Distributed particle filter for bearing-only tracking

Jun Ye Yu

May 2, 2018

1 Introduction

In this report we present four distributed particle filters for single-target bearing-only tracking. The first filter factorizes the joint log-likelihood function using six sufficient statistics that can be computed using distributed summation. The second filter uses likelihood consensus to encode the particle log-likelihoods using a small number of basis functions. The third filter constructs a graph over all particles and the Eigenvectors of the resulting Laplacian matrix are used to encode the particle log-likelihood using a minimal number of coefficients. Finally, the fourth filter groups all particles into clusters and computes the cluster joint likelihood. The individual particle weights are then recovered via convex minimization. For the remainder of the report, we refer to the four particle filters as **CSSpf** [1], **LCpf** [2], **LApf** [3] and **Clusterpf** [4] respectively. We also include the centralized *bootstrap particle filter* (**BSpf**) as baseline.

The remainder of the report is organized as follows. Sec. 2 defines the tracking problem. Sec. 3 presents the particle filters. Sec. 4 compares the filters' performance and Sec. 6 concludes the report.

2 Problem statement

A network of S sensors collaboratively track a single moving target over time. The sensors have fixed position $[x_s, y_s], s = 1...S$. The target state at time k is modeled as $X(k) = [x_t(k), y_t(k), \dot{x}_t(k), \dot{y}_t(k)]$ where $x_t(k)$, $y_t(k)$ are the target position and $\dot{x}_t(k)$, $\dot{y}_t(k)$ are its velocity.

At time k, the target transitions to new state $X(k)$ with probability $f(X(k)|X(k-1))$ which depends on the target dynamic model. Each sensor s also receives a noisy measurement $z_s(k)$ with likelihood $f(z_s(k)|H_s(X(k))$ where $H_s(\cdot)$ is the (possibly sensor-dependent) measurement function. The sensors have unity target detection probability and receive no clutter measurement.

The objective is to estimate the posterior target density $p(X(k)|z_1(k), ..., z_S(k))$ at each time step k.

3 Distributed particle filters for bearing-only tracking

In a particle filter, the posterior target density is modeled using a set of N particles with normalized weights $\{X_i(k), w_i(k)\}_{i=1}^N$, and the objective is to recursively estimate the posterior particle weights. This in turn requires the computation of joint log-likelihood:

$$w_i(k) \propto \log(f(z_1(k), ...z_S(k)|X_i(k))) = \sum_{s=1}^{S} \log(f(z_s(k)|X_i(k))) \tag{1}$$

where measurements from different sensors are considered to be conditionally independent.

For the remainder of this section, we present four distributed particle filters which compute the joint log-likelihood in different manners. We omit time step indice k where there is no ambiguity. For convenience of notation, let $\gamma_s = [\log(f(z_s|X_1), ... \log(f(z_s|x_N))]^T$ denote the column vector of the N particle log-likelihoods at sensor s. Similarly, let $\gamma = [\log(f(z_1, ..., z_S|X_1), ... \log(f(z_1, ..., z_S|x_N))]^T$ denote the column vector of joint log-likelihood.

3.1 Constraint sufficient statistics particle filter

In the CSSpf, the likelihood function is approximated as follows [1]

$$\log(f(z_s|X) \approx \sum_{j=1}^{6} G_{s,j} F_j(X) \tag{2}$$

$$\begin{aligned}
G_{s,1} &= (Z_{s,\theta})^2/R_\theta \\
G_{s,2} &= \cos^2(Z_{s,\theta})/R_\theta \\
G_{s,3} &= \sin^2(Z_{s,\theta})/R_\theta \\
G_{s,4} &= \sin(Z_{s,\theta})\cos(Z_{s,\theta})/R_\theta \\
G_{s,5} &= Z_{s,\theta}\cos(Z_{s,\theta})/R_\theta \\
G_{s,6} &= Z_{s,\theta}\sin(Z_{s,\theta})/R_\theta \\
Z_{s,\theta} &= y_s \sin(z_s) - x_s \cos(z_s) \\
R_\theta &= E((x_t - x_s)^2 + (y_t - y_s)^2)(1 - \exp^{-2\sigma_\theta^2})/2
\end{aligned}$$

$$\begin{aligned}
F_1(X) &= 1 \\
F_2(X) &= x_t^2 \\
F_3(X) &= y_t^2 \\
F_4(X) &= -2x_t y_t \\
F_5(X) &= 2x_t \\
F_6(X) &= -2y_t
\end{aligned}$$

where the expectation term in $R_\theta(\cdot)$ is taken over all particles X_i. The functions $F_j(X)$ depend only on target state X and are known to all sensors. The sufficient statistics $G_{s,j}$ depend only on local information from sensor s. In other words, we approximate the log-likelihood function by the weighted combination of six basis functions $F_j(X)$ with corresponding weight coefficients $G_{s,j}$.

This formulation leads to the following approximate joint log-likelihood function

$$\log(f(z_1, ..., z_S|X) \approx \sum_{j=1}^{6} F_j(X) \left(\sum_{s=1}^{S} G_{s,j} \right) \tag{3}$$

where the summation terms $\left(\sum_{s=1}^{S} G_{s,j}\right)$ can be interpreted as the global sufficient statistics. These global sufficient statistics can be computed in a distributed manner by running six consensus algorithms in parallel. The six basis functions of CSSpf are specifically tailored for bearing-only tracking. For other measurement model, re-derivation of the filter is required.

3.2 Likelihood consensus particle filter

In LCpf, we approximate the log-likelihood function as follows:

$$\log(f(z_s|X)) \approx \sum_{j=1}^{J} \alpha_{s,j} \beta_j(X) \tag{4}$$

where $\beta_j(X)$ is the j^{th} sensor-independent basis function and $\alpha_{s,j}$ is the corresponding coefficient that encompasses all the local information of sensor s. We note that the CSSpf can be seen as a special case of LCpf and that $\alpha_{s,j}$ is analogous to $G_{s,j}$ and $\beta_J(X)$ is analogous to $F_j(X)$.

For each sensor s, we construct the column vector $\gamma_s = [\log(f(z_s|X_1)), ... \log(f(z_s|X_N))]^T$ where T denotes the transpose operation. Given the N particles X_i, we construct the $N \times J$ matrix Φ as follows:

$$\Phi = \begin{pmatrix} \beta_1(X_1) & ... & \beta_J(X_1) \\ ... & ... & ... \\ \beta_1(X_N) & ... & \beta_J(X_N) \end{pmatrix} \tag{5}$$

We seek a set of coefficients $\alpha_s = [\alpha_{s,1}, ...\alpha_{s,J}]^T$ such that the approximation error $\gamma_s - \Phi\alpha_s$ is minimized. Using the least-square approach yields the coefficients vector

$$\alpha_s = (\Phi^T\Phi)^{-1}\Phi^T\gamma_s \tag{6}$$

As in the case of CSSpf, computing the joint log-likelihood amounts to running J consensus algorithms in parallel and computing $\sum_{s=1}^{S} \alpha_{s,j}, j = 1...J$.

3.3 Laplacian approximation particle filter

In LApf, we consider each particle X_i a vertex on a graph. The *Delaunay triangulation* (DT) is used to generate the graph edges. The resulting Laplacian matrix is used to construct a transformation that encodes particle log-likelihoods using a minimal number of coefficients.

Let L denote the Laplacian matrix of the DT graph. The eigenvectors of L are used to transform particle log-likelihoods into Laplacian domain. Using all N eigenvectors is obviously counterproductive since we incur the computational overhead of eigendecomposition and achieve no reduction in communication overhead (i.e., we still have to aggregate N coefficients).

Assume that $m \leq N$ eigenvectors are used as the basis of transformation and let E_m denote the resulting matrix where each column is an eigenvector. We compute the local coefficients at sensor s as follows:

$$\alpha_s = E_m^T\gamma_s \tag{7}$$

The global coefficients are the summation of local coefficients across all S sensors: $\alpha = \sum_s \alpha_s$. Finally, the approximate joint log-likelihood can be computed as follows:

$$\hat{\gamma} = E_m\alpha = E_m \sum_s \alpha_s \tag{8}$$

Since the particle log-likelihoods can be considered as a smooth signal over the graph (i.e., particles close to each other have similar log-likelihoods), most of their energy should be concentrated in the coefficients corresponding to lower frequency basis vectors. In other words, we should retain the m eigenvectors corresponding to the m smallest eigenvalues.

We note that LApf is similar to CSSpf in that both filters encode the particle log-likelihoods directly using a minimal number of coefficients. However, for LApf, all sensors must be synchronized so that they have the same particles; otherwise they would obtain a different particle graph and by extension different eigenvectors for the encoding. The CSSpf has no such restrictions.

3.4 Clustering particle filter

In Clusterpf, the particles are grouped into C clusters based on their position. The sensors reach consensus on the cluster log-likelihoods rather than individual particle log-likelihoods. For $C \ll N$, significant reduction in communication overhead can be achieved.

We follow the approach in [2]. The log-likelihood of each cluster is equal to the aggregate log-likelihoods of its constituent particles. Let γ^c denote the joint log-likelihood of the clusters after consensus. Let A_C denote the $C \times N$ cluster assignment matrix where $A_C(i,j) = 1$ if particle j belongs to cluster i.

In order to recover the individual particle joint log-likelihoods γ, we again construct DT graph, compute the Laplacian matrix L, and then solve the following convex minimization problem:

$$\underset{\gamma}{\text{minimize}} \quad \gamma^T L\gamma \tag{9}$$

$$\text{subject to} \quad A_C\gamma = \gamma^c \tag{10}$$

In other words, we seek to assign particle log-likelihood values that are smooth with respect to particle proximity while ensuring the aggregate particle values are equal to the cluster value. As in the case of LApf, the Clusterpf requires that all sensors have the same particles.

3.5 Computational overhead

In this section we compare the computational overhead of the four filters. More specifically, we compare the overhead for particle log-likelihood computation.

Consider first CSSpf. Each sensor computes the six local sufficient statistics with complexity $O(6N)$. These statistics are then aggregated via distributed consensus with complexity $O(6S*NGossip)$. Finally, the log-likelihoods are computed at all sensors using the global sufficient statistics with complexity $O(6S*N)$. Thus, the overall complexity of CSSpf is thus $O(6S*N+S*NGossip+6S*N)$. Since $N > NGossip$ in general, the complexity is dominated by $\underline{O(6S*N)}$.

