An Auxiliary Particle Filtering Algorithm With Inequality Constraints

Baibing Li, *Member, IEEE*, Cunjia Liu, *Member, IEEE*, and Wen-Hua Chen, *Senior Member, IEEE*

***Abstract*—For nonlinear non-Gaussian stochastic dynamic systems with inequality state constraints, this technical note presents an efficient particle filtering algorithm, constrained auxiliary particle filtering algorithm. To deal with the state constraints, the proposed algorithm probabilistically selects particles such that those particles far away from the feasible area are less likely to propagate into the next time step. To improve on the sampling efficiency in the presence of inequality constraints, it uses a highly effective method to perform a series of constrained optimization so that the importance distributions are constructed efficiently based on the state constraints. The caused approximation errors are corrected using the importance sampling method. This ensures that the obtained particles constitute a representative sample of the true posterior distribution. A simulation study on vehicle tracking is used to illustrate the proposed approach.**

***Index Terms*—Auxiliary particle filter, Bayesian inference, inequality constraints, sequential Monte Carlo, state space models.**

I. Introduction

In practice, nonlinear stochastic dynamic systems are often restricted to a sub-area of the state space. This is usually the consequence of some physical restrictions on the systems of interests. For example, in vehicle tracking, vehicles on a road are expected to move within the boundaries of the road and to follow the speed limit (see, e.g., [1]). Because state constraints reduce the variability of the state vector, incorporating the constraint-related information into state estimation can usually improve on the accuracy of state filtering [2].

State estimation with state constraints is very challenging and has attracted a number of researchers. With the conventional filtering approaches such as the extended Kalman filter (EKF) and unscented Kalman filter (UKF), many approaches have been developed to deal with state constraints but most of them focus on linear and/or equality constraints. This includes the approaches of reparameterizing, pseudo-measurement, optimization, projection and truncation. Recently a truncation-based method has been developed in [3]. It is based on the above conventional filters but a novel Monte Carlo method is used to deal with the inequality constraints. Moving horizon estimator (MHE), on the other hand, performs state estimation by solving an optimization problem that can inherently handle constraints. However, the multi-stage nonlinear optimization in MHE incurs excessive computational burden for online applications [4]. See [5] for an overview on the conventional filtering approaches with state constraints.

In the literature, there are only a few particle filters developed for dealing with state constraints, among which the pioneering research work in [6] and [7] is particularly worth noting. In [6], a simple algorithm using the acceptance-rejection method was developed. The acceptance-rejection method is commonly used in offline Monte Carlo methods where computation time is not an issue. However, it is in general not suitable for online applications because the time required for drawing a particle that satisfies the state constraints could be prohibitively long. To circumvent this problem, a novel approach was proposed in [7] that consists of two stages. First, particle candidates are drawn without using the state constraints; then the candidates that violate the constraints are projected into the feasible area defined by the constraints using a series of optimization. This method incurs a high computational cost and the particles obtained using the deterministic projection are no longer a representative sample of the posterior distribution of the state vector.

Recently, [8] has made some solid advances in particle filtering with state constraints and proposed three algorithms. The first method is a Gibbs-sampling-based algorithm that is more relevant to this technical note as the second method in [8] is based on the acceptance-rejection and the posterior updating in the third method does not use the constraint information. To draw each particle at each time step, a Gibbs sampler is embedded in the Gibbs-sampling-based algorithm so that only one variable is considered in each iteration of the Gibbs sampler. As a result, sampling within the state space reduces to a univariate problem, i.e. sampling within an interval, that can be dealt with straightforwardly. This, however, comes at a price of high computational demand. This is because the Gibbs sampler is an iterative procedure that is usually used as an offline method. Embedding a Gibbs sampler for generating particles incurs excessive computation costs. Moreover, within each iteration of the Gibbs sampler, it needs to solve the state constraints for the end-points of the constraint interval of each state variable. This has a high computational demand when the analytical solutions are not available.

The Gibbs-sampling-based algorithm also has some restrictive requirements on the systems under investigation. It is required that: (a) both the conditional cumulative distribution of each state variable and its inverse function can be evaluated

Manuscript received June 23, 2016; revised October 6, 2016; accepted October 28, 2016. Date of publication November 14, 2016; date of current version August 28, 2017. This work was supported by the U.K. Engineering and Physical Sciences Research Council (EPSRC) Autonomous and Intelligent Systems programme (grant number EP/J011525/1) with BAE Systems as the leading industrial partner. Recommended by Associate Editor S. S. Saab.

B. Li is with the School of Business and Economics, Loughborough University, Loughborough LE11 3TU, U.K. (e-mail: b.li2@lboro.ac.uk).

C. Liu and W.-H. Chen are with the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough LE11 3TU, U.K. (e-mail: c.liu5@lboro.ac.uk; w.chen@lboro.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2016.2624698

analytically; (b) these conditional cumulative distributions can be sampled easily; and (c) the constraint region for the univariate sampling is a single interval. In practice these assumptions may not be valid. Take requirement (c) as an example: the obtained constraint region may consist of several non-overlapped intervals because a nonlinear constraint function in general leads to more than one solution.

In this technical note, we propose a new method, constrained auxiliary particle filtering algorithm (CAPFA). Similar to the ordinary auxiliary particle filter (APF), the CAPFA consists of resampling and sampling stages at each time step. In the resampling stage, it probabilistically selects particles such that particles with a lower likelihood of output measurements and/or far away from the feasible area are less likely to survive and propagate into the next time step. For this end, we carry out a series of constrained optimization to work out the center (as measured by the mode) of each transition distribution within the feasible area from which the particles are to be drawn. To improve on the sampling efficiency in the sampling stage, we form each importance distribution in a way such that it is centered at the mode of the corresponding restricted transition distribution. In doing so, the chance of drawing a feasible particle is much higher as the mode lies within the feasible area. Finally we use the importance sampling method to correct the errors caused in the previous stages.

In comparison with [8], the proposed method is computationally more efficient and hence suitable for online applications. Its assumptions on the system are also less restrictive and hence it can be applied to a wider range of problems.

This technical note is structured as follows. Section II is devoted to problem formulation. In Section III we present the main results and in Section IV we conduct a simulation study for algorithm evaluation. Finally we conclude this technical note in Section V.

