52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

# Improved Auxiliary and Unscented Particle Filter Variants

Alexandros C. Charalampidis and George P. Papavassilopoulos

***Abstract*—This paper proposes some modifications to the Auxiliary Particle Filter and the Unscented Particle Filter. For the APF, based on some error bound considerations it is suggested that the auxiliary weights are taken into account not proportionally but nonlinearly. An improved way to compensate for the auxiliary weights after the resampling is also proposed. For the UPF, a method is introduced to compute the covariance matrices of the particles not using the UKF equations for each particle separately, but so as to optimize the characteristics of the total distribution. The application of the modified filters to an example shows that the proposed changes lead to performance increase.**

***Index Terms*—Particle filtering, Nonlinear filters, State estimation, Resampling, Numerical methods, Auxiliary Particle Filter, Unscented Particle Filter.**

## I. Introduction

State estimation is a task important for a vast variety of applications, ranging from biomedical to space technology and from autonomous robots to chemical processes. In practice, measurements are noisy and real systems are affected by disturbance (process noise). Such noise are in many cases modelled as stochastic processes.

The sensors in many cases provide measurements periodically, while the calculations involved are nowadays done using digital electronics, so it is reasonable that most literature has focused on the discrete-time case, using a discretized model of the actual system if it is a continuous-time system.

The Kalman Filter (KF) [1]–[4] provides an exact solution to the recursive state estimation problem for linear systems with Gaussian noise and initial condition[1]. In that case the state distribution is also Gaussian and it can be fully described by its mean and covariance matrix. It is very important that the solution is recursive, because otherwise each time a new measurement would be obtained, all data should be analysed again leading to huge computational cost. The problem can be also solved exactly if the state space is finite.

As explained in the next section, Bayes Rule provides a recursive solution for nonlinear systems, but in the form of some integrals which can be numerically evaluated only for low-dimensional systems. Some techniques, including the Extended Kalman Filter (EKF, [6]) and the Unscented Kalman Filter (UKF, [7], [8]), use a Gaussian to approximate the state distribution. Although these techniques have been widely studied and used in applications [9]–[15], there exist cases, for example when the state probability density function (pdf) is multimodal, in which this approximation cannot be made and more computationally intensive techniques [16]–[20] have to be used.

The most prominent of these techniques is Particle Filtering [17], [19], [20]. In this technique, the state distribution is approximated using a weighted sum of Dirac functions (called particles). It is a stochastic simulation technique, in the sense that the location of each particle is propagated according to the system equation and setting the disturbance value equal to a random sample of the process noise distribution. The weights are then adapted according to the likelihood of the measurement.

The crucial step, first introduced in [17], is that of resampling, namely after the update new particles with equal weights are randomly drawn from the estimated a posteriori distribution. This is necessary to avoid the phenomenon of "weight degeneracy", in which after a number of steps all particles, except of one, have negligible weight. Using resampling, more particles are likely to be generated in the regions of the state space where the probability density is higher, as desired, while after resampling the particles have equal weights since they have been selected randomly from the same distribution.

Obviously the approximation is more accurate for a large number of particles, but then the algorithm is computationally intensive, so there is a need of improved versions of the algorithm, exploiting better the computational power. Thus, the algorithm has to provide a good approximation without too many particles, while the computations for the renewal of the particles have to be not too complicated. Indeed, several variants have appeared in the bibliography [19], [20]. This paper provides improved versions of two commonly used variants.

More specifically, in Section III some modifications to the Auxiliary Particle Filter (APF, [19]–[21]) are proposed. The APF has been designed to tackle the fact that from some particles created by the prediction step, possibly many will get a very small weight from the correction step (because the likelihood of the new measurement will be very small). To that end, an estimation of these weights is made and more particles are created from those particles for which it is estimated that they will lead to greater weights. This is something taken into account by the filter in the calculation of the weights after resampling. The proposed modifications concern firstly how much more the particles with greater estimated weight should be preferred, and secondly how should it be compensated the fact that some of the particles

[1]The linear Gaussian case is one of the few that can be also solved in the continuous time [5].

978-1-4673-5716-6/13/$31.00 ©2013 IEEE

derived from the previous step have been taken more into account (apart from the greater likelihood to which they may lead).

Section IV concerns the Unscented Particle Filter (UPF, [22]). This filter has features both of a particle filter and of a Gaussian sum filter. To each particle corresponds a Gaussian distribution, whose mean and covariance matrix are computed using the equations of the Unscented Kalman Filter. However, from the distribution obtained from the correction step, a sample is drawn randomly, and the particle weight is computed as in Particle Filters. In this paper a correction is proposed with respect to the covariance matrix, so as to take into account the fact that the particles are not independent, each one constituting a part of the whole filter, thus the equations of the Unscented Kalman Filter are not the most suitable.

In the last section an example of a nonlinear system is studied. This system has been examined extensively in the literature. The modified techniques are compared with the standard ones both in terms of performance as well as of computational cost. It turns out that the proposed changes improve the performance of the filters, with limited or no increase in computational cost.

The following section provides the formulation of the problem, introduces the notation used in the rest of the paper, and presents some basic results in the area.

