Particle Filter with LMMSE Importance Sampling

Bryan Pogorelsky
Dept. of Aerospace Engineering & Engineering Mechanics
The University of Texas at Austin
Austin, TX, USA
pogorelsky@utexas.edu

Kristen Michaelson
Dept. of Aerospace Engineering & Engineering Mechanics
The University of Texas at Austin
Austin, TX, USA
kmichaelson@utexas.edu

Renato Zanetti
Dept. of Aerospace Engineering & Engineering Mechanics
The University of Texas at Austin
Austin, TX, USA
renato@utexas.edu

***Abstract*—Nonlinear estimation can be performed in many ways, with the particle filter being one of the most common. It is well known that ignoring the latest measurement value in the choice of importance density can result in poor particle filter performance for certain classes of challenging problems. In this paper a novel particle filter with importance density based on the linear minimum mean square error (LMMSE) estimator is presented. Performance is evaluated using Monte Carlo simulations of a highly nonlinear growth model. The proposed algorithm is compared with existing particle filter formulations using metrics for accuracy, consistency, and particle diversity.**

***Index Terms*—Nonlinear Estimation, Particle Filters, Sequential Monte Carlo Methods**

I. Introduction

Estimation of complex dynamical systems has long been a topic of interest in the science and engineering community. These systems are often nonlinear and non-Gaussian in nature, making the task of estimation difficult. For linear systems with additive Gaussian noise, the Kalman filter provides the optimal solution [1]. In this case, the Kalman filter is the linear minimum mean square error (LMMSE) estimator: the optimal linear (affine) filter that minimizes a mean squared error (MSE) performance index. For systems that are nonlinear or non-Gaussian, the LMMSE solution is not typically available in closed form. Thus, approximations are necessary. Alternative formulations of the Kalman filter, such as the extended Kalman filter (EKF), can be used for nonlinear systems but do not provide an optimal solution [2]. If a system is highly nonlinear, the EKF can diverge when the local linearization is a poor representation of the true nonlinear function.

Other more robust approaches exist that employ the Kalman filter framework to tackle the estimation of nonlinear systems; these include the unscented Kalman filter (UKF) [3], quadrature Kalman filter [4], and the cubature Kalman filter [5]. While the EKF relies on linearizations of the dynamics and measurement models, these filters employ Gaussian statistical approximation of the nonlinear functions using a set of deterministic regression points. They often produce good results, but situations arise when the system is highly nonlinear and the resulting probability density functions (PDFs) are highly non-Gaussian.

Relaxing the requirement of a linear estimator results in the optimal estimate, also in a MSE sense, being the conditional mean. Calculation of the conditional mean in the presence of nonlinearities is also not typically possible in closed form, and approximations need to be made. One such approximation is given by Monte Carlo methods in which PDFs are approximated as a discrete set of random samples. Statistics (e.g., mean and covariance) can be calculated from these samples. The main advantage of these methods is that they are not subject to any linearity or Gaussianity assumptions [6]. If one were able to sample directly from a posterior distribution, then calculating the mean or covariance of that conditional distribution at any time would be trivial. In practice, samples are drawn from an importance distribution. Sequential importance sampling (SIS) is a recursive process in which samples are drawn from an importance distribution and weights are appropriately chosen to accompany each of the samples. The main drawback of SIS is that sample degeneracy eventually occurs, meaning that very few particles (possibly one) have significant importance weight.

To combat this, sequential importance sampling with resampling (SISR) employs a resampling step to diversify the samples at a particular time step [7]. SISR algorithms are also known as particle filters. Many particle filters rely on process noise to diversify the samples, and hence the performance of SISR algorithms are typically dependent on (1) the presence of "enough" process noise, and (2) the choice of a good importance distribution. One widely used particle filter is the bootstrap particle filter (BPF) [8]. The BPF conveniently uses the transition density as the importance density when particle weights are updated. A consequence of this choice is that the most recent measurement is not accounted for in the importance distribution. Hence, the BPF's performance is adequate when there is "enough" measurement noise. Without

This work was sponsored in part by NASA under award number 80NSSC20M0211.

enough measurement noise, the BPF suffers from particle impoverishment.

The auxiliary particle filter uses an intermediate auxiliary variable in the selection of the importance distribution to account for the most recent measurement [9]. If particle diversity is low after resampling, it may be increased by sampling small perturbations from a continuous distribution. This is called a regularized particle filter [10]. The regularization step can be implemented regardless of the importance density chosen as it is performed after particle weights are updated.