II. Problem Formulation

Consider a nonlinear stochastic dynamic system:

$$x_{k+1} = f_k(x_k, u_k) + \omega_k \tag{1}$$

with the observation equation

$$z_k = h_k(x_k) + \upsilon_k, \tag{2}$$

where $x_k = [x_{1,k}, \ldots, x_{n,k}]^T \in \Re^n$, $u_k = [u_{1,k}, \ldots, u_{m,k}]^T \in \Re^m$, and $z_k = [z_{1,k}, \ldots, z_{l,k}]^T \in \Re^l$ are the vectors of state variables, inputs, and outputs at time step k $(k = 0, 1, 2, \ldots)$ respectively. $f_k : \Re^n \times \Re^m \to \Re^n$ is a vector of possibly nonlinear functions of x_k and u_k, and $h_k : \Re^n \to \Re^l$ is a vector of possibly nonlinear functions of x_k. The process noise $\omega_k \in \Re^n$ and measurement noise $\upsilon_k \in \Re^l$ are assumed to be absolutely continuous random vectors with a zero mean and mutually independent. Their probability density functions are denoted as $p_\omega(\omega_k)$ and $p_\upsilon(\upsilon_k)$ which may be time-dependent but for notational simplicity the subscript for time step k is suppressed. We assume that $p_\omega(\omega_k)$ is log-concave and has continuous second-order partial derivatives. The initial state x_0 is independent of ω_k and υ_k with a known probability density function $p(x_0)$. Let $x_{0:k} = \{x_0, x_1, \ldots, x_k\}$ and $z_{1:k} = \{z_1, \ldots, z_k\}$ denote the set of all states and the set of all output measurements up to time k respectively.

We denote the distribution of x_k conditional on x_{k-1} by $p(x_k|x_{k-1})$ which can be derived from (1). Likewise, we denote the distribution of z_k conditional on x_k by $p(z_k|x_k)$ which can be derived from (2); see, e.g., [9] and [10] for further discussion.

Suppose that x_k satisfies the following constraints:

$$g_k(x_k) \le 0, \tag{3}$$

where $g_k = [g_{1,k}, \ldots, g_{M,k}]^T : \Re^n \to \Re^M$ is a vector of possibly nonlinear functions of x_k. Each $g_{j,k}(x_k)$ is assumed to have continuous second-order partial derivatives. Let $\mathcal{B}_k = \{x_k | g_k(x_k) \le 0\}$ denote the feasible area of x_k that satisfies the constraints. Following the literature (e.g., [2], [10]), the inequality (3) is treated as pseudo-measurements:

$$y_k = \begin{cases} 1 & \text{if} \quad x_k \in \mathcal{B}_k; \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Let $p_g(y_k|x_k) = \mathbb{I}\{x_k \in \mathcal{B}_k\}$ denote the unnormalized density function of the pseudo-measurements y_k, where $\mathbb{I}\{x_k \in \mathcal{B}_k\}$ is an indicator function, having 1 if $x_k \in \mathcal{B}_k$ and 0 otherwise.

III. Main Results

We aim to obtain the posterior distribution $p(x_{0:k}|z_{1:k})$ of the sequence of state vector $x_{0:k}$ at each time step k for given observation sequence $z_{1:k}$, the initial state vector x_0, constraints (3), and knowledge about system (1)–(2).

When the system is nonlinear/non-Gaussian, usually $p(x_{0:k}|z_{1:k})$ is analytically intractable, and hence a numerical solution is required in practice. Theoretically this can be obtained by drawing a representative sample of size N from $p(x_{0:k}|z_{1:k})$, termed particles. Unfortunately, for many complex practical problems, it is usually impossible to draw such a sample directly from $p(x_{0:k}|z_{1:k})$. In the particle filtering, importance sampling is used to circumvent this difficulty, where the particles $x_{0:k}^i$ $(i = 1, \ldots, N)$ are drawn from an importance distribution $\pi(x_{0:k}|z_{1:k})$ instead of $p(x_{0:k}|z_{1:k})$ *per se*. A correction step with a set of weights w_k^i $(i = 1, \ldots, N)$ is required to ensure that the obtained particles constitute a representative sample of the true posterior distribution:

$$p(x_{0:k}|z_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_{0:k} - x_{0:k}^i), \tag{5}$$

where $\delta(.)$ is the Dirac measure and the importance weights are given by $w_k^i \propto p(x_{0:k}^i|z_{1:k}) / \pi(x_{0:k}^i|z_{1:k})$. The importance weights need to be normalized, i.e. w_k^i are replaced with $w_k^i / \sum_{j=1}^{N} w_k^j$, so that the normalized weights sum to one.

The approximation in (5) is based on the law of large number. A common problem with the particle filtering is the degeneracy phenomenon, where after a few iterations, all but one particle will have a negligible weight, leading to a very poor approximation to the true posterior distribution in (5) and hence poor quality of statistical inference for the state vector. The degree of degeneracy is characterized by the effective sample size (ESS) defined to be $N_{\text{eff}} = N/[1 + \text{Var}(w_k(p, \pi))]$, where $w_k(p, \pi) = p(x_{0:k}|z_{1:k})/\pi(x_{0:k}|z_{1:k})$ is the true weight function [11]. In numerical computation, ESS is estimated by $\hat{N}_{\text{eff}} = [\sum_{i=1}^{N}(w_k^i)^2]^{-1}$. When designing algorithms for particle filtering, one should endeavour to ensure that the ESS is as large as possible [12].

A. Why Use Inequality Constraints in State Filtering?

Following [2] and [10], we consider a dynamic system (1) and treat the constraints (3) as pseudo measurements. During the filtering, the pseudo measurements (4) together with the output measurements (2) will be used to form the likelihood function to draw inference about the state vector. We will show that, for any importance distribution for which state constraints (3) are not taken into account, one can always construct a modified importance distribution so that the particles drawn from the modified importance distribution can better represent the posterior distribution. For ease of exposition, the measurements $z_1, \ldots, z_k$ are not written explicitly and the subscript for time step k is dropped in this subsection.

Consider an approximate posterior distribution which pools the prior information and the observation equation (2) without consideration of constraints (3). Let $\tilde{p}(x)$ with $\int \tilde{p}(x)\,\mathrm{d}x = 1$ denote the probability density function of this approximate posterior distribution. Suppose that an importance distribution $\tilde{\pi}(x)$ with $\int \tilde{\pi}(x)\mathrm{d}x = 1$ is used to draw the particles.