## II. Background

Let us consider a system described by

$$x_{k+1} = f(x_k) + w_k,\ k = 0, 1, 2, \dots \tag{1}$$

$$y_k = h(x_k) + v_k,\ k = 1, 2, \dots \tag{2}$$

where $x_k$ is the state of the system and $y_k$ the measured output at time $k$. $w_k$ is the disturbance, also referred to as process noise, and $v_k$ is the measurement noise.

In this paper it is assumed that the random variables $x_0$, $w_k$, $k = 0, 1, 2, \dots$ and $v_k$, $k = 1, 2, \dots$ are mutually independent and normally distributed with known parameters. Furthermore, $w_k$ and $v_k$ have zero mean. Measurements are available from time $k = 1$ onwards.

Suppose that $p_{X_0}(x_0)$ is the probability density function (pdf) of $x_0$, $p_V(v_k)$ is the pdf of the measurement noise and $p_W(w_k)$ is the pdf of the process noise. It holds $p_{Y|X}(y_k|x_k) = p_V(y_k - h(x_k))$ and $p_{X_{k+1}|X_k}(x_{k+1}|x_k) = p_W(x_{k+1} - f(x_k))$. The subscripts of probability density functions will be omitted for convenience. Let us define $y_{1:k} = \{y_1, y_2, \cdots, y_k\}$.

Then, according to Bayes Rule ([23], see also [24] and [25]) the following recursive equations hold:

$$p(x_{k+1}|y_{1:k}) = \int p(x_{k+1}|x_k)p(x_k|y_{1:k})dx_k, \tag{3}$$

$$p(x_{k+1}|y_{1:k+1}) = p(y_{k+1}|x_{k+1})p(x_{k+1}|y_{1:k})/c_k, \tag{4}$$

where

$$c_k = \int p(y_{k+1}|x_{k+1})p(x_{k+1}|y_{1:k})dx_{k+1}. \tag{5}$$

However the above integrals cannot be evaluated analytically. Numerical integration for a sufficiently dense mesh of $x_k$ at each time step is also impractical except for the case of low-dimensional $x_k$, so (3)–(5) are mainly of theoretical interest.

As stated in the introduction, the problem can be solved exactly when the system is linear. This is done as follows: Suppose that a dynamical system is described by $x_{k+1} = A_k x_k + B_k + w_k$, $y_k = C_k x_k + D_k + v_k$ where $w_k$ and $v_k$ are normally distributed with zero mean, while their covariance matrices are $Q$ and $R$ respectively. Suppose also that it is known that $x_k$ follows the normal distribution with mean $\hat{x}_k$ and covariance $P_{x_k}$. Then, a priori with respect to $y_{k+1}$, $x_{k+1}$ follows the normal distribution with mean $\hat{x}^-_{k+1}$ and covariance $P^-_{x_{k+1}}$ given by $\hat{x}^-_{k+1} = A_k\hat{x}_k + B_k$, $P^-_{x_{k+1}} = A_k P_{x_k} A_k^T + Q$. The predicted value of $y_{k+1}$ is then $\hat{y}^-_{k+1} = C_{k+1}\hat{x}^-_{k+1} + D_k$ while its covariance is $P^-_{y_{k+1}} = C_{k+1}P^-_{x_{k+1}}C^T_{k+1} + R$, and the cross covariance of $x_{k+1}$ and $y_{k+1}$ is equal to $P^-_{x_{k+1}y_{k+1}} = P^-_{x_{k+1}}C^T_{k+1}$. The value of $y_{k+1}$ can be then used to refine the distribution, according to the following equations:

$$\begin{aligned} K_{k+1} &= P^-_{x_{k+1}y_{k+1}}P^{-1}_{y_{k+1}}, \\ \hat{x}_{k+1} &= \hat{x}^-_{k+1} + K_{k+1}\left(y_{k+1} - \hat{y}^-_{k+1}\right), \\ P_{x_{k+1}} &= P^-_{x_{k+1}} - K_{k+1}P^-_{y_{k+1}}K^T_{k+1}. \end{aligned} \tag{6}$$

*Remark 1:* If the Gaussian assumption is removed, KF gives the linear minimum covariance estimate of the state [1], [3].

For nonlinear systems, even if the noise is Gaussian, the state distribution is not Gaussian. As stated in the introduction, some techniques have been developed which approximate the state distribution using a Gaussian, and in each step they renew its parameters. Since this paper deals with Particle Filtering, no more details are provided here.

*Remark 2:* One more reason that approximate techniques try to estimate the mean value of the state is that it is the least square error estimator. Specifically, if $x$ is a r.v., $\mathbb{E}[x]$ is its expected value and $\hat{x}$ an estimate of $x$, it holds that $\mathbb{E}\left[(x-\hat{x})(x-\hat{x})^T\right] = \mathrm{V}[x] + (\mathbb{E}[x] - \hat{x})(\mathbb{E}[x] - \hat{x})^T$. Few techniques that do not make the Gaussian approximation the state distribution were introduced few years after the appearance of KF. In [26], the pdf of the state is approximated by the pdf of a Gaussian multiplied by a polynomial. The prediction and correction steps adapt also the polynomial coefficients. The one-dimensional case is studied, yet complex formulae arise. Additionally, the functions obtained may not be probability density functions (pdf), as it is not guaranteed that they are non-negative.