Another method to account for the most recent measurement in the importance distribution is to treat each particle as the mean of a Gaussian distribution that is propagated and updated with Kalman filter equations. More specifically, this can be accomplished by utilizing the EKF equations as was done in [11] to arrive at the Extended Kalman Particle Filter (EKPF). Instead of taking the propagated particles as the importance distribution, as is done in the BPF, a Kalman update is performed on each particle where its posterior distribution is assumed to be Gaussian with mean and covariance given by the EKF formulation. After this step, importance samples are drawn from each of the Gaussian posteriors to calculate the new values of the weights.

The authors of [11] noted that the EKPF did not have adequate performance; thus, the Unscented Particle Filter (UPF) was proposed [12]. In this formulation, each particle is propagated using the UKF equations rather than the EKF equations. The filter performance was observed to be significantly improved over the EKPF. In both the EKPF and UPF formulations, the choice of initial covariance of the particles was left unspecified. In the associated open source code for [11], the initial covariance of each particle is set to the initial total covariance of the states. This is artificially large and partially explains why the UPF outperforms the EKPF.

In this work we also propose using a LMMSE algorithm to generate a Gaussian posterior used as the importance distribution. This new approach mitigates the need for "enough" measurement noise. We will focus on the common scenario where measurement and process noises are additive and Gaussian and we will show that the derived algorithm is different from the EKPF. The main contributions of this work are:

- The proposed algorithm does not carry covariance matrices associated with each particle.
- The proposed importance sampling method is more computationally efficient than those described in [13].

Ref. [14] describes an algorithm similar to ours without including implementation details.

The remainder of this paper is structured as follows: Section II presents a mathematical description of the proposed filter, and Section III applies the filter to two nonlinear and/or non-Gaussian systems. In each example, the proposed filter is compared to the BPF and the UPF using accuracy and consistency criteria. Finally, concluding remarks are presented on the results of the experiments.

II. Particle Filter with LMMSE Importance Sampling

Throughout this paper we consider discrete-time nonlinear dynamics and measurement model. The dynamics are given by

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}) + \boldsymbol{\nu}_k \tag{1}$$

where $\mathbf{f}_k$ is some nonlinear function and $\boldsymbol{\nu}_k$ is the process noise with PDF $p_{\boldsymbol{\nu}_k}(\boldsymbol{\nu}_k)$ and covariance matrix $\mathbf{Q}_k$. The measurements are given by

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \boldsymbol{\eta}_k \tag{2}$$

where the nonlinear measurement model is $\mathbf{h}_k(\mathbf{x}_k)$ and $\boldsymbol{\eta}_k$ is the measurement noise with PDF $p_{\boldsymbol{\eta}_k}(\boldsymbol{\eta}_k)$ and covariance matrix $\mathbf{R}_k$. It is assumed that process and measurement noise are white sequences, uncorrelated from each other, and uncorrelated from $\mathbf{x}_0$.

Initializing the filter consists of first drawing N i.i.d samples, or particles, from the known initial probability density $p_{\mathbf{x}_0}(\mathbf{x}_0)$, with particle weights initialized as

$$w^i = \frac{1}{N}. \tag{3}$$

Each particle is propagated forward in time as done in the bootstrap particle filter. For each propagated particle the LMMSE equations below are now applied,

$$\hat{\mathbf{x}} = \mathrm{E}\{\mathbf{x}\} + \boldsymbol{\Sigma}_{\mathbf{xy}}\boldsymbol{\Sigma}_{\mathbf{xy}}^{-1}(\mathbf{y} - \mathrm{E}\{\mathbf{y}\}) \tag{4}$$

$$\mathbf{P} = \boldsymbol{\Sigma}_{\mathbf{xx}} - \boldsymbol{\Sigma}_{\mathbf{xy}}\boldsymbol{\Sigma}_{\mathbf{yy}}^{-1}\boldsymbol{\Sigma}_{\mathbf{yx}} \tag{5}$$