Consider next LCpf. Let J denote the number of basis functions. Each sensor needs to generate a $N \times J$ matrix to compute the local coefficients. Then J coefficients are aggregated via distributed consensus over $NGossip$ iterations. The log-likelihoods are finally computed from the global coefficients. The overall complexity is thus $O(S*N*J+S*NGossip*J+S*N) \subset \underline{O(S*N*J)}$.

Consider next LApf. The Delaunay triangulation for graph construction has complexity $O(N\log(N))$. The eigenvalue decomposition has complexity $O(N^3)$. Assume m eigenvectors are used to decode the local log-likelihoods. Then m scalars are aggregated via distributed consensus. The joint log-likelihoods are then recovered from the m aggregate scalars. The overall complexity is thus $O(S*N\log(N)+S*N^3+S*m*N+S*m*NGossip+S*m*N) \subset \underline{O(S*N^3)}$.

Finally consider Clusterpf. Particle clustering has complexity $O(N*C*4*I)$ where I is the number of clustering iterations (with default value of 100) and the constant 4 is the target state dimension. C cluster log-likelihoods are then aggregated across all sensors. We again have Delaunay triangulation for graph construction. The log-likelihood is recovered via convex minimization with complexity $O(\sqrt{N})$. The overall complexity is thus $O(S*N*C*4*I+S*C*NGossip+S*N\log(N)+S\sqrt{N}) \subset O(S*N*C*I+S*N\log(N))$. Since we can assume that $C*I > \log(N)$ (i.e., for $N = 10^4$, $C = 4$, $C*I > \log(N)$ for $I > 2$), the complexity is dominated by $\underline{O(S*N*C*I)}$

Overall, CSSpf has the lowest computational overhead whereas LApf has the highest overhead due to the eigenvalue decomposition. For $6 < J \leq N$, LCpf has the second lowest overhead.

4 Performance evaluation

4.1 Simulation setup

The target state evolves over time following a discrete-time model:

$$X(k+1) = F(X(k)) + \xi(k) \tag{11}$$

where $F(X(k))$ is the dynamic model and $\xi(k)$ is the zero-mean Gaussian process noise. The simulated target randomly switches between two different motion models: constant velocity with probability $P_{cv} = 0.05$ and coordinated turn with probability $1 - P_{cv} = 0.95$.

For constant velocity, we have

$$F(X(k)) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

For coordinated turn, we have

$$F(X(k)) = \begin{bmatrix} 1 & 0 & \frac{\sin(\Omega)}{\Omega(k)} & -\frac{1-\cos(\Omega(k))}{\Omega(k)} \\ 0 & 1 & \frac{1-\cos(\Omega(k))}{\Omega(k)} & \frac{\sin(\Omega(k))}{\Omega(k)} \\ 0 & 0 & \cos(\Omega(k)) & -\sin(\Omega(k)) \\ 0 & 0 & \sin(\Omega(k)) & \cos(\Omega(k)) \end{bmatrix} \tag{13}$$

where $\Omega(k)$ is the turning rate

$$\Omega(k) = \frac{a}{\sqrt{\dot{x}^2(k) + \dot{y}^2(k)}} \tag{14}$$

with $a = 0.5$ being the maneuver acceleration parameter.

All sensors receive noisy bearing measurements (in radians) from the target.

$$H_s(X(k)) = \arctan 2\left(\frac{x_t - x_s}{y_t - y_s}\right) + \eta(k) \tag{15}$$

An alternate measurement model we consider is the range measurements (in km).

$$H_s(X(k)) = \sqrt{(x_t - x_s)^2 + (y_t - y_s)^2} \tag{16}$$

The process and measurement noises $\xi(k)$ and $\eta(k)$ have covariance matrices Q and R respectively.

$$Q = \sigma_a^2 \begin{bmatrix} \frac{1}{3} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{3} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 1 & 0 \\ 0 & \frac{1}{2} & 0 & 1 \end{bmatrix} \tag{17}$$

$$R_{\text{bearing}} = \sigma_\theta^2 \tag{18}$$

$$R_{\text{range}} = \sigma_r^2 \tag{19}$$

where $\sigma_a = 10^{-4}$, $\sigma_\theta = 0.0873$ rad $= 5$ degree, and $\sigma_r = 5$ km.

4.2 Algorithm setup

All particle filters use a total of $N = 500$ particles. At time step 1, we generate the initial particles using the true target state: $X_i(1) \sim \mathcal{N}(X(1), R_{\text{initial}})$ with $R_{\text{initial}} = \text{diag}([0.5^2, 0.5^2, 0.05^2, 0.05^2])$.

For LCpf, we use a set of basis functions involving all permutations of $x_t^i y_t^j$ with $0 \le i, j \le d$ where d is some user-specified max degree. (i.e., For $d = 2$, the basis functions would be $\beta_1(X) = x_t^0 y_t^0 = 1, \beta_2(X) = x_t^0 y_t^1 = y_t, ..., \beta_9(X) = x_t^2 y_t^2$.) Note that, due to our choice of basis functions, all particles must remain synchronized across all sensors as in the case of LApf and Clusterpf. For LApf, we construct a DT graph and retain $m < N$ eigenvectors as the basis of Laplacian transformation. For Clusterpf, all particles are grouped into C clusters and a DT graph is constructed to recover individual particle weights.

The random number generators are synchronized to ensure that the particles remain the same across sensors. Distributed summation is performed using gossip algorithms. At each time step, we perform *NGossip* gossip iterations. At each gossip iteration, each sensor i broadcasts its local values G_i, receives broadcasts from its neighbors, and then updates its local values as a weighted aggregate:

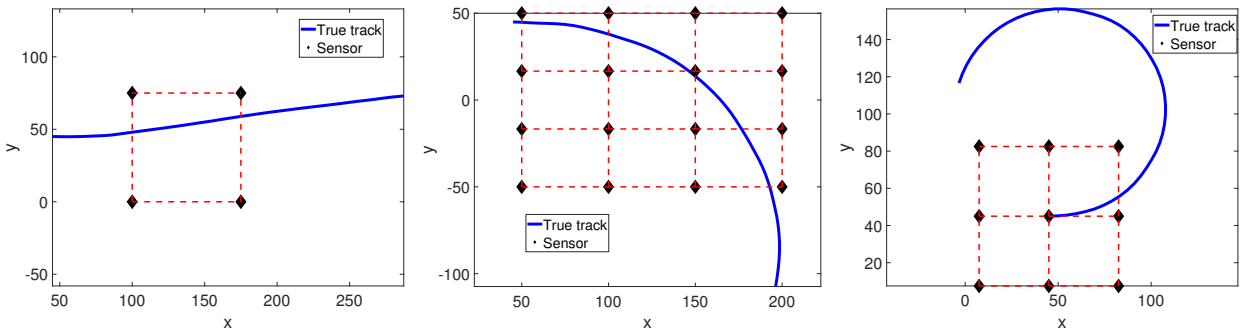
$$G_{i,\text{new}} = w_{ii} G_{i,\text{old}} + \sum_{j \in N_i} w_{ij} G_{j,\text{old}} \tag{20}$$

$$w_{ij} = \begin{cases} \frac{1}{1+max(d_i,d_j)} & j \in N_i \\ 1 - \sum_{j \in N_i} w_{ij} & i = j \\ 0 & j \notin N_i \end{cases} \tag{21}$$

where N_i denotes the set of neighboring sensors of sensor i, $d_i = |N_i|$, and Metropolis weight is used for the update. After a total of *NGossip* iterations, a max consensus algorithm is run to ensure all sensors obtain the same values.

In our codes, we do not implement a loop for the gossip iterations. Instead we define an update matrix W where $W(i, j) = w_{ij}$. Therefore, given initial values $G_{\text{initial}} = [G_1, ..., G_S]^T$, the final values can be easily computed as

$$G_{\text{final}} = W^{NGossip} G_{\text{initial}} \tag{22}$$



(a) Track 1 (b) Track 2 (c) Track 3

Figure 1: Target tracks (blue curve) and sensor positions (black diamond). Sensors connected by red dashed lines are within broadcast range of each other.

In the remainder of the section, we run a number of Monte Carlo simulations to evaluate the performance of the four filters. The track remains the same in each trial; but the measurements differ. We evaluate the algorithms' performances using the *root mean squared error* (RMSE) of position estimate and discrepancy of normalized particle weights, $||w_{\text{true}} - w_{\text{approx}}||_2$, averaged over all time steps and all trials. The first metric is self-explanatory and the second metric is meant to evaluate the impact of different filters' encoding scheme on the overall tracking performance.

4.3 Performance comparison between filters

In this section, we compare the four filters against each other. We use a centralized bootstrap particle filter as baseline. Finally, we include a modified version of LCpf where we use the Gram-Schmidt process to obtain an orthonormal encoding matrix Φ' (i.e., $||\Phi'(:,i)||_2 = 1, i = 1, ...J$). This filter is referred to as LCpf-GS. We consider three different test tracks shown in Fig. 1.

Fig. 2 shows the average RMSE and weight error with respect to communication overhead per sensor per time step for all algorithms for track 1. Note that we do not consider the CSSpf for range-only tracking.

