Proceeding of the IEEE 28th
Canadian Conference on Electrical and Computer Engineering
Halifax, Canada, May 3-6, 2015

Performance Investigation on Constraint Sufficient Statistics Distributed Particle Filter

Jun Ye Yu, Michael Rabbat, Mark Coates and Stephane Blouin

***Abstract*— The constraint sufficient statistics distributed particle filter is a novel and effective solution for bearings-only single target tracking. The algorithm achieves a significant reduction in communication overhead by factorizing the likelihood function without suffering a major decrease in accuracy. However, the algorithm has some limitations which we discuss and explore in this paper. In particular, the algorithm has a bias induced via the approximate likelihood calculation, depending on the geometry of the sensors relative to the target.**

I. INTRODUCTION

Bearings-only target tracking is a key task in many applications (e.g. submarine tracking, aircraft surveillance and whale migration monitoring, etc.). The objective is to estimate the state of a maneuvering target using noisy bearing measurements. If multiple sensors are used for the task, the measurements from all sensors should be processed to improve tracking results. Often, a distributed solution is preferred over a centralized solution because it is more robust to sensor failures. Rather than relaying all measurements to a fusion center which can be a potential bottleneck, sensors communicate among themselves and exchange information via gossip algorithms.

Distributed particle filters are an area of active interest [1]. Each sensor uses a set of weighted particles to model the posterior distribution of the target state. These weights are calculated using each sensor's own received bearing measurements. In order to reach consensus on the particle weights (based on measurements from all sensors), sensors exchange particle weights or key statistics of the weight distribution using gossip algorithms, which may require considerable communication overhead.

A variety of approaches to reduce the overhead have been proposed in the literature. In particular, the *constraint sufficient statistics distributed particle filter* (CSSDPF) [2], [3] achieves significant communication reduction without suffering a major decrease in accuracy. CSSDPF approximates the likelihood function as a linear function of six sufficient statistics. Thus, sensors only need to reach consensus on the six statistics to obtain the same likelihood function, so there is no need to gossip about the individual particle weights. Depending on the number of particles, the communication overhead can be potentially reduced by two orders of magnitude. However, in order to linearize the likelihod function and reduce overhead, CSSDPF makes use of two approximations in its derivation [4], [2]. In this paper, we explore and investigate how these approximations affect the performance of CSSDPF. More specifically, we show that the algorithm has an inherent bias induced by the approximate likelihood function which degrades the algorithm's performance and we investigate how this bias can be mitigated. We point out that, while the algorithm considered in this algorithm uses spherical measurement model [5], this bias also manifests itself when planar measurement model is used instead.

The rest of the paper is organized as follows. Section II describes CSSDPF. Section III compares the performance of the algorithm with and without the approximations and point out the limitations of the algorithm. We explore these limitations in detail in Section IV. Finally, Section V concludes the paper.

This work was supported by PWGSC contract W7707-145675/001/HAL.

J.Y. Yu, M.G. Rabbat, and M.J. Coates are with the Department of Electrical and Computer Engineering, McGill University, Montreal, Canada. jun.y.yu@mail.mcgill.ca, michael.rabbat@mcgill.ca, mark.coates@mcgill.ca

S. Blouin is with the Defence Research and Development Canada Atlantic Research Centre, Dartmouth, Canada. stephane.blouin@drdc-rddc.gc.ca

II. CSSDPF

In this section, we present briefly the CSSDPF algorithm for tracking single target on spherical surface. Interested readers are referred to [4] for more details.

To account for the curvature of Earth, CSSDPF [4] uses the spherical measurement model [5] to compute the target bearing at sensor k, θ^k, defined as

$$\arctan\left(\frac{\sin(x_1 - s_1^k)\cos(x_2)}{\cos(s_2^k)\sin(x_2) - \sin(s_2^k)\cos(x_2)\cos(x_1 - s_1^k)}\right) \quad (1)$$

where (x_1, x_2) are the target longitude and latitude and (s_1^k, s_2^k) are the sensor coordinates. All coordinates as well as the bearing measurements are expressed in radians. The bearing, θ, is clockwise positive with 0 radians corresponding to true North.

All bearing measurements are assumed to be corrupted by zero-mean additive Gaussian noise with variance σ^2, so the received noisy bearing is defined as

$$z^k = \theta^k + \zeta^k \quad (2)$$

where $\zeta^k \sim \mathcal{N}(0, \sigma^2)$ is the measurement noise.