[16] proposes the approximation of the pdf by the sum of the pdf of Gaussians. The pdf obtained are now guaranteed to have the attributes that a function must have so that it is a pdf. Several variants are presented in [16] and the case of linear systems with non-Gaussian noise is studied in detail. In all variants the filter is functioning deterministically.

In the '90s, the Particle Filter (PF) appeared. Today the respective literature is very extensive (see, e.g., the reviews

[19], [20]). Many variants exist, often called with other names, such as "Condensation Filter", "Sequential Monte Carlo Filter", "Sequential Imputations" as well as others.

In the PF, the pdf of the state is approximated by a set of particles, each of them representing a Dirac function with a corresponding weight, namely the approximation is

$$p(x_k|y_{1:k}) \approx \sum_{i=1}^{N} W_k^i \delta_{x_k^i}(x_k). \quad (7)$$

Then, the mean value of a function $g$ of the state can be estimated using the following equation.

$$\mathbb{E}[g(x_k)] \approx \sum_{i=1}^{N} W_k^i g(x_k^i) \quad (8)$$

It is reasonable to conjecture that with a sufficiently large number of particles, the approximation will be satisfactory, however the asymptotic analysis of the PF is a difficult problem [27], [28].

There exist many different algorithms (see, e.g., [19], [20], [22]) for the renewal of $W_k^i$ and $x_k^i$. The simplest such algorithm, and probably the first which appeared in the literature [17] is as follows:

Initially, $N$ random samples are drawn according to the initial state distribution. This set is used as an approximation of the initial state distribution. Then, as in the aforementioned filters, a prediction and a correction step are done separately.

For the prediction step, let $x_k^i, i = 1, \ldots, N$ be the set that approximates the distribution of $x_k$ (the weights at this point will be always equal, as they are also for $k = 0$). For each particle, a disturbance sample $w_k^i$ is drawn according to the disturbance distribution, and the new particle $x_{k+1}^{i,-}$ is computed. The set of the new particles constitutes the approximation of the a priori with respect to the measurement $y_{k+1}$ distribution of $x_{k+1}$.

For the correction step, the likelihood $l_k^i = p(y_{k+1}|x_{k+1}^{i,-})$ of each particle is initially computed. Normalizing so that the sum of all weights is equal to 1 yields

$$W_{k+1}^i = \frac{p(y_{k+1}|x_{k+1}^{i,-})}{\sum_{n=1}^{N} p(y_{k+1}|x_{k+1}^{n,-})}. \quad (9)$$

The set of particles $x_{k+1}^{i,-}$ with weights $W_{k+1}^i$ constitutes an approximation of the a posteriori distribution of $x_{k+1}$. The correction step is completed with the resampling process, which serves to avoid the "weight degeneracy" phenomenon [29], [30]. Indeed, if the above renewal procedure were repeated perpetually, and the weights were repeatedly multiplied, then after many steps all particles except for one would have almost zero weight (more detailed explanation of this phenomenon may be found in the related literature [29]).

The resampling procedure consists of drawing $N$ random samples from the distribution $\sum_{i=1}^{N} W_{k+1}^i \delta_{x_{k+1}^i}(x_{k+1})$, namely for all particles $x_{k+1}^j, j = 1, \ldots, N$ it holds that $P(x_{k+1}^j = x_{k+1}^{i,-}) = W_{k+1}^i$. This way, it is more probable that more new particles will be created from the particles which had high likelihood, thus more emphasis is given to regions of the state space where it is more probable that the state of the system lies in. The new particles, since have been chosen randomly, have equal weights.

## III. Modified Auxiliary Particle Filter

In this section, the Auxiliary Particle Filter is first presented in more detail and then the proposed modifications follow. It is assumed that the system whose state is to be estimated is described by (1)-(2).

Let $x_{k-1}^i, W_{k-1}^i, i = 1 \ldots N_p$ the particles and weights derived from step $k-1$, and $y_k$ the output value for time step $k$. In the simplest variants of the Particle Filter, at this point, from each particle a new particle would be derived according to the (stochastic) dynamics of the system, and this procedure would constitute the prediction step. However, the Auxiliary Particle Filter algorithm at this point estimates how important will be the contribution of the particles which will arise. The particle that will arise from $x_{k-1}^i$ is a r.v. equal to $f(x_{k-1}^i)+w_{k-1}^i$ ($w_{k-1}^i$ is not to be confused with $W_{k-1}^i$). As a representative value of this r.v. can be considered its mean value, its mode (they coincide for a Gaussian distribution) or a random realization.

Let $\mu_k^i$ this representative value. Then it is reasonable to consider, especially if the variance of the disturbance is relatively small, that the $i$-th particle will have a likelihood about $p(y_k|\mu_k^i)$. To this purpose, the weight $W_{k-1}^i p(y_k|\mu_k^i)$ is assigned to the $i$-th particle ($W_{k-1}^i$ from the previous step and $p(y_k|\mu_k^i)$ from this auxiliary procedure). According to these weights, $x_{k-1}^i, i = 1 \ldots N_p$ are resampled and the particles with indices $j^i, , i = 1 \ldots N_p$ are derived. The indices of particles with large weights are expected to appear more times, while the indices of particles with small weights have a significant probability to not appear at all.

