

Analytically-Guided-Sampling Particle Filter Applied to Range-only Target Tracking

Guoquan P. Huang and Stergios I. Roumeliotis

Abstract—Particle filtering (PF) is a popular nonlinear estimation technique and has been widely used in a variety of applications such as target tracking. Within the PF framework, one critical design choice that greatly affects the filter’s performance is the selection of the proposal distribution from which particles are drawn. In this paper, we advocate the proposal distribution to be a Gaussian-mixture-based approximation of the posterior probability density function (pdf) after taking into account the most recent measurement. The novelty of our approach is that each Gaussian in the mixture is determined *analytically* to match the modes of the underlying unknown posterior pdf. As a result, particles are sampled along the *most probable* regions of the state space, hence reducing the probability of particle depletion. We adapt this proposal distribution into a new PF, termed Analytically-Guided-Sampling (AGS)-PF, and apply it to the particular problem of range-only target tracking. Both Monte-Carlo simulation and real-world experimental results validate the superior performance of the proposed AGS-PF over other state-of-the-art PF algorithms.

I. INTRODUCTION AND RELATED WORK

Particle filtering has become an increasingly popular nonlinear estimation approach used in a wide range of applications such as target tracking [1]–[5]. A particle filter (PF) seeks to approximate the posterior probability density function (pdf) by a set of random samples (particles) and updates its estimate recursively in time. Within the sequential importance sampling (SIS) framework, one critical step is to design an appropriate *proposal distribution* (or importance density), which is used to draw particles for the next time step. Clearly, from the Bayesian filtering perspective, the best choice of the proposal distribution is the posterior pdf itself, which, however, in general is intractable to evaluate analytically. In this paper, we focus on formally designing a proposal distribution that better approximates the posterior pdf so as to improve the PF’s performance.

Since, in general, it is also difficult to sample from the *optimal* proposal distribution that minimizes the variances of the particles’ weights conditioned on the trajectory and all available measurements [6], many, often *ad hoc*, choices of the proposal distribution are described in the literature. Typically, the prior pdf is used, which results in the standard (bootstrap) PF weighted by the measurement likelihood [7]. If, however, the prior is uninformative, the generated particles may not be able to sample the state space sufficiently. Specifically, when far from a mode of the posterior pdf,

the weights of the particles decay quickly and lead to particle depletion (i.e., only a few particles have significant weights) [3]. As a result, the very few surviving particles are unable to appropriately represent the underlying posterior pdf, which may cause filter *inconsistency*.¹ In general, in order to converge to meaningful estimates, the standard PF requires using a large number of particles, and thus has high processing requirements. In cases where the posterior pdf is closer than the prior to the measurement likelihood, then using the likelihood, instead of the prior, as the proposal distribution often improves performance [3]. Alternatively, a Gaussian proposal distribution can be obtained by local linearization [6], based on which the unscented PF (UPF) [9] was introduced. The UPF employs the unscented Kalman filter (UKF) or the extended Kalman filter (EKF) to generate the proposal distribution that takes into account the latest measurements, and thus better approximates the posterior pdf. Similar ideas were also exploited in [10]–[12]. However, often due to the multimodal nature of the posterior pdf, the particles sampled from the UKF/EKF posterior pdf do not necessarily capture all the true posterior modes, which may degrade the UPF’s performance.

The closest to the work presented in this paper is the Gaussian sum PF (GSPF) [13] – which essentially is a bank of Gaussian PFs (GPFs) [14] – has been derived based on the concept of the Gaussian sum filter (GSF) [15]. Specifically, by assuming that the prior pdf can be represented as the sum of Gaussian distributions, the GSPF updates each distribution using the particles that are sampled, for example, from the corresponding prior pdf and weighted by the measurement likelihood. However, the GSPF does not provide a measurable criterion about how many Gaussian distributions are needed, and most importantly, where to choose them. These critical issues are addressed in the Analytically-Guided-Sampling (AGS)-PF introduced in this paper.

In particular, the proposed AGS-PF efficiently utilizes the available computational resources by employing a small number of particles drawn from the *most probable* hypotheses about the estimated state, i.e., the AGS-PF samples the most likely regions of the state space. The key idea behind our approach is to employ a Gaussian mixture to approximate the posterior pdf, where each Gaussian is determined *analytically* and corresponds to a mode of the posterior pdf. Specifically, we first formulate and convert the nonlinear cost function of the maximum a posteriori (MAP) optimization

This work was supported by the University of Minnesota through the Digital Technology Center (DTC) and AFOSR MURI FA9550-10-1-0567.

G. P. Huang is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. Email: ghuang@csail.mit.edu. S. I. Roumeliotis is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA. Email: stergios@cs.umn.edu.

¹Consistency is one of the primary criteria for evaluating the performance of an estimator. As defined in [8], an estimator is *consistent* if the estimation errors are zero-mean and have covariance smaller or equal to the one calculated by the estimator.

problem for the current state into *polynomial* form, and then employ algebraic-geometry techniques [16] to analytically compute all the modes of the posterior pdf. Subsequently, we use a Gaussian mixture as the proposal distribution to approximate the posterior distribution. Each Gaussian component matches one mode of the posterior pdf, and its covariance is computed as the inverse of the Hessian matrix of the MAP problem. This analytically-determined proposal distribution provides a better approximation to the posterior pdf, because it not only takes into account the current measurement but also matches all the modes of the posterior pdf. Therefore, the particles drawn from this proposal distribution sample the most probable regions of the state space. Simulation and experimental results demonstrate that the AGS-PF significantly improves the performance in the case of range-only target tracking. We stress that apart from the particular application of range-only target tracking treated here, the proposed analytically-guided sampling scheme is applicable to a broad class of nonlinear estimation problems in robotics and computer vision that can be expressed in (or converted into) polynomial form.

II. ANALYTICALLY-GUIDED SAMPLING-BASED PARTICLE FILTERING

In this section, we present a novel analytically-guided sampling scheme that consists of a Gaussian-mixture-based proposal distribution whose modes are determined analytically to match those of the posterior pdf. As a result, the new AGS-PF effectively focuses the available computational resources on the most probable regions of the state space. In what follows, we begin with a brief overview of the generic PF, and then describe our novel sampling scheme which can readily be integrated into the PF framework.

A. Particle Filtering

A PF seeks to approximate the posterior distribution of the entire state trajectory, $p(\mathbf{x}_{0:k}|z_{0:k})$, sequentially in time, using a set of M weighted samples (particles), $\{\mathbf{x}_{0:k}^{[j]}\}_{j=1}^M$, where $\mathbf{x}_{0:k}^T \triangleq [\mathbf{x}_0^T \cdots \mathbf{x}_k^T]$ denotes all the states up to time-step k , and $z_{0:k}$ denotes all the measurements in the time interval $[0, k]$. To this end, relying on sequential importance resampling (SIR), the PF generally requires three sequential steps to update its estimate (see [3]–[5], and Algorithm 1): Firstly, it draws particles for the next time step from a proposal distribution, $\pi(\mathbf{x}_{0:k}|z_{0:k})$, which is a critical design choice. Secondly, it assigns a weight to each particle in order to account for the fact that the proposal distribution is usually different from the true posterior pdf. Lastly, it performs resampling to multiply (or discard) particles with high (or low) weights.