Eq. 2 can be rewritten as

$$\sin(2z^k)L_2^k - \cos(2z^k)L_1^k = \sin(2\zeta^k)L_2^k + \cos(2\zeta^k)L_1^k \quad (3)$$

where

$$L_1^k(x^i) = \sin(x_1^i - s_1^k)\cos(x_2^i) \quad (4)$$

$$L_2^k(x^i) = \cos(s_2^k)\sin(x_2^i) - \sin(s_2^k)\cos(x_2^i)\cos(x_1^i - s_1^k) \quad (5)$$

CSSDPF approximates the left-hand side of Eq. 3 as following a Gaussian distribution, so the log-likelihood of a particle x^i given bearing z^k measured at sensor k can be expressed as a linear function of six sufficient statistics [4]

$$\log p(z^k|x^i) \approx -\sum_{j=1}^{6} G_j(x^i)\frac{C_j^k}{2R_k(x^i)} - \frac{\log(2\pi R_k(x_i))}{2} \tag{6}$$

where $R_k(x^i)$ is the variance for the approximate Gaussian likelihood function and is defined as

$$\begin{aligned} R_k(x^i) =& \frac{1-e^{-8\sigma^2}}{2}(L_2^k(x^i))^2 + \frac{1+e^{-8\sigma^2}}{2}(L_1^k(x^i))^2 \\ & - e^{-4\sigma^2}(L_1^k(x^i))^2 \end{aligned} \tag{7}$$

We omit the expressions for C_j^k and $G_j(x^i)$ since they do not pertain to our discussion (see [4] for details). We simply note that C_j^k depends only on the coordinates of sensor k and measurements z^k and that $G_j(x^i)$ only depends on particle x^i.

Finally, the likelihood of particle x^i given measurements at all sensors is

$$\begin{aligned} \log p(z^1, ...z^K|x^i) \approx & -\sum_{j=1}^{6}\left(G_j(x^i)\sum_{k=1}^{K}\frac{C_j^k}{2R_k(x^i)}\right) \\ & -\sum_{k=1}^{K}\frac{\log(2\pi R_k(x_i))}{2} \end{aligned} \tag{8}$$

Since $R_k(x^i)$ is particle-dependent, in order to calculate particle weights using Eq. 8 in a distributed fashion, sensors must compute six statistics ($C_j^k/R_k(x^i)$) for *each particle*. If there are N particles, then all sensors must reach consensus on $6N$ scalar values, which is clearly non-optimal as we can simply gossip on the N particle weights instead.

In order to reduce communication overhead, the particle-dependent term $R_k(x^i)$ is approximated by its weighted average $\hat{R}_k = \sum_{i=1}^{n} R_k(x^i)w_i$ so the likelihood becomes

$$\begin{aligned} \log p(z^1, ...z^K|x^i) \approx & -\sum_{j=1}^{6}\left(G_j(x^i)\sum_{k=1}^{K}\frac{C_j^k}{2\hat{R}_k}\right) \\ & -\sum_{k=1}^{K}\frac{\log(2\pi\hat{R}_k)}{2} \end{aligned} \tag{9}$$

where, in practice, the log normalization term is dropped since it is constant and independent of x^i. This new likelihood function, unlike Eq. 8, can be fully characterized by $C_j^k/\hat{R}_k$ which is particle-independent. Thus the sensors only need to gossip on these terms to compute their sum.

III. Impact of Approximation

In this section, we investigate the impact of the approximations on the performance of CSSDPF. We consider the bootstrap particle filter (with no approximation) with spherical measurement model (Eq. 1) as baseline. Next, we implement the CSSDPF with Eq. 8 as the likelihood function and refer to it as CSSDPF$_{\text{exact}}$ for the remainder of the paper. Finally, we implement the CSSDPF with Eq. 9 as the likelihood function and refer to it as CSSDPF$_{\text{approx}}$. To exclude additional error from running a finite number of gossip iterations, for the two CSSDPF algorithms, we compute the six sufficient statistics ($\sum_{k=1}^{K} C_j^k/\hat{R}_k, j = 1, ...6$) exactly.

We use both simulated data and data from a sea trial. We choose *mean absolute error* (MAE) as our performance metric. Unless otherwise specified, all tests are performed with 200 random trials. Note that the MAE is measured in kilometers between the estimated target position and true target position. For the simulated data, the track is generated first and remains fixed through all trials. However, the noisy sensor measurements and process noise differ at each trial. For the sea trial, we have only one single track and one set of measurements, thus they remain fixed through all trials, and the only randomness is in the particle filter.