From the particles derived by the resampling, $x_{k-1}^{j^i}$, new particles $x_k^i = f(x_{k-1}^{j^i})+w_{k-1}^i, i = 1 \ldots N_p$ are derived. The measurement $y_k$ has a likelihood of $p(y_k|x_k^i)$. But, because it has to be taken into account that the particle $x_{k-1}^{j^i}$, from which $x_k^i$ was derived, had $p(y_k|\mu_k^i)$ as a factor in its weight, the weight of the $i$-th particle in step $k$ is $W_k^i = \frac{p(y_k|x_k^i)}{p(y_k|\mu_k^{j^i})}$. The algorithm is presented in diagram form in [19], [20].

The first proposed change concerns this last point, namely the weight assigned to $x_k^i$. The above algorithm compensates the factor $p(y_k|\mu_k^{j^i})$ due to the estimated weighted by dividing with this factor. This, however, has the disadvantage that the number of particles actually derived from $x_{k-1}^{j^i}$ is not taken into account. Thus, if from this particle $s^{j^i}$ new particles arise through resampling, each one will have a weight (before the correction step) equal to $\frac{1}{p(y_k|\mu_k^{j^i})}$, therefore their total weight will be $s^{j^i} \cdot \frac{1}{p(y_k|\mu_k^{j^i})}$. This weight is a r.v. with respect to the resampling, since the number $s^{j^i}$ is not predetermined. This feature, although the expected value

of this total weight is proportional to

$$W_{k-1}^{j^i} p(y_k|\mu_k^{j^i}) \frac{1}{p(y_k|\mu_k^{j^i})} = W_{k-1}^{j^i} \quad (10)$$

(the expected value of $s^{j^i}$ is proportional to $W_{k-1}^{j^i} p(y_k|\mu_k^{j^i})$ since this was the weight of $x_{k-1}^{j^i}$ in the resampling) is not desirable, because the variance of the r.v. $s^{j^i}$ introduces noise (even though it may be a small quantity) into the filter.

*Remark 3:* The above weights are not normalized. What is important, however, is that for two different particles, $x_{k-1}^{j^{i_1}}$ and $x_{k-1}^{j^{i_2}}$, the expected values of the weights of the particles derived from these two pre-resampling particles, have a ratio equal to $W_{k-1}^{j^{i_1}}/W_{k-1}^{j^{i_2}}$. In any case, after the procedure under study, weight normalization follows. Similarly for the following case.

The proposed modification is to set the weight of $x_k^i$ after resampling to be $\frac{W_{k-1}^{j_i}}{s^{j_i}}$. Then the total weight of the particles derived from $x_{k-1}^{j^i}$ will be certainly $s^{j^i}\frac{W_{k-1}^{j_i}}{s^{j_i}} = W_{k-1}^{j_i}$. It is possible to assign the aforementioned value because the values are set after resampling, therefore the value of $s^{j_i}$ has been determined. It is only required that the resampling algorithm counts the number of copies derived from each initial particle, something that does not incur significant cost.

The expediency of the proposed modification can be highlighted with the following example. Assume that two different particles, $x_{k-1}^{i_1}$ and $x_{k-1}^{i_2}$, have equal weights before resampling (for simplification it is also assumed that they have equal weights from the previous step), and these weights are such that according to the total number of particles, 1.5 particles correspond to each of these two particles, namely $W_{k-1}^{i_1} = W_{k-1}^{i_2}, W_{k-1}^{i_1} p(y_k|\mu_k^{i_1}) = W_{k-1}^{i_2} p(y_k|\mu_k^{i_2}) = \frac{1.5}{N_p}\sum_{i=1}^{N_p} W_{k-1}^i p(y_k|\mu_k^i)$. It is possible that from the first particle one new particle is derived, while from the second two, i.e. $s^{i_1} = 1, s^{i_2} = 2$. The algorithm which is mostly used in the literature will assign to the three new particles equal weights, consequently $x_{k-1}^{i_2}$ will indirectly receive a weight double than that of $x_{k-1}^{i_1}$. On the contrary, the proposed algorithm will give to the particles derived from $x_{k-1}^{i_2}$ half weight, thus the total weights will be equal.

The second proposed modification concerns how much the resampling is influenced by the auxiliary weights $p(y_k|\mu_k^i)$. The existing algorithms create a number of particles proportional to the auxiliary weight $p(y_k|\mu_k^i)$. However, this is not necessarily optimal. Assume that from the previous step, there have been derived particles in two different regions of the state space, and that with the new measurement their auxiliary weights are different. In any case, it is reasonable to create more particles in the region where the weight is greater. To find how many more should be created in the region with the greater weight, it is necessary to know how good the approximation of the distribution in a region is, as a function of the number of particles in it.

In the literature, it is often considered that the deviation is inversely proportional to the number of particles. This is done because under suitable assumptions [27], there are bounds of the mean square error of the estimation of the mean of a function, which are inversely proportional to the number of particles. Therefore, the rms value of the deviation is inversely proportional to the square root of the number of particles.