where $\boldsymbol{\Sigma}_{\mathbf{xx}}$, $\boldsymbol{\Sigma}_{\mathbf{xy}}$, $\boldsymbol{\Sigma}_{\mathbf{xy}}$ are the covariance of the prior density, cross-covariance of prior and measurements, and the covariance of the measurements respectively. Many choices of approximations exist to calculate $\mathrm{E}\{\mathbf{y}\}$, $\boldsymbol{\Sigma}_{\mathbf{xy}}$, and $\boldsymbol{\Sigma}_{\mathbf{xy}}$ (linearization, unscented transformation, Gaussian quadrature, etc.). Assuming the process noise covariance $\mathbf{Q}_k$ is small enough such as linearization of $\mathbf{h}_k$ is valid for any likely outcome of $\mathbf{x}_k^i$, the EKF formulation can be used as specified below for each particle:

$$\mathbf{W}_k^i = \mathbf{H}_k^i\mathbf{Q}_k(\mathbf{H}_k^i)^{\mathrm{T}} + \mathbf{R}_k \tag{6}$$

$$\mathbf{K}_k^i = \mathbf{Q}_k(\mathbf{H}_k^i)^{\mathrm{T}}(\mathbf{W}_k^i)^{-1} \tag{7}$$

$$\bar{\mathbf{x}}_k^i = \mathbf{f}_k(\mathbf{x}_{k-1}^i) + \boldsymbol{\nu}_k^i \tag{8}$$

$$\hat{\mathbf{x}}_k^i = \bar{\mathbf{x}}_k^i + \mathbf{K}_k^i\Big(\mathbf{y}_k - \mathbf{h}_k(\bar{\mathbf{x}}_k^i)\Big) \tag{9}$$

$$\mathbf{P}_k^i = \Big(\mathbf{I} - \mathbf{K}_k^i\mathbf{H}_k^i\Big)\mathbf{Q}_k\Big(\mathbf{I} - \mathbf{K}_k^i\mathbf{H}_k^i\Big)^{\mathrm{T}} + \mathbf{K}_k^i\mathbf{R}_k\mathbf{K}_k^{i\mathrm{T}} \tag{10}$$

where

$$\mathbf{H}_k^i = \frac{\partial\mathbf{h}_k(\mathbf{x})}{\partial\mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_k^i}. \tag{11}$$

Importance samples $\mathbf{x}_k^i$ are then drawn from the importance distribution $\pi\left(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k\right)$ defined as a Gaussian distribution parameterized by the mean and covariance above

$$\pi\left(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{y}_k\right) = \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_k^i, \mathbf{P}_k^i\right). \tag{12}$$

The importance weights are calculated as:

$$w_k^i \propto \frac{p\left(\mathbf{y}_k|\mathbf{x}_k^i\right) p\left(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i\right)}{\pi\left(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{y}_k\right)} \tag{13}$$

where

$$p\left(\mathbf{y}_k|\mathbf{x}_k^i\right) = p_{\boldsymbol{\eta}_k}\left(\mathbf{y}_k - \mathbf{h}_k(\mathbf{x}_k^i)\right) \tag{14}$$

and

$$p\left(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i\right) = p_{\boldsymbol{\nu}_k}\left(\mathbf{x}_k^i - \mathbf{f}_k(\mathbf{x}_{k-1}^i)\right). \tag{15}$$

The unnormalized importance weights w_k^i are then normalized to calculate the state estimate and associated covariance as follows:

$$\hat{\mathbf{x}}_k = \Sigma_{i=1}^N \tilde{w}_k^i \mathbf{x}_k^i \tag{16}$$

$$\mathbf{P}_k = \Sigma_{i=1}^N \tilde{w}_k^i \left[\mathbf{x}_k^i - \hat{\mathbf{x}}_k\right]\left[\mathbf{x}_k^i - \hat{\mathbf{x}}_k\right]^{\mathrm{T}} \tag{17}$$

where $\tilde{w}_k^i$ are the normalized weights. A resampling scheme of the user's choice is then implemented and all the particle weights are reset to $1/N$ before commencing the next time time step of the filter. An important note is that unlike the EKPF and UPF, the particles' covariance is not kept for the next filter time step and is effectively reset to $\mathbf{0}$, with the process noise covariance, $\mathbf{Q}_k$, impacting the *a priori* uncertainty of each particle. Additionally, while this approach shares some commonality with [13], the proposed algorithm is different from the 3 algorithms introduced in [13]. Only the third algorithm of [13] used importance sampling but unlike the algorithm proposed here it did not condition the distribution on the value of the state at the previous time ($\hat{\mathbf{x}}_{k-1}^i$) resulting in a different and much more computationally intensive algorithm.