From Bayes' theorem, the true posterior probability density function where the pseudo-measurements are taken into consideration is given by $p(x|y) = c_0\tilde{p}(x)p_g(y|x)$ with $c_0 = [\int \tilde{p}(x)p_g(y|x)\,\mathrm{d}x]^{-1}$. Now we take into account the constraints to construct a modified importance distribution: $\pi(x) = c\tilde{\pi}(x)p_g(y|x)$, where $c = [\int \tilde{\pi}(x)p_g(y|x)\,\mathrm{d}x]^{-1} \geq 1$. The corresponding weight function is $w(x) = p(x|y)/\pi(x)$. In addition, when using the importance distribution $\tilde{\pi}(x)$, the weight function is $\tilde{w}(x) = p(x|y)/\tilde{\pi}(x)$. The following theorem shows that $\text{Var}[w(x)] \leq \text{Var}[\tilde{w}(x)]$.

Theorem 1: Let $p_g(y|x)$ be the probability density function of the pseudo-measurements defined in (4), and $\tilde{p}(x)$ and $\tilde{\pi}(x)$ be the approximate posterior distribution and the corresponding importance distribution where the constraints are ignored. Let $p(x|y) = c_0\tilde{p}(x)p_g(y|x)$ with $c_0 = [\int \tilde{p}(x)p_g(y|x)\,\mathrm{d}x]^{-1}$ be the true posterior distribution where the constraints are taken into consideration. Then one can always construct a modified importance distribution $\pi(x) = c\tilde{\pi}(x)p_g(y|x)$ where $c = [\int \tilde{\pi}(x)p_g(y|x)\,\mathrm{d}x]^{-1}$, with weight $w(x) = p(x|y)/\pi(x)$, such that $\text{Var}[w(x)] \leq \text{Var}[\tilde{w}(x)]$, where $\tilde{w}(x) = p(x|y)/\tilde{\pi}(x)$ is the weight function of the importance distribution $\tilde{\pi}(x)$.

Proof: First, we show $\mathrm{E}w^2 \leq \mathrm{E}\tilde{w}^2$. We note

$$\mathrm{E}w^2 = \int w^2\pi(x)\,\mathrm{d}x = \frac{c_0^2}{c}\int \frac{\tilde{p}^2(x)p_g(y|x)}{\tilde{\pi}(x)}\,\mathrm{d}x,$$

$$\mathrm{E}\tilde{w}^2 = \int \tilde{w}^2\tilde{\pi}(x)\,\mathrm{d}x = c_0^2\int \frac{\tilde{p}^2(x)p_g^2(y|x)}{\tilde{\pi}(x)}\,\mathrm{d}x.$$

Since $p_g^2(y|x) \equiv p_g(y|x)$ and $c \geq 1$, we obtain $\mathrm{E}w^2 \leq \mathrm{E}\tilde{w}^2$. Then it follows by noting $\mathrm{E}w = \mathrm{E}\tilde{w} = 1$. ■

From Theorem 1, by taking into account the state constraints during the sampling, the weights are always more evenly distributed.

Corollary 1: Let N_{eff} and $\tilde{N}_{\text{eff}}$ denote the ESS values corresponding to weight functions $w(x)$ and $\tilde{w}(x)$ respectively. Then under the assumptions of Theorem 1, we have $N_{\text{eff}} \geq \tilde{N}_{\text{eff}}$.

The proof is immediate from Theorem 1. Corollary 1 indicates that the ESS is always larger when the state constraints are taken into consideration. Because particle filtering is based on the law of large numbers, Corollary 1 has important implications: a smaller ESS of particles will in general lead to a poorer representation of the posterior distribution with inferior quality of state estimation.

Now define $G_{\text{eff}} = 100 \times [(N_{\text{eff}} - \tilde{N}_{\text{eff}})/\tilde{N}_{\text{eff}}]\%$ as the gain in ESS when the state constraints are taken into account in the particle filtering. The benefits of using the constraint-related information in particle filters can be measured as follows.

Corollary 2: Under the assumptions of Theorem 1, the gain in ESS by using the constraint-related pseudo-measurement information in the particle filtering is given by

$$G_{\text{eff}} = 100(c - 1)\%,$$

where $c = [\int \tilde{\pi}(x)p_g(y|x)\,\mathrm{d}x]^{-1} \geq 1$.

Proof: Let $a = c_0^2 \int [\tilde{p}^2(x)p_g(y|x)/\tilde{\pi}(x)]\,\mathrm{d}x$. From the proof of Theorem 1, we note that $\text{Var}[w(x)] = c^{-1}a - 1$ and $\text{Var}[\tilde{w}(x)] = a - 1$. Then by definition, we obtain $N_{\text{eff}} = cN/a$ and $\tilde{N}_{\text{eff}} = N/a$. ■

When the state constraints are accounted for in particle filtering, Corollary 2 shows that the smaller the feasible area is, the more the gain in ESS can be obtained when the state constraints are taken into account.

B. A Brief Summary of the Auxiliary Particle Filter

Consider system (1)–(2) without any constraints. The posterior distribution at time k can be derived using Bayes' rule:

$$p(x_{0:k}|z_{1:k}) \propto \sum_{i=1}^{N} w_{k-1}^i p(z_k|x_k)\,p(x_k|x_{k-1}^i). \tag{6}$$

In the sequential Monte Carlo particle filtering, the importance weights for system (1)–(2) is updated recursively: $w_k^i \propto w_{k-1}^i \;\; p(z_k|x_k^i)p(x_k^i|x_{k-1}^i) \;\; /\pi(x_k^i|x_{0:k-1}^i, z_{1:k})$, where $\pi(x_k|x_{0:k-1}, z_{1:k})$ is an importance distribution. A commonly used importance distribution is $\pi(x_k|x_{0:k-1}, z_{1:k}) = p(x_k|x_{k-1})$; the weight updating equation reduces to $w_k^i \propto w_{k-1}^i p(z_k|x_k^i)$ for this choice. This importance distribution is chosen in a way that does not take into account the output measurements $z_{1:k}$. Hence, it may be inefficient since the state space is explored without any knowledge of the observations.

A distinctive nature of the APF is that its importance distributions incorporate such information. The auxiliary particle filter (APF) was initially proposed in [13] and [14]. The APF algorithm draws particles on the basis of (6) in two stages [12], [13]: (a) draw indices I_i with the particle selection probabilities proportional to $w_{k-1}^i p(z_k|x_{k-1}^i)$; and (b) draw each particle x_k^i from $p(x_k|x_{k-1}^{I_i})$ for given index I_i.