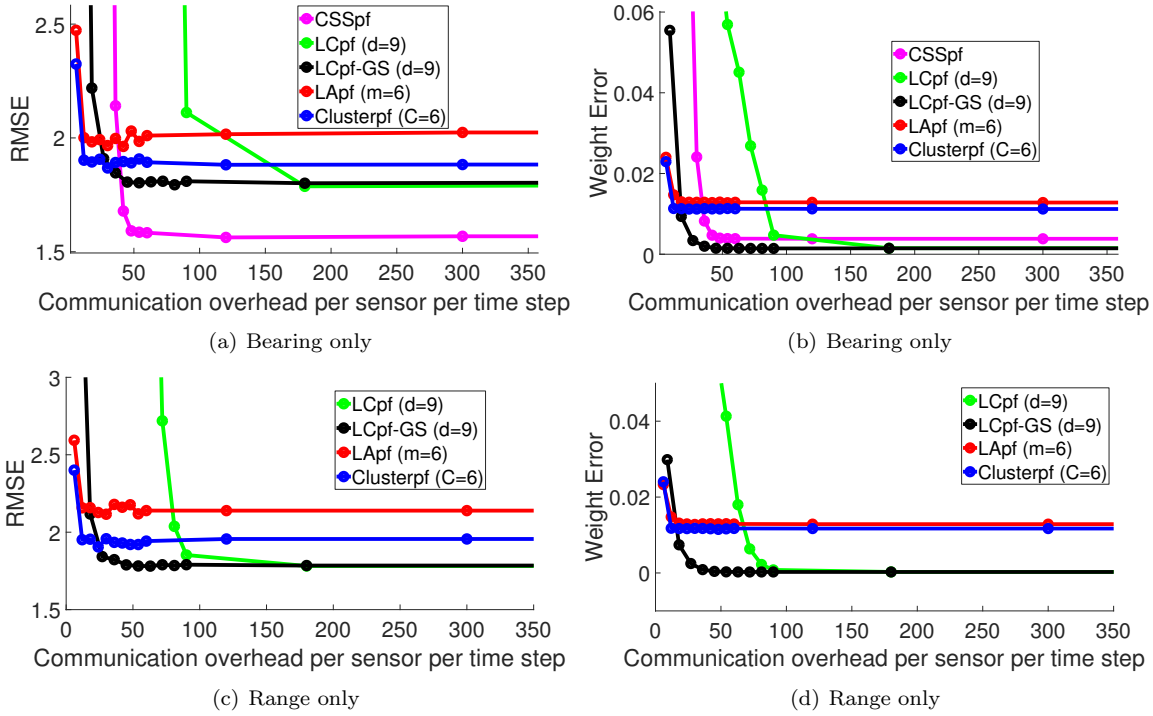
Consider first the bearing-only scenario. For overhead < 25, Clusterpf and LApf have the lowest RMSE. Note that these two filters have low RMSE even at $NGossip = 1$ (i.e., communication overhead = 1*6 = 6). For overhead > 50, CSSpf has the best performance by a significant margin. The LCpf requires the highest communication overhead to achieve adequate tracking performance. For all filters, we can notice a correspondence between weight error and RMSE (i.e., higher error equals higher RMSE) as expected; but two filters having the same weight error do not necessarily have the same RMSE. For instance, for overhead = 100, LCpf has low weight error on-par with that of CSSpf, yet its RMSE is significantly higher.

Consider next the range-only scenario. We observe the same trends as in the bearing-only scenario; although the LCpf requires significantly lower communication overhead to achieve robust tracking performance.

Fig. 3 shows the tracking results for track 2. For overhead < 200, LApf and Clusterpf have the lowest RMSE. The LCpf-GS achieves similar RMSE as LAPf once the overhead exceeds 200. For LCpf, over 100 Gossip iterations are required to achieve adequate tracking performance. Again, for each filter, there is a clear correspondence between RMSE and weight error. These trends are consistent in both bearing-only and range-only tracking.

Fig. 4 shows the tracking results for track 3. Again, LApf and Clusterpf achieve robust tracking performance even with very low communication overhead. LCpf achieves good performance for $NGossip \geq 50$. In bearing-only tracking, CSSpf has poor performance even for $NGossip \geq 100$ even though its weight error is lowest for $NGossip \geq 30$.

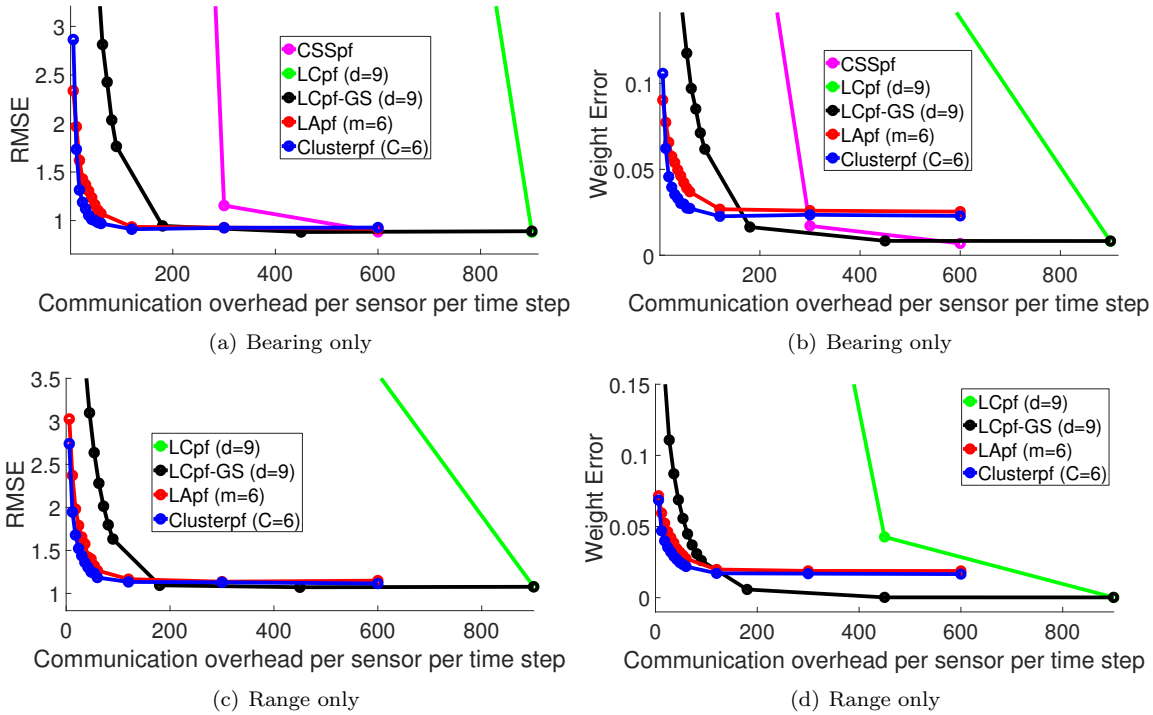
In all three test tracks, the LApf and Clusterpf yield robust tracking performance even with very low



(a) Bearing only (b) Bearing only

(c) Range only (d) Range only

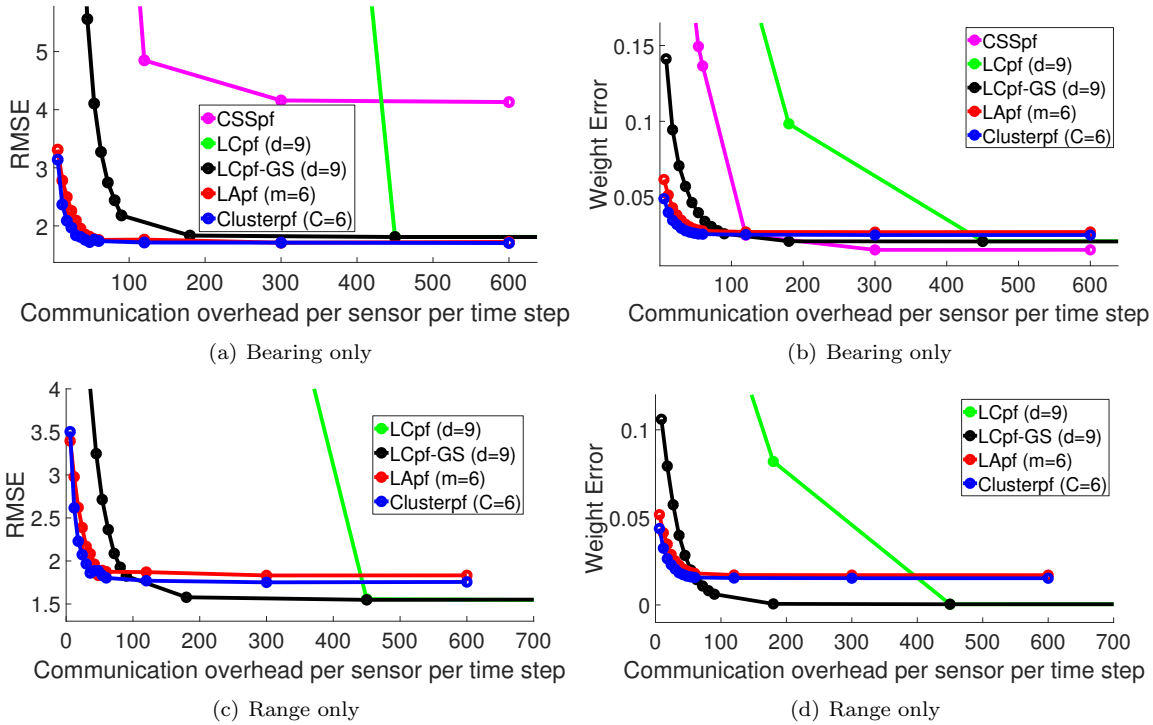
Figure 2: Average RMSE and particle weights error with respect to total communication overhead per time step for track 1 ($N = 500$, 200 Monte Carlo trials).



(a) Bearing only (b) Bearing only

(c) Range only (d) Range only

Figure 3: Average RMSE and particle weights error with respect to total communication overhead per time step for track 2 ($N = 500$, 200 Monte Carlo trials).



(a) Bearing only

(b) Bearing only

(c) Range only

(d) Range only

Figure 4: Average RMSE and particle weights error with respect to total communication overhead per time step for track 3 ($N = 500$, 200 Monte Carlo trials).

communication overhead. The other filters may outperform LApf and Clusterpf albeit at the cost of higher overhead. In particular, the LCpf requires significantly higher communication overhead to achieve similar RMSE as the other filters.

Fig. 5 shows the average runtime of the filters. We omit the results for tracks 1 and 2 since the trends are similar. The LApf has the highest runtime due to the overhead of eigendecomposition. The Clusterpf has the second highest runtime. The other filters have similar low runtime. We note that increasing $NGossip$ does not increase total runtime due to our implementation of the gossip algorithms.

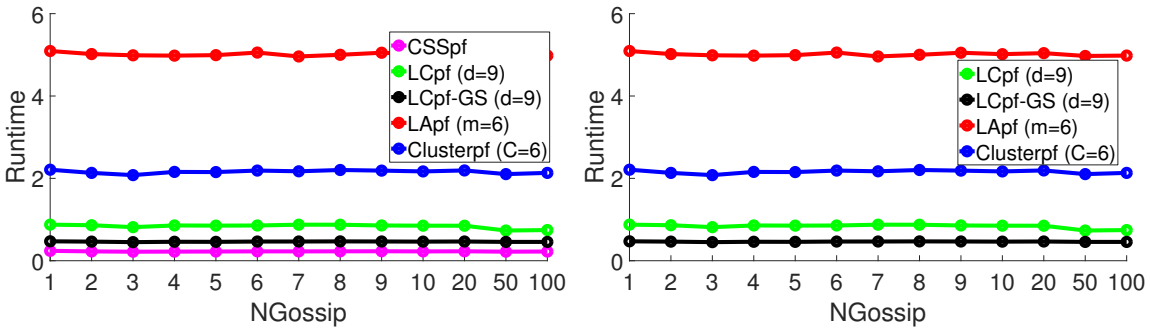
5 Bounding the error propagation

In this section, we seek to bound the error in the presented particle filters. Let $\hat{\gamma}(x_i)$ denote the approximate joint particle log-likelihood for particle x_i and let $\gamma(x_i)$ denote its true joint particle log-likelihood. Our objective is to derive an upper bound for $\frac{||\hat{\gamma}(x_i)-\gamma(x_i)||}{||\gamma(x_i)||}$ for all particles.

