# Sigma-point multiple particle filtering

Luis Úbeda-Medina, Ángel F. García-Fernández, Jesús Grajal

**Abstract**

In this paper, we introduce two new particle filtering algorithms for high-dimensional state spaces in the multiple particle filtering approach. In multiple particle filtering, the state space is partitioned and a different particle filter is used for each component of the partition. At each time step, all particle filters share information about their marginal densities so that they can adequately approximate the filtering recursion. In this paper, we propose a second order approximation to the involved densities based on sigma-point integration methods. We then introduce two different particle filters that make use of this strategy. Finally, we demonstrate their remarkable performance through simulations of a multiple target tracking scenario with a sensor network.

*Keywords:* Particle filters, curse of dimensionality, unscented transform, sigma-point, multiple particle filter.

## 1. Introduction

Many problems in science and engineering require the estimation of the state of a dynamic system based on a sequence of noisy measurements. Advances in this field have allowed the development of a plethora of applications in navigation, aerospace engineering, remote surveillance, telecommunications, control theory and finance, among others [1]. Estimation theory provides different methods to tackle the state estimation problem [2]. However, it is usually the case that sensor measurements are gradually made available at progressive time instants, so that

*Luis Úbeda-Medina and Jesús Grajal are with the Departamento de Señales, Sistemas y Radiocomunicaciones, ETSI de Telecomunicación, Universidad Politécnica de Madrid, Ciudad Universitaria, 28040 Madrid, Spain. Email: luis.ubeda@upm.es, jesus.grajal@upm.es.

**Ángel F. García-Fernández is with the Department of Electrical Engineering and Electronics, University of Liverpool, Brownlow Hill, L69 3GJ, Liverpool, UK and with the ARIES Research Center, Universidad Antonio de Nebrija, C/ Pirineos 55, 28015 Madrid, Spain. Email: angel.garcia-fernandez@liverpool.ac.uk.

This work has been supported by the Spanish Ministry of Science, Innovation and Universities within the project TEC2017-87061-C3-1-R (CIENCIA/AEI/FEDER, UE)

online estimators given past states are usually preferred, making Bayesian techniques in estimation theory particularly suitable for this problem.

The Bayesian filtering approach to the estimation of the current state of a dynamic system requires the sequential computation of the posterior probability density function (PDF), which contains all the information of interest about the current state of the system given the sequence of measurements up to the current time [1]. However, the posterior PDF does not generally have a closed form expression and is not described by a finite set of parameters unless some assumptions hold [3]. Thus, approximations are usually required in order to face this problem in a general setting.

Nonlinear Kalman filters can be used to deal with nonlinear problems, of low and high dimensionality, using a Gaussian approximation [4–6]. These methods can work very well if the posterior PDF is roughly Gaussian [1] and nonlinearities are mild [7].

Particle filters (PFs) [8–10] are a widely used tool to generally approximate the Bayesian filtering recursion, with general posterior densities, not necessarily Gaussian or unimodal. However, the performance of PFs severely decreases if the dimension of the state space is high, an effect which is usually referred to as the curse of dimensionality [11]. Therefore, it is of interest to develop strategies within the particle filtering framework that can satisfactorily deal with high-dimensionality.

In an attempt to tackle the curse of dimensionality, some useful techniques can be applied to PFs. A common approach in systems with linear/Gaussian substructures is to use Rao-Blackwellization [12, 13]. Another effective approach in the literature to soften the effects of dimensionality in the performance of PFs is to assume that, given a partition of the state of the dynamic system, the different components of the partition are posterior independent [14–18].

The approach of multiple filtering [19–25] has also been proposed to tackle the problem of high dimensional state estimation. Multiple filters also partition the state space into different components to alleviate the curse of dimensionality. However, instead of considering the joint posterior PDF of the complete state, as in the previous posterior independence methods, they separately approximate the marginal posterior PDFs of each component of the partition with an individual filter. The dimension of each of the individual state spaces is thus kept low so that the effects of the curse of dimensionality are mitigated. Multiple filters that make use of particle filters are referred to as multiple particle filters (MPFs).

Computation of the involved marginalization integrals in multiple filtering does not generally admit an analytical result, and approximations via Monte Carlo sampling can result in computationally expensive methods [26]. It is then of high interest in MPFs to develop tractable, accurate approximations to deal with these marginalization integrals. In the following, we proceed to review different approaches to the marginalization step, which result in different MPF algorithms.

A first order approximation to perform the marginalization was considered in the original MPF [19, 20]. Each individual PF receives the predicted mean of the state of the other components of the partition, and use them to compute an approximation of the marginalization integrals. The performance of this MPF can severely decline if the predicted PDF of the state of the components of the partition is not adequately represented by its predicted mean, for instance, when the state uncertainty is relatively high.

An improvement over this first order MPF, coined as the C-PF-PROP, was presented in [22]. This method considers a second order approximation to the involved integrals, so that each PF in the method receives not only the predicted means of the adjacent components of the partition but also the covariance matrices of the predictions. An important drawback of the C-PF-PROP is that it requires complex analytical derivations, which have only been provided for the received signal strength indicator measurement model [27]. In addition, a general procedure of how to extend C-PF-PROP to other measurement models is not explicitly mentioned in [27] so it is not straightforward to use it with other measurement models.

In this paper, we address the shortcomings of the original MPF and C-PF-PROP and develop two multiple particle filters that exchange second order information and can be used with additive measurement models, without the need of analytical derivations. In the proposed methods, we first indicate how we can make use of sigma-point integration methods to efficiently compute the marginalization integrals [1, 4, 5, 28, 29]. The resulting filter presented in this paper is referred to as the sigma-point MPF (SP-MPF). The second contribution of this paper is the inclusion of auxiliary sampling to further improve the performance of SP-MPF [30]. This method is referred to as the sigma-point multiple auxiliary PF (SP-MAPF). We demonstrate via simulations that the two proposed algorithms outperform other PFs in the literature.

The rest of the paper is organized as follows. In Section 2, the general setting of the problem is presented and previous solutions based on MPFs are reviewed. Section 3 introduces the two sigma-point based MPFs presented in this paper. In Section 4, the performance of the presented

methods is tested against other PFs in the literature via simulations. Finally, conclusions are drawn in Section 5.

## 2. Background on multiple particle filtering

In this section, the Bayesian approach to the estimation of the current state of a dynamic system is first considered. The state at time $k$ is described by the state vector $\mathbf{X}^k \in \mathbb{R}^{n_x}$, which is observed through the measurements $\mathbf{z}^k \in \mathbb{R}^{n_z}$, where $n_x$ and $n_z$ are the dimensions of the state space and the measurement vector, respectively. The dynamic system is modeled by the dynamic and measurement equations

$$\mathbf{X}^k = \mathbf{f}\left(\mathbf{X}^{k-1}, \mathbf{w}^{k-1}\right) \tag{1}$$

$$\mathbf{z}^k = \mathbf{h}\left(\mathbf{X}^k, \mathbf{v}^k\right) \tag{2}$$

where the dynamic function $\mathbf{f}(\cdot)$ describes the evolution of the state through time and the measurement function $\mathbf{h}(\cdot)$ describes the dependence of the observations with the state of the system. The functions $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ might be nonlinear functions, $\mathbf{w}^{k-1}$ is the process noise vector of the state at time $k-1$, and $\mathbf{v}^k$ is the measurement noise vector at time $k$ [8]. Noise vectors are independent and have zero mean.

Let $\mathbf{z}^{1:k} = (\mathbf{z}^1, ..., \mathbf{z}^k)$ be the sequence of measurements up to time $k$. The posterior can be computed via Bayesian filtering, using a two-step recursion [1]. Thus, using the Chapman-Kolmogorov equation and Bayes' rule, the posterior PDF at time $k$ can be obtained as

$$p(\mathbf{X}^k|\mathbf{z}^{1:k-1}) = \int p(\mathbf{X}^k|\mathbf{X}^{k-1}) p(\mathbf{X}^{k-1}|\mathbf{z}^{1:k-1}) d\mathbf{X}^{k-1} \tag{3}$$

$$p(\mathbf{X}^k|\mathbf{z}^{1:k}) \propto p(\mathbf{z}^k|\mathbf{X}^k) p(\mathbf{X}^k|\mathbf{z}^{1:k-1}) \tag{4}$$

where $\propto$ denotes proportionality, $p(\mathbf{X}^k|\mathbf{X}^{k-1})$ is obtained from (1) and the likelihood function $p(\mathbf{z}^k|\mathbf{X}^k)$ is obtained from (2). The initial state PDF $p(\mathbf{X}^0)$ is also known in order to initialize the recursion.

Directly computing or approximating (3) and (4) can be a difficult task due to the possible nonlinearities in (1) and (2), specially when the dimensionality of the involved spaces is high. Multiple filters [19–22, 24, 25] address this problem by considering that the state of the dynamic system can be partitioned into $t$ components as [24]

$$\mathbf{X}^k = \left[(\mathbf{x}_1^k)^T, (\mathbf{x}_2^k)^T, ..., (\mathbf{x}_t^k)^T\right]^T \tag{5}$$

where $\mathbf{x}_j^k$ is the state of the $j$-th component at time $k$, and $T$ stands for the vector transpose.

We assume that the dynamic model is such that Equation (1) can be written as

$$\mathbf{x}_j^k = \mathbf{f}_j(\mathbf{x}_j^{k-1}, \mathbf{w}_j^{k-1}) \tag{6}$$

where $\mathbf{f}_j(\cdot)$ describes the evolution of the $j$-th component of the state through time and $\mathbf{w}_j^{k-1}$ is an independent noise vector for the $j$-th component. According to (6), the state of each component evolves independently from the rest so that the transition density is

$$p(\mathbf{X}^k|\mathbf{X}^{k-1}) = \prod_{j=1}^{t} p(\mathbf{x}_j^k|\mathbf{x}_j^{k-1}). \tag{7}$$

While (7) is not a constraint of multiple filters, which can be adapted to consider a broader family of dynamic models [20, 21], it is still a common assumption in multiple filtering [19, 22, 24, 31]. In addition, multiple filters assume that

- A1: The posterior PDF is independent so that

$$p(\mathbf{X}^k|\mathbf{z}^{1:k}) = \prod_{j=1}^{t} p(\mathbf{x}_j^k|\mathbf{z}^{1:k}). \tag{8}$$

It should be noted that, due to the correlation arising from processing the measurements, the posterior PDF is not generally independent as indicated by A1. Nevertheless, it is shown in [16] that, for a Gaussian posterior, the posterior independence assumption improves performance for a high enough state dimension or a low enough number of particles. This assumption has been demonstrated to be beneficial in other high-dimensional particle filters [14, 15, 18].