The predictive likelihood $p(z_k|x_{k-1}^i)$ is defined to be $\int p(z_k|x_k)p(x_k|x_{k-1}^i)\,\mathrm{d}x_k$. In many applications, it is difficult to evaluate this predictive likelihood. One simple solution is to use the approximation $p(z_k|\lambda_k^i)$, where λ_k^i is the center (as measured by mean, median, or mode) of $p(x_k|x_{k-1}^i)$ (e.g., [9]). Note that when $p_\omega(\omega_{k-1})$ is Gaussian, $p(x_k|x_{k-1}^i)$ is also Gaussian. In this case, the mean, median, and mode of $p(x_k|x_{k-1}^i)$ are all equal, and the centers of the importance distributions can be obtained straightforwardly, i.e. $\lambda_k^i = f_{k-1}(x_{k-1}^i, u_{k-1})$. A correction via the importance weights $w_k^i = p(z_k|x_k^i)/p(z_k|\lambda_k^{I_i})$ is required to ensure the set of obtained particles represents the true posterior distribution.

The APF can be regarded as an interchange of the importance sampling and resampling procedures in the generic particle filtering (see, e.g., [12], [15]): the resampling stage is performed prior to the sampling stage. Consequently only those particles with a higher particle selection probability are likely to survive and propagate into the next step.

C. Design of the CAPFA

We now take into account state constraints and develop a new filtering algorithm CAPFA based on the APF. The design of the CAPFA will take advantages of the APF to improve on the sampling efficiency in the presence of state constraints.

Following [2], [10], we consider the inequality constraints (3) as pseudo-measurements with the density function $p_g(y_k|x_k)$. This leads to a generalized likelihood function $p(z_k|x_k)p_g(y_k|x_k)$. Applying Bayes' rule to pool this generalized likelihood with the prior distribution $p(x_k|x_{k-1})$, we obtain the posterior distribution of $x_{0:k}$ at time step k:

$$p(x_{0:k}|z_{1:k}) \propto \sum_{i=1}^{N} w_{k-1}^i p(z_k|x_k)p_g(y_k|x_k)\,p(x_k|x_{k-1}^i). \quad (7)$$

The key idea of the CAPFA is to select particles to propagate into the next time step by using the constraint-related information. Bearing in mind that the constraints are now treated as pseudo-measurements that modify the likelihood to be $p(z_k|x_k)p_g(y_k|x_k)$, we follow the design of the APF and select the particles in the CAPFA with probabilities proportional to this modified likelihood multiplied by the corresponding weight w_{k-1}^i. In addition, similar to the APF, the unknown x_k at this stage is replaced with the center of the restricted transition distribution $p_g(y_k|x_k)\,p(x_k|x_{k-1}^i)$, denoted as λ_k^i. In summary, the particles in the resampling stage of the CAPFA are selected with probabilities proportional to $\psi_k^i \triangleq w_{k-1}^i p(z_k|\lambda_k^i)\, p(\lambda_k^i|x_{k-1}^i)$.

The overall structure of the CAPFA is designed in a similar way to the APF: (a) in the resampling stage: we find the centers λ_k^i of the restricted transition distributions $p_g(y_k|x_k)\,p(x_k|x_{k-1}^i)$, and draw indices I_i with the particle selection probabilities proportional to ψ_k^i; and (b) in the sampling stage: we draw a particle from each $p_g(y_k|x_k)p(x_k|x_{k-1}^{I_i})$ for given I_i. These two issues will be investigated in the next two subsections.

D. Resampling Stage: Locating the Center of Each $p_g(y_k|x_k)\,p(x_k|x_{k-1}^i)$

Mathematically, the mode λ_k^i is defined to be the point within the feasible area $\mathcal{B}_k$ that minimises $-\log[p(x_k|x_{k-1}^i)]$ $(i = 1, \ldots, N)$. At the first glance, the task of searching the mode λ_k^i looks daunting as N is usually large. However, there is no need to get the exact solution for each optimization problem because the solution is used only for forming an importance distribution, and any approximation error can be corrected using the weight correction method in the importance sampling. In our experience, one-step iteration will usually suffice; the benefit of using more iterations, if any, is marginal. See the simulation study in Section IV for details.

Specifically, we define $f_k^i \triangleq f_{k-1}(x_{k-1}^i, u_{k-1})$. Let D_k^i and V_k^i denote the first and second derivatives of $-\log[p(x_k|x_{k-1}^i)]$ evaluated at f_k^i. Now we expand $-\log[p(x|x_{k-1}^i)]$ about f_k^i to its second-order and use this second-order expansion to approximate the original optimization problem for λ_k^i.

We suppress subscript k for notational simplicity. The original optimization problem for λ^i can be approximated by a series of constrained optimization problems P_i $(i = 1, \ldots, N)$, each having a quadratic objective function:

$$\min_x (x - f^i)^T D^i + \frac{1}{2}(x - f^i)^T V^i (x - f^i)$$
$$\text{s.t.} \quad g_j(x) \le 0 \quad j = 1, \ldots, M. \quad (8)$$

For problems P_i $(i = 1, \ldots, N)$, if $f^i - (V^i)^{-1}D^i$ falls into the feasible area, it is the exact optimal solution. Let $\mathcal{A}$ denote all the indices i that $f^i - (V^i)^{-1}D^i$ is outside the feasible area, i.e. $\mathcal{A} = \{i | f^i - (V^i)^{-1}D^i \notin \mathcal{B}\}$. In the rest of this subsection, we only focus on problems P_i for $i \in \mathcal{A}$.

We use a simple interior-point method (SIPM) described below to solve problem (8) for $i \in \mathcal{A}$; see [16] for an overview on various interior-point methods. We choose a barrier function as $B(x) = -1/x$. Then solving problem (8) can be converted to solving the following problem for a suitably chosen $\gamma > 0$:

$$\min_x \left\{ (x - f^i)^T D^i + \frac{1}{2}(x - f^i)^T V^i (x - f^i) - \gamma \sum_{j=1}^{M} g_j^{-1}(x) \right\}. \quad (9)$$

Suppose that an initial value $x^{(0)} \in \mathcal{B}$ has been chosen. The Newton method can be used to solve (9) and it terminates after the first iteration. The one-step solution is taken as an approximate solution:

$$\lambda^i = x^{(0)} + \alpha^* d^i \quad (10)$$

where α^* is the step length. $d^i = -(H^i)^{-1}\nabla J^i|_{x=x^{(0)}}$ is the search direction and $H^i \triangleq \nabla^2 J^i$, where J^i denotes the objective function of problem (9).

There are two issues to be addressed: (a) the choice of step length α^*; and (b) the choice of the initial value $x^{(0)}$.