All distributed filters presented in this paper follow the same general steps:

1. Each sensor computes local particle log-likelihoods
2. Each sensor encodes the local log-likelihoods into m coefficients
3. Gossip and max-consensus algorithms are used to compute the m aggregate coefficients.
4. Approximate joint log-likelihoods are recovered from the aggregate coefficients.

Therefore, the discrepancy between true log-likelihoods $\gamma(x_i)$ and the final approximate value $\hat{\gamma}(x_i)$ comes from two sources: the encoding error and the gossiping error. The first error is the discrepancy between true



(a) Bearing only (b) Bearing only

Figure 5: Average total runtime for track 3

log-likelihoods and its reconstructions using m coefficients. The second error occurs when sensors do not obtain the true aggregate coefficient values after a finite number of gossiping iterations.

Consider the encoding error first. Let $\gamma_m(x_i)$ denote the approximate log-likelihood recovered from m coefficients. More specifically, we have

$$\gamma_m^{\text{LC}} = \Psi^{\text{LC}} \alpha^{\text{LC}} \tag{23}$$

$$\gamma_m^{\text{LC-GS}} = \Psi^{\text{LC-GS}} \alpha^{\text{LC-GS}} \tag{24}$$

$$\gamma_m^{\text{LA}} = \Psi^{\text{LA}} \alpha^{\text{LA}} \tag{25}$$

$$\gamma_m^{\text{LC}} = \Psi^{\text{LC}} \alpha^{\text{LC}} \tag{26}$$

$$\gamma_m^{\text{Cluster}} = \arg\min_{\gamma} \gamma^T L \gamma \quad \Phi^{\text{Cluster}} \gamma = \alpha^{\text{LC}} \tag{27}$$

$$\gamma_m^{\text{CSS}} = \Psi^{\text{CSS}} \alpha^{\text{CSS}} \tag{28}$$

We assume that the following bound on encoding error holds for all particles x_i

$$|\gamma_m(x_i) - \gamma(x_i)| \leq \delta_m |\gamma(x_i)| \tag{29}$$

The ratio term $\delta_m \leq 1$ depends on the specific filter and the number of coefficients m. We can rewrite Eq. (29) to yield

$$(1 - \delta_m)|\gamma(x_i)| \leq |\gamma_m(x_i)| \leq (1 + \delta_m)|\gamma(x_i)| \tag{30}$$

Next, we consider the gossiping error. Let $\hat{\alpha}$ denote the coefficients obtained after gossip and max-consensus iterations. This gives us the approximate log-likelihoods $\hat{\gamma}$. We seek to upper bound the following error ratio

$$\frac{|\gamma_m(x_i) - \hat{\gamma}(x_i)|}{|\gamma_m(x_i)|} \leq \delta_{Gossip} \tag{31}$$

First, we assume that the following bound applies for all coefficients:

$$\frac{|\hat{\alpha}_j - \alpha_j|}{|\alpha_j|} \leq \beta \tag{32}$$

where β depends on the number of gossip iterations. More specifically, let $\hat{\alpha}(t)$ be the coefficient obtained after t gossip iterations and max-consensus iterations and let α_j denote the true coefficient value. Let S denote the total number of sensors, let A denote the $S \times S$ averaging matrix used in the gossiping algorithm and let ρ_A denote its second largest eigenvalue in modulus. From [5], we have

$$\frac{|\hat{\alpha}_j(t) - \alpha_j|}{|\alpha_j|} \leq \beta \tag{33}$$

as long as

$$t \geq \frac{1.5\log(S) + \log(\frac{S-1}{\beta})}{\log(1/\rho_A)} \tag{34}$$

Consider LApf, LCpf, LCpf-GS and CSSpf. Let Ψ_i denote the i^{th} row of encoding matrix Ψ and let Ψ_{ij} denote the j^{th} entry of row vector Ψ_i. We can derive an upper bound for the gossiping error as follows:

$$\begin{aligned}
|\hat{\gamma}(x_i) - \gamma_m(x_i)| &= \left| \sum_{j=1}^{m} \Psi_{ij}(\hat{\alpha}_j - \alpha_j) \right| \\
&\leq \sum_{j=1}^{m} |\Psi_{ij}(\hat{\alpha}_j - \alpha_j)| \\
&\leq \sum_{j=1}^{m} |\Psi_{ij}| \quad |\hat{\alpha}_j - \alpha_j| \\
&\leq \beta \sum_{j=1}^{m} |\Psi_{ij}| \quad |\alpha_j|
\end{aligned} \tag{35}$$

where the first inequality follows from triangle inequality and the last inequality follows from Eq.31.

Dividing both sides by $|\gamma_m(x_i)|$ yields:

$$\begin{aligned}
\frac{|\hat{\gamma}(x_i) - \gamma_m(x_i)|}{|\gamma_m(x_i)|} &\leq \frac{\beta \sum_{j=1}^{m} |\Psi_{ij}| \quad |\alpha_j|}{|\gamma_m(x_i)|} \\
&\leq \frac{\sum_{j=1}^{m} \beta\, |\Psi_{ij}|\, |\alpha_j|}{|\gamma(x_i)(1-\delta_m)|} \\
&\leq \frac{\sum_{j=1}^{m} \beta\, |\Psi_{ij}|\, |\alpha_j|}{\min_{x_i} |\gamma(x_i)(1-\delta_m)|}
\end{aligned} \tag{36}$$

Consider first the numerator in Eq. (36). Higher value of β suggests higher error in aggregated coefficients and increases δ_{gossip} as expected. More interestingly, both $|\Psi_{ij}|$ and $|\alpha_j|$ also increase δ_{gossip}. This matches our observations that filters using orthonormal encoding matrix require lower communication overhead. In the denominator, lower encoding error δ_m reduces δ_{gossip}.

An alternate bound can be derived from the infinity norm:

$$\begin{aligned}
\frac{|\hat{\gamma}(x_i) - \gamma_m(x_i)|}{|\gamma_m(x_i)|} &\leq \frac{||\Psi(\hat{\alpha} - \alpha)||_\infty}{|\gamma_m(x_i)|} \\
&\leq \frac{||\Psi||_\infty ||\hat{\alpha} - \alpha||_\infty}{(1-\delta_m)|\gamma(x_i)|} \\
&\leq \frac{\beta ||\Psi||_\infty ||\alpha||_\infty}{(1-\delta_m)|\min_{x_i} \gamma(x_i)|}
\end{aligned}$$

Putting everything together, we obtain

$$\frac{|\hat{\gamma}(x_i) - \gamma(x_i)|}{|\gamma(x_i)|} = \frac{|\hat{\gamma}(x_i) - \gamma_m(x_i) + \gamma_m(x_i) - \gamma(x_i)|}{|\gamma(x_i)|} \tag{37}$$

$$\leq \frac{|\hat{\gamma}(x_i) - \gamma_m(x_i)|}{|\gamma(x_i)|} + \frac{|\gamma_m(x_i) - \gamma(x_i)|}{|\gamma(x_i)|} \tag{38}$$

$$\leq (1+\delta_m)\frac{|\hat{\gamma}(x_i) - \gamma_m(x_i)|}{|\gamma_m(x_i)|} + \frac{|\gamma_m(x_i) - \gamma(x_i)|}{|\gamma(x_i)|} \tag{39}$$

$$\leq (1+\delta_m)\delta_{gossip} + \delta_m = \delta \tag{40}$$

Consider Eq. (40). When both the encoding and gossiping errors are present, they have a cascading effect on the overall error propagation. The overall error is bounded (i.e., less than unity) when both terms are bounded and less than unity. In fact, $0 < \gamma(x_i) \leq 1$ and $\delta \leq 1$ are two of the conditions required to establish a time-uniform bound on the error of particle filters [5]. If the following two conditions are also satisfied: 1. Markov chain associated with state evolution undergoes sufficient mixing within a finite number of time steps and 2. state estimate function $f(x_i)$ can be suitably scaled and bounded such that $\sup_{x_i} |f(x_i)| < 1$); then a distributed particle filter based on a consensus algorithm does not induce instability in terms of error propagation and the error remains bounded over time [5]).

To validate the derived error bounds, we run the following simulations. We run a centralized BSpf to track the target using bearing measurements. At each time step, for comparison, we re-compute the particle weights using the four distributed filters. we also report the following the metric:

1. $\max_{x_i} \frac{|\gamma_m(x_i)-\gamma(x_i)|}{|\gamma(x_i)|}$: estimate of δ_m
2. $\max_{x_i} \frac{|\gamma_m(x_i)-\hat{\gamma}(x_i)|}{|\gamma_m(x_i)|}$: estimate of δ_{gossip}
3. $\max_{\alpha_j} \frac{|\hat{\alpha}_j-\alpha_j|}{|\alpha_j|}$: estimate of β
4. $\frac{\max_{x_i} \sum_{j=1}^{m} \beta|\Psi_{ij}||\alpha_j|}{\min_{x_j} |\gamma(x_j)(1-\delta_m)|}$: upper limit of δ_{gossip} for LApf, LCpf, LCpf-GS and CSSpf
5. $\frac{\beta||\Psi||_\infty||\alpha||_\infty}{\min_{x_j} |\gamma(x_j)(1-\delta_m)|}$: alternate upper limit of δ_{gossip} for LApf, LCpf, LCpf-GS and CSSpf
6. $\max_{x_i} \frac{|\hat{\gamma}(x_i)-\gamma(x_i)|}{|\gamma(x_i)|}$: discrepancy in log-likelihood
7. $(1 + \delta_m)\delta_{gossip} + \delta_m$: estimate of δ

Fig. 6 shows the results for track 1. All data points are averaged over 40 MC trials. Consider first δ_m. The LCpf and LCpf-GS have the lowest encoding error followed by CSSpf. The LApf and Clusterpf have the highest encoding error. Nonetheless, $\delta_m < 1$ for all filters across all time steps. Note that, in our set-up, the actual tracking is done using a centralized BSpf. Therefore, $\gamma_m(x_i)$ and by extension δ_m remains constant over different values of NGossip.

