing estimates were worthy of reporting to other nodes, typically in a time interval around CPA (closest point of approach). The bearings marked with blue dots were transmitted to the other nodes along with their uncertainties. We also see some detections of other targets farther away, but these bearings were not transmitted to other nodes.

We focus on node 1 receiving the “blue” bearings from nodes 2 and/or 3. Node 1 can combine bearings from other nodes with *all* its own bearing estimates, not only the “blue” ones. Herein lies a possible benefit of distributed fusion over centralized: In a centralized fusion scheme it would put a heavy toll on the communication network to always transmit all bearing estimates (and not only the “blue” ones) to the fusion centre, whereas in distributed fusion each node’s local fusion processor can also make use of the information it did not find worthy of reporting just based on its own knowledge.

Fig. 2 shows histograms of bearing errors in selected time intervals in the real-world data set. Note that the data sets are small, and their uncertainty value σ represents both an unknown (local) bias value and variation from sample to sample, therefore they do not fit well to an analytical zero-mean probability density function (pdf) with σ as standard deviation. We apply the analytical zero-mean von Mises pdf in further tracking, acknowledging that we thereby make a coarse approximation. The von Mises distribution [4] is more suitable to circular data than the Gauss distribution, and as Fig. 2 shows they are quite similar when we do not focus on the tails.