Thus, if there exists a region with probability $p_1$ and another region with probability $p_2$, and if $n_1$ particles exist in the first and $n_2$ in the second, it is possible to consider that the deviation will be proportional to $\frac{p_1}{n_1} + \frac{p_2}{n_2}$ or $\frac{p_1}{\sqrt{n_1}} + \frac{p_2}{\sqrt{n_2}}$. If $n$ is the total number of particles to be devoted for the approximation in these two regions, then for the first case $\frac{p_1}{n_1} + \frac{p_2}{n-n_1}$ has to be minimized. Differentiating this yields

$$\frac{d}{dn}\left(\frac{p_1}{n_1} + \frac{p_2}{n-n_1}\right) = -\frac{p_1}{n_1^2} + \frac{p_2}{(n-n_1)^2} = -\frac{p_1}{n_1^2} + \frac{p_2}{n_2^2}, \quad (11)$$

showing that we have a minimum for $n_1/n_2 = \sqrt{p_1/p_2}$. Thus it may be concluded that it is reasonable to use auxiliary weights proportional to $\sqrt{p(y_k|\mu_k^i)}$.

*Remark 4:* In the above minimization problem, it is possible that the value obtained through the differentiation procedure does not correspond to integer $n_1, n_2$. Then the minimum will be for one of the integer values aside the non-integer minimum. But, since the algorithm actually determines weights for the resampling, which need not be integer, there is no problem in using weights equal to $\sqrt{p(y_k|\mu_k^i)}$. Similarly for the following case.

If $\frac{p_1}{\sqrt{n_1}} + \frac{p_2}{\sqrt{n_2}}$ is to be minimized, differentiating yields

$$\frac{d}{dn}\left(\frac{p_1}{\sqrt{n_1}} + \frac{p_2}{\sqrt{n-n_1}}\right) = \frac{1}{2}\left(-\frac{p_1}{{n_1}^{3/2}} + \frac{p_2}{(n-n_1)^{3/2}}\right) = \frac{1}{2}\left(-\frac{p_1}{n_1^{3/2}} + \frac{p_2}{n_2^{3/2}}\right). \quad (12)$$

This viewpoint, consequently, suggests using auxiliary weights proportional to $\sqrt[3]{p(y_k|\mu_k^i)^2}$.

*Remark 5:* Taking into account that it is more reasonable to consider as independent the deviations in different regions, and then the mean square values are added and not the rms values, the last viewpoint may not seem equally natural to that with the mean square values. In any case, however, these bounds only serve to find a reasonable function through which the likelihoods are taken into account, it is thus of purpose to examine this approach, too.

The modifications proposed in this section are applied in a numerical example in Section V.

## IV. Modified Unscented Particle Filter

The Unscented Particle Filter [22] is a technique having some common features with the Gaussian sum techniques. To each particle corresponds a Gaussian distribution. For the prediction step, its parameters are updated as in the Unscented Kalman Filter, separately for each particle. The correction step starts again with the correction step of the Unscented Kalman Filter for each particle separately. However, after

this procedure, from each distribution that arises, a sample is drawn randomly, which constitutes the new particle, while its weight is calculated as in the Particle Filters.

Namely, if the $i$-th particle for time step $k-1$ was $x_k^{i-1}$, and the new distribution has mean $\hat{x}_k^i$ and covariance matrix $P_k^i$, $\tilde{x}_k^i \sim \mathcal{N}(\hat{x}_k^i, P_k^i)$ will be drawn randomly and its corresponding weight is

$$W_k^i = \frac{p(y_k|\tilde{x}_k^i)p(\tilde{x}_k^i|x_k^{i-1})}{p(\tilde{x}_k^i|\hat{x}_k^i, P_k^i)}, \tag{13}$$

where $p(\tilde{x}_k^i|\hat{x}_k^i, P_k^i)$ is the value of the p.d.f. $\mathcal{N}(\hat{x}_k^i, P_k^i)$ evaluated at $\tilde{x}_k^i$.

The approximation of the a posteriori distribution for the time step $k$ is provided by $\tilde{x}_k^i, W_k^i$. Before the repetition of the procedure for the time $k+1$, a resampling is done, through which $x_k^i = \tilde{x}_k^{j_i}$ arise. To $x_k^i$, the algorithm of the Unscented Particle Filter assigns the covariance matrix $P_k^{j^i}$, which will be used in the equations of the Unscented Kalman Filter at the next step.

The rationale of the above algorithm is that the Gaussian distributions $\mathcal{N}(\hat{x}_k^i, P_k^i), i = 1, \ldots, N_p$ consist a relatively good approximation of the a posteriori distribution, it is therefore good to sample the particles from them, since it is optimal to be sampled from the a posteriori distribution. Obviously, the a posteriori distribution can be approximated by the sum

$$\sum_{i=1}^{N_p} \frac{1}{N_p} \mathcal{N}(\hat{x}_k^i, P_k^i) \tag{14}$$

and not by each distribution separately. However, the Unscented Kalman Filter equations have been used for each particle separately, therefore the covariance matrices corresponding to the particles are closer to the covariance matrix of the total distribution, and not to the covariance matrix that each particle should have so that the sum (14) constitutes a good approximation of the a posteriori distribution.

The following viewpoint provides a way to choose the covariance matrices so that the total distribution approximates the a posteriori distribution. Assume that the particle locations are independent and follow the a posteriori distribution, which has mean $\bar{x}$ and covariance matrix $P$. This is the desired, although in practice it does not fully hold. If a r.v. $v_i$ with mean $x_i$ and covariance $P_i$ corresponds to each particle, then the approximation of the a posteriori distribution is the r.v. $v = \sum_{i=1}^{N_p} I_i v_i$, where each of $I_i, i = 1, \ldots, N_p$ is equal to 1 with probability $\frac{1}{N_p}$, otherwise it is equal to 0, while always exactly one of them is equal to 1, namely they constitute the indicator functions of the events $[v = v_i]$.