The dynamic model of the filter is a combination of two different motion models [6]. At each time instant, with probability P_{cv}, the target moves at constant velocity; otherwise, the target turns at constant rate. Let $x_t = [x_{1,t}, x_{2,t}, \dot{x}_{1,t}, \dot{x}_{2,t}]$ define the target state at time t where x_1, x_2 are the target's longitude and latitude and $\dot{x}_1, \dot{x}_2$ are the corresponding velocities. The target state evolves over time as

$$x_{t+1} = F_t^m x_t + \varepsilon_t \tag{10}$$

where ε_t is the process noise. If the target moves at constant velocity, then $F_t^m = F_t^1$ where

$$F_t^1 = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{11}$$

Otherwise, $F_t^m = F_t^2$ where

$$F_t^2 = \begin{bmatrix} 1 & 0 & \frac{\sin(\Omega_t T)}{\Omega_t} & -\frac{1-\cos(\Omega_t T)}{\Omega_t} \\ 0 & 1 & \frac{1-\cos(\Omega_t T)}{\Omega_t} & \frac{\sin(\Omega_t T)}{\Omega_t} \\ 0 & 0 & \cos(\Omega_t T) & -\sin(\Omega_t T) \\ 0 & 0 & \sin(\Omega_t T) & \cos(\Omega_t T) \end{bmatrix}, \tag{12}$$

where T is the sampling interval ($T = 1$ minute in the simulated and sea trial scenarios) and Ω_t is the turning rate and is defined as

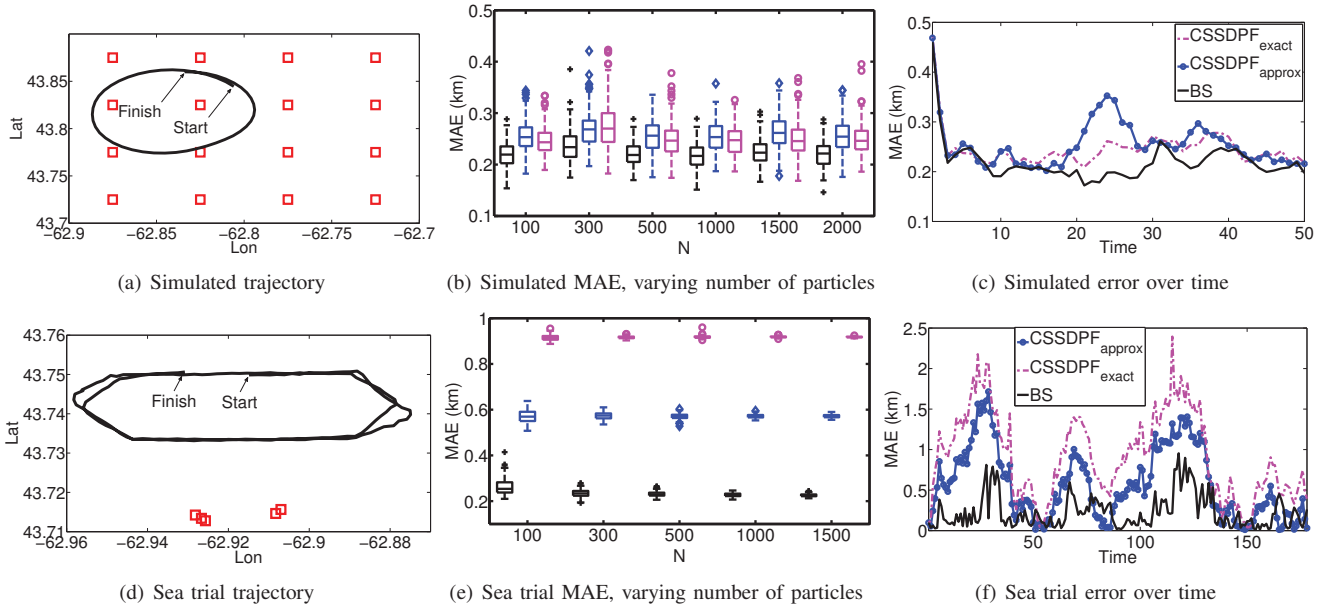
$$\Omega_t = \frac{a}{\sqrt{\dot{x}_{1,t}^2 + \dot{x}_{2,t}^2}} \tag{13}$$

with a being the manoeuvre acceleration parameter (with negative value corresponding to a clockwise turn).