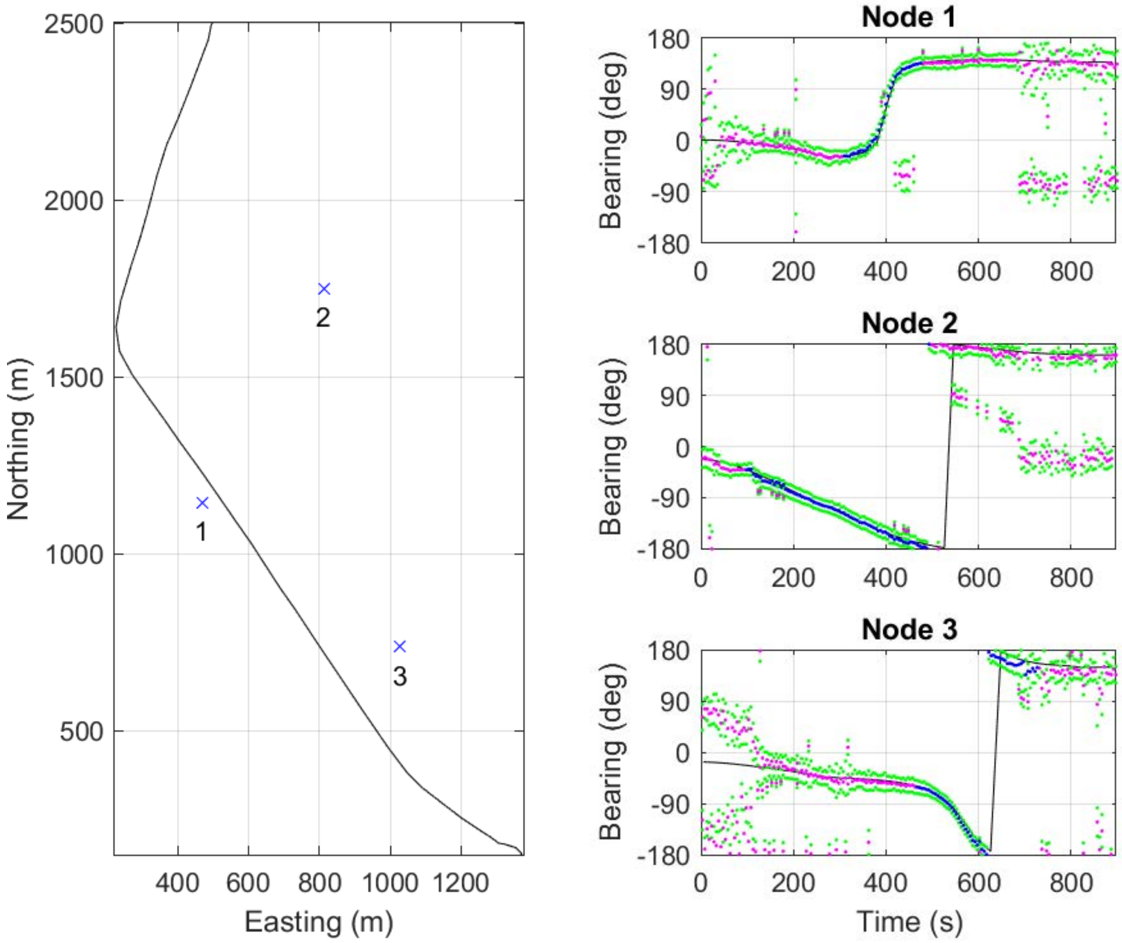

Fig. 1: ***Left:*** *Case study geometry. Crosses denote seafloor sensor nodes, black line shows target's GPS track.* ***Right:*** *Bearing estimates (magenta and blue dots) computed by the three nodes. Blue dots show bearings reported to the other nodes. Green dots show bearing uncertainty (±1σ) estimated by the nodes. Black lines show actual bearing to target.*

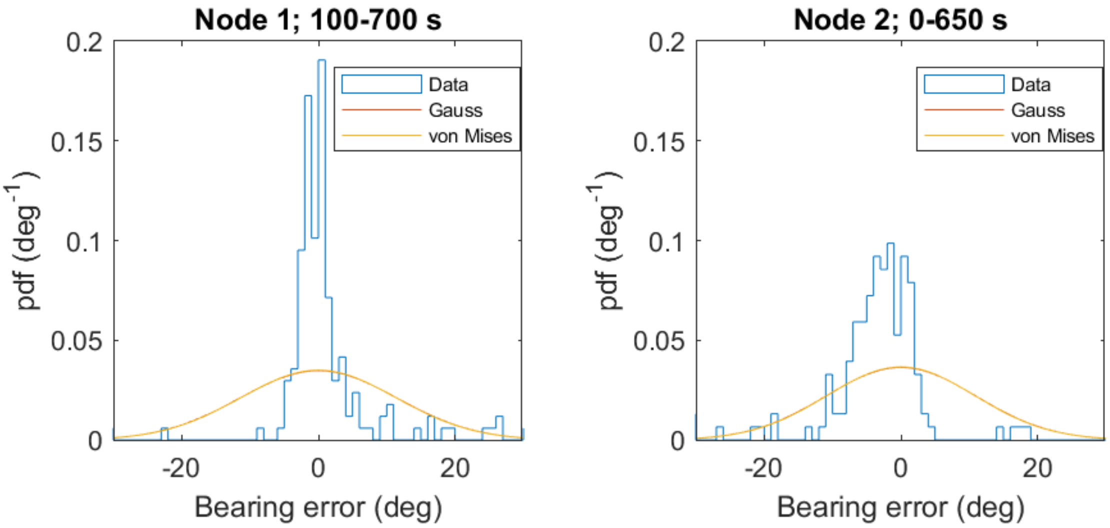

Fig. 2: Histograms of bearing errors over selected time intervals, along with Gauss and von Mises distributions (almost overlapping curves) with the average σ value reported as uncertainty over the same intervals. There are also sporadic bearing errors outside the plotted x axis limits.

2.2. Synthetic data

To also assess algorithm performance on a cleaner but less realistic data set, we generate synthetic data corresponding to the same geometry as the real-world data set, shown in Fig. 1 (left). We assume each node produces a bearing estimate every 5 s, drawn from a pdf which is a mixture between a uniform distribution and a von Mises distribution: $p(\theta) = \alpha \mathrm{U}(-\pi, \pi) + (1-\alpha)\mathrm{vM}(\mu, \kappa)$ [11], where U(a,b) is the uniform distribution from a to b, and vM(μ,κ) is the von Mises distribution with mean μ and an equivalent concentration parameter κ computed from the standard deviation σ using Eqns. (3.30) and (4.40) in [4].

The uniformly distributed element of the mixture model corresponds to the possibility of detecting other targets or noise instead of the target of interest. We let the distributed fusion processor know the standard deviation σ, but let it assume α to be zero. In the real-world data set of Fig. 1 (right), we see that α is close to zero when the target of interest is near CPA (closest point of approach), but larger elsewhere. We assume that node 2 and 3 send bearings to node 1 for the same time intervals as for the real-world data set (blue points in Fig. 1).