Making use of (8) and (7), the predicted PDF in (3) can be written as

$$\begin{aligned} p(\mathbf{X}^k|\mathbf{z}^{1:k-1}) &= \prod_{j=1}^{t} \left[ \int p(\mathbf{x}_j^k|\mathbf{x}_j^{k-1}) p(\mathbf{x}_j^{k-1}|\mathbf{z}^{1:k-1}) d\mathbf{x}_j^{k-1} \right] \\ &= \prod_{j=1}^{t} p(\mathbf{x}_j^k|\mathbf{z}^{1:k-1}) \end{aligned} \tag{9}$$

where $p(\mathbf{x}_j^k|\mathbf{z}^{1:k-1})$ is the predicted density of the $j$-th component of the state at time $k$ based on measurements up to time $k-1$.

The aim of multiple filters is to separately approximate the marginal posterior PDF $p(\mathbf{x}_j^k|\mathbf{z}^{1:k})$ for each component of the partition in (5). To explain this procedure, we define vector

$$\mathbf{X}_{-\{j\}}^k = \left[ (\mathbf{x}_1^k)^T, ..., (\mathbf{x}_{j-1}^k)^T, (\mathbf{x}_{j+1}^k)^T, ..., (\mathbf{x}_t^k)^T \right]^T, \tag{10}$$

in which the state of the $j$-th component has been removed from $\mathbf{X}^k$. Using (4) and (9), the

marginal posterior PDF of $\mathbf{x}_j^k$ can then be computed as

$$\begin{aligned} p(\mathbf{x}_j^k|\mathbf{z}^{1:k}) &= \int p(\mathbf{X}^k|\mathbf{z}^{1:k}) d\mathbf{X}_{-\{j\}}^k \\ &\propto p(\mathbf{x}_j^k|\mathbf{z}^{1:k-1}) \int p(\mathbf{z}^k|\mathbf{X}^k) p(\mathbf{X}_{-\{j\}}^k|\mathbf{z}^{1:k-1}) d\mathbf{X}_{-\{j\}}^k \end{aligned} \tag{11}$$

where using (9)

$$p(\mathbf{X}_{-\{j\}}^k|\mathbf{z}^{1:k-1}) = \prod_{l=1, l\neq j}^{t} p(\mathbf{x}_l^k|\mathbf{z}^{1:k-1}). \tag{12}$$

Approximating these marginal posterior distributions instead of the high dimensional joint state posterior PDF generally results in a better performing particle filter [19–22]. MPFs use an individual PF to approximate each marginal posterior PDF of the different components of the partition in (5). However, in order to efficiently achieve its purpose, the $j$-th PF of a MPF approximates the marginal posterior PDF of $\mathbf{x}_j^k$ in an indirect way by considering the whole sequence of states up to time $k$, $\mathbf{x}_j^{0:k} = (\mathbf{x}_j^0, ..., \mathbf{x}_j^k)$. The advantage of this approach is that it is no longer necessary to compute the integral in the prediction step, see (3), so that [8, 10]

$$p(\mathbf{x}_j^{0:k}|\mathbf{z}^{1:k-1}) = p(\mathbf{x}_j^k|\mathbf{x}_j^{k-1}) p(\mathbf{x}_j^{0:k-1}|\mathbf{z}^{1:k-1}) \tag{13}$$

and

$$p(\mathbf{X}^{0:k}|\mathbf{z}^{1:k-1}) = \prod_{j=1}^{t} p(\mathbf{x}_j^{0:k}|\mathbf{z}^{1:k-1}) \tag{14}$$

where (14) corresponds to the extension of Assumption A1 to state sequences. The same discussion regarding the motivation of A1 (paragraph after (8)) applies to (14).

Given that the measurements only depend on the current state of the dynamic system, the marginal posterior PDF can be written as

$$\begin{aligned} p(\mathbf{x}_j^{0:k}|\mathbf{z}^{1:k}) &\propto \int p(\mathbf{z}^k|\mathbf{X}^k) \prod_{l=1}^{t} \left[ p(\mathbf{x}_l^{0:k}|\mathbf{z}^{1:k-1}) \right] d\mathbf{X}_{-\{j\}}^{0:k} \\ &= p(\mathbf{x}_j^k|\mathbf{x}_j^{k-1}) p(\mathbf{x}_j^{0:k-1}|\mathbf{z}^{1:k-1}) \int p(\mathbf{z}^k|\mathbf{X}^k) p(\mathbf{X}_{-\{j\}}^k|\mathbf{z}^{1:k-1}) d\mathbf{X}_{-\{j\}}^k. \end{aligned} \tag{15}$$

where $\mathbf{X}_{-\{j\}}^{0:k} = (\mathbf{X}_{-\{j\}}^0, ..., \mathbf{X}_{-\{j\}}^k)$ represents the sequence of states up to time $k$ of all the components of the partition except for the $j$-th.

MPFs use a different particle representation for each marginal posterior PDF. Therefore, we have $t$ different PFs with a set of $N$ weighted samples (particles) $\{(\mathbf{x}_{j,1}^{0:k}, \omega_{j,1}^k), ..., (\mathbf{x}_{j,N}^{0:k}, \omega_{j,N}^k)\}$, where $\mathbf{x}_{j,i}^{0:k}$ stands for the $i$-th particle of the $j$-th filter at time $k$, with associated weight $\omega_{j,i}^k$.

Then, the marginal posterior PDF of $\mathbf{x}_j^{0:k}$ is approximated as

$$p(\mathbf{x}_j^{0:k}|\mathbf{z}^{1:k}) \approx \sum_{i=1}^{N} \omega_{j,i}^k \delta(\mathbf{x}_j^{0:k} - \mathbf{x}_{j,i}^{0:k}). \tag{16}$$

Approximating (15) via direct Monte Carlo sampling is quite demanding computationally as we need to sample a large state space to compute the marginalization integral in (15). Therefore, MPFs usually make use of some approximations that enable the use of several parallel computationally efficient PFs which share information among themselves to approximate (15). We proceed to review two methods in the literature: the first order MPF and C-PF PROP.

### *2.1. MPF: A first order approximation*

In the MPF [19], at time $k-1$, the $l$-th PF approximates the predicted value of $\mathbf{x}_l^k$. There are different ways of obtaining these predictions. Here, we consider

$$\hat{\mathbf{x}}_l^k \approx \sum_{i=1}^{N} \omega_{l,i}^{k-1} \cdot \mathbf{x}_{l,i}^{k|k-1} \tag{17}$$

where $\mathbf{x}_{l,i}^{k|k-1}$ is obtained using the dynamic model to propagate particle $\mathbf{x}_{l,i}^{k-1}$ as

$$\mathbf{x}_{l,i}^{k|k-1} \sim p(\mathbf{x}_l^k|\mathbf{x}_{l,i}^{k-1}). \tag{18}$$

Thus, $\hat{\mathbf{x}}_l^k$ is the predicted state of the $l$-th component of the state at time $k$ given its particle posterior approximation at time $k-1$. Once these predictions have been computed, they are sent to the rest of the PFs. At time $k$, the $j$-th PF receives all these estimates of the states of the adjacent components and builds

$$\hat{\mathbf{X}}_{-\{j\}}^k = \left[(\hat{\mathbf{x}}_1^k)^T, ..., (\hat{\mathbf{x}}_{j-1}^k)^T, (\hat{\mathbf{x}}_{j+1}^k)^T, ..., (\hat{\mathbf{x}}_t^k)^T\right]^T. \tag{19}$$

The $j$-th PF then makes use of the following first order approximation

$$p(\mathbf{X}_{-\{j\}}^k|\mathbf{z}^{1:k-1}) \approx \delta\left(\mathbf{X}_{-\{j\}}^k - \hat{\mathbf{X}}_{-\{j\}}^k\right). \tag{20}$$

Using (20), the marginal posterior of $\mathbf{x}_j^{0:k}$, see (15), becomes

$$p(\mathbf{x}_j^{0:k}|\mathbf{z}^{1:k}) \propto p(\mathbf{z}^k|\mathbf{x}_j^k, \hat{\mathbf{X}}_{-\{j\}}^k) p(\mathbf{x}_j^k|\mathbf{x}_j^{k-1}) p(\mathbf{x}_j^{0:k-1}|\mathbf{z}^{1:k-1}). \tag{21}$$

It should be noted that (21) corresponds to one step of the Bayesian recursion with a modified likelihood function $p(\mathbf{z}^k|\mathbf{x}_j^k, \hat{\mathbf{X}}_{-\{j\}}^k)$. Then, particles at time $k$ are sequentially drawn from a proposal or importance function $q_j(\mathbf{x}_j^{0:k}|\mathbf{z}^{1:k})$ which is chosen to factorize as

$$q_j(\mathbf{x}_j^{0:k}|\mathbf{z}^{1:k}) = p(\mathbf{x}_j^k|\mathbf{x}_j^{k-1}) q_j(\mathbf{x}_j^{0:k-1}|\mathbf{z}^{1:k-1}). \tag{22}$$

Particles $\mathbf{x}_{j,i}^{0:k-1}$ in (16) were drawn at the previous time step from $q_j(\mathbf{x}_j^{0:k-1}|\mathbf{z}^{1:k-1})$, so that a new

particle, $\mathbf{x}_{j,i}^{0:k}$, is drawn by sampling from $p(\mathbf{x}_{j,i}^{k}|\mathbf{x}_{j,i}^{k-1})$ and appending it to the previous $i$-th particle [8, 19]. Particle weights are computed according to the principle of importance sampling as

$$\omega_{j,i}^{k} \propto \frac{p(\mathbf{x}_{j,i}^{0:k}|\mathbf{z}^{1:k})}{q_j(\mathbf{x}_{j,i}^{0:k}|\mathbf{z}^{1:k})} \tag{23}$$

$$= p(\mathbf{z}^k|\mathbf{x}_{j,i}^{k}, \hat{\mathbf{X}}_{-\{j\}}^{k})\omega_{j,i}^{k-1}. \tag{24}$$

This recursion is repeated for each component of the partition at each time step.

