Particle Filtering for Multisensor Data Fusion with Switching Observation Models. Application to Land Vehicle Positioning

François Caron, Manuel Davy⋆, Emmanuel Duflos and Philippe Vanheeghe
INRIA-FUTURS "SequeL"
Laboratoire d'Automatique, Génie Informatique et Signal, UMR CNRS 8146
Ecole Centrale de Lille
BP 48, Cité Scientifique, 59651 Villeneuve d'Ascq Cedex
Francois.Caron@ec-lille.fr, Manuel.Davy@ec-lille.fr
Emmanuel.Duflos@ec-lille.fr, Philippe.Vanheeghe.fr

Abstract

This paper concerns the sequential estimation of a hidden state vector from noisy observations delivered by several sensors. Different from the standard framework, we assume here that the sensors may switch autonomously between different sensor states, that is, between different observation models. This includes sensor failure or sensor functioning conditions change. In our model, sensor states are represented by discrete latent variables, which prior probabilities are Markovian. We propose a family of efficient particle filters, for both synchronous and asynchronous sensor observations, as well as for important special cases. Moreover, we discuss connections with previous works. Finally, we study thoroughly a wheel land vehicle positioning problem where the GPS information may be unreliable because of multipath/masking effects.

EDICS: SEN - FUS

Index Terms

Sequential Monte Carlo Methods, Particle Filter, Multisensor System, Data Fusion, Global Positioning System, Switching Observation Model, Fault Detection.

I. Introduction

Statistical Bayesian filtering aims at computing the posterior probability density function (pdf) of a state vector $\mathbf{x}_t \in \mathcal{X}$ from sequentially obtained sensor measurements, called *observations*. It relies, in the state-space formulation, on a state evolution model

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{v}_t) \tag{1}$$

where f is a nonlinear function and $\mathbf{v}_t$ is a white noise. In the following, the time index t is assumed to be an integer, that is, $t \in \{1, 2, 3, \ldots\}$. Eq. (1) can be equivalently written in terms of the conditional pdf $\mathbf{x}_t \sim p(\mathbf{x}_t|\mathbf{x}_{t-1})$, where $\sim$ means 'statistically distributed according to'. The state $\mathbf{x}_t$ is assumed to evolve in a very general state space $\mathcal{X}$, which might be continuous and/or discrete[1].

In complex problems, a single sensor is usually unable to provide full knowledge about the hidden state. It is then necessary to achieve the fusion of several observations $\mathbf{z}_{k,t}$ $(k = 1, \ldots, n)$ provided by n sensors. Each observation is related to the hidden state by an observation probability law $p(\mathbf{z}_{k,t}|\mathbf{x}_t)$.

Sensors delivering observations are generally assumed to be in their nominal state of work (i.e., a nominal observation pdf $p_{k,1}(\mathbf{z}_{k,t}|\mathbf{x}_t)$ for each sensor $k = 1, .., n$). For example, the observation pdf of a camera may be identified during a bright and non-smoggy day. However, in realistic contexts, the sensor functioning conditions may change, resulting in an observation pdf that is quite different from the nominal one. This usually happen because of, e.g., external environment changes or sensor damage. In such cases, it is necessary to detect that the sensor is no more in its nominal state of work and, whenever possible, identify the true state of work of each sensor in order to avoid dramatic state estimation errors. In our framework, this comes down to selecting an observation pdf $p_{k,j}(\mathbf{z}_{k,t}|\mathbf{x}_t)$ in a list. In the camera example discussed above, the statistical relationship between the state $\mathbf{x}_t$ and an observation $\mathbf{z}_{k,t}$ $(k = 1, \ldots, n)$ may undergo changes along the day, due to smog, dust or dawn. Other such examples are provided in the Results Section below. The case of fatal sensor failure is also considered as a major sub-case in the following.

A. *Problem statement*

In this paper, we address sequential estimation problems where a sensor may switch between states of work, using Bayesian filtering. These states of work are denoted by discrete variables $c_{k,t}$, where we adopt the convention $c_{k,t} \in \{0, \ldots, d_k\}$ for $k = 1, \ldots, n$. In the following, the $c_{k,t}$'s are also referred to as *latent variables*, or *sensor states*. Of course, the observation model is determined by $c_{k,t}$. More precisely, for each sensor $k = 1, \ldots, n$, there are d_k possible observation models, denoted (for sensor state $c_{k,t} \in \{1, \ldots, d_k\}$)

$$\mathbf{z}_{k,t} = h_{k,c_{k,t},t}(\mathbf{x}_t) + \mathbf{w}_{k,c_{k,t},t} \tag{2}$$

where $h_{k,c_{k,t},t}$ and $\mathbf{w}_{k,c_{k,t},t}$ are the (nonlinear) observation function and observation noise of the observation model $c_{k,t}$ of sensor k at time t. We adopt the following convention for the definition of the latent variables $c_{k,t}$:

$$\begin{cases} c_{k,t} = 0 \text{ if the observation } \mathbf{z}_{k,t} \text{ is independent of } \mathbf{x}_t \\ c_{k,t} = 1 \text{ if the sensor } k \text{ is in its nominal state of work} \\ c_{k,t} = j, \;\; j \in \{2, .., d_k\} \text{ if the sensor } k \text{ is in} \\ \qquad\qquad\qquad\qquad\quad \text{its } j^{\text{th}} \text{ state of work} \end{cases}$$

[1] For the sake of clarity, we use a slight abuse of notation throughout the paper: though the state/observations may contain discrete components, we denote the state probability distribution as a pdf with $p(\cdot|\cdot)$.

When a sensor undergoes fatal failure, that is, when $\mathbf{z}_{k,t}$ is statistically independent of $\mathbf{x}_t$, $\mathbf{z}_{k,t}$ follows a vague pdf denoted $p_{k,0}(\mathbf{z}_{k,t})$. In the general case, the joint observation pdf for all the n sensors is written compactly as

$$\mathbf{z}_t \sim p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{c}_t) \tag{3}$$

where

$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{c}_t) = \prod_{k=1}^{n} p(\mathbf{z}_{k,t}|\mathbf{x}_t, c_{k,t}) \tag{4}$$

and

$$p(\mathbf{z}_{k,t}|\mathbf{x}_t, c_{k,t}) = \delta_0(c_{k,t})p_{k,0}(\mathbf{z}_{k,t}) + \sum_{j=1}^{d_k} \delta_j(c_{k,t})p_{k,j}(\mathbf{z}_{k,t}|\mathbf{x}_t) \tag{5}$$

with $\mathbf{z}_t = \{\mathbf{z}_{1,t}, \ldots, \mathbf{z}_{n,t}\}$ and $\mathbf{c}_t = \{c_{1,t}, \ldots, c_{n,t}\}$. δ_a is the Dirac delta function centered on a.

The state transition equation (1) and the observation equation (3) being defined, the sequential Bayesian filtering model is almost completely set. In order to make it complete, the prior probability that $c_{k,t}$ (for $k = 1, \ldots, n$) is in a given state j $(j = 0, \ldots, d_k)$ is denoted $\alpha_{k,j,t}$. The $\alpha_{k,j,t}$'s represent the confidence we have in each sensor to be in a given state, as

$$\Pr(c_{k,t} = j) = \alpha_{k,j,t}, \; 0 \leq j \leq d_k \tag{6}$$

where $\alpha_{k,j,t} \geq 0$ and $\sum_{j=0}^{d_k} \alpha_{k,j,t} = 1$.

Let $\boldsymbol{\alpha}_t$ be the vector built from the individual $\alpha_{k,j,t}$'s for $k = 1, \ldots, n$, and $j = 0, \ldots, d_k$. These confidence variables are, in practice, quite difficult to tune a priori, because the sensors external conditions may change rapidly and unexpectedly. In order to tune them adaptatively, we propose to define Markov evolution model for the vector of probabilities $\boldsymbol{\alpha}_t$, as follows

$$\boldsymbol{\alpha}_t \sim p(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1}) \tag{7}$$

Of course, the pdf $p(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1})$ is defined so that the $\alpha_{k,j,t}$'s are probabilities for all t (i.e., they are positive and sum up to one for each sensor k).

B. Estimation objectives and algorithms

Bayesian filtering aims at estimating the posterior pdf $p(\mathbf{x}_{0:t}, \mathbf{c}_{1:t}, \boldsymbol{\alpha}_{0:t}|\mathbf{z}_{1:t})$ with a special interest in the marginal posterior $p(\mathbf{x}_t|\mathbf{z}_{1:t})$. (The notation $\mathbf{a}_{t_1:t_2}$ denotes all the elements of a vector $\mathbf{a}$ between times t_1 and t_2.) In general, $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ has a very complex shape, and cannot be calculated in closed-form. Sequential Monte Carlo methods for Bayesian filtering [1], [2], [3], [4], [5] (also called *particle filters*) provide a numerical approximation of this marginal pdf using a set of weighted random samples, called *particles*. More precisely, the particles $\widetilde{\mathbf{x}}_{0:t}^{(i)}$, with weights $\widetilde{w}_t^{(i)}$, approximate the posterior pdf $p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t})$ thanks to the empirical distribution $P_N(d\mathbf{x}_{0:t})$

$$P_N(d\mathbf{x}_{0:t}) \simeq \sum_{i=1}^{N} \widetilde{w}_t^{(i)} \delta_{\widetilde{\mathbf{x}}_{0:t}^{(i)}}(d\mathbf{x}_{0:t}) \tag{8}$$

(for the sake of notational simplicity, we assume that the state reduces to $\mathbf{x}_t$, up to the end of this subsection). This empirical distribution can be used to compute Bayesian estimates of the state vector $\mathbf{x}_{0:t}$. For example the minimum mean squared error (MMSE) estimate is

$$\begin{aligned}\mathbb{E}_{p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t})}\left[\mathbf{x}_{0:t}\right] &= \int_{\mathcal{X}^{t+1}} \mathbf{x}_{0:t} p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t}) d\mathbf{x}_{0:t} \\ &\approx \int_{\mathcal{X}^{t+1}} \mathbf{x}_{0:t} P_N(d\mathbf{x}_{0:t}) = \sum_{i=1}^{N} \widetilde{w}_t^{(i)} \widetilde{\mathbf{x}}_{0:t}^{(i)} \end{aligned} \tag{9}$$

This approximation becomes more precise as the number N of particles increases [6]. Particle filters have many good properties that make them attractive for practical problems.

- As opposed to Kalman filters-based algorithms (Extended [7] or Unscented [8] Kalman filters), they do not assume Gaussian distribution. They can be implemented for arbitrary (multimodal or highly skewed) posterior pdf, which arise in particular when faced with nonlinear observation and evolution models or non-Gaussian noises [3]
- They can deal with hybrid state vector (both continuous and discrete), see e.g. [9]
- The tradeoff between estimation accuracy and computational load comes down to adjusting the number of particles [3]

The standard particle filter algorithm is based on sequential importance sampling, with weight normalization. It is presented in Algorithm 1 below.

Algorithm 1: Standard Particle filter

```
% Step 1.1 Initialization
• For particles $i = 1, .., N$, do
    – Sample the initial state according to some initial distribution $\pi_0$, that is $\mathbf{x}_0^{(i)} \sim \pi_0(x_0)$
    – Compute the initial weights $w_0^{(i)} = \frac{p(\mathbf{x}_0^{(i)})}{\pi_0(\mathbf{x}_0^{(i)})}$
• Normalize the weights so that $\sum_{i=1}^{N} w_0^{(i)} = 1$
• For $t = 1, 2, ...$ do
% Step 1.2 Iterations
    – For particles $i = 1, .., N$ do
        ∗ Sample the state at time $t$ using the importance distribution $q$, that is $\widetilde{\mathbf{x}}_t^{(i)} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{z}_{1:t})$
        ∗ Update the weight
              $$\widetilde{w}_t^{(i)} \propto \widetilde{w}_{t-1}^{(i)} \frac{p(\mathbf{z}_t|\widetilde{\mathbf{x}}_t^{(i)}) p(\widetilde{\mathbf{x}}_t^{(i)}|\widetilde{\mathbf{x}}_{t-1}^{(i)})}{q_t(\widetilde{\mathbf{x}}_t^{(i)}|\widetilde{\mathbf{x}}_{t-1}^{(i)})}$$
          where the proportionality constant is such that $\sum_{i=1}^{N} \widetilde{w}_t^{(i)} = 1$
    % Step 1.3 Resampling
    – Compute
              $$N_{\text{eff}} = \left[\sum_{i=1}^{N} \left(\widetilde{w}_t^{(i)}\right)^2\right]^{-1} \tag{10}$$
```

  - If $N_{\text{eff}} \leq \eta$ then resample the particles: duplicate particles with large weights and suppress particles with low weights. The resulting particles are denoted $\mathbf{x}_t^{(i)}$ and their weights are $w_t^{(i)}=\frac{1}{N}$. (η is a threshold set to, e.g., $0.8 \times N$.)
  - Otherwise, rename particles, that is, set $\mathbf{x}_t^{(i)} \leftarrow \widetilde{\mathbf{x}}_t^{(i)}$.