2.3. Communication pattern

The communication pattern in the scenario described above consists of three messages: Node 2 and node 3 each send one message with a series of "blue" bearing estimates[1]. These two messages are received by node 1, which combines with its own bearing estimates, runs a tracker on the data, and produces a third message with track information which is transmitted further in the network. This is one step of a larger distributed fusion approach, the remainder of which is outside the scope of this paper.

3. TRADITIONAL TRACKING BASED ON SECOND-ORDER STATISTICS

From the estimates of bearings and bearing uncertainties from the two nodes at a point in time, we compute the crossfix and its covariance matrix using Eqns. (8)-(14) in [5]. The covariance matrix represents the second-order statistics, which can be represented by uncertainty ellipses. This constitutes an approximation, as the actual pdf will be more complicated. To compute the actual pdf numerically for comparison, we assume that the target position is *a priori* uniformly distributed on a Cartesian grid, except in regions close to a node (to avoid singularities). It can then be shown that, on the supported region, we have

$$p(x,y \mid \theta_1, \theta_2) \propto p_1(\theta_1(x,y)) p_2(\theta_2(x,y)) \left| J(x,y) \right| \tag{1}$$

$$\left| J(x,y) \right| = \frac{(y-y_2)(x-x_1)-(y-y_1)(x-x_2)}{r_1^2 r_2^2}$$

where θ_i and r_i are the bearing and distance from node i to each point (x,y), $p_i(\theta)$ are the bearing pdfs, (x_i,y_i) are the node positions, and $|J(x,y)|$ is the determinant of the Jacobian given by Eqns. (12)-(14) in [5]. In Fig. 3 we show an example comparison between the actual pdf and the uncertainty ellipses assuming second-order statistics only, where the non-elliptical nature of the actual pdf can be clearly seen.

[1] Bearings can be exchanged between nodes more often, but communicating one message for every 5-s time step between bearing estimates would probably be too frequent for an underwater acoustic communication network.

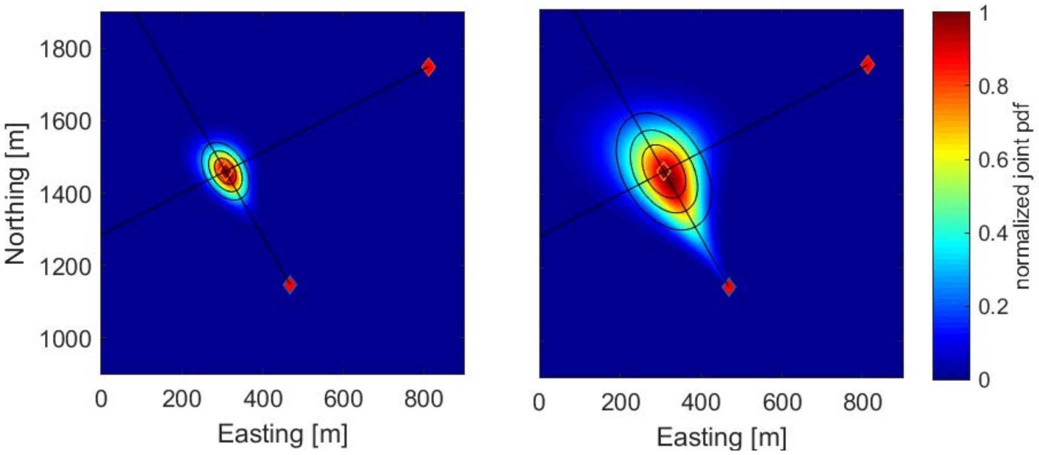

Fig. 3: Crossfix and pdfs for a point along the target trajectory, assuming von Mises distributed bearings with σ=5° (left) and σ=10° (right). The error ellipses shown indicate 25%, 50% and 75% quantiles if we use second-order statistics only.