The process noise $\varepsilon_t \in \mathbb{R}^4$ is zero-mean Gaussian with covariance matrix Q given by [7]

$$Q = \sigma_a^2 \begin{bmatrix} \frac{T^3}{3} & 0 & \frac{T^2}{2} & 0 \\ 0 & \frac{T^3}{3} & 0 & \frac{T^2}{2} \\ \frac{T^2}{2} & 0 & T & 0 \\ 0 & \frac{T^2}{2} & 0 & T \end{bmatrix}. \tag{14}$$

At each time step, particles are resampled and disturbed with zero-mean Gaussian noise with variance σ_p^2 for regularization.



(a) Simulated trajectory (b) Simulated MAE, varying number of particles (c) Simulated error over time

(d) Sea trial trajectory (e) Sea trial MAE, varying number of particles (f) Sea trial error over time

Fig. 1. Results for experiments on (a) a simulated trajectory and (d) data from an at-sea trial. Red squares are the sensors. Panels (b) and (e) show boxplots of the MAE as a function of the number of particles for 200 Monte Carlo trials. The box indicates the 25th, 50th (median), and 75th percentiles; outliers are indicated by individual symbols. For each value of N, the group of boxplots corresponds to (from left to right): BS (black $+$), CSSDPF$_{\text{approx}}$ (blue $\Diamond$), and CSSDPF$_{\text{exact}}$ (magenta $\circ$). Panels (c) and (f) show the average absolute position error over time for $N = 1000$ particles.

Note that all three particle filters use this dynamic model and the measurement model defined in Eq. 1 and they only differ in the likelihood evaluation.

A. Simulated track

We have 16 sensors placed in a 4×4 grid tracking a single target over a circular trajectory in the clockwise direction over 50 time steps. All sensors run a particle filter with N particles, and the particles at different nodes remain synchronized by using the same seed for the random number generators. Each sensor receives a variable number of measurements, with an average total of 5 measurements for all sensors at each time step. Unless otherwise specified, for the simulated scenario we use the following parameter values: $\sigma_\theta = 5$, $P_{cv} = 0.1$, $a = -0.001$, $\sigma_a = 10^{-4}$, $\sigma_p = 10^{-6}$, and measurement noise variance $\sigma^2 = 0.0076$ (equivalent to a standard deviation of 5 degrees). Fig. 1 displays the performance comparison. Bootstrap has the best performance but the gap is very small. The two CSSDPF algorithms, on the other hand, have very similar performance. Increasing the number of particles does not seem to improve the performance by any significant amount.

B. Sea trial track

In the sea trial data set [8], five sensors track a single target moving in a roughly clockwise trajectory over 178 time steps. The sensors drift over time and record their locations using on-board GPS. We also have access to the GPS readings aboard the target which we use as ground truth. At each time step (one minute intervals), each sensor node uses an array of underwater acoustic sensors to obtain a collection of bearing measurements. The bearing errors are filtered and the resulting bearing measurements used by the particle filter have errors in the range of -0.087 to 0.087 radians (-5 to 5 degrees). The model parameters for the particle filters are $\sigma^2 = 0.0076$, $P_{cv} = 0.77$, $a = -0.0009$, $\sigma_a = 10^{-3}$ and $\sigma_p = 2 \times 10^{-4}$. These parameters are obtained by fitting the dynamic model to the known target trajectory. Fig. 1 displays the performance comparison. Both CSSDPF algorithms suffer a performance degradation which may be attributed to dynamic model mismatch (i.e. several sharp turns in the trajectory) and non-Gaussian measurement noise. However, what is worth noticing is that CSSDPF$_{\text{exact}}$ actually has the worst performance by a large margin. Furthermore, as Fig. 1(f) shows, CSSDPF$_{\text{exact}}$ has consistently the highest MAE over the entire track. This is quite surprising since we expect that the likelihood function using $R_k(x^i)$ would outperform the one using the approximation $\hat{R}_k$.

IV. Performance Investigation

In this section, we investigate the cause of CSSDPF$_{\text{exact}}$'s performance degradation in the sea trial track. Since the key difference between CSSDPF$_{\text{approx}}$ and CSSDPF$_{\text{exact}}$ is the use of $\hat{R}_k$ versus $R_k(x^i)$ in the likelihood function, we focus on the impact of this approximation.