As mentioned before, one of the main challenges in PFs is designing an appropriate proposal distribution. Even though numerous choices can be made, one typically requires that the proposal distribution has the following form in order to be amenable to recursive computation [4]:

$$\pi(\mathbf{x}_{0:k}|z_{0:k}) = \pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, z_{0:k})\pi(\mathbf{x}_{0:k-1}|z_{0:k-1}) \quad (1)$$

It is well known that the curse of dimensionality can quickly make the particles too sparse to represent the posterior pdf (i.e., particle depletion). In practice, it is common to approximate the proposal distribution $\pi(\mathbf{x}_{0:k}|z_{0:k})$, by fixing the past trajectory $\mathbf{x}_{0:k-1}$ and only sampling the current state \mathbf{x}_k , i.e., using $\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, z_{0:k})$. It has been proven in [6] that the *optimal* proposal distribution for the current state, with respect to minimizing the variance of the particles' weights, is in the form of a conditional pdf conditioned on the past trajectory and all the measurements:

$$\pi_{\text{opt}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}, z_{0:k}) = p(\mathbf{x}_k|\mathbf{x}_{0:k-1}, z_{0:k}) \quad (2)$$

Based on (1) as well as the common assumptions that the motion model is a Markov process [see (18)] and that the measurements are conditionally independent given the states [see (20)], the (unnormalized) importance weight of the j -th particle is computed recursively as follows [4]:

$$w_k^{[j]} = \frac{p(\mathbf{x}_{0:k}^{[j]}|z_{0:k})}{\pi(\mathbf{x}_{0:k}^{[j]}|z_{0:k})} \propto w_{k-1}^{[j]} \frac{p(\mathbf{x}_k^{[j]}|\mathbf{x}_{k-1}^{[j]})p(z_k|\mathbf{x}_k^{[j]})}{\pi(\mathbf{x}_k^{[j]}|\mathbf{x}_{0:k-1}^{[j]}, z_{0:k})} \quad (3)$$

To summarize, a generic PF is outlined in Algorithm 1.

Algorithm 1 A generic particle filtering algorithm

- 1: **loop**
 - 2: Draw particles $\{\mathbf{x}_k^{[j]}\}_{j=1}^M \sim \pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, z_{0:k})$
 - 3: Compute weights via (3), and normalize weights
 - 4: Resample particles based on weights
 - 5: **end loop**
-

In general, the optimal proposal distribution (2) is not available analytically or in a suitable form for efficient sampling, and one may choose (infinitely many) other possible distributions to approximate it. As mentioned before, a common choice is to sample from the state-transition prior distribution (motion model), i.e., $\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, z_{0:k}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$. In this case the weight is simply proportional to the measurement likelihood, $p(z_k|\mathbf{x}_k)$ [see (3)]. However, such a choice may easily lead to filter inconsistency (see Sections IV and V). To address this issue, in the following, we design an analytically-determined proposal distribution by taking the current measurement into account as well as matching all the modes of the posterior pdf.

B. Analytically-Guided Sampling Scheme

Our choice of Gaussian-mixture-based proposal distribution is motivated by the following Gaussian sum theorem (see Theorem 4.1 of [15], p.214):²

Theorem 2.1: For a measurement model with additive Gaussian noise [e.g., see (20)], i.e., $z_k = h(\mathbf{x}_k) + v_k$, where $v_k \sim \mathcal{N}(0, \sigma_k^2)$, and a prior pdf given by $p(\mathbf{x}_k|z_{0:k-1}) = \sum_{i=1}^m \alpha_i \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}^{(i)}, \mathbf{P}_{k|k-1}^{(i)})$, the posterior pdf $p(\mathbf{x}_k|z_{0:k})$ approaches the Gaussian sum $\sum_{i=1}^m \alpha'_i \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{(i)}, \mathbf{P}_{k|k}^{(i)})$ uniformly in \mathbf{x}_k and z_k as

²Throughout this paper, the subscript $\ell|j$ refers to the estimate of a quantity at time-step ℓ , after all measurements up to time-step j have been processed. \hat{x} is used to denote the estimate of a random variable x . $\mathbf{0}_{m \times n}$ denotes a $m \times n$ matrix of zeros.

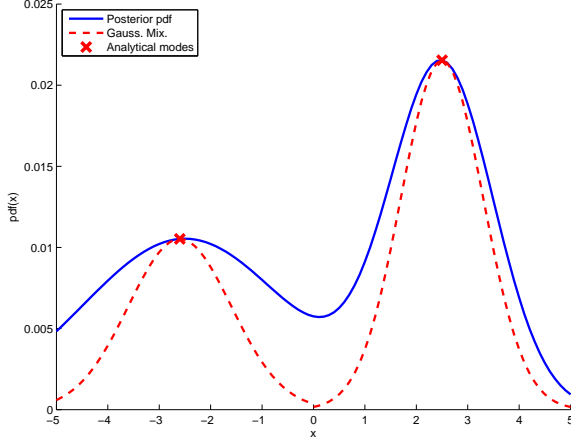


Fig. 1. Illustration of the proposed analytically-determined proposal distribution that uses a Gaussian mixture to approximate the posterior distribution. The modes of the posterior pdf – which are computed analytically – are also used as the modes of the proposed proposal distribution. Note that for visualization the plotted Gaussians of the mixture are scaled so that their modes also coincide along y-axis with those of the posterior pdf.