C. Main contributions and paper organization

This paper proposes several contributions. First, in Section II, we present the general framework, and the statistical models chosen. Second, the switching sensor states model presented above is fully explored, and a family of particle filtering algorithms especially designed for this model are presented, see Section III. In particular, we propose efficient importance distributions for the state $\mathbf{x}_t$, the latent variables $\mathbf{c}_t$ and the probabilities $\boldsymbol{\alpha}_t$. Third, we consider the important case where the sensors deliver their observations in an asynchronous way. Indeed, in practice, there might be a different sampling device for each sensor, and these devices may not have the same sampling frequencies, nor the same clocks, and the observations are collected at different times for different sensors. We also discuss several important subcases:

- Binary valid/fail sensor states. In most applications, users have one observation model for each sensor, corresponding to the nominal state of work. The important matter concerns possible sensor failures. In this case, the state variable of sensor k is binary, i.e., $c_{k,t} \in \{0, 1\}$ for all t.
- A Rao-Blackwellized algorithm [3] is proposed, for cases where the state evolution and observation models are linear and Gaussian. This enables easy sensor failure detection in already existing Kalman filter-based systems.
- Switching variance observation models are investigated, i.e. when the statistics of the additive noise $w_{k,j,t}$ switch between different values, while the observation function $h_{k,j,t} = h_{k,t}$ remains the same. These noise characteristics changes may be caused by, e.g., external environment conditions which bring supplementary noise.

In Section IV, we propose a survey of related approaches. In particular, the connections of our approach to Jump systems with fixed priors, Jump Markov (Linear) Systems, and that of Hue et. al. [10] are discussed, as well as the differences and relevance of our statistical jump model with respect to these models. Simulations concerning two academic, synthetic data where sensors have two or three states of work, including sensor fatal failure are presented in order to show the accuracy of the method. Comparisons are provided with Jump systems with fixed priors and Jump Markov Systems. Various interpretations and possible extensions are also proposed. In Section V, a real example is fully addressed. It concerns the localization of a land wheel vehicle where the sensors provide measurements of steering, speed and GPS position, potentially corrupted due to multipath or mask effects. Several state-of-the-art Bayesian estimation methods are tested, and our particle filtering approach, together with an unscented-based approximation shows to have the best performance.

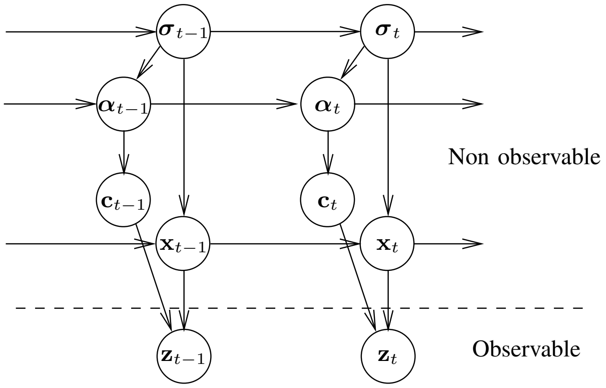

Fig. 1. Overall Bayesian sequential model for multisensor fusion with switching observation models. The discrete vector $\mathbf{c}_t$ indicates the observation model of each sensor, with prior probabilities $\boldsymbol{\alpha}_t$. The evolution of the hidden state $\mathbf{x}_t$ and $\boldsymbol{\alpha}_t$ is tuned by the hyperparameter vector $\boldsymbol{\sigma}_t$ which also evolves. The aim of this model is to estimate accurately the hidden state $\mathbf{x}_t$ from the sensor measurements $\mathbf{z}_t$ even though the sensor state of work may be switching.

II. SEQUENTIAL BAYESIAN MODELS

In this section, we give details about the switching observation model. Moreover, we propose several explicit evolution pdfs for the probabilities $\boldsymbol{\alpha}_t$, and prior pdfs over the observations.

The parameters to be estimated are essentially the state vector $\mathbf{x}_t$ and the discrete variables $\mathbf{c}_t$. We propose to apply sequential Bayesian estimation of these parameters, based on the following evolution models

$$\mathbf{x}_t \sim p(\mathbf{x}_t|\mathbf{x}_{t-1},\boldsymbol{\sigma}_t) \tag{11}$$

$$\mathbf{c}_t \sim \Pr(\mathbf{c}_t|\boldsymbol{\alpha}_t) \tag{12}$$

$$\boldsymbol{\alpha}_t \sim p(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1},\boldsymbol{\sigma}_t) \tag{13}$$

$$\boldsymbol{\sigma}_t \sim p(\boldsymbol{\sigma}_t|\boldsymbol{\sigma}_{t-1}) \tag{14}$$

This general dynamic model is summarized in Fig. 1. It should be noted that, through the incorporation of the Markov prior over $\boldsymbol{\alpha}_t$, the sensor states $\mathbf{c}_t$ are not i.i.d.. Actually, $\mathbf{c}_t$ depends on the past values $\mathbf{c}_0,\dots,\mathbf{c}_{t-1}$ by marginalizing out the $\boldsymbol{\alpha}_0,\dots,\boldsymbol{\alpha}_t$. These evolution models (also called *prior pdfs*) are:

- $p(\mathbf{x}_t|\mathbf{x}_{t-1},\boldsymbol{\sigma}_t)$ is the state evolution model, defined in Eq. (1) for $t=1,2,\ldots$. At time 0, it is assumed that $\mathbf{x}_0 \sim p_0(\mathbf{x}_0)$.

- $\Pr(\mathbf{c}_t|\boldsymbol{\alpha}_t)$ is the prior over the sensor state variables $c_{k,t}$, $k=1,\dots,n$. It is defined as

$$\Pr(\mathbf{c}_t|\boldsymbol{\alpha}_t) = \prod_{k=1}^{n} \Pr(c_{k,t}|\boldsymbol{\alpha}_{k,t}) \tag{15}$$

with

$$\Pr(c_{k,t}|\boldsymbol{\alpha}_{k,t}) = \sum_{j=0}^{d_k} \delta_j(c_{k,t})\alpha_{k,j,t} \tag{16}$$

where $\boldsymbol{\alpha}_{k,t} = \{\alpha_{k,0,t}...\alpha_{k,d_k,t}\}$. The choice of probabilities $\alpha_{k,j,t}$ is extremely sensitive for the correct estimation of sensors state of work. Moreover, the $\alpha_{k,j,t}$'s might evolve, because sensor reliability decreases as the sensor becomes older, for example. Consequently, these probabilities are considered as unknown parameters with a Markov prior, and they are also estimated.

- The evolution model $p(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1})$ for $\boldsymbol{\alpha}_t$ is as a Dirichlet distribution for each sensor $k = 1, \dots, n$

$$(\alpha_{k,0,t}, \dots, \alpha_{k,d_k,t}) \sim \mathcal{D}(\sigma^{\alpha}_{k,t}\alpha_{k,0,t-1}, \dots, \sigma^{\alpha}_{k,t}\alpha_{k,d_k,t-1}) \tag{17}$$

where $\sigma^{\alpha}_{k,t}$ is a sensor specific coefficient which adjusts the spread of the $\alpha_{k,j,t}$'s. We recall the definition of a Dirichlet distribution for a set of random variables $(b_0, .., b_p) \sim \mathcal{D}(a_0, .., a_p)$

$$\mathcal{D}(a_0, .., a_p) = \frac{\Gamma(\sum_{l=0}^{p} a_l)}{\prod_{l=0}^{p} \Gamma(a_l)} \prod_{l=0}^{p} b_l^{a_l - 1} \delta_1(\sum_{l=1}^{p} b_l) \tag{18}$$

where Γ is the gamma function. In the important special case where $c_{k,t} \in \{0, 1\}$ $(d_k = 1)$, the Dirichlet distribution reduces then to a beta distribution.

At time $t = 0$, we assume that $\alpha_{k,0} \sim p_0(\alpha_{k,0})$. The coefficient $\sigma^{\alpha}_{k,t}$ tunes the evolution of $\alpha_{k,t}$. In order to avoid setting a fixed value for $\sigma^{\alpha}_{k,t}$ (and thus fixed dynamics for $\alpha_{k,t}$) this coefficient is also estimated, as explained below, Eq (19).

- $p(\boldsymbol{\sigma}_t|\boldsymbol{\sigma}_{t-1})$ is the hyperparameter transition equation. In many problems, it is assumed that the evolution noise $\mathbf{v}_t$ is Gaussian. In this case, each scalar component of $\mathbf{v}_t$ is assumed to have variance $(\sigma^v_t)^2$. As these variances have an important influence on the Bayesian filter behavior, it may be important to consider them as unknowns. For each of these variances, the evolution model may be

$$\log(\sigma^v_t) = \log(\sigma^v_{t-1}) + \lambda^v \tag{19}$$

where λ^v is a zero mean white Gaussian noise with fixed and known variance. The $\log$ function is used to ensure that the variances remain positive. This can also be applied to the hyperparameter that tunes the evolution of $\boldsymbol{\alpha}_t$, denoted $\sigma^{\alpha}_{k,t}$ above. A similar hyperparameter evolution model may be found in [9] where it is shown that letting the hyperparameters follow this model robustifies the estimation. The shorthand $\boldsymbol{\sigma}_t = \{\sigma^v_t, \sigma^{\alpha}_{k,t}|k = 1, .., n\}$ denotes the full set of hyperparameters. In the following, the hyperparameter σ^v_t will be assumed as known and fixed (considering that it is not the central point of this article) and the shorthand $\boldsymbol{\sigma}_t = \{\sigma^{\alpha}_{k,t}|k = 1, .., n\}$.

The model defined above allows, in theory, the estimation of the augmented state composed of the state vector $\mathbf{x}_t$, the discrete latent variables $\mathbf{c}_t$, the confidence coefficients $\boldsymbol{\alpha}_t$ and the hyperparameters $\boldsymbol{\sigma}_t$. In the general case, it is not possible to calculate the posterior $p(\mathbf{x}_{0:t}, \mathbf{c}_{1:t}, \boldsymbol{\alpha}_{0:t}, \boldsymbol{\sigma}_{0:t}|\mathbf{z}_{1:t})$ in closed form at each time step. Particle filters [3] are widely used and provide the most convincing approach to tackle such problems. In the following section, we present several algorithms for synchronous/asynchronous sensor measurements, and for important special cases.

III. Particle Filtering algorithms for switching observation models

In this section, we present several particle filtering algorithms for switching observation models. The cases of synchronous and asynchronous measurements are resp. addressed in Subsections III-A and III-B, where dedicated algorithms are presented. In Subsection III-C, importance distributions are presented, some being designed for the general switching observation model framework, and some especially for the special cases a) Binary case valid/failing ($c_{k,t} \in \{0,1\}$), b) conditionally linear and Gaussian System and c) Switching observation variances.

A. SISR algorithm for synchronous sensors

In some cases, the measurements provided by the n sensors are synchronous, that is, all the sensors have the same sampling frequency and the measurements from all the sensors arrive at the same time. The corresponding algorithm is presented in Algorithm 2 below.

Algorithm 2: Particle filter for switching observation models – Synchronous case

```
% Step 2.1 Initialization
• For particles $i = 1,..,N$ sample[2] $x_0^{(i)} \sim p_0(x_0)$
• For particles $i = 1,..,N$, sample $\boldsymbol{\sigma}_0^{(i)} \sim p_0(\boldsymbol{\sigma}_0)$
• For particles $i = 1,..,N$, sample $\boldsymbol{\alpha}_0^{(i)} \sim p_0(\boldsymbol{\alpha}_0|\boldsymbol{\sigma}_0^{(i)})$
• Set the initial weights $w_0^{(i)} \leftarrow \frac{1}{N}$
% Step 2.2 Iterations
• For t=1,2,... do
    – For particles $i = 1,..,N$, do
        ∗ Sample the sensor state variable $\widetilde{\mathbf{c}}_t^{(i)} \sim q(\mathbf{c}_t|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{t-1}^{(i)}, \mathbf{z}_t)$
        ∗ Sample the state vector $\widetilde{\mathbf{x}}_t^{(i)} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \widetilde{\mathbf{c}}_t^{(i)}, \mathbf{z}_t)$
        ∗ Sample the probabilities $\widetilde{\boldsymbol{\alpha}}_t^{(i)} \sim q(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1}^{(i)}, \widetilde{\mathbf{c}}_t^{(i)}, \boldsymbol{\sigma}_{t-1}^{(i)})$
        ∗ Sample the hyperparameter vector $\widetilde{\boldsymbol{\sigma}}_t^{(i)} \sim q(\boldsymbol{\sigma}_t|\boldsymbol{\sigma}_{t-1}^{(i)}, \widetilde{\boldsymbol{\alpha}}_t^{(i)}, \boldsymbol{\alpha}_{t-1}^{(i)})$
    – For particles $i = 1,..,N$, update the weights
        $$\widetilde{w}_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(\mathbf{z}_t|\widetilde{\mathbf{x}}_t^{(i)}, \widetilde{\mathbf{c}}_t^{(i)}) p(\widetilde{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\widetilde{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \widetilde{\mathbf{c}}_t^{(i)}, \mathbf{z}_t) q(\widetilde{\mathbf{c}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{t-1}^{(i)}, \mathbf{z}_t)} \times \frac{p(\widetilde{\mathbf{c}}_t^{(i)}|\widetilde{\boldsymbol{\alpha}}_t^{(i)}) p(\widetilde{\boldsymbol{\alpha}}_t^{(i)}|\boldsymbol{\alpha}_{t-1}^{(i)}, \widetilde{\boldsymbol{\sigma}}_t^{(i)}) p(\widetilde{\boldsymbol{\sigma}}_t^{(i)}|\boldsymbol{\sigma}_{t-1}^{(i)})}{q(\widetilde{\boldsymbol{\alpha}}_t^{(i)}|\boldsymbol{\alpha}_{t-1}^{(i)}) q(\widetilde{\boldsymbol{\sigma}}_t^{(i)}|\boldsymbol{\sigma}_{t-1}^{(i)}, \widetilde{\boldsymbol{\alpha}}_t^{(i)}, \boldsymbol{\alpha}_{t-1}^{(i)})} \qquad (20)$$
    – Normalize the weights so that $\sum_{i=1}^{N} \widetilde{w}_t^{(i)} = 1$
    % Step 2.3 Resampling
    – Compute $N_{\text{eff}}$ as in Eq. (10) and perform particle resampling whenever $N_{\text{eff}} < \eta$ , see Step 1.3 in Algorithm 1.
```

The importance distributions $q(\cdot)$ used in Algorithm 2 are presented in Subsection III-C below.