III. Numerical Simulation

To evaluate the the performance of the proposed algorithm, a numerical simulation is conducted. A Monte Carlo analysis is used to gather sample performance statistics of the proposed filter. The two main metrics that are used to determine the filter's performance are accuracy and consistency. Accuracy is the measure of how close the filter state estimate is to the true state. Consistency on the other hand, is the measure of how close the filter's predicted covariance is to the true error covariance which can be found from Monte Carlo simulation.

To measure the accuracy, the Monte Carlo averaged root mean square error (RMSE) is used,

$$\text{RMSE}(k) = \sqrt{\frac{1}{N_m}\sum_{j=1}^{N_m}\left\|\mathbf{x}_k^j - \hat{\mathbf{x}}_k^j\right\|_2^2} \tag{18}$$

where N_m is the number of Monte Carlo simulations, $\mathbf{x}_k^j$ are the true states and $\hat{\mathbf{x}}_k^j$ the estimated states, both referenced to the jth Monte Carlo run and kth time step.

For the evaluation of consistency, the noncredibility index (NCI) is used [13] [15]. The index is defined as,

$$\begin{aligned}\text{NCI}_k = \frac{1}{N_m}\sum_{j=1}^{N_m}\Big[10\log_{10}((\mathbf{x}_k^j - \hat{\mathbf{x}}_k^j)^T(\mathbf{P}_k^j)^{-1}(\mathbf{x}_k^j - \hat{\mathbf{x}}_k^j)) \\ -10\log_{10}((\mathbf{x}_k^j - \hat{\mathbf{x}}_k^j)^T(\mathbf{\Sigma}_k)^{-1}(\mathbf{x}_k^j - \hat{\mathbf{x}}_k^j))\Big]\end{aligned} \tag{19}$$

where $\mathbf{P}_k^j$ is the filter's error covariance at the jth Monte Carlo run and kth time step. $\mathbf{\Sigma}_k$ is the ideal error covariance calculated from the samples of the Monte Carlo simulations. When the filter is consistent (the difference between the true and predicted covariance is small) the NCI will tend towards zero.

Additionally, the number of effective particles during each time step is also evaluated using the below formulation.

$$N_{\text{eff}} = \frac{1}{\sum_{j=1}^{N_m}\left(\tilde{w}_k^i\right)^2} \tag{20}$$

If the number of effective particles is low, this is an indicator the particle impoverishment is occurring where a very small number of particles have much of the weight.

A. Univariate Nonstationary Growth Model

The following highly nonlinear univariate dynamic system and measurement model have often been used to evaluate nonlinear filters' performance [13] [8] [16]

$$x_k = \frac{x_{k-1}}{2} + 25\frac{x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2(k-1)) + \nu_{k-1} \tag{21}$$

$$y_k = \frac{x_k^2}{20} + \eta_k \tag{22}$$

where ν_{k-1} and η_k are the process and measurement noises, respectively. These independent noises are distributed as follows,

$$\nu_{k-1} \sim \mathcal{N}(0,1) \tag{23}$$

$$\eta_k \sim \mathcal{N}(0,1) \tag{24}$$

and the initial state of the system is assumed to be a Gaussian distribution with $\mu_0 = 0$ and $P_0 = 1$.

For this example, 200 Monte Carlo simulations were performed for a time span of $k = [0, 50]$ and the bootstrap particle filter, the unscented particle filter, and the LMMSE particle filter were tested. Each filter uses 100 particles. An important note is that for all filters, resampling is conducted after each time step using the same systematic resampling algorithm [17]. Figure 1 shows a time history of the RMSE of the 200 simulations. The LMMSE PF outperforms the BPF throughout a large majority of the time span. Figure 2 shows the consistency of the filters. For nearly the entire span of the simulation, the LMMSE PF's NCI value is the closest to zero, which is indicative that the predicted filter covariance is closer to the actual observed Monte Carlo covariance than the BPF and the UPF. Finally, Figure 3 shows that for all time, the LMMSE PF has a greater number of effective particles than the BPF and the UPF, meaning particle diversity is higher.