Consider next the gossiping error. For LApf, LCpf-GS and Clusterpf, $\delta_{\text{gossip}} < 1$ for all values of NGossip. For CSSpf and LCpf, δ_{gossip} falls below 1 only when NGossip exceeds 3 and 8. We also plot the two derived upper limits and Eq. 36 is shown to be a fairly tight bound. Nonetheless, both upper bounds display trends consistent with the actual values.

Fig. 6 (d) shows the average value of $\beta = \max_j |\hat{\alpha_j} - \alpha_j|/|\alpha_j|$. For track 1, we have $\rho_A = 0.33$. Therefore, Eq. (34) suggests that a minimum of 3 gossip iterations is required to obtain $\beta < 1$ which matches the simulation results.

If we compare the results of Fig. 6 (c) and (d), we note that, even though the β is similar for all algorithms at all values of NGossip, the gossiping error δ_{gossip} can still be quite different across the algorithms. This suggests that additional factors have an important impact on gossiping error.

Consider Eq. (36) and Eq. (37). In the ideal case of $\delta_m = 0$ (i.e., perfect reconstruction of log-likelihoods from m coefficients), the denominator of the bound is the same for all algorithms and we only focus on the numerator. While Eq. (36) provides a tighter bound, the analysis is difficult because the summation involves products of two correlated sequences and there is no obvious bound on the individual entry of Ψ and α; we thus focus on Eq. (37). Since $\delta_{\text{gossip}} \propto \beta||\Psi||_\infty||\alpha||_\infty$, different filters with different encoding matrix Ψ (and corresponding coefficients α) would require different values of β in order to achieve the same gossiping error.

Fig. 6 (e) shows the average value of $||\Psi||_\infty||\alpha||_\infty$ over time for CSSpf, LCpf, LCpf-GS, and LApf. The LCpf unsurprisingly has the largest curve by several order of magnitude followed by CSSpf. Conversely, the LApf has the lowest curve followed by LCpf-GS. Therefore, at low NGossip, even if β is large, LApf has low gossiping error.

Fig. 6 (f) shows the overall error δ. For LApf, LCpf-GS and Clusterpf, $\delta < 1$ for all values of NGossip. For LCpf and CSSpf, the error becomes bounded only when NGossip exceeds 7 and 3 respectively. For all filters, the upper bound $(1 + \delta_m)\delta_{\text{gossip}} + \delta_m$ is very close to the true value. As δ_m is small for all filters, at low NGossip, the gossiping error δ_{gossip} becomes the dominating factor and LCpf-GS has the lowest δ. At higher NGossip, the encoding error becomes the dominating factor and LCpf has lower δ than LApf and Clusterpf.

Fig. 6 (g) shows the average discrepancy in normalized particle weights. The curves for CSSpf and LCpf are significantly higher than the other curves until NGossip > 4 and NGossip > 8 as expected.

Fig. 7 shows the relevant results for track 2. Consider first δ_m. For all filters, the encoding error spikes for time steps 22 to 23. Otherwise, δ_m remains below 1. Again, LCpf and LCpf-GS have the lowest encoding error followed by CSSpf, LApf and Clusterpf.

Consider next δ_{gossip}. For Clusterpf, the gossiping error is consistently below unity. For LApf and LCpf-GS, the gossiping error drops below 1 for NGossip>4; although the gossiping error for both filters is fairly close to 1 even at low NGossip. For CSSpf and LCpf, again a much higher communication overhead is required to reduce δ_{gossip} below 1.

Again, LCpf has the highest curve for $||\Psi||_\infty ||\alpha||_\infty$ by several orders of magnitude. Since the gap between LCpf and LApf is considerably larger than in the case of track 1, LCpf requires even more communication overhead to achieve low δ_{gossip} than in track 1 (i.e., NGossip$\geq$ 75 compared to NGossip$\geq$ 20 in track 1).

At low NGossip where gossiping error is the dominating factor, Clusterpf has the lowest curve for δ followed by LApf and LCpf-GS. At higher NGossip, LCpf-GS has the lowest δ as it has the lowest encoding error.

Finally, Fig. 8 shows the results for track 3. The δ_m remains below unity for all filters for all time steps. For LCpf-GS, LApf and Clusterpf, the gossiping error remains below 1 even at NGossip $= 1$. For CSSpf and LCpf, much higher communication overhead is required. These trends are also reflected in the overall δ and normalized particle weights discrepancy.

We repeat the same experiments using range measurements. Fig. 9-11 show the results for the three tracks. We notice the same overall trends as in the bearing-only tracking case.

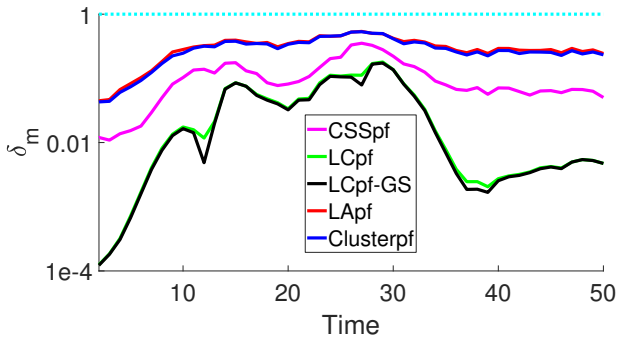
6 Conclusion

In this report, we present four distributed particle filters for single-target bearing-only tracking. CSSpf approximates the log-likelihood function using six sufficient statistics. LCpf uses likelihood consensus to encode the particle log-likelihoods. LApf constructs a graph over all particles and uses the eigenvectors of resulting Laplacian matrix to encode the particle log-likelihood. Finally, Clusterpf groups particles into clusters, computes the cluster joint log-likelihood and recovers individual particle weights using convex minimization.

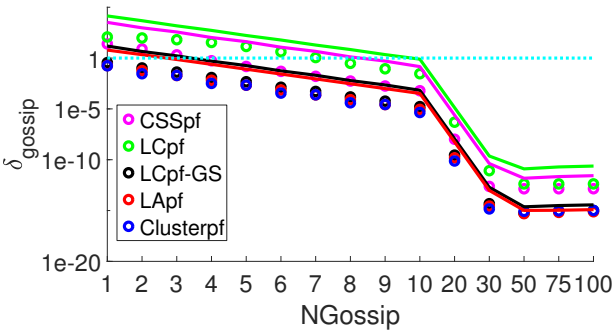
We study each individual algorithm's performance and compare them against each other. LCpf is very fast, but is highly susceptible to gossiping error and requires a much higher communication overhead to achieve adequate tracking performance. The LApf and Clusterpf yield robust tracking performance in all tested scenarios; but they also have the highest runtime by a large margin. More interestingly, when minimal communication overhead is permitted, LApf and Clusterpf are still able to yield robust performance while the other filters would break down. These results make LApf and Clusterpf extremely attractive in energy-constrained tracking scenarios.

References

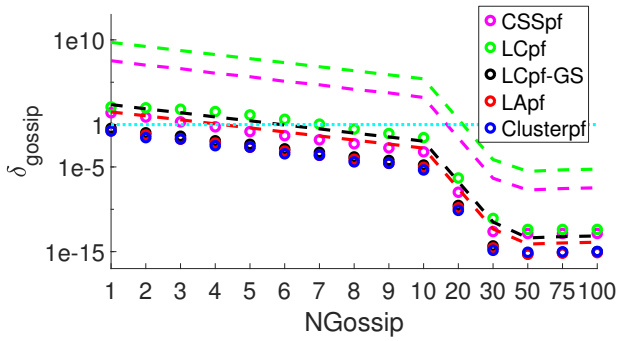
[1] A. Mohammadi and A. Asif, "A constraint sufficient statistics based distributed particle filter for bearing only tracking," in *IEEE Int. Conf. Communications (ICC)*, Ottawa, ON, Canada, Jun 2012, pp. 3670–3675.



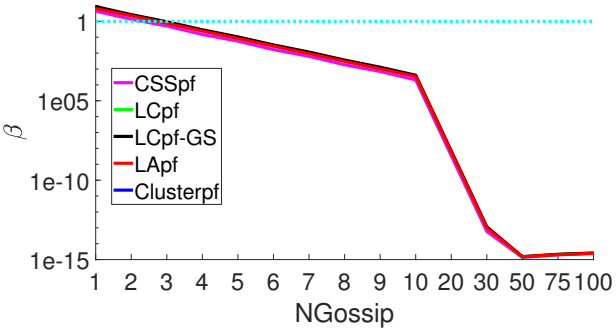
(a) Encoding error δ_m



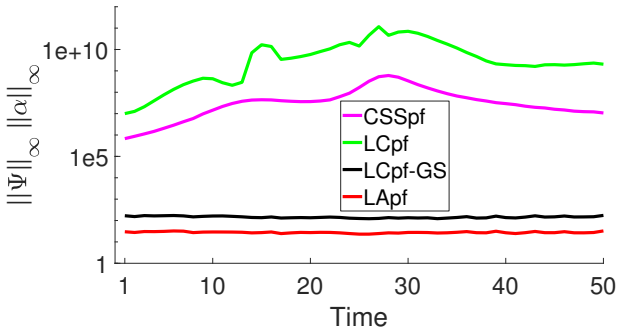
(b) Gossiping error δ_{gossip}, circles represent true gossiping errors and sold lines represent the theoretical upper bound



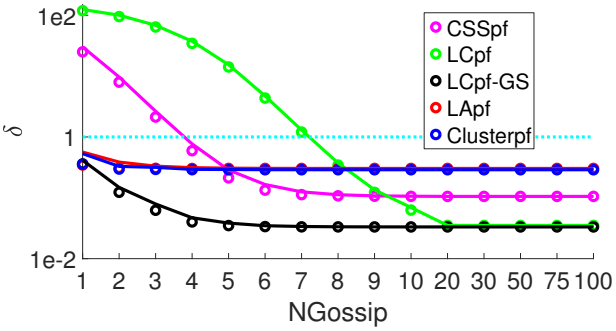
(c) Gossiping error δ_{gossip}, circles represent true gossiping errors and sold lines represent the alternate theoretical upper bound based on infinity norm