$\mathbf{P}_{k|k-1}^{(i)} \rightarrow 0$, for $i = 1, \dots, m$, where the mean, covariance, and weight are computed as follows:

$$\hat{\mathbf{x}}_{k|k}^{(i)} = \hat{\mathbf{x}}_{k|k-1}^{(i)} + \mathbf{K}_k^{(i)} \left(z_k - h(\hat{\mathbf{x}}_{k|k-1}^{(i)}) \right) \quad (4)$$

$$\mathbf{P}_{k|k}^{(i)} = \mathbf{P}_{k|k-1}^{(i)} - \mathbf{K}_k^{(i)} \mathbf{H}_k^{(i)} \mathbf{P}_{k|k-1}^{(i)} \quad (5)$$

$$\mathbf{K}_k^{(i)} = \mathbf{P}_{k|k-1}^{(i)} \mathbf{H}_k^{(i)T} \left(\mathbf{H}_k^{(i)} \mathbf{P}_{k|k-1}^{(i)} \mathbf{H}_k^{(i)T} + \sigma_k^2 \right)^{-1} \quad (6)$$

$$\mathbf{H}_k^{(i)} = \nabla_{\mathbf{x}_k} h \Big|_{\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1}^{(i)}} \quad (7)$$

$$\alpha'_i = \frac{\alpha_i \beta_i}{\sum_{i=1}^m \alpha_i \beta_i} \quad (8)$$

$$\beta_i = \mathcal{N} \left(z_k; h(\hat{\mathbf{x}}_{k|k-1}^{(i)}), \mathbf{H}_k^{(i)} \mathbf{P}_{k|k-1}^{(i)} \mathbf{H}_k^{(i)T} + \sigma_k^2 \right) \quad (9)$$

Based on this theorem, under mild assumptions, the Gaussian mixture can provide a good approximation to the posterior pdf. Hence, we propose to use a Gaussian mixture as the proposal distribution in the AGS-PF, while the novelty here is that we employ tools from algebraic geometry to analytically determine the modes of the posterior pdf, which are then used as the modes of the proposal distribution (see Fig. 1).

In particular, given that the prior pdf in the PF is approximated by a set of particles [see (10)], $\{\mathbf{x}_{k\ominus}^{[j]}, w_{k\ominus}^{[j]}\}_{j=1}^M$, we find a Gaussian mixture to approximate this distribution, which can be achieved, e.g., by clustering the M particles into m groups and then fitting a Gaussian to each group.

$$p(\mathbf{x}_k | z_{0:k-1}) \simeq \sum_{j=1}^M w_{k\ominus}^{[j]} \delta(\mathbf{x}_k - \mathbf{x}_{k\ominus}^{[j]}) \quad (10)$$

$$\simeq \sum_{i=1}^m \alpha_i \underbrace{\mathcal{N} \left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}^{(i)}, \mathbf{P}_{k|k-1}^{(i)} \right)}_{p^{(i)}(\mathbf{x}_k | z_{0:k-1})} \quad (11)$$

where $\alpha_i = \sum_{\mathbf{x}_{k\ominus}^{[j]} \in \text{group } i} w_{k\ominus}^{[j]}$, and $\delta(\cdot)$ is the Dirac delta function. Using Bayes' rule and the above Gaussian mixture approximation of the prior pdf (11), the posterior pdf can be

approximated as follows:

$$\begin{aligned} p(\mathbf{x}_k | z_{0:k}) &\propto p(z_k | \mathbf{x}_k) p(\mathbf{x}_k | z_{0:k-1}) \\ &\simeq \sum_{i=1}^m \alpha_i \underbrace{p(z_k | \mathbf{x}_k) p^{(i)}(\mathbf{x}_k | z_{0:k-1})}_{\propto p^{(i)}(\mathbf{x}_k | z_{0:k})} \\ &\simeq \sum_{i=1}^m \sum_{\ell=1}^{n_i} \alpha'_{i\ell} \mathcal{N} \left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{(i\ell)}, \mathbf{P}_{k|k}^{(i\ell)} \right) \quad (12) \end{aligned}$$

It is important to point out that due to the nonlinearity of the measurement model, for each Gaussian distribution in the prior mixture, $p^{(i)}(\mathbf{x}_k | z_{0:k-1})$, the corresponding posterior pdf, $p^{(i)}(\mathbf{x}_k | z_{0:k})$, very often is a multi-modal, rather than a unimodal, distribution. To take this fact into account, in (12) we use n_i Gaussian distributions to approximate the i -th posterior pdf, $p^{(i)}(\mathbf{x}_k | z_{0:k})$, whose weights, $\alpha'_{i\ell}$, are computed based on (8) but with appropriate normalization, i.e., $\alpha'_{i\ell} = \frac{\alpha_i \beta_i}{\sum_{i=1}^m \sum_{\ell=1}^{n_i} \alpha_i \beta_i}$. Note that n_i is *analytically determined* as the number of modes of the i -th posterior pdf, rather than arbitrarily chosen as in most Gaussian sum filters such as the GSPF [13]. Here the index i_ℓ denotes the ℓ -th Gaussian of the i -th posterior pdf. Thus, a mixture of $n = \sum_{i=1}^m n_i$ Gaussian distributions is used to approximate the posterior pdf, which is then used as the proposal distribution in the proposed AGS-PF.

We now aim to analytically compute all the modes, $\hat{\mathbf{x}}_{k|k}^{(i\ell)}$, ($\forall i = 1, \dots, m$ and $\forall \ell = 1, \dots, n_i$), of the i -th posterior pdf, $p^{(i)}(\mathbf{x}_k | z_{0:k})$, i.e., to solve the following one-time-step MAP problem analytically:

$$\max_{\mathbf{x}_k} p^{(i)}(\mathbf{x}_k | z_{0:k}) \propto p^{(i)}(\mathbf{x}_k | z_{0:k-1}) p(z_k | \mathbf{x}_k) \quad (13)$$

Exploiting the Gaussianity of the measurement noise along with the Gaussian approximation of the i -th element, $p^{(i)}(\mathbf{x}_k | z_{0:k-1})$, of the approximate prior (11), the MAP problem (13) is equivalent to the following nonlinear least-squares problem [17]:

$$\min_{\mathbf{x}_k} \frac{1}{2} \|\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}^{(i)}\|_{\mathbf{P}_{k|k-1}^{(i)}}^2 + \frac{1}{2} \|z_k - h(\mathbf{x}_k)\|_{\sigma_k^2}^2 \quad (14)$$

where we have employed the notation $\|\mathbf{a}\|_{\mathbf{M}}^2 \triangleq \mathbf{a}^T \mathbf{M}^{-1} \mathbf{a}$. Note that for a broad class of nonlinear estimation problems arising in robotics and computer vision, we can transform or convert (14) into polynomial form and then solve for all the local minima (corresponding to all the modes) analytically using algebraic geometry techniques [16].

Once we analytically find all the modes, $\hat{\mathbf{x}}_{k|k}^{(i\ell)}$, of the i_ℓ -th Gaussian component of the posterior pdf (proposal distribution), we compute the corresponding covariance, $\mathbf{P}_{k|k}^{(i\ell)}$, from the inversion of the Hessian matrix of (14) as follows:

$$\mathbf{P}_{k|k}^{(i\ell)} = \left(\mathbf{P}_{k|k-1}^{(i)}{}^{-1} + \sigma_k^{-2} \mathbf{H}_k^{(i\ell)T} \mathbf{H}_k^{(i\ell)} \right)^{-1} \quad (15)$$

where the measurement Jacobian, $\mathbf{H}_k^{(i\ell)}$, is evaluated at the i_ℓ -th analytically-computed mode, $\hat{\mathbf{x}}_{k|k}^{(i\ell)}$ [see (7)]. Based on the matrix inversion lemma [18], it is not difficult to see that (15) is precisely the standard EKF covariance update equation [8] [also see (5)].