A. Impact of $R_k(x^i)$

Recall that $R_k(x^i)$ depends on the particular particle x^i. In our first test, we select one single sensor and construct a square grid around it. For each point in the grid, we compute $R_k(x^i)$. Fig. 2 shows the results. We note that the values of $R_k(x^i)$ seem to depend only on the latitude of individual particles as we can clearly distinguish horizontal slices with different colors. In addition, the closer x^i is to the sensor, the smaller $R_k(x^i)$ becomes. This trend is fairly consistent over a variety of sensor locations and grid sizes. Thus we

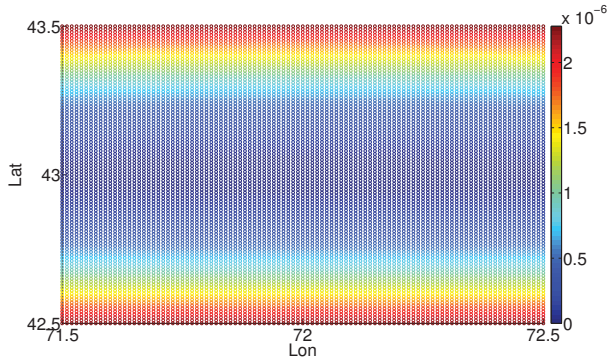


Fig. 2. Values of $R_k(x^i)$ for particles around a single sensor placed at the center ([72, 43]). Each point represents one particle with its color corresponding to the value of $R_k(x^i)$.

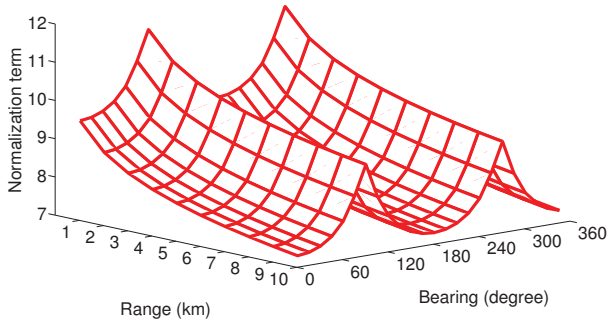
conjecture that $R_k(x^i)$ depends on the geometry between the sensor and the particle.

Consider Eq. 8. We denote the two components, $-\sum_{j=1}^{6} G_j(x^i)\frac{C_j^k}{2R_k(x^i)}$ and $-\frac{\log(2\pi R_k(x^i))}{2}$, as the measurement and normalization terms respectively. For our second test, we fix one sensor and select targets placed 10 km away with bearings from 0 to 360 degrees. For each target, we generate 10 particles with the same bearing at different distance from the sensor (with one particle placed directly at the target position). Then we compute the likelihood and normalization terms for each particle.

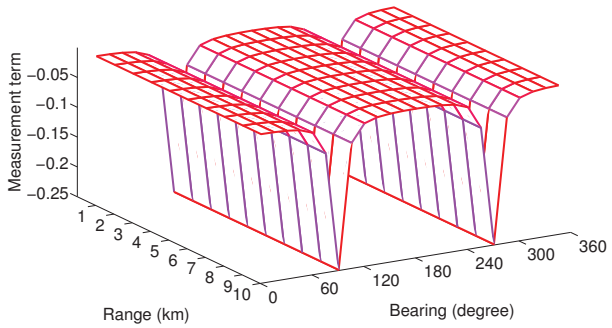
As Fig. 3(a) shows, the normalization term is larger for particles closer to the sensor even though they are actually further away from the target. This bias is persistent over the entire range of bearings and is consistent with our previous observation that $R_k(x^i)$ depends on the geometry between the sensor and the particle. We note that this dependence on distance does not exist in CSSDPF$_{\text{approx}}$ since it uses a constant variance, $\hat{R}_k$, in the likelihood function. On the other hand, as Fig. 3(b) shows, the measurement term has no dependence on the distance between sensor and particle. All particles with identical bearing have the same value of measurement term and this trend is consistent over all bearing angles.

Based on these observations, we conjecture that the normalization term induces a significant bias on the particle weights based on the geometry between the sensors and the particles. More specifically, assume a set of particles have the same bearing as the target relative to the sensor. Their measurement term is equal, but particles closer to the sensor have a bigger normalization term and hence have higher weight. This bias would manifest itself regardless of the actual distance between the target and the sensor, since bearing measurements provide no information about the range of the target. In fact, we could have chosen any random range for the target points in our test and we would have obtained the exact same figures as Fig. 3(a) and Fig. 3(b).