[2] The pdfs $p_0(\cdot)$ used in the initialization can be any pdf over the state space, e.g., uniform or derived from some heuristics.

B. SISR algorithm for asynchronous sensors

In many cases, the sensors are equipped with individual sampling devices which do not have the same sampling frequencies nor the same clock. As a result, the measurements originating from each of the sensors are available at different times. This requires a specific particle filtering algorithm, which is presented in Algorithm 3 below.

Algorithm 3: Particle filter for switching

observation models – Asynchronous case

```
% Step 3.1: Initialization
• This step is similar to Step 2.1 in Algorithm 2.
% Step 3.2: Iterations
• For t=1,2,... do
    – Await the arrival of a new measure $\mathbf{z}_{k,t}$, delivered by sensor $k$, $k = 1, \dots, n$ and, for particles $i = 1, .., N$, do
        ∗ Sample the state of sensor $k$ as $\widetilde{c}_{k,t}^{(i)} \sim q(c_{k,t}|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)}, \mathbf{z}_{k,t})$
        ∗ Sample the state variable $\widetilde{\mathbf{x}}_t^{(i)} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \widetilde{c}_{k,t}^{(i)}, \mathbf{z}_{k,t})$
        ∗ Sample the sensor state probabilities $\widetilde{\boldsymbol{\alpha}}_{k,t}^{(i)} \sim q(\boldsymbol{\alpha}_{k,t}|\boldsymbol{\alpha}_{k,t-1}^{(i)}, \widetilde{c}_{k,t}^{(i)}, \boldsymbol{\sigma}_{t-1}^{(i)})$ for sensor $k$
        ∗ Sample the relevant hyperparameters $\widetilde{\sigma}_{k,t}^{\alpha\ (i)} \sim q(\sigma_{k,t}^{\alpha}|\sigma_{k,t-1}^{\alpha\ (i)}, \widetilde{\boldsymbol{\alpha}}_{k,t}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)})$
    – For particles $i = 1, .., N$, update the weights
```

$$\begin{aligned}\widetilde{w}_t^{(i)} \propto w_{t-1}^{(i)} & \frac{p(\mathbf{z}_{k,t}|\widetilde{\mathbf{x}}_t^{(i)}, \widetilde{c}_{k,t}^{(i)})p(\widetilde{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\widetilde{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \widetilde{c}_{k,t}^{(i)}, \mathbf{z}_{k,t})} \\ & \times \frac{p(\widetilde{c}_{k,t}^{(i)}|\widetilde{\boldsymbol{\alpha}}_{k,t}^{(i)})p(\widetilde{\boldsymbol{\alpha}}_{k,t}^{(i)}|\boldsymbol{\alpha}_{k,t-1}^{(i)}, \widetilde{\sigma}_{k,t}^{\alpha\ (i)})}{q(\widetilde{c}_{k,t}^{(i)}|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)}, \mathbf{z}_{k,t})q(\widetilde{\boldsymbol{\alpha}}_{k,t}^{(i)}|\boldsymbol{\alpha}_{k,t-1}^{(i)})} \\ & \times \frac{p(\widetilde{c}_{k,t}^{(i)}|\widetilde{\boldsymbol{\alpha}}_{k,t}^{(i)})p(\widetilde{\sigma}_{k,t}^{\alpha\ (i)}|\sigma_{k,t-1}^{\alpha\ (i)})}{q(\widetilde{\sigma}_{k,t}^{\alpha\ (i)}|\sigma_{k,t-1}^{\alpha\ (i)}, \widetilde{\boldsymbol{\alpha}}_t^{(i)}, \boldsymbol{\alpha}_{t-1}^{(i)})}\end{aligned} \tag{21}$$

```
      Normalize the weights so that $\sum_{i=1}^N \widetilde{w}_t^{(i)} = 1$
    % Step 3.3 Resampling
    – This step is similar to Step 2.3 in Algorithm 2.
```

The principle of this algorithm is quite close to that of Algorithm 2. In the latter algorithm, however, the state vector $\mathbf{x}_t$ is updated each time a new observation arrives from one of the sensors. In practice, Algorithm 3 is more likely to be used, as many new sensors incorporate a sampling/pre-processing unit.

The importance distributions used in Algorithm 3, as well as those of Algorithm 2, are described in the next subsection.

C. Importance distributions

In Algorithms 2 and 3, particles are extended from time $t-1$ to time t using importance distributions denoted $q(\cdot)$. We have defined one for each of the variables, namely $q(\mathbf{c}_t|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{t-1}^{(i)}, \mathbf{z}_t)$, $q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \widetilde{\mathbf{c}}_t^{(i)}, \mathbf{z}_t)$, $q(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1}^{(i)}, \widetilde{\mathbf{c}}_t^{(i)}, \boldsymbol{\sigma}_{t-1}^{(i)})$ and $q(\boldsymbol{\sigma}_t|\boldsymbol{\sigma}_{t-1}^{(i)}, \widetilde{\boldsymbol{\alpha}}_t^{(i)}, \boldsymbol{\alpha}_{t-1}^{(i)})$. Their choice is of paramount importance for particle filtering algorithm efficiency [2]. The selection of the state importance distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \widetilde{\mathbf{c}}_t^{(i)}, \mathbf{z}_t)$ is not specific to our framework, i.e., it may be one of the standard choices [2], [3]: the optimal importance distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \widetilde{\mathbf{c}}_t^{(i)}, \mathbf{z}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \widetilde{\mathbf{c}}_t^{(i)}, \mathbf{z}_t)$, or approximations of it obtained by applying Extended/Unscented Kalman filtering [2], [11].

The choice of the importance distribution $q(\mathbf{c}_t|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{t-1}^{(i)}, \mathbf{z}_t)$ is crucial and it is specific to our framework. A first remark is that the number of possible values of vector $\mathbf{c}_t$ is $\prod_{k=1}^{n}(1+d_k)$, and testing each possible combination is not realistic. For optimal efficiency, it is necessary that the distribution $q(\mathbf{c}_t|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{t-1}^{(i)}, \mathbf{z}_t)$ proposes the most probable configurations, that is, the most probable states $c_{k,t}$ for each sensor, $k = 1, \ldots, n$. Since the sensors are likely to switch state independently of the other sensors, this importance distribution is written

$$q(\mathbf{c}_t|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{t-1}^{(i)}, \mathbf{z}_t) \;=\; \prod_{k=1}^{n} q(c_{k,t}|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)}, \mathbf{z}_{k,t}) \tag{22}$$

where the individual importance distributions $q(c_{k,t}|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)}, \mathbf{z}_{k,t})$ are also used in Algorithm 3. The optimal importance distribution is $q(c_{k,t}|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)}, \mathbf{z}_{k,t}) = \Pr(c_{k,t}|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)}, \mathbf{z}_{k,t})$, defined by

$$\Pr(c_{k,t}|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)}, \mathbf{z}_{k,t}) = \frac{\alpha_{k,c_{k,t},t-1}^{(i)} p(\mathbf{z}_{k,t}|c_{k,t}, \mathbf{x}_{t-1}^{(i)})}{\sum_{j=0}^{d_k} \alpha_{k,j,t-1}^{(i)} p(\mathbf{z}_{k,t}|j, \mathbf{x}_{t-1}^{(i)})} \tag{23}$$

As this optimal importance distribution cannot be computed analytically, we use an approximation of it as importance function. Using an EKF step, the pdfs $p(\mathbf{z}_{k,t}|j, \mathbf{x}_{t-1}^{(i)})$ for $j = 0, \ldots, d_k$ are approximated by

$$p(\mathbf{z}_{k,t}|j, \mathbf{x}_{t-1}^{(i)}) \simeq \mathcal{N}(h_{k,j,t}(\widehat{\mathbf{x}}_{t|t-1}^{(i)}), S_{k,j,t}^{(i)})$$

where $\widehat{\mathbf{x}}_{t|t-1}^{(i)} = f(\mathbf{x}_{t-1}^{(i)})$ is the state prediction and $S_{k,j,t}^{(i)}$ is the approximated innovation covariance matrix defined by

$$S_{k,j,t}^{(i)} = \nabla h_{k,j,t}^{(i)} Q_t^{(i)} \nabla h_{k,j,t}^{(i)\,T} + R_{k,j,t}^{(i)}$$

where $R_{k,j,t}^{(i)}$ is the covariance matrix of the additive noise $\mathbf{w}_{k,j,t}^{(i)}$ for sensor k and particle i (which reduces to $\sigma_{k,j,t}^{(i)}$ in case of scalar observations), $Q_t^{(i)}$ is the covariance matrix of the state evolution noise $\mathbf{v}_t$ (which reduces to its diagonal components $\{\sigma_t^{v\,(i)}\}$ in case of independent state components evolution) and

$$\nabla h_{k,j,t}^{(i)} = \left.\frac{\partial h_{k,j,t}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\widehat{\mathbf{x}}_{t|t-1}^{(i)}} \tag{24}$$

The importance distribution $q(c_{k,t}|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)}, \mathbf{z}_{k,t})$ is thus defined by

$$q(c_{k,t}|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)}, \mathbf{z}_{k,t}) = \frac{\alpha_{k,c_{k,t},t-1}^{(i)} \mathcal{N}(h_{k,c_{k,t},t}(\widehat{\mathbf{x}}_{t|t-1}^{(i)}), S_{k,c_{k,t},t}^{(i)})}{\sum_{j=0}^{d_k} \alpha_{k,j,t-1}^{(i)} \mathcal{N}(h_{k,j,t}(\widehat{\mathbf{x}}_{t|t-1}^{(i)}), S_{k,j,t}^{(i)})} \tag{25}$$

The importance distributions $q(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1}^{(i)}, \widetilde{\mathbf{c}}_t^{(i)}, \boldsymbol{\sigma}_{t-1}^{(i)})$ and $q(\boldsymbol{\sigma}_t|\boldsymbol{\sigma}_{t-1}^{(i)}, \widetilde{\boldsymbol{\alpha}}_{k,t}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)})$ can be chosen as the optimal importance distributions, that is those that minimize the variance of the weights, conditional on the observations [3].

Actually, these can be computed in closed-form, and reject sampling [12] may be implemented whenever direct sampling is not possible. Considering that the Dirichlet distribution is conjugate to the multinomial distribution, the optimal importance distributions $q(\boldsymbol{\alpha}_{k,t}|\boldsymbol{\alpha}_{k,t-1}^{(i)}, c_{k,t}^{(i)}, \sigma_{k,t}^{\alpha\ (i)})$ for each sensor are given by

$$q(\boldsymbol{\alpha}_{k,t}|\boldsymbol{\alpha}_{k,t-1}^{(i)}, c_{k,t}^{(i)}, \sigma_{k,t}^{\alpha\ (i)}) = \mathcal{D}(\sigma_{k,t}^{\prime\ \alpha\ (i)} \boldsymbol{\alpha}_{k,t-1}^{\prime(i)}) \tag{26}$$

where $\sigma_{k,t}^{\prime\ \alpha\ (i)} = \sigma_{k,t}^{\alpha\ (i)} + 1$ and $\alpha_{k,j,t-1}^{\prime(i)} = \frac{\sigma_{k,t}^{\alpha\ (i)}}{\sigma_{k,t}^{\alpha\ (i)}+1} \alpha_{k,j,t-1}^{(i)} + \frac{1}{\sigma_{k,t}^{\alpha\ (i)}+1} \delta_{c_{k,t}^{(i)}}(j)$ for $j = 1, \ldots, d_k$. For the full set of sensors,

$$q(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1}^{(i)}, \mathbf{c}_t^{(i)}, \boldsymbol{\sigma}_t^{\alpha\ (i)}) = \prod_{k=1}^{n} q(\boldsymbol{\alpha}_{k,t}|\boldsymbol{\alpha}_{k,t-1}^{(i)}, c_{k,t}^{(i)}, \sigma_{k,t}^{\alpha\ (i)}) \tag{27}$$

Concerning the hyperparameter $\sigma_{k,t-1}^{\alpha}$, the optimal importance distribution is

$$\begin{aligned} q(\log(\sigma_{k,t}^{\alpha})|\boldsymbol{\alpha}_{k,t}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)}, \sigma_{k,t-1}^{\alpha\ (i)}) \ &= \ \frac{\mathcal{D}(\boldsymbol{\alpha}_{k,t}^{(i)}; \sigma_{k,t}^{\alpha} \boldsymbol{\alpha}_{k,t-1}^{(i)})}{\mathcal{D}(\boldsymbol{\alpha}_{k,t}^{(i)}; \sigma_{k,t-1}^{\alpha\ (i)} \boldsymbol{\alpha}_{k,t-1}^{(i)})} \\ &\quad \times \mathcal{N}(\log(\sigma_{k,t}^{\alpha}); \log(\sigma_{k,t-1}^{\alpha\ (i)}), \lambda^{\alpha}) \end{aligned} \tag{28}$$

D. *Important special cases*

In this subsection, we present additional details for three special cases, which have some importance in applications.