### *2.2. C-PF-PROP: A second order approximation*

The MPF can achieve a very good performance, but it deteriorates if $p(\mathbf{X}_{-\{j\}}^{k}|\mathbf{z}^{1:k-1})$ is not narrowly concentrated around $\hat{\mathbf{X}}_{-\{j\}}^{k}$, i.e. if the approximation in (20) is not accurate. Under a linear-Gaussian dynamic model, $p\left(\mathbf{x}_l^k|\mathbf{x}_l^{k-1}\right) = \mathcal{N}\left(\mathbf{x}_l^k; \mathbf{A}\mathbf{x}_l^{k-1}, \mathbf{Q}\right)$, where $\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{C})$ is a Gaussian PDF with mean $\hat{\mathbf{x}}$ and covariance matrix $\mathbf{C}$ evaluated at $\mathbf{x}$, the C-PF-PROP [22] addresses this limitation using a second order approximation in (20). That is, it makes the approximation

$$p(\mathbf{X}_{-\{j\}}^{k}|\mathbf{z}^{1:k-1}) \approx \prod_{l=1, l\neq j}^{t} \mathcal{N}\left(\mathbf{x}_l^k; \hat{\mathbf{x}}_l^k, \hat{\mathbf{C}}_l^k\right), \tag{25}$$

and

$$\hat{\mathbf{x}}_l^k = \mathbf{A}\bar{\mathbf{x}}_l^{k-1} \tag{26}$$

$$\hat{\mathbf{C}}_l^k = \mathbf{A}\bar{\mathbf{C}}_l^{k-1}\mathbf{A}^T + \mathbf{Q} \tag{27}$$

where

$$\bar{\mathbf{x}}_l^{k-1} = \sum_{i=1}^{N} \omega_{l,i}^{k-1}\mathbf{x}_{l,i}^{k-1} \tag{28}$$

$$\bar{\mathbf{C}}_l^{k-1} = \sum_{i=1}^{N} \omega_{l,i}^{k-1} \cdot (\mathbf{x}_{l,i}^{k-1} - \bar{\mathbf{x}}_l^{k-1}) \cdot (\mathbf{x}_{l,i}^{k-1} - \bar{\mathbf{x}}_l^{k-1})^T. \tag{29}$$

The $j$-th PF uses all the received information to build $\hat{\mathbf{X}}_{-\{j\}}^{k}$ and $\hat{\mathbf{C}}_{-\{j\}}^{k}$, which are respectively the collection of all the first and second predicted moments of the components of the marginal states, except for the $j$-th. Using (25) in (15), C-PF-PROP then approximates the marginal posterior PDFs by

$$p(\mathbf{x}_j^{0:k}|\mathbf{z}^{1:k}) \propto p(\mathbf{x}_j^{0:k}|\mathbf{z}^{1:k-1})p(\mathbf{z}^k|\mathbf{x}_j^k; \hat{\mathbf{X}}_{-\{j\}}^{k}, \hat{\mathbf{C}}_{-\{j\}}^{k}) \tag{30}$$

where we use the notation

$$p(\mathbf{z}^k|\mathbf{x}_j^k; \hat{\mathbf{X}}_{-\{j\}}^{k}, \hat{\mathbf{C}}_{-\{j\}}^{k}) \triangleq \int p(\mathbf{z}^k|\mathbf{X}^k) \prod_{l=1, l\neq j}^{t} \mathcal{N}\left(\mathbf{x}_l^k; \hat{\mathbf{x}}_l^k, \hat{\mathbf{C}}_l^k\right) d\mathbf{X}_{-\{j\}}^{k}. \tag{31}$$

As in the MPF, particles are drawn from the importance sampling function in (22), so that using (23), particle weights are computed as

$$\omega_{j,i}^{k} \propto p(\mathbf{z}^{k}|\mathbf{x}_{j,i}^{k}; \hat{\mathbf{X}}_{-\{j\}}^{k}, \hat{\mathbf{C}}_{-\{j\}}^{k})\omega_{j,i}^{k-1}. \tag{32}$$

Equation (31) does not generally have an analytical solution, but an approximation was provided in [22] for the received signal strength indicator (RSSI) measurement model.

## 3. Sigma-point multiple particle filters

In this section, we introduce two new MPFs with a second order approximation to the marginal PDFs of the components of the state in $\mathbf{X}_{-\{j\}}^{k}$ using sigma-point methods [1]. A brief review of sigma-point methods is first provided in Section 3.1. The two MPFs proposed in this work are presented in Sections 3.2 and 3.3, respectively.

### *3.1. Sigma-point integration methods*

Sigma-point methods [1, 32] are numerical integration tools to compute the moments of a distribution that undergoes a nonlinear transformation. Let us consider a random variable $\mathbf{x}$ with PDF $p(\mathbf{x})$ that is transformed through a function $\mathbf{z} = \mathbf{h}(\mathbf{x})$. Sigma-point methods use a set of $m$ sigma-points with their corresponding weights, $\mathcal{X} = \{(\mathcal{X}^1, w^1), ..., (\mathcal{X}^m, w^m)\}$, which match the mean $\bar{\mathbf{x}}$ and covariance matrix $\mathbf{P}$, i.e., the first two moments of $p(\mathbf{x})$, and approximate the first two moments of $\mathbf{z}$ as

$$\begin{aligned} \bar{\mathbf{z}} &= \int \mathbf{h}(\mathbf{x})\, p(\mathbf{x}) d\mathbf{x} \qquad (33) \\ &\approx \sum_{i=1}^{m} w^i \mathbf{h}(\mathcal{X}^i) \\ \mathbf{S} &= \int (\mathbf{h}(\mathbf{x}) - \bar{\mathbf{z}})(\mathbf{h}(\mathbf{x}) - \bar{\mathbf{z}})^T\, p(\mathbf{x}) d\mathbf{x} \qquad (34) \\ &\approx \sum_{i=1}^{m} w^i \left(\mathbf{h}(\mathcal{X}^i) - \bar{\mathbf{z}}\right)\left(\mathbf{h}(\mathcal{X}^i) - \bar{\mathbf{z}}\right)^T. \end{aligned}$$

There are many ways of selecting sigma-points and their corresponding weights, with different properties regarding computational burden and accuracy, for example, the unscented transform [32], the Gauss-Hermite quadrature rule [5], and the cubature rule [6] among others [1]. The unscented transform and cubature rule have the important property of requiring a number of sigma-points which grows linearly with the dimension of the state-space. On the contrary, the

number of sigma-points in Gauss-Hermite integration grows exponentially with the dimension of the state [1], often making its use in high-dimensional state spaces prohibitive.

### *3.2. SP-MPF: A sigma-point second order approximation*

The sigma-point MPF (SP-MPF), as C-PF-PROP, is a second order approximation to multiple particle filtering. However, unlike the latter, it does not require derivation of any analytical approximations for (31). Instead, it uses sigma-point integration, which has been shown to outperform analytical approximations in nonlinear Kalman filters [4].

First, it is useful to define

$$l_j(\mathbf{z}^k|\mathbf{x}_j^k) \triangleq \int p(\mathbf{z}^k|\mathbf{X}^k) \prod_{l=1,l\neq j}^{t} p(\mathbf{x}_l^k|\mathbf{z}^{1:k-1}) d\mathbf{X}_{-\{j\}}^k \tag{35}$$

which represents the likelihood for $\mathbf{x}_j^k$ after removing the influence on the measurements of the rest of the components of the state. Using (35), the marginal posterior for SP-MPF can be expressed as

$$p(\mathbf{x}_j^{0:k}|\mathbf{z}^{1:k}) \propto l_j(\mathbf{z}^k|\mathbf{x}_j^k) p(\mathbf{x}_j^k|\mathbf{x}_j^{k-1}) p(\mathbf{x}_j^{0:k-1}|\mathbf{z}^{1:k-1}). \tag{36}$$

The developed particle filters in this section require additive sensor models, which make the assumption

- A2: Equation (2) can be expressed as

  $$\mathbf{z}^k = \sum_{l=1}^{t} \mathbf{h}_l\left(\mathbf{x}_l^k\right) + \mathbf{v}^k \tag{37}$$

  where $\mathbf{h}_l(\mathbf{x}_l^k)$ is the contribution to the sensor measurements of the $l$-th component of the state and $\mathbf{v}^k$ is a zero-mean noise with known covariance matrix $\mathbf{R}^k$. Note that Equation (37) indicates that additivity of the measurements $\mathbf{h}_l(\mathbf{x}_l^k)$ is required.

In order to provide a second order approximation to the marginalization integrals, we consider the computation of $\bar{\mathbf{y}}_j^k\left(\mathbf{x}_j^k\right)$ and $\hat{\mathbf{S}}_j^k$, which respectively denote the mean and covariance of the conditional distribution $l_j(\mathbf{z}^k|\mathbf{x}_j^k)$. We show in Appendix 1 that these quantities can be computed as

$$\begin{aligned} \bar{\mathbf{y}}_j^k\left(\mathbf{x}_j^k\right) &= \int \mathbf{z}^k l_j(\mathbf{z}^k|\mathbf{x}_j^k) d\mathbf{z}^k \\ &= \mathbf{h}_j\left(\mathbf{x}_j^k\right) + \sum_{l=1,l\neq j}^{t} \bar{\mathbf{z}}_l^k \end{aligned} \tag{38}$$

where

$$\bar{\mathbf{z}}_l^k = \int \mathbf{h}_l(\mathbf{x}_l^k) p(\mathbf{x}_l^k|\mathbf{z}^{1:k-1}) d\mathbf{x}_l^k. \tag{39}$$

On the other hand, $\hat{\mathbf{S}}_j^k$ can be written as

$$\begin{aligned} \hat{\mathbf{S}}_j^k &= \int \left(\mathbf{z}^k - \bar{\mathbf{z}}^k\right)\left(\mathbf{z}^k - \bar{\mathbf{z}}^k\right)^T l_j(\mathbf{z}^k|\mathbf{x}_j^k) d\mathbf{z}^k \\ &= \mathbf{R}^k + \sum_{l=1, l\neq j}^{t} \mathbf{S}_l^k \end{aligned} \tag{40}$$

where

$$\mathbf{S}_l^k = \int \left[\left(\mathbf{h}_l(\mathbf{x}_l^k) - \bar{\mathbf{z}}_l^k\right)\left(\mathbf{h}_l(\mathbf{x}_l^k) - \bar{\mathbf{z}}_l^k\right)^T\right] p(\mathbf{x}_l^k|\mathbf{z}^{1:k-1}) d\mathbf{x}_l^k. \tag{41}$$

Note that the computation of $\bar{\mathbf{z}}_l^k$ and $\mathbf{S}_l^k$, respectively the contributions of $\mathbf{x}_l^k$ to the mean and covariance of $\mathbf{z}^k$ in (39) and (41), correspond to the type of integrals we can conveniently compute using sigma-points, as pointed out in Section 3.1.