In the sea trial track, since all five sensors are clustered south of the track at all times, the bias from the normalization



(a) Normalization term

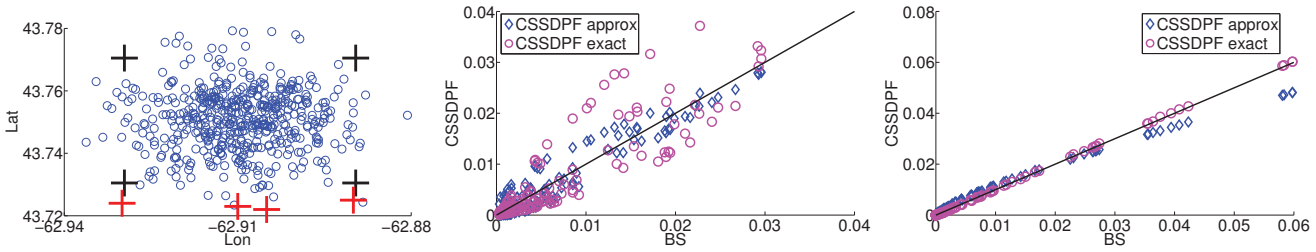


(b) Measurement term

Fig. 3. Values of normalization term (a) and measurement term (b) for particles with given bearing and range relative to a fixed sensor.

terms likely aggregates. Thus, particles closer to the sensors end up with much larger weights compared to particles further away. In the simulated track however, since sensors are spread around the track in a grid, the bias from the normalization terms may cancel out. For instance, consider particles between two sensors A and B. Each particle may gain a large normalization term from A but a small normalization term from B or vice versa. However, a favorable placement of sensors would still not fully nullify the induced bias and the algorithm's performance certainly degrades as a result. Again, we note that CSSDPF$_{\text{approx}}$ does not suffer from this bias since $\hat{R}_k$ (and by extension the normalization term) is constant for all particles.

To validate our conjecture, we conduct a simple test. We place four sensors around a set of particles and compute the particle weights using the true bearings (without measurement noise) with the three likelihood functions (BS, CSSDPF$_{\text{approx}}$ and CSSDPF$_{\text{exact}}$). In the first case (Fig. 4(b)), the four sensors are placed in a cluster south of the target. In the second case (Fig. 4(c)), the four sensors surround the target. For a fair comparison, the particles are the same in both cases. In the first case, particles have very different weights under BS and CSSDPF$_{\text{exact}}$. By simply shifting the sensor positions, the two algorithms end up yielding nearly identical particle weights. On the other hand, CSSDPF$_{\text{approx}}$ is fairly robust to sensor placement. However, under ideal sensor placement, the use of $\hat{R}_k$ does induce a performance penalty as shown by the discrepancy between BS and CSSDPF$_{\text{approx}}$ in Fig. 4(c).



(a) Sensor placement (b) Sensors are clustered (c) Sensors are spread out

Fig. 4. Impact of sensor placements (a) on algorithm performance. The sensors are spread out (black) in (b) and clustered (red) in (c). Panels (b) and (c) compare the particle weights under the three algorithms. The particles are sorted in decreasing weight under bootstrap and only the top 150 particles are shown. Red line is the 45 degree diagonal line.

As an additional test, we take the sea trial track and place 4 sensors around the track (See Fig. 5(a). We limit the number of sensors so the performance is not influenced by additional bearings). We then generate the corresponding true bearing measurements and add white Gaussian noise (with $\sigma = 5$ degrees). Fig. 5(b) compares the MAE of the three algorithms with respect to number of particles. We note a significant performance improvement for all three algorithms. In addition, for small N, CSSDPF$_{\text{exact}}$ actually outperforms CSSDPF$_{\text{approx}}$.

The previous two tests confirm that, depending on the placement of sensors, the normalization term can induce non-negligible bias on the particle weighs and by extension significantly impact the performance of CSSDPF$_{\text{exact}}$.

B. Impact of σ

In order to improve the performance of CSSDPF$_{\text{exact}}$, we need to mitigate the impact from the bias induced by $R_k(x^i)$. Consider Eq. 7. Given x^i and sensor k, $L_1^k(x^i)$ and $L_2^k(x^i)$ are fixed. That leaves σ^2, the variance parameter in the measurement model used by the particle filter.