III. APPLICATION: RANGE-ONLY TARGET TRACKING

In this section, we apply the AGS-PF presented in the previous section to the particular problem of range-only target tracking, to illustrate in detail the key idea of *analytically* determining the proposal distribution.

A. Problem Formulation

Consider a single sensor moving in a plane and estimating the state (position, velocity, etc.) of a moving target, by processing the available range measurements. In this work, we study the case of *global* tracking, i.e., the position of the target is expressed with respect to a fixed (global) frame of reference, instead of a relative *sensor-centered* one. We hereafter assume that the tracking sensor's position is known with high accuracy in the global frame of reference (e.g., from GPS). The target state at time-step k is defined as a vector of dimension $2N$, where $N - 1$ is the highest order of the time derivative of the target position described by a known stochastic motion model, and can include components such as position, velocity, and acceleration.

$$\mathbf{x}_k = [x_{T_k} \ y_{T_k} \ \dot{x}_{T_k} \ \dot{y}_{T_k} \ \ddot{x}_{T_k} \ \ddot{y}_{T_k} \ \cdots]^T \quad (16)$$

$$= [\mathbf{p}_{T_k}^T \ \mathbf{d}_{T_k}^T]^T \quad (17)$$

where $\mathbf{p}_{T_k} \triangleq [x_{T_k} \ y_{T_k}]^T$ is the target position, and $\mathbf{d}_{T_k} \triangleq [\dot{x}_{T_k} \ \dot{y}_{T_k} \ \ddot{x}_{T_k} \ \ddot{y}_{T_k} \ \cdots]^T$ denotes all the higher-order time derivatives of the target position. In the following, the target stochastic motion model and the sensor measurement model are explained.

1) *Motion model*: We consider the case where the target moves randomly but assume that the stochastic model describing the motion of the target (e.g., constant acceleration or constant velocity [8]) is known. In particular, the discrete-time state propagation equation is generically given by the following linear form:

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1} \mathbf{x}_{k-1} + \mathbf{G}_{k-1} \mathbf{w}_{k-1} \quad (18)$$

where \mathbf{w}_{k-1} is zero-mean white Gaussian noise with covariance \mathbf{Q}_{k-1} . The state transition matrix, $\mathbf{\Phi}_{k-1}$, and the process noise Jacobian, \mathbf{G}_{k-1} , that appear in the preceding expression depend on the motion model used [8]. We will make no further assumptions on these matrices other than that their values are known. It is clear that this is a special case of a Markov model and can be written in a general conditional pdf form, $\mathbf{x}_k \sim p(\mathbf{x}_k | \mathbf{x}_{k-1})$.

2) *Measurement model*: In this work, we are interested in the case where a single sensor measures its distance to the target. Statistically, such a measurement follows a conditional pdf, $z_k \sim p(z_k | \mathbf{x}_k)$, and its function is given by:

$$z_k = \sqrt{(x_{T_k} - x_{S_k})^2 + (y_{T_k} - y_{S_k})^2} + v_k \quad (19)$$

$$\triangleq h(\mathbf{x}_k) + v_k \quad (20)$$

where $\mathbf{p}_{S_k} \triangleq [x_{S_k} \ y_{S_k}]^T$ is the known sensor position expressed in the global frame of reference, and v_k is the zero-mean white Gaussian measurement noise, and its variance is σ_k^2 , i.e., $v_k \sim \mathcal{N}(0, \sigma_k^2)$.

B. Computing the Modes of the Proposal Distribution

We now focus on solving (14) analytically in order to determine all the modes of the posterior pdf. By observing that the range measurement depends only on the target position [see (20)], we first decouple the target position \mathbf{p}_{T_k} and the remaining states \mathbf{d}_{T_k} in solving (14), so as to simplify the ensuing derivations.

Lemma 3.1: Minimizing (14) is equivalent to sequentially solving the following two problems:

- 1) Solving the following minimization problem with respect to the target position only:

$$\min_{\mathbf{p}_{T_k}} \frac{1}{2} \|\mathbf{p}_{T_k} - \hat{\mathbf{p}}_{T_k|k-1}^{(i)}\|_{\mathbf{P}_{\mathbf{p}\mathbf{p}|k-1}^{(i)}}^2 + \frac{1}{2} \|z_k - h(\mathbf{p}_{T_k})\|_{\sigma_k^2}^2 \quad (21)$$

where $\mathbf{P}_{\mathbf{p}\mathbf{p}|k-1}^{(i)}$ is the covariance matrix corresponding to the target position, obtained by partitioning the covariance matrix as: $\mathbf{P}_{k|k-1}^{(i)} \triangleq \begin{bmatrix} \mathbf{P}_{\mathbf{p}\mathbf{p}|k-1}^{(i)} & \mathbf{P}_{\mathbf{p}\mathbf{d}|k-1}^{(i)} \\ \mathbf{P}_{\mathbf{d}\mathbf{p}|k-1}^{(i)} & \mathbf{P}_{\mathbf{d}\mathbf{d}|k-1}^{(i)} \end{bmatrix}$.

- 2) Using the ℓ -th optimal target position of (21), $\hat{\mathbf{p}}_{T_k|k}^{(i_\ell)}$, to determine the remaining states:

$$\hat{\mathbf{d}}_{T_k|k}^{(i_\ell)} = \hat{\mathbf{d}}_{T_k|k-1}^{(i)} - \sum_{\mathbf{d}\mathbf{d}|k-1}^{(i)} \sum_{\mathbf{d}\mathbf{p}|k-1}^{(i)} (\hat{\mathbf{p}}_{T_k|k}^{(i_\ell)} - \hat{\mathbf{p}}_{T_k|k-1}^{(i)}) \quad (22)$$

where the following partitioning of the information

$$\text{matrix is used: } \mathbf{P}_{k|k-1}^{(i)} \triangleq \begin{bmatrix} \sum_{\mathbf{p}\mathbf{p}|k-1}^{(i)} & \sum_{\mathbf{p}\mathbf{d}|k-1}^{(i)} \\ \sum_{\mathbf{d}\mathbf{p}|k-1}^{(i)} & \sum_{\mathbf{d}\mathbf{d}|k-1}^{(i)} \end{bmatrix}.$$

Proof: See [17]. \blacksquare

We thus see from the above lemma that the size of the nonlinear problem has dramatically decreased from $2N$ for (14) to a constant size of 2 for (21). Moreover, the analytical solution for the target position is independent of its higher-order time derivatives, regardless of the stochastic target motion model.