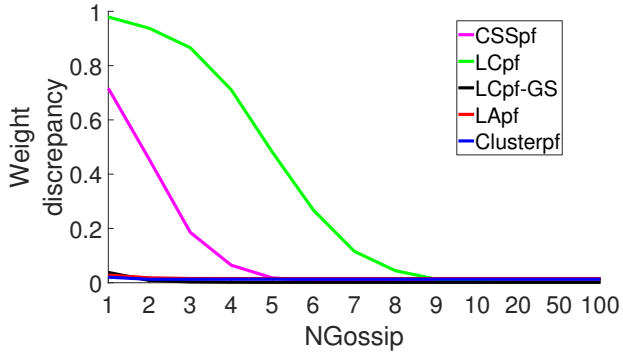
(d) β, the error ratio of coefficients α



(e) Average $||\Psi||_\infty||\alpha||_\infty$ over time

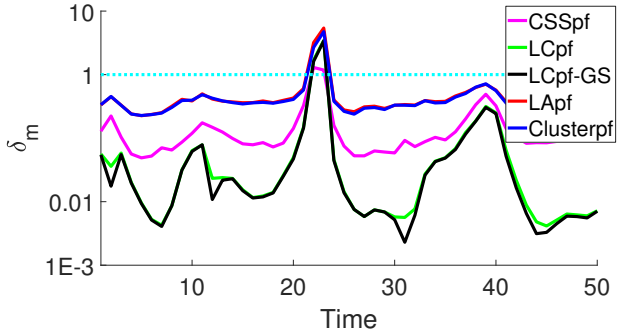


(f) Log-likelihood error δ, circles represent true errors and solid lines represent the theoretical higher bound $(1+\delta_m)\delta_{\text{gossip}}+\delta_m$

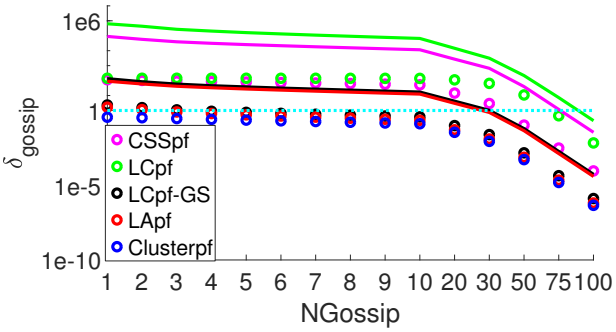


(g) Average normalized particle weights discrepancy $||w_{\text{true}} - w_{\text{approx}}||_2$

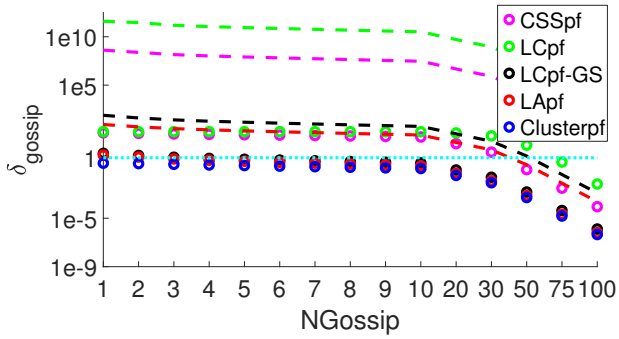
Figure 6: Distributed filter error bound with respect to *NGossip* and time for track 1 averaged over 40 trials. $N = 500$, $d = 2, m = C = 6$.



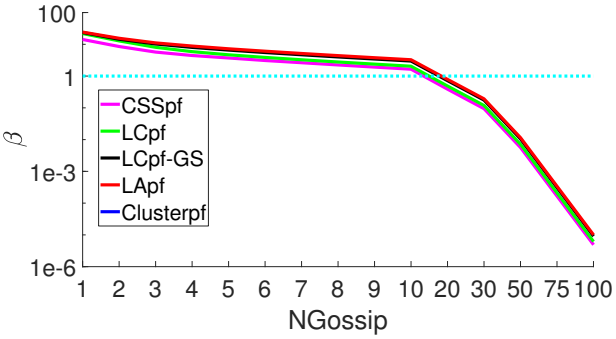
(a) Encoding error δ_m



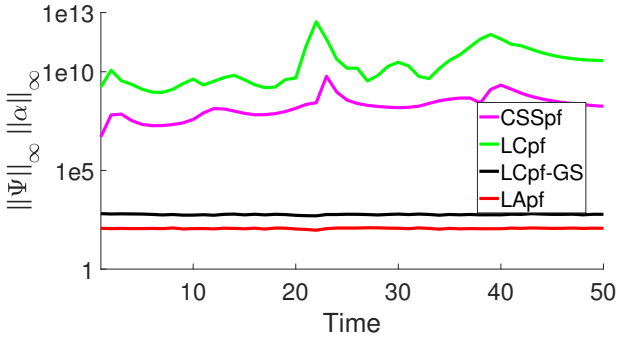
(b) Gossiping error δ_{gossip}, circles represent true gossiping errors and sold lines represent the theoretical upper bound



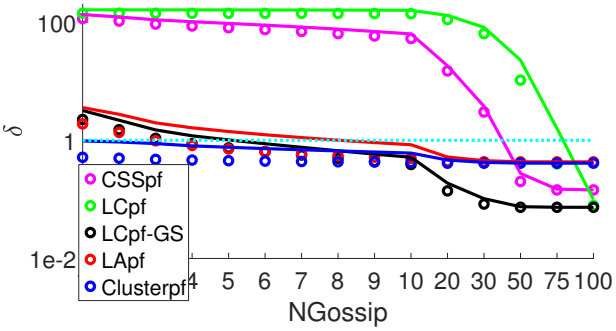
(c) Gossiping error δ_{gossip}, circles represent true gossiping errors and sold lines represent the alternate theoretical upper bound based on infinity norm



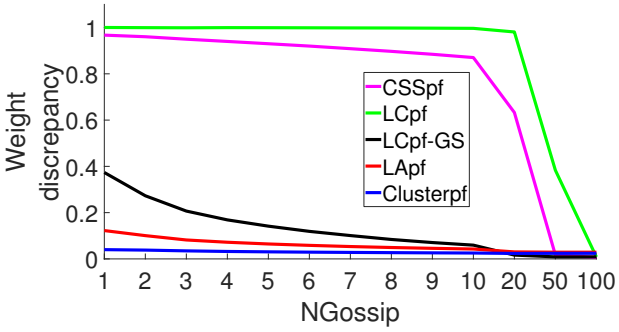
(d) β, the error ratio of coefficients α



(e) Average $||\Psi||_\infty||\alpha||_\infty$ over time

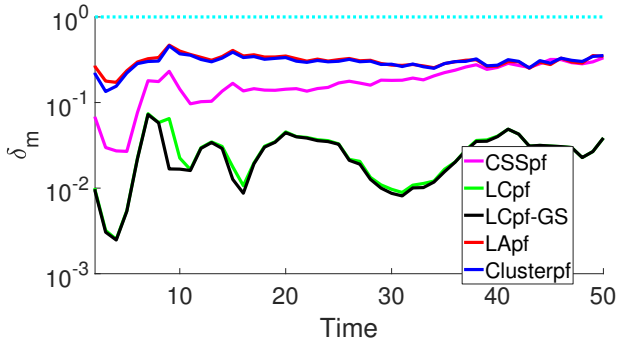


(f) Log-likelihood error δ, circles represent true errors and solid lines represent the theoretical higher bound $(1+\delta_m)\delta_{\text{gossip}}+\delta_m$

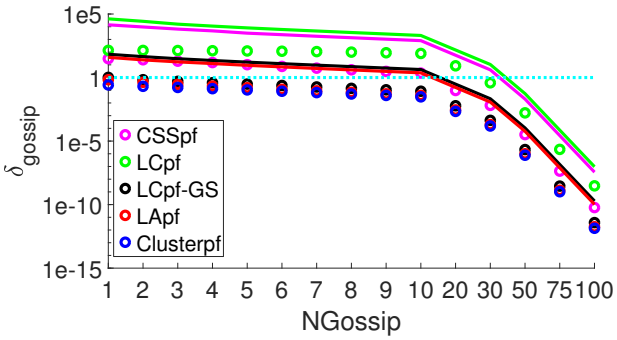


(g) Average normalized particle weights discrepancy $||w_{\text{true}} - w_{\text{approx}}||_2$

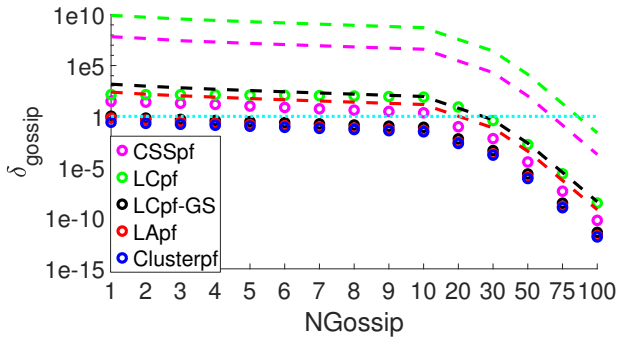
Figure 7: Distributed filter error bound with respect to *NGossip* and time for track 2 averaged over 40 trials. $N = 500$, $d = 2, m = C = 6$.