Then, for given $x_i, i = 1, \ldots, N_p$, the covariance matrix of the total distribution is

$$\mathrm{V}[v] = \mathrm{V}\left[\sum_{i=1}^{N_p} I_i v_i\right] =$$

$$= \mathbb{E}\left[\left(\sum_{i=1}^{N_p} I_i v_i - \frac{1}{N_p}\sum_{i=1}^{N_p} x_i\right)\left(\sum_{i=1}^{N_p} I_i v_i - \frac{1}{N_p}\sum_{i=1}^{N_p} x_i\right)^T\right]$$

$$= \mathbb{E}\left[\left(\sum_{i=1}^{N_p}\left(I_i v_i - \frac{x_i}{N_p}\right)\right)\left(\sum_{i=1}^{N_p}\left(I_i v_i - \frac{x_i}{N_p}\right)\right)^T\right] =$$

$$= \sum_{i=1}^{N_p}\sum_{j=1}^{N_p} \mathbb{E}\left[\left(I_i v_i - \frac{x_i}{N_p}\right)\left(I_j v_j - \frac{x_j}{N_p}\right)^T\right] \tag{15}$$

For $i = j$ it holds that

$$\mathbb{E}\left[\left(I_i v_i - \frac{x_i}{N_p}\right)\left(I_i v_i - \frac{x_i}{N_p}\right)^T\right] =$$

$$= \mathbb{E}\left[I_i^2 v_i v_i^T - I_i \frac{1}{N_p} v_i x_i^T - I_i \frac{1}{N_p} x_i v_i^T + \frac{1}{N_p^2} x_i x_i^T\right] =$$

$$= \frac{\mathbb{E}\left[v_i v_i^T\right]}{N_p} - \frac{1}{N_p^2} x_i x_i^T = \frac{P_i + x_i x_i^T}{N_p} - \frac{1}{N_p^2} x_i x_i^T, \tag{16}$$

while for $i \neq j$

$$\mathbb{E}\left[\left(I_i v_i - \frac{x_i}{N_p}\right)\left(I_j v_j - \frac{x_j}{N_p}\right)^T\right] =$$

$$= \mathbb{E}\left[I_i I_j v_i v_j^T - I_i \frac{1}{N_p} v_i x_j^T - I_j \frac{1}{N_p} x_i v_j^T + \frac{1}{N_p^2} x_i x_j^T\right] =$$

$$= -\frac{1}{N_p^2} x_i x_j^T \tag{17}$$

Therefore $\mathrm{V}[v]$ depends on the values of $x_i$. It holds that $\mathbb{E}\left[x_i x_i^T\right] = P + \bar{x}\bar{x}^T$, while for $i \neq j$ it holds that $\mathbb{E}\left[x_i x_j^T\right] = \bar{x}\bar{x}^T$. Thus it follows that

$$\mathrm{V}[v] = \frac{\sum_{i=1}^{N_p} P_i}{N_p} + N_p\left(\frac{1}{N_p} - \frac{1}{N_p^2}\right)(P + \bar{x}\bar{x}^T) -$$

$$- \frac{N_p^2 - N_p}{N_p^2}\bar{x}\bar{x}^T = \frac{\sum_{i=1}^{N_p} P_i}{N_p} + \left(1 - \frac{1}{N_p}\right)P. \tag{18}$$

It is, therefore, concluded that $\mathrm{V}[v] = P \Leftrightarrow \sum_{i=1}^{N_p} P_i = P$. Furthermore, for the case where $P_i, i = 1, \ldots, N_p$, are equal, the condition is $P_i = \frac{1}{N_p} P$.

The above analysis shows that having each one of the matrices $P_i$ approximately equal to $P$ is not optimal, on the contrary they have to be equal to about $\frac{1}{N_p} P$. Given that the Unscented Kalman Filter equations seek to approximate this value, it is reasonable, after the resampling, to multiply the matrices $P_k^i$ by $\frac{\alpha}{N_p}$. The factor $\alpha$ is introduced because the convergence is not done in one step, therefore beginning from a matrix of the order of $\frac{P}{N_p}$, after the next step the matrix will be of the order of $\frac{1}{\alpha} P$. The value of $\alpha$ can be

Table I
TABLE OF THE RMS VALUES OF THE ESTIMATION ERROR FOR SEVERAL PARTICLE FILTER VARIANTS

| Filter | Mean Value | Standard Deviation | Maximum Value | Comp. Time [2] (ms) |
|---|---|---|---|---|
| PF-SIR | 5.41541 | 1.34547 | 11.63975 | 0.18850 (3.56) |
| APF | 5.37662 | 1.18517 | 9.68630 | 0.24240 |
| APF-$\sqrt{\ }$ | 5.31901 | 1.13784 | 9.32619 | 0.24800 |
| APF-$\sqrt[3]{\ }^{-2}$ | 5.33066 | 1.13710 | 9.33985 | 0.31380 |
| APF-MR | 5.33339 | 1.16558 | 9.42745 | 0.28180 |
| APF-MR-$\sqrt{\ }$ | 5.18590 | 1.11002 | 9.24750 | 0.28530 |
| APF-MR-$\sqrt[3]{\ }^{-2}$ | 5.19944 | 1.13456 | 9.58607 | 0.35500 |
| UPF | 5.90033 | 1.78609 | 13.34652 | 20.90770 |
| UPF - $\alpha = 0.1$ | 5.08705 | 1.37431 | 13.44550 | 20.91310 |
| UPF - $\alpha = 0.2$ | 5.03754 | 1.27727 | 12.69427 | 20.90010 |
| UPF - $\alpha = 1$ | 5.12051 | 1.27182 | 12.59017 | 20.90210 |
| UPF - $\alpha = 0$ | 5.02658 | 1.20974 | 12.95750 | 12.98660 |