To solve (21) analytically, we first introduce a new variable $\rho_k = h(\mathbf{p}_{T_k})$. Then, (21) is equivalent to the following *constrained* minimization problem:³

$$\min_{\mathbf{p}_{T_k}, \rho_k} \frac{1}{2} \|\mathbf{p}_{T_k} - \hat{\mathbf{p}}_{T_k|k-1}^{(i)}\|_{\mathbf{P}_{\mathbf{p}\mathbf{p}|k-1}^{(i)}}^2 + \frac{1}{2} \|z_k - \rho_k\|_{\sigma_k^2}^2 \quad (23)$$

$$\text{s.t. } \rho_k^2 = (x_{S_k} - x_{T_k})^2 + (y_{S_k} - y_{T_k})^2, \rho_k \geq 0 \quad (24)$$

which we solve by employing the method of Lagrange multipliers [19]. Specifically, without loss of generality, by assuming $\mathbf{P}_{\mathbf{p}\mathbf{p}|k-1}^{(i)} \triangleq \mathbf{Diag}(s_1, s_2)$, the Lagrangian function is constructed as follows:⁴

$$\mathcal{L}(x_{T_k}, y_{T_k}, \rho_k, \lambda) = \frac{s_1}{2} (x_{T_k} - \hat{x}_{T_k|k-1}^{(i)})^2 + \frac{s_2}{2} (y_{T_k} - \hat{y}_{T_k|k-1}^{(i)})^2 + \frac{(z_k - \rho_k)^2}{2\sigma_k^2} + \lambda (\rho_k^2 - (x_{S_k} - x_{T_k})^2 - (y_{S_k} - y_{T_k})^2) \quad (25)$$

³It is important to note that similar derivations of the analytical approach for solving (14) in this case can be found in our previous work [17]. For completeness, we briefly describe this approach here.

⁴We can always diagonalize $\mathbf{P}_{\mathbf{p}\mathbf{p}|k-1}^{(i)}$ by applying a 2D rotational transformation, which does not affect the distance measurements. Moreover, we here temporarily omit the positivity constraint on d , which will be used later for determining the feasible solutions.

where λ is the Lagrange multiplier. Setting the derivatives of $\mathcal{L}(\cdot)$ with respect to the four optimization variables to zero, and performing simple algebraic manipulations, we obtain:

$$\frac{\partial \mathcal{L}}{\partial x_T} = 0 \Rightarrow x_{T_k} = \frac{s_1 \hat{x}_{T_k|k-1}^{(i)} - 2\lambda x_{S_k}}{s_1 - 2\lambda} \quad (26)$$

$$\frac{\partial \mathcal{L}}{\partial y_T} = 0 \Rightarrow y_{T_k} = \frac{s_2 \hat{y}_{T_k|k-1}^{(i)} - 2\lambda y_{S_k}}{s_2 - 2\lambda} \quad (27)$$

$$\frac{\partial \mathcal{L}}{\partial \rho_k} = 0 \Rightarrow \rho_k = \frac{z_k}{1 + 2\sigma_k^2 \lambda} \quad (28)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \Rightarrow 0 = \rho_k^2 - (x_{S_k} - x_{T_k})^2 - (y_{S_k} - y_{T_k})^2 \quad (29)$$

We substitute (26)-(28) into (29) and multiply both sides of (29) with $(1 + 2\sigma_k^2 \lambda)^2 (s_1 - 2\lambda)^2 (s_2 - 2\lambda)^2$, to obtain a fourth-order *univariate* polynomial in λ :⁵

$$0 = f(\lambda) = \sum_{i=0}^4 a_i \lambda^i \quad (30)$$

where $a_i, i = 0, \dots, 4$, are the coefficients expressed in terms of the known quantities $s_1, s_2, z_k, \sigma_k, \hat{x}_{T_k|k-1}^{(i)}, \hat{y}_{T_k|k-1}^{(i)}, x_{S_k}$, and y_{S_k} . Since $f(\lambda)$ is quartic, we compute its roots in closed form [17].

Although, in general, there exist 4 solutions for λ and thus 4 solutions for x_{T_k}, y_{T_k} and ρ_k , as they depend injectively on λ [see (26)-(28)], we only need to consider the pairs (x_{T_k}, y_{T_k}) that correspond to real solutions for λ and to a nonnegative ρ_k [see (24)]. Moreover, since some of these solutions could be local maxima or saddle points, the second-order derivative test [19] is employed to identify the minima. In fact, it was shown in [17] that there are at most 2 local minima for the problem (21), i.e., $n_i \leq 2$. Finally, once we determine all the local minima for the target position, we can compute the corresponding estimates for the higher-order position derivatives via (22).

C. Computing the Covariance of Each Gaussian in the Mixture (Proposal Distribution)

In order to obtain the covariance, $\mathbf{P}_{k|k}^{(i_\ell)}$, of the i_ℓ -th Gaussian component of the proposal distribution (15), we compute the measurement Jacobian, $\mathbf{H}_k^{(i_\ell)}$, as follows [see (20)]:

$$\mathbf{H}_k^{(i_\ell)} = \begin{bmatrix} (\hat{\mathbf{p}}_{T_k|k}^{(i_\ell)} - \mathbf{p}_{S_k})^T & \mathbf{0}_{1 \times (2N-2)} \\ \|\hat{\mathbf{p}}_{T_k|k}^{(i_\ell)} - \mathbf{p}_{S_k}\| & \end{bmatrix} \quad (31)$$

where the first block column of $\mathbf{H}_k^{(i_\ell)}$ is the Jacobian with respect to the target position, while the Jacobian with respect to the higher-order derivatives of the target position is $\mathbf{0}_{1 \times (2N-2)}$ since they are not involved in the measurement model [see (20)].

⁵It is important to note that if any of the denominators of (26)-(28) becomes zero while the corresponding numerator is nonzero, the target is located at infinity. Moreover the cost of (24) also becomes infinity and hence attains the global maximum, which is not interesting to us. On the other hand, there exists the degenerate case where both the numerator and denominator of (26) or (27) become zero, which can be avoided through an appropriate coordinate transformation.

D. Range-only Target Tracking Using the AGS-PF

We now apply the AGS-PF to the range-only target tracking problem. Note that since the analytically-determined proposal distribution derived in the previous section not only takes into account the current measurement but also matches all the modes of the posterior pdf, it provides a better approximation to the posterior pdf, and thus the AGS-PF is expected to perform better than the standard PF.