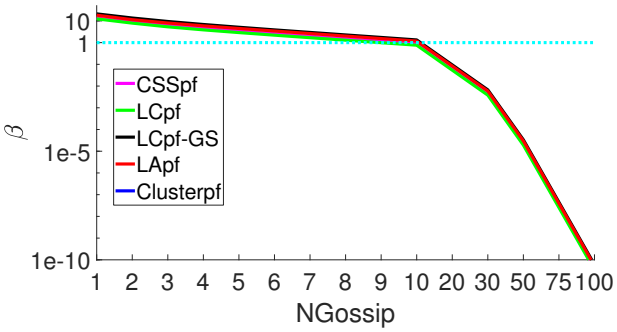
(a) Encoding error δ_m



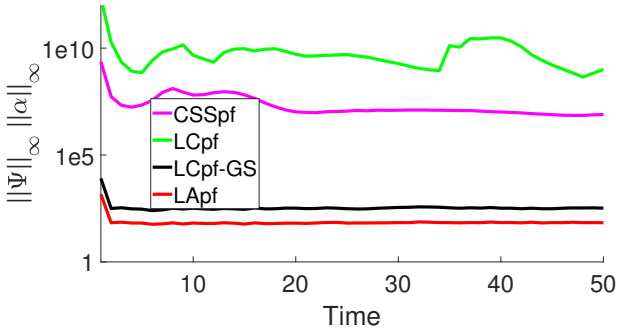
(b) Gossiping error δ_{gossip}, circles represent true gossiping errors and sold lines represent the theoretical upper bound



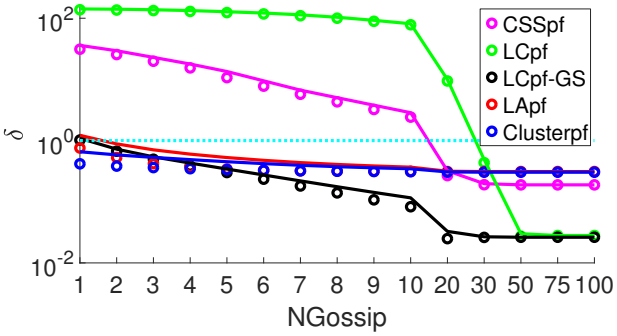
(c) Gossiping error δ_{gossip}, circles represent true gossiping errors and sold lines represent the alternate theoretical upper bound based on infinity norm



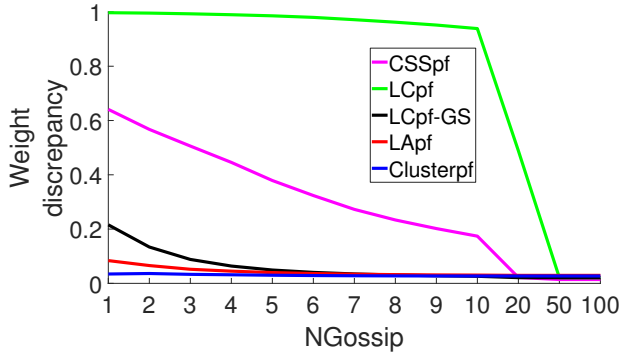
(d) β, the error ratio of coefficients α



(e) Average $||\Psi||_\infty ||\alpha||_\infty$ over time

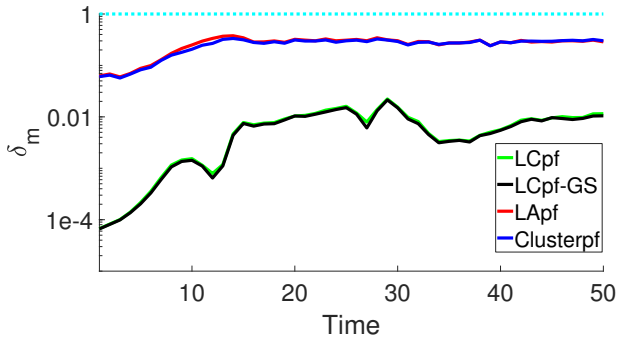


(f) Log-likelihood error δ, circles represent true errors and solid lines represent the theoretical higher bound $(1+\delta_m)\delta_{\text{gossip}}+\delta_m$

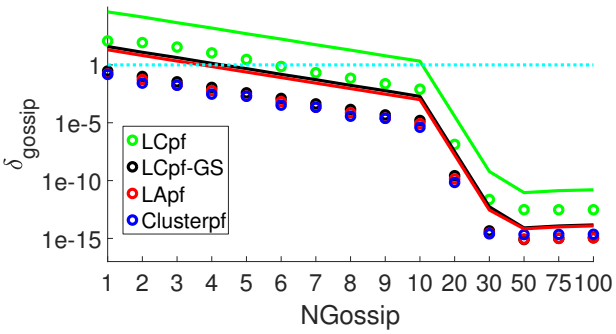


(g) Average normalized particle weights discrepancy $||w_{\text{true}} - w_{\text{approx}}||_2$

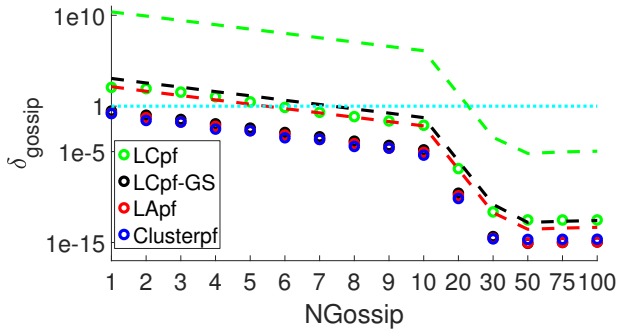
Figure 8: Distributed filter error bound with respect to *NGossip* and time for track 3 averaged over 40 trials. $N = 500$, $d = 2, m = C = 6$.



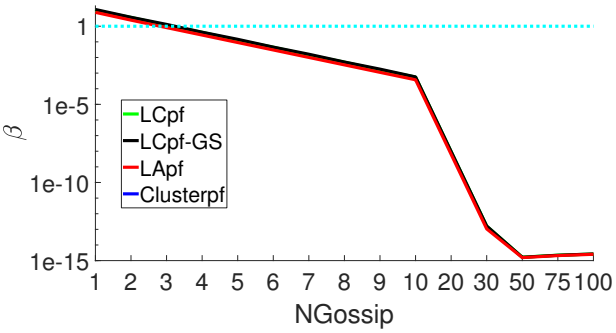
(a) Encoding error δ_m



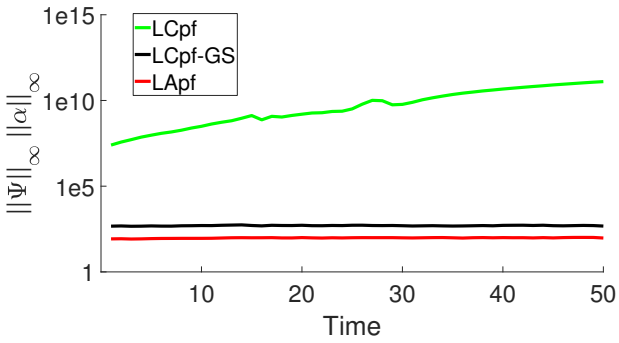
(b) Gossiping error δ_{gossip}, circles represent true gossiping errors and sold lines represent the theoretical upper bound



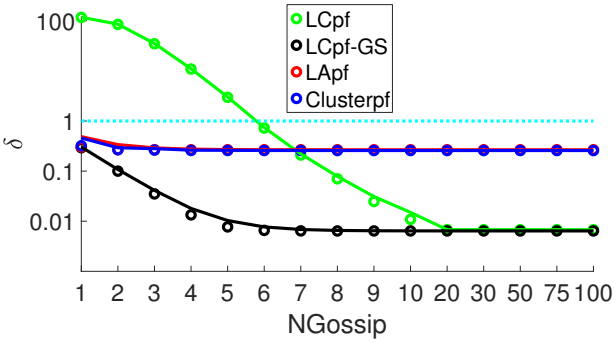
(c) Gossiping error δ_{gossip}, circles represent true gossiping errors and sold lines represent the alternate theoretical upper bound based on infinity norm



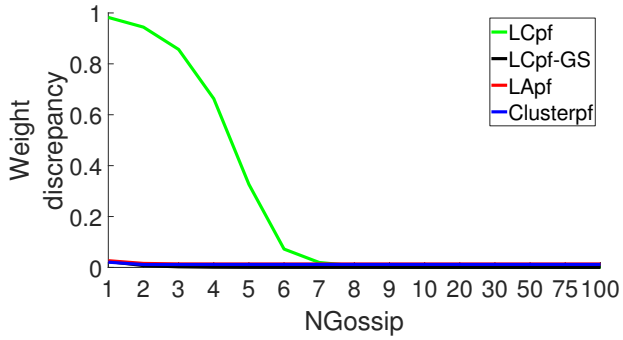
(d) β, the error ratio of coefficients α



(e) Average $||\Psi||_\infty ||\alpha||_\infty$ over time

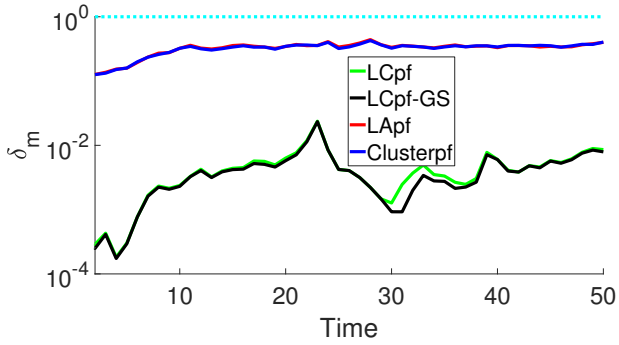


(f) Log-likelihood error δ, circles represent true errors and solid lines represent the theoretical higher bound $(1+\delta_m)\delta_{\text{gossip}}+\delta_m$

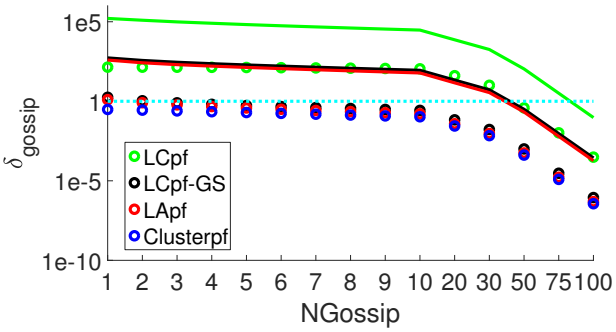


(g) Average normalized particle weights discrepancy $||w_{\text{true}} - w_{\text{approx}}||_2$

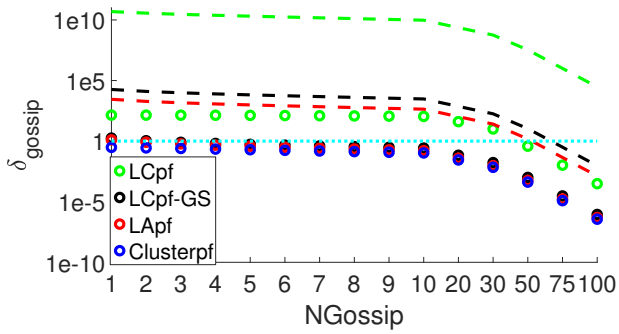
Figure 9: Distributed filter error bound with respect to *NGossip* and time for track 1 averaged over 40 trials. $N = 500$, $d = 2, m = C = 6$. Sensors receive range measurements.