For the selection of α^*, we use the widely-used linearization method, i.e. we linearize constraint functions $g_j(x)$ and solve (8) with the linearized constraint functions. It is easy to show

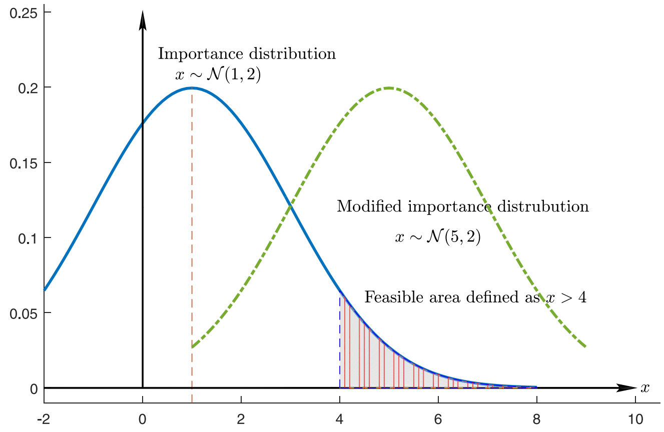

Fig. 1. Illustration of the CAPFA algorithm: the importance distribution is shifted from $N(1,2)$ to $N(5,2)$ such that its center falls into the the feasible area $[4,+\infty)$. The chance that the candidate particles drawn from $N(5,2)$ fall into $[4,+\infty)$ is higher.

that the optimal solution to this problem is given by

$$\alpha^* = \min_j \left\{ \frac{-g_j(x^{(0)})}{(d^i)^T \nabla g_j(x^{(0)})} \mid \text{for all } (d^i)^T \nabla g_j(x^{(0)}) > 0 \right\} \tag{11}$$

or $\alpha^* = 1$ if $x^{(0)} + d^i \in \mathcal{B}$.

Occasionally for the step length α^* obtained in (11), we have $\lambda^i = x^{(0)} + \alpha^* d^i \notin \mathcal{B}$. In this case, we shrink α^* by a preselected factor β $(0 < \beta < 1)$ for a couple of times until all the constraints are satisfied. As this is a linear search problem, we suggest using either the bisection method $\beta = 0.5$ or golden section method $\beta = 0.618$.

Since all the optimization problems P_i have a common feasible area, a common initial point $x^{(0)}$ can be used for all P_i. We suggest using the weighted mean or mode (for the case that the mean lies outside the feasible area) of all the feasible f^i as the initial point $x^{(0)}$. Specifically, let us re-introduce the subscript k for each time step for the clarity purposes. We define $\mathcal{C}_k = \{i | f_k^i \in \mathcal{B}_k\}$ and calculate $\bar{f}_k = \sum_{i \in \mathcal{C}_k} w_k^i f^i / \sum_{i \in \mathcal{C}_k} w_k^i$. If $\bar{f}_k \in \mathcal{B}_k$, then we choose $x_k^{(0)} = \bar{f}_k$; otherwise, we find $i^* = \arg\max_i (w_k^i | i \in \mathcal{C}_k)$ and choose $x_k^{(0)} = f_k^{i*}$.

E. Sampling Stage: Drawing Particles From Restricted Transition Distributions

Now we turn to consider how to efficiently sample from each unnormalized restricted transition distribution $p_g(y_k|x_k)\, p(x_k|x_{k-1}^{I_i})$.

A simple approach would be to use the acceptance-rejection method. However, the computational cost for this method could be prohibitively high if the center of $p(x_k|x_{k-1}^{I_i})$ is far away from $\mathcal{B}_k$. To address this issue, we use an importance-sampling-based method as follows. Consider any particle with index I_i. We construct an importance distribution $\pi(x_k|\lambda_k^{I_i})$ for the particle as $p(x_k|x_{k-1}^{I_i})$ with its mean translated to $\lambda_k^{I_i}$:

$$\pi(x_k|\lambda_k^{I_i}) = p(x_k - \lambda_k^{I_i} + f_{k-1}(x_{k-1}^{I_i}, u_{k-1})|x_{k-1}^{I_i}).$$

Hence, rather than to draw a particle from the unnormalized distribution $p_g(y_k|x_k)p(x_k|x_{k-1}^{I_i})$, we draw the particle from the importance distribution $\pi(x_k|\lambda_k^{I_i})$. Since $\pi(x_k|\lambda_k^{I_i})$ has the center of $\lambda_k^{I_i} \in \mathcal{B}_k$, it is more likely that the candidate particle drawn from $\pi(x_k|\lambda_k^{I_i})$ will fall into the feasible area, hence improving on the sampling efficiency, as illustrated in Fig. 1.

Algorithm 1: The CAPFA Algorithm.

```
Initialization;
for $k = 1 : K$ do
    set $\mathcal{A}_k = \{i | f_k^i - (V_k^i)^{-1} D_k^i \notin \mathcal{B}_k\}$;
    for $i = 1 : N$ do
        if $i \notin \mathcal{A}_k$ then
            set $\lambda_k^i = f_k^i - (V_k^i)^{-1} D_k^i$;
        else
            solve (8) for $\lambda_k^i$ using the SIPM;
        end
    end
    for $i = 1 : N$ do
        draw $I_i$ with probabilities proportional to $\psi_k^i$;
        draw $x_k^i$ from $\pi(x_k|\lambda_k^{I_i})$;
        set $w_k^i$ using (12) and (13);
        set $x_{0:k}^i = (x_{0:k-1}^{I_i}, x_k^i)$;
    end
    normalize $w_k^i = w_k^i / \sum_{i=1}^N w_k^i$;
    resample particles according to $w_k^i$;
end
```

Because particle x_k^i is drawn from $\pi(x_k|\lambda_k^{I_i})$, the importance weight needs to be adjusted by a factor

$$\rho_k^i = p_g(y_k|x_k^i) p(x_k^i|x_{k-1}^{I_i}) / \pi(x_k^i|\lambda_k^{I_i}) \tag{12}$$

to ensure that the obtained particles as a whole constitute a representative sample of the true posterior distribution. Combining ψ_k^i with ρ_k^i, we obtain the importance weight for each particle x_k^i $(i = 1, \ldots, N)$ in the CAPFA:

$$w_k^i = \frac{\rho_k^i p(z_k|x_k^i)}{p(z_k|\lambda_k^{I_i}) p(\lambda_k^{I_i}|x_{k-1}^{I_i})}. \tag{13}$$

Finally, due to the indicator function in (12), we resample the obtained particles. After the resampling, all particles fall into the feasible area with a non-zero weight. The CAPFA is outlined in Algorithm 1.