To understand the impact of σ on the algorithm's performance, we do the following test. We take the sea trial track and generate two identical sets of noise-free bearing measurements. The first set is then corrupted by additive zero-mean Gaussian noise (with standard deviation of 5 degrees) and the second set is corrupted by uniformly random noise (within range of $[-5, 5]$ degrees). As a baseline, we also use the filtered noisy bearings (with error not exceeding 5 degrees) from the sea trial dataset. The three algorithms assume that the measurement noise is Gaussian with variance σ^2. Fig. 6 displays the resulting performance for different σ. For all three scenarios, the performance of CSSDPF$_{\text{exact}}$ improves considerably by decreasing σ while the other two algorithms do not appear affected. For $\sigma = 1, 2$, CSSDPF$_{\text{exact}}$ achieves similar performance as bootstrap filter.

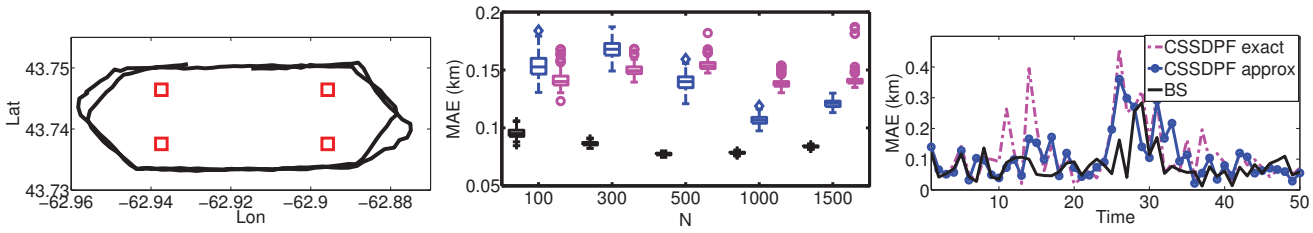
This performance improvement can be explained as follows. For smaller σ, there is a higher penalty in the measurement term for particles that explain poorly the received measurements (i.e. bearing of the particle relative to the sensor is very different from the received bearing). Thus, as we reduce σ, particles with large bearing errors still get a high normalization term if they are close to the sensors, but have reduced weights as a result of increasing penalty from the measurement term.

We conduct one final test in which we evaluate the three algorithms' performance on the simulated track. Recall that the measurements in the simulated data set are corrupted by zero-mean Gaussian noise with standard deviation of 5 degrees. Again, the three algorithms model the measurement noise as being zero-mean Gaussian with variance σ^2. Fig. 7 shows the performance of the three algorithms with respect to different σ. In this case, rather than seeing an improvement, the performance of CSSDPF$_{\text{exact}}$ actually becomes much worse when σ is small whereas the three algorithms achieve similar performance for $\sigma = 4, 5$.

We conjecture that the difference in sensor placements between the simulated and sea trial tracks is the main cause of this discrepancy. In the simulated track, the sensors are already spread and the bias is mitigated. Thus, by setting σ small, we have a model mismatch and the filter's performance degrades. However, CSSDPF$_{\text{approx}}$ is less sensitive to model mismatch. For the sea trial track, reducing σ causes the particle filter to reject more particles that are close to the sensors (high normalization term) but have relatively large bearing errors (high penalty in measurement term). In other words, when we reduce σ, the higher penalty in measurement term can compensate the bias from normalization term. However, a balance is clearly required. Reducing σ to the limit would heavily penalize all particles and effectively reject all received measurements. The particle filter then depends entirely on the dynamic model and proper initialization, which would no doubt lead to performance degradation. Our tests show that, for the simulated and sea trial tracks, CSSDPF$_{\text{exact}}$ can achieve reasonable performance for $\sigma = 3$.

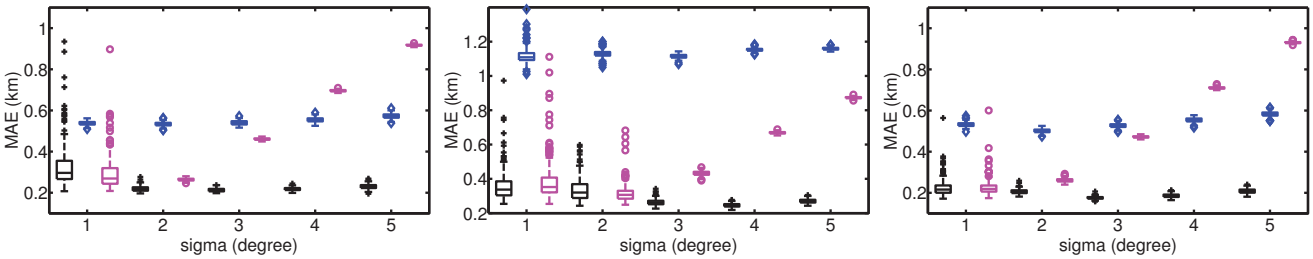
V. Conclusion

In this paper, we investigate the performance of CSSDPF for single-target bearing-only tracking. Our work shows that, as a result of the approximate likelihood function, CSSDPF$_{\text{exact}}$ has a bias induced by the geometry between the sensors and the particles where, given the same bearing, particles closer to the sensor have higher weights regardless of their actual distance to the target. This bias can significantly degrade the algorithm's performance when the sensors are not suitably placed (i.e. clustered so the bias aggregates).