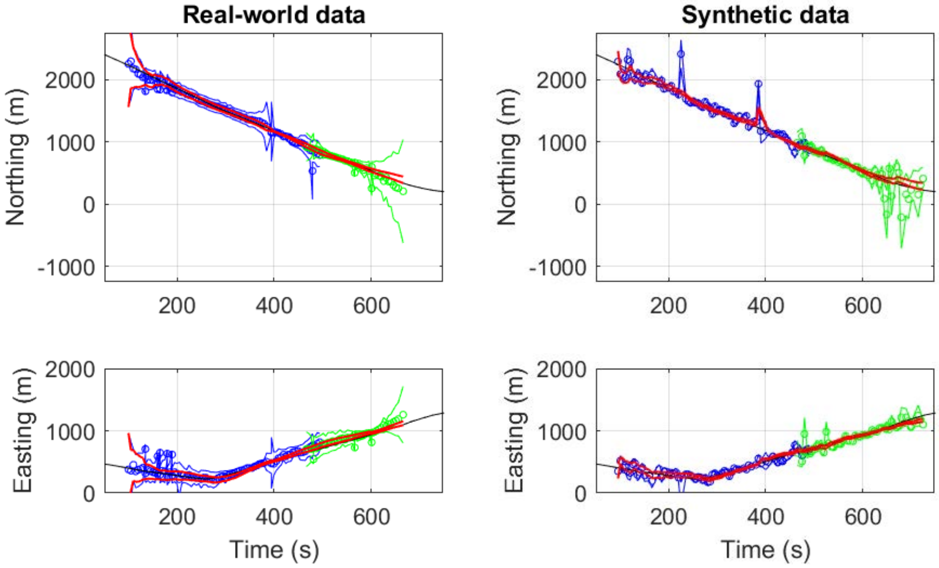

Fig. 4: Crossfixes and Kalman filter track computed by node 1. Blue(green): Crossfixes by combining own bearings with those received from node 2(3). Circles indicate crossfix position and lines indicate position ±1 std.dev. along each direction. Red: Track output from Kalman filter. Position not shown; lines indicate position ±1 std.dev. along each direction. Black line: Actual target track

We let our node 1 compute crossfixes (*x*,*y*) and covariance matrixes **R** based on its own stored bearing estimates (all points for node 1 in Fig. 1) and the estimates received from nodes 2 and 3 (blue points in Fig. 1), for pairs of bearings which are sufficiently close in time. The crossfixes are used as measurements in a Kalman filter, along with covariance matrices representing the measurement uncertainty. Sometimes a bearing from node 2 and a bearing from node 3 are matched with the same bearing from node 1. We neglect the fact that these two crossfixes are statistically dependent. Sometimes the bearings may not cross or the crossfix uncertainty is very large (>1000 m) in which case they are discarded.

Fig. 4 shows the result from node 1 crossfixing its own bearings with bearings received from nodes 2 and 3, in blue and green, respectively. A state-space model with four states (the x- and y- coordinates and their derivatives) is used together with a constant velocity motion model. The output from the Kalman filter tracker is shown in red in Fig. 4. For clarity we do not show the track coordinate here, but only the lines that represent the uncertainty.

For the real-world data, when comparing the crossfixes to ground truth they are always within the reported uncertainty. This uncertainty is larger when the target is farthest from node 1 because (a) node 1's own bearing uncertainty then becomes larger, and (2) the geometry becomes more collinear. The tracker output is within the reported uncertainty compared to ground truth. For the synthetic data (with parameters α=0.08 and σ=5°), the outliers due to non-zero α have too small reported uncertainties, and sometimes confuse the tracker. However, it quickly converges towards the correct track again.

4. PARTICLE FILTER APPROACHES

4.1. Basic particle filter

Particle filters are based on the idea of discretizing the probability distribution of the target state vector. This enables non-linear measurement equations and non-Gaussian noise to be easily included into the framework. This is utilized here to update the particles directly from the bearing estimates without estimating the cross-fixes. This enables a more flexible utilization of the measurement data with no explicit demand for time alignment of measurement data, and allows us to do measurement update even with bearings from a single node only (for introduction to particle filters for tracking, see [8]). We also avoid re-using the same bearing as describe above for the Kalman filter. The particle filter used here is similar to the basic particle filter of [11]. One difference with [11] is the initialization. Here we initialize the particles based on the first measurement in a pie-shaped area assuming a distance from the node of 15 to 1500 meters and an azimuth angle distributed according to the von Mises distribution. The particles are naturally propagated through the process update equation. The update step is done by calculating the likelihood of the latest arrived measurement with respect to each particle which is followed by systematic resampling. The likelihood is calculated using the von Mises distribution with α=0 and κ obtained from the uncertainty estimate of the bearing estimator.

The particle filter was applied to the real dataset described above. The root-mean squared (RMS) error of the Kalman filter was 42.8 m while for the particle filter described here it is 56.3 m. This is a bit surprising given all the benefits of the particle filter we just described. However, this could be due to modelling errors or be unique to this data set.