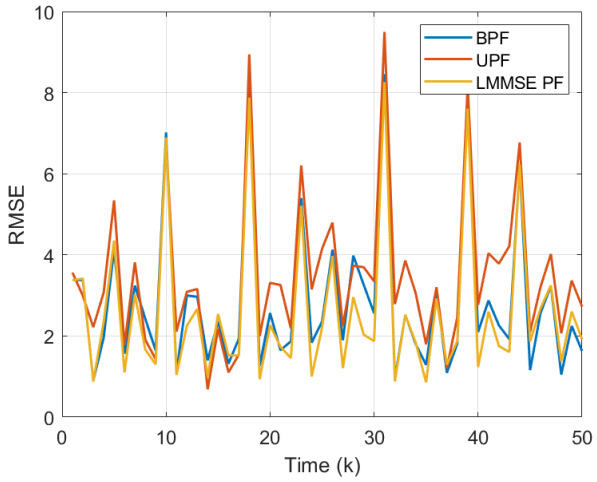

Fig. 1. Time history of Monte Carlo averaged RMSE for 200 random realizations

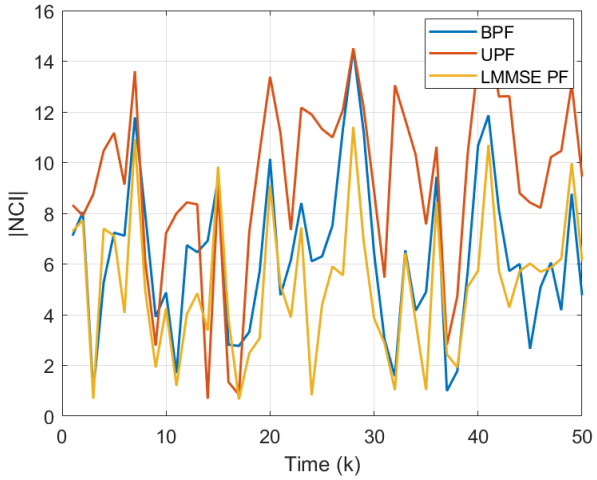

Fig. 2. Time history of Monte Carlo averaged absolute NCI for 200 random realizations

In Table I a summary of the results for this Monte Carlo simulation can be seen. These are calculated by time averaging the data displayed on the time history figures and show the same trends.

TABLE I
UNIVARIATE NONSTATIONARY GROWTH MODEL RESULTS

	RMSE	NCI	N_{eff}
LMMSE PF	2.616	5.184	77.707
Unscented PF	3.458	9.309	55.705
Bootstrap PF	2.814	6.300	59.848

The benefits of the new algorithm when compared with the BPF are evident when the value of the measurement noise variance is reduced to $R = 0.1$. With this value of the measurement noise, the BPF's likelihood function occasionally underflows (i.e. after a measurement update, all the new weights are zero) with a 64 bit float representation. The LMMSE PF rarely underflows, since it is able to maintain a higher particle diversity. In each algorithm, when the weights underflow, they are all reset to $1/N$; effectively causing the filter to ignore the measurement.

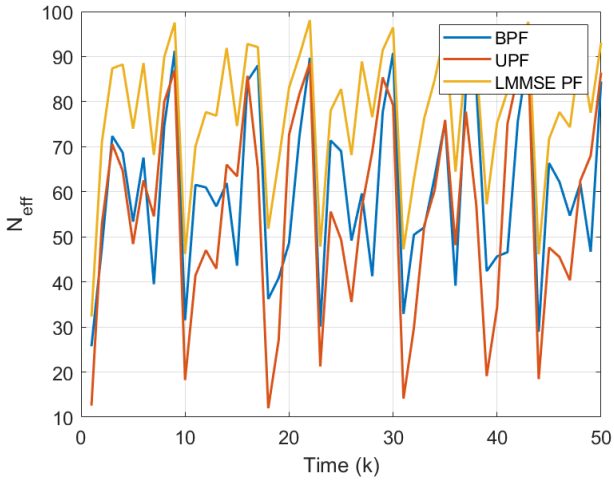

Fig. 3. Time history of Monte Carlo averaged absolute number of effective particles for 200 random realizations