In order to use sigma-point methods to evaluate integrals (39) and (41), we require the mean $\hat{\mathbf{x}}_l^k$ and covariance $\hat{\mathbf{C}}_l^k$ matrix of the prior distribution of each component of the partition. The presented method does not require the dynamic model to be linear-Gaussian, so that instead of using (26) and (27), we consider the expression in (17) to compute $\hat{\mathbf{x}}_l^k$, while $\hat{\mathbf{C}}_l^k$ is computed as

$$\hat{\mathbf{C}}_l^k = \sum_{i=1}^{N} \omega_{l,i}^{k-1} \cdot \left(\mathbf{x}_{l,i}^{k|k-1} - \hat{\mathbf{x}}_l^k\right) \cdot \left(\mathbf{x}_{l,i}^{k|k-1} - \hat{\mathbf{x}}_l^k\right)^T \tag{42}$$

where $\mathbf{x}_{l,i}^{k|k-1}$ is obtained using (18) and $\hat{\mathbf{x}}_l^k$ from (17).

Making use of sigma-point methods, we proceed to estimate $\bar{\mathbf{z}}_l^k$ and $\mathbf{S}_l^k$. For each component of the partition, we can use any sigma-point integration method [5, 6, 32] to select a set of $m_l$ weighted sigma-points $\mathcal{X}_l = \{(\mathcal{X}_l^1, w_l^1), ..., (\mathcal{X}_l^{m_l}, w_l^{m_l})\}$ that match $\hat{\mathbf{x}}_l^k$ and $\hat{\mathbf{C}}_l^k$, which are given by (17) and (42). Then, sigma-points are propagated through $\mathbf{h}_l(\cdot)$, so that $\bar{\mathbf{z}}_l^k$ and $\mathbf{S}_l^k$ can be approximated as

$$\bar{\mathbf{z}}_l^k \approx \sum_{i=1}^{m_l} w_l^i \mathbf{h}_l(\mathcal{X}_l^i) \tag{43}$$

$$\mathbf{S}_l^k \approx \sum_{i=1}^{m_l} w_l^i \left(\mathbf{h}_l(\mathcal{X}_l^i) - \bar{\mathbf{z}}_l^k\right) \cdot \left(\mathbf{h}_l(\mathcal{X}_l^i) - \bar{\mathbf{z}}_l^k\right)^T. \tag{44}$$

Thus, we can compute a second order Gaussian approximation to (35) using sigma-points as [1]

$$l_j(\mathbf{z}^k|\mathbf{x}_j^k) \approx l_j^N(\mathbf{z}^k|\mathbf{x}_j^k) \tag{45}$$

**Algorithm 1** PF for the $j$-th component in SP-MPF

- Initialize particles according to $p(\mathbf{x}_j^0)$

- Compute $\hat{\mathbf{x}}_j^1$ and $\hat{\mathbf{C}}_j^l$ using (17) and (42)

- Obtain a set of sigma-points and weights $\mathcal{X}_j$, whose first two moments match $\hat{\mathbf{x}}_j^1$ and $\hat{\mathbf{C}}_j^l$

- Compute $\bar{\mathbf{z}}_j^1$ and $\mathbf{S}_j^l$ using (43) and (44) using a sigma-point method

- Send $\bar{\mathbf{z}}_j^1$ and $\mathbf{S}_j^l$ to the other $t-1$ PFs.

**for** $k = 1, ..., k_{end}$ **do** ▷ $k_{end}$ is the final time step

- Collect new measurements $\mathbf{z}^k$ from sensors

**for** $l = 1, ..., t \quad l \neq j$ **do**

- Receive $\bar{\mathbf{z}}_l^k$ and $\mathbf{S}_l^k$ from the $l$-th PF.

**for** $i = 1, ..., N$ **do**

- Draw a sample $\mathbf{x}_{j,i}^k$ using $p(\mathbf{x}_j^k|\mathbf{x}_{j,i}^{k-1})$

- Compute $l_j^N(\mathbf{z}^k|\mathbf{x}_{j,i}^k)$ using (45)

- Compute $\omega_{j,i}^k = \omega_{j,i}^{k-1} l_j^N(\mathbf{z}^k|\mathbf{x}_{j,i}^k)$

**for** $i = 1, ..., N$ **do**

- Normalize $\omega_{j,i}^k = \frac{\omega_{j,i}^k}{\sum_{i=1}^N \omega_{j,i}^k}$

-Compute $\hat{N}_{eff}^j = \frac{1}{\sum_{i=1}^N (\omega_{j,i}^k)^2}$

**if** $\hat{N}_{eff}^j < \Gamma$ **then**

-Resample the particle set $\{(\mathbf{x}_{j,1}^k, \omega_{j,1}^k), ..., (\mathbf{x}_{j,N}^k, \omega_{j,N}^k)\}$)

- Compute $\hat{\mathbf{x}}_j^{k+1}$ and $\hat{\mathbf{C}}_j^{k+1}$ using (17) and (42)

- Obtain a set of sigma-points and weights $\mathcal{X}_j$, whose first two moments match $\hat{\mathbf{x}}_j^{k+1}$ and $\hat{\mathbf{C}}_j^{k+1}$ using a sigma-point method

- Compute $\bar{\mathbf{z}}_j^{k+1}$ and $\mathbf{S}_j^{k+1}$ using (43) and (44)

- Send $\bar{\mathbf{z}}_j^{k+1}$ and $\mathbf{S}_j^{k+1}$ to the other $t-1$ PFs.

$$= \mathcal{N}\Big(\mathbf{z}^k; \mathbf{h}_j(\mathbf{x}_j^k) + \sum_{l=1, l\neq j}^{t} \bar{\mathbf{z}}_l^k, \mathbf{R}^k + \sum_{l=1, l\neq j}^{t} \mathbf{S}_l^k\Big).$$

At time $k$, the $j$-th PF receives this estimation of $\bar{\mathbf{z}}_l^k$ and $\mathbf{S}_l^k$ from the other $t-1$ PFs so that it can approximate $l_j^{\mathcal{N}}(\mathbf{z}^k|\mathbf{x}_j^k)$. SP-MPF also draws particles from the importance density (22) so that the particle weight update equation in (23) becomes

$$\begin{aligned} \omega_{j,i}^k &\propto \frac{l_j^{\mathcal{N}}(\mathbf{z}^k|\mathbf{x}_j^k) p(\mathbf{x}_{j,i}^k|\mathbf{x}_{j,i}^{k-1}) p(\mathbf{x}_{j,i}^{0:k-1}|\mathbf{z}^{1:k-1})}{p(\mathbf{x}_{j,i}^k|\mathbf{x}_{j,i}^{k-1}) q(\mathbf{x}_{j,i}^{0:k-1}|\mathbf{z}^{1:k-1})} \\ &= \omega_{j,i}^{k-1} l_j^{\mathcal{N}}(\mathbf{z}^k|\mathbf{x}_j^k). \end{aligned} \tag{46}$$

The $j$-th PF finally computes $\hat{\mathbf{x}}_j^{k+1}$ and $\hat{\mathbf{C}}_j^{k+1}$, the first two moments of the predicted PDF of the state at time $k+1$, using (17) and (42). Then, it draws its own set of sigma-points $\mathcal{X}_j$ that match $\hat{\mathbf{x}}_j^{k+1}$ and $\hat{\mathbf{C}}_j^{k+1}$ and use them to approximate $\bar{\mathbf{z}}_j^{k+1}$ and $\mathbf{S}_j^{k+1}$ using (43) and (44) to transmit them to the other $t-1$ PFs, which would use them in their own recursion at time $k+1$.

As a result of the way particles are drawn from $q_j(\mathbf{x}_j^{0:k}|\mathbf{z}^{1:k})$, approximating $p(\mathbf{x}_j^k|\mathbf{z}^{1:k})$ only requires to store the states of the particles up to time $k-1$, so that implementations can be memory-efficient. However, the underlying state-space which the filter is dealing with is in fact that of $\mathbf{x}_j^{0:k}$. The dimension thus grows at every time step, which provokes the so called particle degeneracy [33].

To prevent particle degeneracy, an adaptive resampling scheme based on the monitoring of the effective sample size $N_j^{eff}$ is considered [10, 34]. One cannot generally exactly evaluate $N_j^{eff}$, however a commonly accepted estimate is given by [8]

$$\hat{N}_j^{eff} = \frac{1}{\sum_{i=1}^{N} (\omega_{j,i}^k)^2}, \tag{47}$$

which takes values in the interval $[1, N]$. If $\hat{N}_j^{eff}$ falls below a given threshold, $\Gamma$, particles are resampled. This restores the effective sampling size to $N$, preventing particle degeneracy, but results in a loss of diversity among particles. The pseudocode for SP-MPF is provided in Algorithm 1

### *3.3. SP-MAPF: Introducing auxiliary sampling in the SP-MPF*

In this section, we propose a version of SP-MPF that uses auxiliary sampling [30]: SP-MAPF. The use of an auxiliary variable allows for the consideration of the actual measurements in the drawing of samples from the marginal posterior PDF, mimicking the way samples would be

Table 1: Principal features of the different multiple particle filtering algorithms

| **Filter** | **MPF** | **C-PF-PROP** | **SP-MPF** | **SP-MAPF** |
|---|---|---|---|---|
| Approximation to Equation (11) | Dirac delta approximation | Analytical approximation | sigma-point methods | sigma-point methods |
| Order of the approximation | mean | mean and covariance | mean and covariance | mean and covariance |
| Auxiliary filtering | × | × | × | ✓ |
| Does not require analytical derivations | ✓ | × | ✓ | ✓ |

drawn from the optimal importance density [8]. This technique has been shown to improve particle filters performance under informative sensor measurements.