(a) Sea trial trajectory with sensors in a grid (b) Sea trial MAE, varying number of particles (c) Sea trial error over time

Fig. 5. Results for experiments on sea trial trajectory with sensors (red squares) spread around the track. Panel (b) shows boxplots of the MAE as function of the number of particles for 200 Monte Carlo trials. The box indicates the 25th, 50th (median), and 75th percentiles; the outliers are indicated by individual symbols. For each value of N, the group of boxplots corresponds to (from left to right): BS (black $+$), CSSDPF$_{\text{approx}}$ (blue $\Diamond$), and CSSDPF$_{\text{exact}}$ (magenta $\circ$). Panels (c) shows the average MAE over time for $N = 1000$ particles.



(a) Sea trial data (b) Gaussian noise (c) Uniform noise

Fig. 6. Results for experiments on sea trial track. Bearing measurements are corrupted with filtered noise value from the sea trial data set (with error not exceeding 5 degrees) (a), zero-mean Gaussian noise with standard deviation of 5 degrees (b) or uniform noise within range $[-5, 5]$ degrees (c). The three panels show boxplots of the MAE as function of noise standard deviation, σ, of the filter measurement model for 200 Monte Carlo trials. For each value of σ, the group of boxplots corresponds to (from left to right): BS (black $+$), CSSDPF$_{\text{approx}}$ (blue $\Diamond$), and CSSDPF$_{\text{exact}}$ (magenta $\circ$).

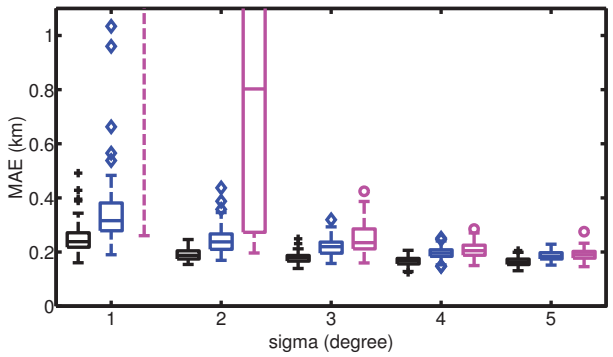


Fig. 7. Performance comparison for simulated track with respect to different σ, $N = 1000$

However, when the bias is present, CSSDPF$_{\text{approx}}$ [4] is shown to be much more robust as it does not have the geometry-related bias. Interesting future work may include developing a method to compensate the bias by extracting information about sensor location relative to the particle.

REFERENCES

[1] O. Hlinka, F. Hlawatsch, and P. Djuric, "Distributed particle filtering in agent networks: A survey, classification, and comparison," *IEEE Signal Processing Mag.*, vol. 30, no. 1, pp. 61–81, Jan 2013.

[2] A. Mohammadi and A. Asif, "A constraint sufficient statistics based distributed particle filter for bearing only tracking," in *IEEE Intl. Conf. on Communications*, June 2012, pp. 3670–3675.

[3] A. Mohammadi, "Distributed Implementation of the Particle Filter with Performance Bounds," Ph.D. dissertation, York Univ., Toronto, ON, 2013.

[4] J.-Y. Yu, M. J. Coates, M. G. Rabbat, and S. Blouin, "A distributed particle filter for bearings-only tracking on spherical surfaces," Elect. and Comput. Eng., McGill Univ., Montréal, QC, Tech. Rep., Oct. 2014.

[5] D. Peters, "Flatlanders in space: Three-dimensional transformations of two-dimensional data," Defense Research and Development Canada, Dartmouth, NS, Tech. Rep. 2005-259, Dec. 2005.

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

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

[8] J.-Y. Yu, D. Üstebay, S. Blouin, M. G. Rabbat, and M. J. Coates, "Distributed underwater acoustic source localization and tracking," in *Proc. Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, USA, Nov. 2013.