B. Univariate Dynamics with Non-Gaussian Process Noise and Non-stationary Observation Model

The second numerical example is that used in Ref. [11] to show the performance advantages of the UPF over the BPF. The dynamics are

$$x_k = 1 + \sin(\omega\pi(k-1)) + \phi_1 x_{k-1} + v_{k-1} \tag{25}$$

where $\omega = 4e-2$, $\phi_1 = 0.5$, and v_{k-1} is a process noise drawn from a gamma distribution:

$$v_{k-1} \sim \Gamma(3,2) \tag{26}$$

The non-stationary observation model is

$$y_k = \begin{cases} \phi_2 x_k^2 + n_k & t \leq 30 \\ \phi_3 x_k - 2 + n_k & t > 30 \end{cases} \tag{27}$$

where $\phi_2 = 0.2$, $\phi_3 = 0.5$, and n_k is an observation noise drawn from a Gaussian distribution:

$$n_k \sim \mathcal{N}(0, .00001) \tag{28}$$

In this observation model, the measurement equation changes after the first 30 time steps.

For this model, 100 Monte Carlo simulations were performed for the UPF, the BPF, and the LMMSE PF for a time span of $k = [0, 60]$. All three filters use 200 particles. Figure 4 shows an example trajectory with each filter's tracking performance.

The initial state x_0 is chosen randomly from a Gaussian distribution $x_0 \sim \mathcal{N}(1, 0.75)$. All the filters are initialized with $\hat{x}_0 = 1$. After the first time step, the UPF and the LMMSE

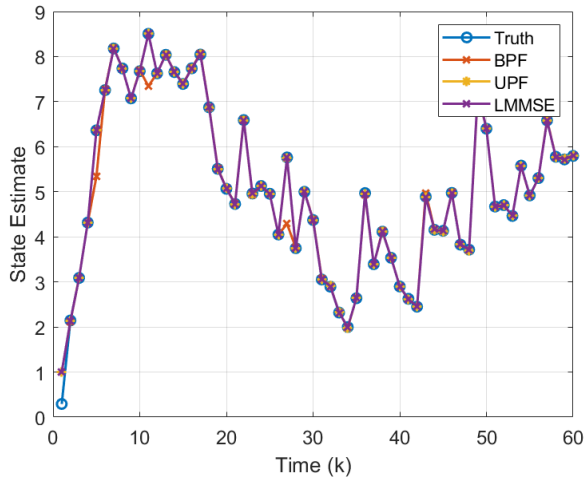

Fig. 4. Sample trajectory with filter performance

PF achieve near-perfect tracking performance. The BPF falters when the state changes too quickly but recovers at the next time step. Figure 5 shows the RMSE performance of each filter over 100 Monte Carlo runs.

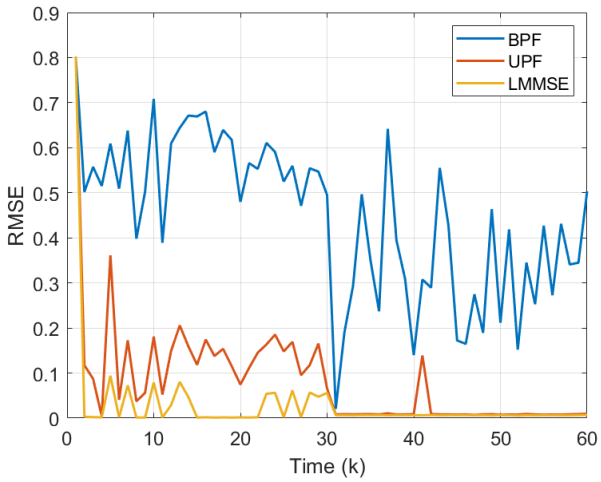

Fig. 5. Time history of Monte Carlo averaged RMSE for 100 realizations of the system (25)-(28)

It is clear from Figure 5 that on average, the LMMSE PF outperforms the UPF in the first half of the simulation. Later, their performance is similar. The BPF performs the worst throughout, although its performance also improves slightly on average in the latter half of the simulation.

Figure 6 compares absolute NCI values for each filter. Once again, the LMMSE PF more closely matches the true covariance of the system than the BPF. The absolute NCI values of the UPF are visibly worse than those of the LMMSE PF in the first 30 time steps. Their NCI values are comparable in the second half of the simulation. Table II summarizes the results in this example.