1) Binary valid/failing sensor states: The binary valid/failing sensor states case is of great importance. For example, in the chemical industry, sensors are immersed into reagents which may be quite corrosive, making sensor faults likely, and difficult to check on-site. Another example is that of autonomous vehicles, which need to check the information from the sensors are valid in order to avoid dramatic state estimation errors which would result in vehicle crash. For this case, we present the specific choices of the importance distributions $q(c_{k,t}|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)}, \mathbf{z}_{k,t})$ and $q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \widetilde{\mathbf{c}}_t^{(i)}, \mathbf{z}_t)$.

Here, the sensor states may be either $j = 0$ or $j = 1$. The importance distribution, $q(c_{k,t}|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)}, \mathbf{z}_{k,t})$ reduces to

$$\begin{aligned} q(c_{k,t}|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)}, \mathbf{z}_{k,t}) &\propto (1 - \alpha_{k,1,t-1}^{(i)}) p_0(\mathbf{z}_{k,t}) \delta_0(c_{k,t}) \\ &\quad + \alpha_{k,1,t-1}^{(i)} \mathcal{N}(h_{k,1}(\widehat{\mathbf{x}}_{t|t-1}^{(i)}), S_{k,1,t}^{(i)}) \delta_1(c_{k,t}) \end{aligned} \tag{29}$$

Once $\widetilde{c}_{k,t}^{(i)}$ has been sampled in Algorithm 2 or in Algorithm 3, the state $\widetilde{\mathbf{x}}_t^{(i)}$ is sampled using $q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \widetilde{\mathbf{c}}_t^{(i)}, \mathbf{z}_t)$, which is a Gaussian distribution whose mean and covariance matrix are given by the multisensor extended Kalman filter, as follows: First, compute the estimate

$$\widehat{\mathbf{x}}_{t|t}^{(i)} = \widehat{\mathbf{x}}_{t|t-1}^{(i)} + \sum_{k=1}^{n} \widetilde{c}_{k,t}^{(i)} K_{k,t}^{(i)} \boldsymbol{\nu}_{k,t}^{(i)} \tag{30}$$

where $\boldsymbol{\nu}_{k,t}^{(i)} = \mathbf{z}_{k,t} - h_{k,1,t}(\widehat{\mathbf{x}}_{t|t-1}^{(i)})$ and the Kalman gain is

$$K_{k,t}^{(i)} = \Sigma_{t|t}^{(i)} \left(\nabla h_{k,1,t}^{(i)}\right)^T R_{k,1,t}^{(i)\ -1} \tag{31}$$

Finally, let

$$\Sigma^{(i)}_{t|t} = \left[\Sigma^{-1}_{t|t-1} + \sum_{k=1}^{n} \widetilde{c}^{(i)}_{k,t} \left(\nabla h^{(i)}_{k,1,t}\right)^{T} R^{-1}_{k,1,t} \nabla h^{(i)}_{k,1,t}\right]^{-1} \tag{32}$$

and set $q(\mathbf{x}_t|\mathbf{x}^{(i)}_{t-1}, \widetilde{\mathbf{c}}^{(i)}_t, \mathbf{z}_t) = \mathcal{N}(\mathbf{x}_t; \widehat{\mathbf{x}}^{(i)}_{t|t}, \Sigma^{(i)}_{t|t})$. The advantage of this approach is that the extended Kalman filter quantities are used to sample both $\widetilde{c}^{(i)}_{k,t}$ and $\widetilde{\mathbf{x}}^{(i)}_t$, which saves computations. The other importance distributions are as explained above.

2) Conditionally linear and Gaussian model: Another important special case is when the Bayesian model is linear and Gaussian, conditional on the sensor state variable $\mathbf{c}_t$ (and on the hyperparameters $\boldsymbol{\sigma}_t$):

$$\mathbf{x}_t = F_t \mathbf{x}_{t-1} + G_t \mathbf{v}_t \tag{33}$$

$$\mathbf{z}_{k,t} = H_{k,j,t} \mathbf{x}_t + \mathbf{w}_{k,j,t} \tag{34}$$

with F_t the state matrix, G_t the noise transition matrix, $\mathbf{v}_t$ a zero mean white Gaussian noise of covariance matrix Q_t, $H_{k,j,t}$ the observation matrix for the sensor k in state j and $\mathbf{w}_{k,j,t}$ the zero mean white Gaussian noise with covariance matrix $R_{k,j,t}$. This case is important when we have an already existing Kalman filter working for a given sensor state j, and we want to add the extra possibility of detecting sensor faults, or state of work switch. The corresponding algorithm implements a bank of interacting Kalman filters, known as a *Rao-Blackwellized* particle filter, or a Mixture Kalman Filter, see [13], [14], [15], [16], [17] for examples. The algorithm below concerns the asynchronous case. The proposal distributions are those described in Subsection III-C.

Algorithm 4: Rao-Blackwellized particle filter for

switching observation models – Asynchronous case

```
% Step 4.1 Initialisation
• For particles $i = 1,..,N$ sample $\widehat{\mathbf{x}}^{(i)}_{0|0} \sim p_0(\widehat{\mathbf{x}}_{0|0})$
• For particles $i = 1,..,N$, sample $\Sigma^{(i)}_{0|0} \sim p_0(\Sigma_{0|0})$
• For particles $i = 1,..,N$, sample $\boldsymbol{\sigma}^{(i)}_0 \sim p_0(\boldsymbol{\sigma}_0)$
• For particles $i = 1,..,N$, sample $\boldsymbol{\alpha}^{(i)}_0 \sim p_0(\boldsymbol{\alpha}_0|\boldsymbol{\sigma}^{(i)}_0)$
• Set the initial weights $w^{(i)}_0 \leftarrow \frac{1}{N}$
% Step 4.2 Iterations
• For t=1,2,... do
    – Await the arrival of a new measure $\mathbf{z}_{k,t}$, delivered by sensor $k$, $k = 1, \ldots, n$ and, for particles $i = 1,..,N$, do
        ∗ Sample the sensor state variable $\widetilde{c}^{(i)}_{k,t} \sim q(c_{k,t}|\widehat{\mathbf{x}}^{(i)}_{t-1|t-1}, \Sigma^{(i)}_{t-1|t-1}, \boldsymbol{\alpha}^{(i)}_{k,t-1}, \mathbf{z}_{k,t})$
        ∗ Sample the probabilities $\widetilde{\boldsymbol{\alpha}}^{(i)}_{k,t} \sim q(\boldsymbol{\alpha}_{k,t}|\boldsymbol{\alpha}^{(i)}_{k,t-1}, \widetilde{c}^{(i)}_{k,t}, \boldsymbol{\sigma}^{(i)}_{t-1})$
        ∗ Sample the hyperparameter vector $\widetilde{\boldsymbol{\sigma}}^{(i)}_t \sim q(\boldsymbol{\sigma}_t|\boldsymbol{\sigma}^{(i)}_{t-1}, \widetilde{\boldsymbol{\alpha}}^{(i)}_{k,t}, \boldsymbol{\alpha}^{(i)}_{k,t-1})$
        ∗ Update mean and covariance matrix with a Kalman filter step $(\widehat{\mathbf{x}}^{(i)}_{t|t}, \Sigma^{(i)}_{t|t})$ =KF$(\widehat{\mathbf{x}}^{(i)}_{t-1|t-1}, \Sigma^{(i)}_{t-1|t-1}, \widetilde{c}^{(i)}_{k,t}, \mathbf{z}_{k,t})$
```

- For particles $i = 1, .., N$, update the weights

$$\begin{aligned}\widetilde{w}_t^{(i)} \propto w_{t-1}^{(i)} & \frac{p(\mathbf{z}_{k,t}|\widetilde{\mathbf{c}}_{1:t}^{(i)}, \mathbf{z}_{1:t-1}) p(\widetilde{c}_{k,t}^{(i)}|\widetilde{\boldsymbol{\alpha}}_{k,t}^{(i)})}{q(\widetilde{c}_{k,t}^{(i)}|\widehat{\mathbf{x}}_{t-1|t-1}^{(i)}, \Sigma_{t-1|t-1}^{(i)}, \boldsymbol{\alpha}_{k,t-1}^{(i)}, \mathbf{z}_{k,t}))} \\ \times & \frac{p(\widetilde{\boldsymbol{\alpha}}_{k,t}^{(i)}|\boldsymbol{\alpha}_{k,t-1}^{(i)}) p(\widetilde{\boldsymbol{\sigma}}_t^{(i)}|\boldsymbol{\sigma}_{t-1}^{(i)})}{q(\widetilde{\boldsymbol{\alpha}}_{k,t}^{(i)}|\boldsymbol{\alpha}_{k,t-1}^{(i)}, \widetilde{c}_{k,t}^{(i)}, \boldsymbol{\sigma}_{t-1}^{(i)}) q(\widetilde{\boldsymbol{\sigma}}_t^{(i)}|\boldsymbol{\sigma}_{t-1}^{(i)}, \widetilde{\boldsymbol{\alpha}}_t^{(i)}, \boldsymbol{\alpha}_{t-1}^{(i)})}\end{aligned} \quad (35)$$

- Normalize the weights so that $\sum_{i=1}^{N} \widetilde{w}_t^{(i)} = 1$

% `Step 4.3` Resampling

- Compute N_{eff} as in Eq. (10) and perform particle resampling whenever $N_{\text{eff}} < \eta$, see Step 1.3 in Algorithm 1.

---

3) Switching observation variances: A last important case is when the observation model h_k, j, t in Eq. (2) of a given sensor k does not change, but the additive white noise pdf may switch between pre-defined distributions [18]. The additive noise of the j^{th} pdf is denoted $\mathbf{w}_{k,j,t}$. An interesting case is when all the $\mathbf{w}_{k,j,t}$, $j = 1, \ldots, d_k$ are zero-mean and monomodal, with different variances (we assume the noises are sorted so that $\mathbf{w}_{k,1,t}$ has the smallest variance whereas $\mathbf{w}_{k,d_k,t}$ the largest). Similar to the above cases, the importance distribution $q(\mathbf{c}_t|\mathbf{x}_{t-1}^{(i)}, \boldsymbol{\alpha}_{t-1}^{(i)}, \mathbf{z}_t)$ can be computed from the approximated optimal pdf using an EKF step.

An alternative solution consist of defining an evolution model for the variance of $\mathbf{w}_{k,t}$ under the sensor state $j = 1$. This simple solution, however, does not enable switching between discrete variance states, as there is a continuity across the noise variances. An attractive solution consists of setting an upper bound on the variance of $\mathbf{w}_{k,t}$ and switching to the state $j = 0$ whenever the variance becomes too large, thus detecting sensor failure.

IV. Discussion

In this section, we first survey previous related works, and we discuss their connection with our approach. Then, we present an alternative interpretation and possible extensions. Some simulation results for synthetic data showing the efficiency of the approach are proposed in Subsection IV-C (the real example of land vehicle positioning is presented in the next section).