determined experimentally. In any case, the choice $\alpha = 1$ is also expected to give good enough results, especially if the convergence is fast enough.

It is also possible to set the matrices $P_k^i$ equal to 0, which is equivalent to $\alpha = 0$. This choice has the advantage that in this case, the prediction step of the Unscented Kalman Filter is done with no computational cost, with the a priori mean equal to $f(x_{k-1}^i)$ and the a priori covariance matrix equal to the covariance matrix of the disturbance. The computational cost reduction gives the possibility to deploy a greater number of particles, in which case the term $\frac{P}{N_p}$ will be even closer to 0.

The proposed modifications are tested in a numerical example in the following section.

## V. NUMERICAL EXAMPLE AND COMMENTS

The system to which the different variants of the Particle Filter will be applied for comparison purposes is described by

$$x_{k+1} = \frac{x_k}{2} + 25\frac{x_k}{1+x_k^2} + 8\cos(1.2k) + w_k, \tag{19}$$

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

and appeared for the first time in [31]. Since, it has been extensively studied in the literature [17]–[20]. $x_0$, $w_k$ and $v_k$ are independent r.v. following the Gaussian distribution with zero mean. $x_0$ has a covariance equal to 0.001 while, as in [17], [19], [20], for the covariances of $w_k$ and $v_k$ it is assumed that $\mathrm{V}[w_k] = 10,\ \mathrm{V}[v_k] = 1$. The system is run up to $k = 100$.

1000 repetitions have been performed and the results are presented in Table I. PF-SIR is the Sampling Importance Resampling Filter of the literature [19]. APF is the Auxiliary Particle Filter, for all whose variants the auxiliary weights are computed for a random realization of the r.v. $f(x_{k-1}^i)+w_{k-1}^i$, as in [19]. $\sqrt{\ }$ and $\sqrt[3]{\ }^{-2}$ denote the proposed in Section III modifications in the way that the auxiliary weight is taken into account, while MR (Modified Reweighting) denotes the proposed modification in the redetermination of the weights after resampling. UPF is the Unscented Particle Filter, and $\alpha$ is the parameter referred in Section IV. The resampling is performed using the algorithm proposed in [18], while as in the software [32], a very small value is added to the likelihoods so as to exclude the event that they are all 0 due to rounding. In all variants 50 particles have been used.

It is concluded that the proposed modifications to the APF improve its performance, both if applied separately as well as if applied in combination, in which latter case the best results are obtained. Concerning the proposed modification to the UPF, among the various values of $\alpha$ tested, the best results were for $\alpha = 0$, choice which, as already mentioned, has also a computational advantage. An interesting fact is that while UPF can, with proper adjustment, yield a small error in the mean case, in the worst case its error is greater even than that of PF-SIR.

The computational costs of the various proposed techniques are different, but for PF and the variants of APF the differences are not big, as the differences among the variants of UPF. However the MATLAB implementation which has been made leads to high cost for UPF, because the computations in this case are not made "vectorized". This has been made because using MATLAB it is easy to compute, for example, all the particle weights simultaneously, or to create all new particles using one function call, while the Unscented Kalman Filter equations were applied separately to each particle in the implementation made. If the particle computations for PF-SIR are also made non-vectorized, the cost is that appearing in parenthesis in Table I.

The conclusion from the numerical results is that the proposed modification can reduce the estimation error, and the choice of the variant that will be used can be made taking into account the performance of the various variants in the problem under study as well as the technical details of the implementation that will be made.

[2]See the penultimate paragraph of this section.

## ACKNOWLEDGMENTS

Alexandros Charalampidis wishes to thank the Special Account of Research of the National Technical University

of Athens for supporting his doctoral research on Estimation Theory.

The research of George P. Papavassilopoulos has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALES, Investing in knowledge society through the European Social Fund. Support has been also provided by the program ARISTEIA, project HEPHAISTOS.

## REFERENCES

[1] Kalman, R. E.: *A new approach to linear filtering and prediction problem*. ASME Journal of Basic Engineering, 82(1):35–45, Mar. 1960.

[2] Anderson, B. D. O. and J. B. Moore: *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ, 1979.

[3] Goodwin, G. C. and K. S. Sin: *Adaptive Filtering Prediction and Control*. Dover Publications, New York, NY, USA, 2009.

[4] Simon, D.: *Optimal State Estimation: Kalman, $H_\infty$ and Nonlinear Approaches*. John Wiley, Hoboken, NJ, 2006.