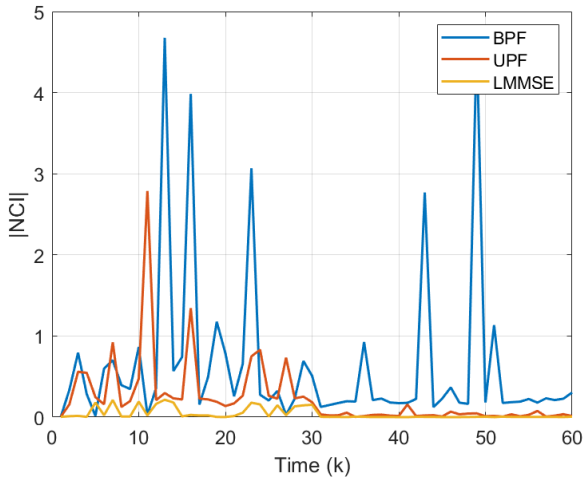

Fig. 6. Time history of Monte Carlo averaged absolute NCI for 100 realizations of the system (25)-(28)

TABLE II
UNIVARIATE DYNAMICS WITH NON-STATIONARY OBSERVATION MODEL RESULTS

	RMSE	NCI
LMMSE PF	0.085	0.039
Unscented PF	0.118	0.234
Bootstrap PF	0.428	0.636

Table II emphasizes that while the LMMSE PF slightly outperforms the UPF in terms of estimation error, it shows remarkable improvement over the UPF in terms of absolute NCI.

IV. CONCLUSIONS

The importance of accounting for the measurement outcome in the importance distribution of a particle filter (PF) is well known and implemented in several algorithms such as the Auxiliary PF (APF), Extended Kalman PF (EKPF), and Unscented PF (UPF). In this paper, a new linear minimum mean square error particle filter is proposed to account for the measurement value in the importance distribution. An EKF approximation of the LMMSE equations is used for each particle where the prior iteration's particle covariances are not kept.

A Monte Carlo simulation was performed to test the efficacy of the proposed algorithm for a highly nonlinear growth model. The LMMSE particle filter outperformed the bootstrap particle filter in terms of accuracy and more significantly for consistency and particle diversity metrics. In the second example, a Monte Carlo simulation was performed to compare the proposed algorithm with the UPF and the BPF for a system with non-Gaussian process noise and a non-stationary observation model. The proposed algorithm demonstrated higher accuracy and better covariance approximation than both the UPF and the BPF.

REFERENCES

[1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[3] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. Ieee, 2000, pp. 153–158.

[4] I. Arasaratnam, S. Haykin, and R. J. Elliott, "Discrete-time nonlinear filtering algorithms using gauss–hermite quadrature," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 953–977, 2007.

[5] I. Arasaratnam and S. Haykin, "Cubature kalman filters," *IEEE Transactions on automatic control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[6] A. Doucet, N. De Freitas, and N. Gordon, "An introduction to sequential monte carlo methods," in *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 3–14.

[7] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[8] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEE proceedings F (radar and signal processing)*, vol. 140, no. 2. IET, 1993, pp. 107–113.

[9] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American statistical association*, vol. 94, no. 446, pp. 590–599, 1999.

[10] N. Oudjane and C. Musso, "Progressive correction for regularized particle filters," in *Proceedings of the Third International Conference on Information Fusion*, vol. 2. IEEE, 2000, pp. THB2–10.

[11] R. Van, A. Doucet, N. de Freitas, E. Wan *et al.*, "The unscented particle filter technical report cued/f-infeng/tr 380," *Cambridge University Engineering Department*, 2000.

[12] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. A. Wan, "The unscented particle filter," in *Advances in neural information processing systems*, 2001, pp. 584–590.

[13] S. Yun and R. Zanetti, "Sequential monte carlo filtering with gaussian mixture sampling," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 9, pp. 2069–2077, 2019.

[14] M. Murata and K. Hiramatsu, "Non-gaussian filters for nonlinear continuous- discrete models," *SICE Journal of Control, Measurement, and System Integration*, pp. 53–61.

[15] X. R. Li and Z. Zhao, "Measuring estimator's credibility: Noncredibility index," in *2006 9th International Conference on Information Fusion*. IEEE, 2006, pp. 1–8.

[16] D. Raihan and S. Chakravorty, "Particle gaussian mixture (pgm) filters," in *2016 19th International Conference on Information Fusion (FUSION)*. IEEE, 2016, pp. 1369–1376.

[17] R. Douc and O. Cappé, "Comparison of resampling schemes for particle filtering," in *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.* IEEE, 2005, pp. 64–69.