Contrary to the previously discussed methods, instead of considering the posterior PDF of the sequence of states as in (15), SP-MAPF obtains samples from $p(\mathbf{x}_j^k|\mathbf{z}^{1:k})$, the marginal posterior PDF of the state at time $k+1$ in (11). Given a particle representation of the marginal posterior PDF at time $k-1$, the $j$-th PF can approximate $p(\mathbf{x}_j^k|\mathbf{z}^{1:k-1})$ in (11) using the Chapman-Kolmogorov equation as

$$\begin{aligned} p(\mathbf{x}_j^k|\mathbf{z}^{1:k-1}) &= \int p(\mathbf{x}_j^k|\mathbf{x}_j^{k-1}) p(\mathbf{x}_j^{k-1}|\mathbf{z}^{1:k-1}) d\mathbf{x}_j^{k-1} \\ &\approx \int p(\mathbf{x}_j^k|\mathbf{x}_j^{k-1}) \sum_{i=1}^{N} \omega_{j,i}^{k-1} \delta(\mathbf{x}_j^{k-1} - \mathbf{x}_{j,i}^{k-1}) d\mathbf{x}_j^{k-1} \\ &= \sum_{i=1}^{N} \omega_{j,i}^{k-1} p(\mathbf{x}_j^k|\mathbf{x}_{j,i}^{k-1}). \end{aligned} \tag{48}$$

As in SP-MPF, at time $k$ the $j$-th PF of SP-MAPF collects the shared estimations $\bar{\mathbf{z}}_l^k$ and $\mathbf{S}_l^k$ from the rest of the components of the partition so that it can also approximate $l_j^{\mathcal{N}}(\mathbf{z}^k|\mathbf{x}_j^k)$ in (45). Using (48), (11) can therefore be written as

$$p(\mathbf{x}_j^k|\mathbf{z}^{1:k}) \propto l_j^{\mathcal{N}}(\mathbf{z}^k|\mathbf{x}_j^k) \sum_{i=1}^{N} \omega_{j,i}^{k-1} p(\mathbf{x}_j^k|\mathbf{x}_{j,i}^{k-1}). \tag{49}$$

SP-MAPF indirectly obtains samples from this PDF using an auxiliary variable, $a_j$, for each component of the state [30], which removes the sum in (49). Thus, SP-MAPF also avoids the increase in computational cost arising from the Chapman-Kolmogorov step integral in (48). There-

fore SP-MAPF samples in a higher dimension from

$$p(\mathbf{x}_j^k, a_j|\mathbf{z}^{1:k}) \propto l_j^{\mathcal{N}}(\mathbf{z}^k|\mathbf{x}_j^k)\omega_{j,a_j}^{k-1} p(\mathbf{x}_j^k|\mathbf{x}_{j,a_j}^{k-1}) \tag{50}$$

where $a_j \in \{1, 2, ..., N\}$ is an index in the mixture in (49).The selection of the auxiliary variable is sound as $p(\mathbf{x}_j^k, a_j|\mathbf{z}^{1:k}) \geq 0$ and integrating (50) over $a_j$, one gets the marginal PDF in (49). SP-MAPF produces samples from an importance density in a higher dimension

$$q_j(\mathbf{x}_j^k, a_j|\mathbf{z}^{1:k}) \propto l_j^{\mathcal{N}}(\mathbf{z}^k|\boldsymbol{\mu}_{j,a_j}^k)\omega_{j,a_j}^{k-1} p(\mathbf{x}_j^k|\mathbf{x}_{j,a_j}^{k-1}) \tag{51}$$

with $\boldsymbol{\mu}_{j,a_j}^k$ being some characterization of $\mathbf{x}_j^k$ given $\mathbf{x}_{j,a_j}^{k-1}$, such as the predicted mean, $\boldsymbol{\mu}_{j,a_j}^k = \mathrm{E}[\mathbf{x}_j^k|\mathbf{x}_{j,a_j}^{k-1}]$, or a sample, $\boldsymbol{\mu}_{j,a_j}^k \sim p(\mathbf{x}_j^k|\mathbf{x}_{j,a_j}^{k-1})$. Samples from $a_j$ are drawn according to the distribution of first-stage weights, $\lambda_{j,i}$, which are given by

$$\lambda_{j,i} \propto \omega_i^{k-1} l_j^{\mathcal{N}}(\mathbf{z}^k|\boldsymbol{\mu}_{j,a_j}^k). \tag{52}$$

Using (50) and (51), the weight update equation in (23) becomes

$$\begin{aligned} \omega_{j,i}^k &\propto \frac{p(\mathbf{x}_{j,i}^k, a_j^i|\mathbf{z}^{1:k})}{q_j(\mathbf{x}_{j,i}^k, a_j^i|\mathbf{z}^{1:k})} \\ &= \frac{l_j^{\mathcal{N}}(\mathbf{z}^k|\mathbf{x}_{j,a_j^i}^k)}{l_j^{\mathcal{N}}(\mathbf{z}^k|\boldsymbol{\mu}_{j,a_j^i}^k)}. \end{aligned} \tag{53}$$

As in SP-MPF, the $j$-th particle filter finally estimates the first two moments of the predicted PDF at the next time step using (17) and (42), and draws a set of sigma-points $\mathcal{X}_j$ to approximate $\bar{\mathbf{z}}_j^{k+1}$ and $\mathbf{S}_j^{k+1}$ using (43) and (44), which are then sent to the other $t-1$ PFs. It should be noted that, if $t = 1$ and the measurement noise is Gaussian, SP-MPF and SP-MAPF corresponds to sequential importance sampling and the auxiliary particle filter, respectively.

The pseudocode for SP-MAPF is given in Algorithm 2. The main features and differences among the different multiple particle filtering algorithms are summarized in Table 1.

### *3.4. On computational complexity*

In this section, we briefly analyze the computational complexity of the proposed methods in comparison to that of MPF (see Section 2.1), as this algorithm is the most basic multiple particle filter. The key aspect is that the computational complexity of the presented methods is $O(N \cdot t)$, i.e., linear with respect to the number of particles, which is a desirable property for a particle filter algorithm, though not necessary [26, 35], and also linear with respect to $t$, the number of components of the partition in (5).

The main increase in computational complexity of SP-MPF with respect to MPF lies in the following required additional steps:

1. The computation of the covariance matrix $\hat{\mathbf{C}}_l^k$, which is $O\left(N \cdot n_{x_l}^2\right)$.
2. The drawing of sigma-points, which considering the use of a Cholesky factorization of $\hat{\mathbf{C}}_l^k$, is $O\left(n_{x_l}^3\right)$ [32].
3. The transformation of sigma-points, whose computational complexity depends on the measurement model, and computation of the mean and covariance of the transformed sigma-points, which are $O\left(m \cdot n_z\right)$ and $O\left(m \cdot n_z^2\right)$, respectively.

The important result of this analysis is that, typically, we have that $N >> m$, $N >> n_{x_l}$ and $N >> n_z$, so that the computational complexity of each individual filter is mainly dominated by the number of particles. As the computational complexities of SP-MPF and MPF are both linear with respect to the number of particles, this means that the increase in computational complexity of SP-MPF with respect to MPF is usually not high as will be shown in Section 4. In addition, if we aim to achieve a required performance goal in our application, which is the case in many engineering problems, SP-MPF allows for the decrease of the number of particles $N$ with respect to MPF. Therefore, the required running time to achieve our performance goal can be effectively lowered with respect to MPF. A discussion considering this aspect based on our simulations can be found in Section 4.

In the case of SP-MAPF the evaluation of the first stage weights is also required, a procedure whose computational complexity is also $O(N)$, but which also depends on the specifics of the measurement model. It is important to note that the computational complexity of SP-MAPF will therefore also remain linear with respect to the number of particles.

## 4. Simulations

In this section, the performance of the proposed methods is analyzed via simulations of a multiple target tracking scenario with fixed and known number of targets. This problem adequately fits in the multiple filtering approach, as the marginal posterior PDF of the state of each target can be individually approximated by a different filter. The developed methods could be used to deal with the surviving targets in multiple target tracking problems with an unknown and varying number of targets [15].

**Algorithm 2** PF for the $j$-th component in SP-MAPF

- Initialize particles according to $p(\mathbf{x}_j^0)$
- Compute $\hat{\mathbf{x}}_j^1$ and $\hat{\mathbf{C}}_j^1$ using (17) and (42)
- Obtain a set of sigma-points and weights $\mathcal{X}_j$, whose first two moments match $\hat{\mathbf{x}}_j^1$ and $\hat{\mathbf{C}}_j^1$ using a sigma-point method
- Compute $\bar{\mathbf{z}}_j^1$ and $\mathbf{S}_j^1$ using (43) and (44)
- Send $\bar{\mathbf{z}}_j^1$ and $\mathbf{S}_j^1$ to the other $t-1$ PFs.

**for** $k = 1, ..., k_{end}$ **do** ▷ $k_{end}$ is the final time step

- Collect new measurements $\mathbf{z}^k$ from sensors

**for** $l = 1, ..., t \quad l \neq j$ **do**

- Receive $\bar{\mathbf{z}}_l^k$ and $\mathbf{S}_l^k$ from the $l$-th PF.