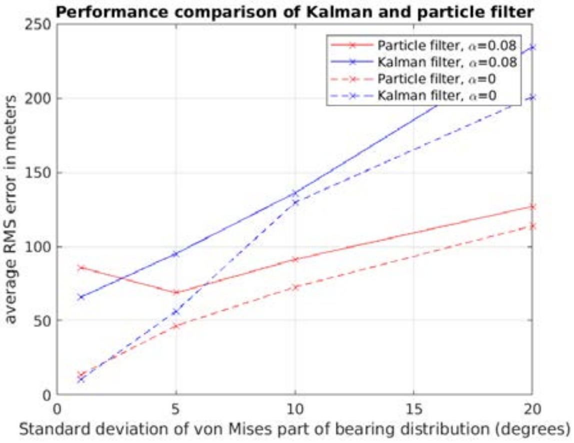

Fig. 5: Comparison of basic particle filter and Kalman filter on synthetic data. Both filters apply α=0 in all cases, while they are given the correct standard deviation.

To assess our earlier claims of the advantages of particle filters, we did a simulation study using the same target track but synthetic bearing data using the mixture model of Sec. 2.2.

The results are shown in Fig. 5 for different values of σ and α. The average RMS error over ten runs are consistently better for the particle than the Kalman filter except for σ=1° and α=0.08. In this case, the particles sometimes get stuck in a point due to uniformly distributed bearings – and the filter is unable to recover. Note that even though the average RMS values (over ten runs) for the particle filter is better than Kalman, a few runs shows a better performance for the Kalman filter.

4.2. Distributed Particle Filter

In this section, we investigate a Distributed Particle Filter (DPF) (developed in [6]) and assess its performance against the real dataset described in Section 2. This filter falls in the category of constraint sufficient statistics (CSS) schemes. Indeed, rather weak constraints must be satisfied [7] for approximating the Global Sufficient Statistics (GSS) through the computation of Local Sufficient Statistics (LSS). Those local statistics are meant to represent the likelihood function as seen by individual nodes. The bootstrap approach [10] is used for the Sequential Importance Sampling (SIS) and re-sampling is performed at each time step based on the Systemic Resampling algorithm of [8]. Regardless of the number of particles used, only the six LSS values need to be communicated to other nodes. Like the basic particle filter described above, but unlike the Kalman filter, no bearing measurement is used twice.

All nodes (including node 1) locally run the DPF algorithm described above. Moreover, we assume that all other nodes hear the bearings transmitted by one node. For convenience purposes, we here also assume that LSS values are exchanged amongst all nodes at every time step. A time-driven approach is taken here, but the approach could also be implemented as playback of past data after receiving a series of bearings from another node. In summary, at each time step each node receives and only uses: (a) the latest LSS values from other nodes, and (b) its own and most recent bearing. In addition to this, a node sometimes receives bearings from one or many other nodes. In real applications, LSS values could only be transmitted along with new bearing data.

The proposed approach uses the conventional discrete-time constant velocity (CV) target motion developed in [9] along with counter-clockwise coordinated turns (CCT) as proposed in [8]. In that setup, the filter state vector is comprised of the estimated target position and velocity, in the latitude and longitude space (the other two filters operate in the X-Y space).