A. Previous works

Switching state-space models, or jump state-space models, have been widely studied in the literature. Such models are composed of

- A discrete random indicator variable $\mathbf{c}_t$, with some statistical structure, which governs the behavior of the statistical state-space model that switches from one model to another
- A state vector $\mathbf{x}_t$ that evolves according to a stochastic evolution model $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{c}_t)$
- An observation vector $\mathbf{z}_t$ that is related to the state vector $\mathbf{x}_t$ through a stochastic observation model $p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{c}_t)$

Different ways of defining the statistical structure of the indicator variable $\mathbf{c}_t$ have been investigated[3]

- The $\mathbf{c}_t$'s may be defined as random iid variables with (known or unknown) fixed prior $\Pr(\mathbf{c}_t)$. In that case we have

$$\Pr(\mathbf{c}_t|\mathbf{c}_{1:t-1}) = \Pr(\mathbf{c}_t)$$

- The $\mathbf{c}_t$'s may be defined as a discrete-time homogeneous Markov chain with (known or unknown) fixed transition matrix, and then

$$\Pr(\mathbf{c}_t|\mathbf{c}_{1:t-1}) = \Pr(\mathbf{c}_t|\mathbf{c}_{t-1})$$

These last models, known as Jump Markov Systems (JMS), have been widely studied in the literature, especially when the state-space models are conditionnally linear and Gaussian (Jump Markov Linear Systems, JMLS) [19], [20], [21]. Many algorithms have been defined to solve this estimation problem, mainly based on Gaussian mixture approximations, like the Interacting Multiple Model (IMM) algorithm [22], [23], or the Generalized Pseudo-Bayes (GPB) algorithms [24]. Efficient sequential Monte Carlo algorithms have then been defined for both JMS [9], [25], [26] and JMLS [27], [15]. These algorithms have been successfully applied in various area like digital communications [28], [29], multitarget tracking [30], [31], [32], fault diagnosis [13], [33], [34], geoscience [35] or image processing [18].

In our model, we do not make such a Markovian assumption for the indicator variable $\mathbf{c}_t$, but for its prior $\boldsymbol{\alpha}_t$. By doing this, one has

$$\Pr(\mathbf{c}_t|\mathbf{c}_{1:t-1}) = \int \Pr(\mathbf{c}_t|\boldsymbol{\alpha}_t)\Pr(\boldsymbol{\alpha}_t|\mathbf{c}_{1:t-1})d\boldsymbol{\alpha}_t$$

and, through $\boldsymbol{\alpha}_t$, $\mathbf{c}_t$ is a priori dependent of the whole trajectory $\mathbf{c}_{1:t-1}$, which is more likely for the problem we are considering. Our model thus introduces a memory over the past sensor states, rather than simply considering the last one. This memory is tuned by the hyperparameters $\sigma^{\alpha}_{k,t}$ (a high value results in a long memory and conversely). This is of particular interest when considering sensor failures. For example, in chemical industry, the confidence in a sensor will generally decrease with time. In land vehicle positioning, the confidence in the GPS should decrease when GPS failures have occurred recently, because it indicates the vehicle is in an area where it is subjected to multipath effects (an urban canyon for example). Our approach is compared to both iid and Markovian indicator variable models in the Section IV-C where it is shown that our non stationary approach is more robust in case of a change of statistical properties of the indicator variables.

Hue et. al. work [10] is the closest to our approach. They used a prior over the probabilities of the latent state (the data association variable in [10], which is the counterpart of $\mathbf{c}_t$ in this paper), and this prior could evolve. However, its evolution model is defined implicitly, whereas we handle it explicitly with the reliability coefficients $\alpha_{k,t}$. Moreover, in [10], a Gibbs sampler is used to update these quantities, whereas we have a computationally cheaper, and efficient, importance distributions, see simulations below.

[3] As pointed out by a reviewer, a possible statistical structure could be an m-order Markov model where $\Pr(\mathbf{c}_t|\mathbf{c}_{1:t-1}) = \Pr(\mathbf{c}_t|\mathbf{c}_{t-m:t-1})$. As we found no reference using such a model for jump systems, we didn't provide a comparison of our approach with this potentially interesting model.

B. Interpretations and possible extensions

An interesting interpretation of our model is that of *mixture of pdfs*. More precisely, for a sensor k, the switching observation model with latent variable $c_{k,t}$ can also be interpreted as a single observation equation denoted $\overline{p}(\mathbf{z}_{k,t}|\mathbf{x}_t, \boldsymbol{\alpha}_{k,t}, \boldsymbol{\sigma}_t)$ whose pdf is the mixture of the individual sensor state pdfs,

$$\overline{p}(\mathbf{z}_{k,t}|\mathbf{x}_t, \boldsymbol{\alpha}_{k,t}, \boldsymbol{\sigma}_t) \;=\; \sum_{j=0}^{d_k} \alpha_{k,j,t} p(\mathbf{z}_{k,t}|\mathbf{x}_t, c_{k,t} = j, \boldsymbol{\sigma}_t) \tag{36}$$

In this interpretation, the variable $c_{k,t}$ is the mixture latent variable, and the weights $\alpha_{k,j,t}$ evolve with time, as well as the individual pdf parameters $\mathbf{x}_t$ and $\boldsymbol{\sigma}_t$. A direct extension of our algorithm concerns unsupervised sequential classification using time-varying mixtures of Gaussian.

C. Illustrating examples

In this subsection, we present simulations for two scenarios: The first one concerns a widely used academic model with two sensors, one of which switches state. The second example concerns a conditionally linear and Gaussian model where the two sensors may fail.

1) First example: Consider the following nonlinear model from [3], [36], [37]

$$x_{t+1} = \frac{1}{2} x_t + 25 \frac{x_t}{1 + x_t^2} + 8 \cos(1, 2(t+1)) + v_t \tag{37}$$

with $v_t \sim \mathcal{N}(0, 10)$ and $x_0 \sim \mathcal{N}(0, 10)$. We assume that two sensors deliver observations. Sensor $\#1$ has two valid and one faulty states of work, given by the following three observation models

$$z_{1,t} = \begin{cases} x_t + w_{1,0,t} \quad w_{1,0,t} \sim \mathcal{U}([-25, 25]) & \text{if } c_{1,t} = 0 \\ \frac{x_t^2}{20} + w_{1,t} \quad w_{1,1,t} \sim \mathcal{N}(0, 1) & \text{if } c_{1,t} = 1 \\ \frac{(x_t - 10)^2}{20} + w_{1,2,t} \quad w_{1,1,t} \sim \mathcal{N}(0, 3) & \text{if } c_{1,t} = 2 \end{cases} \tag{38}$$

where $\mathcal{U}([a, b])$ is the uniform pdf on the interval $[a, b]$. The second sensor has only one nominal and one faulty states of work corresponding to the following observation models

$$z_{2,t} = \begin{cases} x_t + w_{2,0,t} \quad w_{2,0,t} \sim \mathcal{U}([-25, 25]) & \text{if } c_{2,t} = 0 \\ x_t + w_{2,1,t} \quad w_{2,1,t} \sim \mathcal{N}(0, 2) & \text{if } c_{2,t} = 1 \end{cases} \tag{39}$$

Both sensor are assumed not totally reliable and thus may be faulty. Observations from sensors $\#1$ and $\#2$ are synchronous. Algorithm 2 is applied over 100 time iterations with $N = 500$ particles. The importance distribution for $q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \widetilde{\mathbf{c}}_t^{(i)}, \mathbf{z}_t)$ is chosen to be based on an Extended Kalman Filter step. Sensor $\#1$ follows the second observation model in Eq. (38) for $t \in T_1 \cup T_2$ with $T_1 = [10, 30]$ and $T_2 = [50, 70]$ and is faulty for $t \in T_4$ where $T_4 = [70, 80]$. For other time instants, Sensor $\#1$ follows the first observation model in Eq. (38). Sensor $\#2$ is faulty for $t \in T_3$ where $T_3 = [20, 50]$. These faults are simulated by adding a random offset sampled from $\mathcal{U}([-20, 20])$ to the observations $z_{1,t}$ and $z_{2,t}$.

In Fig.'s 2-6, we have reported the MAP/MMSE estimates at each time t of $c_{1,t}$, $c_{2,t}$, $\boldsymbol{\alpha}_t$ and the error e_t, where

$$e_t \;=\; x_t - \sum_{i=1}^{N} w_t^{(i)} x_t^{(i)} \tag{40}$$

The empirical standard deviation used to compute the 2-sigma bounds in Fig.'s 5 and 9 is computed by

$$\sigma_e = \sqrt{\sum_{i=1}^{N} w_t^{(i)} \left(x_t^{(i)} - \sum_{i=1}^{N} w_t^{(i)} x_t^{(i)} \right)^2} \tag{41}$$

As can be seen, the sensor states are accurately estimated, and the mean squared error is kept low, thanks to the ability to accurately switch from one observation model to another.

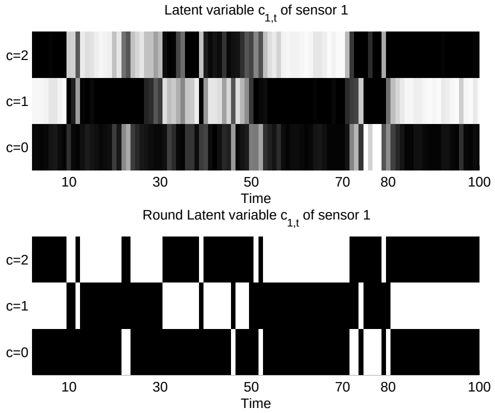

Fig. 2. (Top) Posterior probability of each state $c_{1,t} \in \{0, 1, 2\}$ at each iteration $t = 1, \ldots, 100$. Black corresponds to a zero probability, whereas white corresponds to probability one. (Bottom) MAP estimate of the sensor state $c_{1,t} \in \{0, 1, 2\}$ (white = estimated state). During time intervals $T_1 = [10, 30]$ and $T_2 = [50, 80]$, the true state is $c_{1,t} = 2$, during the time interval $T_4 = [70, 80]$ the true state is $c_{1,t} = 0$, and $c_{1,t} = 1$ for any other time instants. The sensor states are accurately estimated.

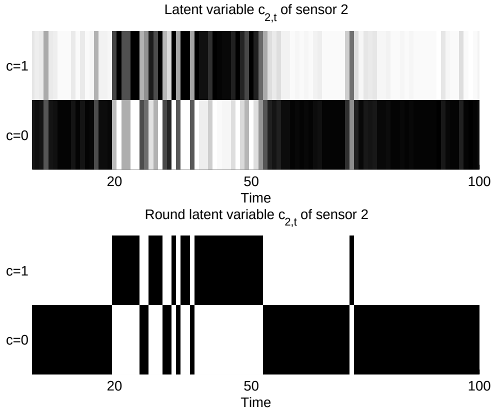

Fig. 3. (Top) Posterior probability of each state $c_{2,t} \in \{0, 1\}$ at each iteration $t = 1, \ldots, 100$. Black corresponds to a zero probability, whereas white corresponds to probability one. (Bottom) MAP estimate of the sensor state $c_{2,t} \in \{0, 1\}$ (white = estimated state). During the time interval $T_3 = [20, 50]$, the sensor if faulty, and it is valid for any other time instant. The sensor state is correctly estimated.

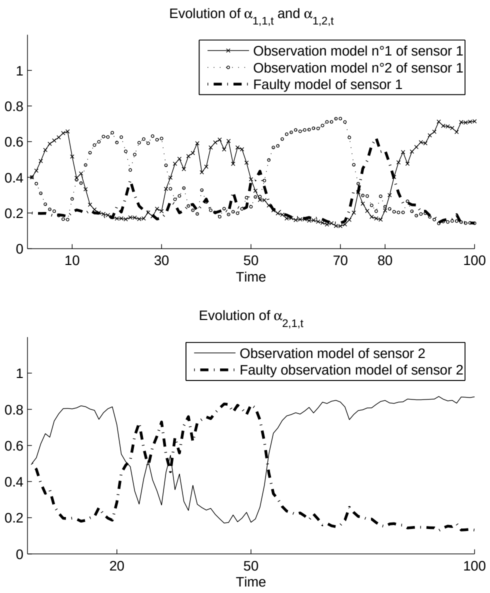

Fig. 4. (Top) MMSE estimate of $\alpha_{1,0,t}$, $\alpha_{1,1,t}$ and $\alpha_{1,2,t}$. During time intervals $T_1 = [10, 30]$ and $T_2 = [50, 80]$, the confidence in state 2 (that is $\alpha_{1,2,t}$) increases, while $\alpha_{1,1,t}$ decreases. The opposite happens outside T_1, T_2 and T_4. Both coefficients $\alpha_{1,1,t}$ and $\alpha_{1,1,t}$ decrease during the interval T_4. During the interval $T_3 = [20, 50]$, the $\alpha_{2,1,t}$ decreases because the sensor is estimated to be faulty. It increases outside of T_3, because the sensor is estimated as valid. (Bottom) MMSE estimate of $\alpha_{2,0,t}$ and $\alpha_{2,1,t}$. During the interval $T_3 = [20, 50]$, the $\alpha_{2,1,t}$ decreases because the sensor is estimated to be faulty. It increases outside of T_3, because the sensor is estimated as valid.