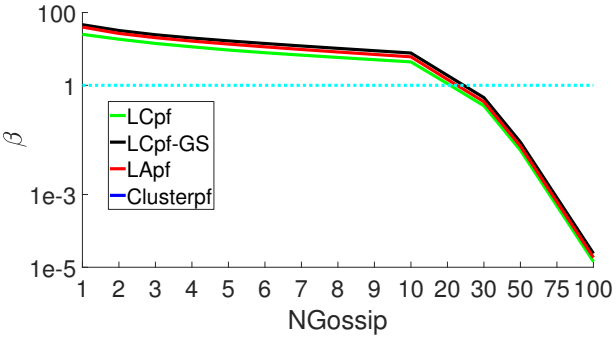
(a) Encoding error δ_m



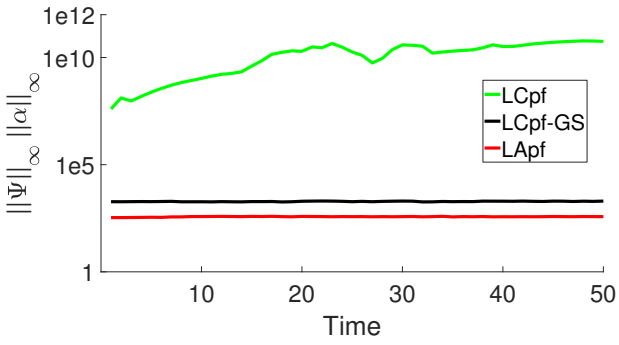
(b) Gossiping error δ_{gossip}, circles represent true gossiping errors and sold lines represent the theoretical upper bound



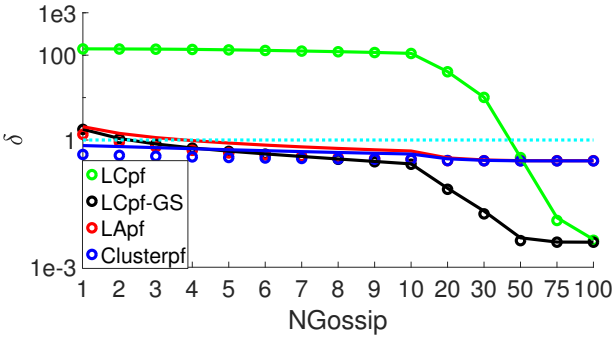
(c) Gossiping error δ_{gossip}, circles represent true gossiping errors and sold lines represent the alternate theoretical upper bound based on infinity norm



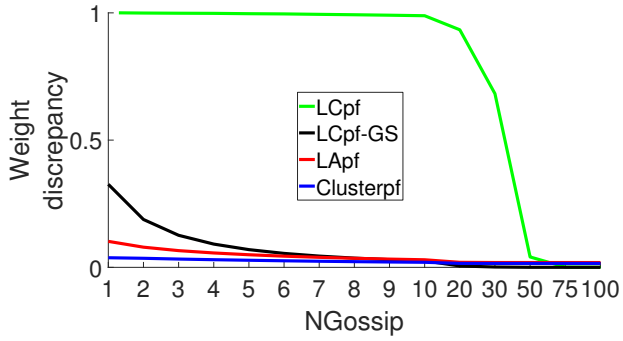
(d) β, the error ratio of coefficients α



(e) Average $||\Psi||_\infty||\alpha||_\infty$ over time

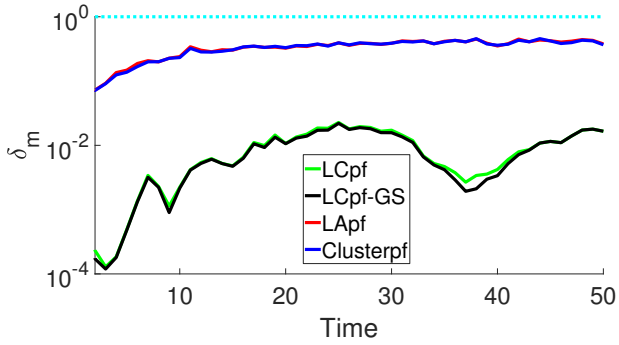


(f) Log-likelihood error δ, circles represent true errors and solid lines represent the theoretical higher bound $(1+\delta_m)\delta_{\text{gossip}}+\delta_m$

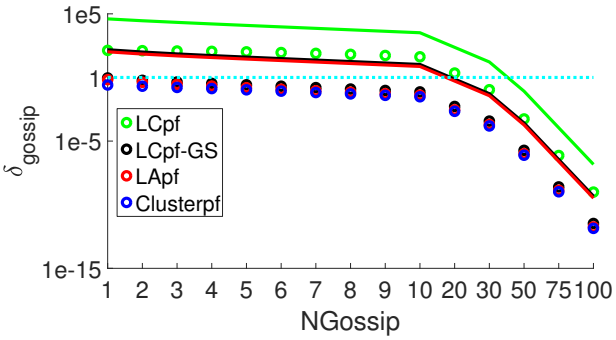


(g) Average normalized particle weights discrepancy $||w_{\text{true}} - w_{\text{approx}}||_2$

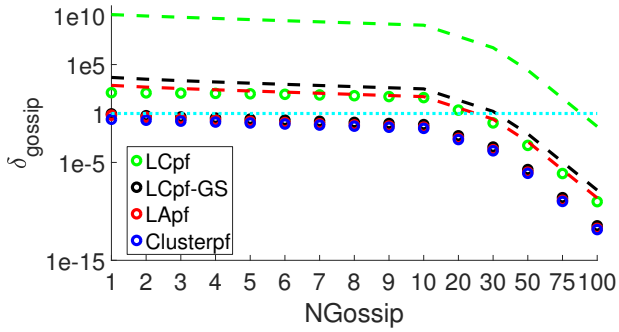
Figure 10: Distributed filter error bound with respect to *NGossip* and time for track 2 averaged over 40 trials. $N = 500$, $d = 2, m = C = 6$.



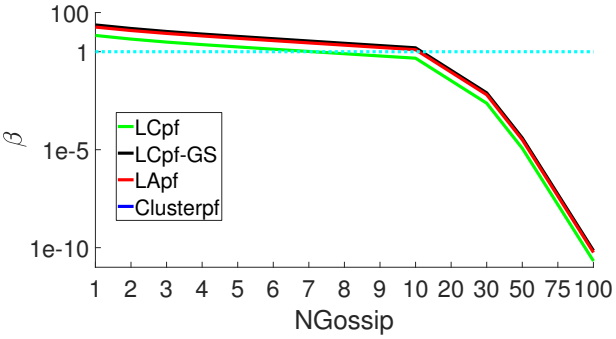
(a) Encoding error δ_m



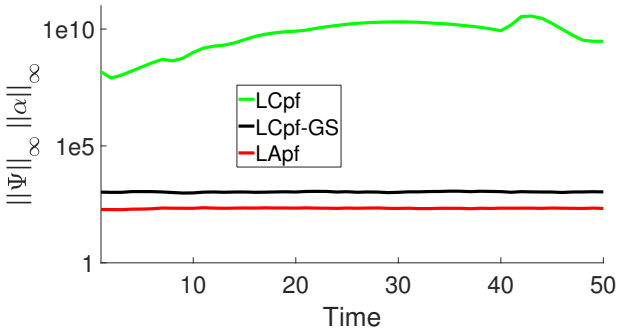
(b) Gossiping error δ_{gossip}, circles represent true gossiping errors and sold lines represent the theoretical upper bound



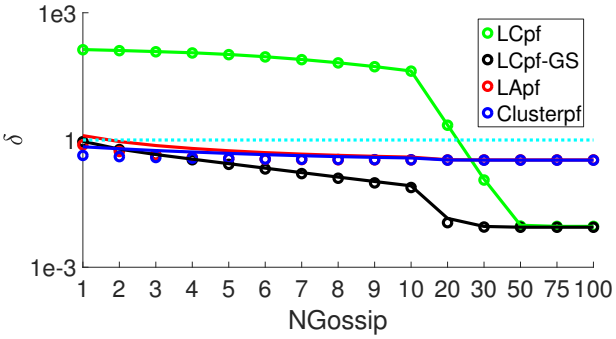
(c) Gossiping error δ_{gossip}, circles represent true gossiping errors and sold lines represent the alternate theoretical upper bound based on infinity norm



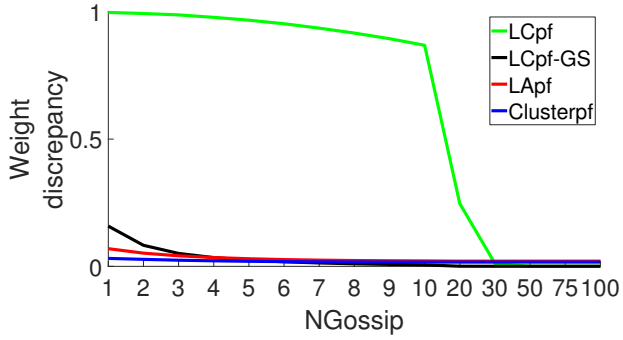
(d) β, the error ratio of coefficients α



(e) Average $||\Psi||_\infty||\alpha||_\infty$ over time



(f) Log-likelihood error δ, circles represent true errors and solid lines represent the theoretical higher bound $(1+\delta_m)\delta_{\text{gossip}}+\delta_m$



(g) Average normalized particle weights discrepancy $||w_{\text{true}} - w_{\text{approx}}||_2$

Figure 11: Distributed filter error bound with respect to *NGossip* and time for track 3 averaged over 40 trials. $N = 500$, $d = 2, m = C = 6$.

[2] O. Hlinka, O. Sluciak, F. Hlawatsch, P. Djuric, and M. Rupp, "Likelihood consensus and its application to distributed particle filtering," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4334–4349, 2012.

[3] M. Rabbat, M. Coates, and S. Blouin, "Graph laplacian distributed particle filtering," in *Signal Process. Conf. (EUPISCO)*, Budapest, Hungary, Aug. 2016, pp. 1493 – 1497.

[4] C. W. Chao, M. Rabbat, and S. Blouin, "Particle weight approximation with clustering for gossip-based distributed particle filters," in *IEEE Int. Workshop Comp Comput. Advances Multi-Sensor Adaptive Process. (CAMSAP)*, Cancun, Mexico, Dec 2015, pp. 85–88.

[5] S. D. Gupta, M. Coates, and M. Rabbat, "Error propagation in gossip-based distributed particle filters," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 3, pp. 148–163, Aug. 2015.