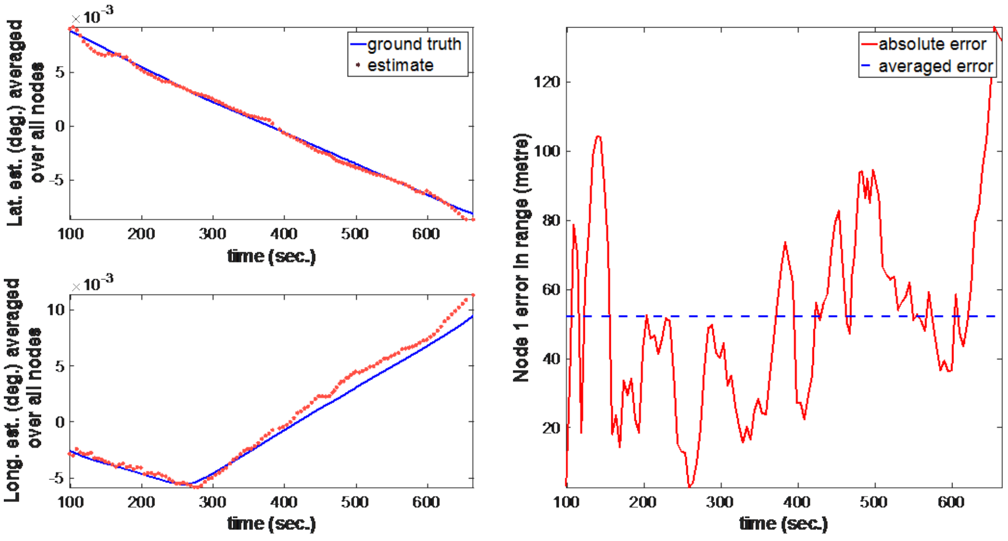

Fig. 6: Simulation results of the Distributed Particle Filter with the real dataset. Left: de-referenced latitude and longitude of both the target and the averaged (over all nodes) target position estimate. Right: Time evolution of the root mean squared (RMS) error from node 1.

Fig. 6 shows results of applying this DPF to the real dataset with 50,000 particles (using 1,000 particles only leads to marginally worse results). The averaged RMS error is 52 m, which is comparable to the other two schemes.

5. CONCLUDING REMARKS

We have discussed three different techniques of locally fusing measurements in bearing-only trackers. Moreover, real and synthetic datasets are used for performing an initial comparison of their relative performances. The synthetic datasets have bearings based on a mixture of uniform and von Mises distributions.

Even though each tracker has distinct internal processing approaches and use of measurement data, they have comparable performance on the real data set. Over synthetic datasets, the basic particle filter seems to outperform the Kalman filter in most cases.

Future work will include analysis of the performance and extensions of the proposed bearing-only trackers, and to experimentally validate the present results.

REFERENCES

[1] **R. Otnes**, NILUS – An Underwater Acoustic Sensor Network Demonstrator System, In *10th International Mine Warfare Technology Symposium*, Monterey, CA, USA, 2012. Online: https://www.ffi.no/no/Publikasjoner/Documents/NILUS2012.pdf

[2] **R. Otnes, E. v. d. Spek, M. E. G. D. Colin, H. S. Dol, C. E. Solberg**, Easily deployable underwater sensor nodes – the NILUS MK 2 Demonstrator System, In *UDT Europe*, Oslo, Norway, 2016.
Online: https://www.ffi.no/no/Publikasjoner/Documents/NILUS_UDT_Europe_2016.pdf

[3] **R. Otnes, J. Eastwood, M. E. G. D. Colin**, Using GStreamer for acoustic signal processing in deployable sensor nodes, In *MTS/IEEE Oceans*, Genoa, Italy, 2016.

[4] **N. I. Fisher**, *Statistical Analysis of Circular Data*, Cambridge University Press, 1996.

[5] **E. Taghavi, R. Tharmarasa, T. Kirubarajan, M. McDonald**, Multisensor-Multitarget Bearing-Only Sensor Registration, *IEEE Tr Aerospace and Electronic Systems*, 52 (4), pp. 1654-1666, 2016.

[6] J. Y. Yu, M. J. Coates, M. G. Rabbat, S. Blouin, A Distributed Particle Filter for Bearings-Only Tracking on Spherical Surfaces, *IEEE Signal Processing Letters*, 23 (3), pp. 326-330, 2016**.**

[7] **A. Mohammadi, A. Asif,** A constraint sufficient statistics based distributed particle filter for bearing only tracking" *IEEE Int. Conf. Communications*, pp. 3670-3675 Jun. 2012.

[8] **B. Ristic, S. Arulampalam, N. J. Gordon,** *Beyond the Kalman filter: Particle filters for tracking applications,* Artech House, 2004.

[9] **Y. Bar-Shalom, X. R. Li, T. Kirubarajan,** *Estimation with Applications to Tracking and Navigation*, Wiley, 2004.

[10] **O. Hlinka, F. Hlawatsch, P. M. Djuric**, Distributed particle filtering in agent networks: A survey, classification, and comparison, *IEEE Signal Processing Magazine*, 30 (1), pp. 61-81, 2013.

[11] **E. Rabe, E. Gudmundson, M. L. Nordenvaad**. Target tracking using bearings-only sensor nodes and a mixture model, in *MTS/IEEE Oceans*, Aberdeen, UK, 2017.