[5] Kalman, R. E. and R. S. Bucy: *New Results in Linear Filtering and Prediction Theory*. Journal of Basic Engineering, 83D:95–108, 1961.

[6] Smith, G. L., S. F. Schmidt, and L. A. McGee: *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle*. Technical Report TR R-135, NASA, 1962.

[7] Julier, S., J. K. Uhlmann, and H. F. Durrant-Whyte: *A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators*. IEEE Transactions on Automatic Control, 45(3):477–482, Mar. 2000.

[8] Julier, S. and J. K. Uhlmann: *Unscented filtering and nonlinear estimation*. Proceedings of the IEEE, 92(3):401–422, Mar. 2004.

[9] Lefebvre, T., H. Bruyninckx, and J. d. Schutter: *Kalman filters for nonlinear systems: a comparison of performance*. International Journal of Control, 77(7):639–653, May 2004.

[10] Simandl, Miroslav and Jindrich Duník: *Derivative-free estimation methods: New results and performance analysis*. Automatica, 45(7):1749–1757, July 2009, ISSN 0005-1098.

[11] Arasaratnam, I. and S. Haykin: *Cubature Kalman Filters*. Automatic Control, IEEE Transactions on, 54(6):1254–1269, June 2009, ISSN 0018-9286.

[12] M. St-Pierre and D. Gingras: *Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system*. In *Proc. 2004 IEEE Intelligent Vehicles Symposium*, pages 831–835, Jun. 2004.

[13] Charalampidis, A. C. and G. P. Papavassilopoulos: *Computationally Efficient Kalman Filtering for a Class of Nonlinear Systems*. IEEE Transactions on Automatic Control, 56(3):483–491, Mar. 2011.

[14] Charalampidis, A. C. and G. P. Papavassilopoulos: *Kalman Filtering for a Generalized Class of Nonlinear Systems and a New Gaussian Quadrature Technique*. IEEE Transactions on Automatic Control, 57(11):2967–2973, Nov. 2012.

[15] Charalampidis, A. C. and G. P. Papavassilopoulos: *Development and Numerical Investigation of New Nonlinear Kalman Filter Variants*. IET Control Theory and Applications, 5(10):1155–1166, 2011.

[16] Sorenson, H. W. and D. L. Alspach: *Recursive Bayesian estimation using Gaussian sums*. Automatica, 7(4):465–479, 1971.

[17] Gordon, N. J., D. J. Salmond, and A. F. M. Smith: *Novel approach to nonlinear/non-Gaussian state estimation*. IEE Proceedings-F, 140(2):107–113, Apr. 1993.

[18] Kitagawa, G.: *Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Modules*. Journal of Computational and Graphical Statistics, 5(1):1–25, Mar. 1996.

[19] Sanjeev Arulampalam, M., S. Maskel, N. Gordon, and T. Clapp: *A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking*. IEEE Transactions on Signal Processing, 50(2):174–188, Feb. 2002.

[20] Cappé, O., S. J. Godsill, and E. Moulines: *An overview of existing methods and recent advances in sequential Monte Carlo*. Proceedings of the IEEE, 95(5):899–924, 2007.

[21] Pitt, M. K. and N. Shephard: *Filtering via simulation: Auxiliary particle filters*. Journal of the American Statistical Association, 94(446):590–599, June 1999.

[22] van der Merwe, R., A. Doucet, N. de Freitas, and E. Wan: *The Unscented Particle Filter*. Technical Report CUED/F-INFENF/TR 380, Cambridge University, Aug. 2000.

[23] Bayes, T.: *An Essay towards solving a Problem in the Doctrine of Chances*. Philosophical Transactions of the Royal Society of London, 53:370–418, 1763.

[24] Billingsley, P.: *Probability and Measure*. Wiley-Interscience, Hoboken, NJ, 3rd edition, 1995.

[25] Ash, R. B.: *Basic Probability Theory*. Dove, Mineola, NY, 2008. Available: www.math.uiuc.edu/∼r-ash/BPT.html.

[26] Sorenson, H. W. and A. R. Stubberud: *Non-linear filtering by approximation of the a posteriori density*. International Journal of Control, 8(1):33–51, 1968.

[27] Crisan, D. and A. Doucet: *A survey of convergence results on particle filtering methods for practitioners*. IEEE Transactions on Signal Processing, 50(3):736–746, Mar. 2002.

[28] Hu, X. L., T. B. Schon, and L. Ljung: *A Basic Convergence Result for Particle Filtering*. IEEE Transactions on Signal Processing, 56(4):1337–1348, Apr. 2008.

[29] Cappé, Olivier: *An Introduction to Sequential Monte Carlo for Filtering and Smoothing*. http://www-irma.u-strasbg.fr/~guillou/meeting/cappe.pdf.

[30] Orhan, Emin: *Particle Filtering*. http://www.bcs.rochester.edu/people/eorhan/notes/particle-filtering.pdf.

[31] Netto, M., L. Gimeno, and M. Mendes: *On the optimal and suboptimal nonlinear filtering problem for discrete-time systems*. Automatic Control, IEEE Transactions on, 23(6):1062–1067, Dec. 1978, ISSN 0018-9286.

[32] de Freitas, N. and R. van der Merwe: *http://www.cs.ubc.ca/∼nando/software/upf_demos.tar.gz*, Aug. 2000. MATLAB Code.