**for** $i = 1, ..., N$ **do**

- Draw $\boldsymbol{\mu}_{j,i}^k$ using: $p(\mathbf{x}_j^k|\mathbf{x}_{j,i}^{k-1})$ or $\mathrm{E}[\mathbf{x}_j^k|\mathbf{x}_{j,i}^{k-1}]$
- Compute $l_j^N(\mathbf{z}^k|\boldsymbol{\mu}_{j,i}^k)$ using (45)
- Compute $\lambda_{j,i}^k = \omega_{j,i}^{k-1} l_j^N(\mathbf{z}^k|\boldsymbol{\mu}_{j,i}^k)$

**for** $i = 1, ..., N$ **do**

- Normalize $\lambda_{j,i}^k = \frac{\lambda_{j,i}^k}{\sum_{i=1}^N \lambda_{j,i}^k}$

**for** $i = 1, ..., N$ **do**

- Sample an index $a_j^i$ from the distribution $(\lambda_{j,1}, \lambda_{j,2}, ..., \lambda_{j,N})$
- Draw a sample $\mathbf{x}_{j,i}^k$ using $p(\mathbf{x}_j^k|\mathbf{x}_{j,a_j^i}^{k-1})$
- Compute $l_j^N(\mathbf{z}^k|\mathbf{x}_{j,a_j^i}^k)$ using (45)
- Compute $\omega_{j,i}^k = \frac{l_j^N(\mathbf{z}^k|\mathbf{x}_{j,a_j^i}^k)}{l_j^N(\mathbf{z}^k|\mu_{j,a_j^i}^k)}$

**for** $i = 1, ..., N$ **do**

- Normalize $\omega_{j,i}^k = \frac{\omega_{j,i}^k}{\sum_{i=1}^N \omega_{j,i}^k}$

- Compute $\hat{\mathbf{x}}_j^{k+1}$ and $\hat{\mathbf{C}}_j^{k+1}$ using (17) and (42)
- Obtain a set of sigma-points and weights $\mathcal{X}_j$, whose first two moments match $\hat{\mathbf{x}}_j^{k+1}$ and $\hat{\mathbf{C}}_j^{k+1}$ using a sigma-point method.
- Compute $\bar{\mathbf{z}}_j^{k+1}$ and $\mathbf{S}_j^{k+1}$ using (43) and (44)
- Send $\bar{\mathbf{z}}_j^{k+1}$ and $\mathbf{S}_j^{k+1}$ to the other $t-1$ PFs.

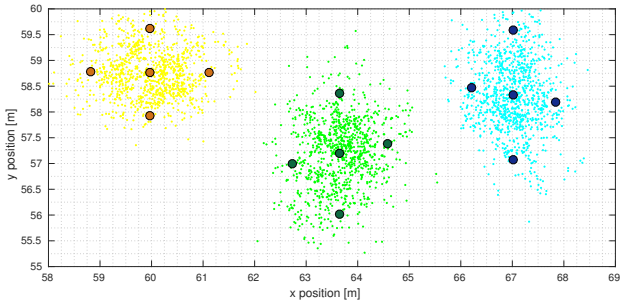



Figure 1: SP-MPF particles (thin dots) and sigma-points (thick dots) in a scenario with 3 different targets at a given time step.

Before discussing performance considerations, we think it is convenient to illustrate how SP-MPF and SP-MAPF work. To this end, Figure 1 shows the inner information managed by the different PFs of the proposed methods in a scenario with 3 targets. As previously explained in Section 3, the marginal PDF of each target is approximated by a PF using a weighted set of particles. In addition, every PF also computes a set of sigma-points (in this case drawn using the unscented transform) that matches the first two moments of its predicted marginal posterior PDF. Figure 1 shows in a different color for each target (or each PF) the particles and sigma-points which represent their marginal PDFs at a given time step.

The considered dynamic and measurement models are introduced in Section 4.1, while the performance in the presented scenario for different PF methods is discussed in Section 4.2.

### *4.1. Dynamic and measurement models*

There is a total of $t$ targets in the scenario, where the state of the $j$-th target at time $k$ is represented by its position and velocity through its state vector, $\mathbf{x}_j^k = [x_j^k, \dot{x}_j^k, y_j^k, \dot{y}_j^k]^T$, with $[x_j^k, y_j^k]^T$ being the position vector and $[\dot{x}_j^k, \dot{y}_j^k]^T$ the velocity vector of target $j$ at time $k$. The motion of each target has been modeled as linear with a nearly constant velocity [2]

$$p(\mathbf{x}_j^{k+1}|\mathbf{x}_j^k) = \mathcal{N}(\mathbf{x}_j^{k+1}; \mathbf{F}\mathbf{x}_j^k, \mathbf{Q}) \tag{54}$$

$$\mathbf{F} = \mathbf{I}_2 \otimes \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix} \qquad \mathbf{Q} = \sigma_u^2 \mathbf{I}_2 \otimes \begin{pmatrix} \tau^3/3 & \tau^2/2 \\ \tau^2/2 & \tau \end{pmatrix}$$

where $\tau$ is the sampling period, $\mathbf{I}_n$ is the $n \times n$ identity matrix, $\otimes$ stands for the Kronecker product and $\sigma_u^2$ is the continuous-time process noise intensity. The PDF at time $k = 0$ of the position the $j$-th target $p(\mathbf{x}_j^0)$ is initialized as Gaussian distribution centered at the true position of the target,

with covariance $\mathbf{Q}$. Measurements at time $k$, $\mathbf{z}^k = [z_1^k, ..., z_{N_s}^k]^T$, are received through $N_s = 169$ sensors in a sensor network displayed in a 2-D regular grid covering 120m×120m, with a spacing of 10m between sensors in both axes, see Figure 2.

Simulations include up to $t = 8$ targets in the scenario whose trajectories cross each other, recreating a demanding multiple target tracking problem. These trajectories were generated according to Equation (54), with $\tau = 1s$ and $\sigma_u = 0.1m/s^{3/2}$. The simulated target trajectories are shown in Figure 2.

We aim to compare the performance of the methods proposed in this paper with the C-PF-PROP method. Therefore, we use the received signal strength indicator measurement model for which the C-PF-PROP analytical expressions are available in [22, 27]. The nonlinear measurement equation (37) for the $i$-th sensor at time $k$ then is

$$z_i^k = h_i(\mathbf{X}^k) + v_i^k \tag{55}$$

$$h_i(\mathbf{X}^k) = \sum_{j=1}^{t} h_{i,j}(\mathbf{x}_j^k) \qquad h_{i,j}(\mathbf{x}_j^k) = \frac{\Phi}{\left(d_{j,i}^k\right)^\alpha + \epsilon}$$

where $v_i^k$ is a zero-mean, unit-variance ($\sigma^2 = 1$), Gaussian-distributed noise, $\Phi$ determines the signal power received by a sensor, $\epsilon$ is a saturation parameter which limits the signal power that can be received from near targets, $\alpha$ is a path-loss coefficient and $d_{j,i}^k$ is the distance from the $j$-th target to the $i$-th sensor. The sensor model parameters have been set to $\Phi = 500$, $\epsilon = 25$ and $\alpha = 2$.

Under the above dynamic and measurement models, nonlinearities arise from the relationship between target positions and measurements in (55), while the relationship between the velocities and the positions of the targets is linear and Gaussian. This allows for the use of Rao-Blackwellization, which improves the performance of the PFs for a given sample size [12, 15]. Therefore, particles in our simulations only contain the position elements of the targets, while velocities are estimated using a Kalman filter for each particle, conditional on its positions up to the current time step.

### *4.2. Simulation results*

In this section, we compare the proposed methods with the following particle filters for high-dimensional spaces: the parallel partition (PP), presented in [15, Sec. III], is a PF which estimates the joint PDF of all targets by independently sampling each target state, the auxiliary PP

(APP) [17, 18] is a version of the PP method which uses an auxiliary variable for each target to improve its importance sampling procedure. These two filters, PP and APP, estimate the joint state space PDF but independently sample the state of each target. The rest of the considered methods fall into the group of the multiple particle filtering approach: the MPF [19, 20], which has been reviewed in Section 2.1, the MAPF [23, 36] which makes use of auxiliary filtering to improve MPF, and C-PF-PROP [22], whose details can also be found in Section 2.2.

We have implemented SP-MPF and SP-MAPF using the unscented transformation [4, 32], the cubature rule [6] and the Gauss-Hermite quadrature rule [5] (of order $n = 3$). The weight of the central sigma-point in the unscented transform has been set to $w_o = 1/3$. It should also be noted that the sigma-points are only drawn in the position elements, due to the Rao-Blackwellization. The obtained results considering the cubature rule and the Gauss-Hermite quadrature rule remain basically unaltered with respect to the use of the unscented transform, so that, for the sake of clarity, their results have not been included in the figures.

The performance of the different methods is characterized by their averaged optimal subpattern assignment (OSPA) [37] position error (with parameters $p = 2$, $c = 10$). The average is computed with respect to the 100 time steps of the trajectories (see Figure 2) of all targets in the scenario, in a Monte Carlo experiment with 5000 runs. Before analyzing the performance of the filters, we first seek to characterize the accuracy of the Gaussian approximation of the marginalization integral in Equation (45). In order to do so, we have performed simulations considering the method in [26, Equation (49)], which computes a particle approximation of the marginalization integrals in Equation (11) (i.e. without a Gaussian approximation). Results show that in a scenario with 2 targets, and using 100 particles, the averaged OSPA position error for SP-MPF and the method in [26] are 1.41.m and 1.39m, respectively. For 3 targets, averaged OSPA errors are respectively 1.37m and 1.38m. Note that the difference in performance between SP-MPF and the filter in [26] is almost negligible, indicating that the Gaussian approximation of SP-MPF is as accurate as the particle integration method in [26]. It should be noted, that the computational complexity of SP-MPF is $O(t \cdot N)$, while the method in [26] is $O(N^t)$. Thus, the use of this direct marginalization method is unaffordable for a high number of components in the partition in (5).