In some applications, the covariance matrix of distribution $p_\omega(\omega_k)$ is rank-deficient. If this is the case, we follow [17] (Appendix B) to transform the problem into a sub-space in which the random variables have a full-rank covariance matrix and to draw the relevant samples in the sub-space.

IV. Simulation Study

In this section, we evaluate the proposed CAPFA using a vehicle-tracking simulation study. Similar problems were also investigated in many previous studies such as [3].

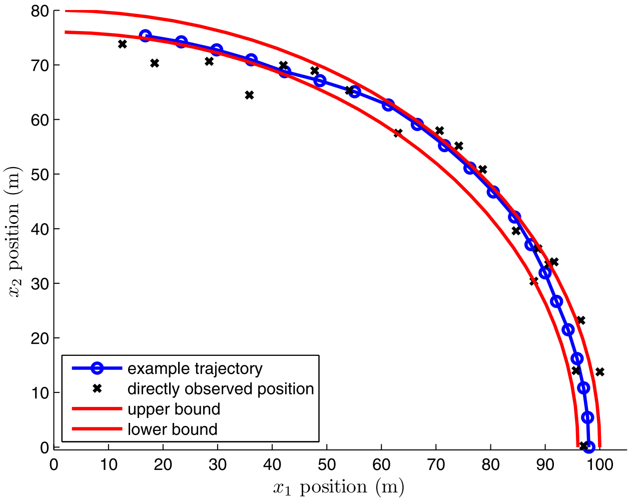

Fig. 2. Tracking scenario and the sample trajectory.

A. Simulation Settings

Consider a typical vehicle tracking problem with a moving vehicle traveling on a bend road section. The boundaries of the road are defined by two ellipse curves centred at the origin of a Cartesian coordinate system in the first quadrant with major radius of $a_1 = 96$ m and $a_2 = 100$ m, and minor radius $b_1 = 76$ m and $b_2 = 80$ m. The example vehicle trajectory is depicted in Fig. 2, where the vehicle starts from the position $(98, 0)$ and travels first in the central area of the road, and then moves close to the right and left part of the road respectively.

The vehicle dynamics are described by the following white noise acceleration motion model:

$$x_{k+1} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.5T^2 & 0 \\ 0 & 0.5T^2 \\ T & 0 \\ 0 & T \end{bmatrix} \omega_k \quad (14)$$

where the state vector $x_k = [x_{1,k} \quad x_{2,k} \quad \dot{x}_{1,k} \quad \dot{x}_{2,k}]^T$ consists of the vehicle position and velocity components in x_1 and x_2 directions, $T = 1$ s is the sampling interval, and ω_k is a two-dimensional Gaussian process noise with zero mean and identity covariance matrix. Given the road boundaries, the state constraints can be written as $(x_{1,k}/a_1)^2 + (x_{2,k}/b_1)^2 \leq 1$ and $(x_{1,k}/a_2)^2 + (x_{2,k}/b_2)^2 \geq 1$. The vehicle is tracked by a range and bearing sensor modeled as:

$$z_k = \begin{bmatrix} \sqrt{x_{1,k}^2 + x_{2,k}^2} \\ \arctan\left(\frac{x_{2,k}}{x_{1,k}}\right) \end{bmatrix} + \upsilon_k \quad (15)$$

where υ_k is a two-dimensional Gaussian noise vector with zero-mean and covariance matrix $R = \mathrm{diag}\{8, 0.002\}$.

For the above tracking problem, simulation experiments were carried out on a PC with 2.5 GHz CPU. 100 Monte Carlo runs were undertaken where different realisations of the noises were generated based on a pre-specified trajectory. In the state estimation, the initial prior distribution was chosen as a Gaussian distribution with mean $\hat{x}_0 = [95, 5, 0, 10]^T$ and covariance matrix $P_0 = \mathrm{diag}\{25, 25, 1, 1\}$.

TABLE I
PERFORMANCE OF CAPFA AVERAGED OVER 100 RUNS USING DIFFERENT OPTIMIZATION METHODS

Method(N)	MSE	SD	PESS (%)	PFP (%)	CT (s)
SIPM(150)	5.253	2.269	76.2	88.4	0.040
SIPM(250)	5.022	2.135	75.3	88.5	0.070
SIPM(500)	4.831	1.936	72.8	88.6	0.133
SQP(150)	5.162	2.216	77.3	82.5	1.227
SQP(250)	4.838	2.019	77.4	82.5	1.975

TABLE II
PERFORMANCE OF CAPFA WITH DIFFERENT LEVELS OF PARAMETER γ AVERAGED OVER 100 RUNS

γ	MSE	SD	PESS (%)	PFP (%)	CT (s)
5	5.014	2.123	73.7	88.5	0.068
10	5.005	2.130	75.3	88.5	0.075
20	5.022	2.135	75.3	88.5	0.070
40	4.988	2.104	76.0	88.5	0.068

The performance of the CAPFA was assessed in terms of accuracy measured by the average mean square error (MSE) of the position-related variables and the corresponding standard deviation (SD) of the MSE over the 100 runs; the latter characterizes the reliability and stability of the filtering results from realization to realization.

The quality of the obtained particles was measured by Percentage of ESS (PESS) defined as the ratio of ESS to N, Percentage of Feasible Particles (PFP) *prior to* the final resampling at each time step, and computation time (CT) measured by the time required in one run of the experiment.

B. The CAPFA With Different Optimization Methods

First, we compare the performance of the CAPFA using the one-step iteration SIPM with $\gamma = 20$ to that of the CAPFA where the optimization was undertaken using a Matlab standard solver, 'fmincon', which uses the Sequential Quadratic Programming (SQP) algorithm to solve a constrained optimization problem. The results are summarised in Table I.

It can be seen from Table I that, for the same particle number N, the benefit of using the standard 'fmincon' solver in the CAPFA was marginal in terms of average MSE, SD and PESS. The main drawback of using the standard optimization technique is the high computational cost, which may be problematic in some real-time applications where the state information needs to be updated rapidly. We also see that the benefit of the improved MSE using the 'fmincon' solver with $N = 150$ or $N = 250$ can be achieved by using the SIPM with a larger N (say $N = 500$) at a much lower computational cost. The case of $N = 500$ with the SQP optimization method is not reported here due to the high computational demand.