Specifically, at time-step $k-1$, we cluster the particles into m groups, e.g., using the K-means algorithm [20]. Here, m is a design choice selected based on the available computational resources. We compute the sample mean and covariance of the i -th group ($i = 1, \dots, m$), and approximate this group by a Gaussian, $\mathcal{N}^{(i)} \triangleq \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}^{(i)}, \mathbf{P}_{k-1|k-1}^{(i)})$, $i = 1, \dots, m$. Then, based on the *linear* motion model (18), we propagate each Gaussian to obtain the prior, $\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}^{(i)}, \mathbf{P}_{k|k-1}^{(i)})$, as follows:

$$\hat{\mathbf{x}}_{k|k-1}^{(i)} = \Phi_{k-1} \hat{\mathbf{x}}_{k-1|k-1}^{(i)} \quad (32)$$

$$\mathbf{P}_{k|k-1}^{(i)} = \Phi_{k-1} \mathbf{P}_{k-1|k-1}^{(i)} \Phi_{k-1}^T + \mathbf{G}_{k-1} \mathbf{Q}_{k-1} \mathbf{G}_{k-1}^T \quad (33)$$

When a new range measurement becomes available, we analytically compute all the modes of the posterior pdf for each of the m Gaussians $\mathcal{N}^{(i)}$, by solving (14) (see Section III-B). Once all the modes of the posterior pdf (and thus the proposal distribution) are determined, we compute the corresponding covariance based on (15) for each Gaussian component of the proposal distribution. Finally, after all the Gaussian components (modes and covariances) are specified, we use them as a proposal distribution to draw particles, $\{\mathbf{x}_k^{[j]}\}_{j=1}^{M_{i_\ell}} \sim \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{(i_\ell)}, \mathbf{P}_{k|k}^{(i_\ell)})$, where M_{i_ℓ} is the number of particles drawn from the i_ℓ -th Gaussian. For simplicity, M_{i_ℓ} is set equal to the number of the particles originating from the i -th group after clustering, i.e., $M_{i_\ell} = M_i$, though a more adaptive scheme (e.g., based on the particles' weights) may be used. In summary, the main steps of the AGS-PF range-only target tracking are outlined in Algorithm 2.

Algorithm 2 AGS-PF for range-only target tracking

Require: Initialize particles by sampling from $p(\mathbf{x}_0)$

- 1: **loop**
 - 2: Cluster particles into m groups using K-means, and fit a Gaussian to each group
 - 3: Propagate each of the m Gaussians (means and covariances) to obtain the priors via (32) and (33)
 - 4: For each group, given a new measurement, analytically determine the proposal distribution as a Gaussian mixture (see Sections III-B and III-C)
 - 5: Draw particles from the analytically-determined proposal distribution (Gaussian mixture)
 - 6: Compute weights via (3), prune particles, and normalize weights
 - 7: Resample particles based on weights
 - 8: **end loop**
-

Note that in the case of range-only tracking, we may find multiple modes of the posterior pdf for each group at each

time step. Hence, the number of Gaussian components of the proposal distribution may be larger than the original number of clusters. This can result in an unbounded growth of particles over time in the worst case, since more and more particles may be generated for each group in the subsequent time steps. In order to keep the number of particles constant and reduce the computational cost of the AGS-PF, at every time step we *prune* out particles with low weights (see Algorithm 2). As a result, the AGS-PF can use substantially *fewer* particles while achieving significantly *better* performance than the standard PF (see Sections IV and V). This is attributed to the fact that the particles drawn from the analytically-determined proposal distribution that captures all the modes of the posterior pdf, are *effectively* sampled from the *most probable* regions of the state space.

IV. SIMULATION RESULTS

A series of Monte-Carlo simulations were conducted under various conditions, in order to validate the capability of the proposed AGS-PF to improve tracking performance. The metrics used to evaluate the filter's performance are: (i) the average root mean squared error (RMSE), and (ii) the normalized (state) estimation error squared (NEES) [8]. Both metrics are computed by averaging over all Monte-Carlo runs for each time step. The average RMSE provides us with a concise metric of the accuracy of a given filter, while the NEES is a criterion for evaluating the filter's consistency. Specifically, it is known that the NEES of a d -dimensional Gaussian random variable follows a χ^2 distribution with d degrees of freedom. Therefore, if an estimator is consistent, we expect that the average NEES for the target state will be close to $2N$ for all k . The larger the deviations of the NEES from these values, the worse the inconsistency of the filter. By studying both metrics, we obtain a comprehensive picture of the performance of the filters under consideration.

In this test, we performed 100 Monte-Carlo simulations and compared three different PFs. During each Monte-Carlo run, all filters process the same data to ensure a fair comparison: (i) the standard (bootstrap) PF with **1000** particles, which uses the prior as the proposal distribution to draw particles from and employs the systematic resampling strategy [7]; (ii) the GSPF [13] using the same number of particles as the standard PF and $m = 10$ Gaussians to represent the underlying distributions, which are initialized by clustering the initial particles into m groups and then fitting a Gaussian to each group; and (iii) the proposed AGS-PF with **500** particles, which uses the analytically-determined proposal distribution as well as systematic resampling. In the AGS-PF, at each time step, we clustered the particles into $m = 10$ groups, equal to the number of Gaussians used in the GSPF. Note that in order to validate the effectiveness of the analytically-determined proposal distribution employed by the AGS-PF, significantly fewer particles were used in the AGS-PF, as compared to the standard PF and the GSPF. Despite this fact, as shown below, the AGS-PF attains substantially better performance than both the standard PF and the GSPF.

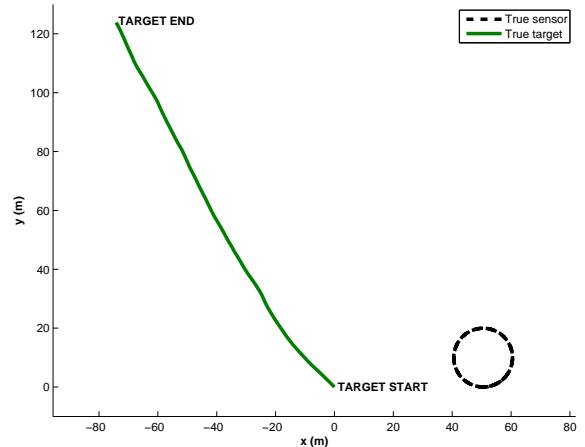


Fig. 2. The trajectories of the target and sensor obtained from one typical realization of the 100 Monte-Carlo simulations.