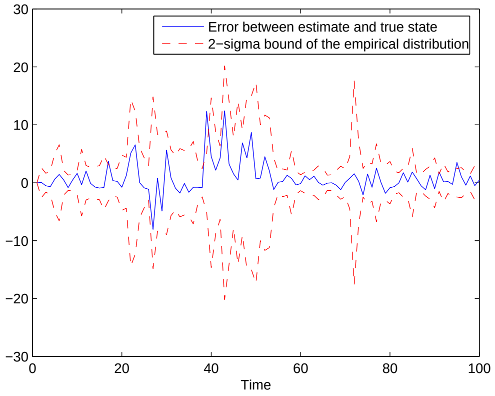

Fig. 5. Evolution of the error e_t computed as in Eq. (40) (solid line). In dashed lines, the 2-sigma bounds are plotted. The 2-sigma bounds increase and decrease, depending on the observation model used (some are less informative than others). In particular, when $t \in T_3 = [20, 50]$, sensor 2 is estimated as faulty, and it does not provide information about the hidden state $\mathbf{x}_t$. Overall, the estimation accuracy is good, even though the sensors models switch.

For this particular example, our approach has been compared to the approach when the prior probabilities of the

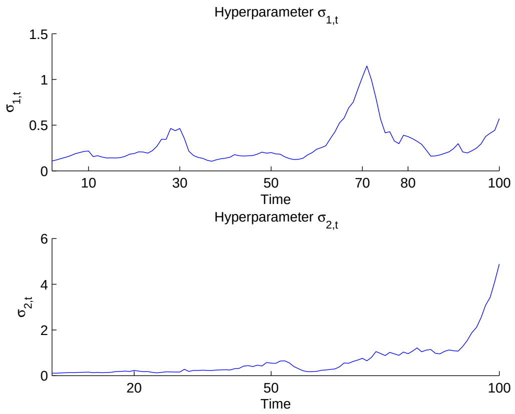

Fig. 6. (Top) Evolution of the hyperparameter $\sigma^{\alpha}_{1,t}$. This hyperparameter tunes the dynamics of the reliability coefficient $\alpha_{1,t}$. More precisely, the value of $\sigma_{1,t}$ decreases at times t=30, 50, 70 80, to allow the reliability coefficients to quickly change. (Bottom) Evolution of the hyperparameter $\sigma^{\alpha}_{2,t}$. This hyperparameter tunes the dynamics of the reliability coefficients $\alpha_{2,t}$.

$\mathbf{c}_t$ are assumed fixed and known. These prior probabilities are set according to the relative frequencies of the states on the interval $[0, 100]$, thus $\boldsymbol{\alpha}'_1 = \left[\begin{array}{ccc} 0.1 & 0.5 & 0.4 \end{array} \right]$ and $\boldsymbol{\alpha}'_2 = \left[\begin{array}{cc} 0.3 & 0.7 \end{array} \right]$ where

$$\Pr(c_{k,t}|c_{k,1:t-1}) = \Pr(c_{k,t}) = \boldsymbol{\alpha}'_k(c_{k,t}+1)$$

The importance function for the $c_{k,t}$'s remain the same as for our approach, except that it priors are then time-invariant and fixed to $\boldsymbol{\alpha}'_k$. Results of this method for the same simulation are shown in Fig.'s 7-9. Similar simulations has been performed 500 times. They showed that our method provides an absolute mean error that is 21% less than the absolute mean error of the fixed prior approach.

2) Second example: Consider the following linear model [7], [14].

$$\mathbf{x}_{t+1} = \left[\begin{array}{cc} 1 & T \\ 0 & 1 \end{array} \right] \mathbf{x}_t + \left[\begin{array}{c} \frac{1}{2}T^2 \\ T \end{array} \right] \gamma_t \qquad (42)$$

with $\gamma_t \sim \mathcal{N}(0, 10)$ and $T = 0.1$ is the sampling period. Two synchronous sensors deliver observations, each of them having one observation model only:

$$z_{1,t} = \left[\begin{array}{cc} 1 & 0 \end{array} \right] \mathbf{x}_t + w_{1,t} \qquad (43)$$

$$z_{2,t} = \left[\begin{array}{cc} 0 & 1 \end{array} \right] \mathbf{x}_t + w_{2,t} \qquad (44)$$

Both sensors are assumed potentially faulty. When the sensors are in their nominal state of work, it is assumed that the noises have pdfs $w_{1,t} \sim \mathcal{N}(0, 2)$ and $w_{2,t} \sim \mathcal{N}(0, 1)$. Whenever these sensors become faulty, the variances are assumed to follow the vague pdfs $w_{1,t} \sim \mathcal{N}(0, 100)$ and $w_{2,t} \sim \mathcal{N}(0, 100)$. This is a conditionally linear and Gaussian model, which can be implemented with the Rao-Blackwellized algorithm similar to Algorithm 4, for the synchronous case, however.

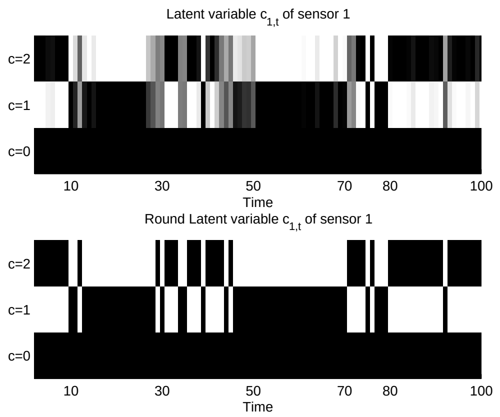

Fig. 7. (Top) Posterior probability of each state $c_{1,t} \in \{0,1,2\}$ at each iteration $t = 1, \ldots, 100$ for the switching model with fixed priors. Black corresponds to a zero probability, whereas white corresponds to probability one. (Bottom) MAP estimate of the sensor state $c_{1,t} \in \{0,1,2\}$ (white = estimated state). The sensor states are less accurately estimated, especially the sensor failure ($c_{1,t} = 0$) during the time interval $T_4 = [70, 80]$

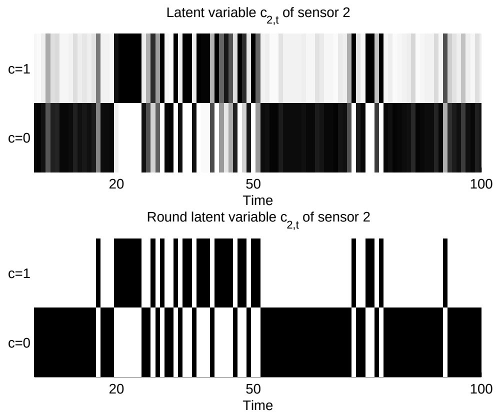

Fig. 8. (Top) Posterior probability of each state $c_{2,t} \in \{0,1\}$ at each iteration $t = 1, \ldots, 100$ for the switching model with fixed priors. Black corresponds to a zero probability, whereas white corresponds to probability one. (Bottom) MAP estimate of the sensor state $c_{2,t} \in \{0,1\}$ (white = estimated state). During the time interval $T_3 = [20, 50]$, the sensor if faulty, and it is valid for any other time instant. The sensor state is less correctly estimated.

We simulated failures of Sensor #1 and #2 with two Markovian models defined by the two transition matrices $\Pi_{[0,100]} = \begin{bmatrix} .85 & .15 \\ .15 & .85 \end{bmatrix}$ on the interval $T_1 = [0, 100]$ and $\Pi_{[100,200]} = \begin{bmatrix} .15 & .85 \\ .85 & .15 \end{bmatrix}$ on the interval $T_2 = [100, 200]$, where

$$\Pr(c_{k,t} = j | c_{k,t-1} = i) = \pi_{i+1,j+1}$$

with $\Pi = (\pi_{i,j})$. The algorithm is run over 200 iterations with $N = 500$ particles. Fig. 10 displays the evolution of

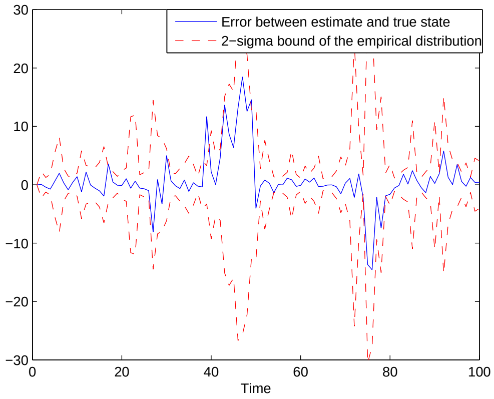

Fig. 9. Evolution of the error e_t computed as in Eq. (40) (solid line) for the switching model with fixed priors. In dashed lines, the 2-sigma bounds are plotted. The 2-sigma bounds increase and decrease, depending on the observation model used (some are less informative than others). In particular, when $t \in T_3 = [20, 50]$, sensor 2 is estimated as faulty, and it does not provide information about the hidden state $\mathbf{x}_t$. The absolute mean error is 20% higher than the error with our model.

the two components $\mathbf{x}_t(1)$ and $\mathbf{x}_t(2)$ of $\mathbf{x}_t$, as well as the observations $z_1(t)$ and $z_2(t)$. The MAP/MMSE estimates of $\mathbf{c}_t$, $\boldsymbol{\alpha}_t$ and $\boldsymbol{\sigma}_t$ are displayed in Fig.'s 11–13, where the errors of position and velocity are defined by Eq. 40 for each component of the vector $\mathbf{x}_t$.

As can be seen, the true sensor states are estimated very accurately, in spite of sensor failures. Simulations performed with the same observations and a standard model (that is, the same model as in Eq.'s (42)–(44), without defining a faulty sensor state) show a much higher error each time the sensors fail (for the sake of brevity this simulations are not reported here ; see also the results in the following section).

Our model has been compared with a Jump Markov Linear System. The transition matrix of the JMLS is set to $\Pi = \Pi_{[0,100]} = \begin{bmatrix} .85 & .15 \\ .15 & .85 \end{bmatrix}$. The algorithm employed is the optimal Rao-Blackwellised particle filter algorithm provided by Doucet et al. [27]. Simulations have been run 500 times and the absolute mean position errors are given in Tab. I for both models.

TABLE I

COMPARISON OF OUR MODEL WITH A JMLS

Model/Interval	T_1	T_2	$T_1 \cup T_2$
Abs. mean error of our model	0.6767	0.6241	0.6504
Abs. mean error of the JMLS	0.6149	0.7870	0.7070
Improvement	−10.05%	20.70%	7.21%

The JMLS model provides better results on the first interval where the indicator variables are sampled according to the fixed transition matrix Π, which was expected as the data are generated according to the evolution model.

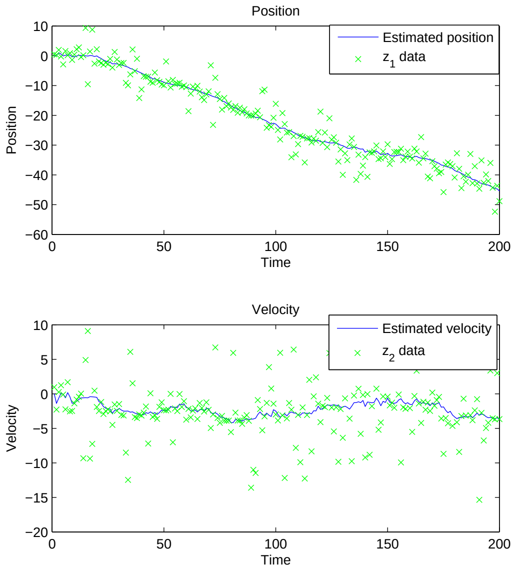

Fig. 10. (Top) Evolution of the MMSE estimate of the first component $\mathbf{x}_t(1)$ of $\mathbf{x}_t$ as well as the observation $z_1(t)$ provided by sensor $\#1$. (Bottom) Evolution of the MMSE estimate of the first component $\mathbf{x}_t(2)$ of $\mathbf{x}_t$ as well as the observation $z_2(t)$ provided by sensor $\#2$.

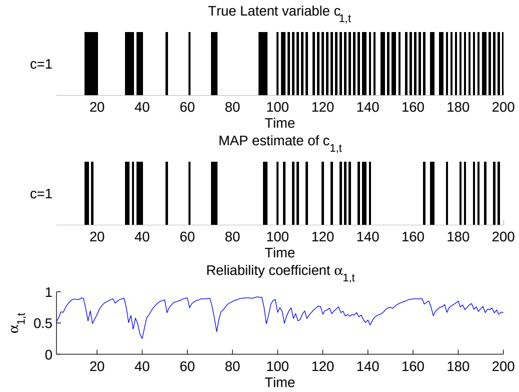

Fig. 11. (Top) True sensor state $c_{1,t}$. Its evolution model follows the Markov transition matrix $\Pi_{[}0, 100]$ on the interval $T_1 = [0, 100]$ and the Markov transition matrix $\Pi_{[}100, 200]$ on the interval $T_2 = [100, 200]$. (Middle) MAP estimate of the the nominal sensor state $c_{1,t} = 1$ (white = estimated state). (Bottom) MMSE estimate of the reliability coefficient $\alpha_{1,1,t}$. This coefficient decreases when the sensor is faulty.

However, on the second interval T_2, our model outperforms the JMLS model, although it doesn't define a Markov structure on the indicator variables. As it is more realistic to consider that the dynamics of the indicator variables may evolve with time, our algorithm provides a more robust estimation algorithm.

In the following section, we present an application of our framework to wheel land vehicle positioning.