We also tried other choices for γ, i.e. $\gamma = 5, 10, 40$. The obtained results for $N = 250$ are displayed in Table II which indicates that the CAPFA is not sensitive to the choice of γ. This robustness property is important in practice. In the rest of the simulations in this section, we set $\gamma = 20$.

TABLE III
COMPARISON OF THE PERFORMANCES OF SEVERAL FILTERS AVERAGED OVER 100 RUNS

Filters	N	MSE	SD	PESS(%)	PFP(%)	CT(s)
APF([13])	150	7.141	2.195	42.7	44.5	0.009
APF([13])	250	7.051	2.852	42.7	44.4	0.014
APF([13])	500	6.770	2.028	43.1	44.9	0.028
CAPFA	150	5.253	2.269	76.2	88.4	0.040
CAPFA	250	5.022	2.135	75.3	88.5	0.070
CAPFA	500	4.831	1.936	72.8	88.6	0.133
CPF(alg.2 [7])	150	7.518	4.939	35.1	59.6	0.080
CPF(alg.2 [7])	250	6.371	2.723	34.6	58.5	0.090
CPF(alg.2 [7])	500	5.570	2.413	34.7	57.8	0.134
CPF(alg.3 [7])	150	7.330	3.769	38.6	59.7	0.080
CPF(alg.3 [7])	250	6.106	2.572	36.8	58.4	0.087
CPF(alg.3 [7])	500	5.529	2.436	36.2	57.7	0.152
CPF(alg.4 [7])	150	7.725	4.180	39.7	58.1	0.014
CPF(alg.4 [7])	250	6.291	2.768	37.9	57.5	0.023
CPF(alg.4 [7])	500	5.593	2.497	36.9	57.1	0.040
CPF(Gibbs [8])	150	5.728	2.335	54.3	100	1.743
CPF(Gibbs [8])	250	5.716	2.145	53.6	100	2.470
CPF(Gibbs [8])	500	5.605	2.104	54.1	100	4.685
tUPF([18])	150	8.262	3.759	57.5	100	0.039
tUPF([18])	250	7.738	3.389	57.1	100	0.042
tUPF([18])	500	7.667	3.343	56.1	100	0.049
tUPF ([19])	500	5.562	2.634	7.3	100	0.750

C. Measuring the Benefits of Employing State Constraints

Now we gauge the gain of using the state constraints in the particle filtering. We compared the APF with and without the constraints during the sampling. The numerical results are displayed in the first two panels of Table III.

The vehicle was required to move within the road. The PFP values in Table III, however, show that about 55% of the APF particles fell outside the road. Hence, the ordinary APF may provide biased information on the position of the vehicle.

The PFP of the CAPFA is about 88% prior to the final re-sampling at each time step. After the final re-sampling, all the particles obtained by the CAPFA fall into the feasible area, leading to $\mathrm{PFP} = 100\%$. The use of the road constraints in the CAPFA substantially increased the ESS. For $N = 250$, the gain in ESS by using the CAPFA compared to the APF was $G_{\mathrm{eff}} = 100\% \times (75.3 - 42.7)/42.7 = 76.4\%$. For the estimation error, the average MSE of the CAPFA was only about $100\% \times 5.022/7.051 = 71.2\%$ of the APF.

D. Comparison With Other Algorithms

In this subsection we compare the proposed CAPFA to some existing particle filters with state constraints.

It was shown in [7] that the constrained particle filter (CPF) in [7] outperformed the acceptance-rejection method in [6] and several conventional filtering methods including the EKF, constrained EKF, UKF, constrained UKE and MHE. Hence, we first consider CPF Algorithm 2 in [7] (denoted as CPF(alg.2 [7])) for which $\alpha = 0.05$ was used in the chi-square test. The results are displayed in the third panel of Table III.

It can be seen that the CPF Algorithm 2 in [7] had a reasonable computation time. This is mainly because the chi-square test is used to reduce the times that the optimization routine is called: when the test is passed, no optimization is undertaken even if there are still infeasible particles. In our simulation study, we noted that this chi-square test was effective for the reduction of computation time at the price that a substantial proportion of the particles did not satisfied the constraints: its PFP was about 59%, hence there were about 41% of the particles lying outside the feasible area. This further caused another important issue: the corresponding PESS of the CPF Algorithm 2 was very low. It was only about 1/3 of the nominal size N. Low PFP and PESS levels affect the quality of inference. This can be seen from Table III that when the total number of particles N was not large (150 and 250), both the MSEs and SDs were relatively high. Note that a high level of SD indicates that the quality of the state filtering was not stable. This problem was alleviated when N (and hence the corresponding effective sample size) became large. This suggests that particle number N need to be set large for the CPF to cope with the lower percentages of ESS and feasible particles. We also tested Algorithms 3 and 4 of the CPF in [7]; they had a similar performances to the CPF Algorithm 2.

We next turn to consider the Gibbs-sampler-based particle filter in [8], denoted as CPF(Gibbs [8]) in Table III. First, we note that this method can guarantee all particles fall into the feasible area and hence its PFP is 100%. It can also be seen from Table III that the PESS and accuracy as measured by MSE and SD were better than that of the CPF Algorithms in [7], but not as good as the CAPFA. More importantly, its computational cost was much higher than all the other filters. The Gibbs-sampler-based particle filter involves: (a) the iterative approach of the Gibbs-sampler for every particle; and (b) determining the constraint interval within each iteration of the Gibbs sampling. To speed up the computation process, the constraint interval was solved analytically in our simulation study and hence the computation in (b) was kept at the minimum level. If it was impossible to analytically solve problem (b), then searching a numerical solution for (b) would incur a much higher computational cost. Overall, it seems that the Gibbs-sampler-based particle filter in [8] is more suitable to offline applications due to its computational demand.

Finally, we considered two particle filters recently proposed in [18] and [19], denoted as tUPF([18]) and tUPF ([19]) respectively in Table III. For both methods, the importance distributions are based on the UKF approach that is further supplemented with a truncation technique to accommodate the state constraints. They have 100% PFP because all the particles outside the feasible are redrawn. We can see from Table III that, in comparison with tUPF([18]), tUPF([19]) is more accurate but it has a higher computational cost. The CAPFA outperforms both of them in terms of MSE, SD, and PESS. The CAPFA has a comparable computation time to tUPF([18]), and is much faster than tUPF([19]). Note that for tUPF([19]), the cases for $N = 150$ and 250 are not reported here because it did not work properly due to a very low ESS.