Figure 3 shows the averaged OSPA position error for the different algorithms with respect to the number of particles when there are 3 and 6 targets in the scenario. It is worth noting that SP-MPF and SP-MAPF both outperform C-PF-PROP, a feature which speaks particularly well of the

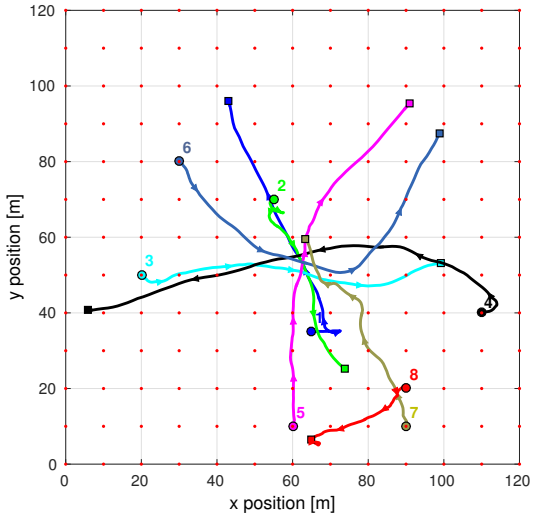
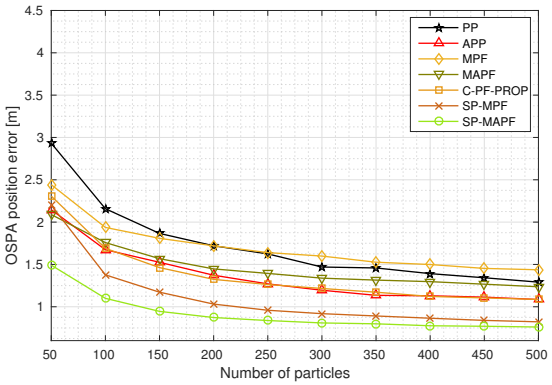


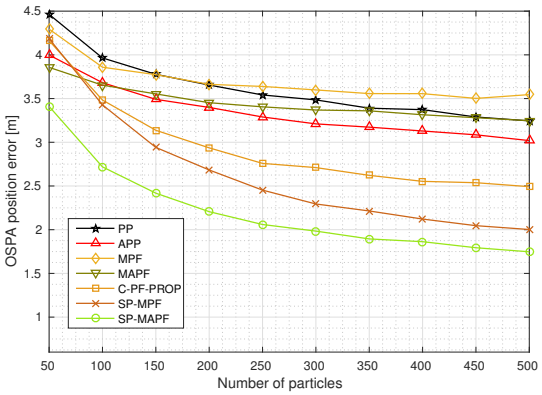

Figure 2: Eight simulated trajectories, consisting of 100 steps. The initial point of each trajectory is indicated by a circle, while the final point is represented by a square. An arrow shows the position of the target at steps 20, 40, 60 and 80 of each trajectory. When simulating a scenario with $t$ targets, target trajectories 1 to $t$ are taken into account. Sensors are represented as red dots.

algorithms presented in this paper, as the design of C-PF-PROP specifically fits the measurement model in (55). It can also be observed that SP-MAPF is the best-performing filter as it gathers all the advantages from the techniques considered in this paper, dealing with the dimensionality effects by multiple particle filtering and Rao-Blackwellization, relying on sigma-point methods to achieve a higher order approach to the marginalization integrals, and using improved importance densities by means of auxiliary filtering. The performance of the filters that estimate the joint state (PP and APP) should not to be overlooked, as they also competently deal with the proposed multiple target tracking problem, especially, for a high number of particles. In addition, some applications can benefit from the availability of an approximation to the joint posterior PDF [38]. The averaged OSPA error against time for the different algorithms when there are 3 targets in the scenario is provided in Figure 4. Note that, as expected, the error generally increases for all algorithms at those time instants when the targets trajectories cross each other, which happens at around time step $k = 50$, see Figure 2. It is in fact in these difficult situations when the biggest difference in performance is observed when comparing SP-MPF or SP-MAPF with the rest of the filters.

In order to further analyze the performance of the filters, the efficiency of the importance




(a)




(b)

Figure 3: Averaged OSPA position error with respect to the number of particles of the different algorithms in a scenario with 3 targets (a) and with 6 targets (b).

sampling schemes of the different multiple particle filters is also considered by means of their effective sample size [10, 34]. Figure 5 shows the mean target effective sample size at each time step, with three targets under track, for the different multiple particle filter algorithms when they use 300 particles. To have a fair comparison in the effective sampling size analysis, MPF, C-PF-PROP and SP-MPF perform a resampling stage after every time step, with the effective sample size being computed prior to the resampling stage. It can be observed that SP-MAPF has the highest effective sample size at all times. Note that at those time instants where the involved PDFs are harder to approximate (i.e. when targets get in close proximity, from time $k = 50$), the difference in the effective sample size from the second-order approximations of SP-MAPF and

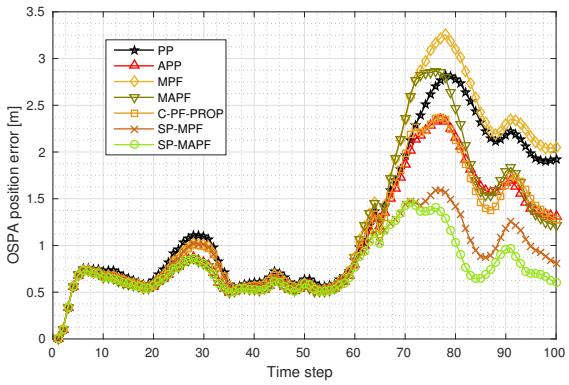



Figure 4: Averaged OSPA position error at each time step of the different algorithms with $N = 300$ particles in a scenario with 3 targets.

SP-MPF respectively broadens with respect to the first order approximations of MAPF and MPF, indicating that the methods presented in this paper achieve a better approximation to the posterior PDF at these time instants. It is also worth noting that SP-MPF also presents a higher effective sample size than C-PF-PROP in these situations, which is in accordance with the tracking error results presented before.

As indicated in Section 3.4, it is important to take into account that these algorithms do not have the same computational resources requirements for a given number of particles. Figure 6 plots the mean execution time of our MATLAB implementation of these methods with an Intel Core 7 at 2.5GHz against the number of particles. Using Figure 6 along with the results of Figure 3a allows us to note that, as the computational load for SP-MPF and C-PF-PROP is almost the same, for a given computational load, SP-MPF achieves a lower error, making its use clearly advisable.

It is also worth highlighting that those algorithms using auxiliary filtering require approximately a 60% additional time compared to their counterparts that do not implement this feature, which is in accordance with the computational complexity insights drawn in Section 3.4. Although the use of auxiliary filtering generally pays off in this scenario, its incidence in computational complexity should be carefully considered. For example, if computational load is a critical factor, choosing SP-MAPF over SP-MPF with 3 targets in the scenario would not further diminish the error, as for the same computational load. For example, the error for SP-MAPF with 200 particles and SP-MPF with 400 particles in Figure 3a is approximately the same. Nevertheless,

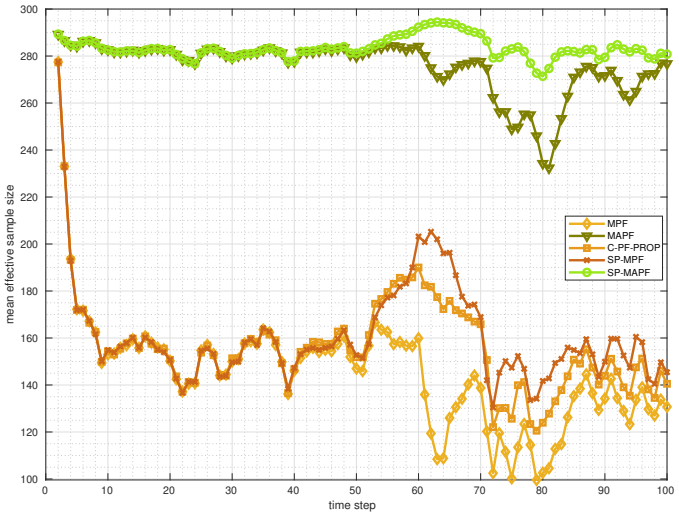



Figure 5: Mean effective sample size for the different multiple particle filters considered in the paper in a scenario with 3 targets and 300 particles. As there is a particle filter for each target, we show the averaged effective sample size for all targets. The average sample size is higher for the filters with auxiliary sampling. Sigma-point integration improves the results when targets get in close proximity (from time step 50).

choosing any of SP-MPF with 400 particles or SP-MAPF with 200 particles would clearly be advisable when compared to the rest of the considered algorithms with a similar computational load: APP with 300 particles, MAPF with 200 particles, C-PF-PROP with 400 particles, PP with 400 particles, or MPF with 500 particles. To illustrate how the different methods react to an increase in the dimensionality of the state-space, Figure 7 shows their averaged OSPA position error using a fixed number of particles with respect to the number of targets in the scenario. It is shown that those filters using higher order approximations to the marginal posterior PDFs outperform the first order approaches.

It is worth noting that although the error apparently increases for all the algorithms when a higher number of targets is considered, this trend is not applicable when the 8th target is incorporated in the scenario, as the averaged error in this case descends. Note that the trajectory of this target in Figure 2 does not get close to targets 1 to 6. This behavior, shown in Figure 2, indicates that the increasing error in Figure 7 is (to a great extent) caused by the difficult situation arising from the repeatedly crossing of targets 1 to 7 in the center of the scenario. Once again, this shows the remarkable ability of SP-MPF and SP-MAPF to deal with these difficult scenarios.

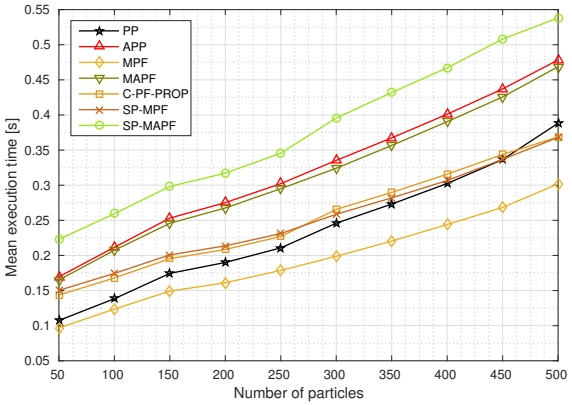



Figure 6: Mean execution time with respect to the number of particles for the different algorithms when $t = 3$ targets are present at the scenario. Execution times consider the estimation of the 100 steps of the trajectories.

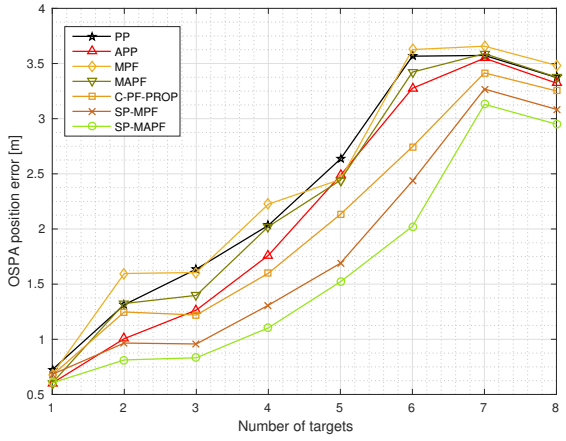



Figure 7: Averaged OSPA position error of the different algorithms with 250 particles in scenarios with 1 to 8 targets.