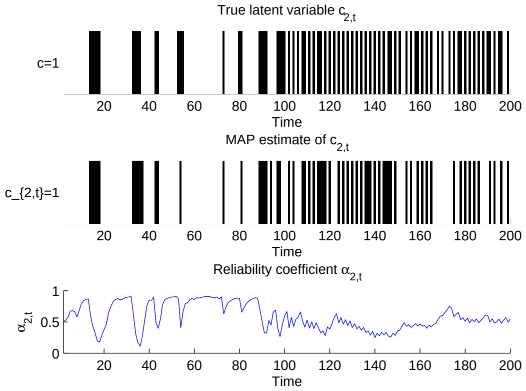

Fig. 12. (Top) True sensor state $c_{2,t}$. Its evolution model follows the Markov transition matrix $\Pi_{[}0,100]$ on the interval $T_1 = [0,100]$ and the Markov transition matrix $\Pi_{[}100,200]$ on the interval $T_2 = [100,200]$. (Middle) MAP estimate of the the nominal sensor state $c_{2,t} = 1$ (white = estimated state). (Bottom) MMSE estimate of the reliability coefficient $\alpha_{2,1,t}$. This coefficient decreases when the sensor is faulty.

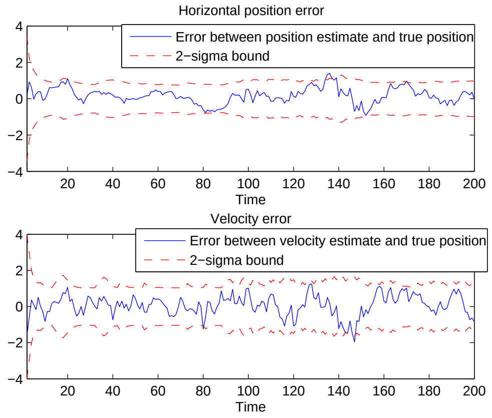

Fig. 13. (Top) Evolution of the position error (solid line). In dashed lines, the 2-sigma bounds are plotted. The 2-sigma bounds increase and decrease, depending on the observation model used (some are less informative than others). (Bottom) Evolution of the velocity error (solid line). In dashed lines, the 2-sigma bounds are plotted. Overall, the estimation accuracy is good, even though the sensors models switch.

V. Application to multisensor fusion for land vehicle positioning

The framework and algorithms presented above are well adapted to Mobile Robotics problems. Developing autonomous vehicles such as drones or automated taxi cabs, is quite important for goods/people transportation or operation in hostile environments (martian exploration, land mines removal). In order to ensure high accuracy positioning and external environment mapping, many sensors, possibly redundant, are required. Typical positioning sensors are GPS, inertial units, magnetic compasses. For obstacle detection, sensors may be radars or cameras. An important characteristics of mobile robotics is that the observations delivered by some sensors may be easily

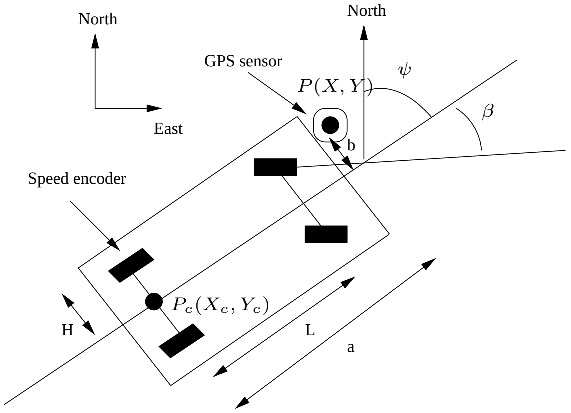

Fig. 14. Nomenclature of the vehicle. ψ is the angle between true North and longitudinal axis of the land vehicle and β is the steering angle. Position is calculated at point $P(X,Y)$. Speed encoder is situated on the left rear wheel and GPS sensor on the front right side of the vehicle. Position of the vehicle is calculated in the North-East coordinate frame.

blurred because of hard external conditions. For example, a camera lens may become dirty because of a sand storm on Mars, and a radar may have additional noise due to snowfalls. Other sensors, like GPS are subject to multipath phenomena in urban canyons and magnetic compass are sensitive to parasite electromagnetic fields: they may deliver totally erroneous measurements in some conditions.

Several previous works apply sequential particle filtering to land vehicle positioning. Yang et al. [38] made a comparative comparison between EKF and particle filter, showing the interest of the latter for the problem considered. Giremus et al. [39] and Gustafsson et al. [16], [17] used a Rao-Blackwellized bootstrap filter. Giremus et al. [40] proposed a fixed-lag Rao-Blackwellised particle filter to estimate multipath biases associated to the GPS measurements. They used a switching observation model whose indicator variables have fixed priors.

In this section, we address the positioning of a land vehicle with four wheels, two of which may steer. The state evolution model is presented in the next subsection.

A. State model based on land vehicle kinematics

The land vehicle considered is depicted in Fig. 14. Its kinematic model is inspired from [41] and [42]. The differential equations governing the kinematics of the vehicle at point P_c are based on an Ackerman steering model of simplified bicycle model (the variables are presented in Tab. II, here taken at point C):

$$\begin{cases} \dot{X}_c(t) = V_c(t)\cos\psi(t) \\ \dot{Y}_c(t) = V_c(t)\sin\psi(t) \\ \dot{\psi}(t) = \frac{V_c(t)}{L}\tan\beta(t) \end{cases}$$

Let $P(X,Y)$ be the position of the vehicle at the GPS antenna (point P). At point P, X and Y are expressed

TABLE II

STATE VARIABLES USED IN THE LAND WHEEL VEHICLE MODEL.

X, Y	North and East positions of the vehicle at point P
V	Speed of the vehicle at point P_C
ψ	Heading of the vehicle
β	Steering angle
T	Sampling time

as functions of X_c and Y_c by

$$X(t) = X_c(t) + a\cos\psi(t) + b\sin\psi(t)$$
$$Y(t) = Y_c(t) + a\sin\psi(t) - b\cos\psi(t)$$

We adopt the following discrete first order approximation (where we note $V_t = V_c(t)$)

$$\begin{aligned}
X_{t+1} &= X_t + T\left[V_t\cos\psi_t + (-a\sin\psi_t + b\cos\psi_t)\frac{V_t}{L}\tan\beta_t\right]\\
Y_{t+1} &= Y_t + T\left[V_t\sin\psi_t + (a\cos\psi_t + b\sin\psi_t)\frac{V_t}{L}\tan\beta_t\right]\\
V_{t+1} &= V_t + T\dot{V}_t\\
\psi_{t+1} &= \psi_t + T\frac{V_t}{L}\tan(\beta_t)\\
\beta_{t+1} &= \beta_t + T\dot{\beta}_t\\
\dot{\beta}_{t+1} &= \dot{\beta}_t + T\ddot{\beta}_t
\end{aligned}$$

These equations may be rewritten as the following nonlinear model

$$\mathbf{x}_{t+1} \;=\; f_1(\mathbf{x}_t) + G_t \cdot \mathbf{v}_t \tag{45}$$

where f_1 is the state transition function, G_t is the noise transfer matrix, the state vector is $\mathbf{x}_t = [X_t,\ Y_t,\ V_t,\ \psi_t,\ \beta_t,\ \dot{\beta}_t,]^T$ and $\mathbf{v}_t = [\dot{V}_t,\ \ddot{\beta}_t]^T$. The state noise $\mathbf{v}_t = [\dot{V}_t,\ \ddot{\beta}_t]^T$ is assumed zero mean white Gaussian with known covariance matrix $Q_t = E[\mathbf{v}_t\mathbf{v}_t^T]$.

B. *Observation models*

The vehicle is equipped with three sensors, see Fig. 14. These sensors are:

- A centimeter-level precision DGPS receiver, providing North and East position of the vehicle
- A speed sensor
- A steering sensor

Speed and steering measurements are synchronous. However, the GPS measurements are asynchronous w.r.t. these other two sensors. The GPS measurements frequency is low (5 Hz) when compared to the Speed/Steering measurement frequency (40 Hz). We detail below the sensor observation equations.

1) GPS sensor model: The observation model for the GPS is given by

$$\mathbf{z}_{1,t} = h_1(\mathbf{x}_t) + \mathbf{w}_{1,t} = \begin{pmatrix} X_t \\ Y_t \end{pmatrix} + \mathbf{w}_{1,t}$$

with $\mathbf{w}_{1,t}$ a zero mean white Gaussian noise of known covariance matrix $R_{1,t} = \begin{bmatrix} 0.002 & 0 \\ 0 & 0.002 \end{bmatrix}$. When the sensor is non-faulty, the likelihood is written as $p(\mathbf{z}_{1,t}|\mathbf{x}_t) = \mathcal{N}(h_1(\mathbf{x}_t), R_{1,t})$.

The GPS receiver cannot be assumed fully reliable due to multipath, diffraction and mask phenomena likely to alter the measurements, especially in urban areas [43]. It is supposed that the GPS sensor has two classes of work:

- A nominal state of work corresponding to the measurement model defined above
- A faulty state of work when data are corrupted by multipath. In that case we define a vague pdf $p_0(\mathbf{z}_{1,t})$, uniform on a large interval centered about the last position estimate

This is thus a sensor for which the discrete variable $c_{1,t}$ is in $\{0, 1\}$ (binary case). The evolution pdf for $\boldsymbol{\alpha}_{1,t}$ is that of Eq. (17).

2) Speed and Steering sensors: The observation models of the speed and steering sensors are respectively

$$z_{2,t} = h_2(\mathbf{x}_t) + w_{2,t} = (1 + \tan(\beta_t)\frac{H}{L})v_t + w_{2,t}$$

and

$$z_{3,t} = h_3(\mathbf{x}_t) + w_{3,t} = \beta_t + w_{3,t}$$

with $w_{2,t}$ and $w_{3,t}$ zero mean white Gaussian noises of assumed known covariances matrix $R_{2,t} = 0.1$ and $R_{3,t} = 0.002$. These sensors are not supposed to be faulty.

C. Estimation objectives and algorithm

Given the model and assumptions about the sensors, the estimation objective is as follows: each time the speed/steering measurements are collected, we estimate $\mathbf{x}_t$ by MMSE. Whenever a GPS measurement is collected, we estimate $\mathbf{x}_t$, $\boldsymbol{\alpha}_{1,t}$ and $\sigma^{\alpha}_{1,t}$ by MMSE and $c_{1,t}$ by MAP.

In this model, components of the state evolution noise have a negligible variance (almost deterministic evolution). This makes the design of an efficient importance distribution $q(\mathbf{x}_t|\mathbf{x}^{(i)}_{t-1}, \widetilde{\mathbf{c}}^{(i)}_t, \mathbf{z}_t)$ quite difficult: on the one hand, using the state evolution pdf $p(\mathbf{x}_t|\mathbf{x}^{(i)}_{t-1})$ as $q(\mathbf{x}_t|\mathbf{x}^{(i)}_{t-1}, \widetilde{\mathbf{c}}^{(i)}_t, \mathbf{z}_t)$ (bootstrap filter choice) is not efficient. On the other hand, UKF and EKF based approaches cannot be applied because the evolution model is highly informative and the particle states $\widetilde{\mathbf{x}}^{(i)}_t$ sampled from such importance pdfs are very likely to have very small weights.

Here, however, the posterior distribution $p(\mathbf{x}_t, \mathbf{c}_t, \boldsymbol{\alpha}_t, \boldsymbol{\sigma}_t|\mathbf{z}_{1:t})$ is almost always monomodal and little skewed. We propose to use an approximate particle filtering approach, whose computational load is reduced, and which is very accurate. This approach, proposed in [44] under the name of Rao-Blackwellised UKF, is inspired by the Rao-Blackwellized particle filter, where the Kalman filters are replaced by Unscented Kalman filters [8], [45], [46], [11]. Here, the state $\mathbf{x}_t$ is estimated by a bank of UKFs, whereas the other parts of the full state (that is, $\mathbf{c}_t$, $\boldsymbol{\alpha}_t$, $\boldsymbol{\sigma}_t$)

are updated using the importance distributions described in Section III, for variables $c_{1:t}, \alpha_{0:t}, \sigma_{0:t}$ are as explained in Section III. The algorithm is presented below.