For the results presented in this section, we adopted a zero-acceleration motion model for the target [8]:

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{w}(t) \quad (34)$$

where

$$\mathbf{x}(t) = \begin{bmatrix} x_T(t) \\ y_T(t) \\ \dot{x}_T(t) \\ \dot{y}_T(t) \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and $\mathbf{w}(t) = [w_x(t) \ w_y(t)]^T$ is zero-mean, white Gaussian noise with covariance $\mathbb{E}[\mathbf{w}(t)\mathbf{w}(\tau)^T] = q\mathbf{I}_2\delta(t - \tau)$, where $q = 1 \left(\frac{\text{m}}{\text{sec}^2}\right)^2 \frac{1}{\text{Hz}}$, and $\delta(t - \tau)$ is the Dirac-delta function. In our implementation, we discretize this continuous-time system model (34) with time step $\Delta t = 0.1$ sec. The initial true target state is $\mathbf{x}_0 = [0 \ 0 \ -5 \ 5]^T$, while the initial estimate of the target state is randomly generated from a Gaussian pdf, $\mathcal{N}(\mathbf{x}_0, \mathbf{P}_{0|0})$, where $\mathbf{P}_{0|0} = 100\mathbf{I}_4$ is the initial covariance of the state estimate. Similarly to [17], [21], we chose a circular sensor trajectory with perfectly known sensor positions for the simulations. Fig. 2 shows the trajectories of the target and sensor in one typical realization of the Monte-Carlo simulations. The standard deviation of the distance-measurement noise was equal to $\sigma = 0.5$ m.

Fig. 3 shows the Monte-Carlo results of the three PFs. It is clear that the standard PF provides inaccurate estimates which are diverging from the ground truth and become inconsistent. As explained before, the poor NEES performance of the standard PF is primarily due to the ill conditioning of the covariances computed from particles whose weights are small or particles that do not span all directions of the state space. As expected, the GSPF performs more accurately than the standard PF. Most importantly, the AGS-PF performs better than both the standard PF and the GSPF, in terms of accuracy (RMSE) and consistency (NEES). This is attributed to the analytically-determined proposal distribution which matches all the modes of the posterior pdf while taking into account the most recent measurements.

Finally, using the same simulation setup as described above, we compared the computational requirements of the

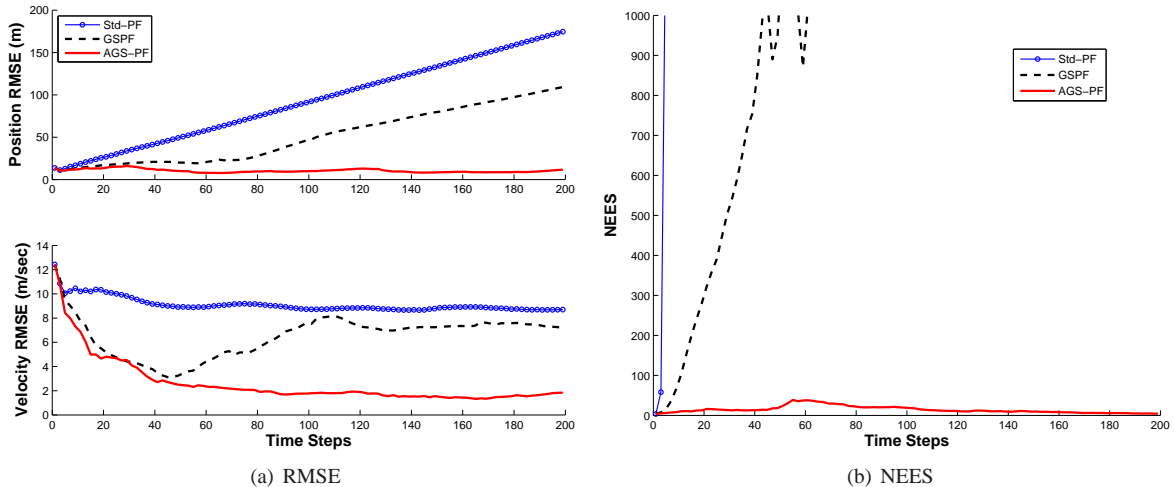


Fig. 3. Monte-Carlo simulation results: The proposed AGS-PF algorithm performs substantially better than its competitors, in terms of both accuracy (RMSE) and consistency (NEES). Note that for clarity of presentation, only the portions of the NEES lines that are within a certain threshold are plotted.

TABLE I
COMPUTATIONAL COST VS. ESTIMATION ACCURACY

	Runtime (sec)	Pos. Est. Err. (m)	Vel. Est. Err. (m/sec)
Std-PF	0.2273	92.3522	9.1060
GSPF	0.1801	51.4123	6.5045
AGS-PF	0.1813	10.3799	2.5885

proposed AGS-PF and its competitors by measuring the CPU running time for a complete update of all filters. Our Matlab implementations running on a Core i7 CPU of 2.67 GHz required an average execution time for each filter shown in Table I. These results were obtained by averaging the CPU runtime over all Monte-Carlo simulations and over all time steps. As compared to the standard PF and the GSPF, the proposed AGS-PF is not only computationally more efficient by using fewer particles but also achieves significantly better tracking performance [see Fig. 3(a), and Table I]. Specifically, as compared to the standard PF and the GSPF, the AGS-PF attains on average 89% and 80% reduction in position estimation error, 71% and 60% reduction in velocity estimation error, while at 20% lower and same order of computational cost, respectively.

V. EXPERIMENTAL RESULTS

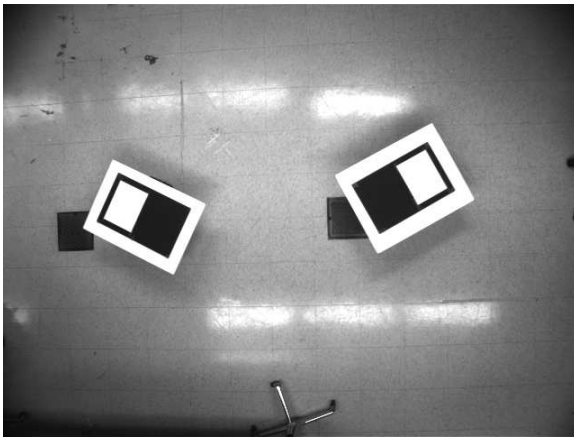
In this section, we present a real-world experiment conducted in a controlled environment to further validate the proposed AGS-PF. During the test, two Pioneer-III robots, one acting as the target and the other serving as the sensor, moved in a rectangular area of $4 \text{ m} \times 2 \text{ m}$, within which the positions of the robots were tracked by an overhead camera. For this purpose, rectangular tracking patterns were mounted on top of the robots and the vision system was calibrated in order to provide ground-truth measurements of the robots' poses in a global coordinate frame. The standard deviation of the noise in these measurements was approximately 0.5 deg for orientation and 0.01 m, along each axis, for position. The target robot drove along a straight line at a constant velocity of $v = 0.1 \text{ m/sec}$, and thus a zero-acceleration

motion model with $q = 0.05 \left(\frac{\text{m}}{\text{sec}^2}\right)^2 \frac{1}{\text{Hz}}$ was used to describe this motion [see (34)], while the sensor robot moved on a circle. Fig. 4(a) shows the experimental setup, and Fig. 4(b) depicts the trajectories of the target and the sensor.