## 5. Conclusions

In this paper, we have introduced two new algorithms, SP-MPF and SP-MAPF, which fit in the multiple particle filtering paradigm providing a second order approximation to the involved marginal PDFs. The proposed algorithms are valid for additive measurement models, do not require the calculation of Jacobians and make use of sigma-point integration methods for accurate marginalization. This enables them to outperform MPF and C-PF-PROP due to a higher accuracy in the marginalization step. Compared to C-PF-PROP, the presented methods have the advantage of being directly applicable to a wider range of models. The performance of the presented methods has been evaluated in a high-dimensional multiple target tracking problem, demonstrating that SP-MPF and SP-MAPF widely outperform previously presented algorithms in the literature.

## Appendix

This appendix shows a derivation of the first and second moments of the marginalization integrals in (38) and (40), which are of interest in multiple filtering. The first moment can be computed as

$$\begin{aligned}
\bar{\mathbf{y}}_j^k\left(\mathbf{x}_j^k\right) &= \int \mathbf{z}^k l_j(\mathbf{z}^k|\mathbf{x}_j^k)d\mathbf{z}^k \qquad (56)\\
&= \int\left[\int \mathbf{z}^k p(\mathbf{z}^k|\mathbf{X}^k)d\mathbf{z}^k\right]\prod_{l=1,l\neq j}^{t} p(\mathbf{x}_l^k|\mathbf{z}^{1:k-1})d\mathbf{X}_{-\{j\}}^k\\
&= \int\sum_{l=1}^{t}\mathbf{h}_l(\mathbf{x}_l^k)\prod_{l=1,l\neq j}^{t} p(\mathbf{x}_l^k|\mathbf{z}^{1:k-1})d\mathbf{X}_{-\{j\}}^k\\
&= \mathbf{h}_j(\mathbf{x}_j^k)+\sum_{l=1,l\neq j}^{t}\int \mathbf{h}_l(\mathbf{x}_l^k)\cdot\prod_{l=1,l\neq j}^{t} p(\mathbf{x}_l^k|\mathbf{z}^{1:k-1})d\mathbf{X}_{-\{j\}}^k\\
&= \mathbf{h}_j(\mathbf{x}_j^k)+\sum_{l=1,l\neq j}^{t}\bar{\mathbf{z}}_l^k
\end{aligned}$$

where $\bar{\mathbf{z}}_l^k$ is given by (39). The second moment can be computed as

$$\begin{aligned}
\hat{\mathbf{S}}_j^k &= \int\left(\mathbf{z}^k-\bar{\mathbf{z}}^k\right)\left(\mathbf{z}^k-\bar{\mathbf{z}}^k\right)^T l_j(\mathbf{z}^k|\mathbf{x}_j^k)d\mathbf{z}^k \qquad (57)\\
&= \int\left[\int\left(\mathbf{z}^k-\bar{\mathbf{z}}^k\right)\left(\mathbf{z}^k-\bar{\mathbf{z}}^k\right)^T p(\mathbf{z}^k|\mathbf{X}^k)d\mathbf{z}^k\right]\cdot\prod_{l=1,l\neq j}^{t} p(\mathbf{x}_l^k|\mathbf{z}^{1:k-1})d\mathbf{X}_{-\{j\}}^k\\
&= \mathbf{R}^k+\int\left[\left(\sum_{l=1,l\neq j}^{t}\left(\mathbf{h}_l(\mathbf{x}_l^k)-\bar{\mathbf{z}}_l^k\right)\right)\left(\sum_{l=1,l\neq j}^{t}\left(\mathbf{h}_l(\mathbf{x}_l^k)-\bar{\mathbf{z}}_l^k\right)\right)^T\right]\cdot\prod_{l=1,l\neq j}^{t} p(\mathbf{x}_l^k|\mathbf{z}^{1:k-1})d\mathbf{X}_{-\{j\}}^k
\end{aligned}$$

$$= \mathbf{R}^k + \sum_{l=1,l\neq j}^{t} \int \left[ \left( \mathbf{h}_l(\mathbf{x}_l^k) - \bar{\mathbf{z}}_l^k \right) \left( \sum_{l=1,l\neq j}^{t} \left( \mathbf{h}_l(\mathbf{x}_l^k) - \bar{\mathbf{z}}_l^k \right) \right)^T \right] \cdot \prod_{l=1,l\neq j}^{t} p(\mathbf{x}_l^k | \mathbf{z}^{1:k-1}) d\mathbf{X}_{-\{j\}}^k$$

$$= \mathbf{R}^k + \sum_{l=1,l\neq j}^{t} \mathbf{S}_l^k$$

where $\mathbf{S}_l^k$ is given by (41).


[1] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[2] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. John Wiley & Sons, Inc., 2001.

[3] F. Daum, "Exact finite-dimensional nonlinear filters," *IEEE Transactions on Automatic Control*, vol. 31, pp. 616–622, jul 1986.

[4] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," in *Proceedings of the IEEE*, vol. 92, pp. 401–422, 2004.

[5] 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.

[6] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, pp. 1254–1269, jun 2009.

[7] M. R. Morelande and Á. F. García-Fernández, "Analysis of Kalman filter approximations for nonlinear measurements," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5477–5484, 2013.

[8] M. 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.

[9] B. Ristic and M. S. Arulampalam, "Tracking a manoeuvring target using angle-only measurements: Algorithms and performance," *Signal Processing*, vol. 83, no. 6, pp. 1223–1238, 2003.

[10] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter, particle filters for tracking applications*. Artech House, 2004.

[11] F. Daum and J. Huang, "Curse of dimensionality and particle filters," *Proceedings of 2003 IEEE Aerospace Conference*, vol. 4, pp. 1979–1993, 2003.

[12] M. R. Morelande, C. M. Kreucher, and K. Kastella, "A Bayesian approach to multiple target detection and tracking," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1589–1604, 2007.

[13] K. J. Kim, "Rao-Blackwellized Gauss-Hermite filter for joint state estimation in cyclic prefixed single-carrier systems," *Signal Processing*, vol. 92, no. 9, pp. 2105–2115, 2012.

[14] M. Orton and W. Fitzgerald, "A Bayesian approach to tracking multiple targets using sensor arrays and particle filters," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 216–223, 2002.

[15] Á. F. García-Fernández, J. Grajal, and M. Morelande, "Two-layer particle filter for multiple target detection and tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 3, pp. 1569–1588, 2013.

[16] W. Yi, M. R. Morelande, L. Kong, and J. Yang, "A computationally efficient particle filter for multitarget tracking using an independence approximation," *IEEE Transactions on Signal Processing*, vol. 61, pp. 843–856, feb 2013.

[17] L. Úbeda-Medina, Á. F. García-Fernández, and J. Grajal, "Generalizations of the auxiliary particle filter for multiple target tracking," in *Proceedings of the17th International Conference on Information Fusion (FUSION)*, 2014.

[18] L. Úbeda-Medina, Á. F. García-Fernández, and J. Grajal, "Adaptive Auxiliary Particle Filter for Track-Before-

Detect With Multiple Targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 5, pp. 2317–2330, 2017.

[19] M. F. Bugallo, T. Lu, and P. M. Djurić, "Target Tracking by Multiple Particle Filtering," *IEEE Aerospace Conference*, pp. 1–7, 2007.

[20] P. M. Djurić, L. Ting, and M. F. Bugallo, "Multiple particle filtering," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 3, pp. 1181–1184, 2007.

[21] P. M. Djurić and M. F. Bugallo, "Particle filtering for high-dimensional systems," *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, CAMSAP 2013*, pp. 352–355, 2013.

[22] J. P. Beaudeau, M. F. Bugallo, and P. M. Djurić, "RSSI-Based Multi-Target Tracking by Cooperative Agents Using Fusion of Cross-Target Information," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5033–5044, 2015.

[23] L. Úbeda-Medina, Á. F. García-Fernández, and J. Grajal, "Target tracking using multiple auxiliary particle filtering," *Proceedings of the 20th International Conference on Information Fusion, FUSION. Xi'an, China.*, pp. 1–8, 2017.

[24] P. Closas, C. Fernández-Prades, and J. Vilá-Valls, "Multiple quadrature Kalman filtering," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6125–6137, 2012.

[25] J. Vilá-Valls, P. Closas, and Á. F. García-Fernández, "Uncertainty Exchange Through Multiple Quadrature Kalman Filtering," *Signal Processing Letters*, vol. 23, no. 12, pp. 1825–1829, 2016.

[26] S. Maskell, M. Rollason, N. Gordon, and D. Salmond, "Efficient particle filtering for multiple target tracking with application to tracking in structured images," *Image and Vision Computing*, vol. 21, no. 10, pp. 931–939, 2003.

[27] J. Beaudeau, M. F. Bugallo, and P. M. Djurić, "Analysis of the cross-target measurement fusion likelihood for RSSI-based sensors," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, no. 3, pp. 1856–1860, 2014.

[28] S. Sadhu, S. Mondal, M. Srinivasan, and T. K. Ghoshal, "Sigma point Kalman filter for bearing only tracking," *Signal Processing*, vol. 86, no. 12, pp. 3769–3777, 2006.

[29] Y. Zhao, "Performance evaluation of Cubature Kalman filter in a GPS/IMU tightly-coupled navigation system," *Signal Processing*, vol. 119, pp. 67–79, 2016.

[30] M. K. Pitt, N. Shephard, and Q. Gate, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, pp. 590–599, 1999.

[31] P. M. Djurić and M. F. Bugallo, "Improved target tracking with particle filtering," *2009 IEEE Aerospace conference*, pp. 1–7, mar 2009.

[32] S. Julier, J. Uhlmann, and H. F. Durrant-whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," in *IEEE Transactions on Automatic Control*, vol. 45, pp. 477–482, 2000.

[33] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[34] A. Kong, J. S. Liu, and W. H. Wong, "Sequential imputations and Bayesian missing data problems," *Journal of the American Statistical Association*, vol. 89, no. 425, pp. 278–288, 1994.

[35] M. Klaas, N. de Freitas, and A. Doucet, "Toward Practical N^2 Monte Carlo: The Marginal Particle Filter," *Proceedings of the Twenty-First Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pp. 308–315,

2005.

[36] X. Chen and S. Godsill, "Multiple dipolar sources localization for MEG using Bayesian particle filtering," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 949–953, 2013.

[37] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, pp. 3447–3457, Aug. 2008.

[38] Á. F. García-Fernández, B.-N. Vo, and B.-T. Vo, "MCMC-based posterior independence approximation for RFS multitarget particle filters," in *Proceedings of the17th International Conference on Information Fusion (FUSION)*, 2014.