E. Impact of a Poor Initial Condition

It is well known that a poorly chosen initial point for a filter can have a negative impact on the performance of the filter, in

TABLE IV
COMPARISON OF THE PERFORMANCES OF SEVERAL PARTICLE FILTERS WITH A POOR INITIAL GUESS AVERAGED OVER 100 RUNS

Filters	N	MSE	PESS(%)	PFP(%)	CT(s)
APF([13])	500	7.197	42.6	44.4	0.021
CAPFA	250	5.857	75.6	88.7	0.074
CAPFA	500	5.153	73.1	88.6	0.148
CPF(alg.2 [7])	500	6.460	33.8	57.6	0.477
CPF(alg.3 [7])	500	6.576	36.2	57.3	0.198
CPF(alg.4 [7])	500	6.482	37.3	56.7	0.044
CPF(Gibbs [8])	500	6.244	49.1	100	4.726
tUPF([18])	500	8.574	55.1	100	0.046
tUPF([19])	500	6.357	6.9	100	0.507

particular during an initial time period. Next, we consider a poor initial guess chosen as a Gaussian distribution with mean $\hat{x}_0 = [90, 15, 0, 10]^T$ and covariance $P_0 = \text{diag}\{100, 100, 1, 1\}$. Due to the poor initial guess, most of the algorithms in Table III require large particle numbers ($N = 500$ or above) to obtain a reasonable result in the test. However, the proposed CAPFA performed well even with $N = 250$ which is therefore included here for the comparison purposes. The results are displayed in IV.

Table IV shows that with a poorer initial guess: (a) the MSE became larger for all the filters; (b) the CAPFA with both $N = 250$ and 500 outperformed the others; (c) the CAPFA was fairly robust against the poor initial condition.

In summary, the simulation study shows that the PESS level of the proposed CAPFA is superior over the existing algorithms, leading to a satisfactory filtering accuracy with a reasonably low computational cost.

V. CONCLUDING REMARKS

In this technical note we have proposed a new particle filtering algorithm, CAPFA, to deal with state estimation with state constraints. We have shown that the CAPFA has three distinguished features for dealing with the state constraints: (a) a highly selective method that chooses particles for propagating into the next time step; (b) an effective optimization method for locating the center of each restricted transition distribution; and (c) a method that efficiently constructs the importance distributions using knowledge of the state constraints.

The simulation study shows that the proposed SIPM method took only a fraction of time used by the 'standard' optimization method with little price of performance degradation. It also demonstrates the benefits of incorporating state constraints in the particle sampling: the CAPFA had a much larger ESS in comparison with the existing particle filters.

This technical note assumes that the probability density function of the system noise and the constraint functions have continuous second-order partial derivatives. Primarily these conditions are used in the SIPM. Hence, if necessary, these assumptions can be relaxed by replacing the SIPM with a suitable derivative-free optimization method. We also note that the assumptions of the proposed method exclude some realistic models of practical interests where (a) the uncertainty in the state equation is modeled as a non-additive noise without a density; and (b) the state constraint is defined implicitly by a feasible set, where it is just possible to check whether a point is inside or outside the feasible set, but no simple explicit expression is available in terms of regular enough constraint functions. Further research is required to overcome these limitations.

ACKNOWLEDGMENT

The authors thank the reviewers for their helpful comments.

REFERENCES

[1] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

[2] I. Kyriakides, D. Morrell, and A. Papandreou-Suppappola, "Sequential Monte Carlo methods for tracking multiple targets with deterministic and stochastic constraints," *IEEE Trans. Signal Processing*, vol. 56, no. 3, pp. 937–948, 2008.

[3] O. Straka, J. Duník, and M. Šimandl, "Truncation nonlinear filters for state estimation with nonlinear inequality constraints," *Automatica*, vol. 48, no. 2, pp. 273–286, 2012.

[4] V. Zavala, C. Larid, and L. Biegler, "A fast moving horizon estimation algorithm based on nonlinear programming sensitivity," *J. Process Control*, vol. 18, no. 9, pp. 876–884, 2008.

[5] D. Simon, "Kalman filtering with state constraints: A survey of linear and nonlinear," *IET Control Theory Appl.*, vol. 4, no. 8, pp. 1303–1318, 2010.

[6] L. Lang, W.-S. Chen, B. R. Bakshi, P. K. Goel, and S. Ungarala, "Bayesian estimation via sequential Monte Carlo sampling constrained dynamic systems," *Automatica*, vol. 43, no. 9, pp. 1615–1622, 2007.

[7] X. Shao, B. Huang, and J. M. Lee, "Constrained Bayesian state estimation: A comparative study and a new particle filter based approach," *J. Process Control*, vol. 20, no. 2, pp. 143–157, 2010.

[8] Z. Zhao, B. Huang, and F. Liu, "Constrained particle filtering methods for state estimation of nonlinear process," *AIChE J.*, vol. 60, no. 6, pp. 2072–2082, 2014.

[9] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[10] F. Papi, M. Bocquel, M. Podt, and Y. Boers, "Fixed-lag smoothing for Bayes optimal knowledge exploitation in target tracking," *IEEE Trans. Signal Processing*, vol. 62, no. 12, pp. 3143–3152, 2014.

[11] A. Kong, J. S. Liu, and W. H. Wong, "Sequential imputations and Bayesian missing data problems," *J. Amer. Stat. Assoc.*, vol. 89, no. 425, pp. 278–288, 1994.

[12] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovsky, Eds. Oxford: Oxford University Press, 2011.

[13] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Amer. Stat. Assoc.*, vol. 94, no. 446, pp. 590–599, 1999.

[14] M. K. Pitt and N. Shephard, "Auxiliary variable based particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. Freitas, and N. Gordon, Eds. New York: Springer, 2001.

[15] A. M. Johansen and A. Doucet, "A note on auxiliary particle filters," *Stat. Probab. Lett.*, vol. 78, no. 12, pp. 1498–1504, 2008.

[16] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge University Press, 2004.

[17] M. Yu, C. Liu, B. Li, and W.-H. Chen, "An enhanced particle filtering method for GMTI radar tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 3, pp. 1408–1420, 2016.

[18] O. Straka, J. Duník, and M. Šimandl, "Truncated unscented particle filter," in *Proc. American Control Conf.*, San Francisco, CA, 2011, pp. 1825–1830.

[19] M. Yu, W.-H. Chen, and J. Chambers, "Truncated unscented particle filter for dealing with non-linear inequality constraints," in *Proc. Sensor Signal Processing for Defence (SSPD'14)*, 2014, pp. 1–5.