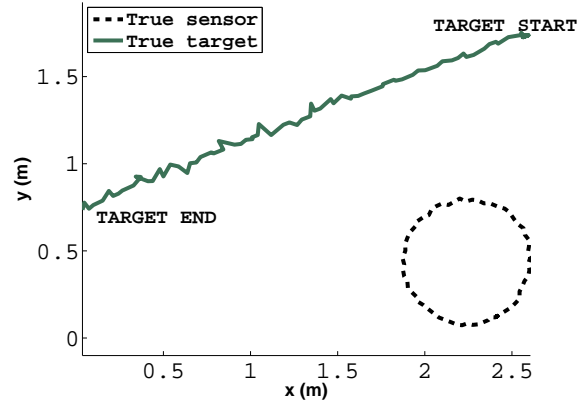
In this experiment, the initial estimate of the target state was set to $\hat{x}_{0|0} = [2.5940 \ 1.7374 \ 0.0003 \ -0.0001]^T$ with covariance $\mathbf{P}_{0|0} = \mathbf{I}_4$. Relative distance measurements were produced synthetically using the differences in the true positions of the target and the sensor, as these were recorded by the overhead camera, with the addition of noise. For the results shown in this section, the distance measurements were corrupted by zero-mean white Gaussian noise, with standard deviation $\sigma = 0.1 \text{ m}$. The same three PFs as in the previous simulation were implemented, and the comparative results obtained from this single-run experiment are presented in Fig. 5. From the experimental results, it becomes clear that the proposed AGS-PF outperforms the standard PF and the GSPF, in terms of both accuracy (RMSE) and consistency (NEES), which agrees with the simulation results presented in the previous section.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced a new AGS-PF, which uses a Gaussian mixture as the proposal distribution, where each Gaussian corresponds to one of the analytically-computed modes of the posterior pdf. Using such proposal distribution, the AGS-PF draws its particles within the most probable regions of the state space. As a result, as compared to the standard PF, the AGS-PF attains better performance while requiring fewer computational resources. We applied this algorithm to range-only target tracking, although the proposed methodology is applicable to a broad class of problems which can be transformed into polynomial form. Simulation and experimental results have demonstrated that the proposed approach significantly outperforms the standard PF and the GSPF, in terms of accuracy, consistency and efficiency. In the future work, we plan to extend the proposed AGS-PF to target tracking in 3D and investigate further improvements for reducing the number of particles.

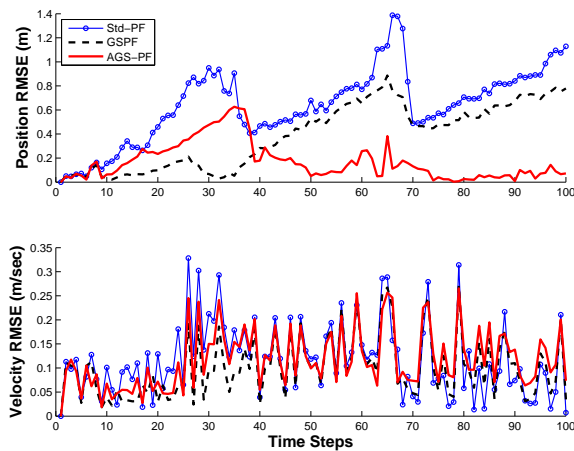


(a)

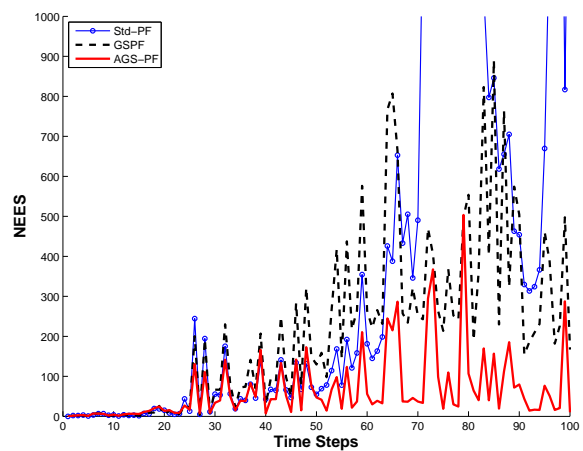


(b)

Fig. 4. Experimental setup: (a) Calibrated image of two Pioneer III robots (one acts as the target while the other is the sensor) with tracking patterns mounted on top of them. (b) Trajectories of the two robots (target and sensor) that move inside a $4\text{ m} \times 2\text{ m}$ arena during the indoor experiment.



(a) RMSE



(b) NEES

Fig. 5. Experimental results: The proposed AGS-PF algorithm performs better than its competitors, in terms of both accuracy (RMSE) and consistency (NEES). Note that for clarity of presentation, only the portions of the NEES lines that are within a certain threshold are plotted.

REFERENCES

- [1] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
- [2] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.
- [3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [4] A. Doucet and A. M. Johansen, *A tutorial on particle filtering and smoothing: fifteen years later*. University Press, 2009.
- [5] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, Jul. 2010.
- [6] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.
- [7] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F: Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, Apr. 1993.
- [8] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001.
- [9] R. van der Merwe, A. Doucet, N. De Freitas, and E. Wan, "The unscented particle filter," in *Proc. of the Advances in Neural Information Processing Systems*, 2001, pp. 584–590.
- [10] Y. Rui and Y. Chen, "Better proposal distributions: object tracking using unscented particle filter," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2001, pp. 786–793.
- [11] F. Wang, Y. Lin, T. Zhang, and J. Liu, "Particle filter with hybrid proposal distribution for nonlinear state estimation," *Journal of Computers*, vol. 6, no. 11, pp. 2491–2501, Nov. 2011.
- [12] M. Montemerlo, "FastSLAM: A factored solution to the simultaneous localization and mapping problem with unknown data association," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Jul. 2003.
- [13] J. H. Kotecha and P. M. Djuric, "Gaussian sum particle filtering," *IEEE Trans. on Signal Processing*, vol. 51, no. 10, pp. 2602–2612, Oct. 2003.
- [14] —, "Gaussian particle filtering," *IEEE Trans. on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.
- [15] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Dover Publications, 2005.
- [16] D. Cox, J. Little, and D. O'Shea, *Using Algebraic Geometry*. Springer, 2005.
- [17] G. P. Huang, K. X. Zhou, N. Trawny, and S. I. Roumeliotis, "A bank of MAP estimators for single-sensor range-only target tracking," in *Proc. of the American Control Conference*, Baltimore, MD, Jun. 30 – Jul. 2, 2010, pp. 6974–6980.
- [18] G. H. Golub and C. F. V. Loan, *Matrix Computations*. The Johns Hopkins University Press, 1996.
- [19] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.
- [20] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley-Interscience, 2000.
- [21] A. Farina, "Target tracking with bearing-only measurements," *Signal Processing*, vol. 78, pp. 61–78, Oct. 1999.