Algorithm 5: Rao-Blackwellized UKF particle filter

for switching observation models – Asynchronous case

```
% Step 5.1 Initialisation
• For particles $i = 1,..,N$ sample $\widehat{\mathbf{x}}_{0|0}^{(i)} \sim p_0(\widehat{\mathbf{x}}_{0|0})$
• For particles $i = 1,..,N$, sample $\Sigma_{0|0}^{(i)} \sim p_0(\Sigma_{0|0})$
• For particles $i = 1,..,N$, sample $\sigma_{1,0}^{\alpha\ (i)} \sim p_0(\sigma_{1,0}^{\alpha})$
• For particles $i = 1,..,N$, sample $\boldsymbol{\alpha}_{1,0}^{(i)} \sim p_0(\boldsymbol{\alpha}_{1,0}|\sigma_{1,0}^{(i)})$
• Set the initial weights $w_0^{(i)} \leftarrow \frac{1}{N}$
% Step 5.2 Iterations
• For t=1,2,... do
    – Await the arrival of new measure $z_{k,t}$, delivered by sensor $\#k$, $k = 1, 2, 3$ and, for particles $i = 1,..,N$, do
    % If arrival of measures $z_{2,t}$ and $z_{3,t}$
        ∗ Update mean $\widehat{x}_{t|t}^{(i)}$ and covariance matrix $\Sigma_{t|t}^{(i)}$ with a Unscented Kalman filter step, that is $(\widehat{x}_{t|t}^{(i)}, \Sigma_{t|t}^{(i)}) = \text{UKF}(\widehat{x}_{t-1|t-1}^{(i)}, \Sigma_{t-1|t-1}^{(i)}, z_{2,t}, z_{3,t})$
        ∗ update of variables related to the GPS sensor
        ∗ set $c_{1,t}^{(i)} \leftarrow c_{1,t-1}^{(i)}$
        ∗ set $\boldsymbol{\alpha}_{1,t}^{(i)} \leftarrow \boldsymbol{\alpha}_{1,t-1}^{(i)}$
        ∗ set $\sigma_{1,t}^{\alpha\ (i)} \leftarrow \sigma_{1,t-1}^{\alpha\ (i)}$
    – For particles $i = 1,..,N$, update the weights
          $\widetilde{w}_t^{(i)} = w_{t-1}^{(i)} p(z_{2,t}|\mathbf{z}_{1:t-1}) p(z_{3,t}|\mathbf{z}_{1:t-1})$
    % If arrival of a GPS measure $\mathbf{z}_{1,t}$
        ∗ Sample the sensor state variable $\widetilde{c}_{1,t}^{(i)} \sim q(c_{1,t}|\widehat{\mathbf{x}}_{t-1|t-1}^{(i)}, \Sigma_{t-1|t-1}^{(i)}, \alpha_{1,t-1}^{(i)}, \mathbf{z}_{1,t})$
        ∗ Sample the probabilities $\widetilde{\boldsymbol{\alpha}}_{1,t}^{(i)} \sim q(\boldsymbol{\alpha}_{1,t}|\boldsymbol{\alpha}_{1,t-1}^{(i)}, \widetilde{c}_{1,t}^{(i)}, \sigma_{1,t-1}^{\alpha\ (i)})$
        ∗ Sample the hyperparameter vector $\widetilde{\sigma}_{1,t}^{\alpha\ (i)} \sim q(\sigma_{1,t}^{\alpha}|\sigma_{1,t-1}^{\alpha\ (i)}, \widetilde{\boldsymbol{\alpha}}_{1,t}^{(i)}, \boldsymbol{\alpha}_{1,t-1}^{(i)})$
        ∗ Update mean $\widehat{x}_{t|t}^{(i)}$ and covariance matrix $\Sigma_{t|t}^{(i)}$ with a Unscented Kalman filter step, that is $(\widehat{x}_{t|t}^{(i)}, \Sigma_{t|t}^{(i)}) = \text{UKF}(\widehat{x}_{t-1|t-1}^{(i)}, \Sigma_{t-1|t-1}^{(i)}, \widetilde{c}_{1,t}^{(i)}, \mathbf{z}_{1,t})$
    – For particles $i = 1,..,N$, update the weights
          $\widetilde{w}_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(\mathbf{z}_{1,t}|\widetilde{c}_{1,1:t}^{(i)}, \mathbf{z}_{1:t-1}) p(\widetilde{c}_{1,t}^{(i)}|\widetilde{\boldsymbol{\alpha}}_{1,t}^{(i)})}{q(\widetilde{c}_{1,t}^{(i)}|\widehat{\mathbf{x}}_{t-1|t-1}^{(i)}, \Sigma_{t-1|t-1}^{(i)}, \alpha_{1,t-1}^{(i)}, \mathbf{z}_{1,t})}$
          $\times \frac{p(\widetilde{\boldsymbol{\alpha}}_{1,t}^{(i)}|\boldsymbol{\alpha}_{1,t-1}^{(i)}, \widetilde{\sigma}_{1,t}^{\alpha\ (i)}) p(\widetilde{\sigma}_{1,t}^{\alpha\ (i)}|\sigma_{1,t-1}^{\alpha\ (i)})}{q(\widetilde{\boldsymbol{\alpha}}_{1,t}^{(i)}|\boldsymbol{\alpha}_{1,t-1}^{(i)}, \widetilde{c}_{1,t}^{(i)}, \sigma_{1,t-1}^{\alpha\ (i)}) q(\widetilde{\sigma}_{1,t}^{\alpha\ (i)}|\sigma_{1,t-1}^{\alpha\ (i)}, \widetilde{\boldsymbol{\alpha}}_{1,t}^{(i)}, \boldsymbol{\alpha}_{1,t-1}^{(i)})}$          (46)
```

- Normalize the weights so that $\sum_{i=1}^{N}\widetilde{w}_t^{(i)}=1$

% `Step 5.3` Resampling

- Compute N_{eff} as in Eq. (10) and perform particle resampling whenever $N_{\text{eff}} < \eta$, see Step 1.3 in Algorithm 1.

D. Results

Algorithm 5 is applied to real sensor measurements. These observations have been collected by the Australian Center for Field Robotics (ACFR)[4]. These very clean data were not altered by the standard observations errors. Two types of errors have been added to GPS signal to test the algorithm:

- Simulation of multipath and diffraction, by addition of a piecewise constant random offset to the real, non altered data, see Fig. 15. Multipath are simulated at times in $T_1 = [50, 60]$, $T_2 = [62, 68]$ and $T_3 = [75, 80]$ (time instants corresponds to GPS data indexes, i.e., it is increased by $t \leftarrow t+1$ each time a GPS observation is collected).
- Simulation of side-slipping of the vehicle, simulated by replacing the real steering angle observations by erroneous ones, over the interval $T_4 = [12, 14]$ (see Fig. 16).

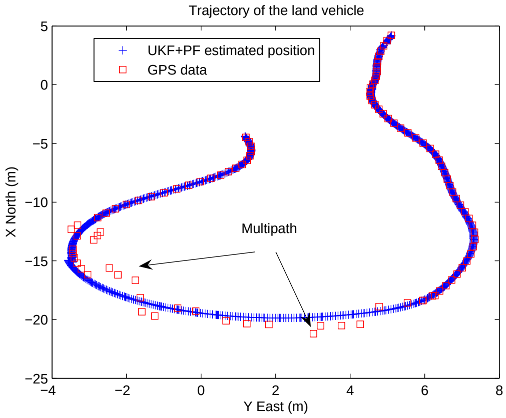

Fig. 15. MMSE Estimation of the trajectory of the vehicle obtained using our algorithm. Squares represent the GPS data. Both are plotted in the North-East coordinate frame. GPS failures occur due to multipath effects, these have been simulated by adding a piecewise constant random bias to the true GPS observation.

The state estimation results of Algorithm 5 have been compared to those obtained by two Unscented Kalman filters. In the first UKF, the ability to detect sensor failures is not implemented. In the second UKF, sensor failures are detected from the normalized quadratic innovation, as detailed in Appendix A.

[4] These data may be found at the Internet address `http://www.acfr.usyd.edu.au/homepages/academic/enebot/dataset.htm`.

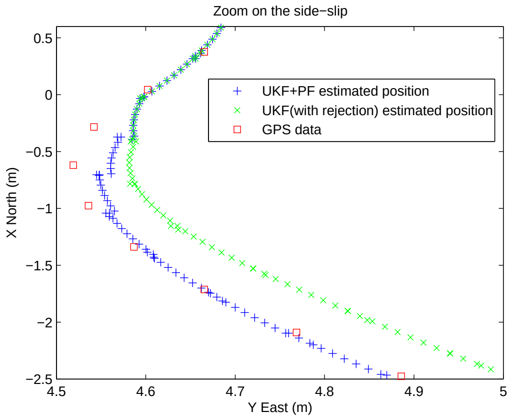

Fig. 16. MMSE Estimation of the trajectory of the vehicle, obtained with our algorithm and with a UKF algorithm which rejects invalid measures. The segment of the trajectory represented here is a magnified view of Fig. 15. Vehicle side-slip is simulated by adding a random offset to the steering measurement angle. In this case, the UKF algorithm rejects GPS data and fails in tracking again the true vehicle trajectory. Our algorithm first rejects GPS data, but comes back to the true trajectory.

Fig. 17 represents the evolution of the posterior probability of $c_{1,t}$ for the GPS sensor. It also shows the MMSE estimate of $\boldsymbol{\alpha}_{1,t}$ (which is here interpreted as a reliability coefficient). During intervals T_1, T_2 and T_3, our algorithm succeeds at detecting GPS failure and $c_{1,t}$ is estimated close to 0. The parameter $\boldsymbol{\alpha}_{1,t}$ drops during these intervals from 80% to 40% GPS reliability.

Fig. 18 compares the positioning of our algorithm and that of the first UKF (which does not enable GPS failure detection). Of course, this UKF performs poorly whenever erroneous measurements arrive, because they are integrated in the fusion process, resulting in large errors during time intervals T_1, T_2 and T_3. Our algorithm, on the other hand, keeps good performance, though the error increases slightly over T_1, T_2 and T_3. The error increase is due to the lack of GPS information, and not to the algorithm itself.

Fig. 19 is similar to Fig. 18. In Fig. 19, however, our algorithm is compared to the UKF with erroneous observations detection, see Appendix A. The latter algorithm considers the GPS measurements as erroneous (steering measurements are assumed reliable), however, it fails at coming back to the true trajectory. This can be explained as follows: due to side-slip, GPS measurements disagree with the evolution model during time interval T_4 and they are rejected by the UKF algorithm. As a consequence, the estimated vehicle trajectory diverges from the true one as time increases, involving rejection of all the following GPS measurements: the UKF algorithm assumes wrongly that the GPS is failing. The same phenomenon happens in case of GPS errors due to multipath problems in time intervals T_1, T_2 and T_3 (see Fig. 19). On the contrary, our algorithm tests all the sensor valid/failure hypotheses. For example, at time $t = 12$, the most likely hypothesis is ”the GPS is faulty”, but at $t = 13$, the most likely hypothesis becomes ”the GPS is valid” because the new GPS observation is coherent with the evolution model. This explains that the position estimated by our algorithm follows the true trajectory (see Fig. 16).

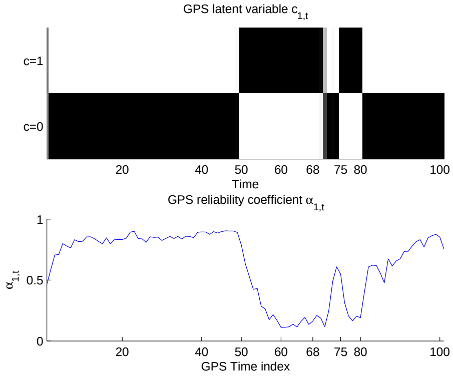

Fig. 17. (Top) Posterior probability of the GPS state $c_{1,t} = 1$. During time intervals $T_1 = [50, 60]$, $T_2 = [62, 68]$ and $T_3 = [75, 80]$, the true state of GPS sensor is $c_{1,t} = 0$ (faulty), and $c_{1,t} = 1$ for any other time instants. The sensor states are accurately estimated. (Bottom) MMSE estimate of the GPS reliability coefficient $\alpha_{1,1,t}$. During the intervals $T_1 = [50, 60]$, $T_2 = [62, 68]$ and $T_3 = [75, 80]$, GPS reliability coefficient $\alpha_{1,1,t}$ decreases because the GPS is detected to be faulty. Outside of these intervals, GPS reliability coefficient increases because GPS is detected to be valid.

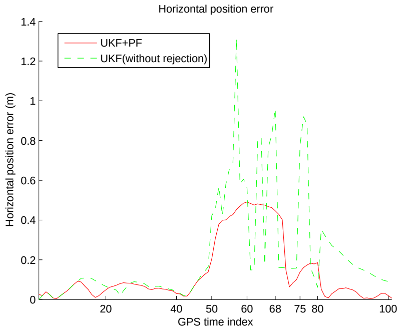

Fig. 18. Comparison of the mean square positioning error between our algorithm and UKF algorithm (without detection of sensor failures). UKF algorithm gives large errors during time intervals $T_1 = [50, 60]$, $T_2 = [62, 68]$ and $T_3 = [75, 80]$ (GPS failures) because it uses these data into the fusion algorithm, as if they were reliable. Our algorithm largely outperforms UKF algorithm.

VI. Conclusion and perspectives

In this paper, we have proposed new model and algorithm for sensor failure detection and, more generally, to switching observation models. We propose a family of algorithms for situations where the sensors deliver measurements in a synchronous way, or in an asynchronous way. We proposed an alternative interpretation of our model in terms of mixture of pdfs. Simulation results for two academic problems, and for a real wheel land vehicle positioning problem show the interest of our model, and the efficiency of the proposed particle filter algorithms. Extensions of this work will concern sequential unsupervised data classification. Further applications are in chemical processes monitoring and visual tracking.

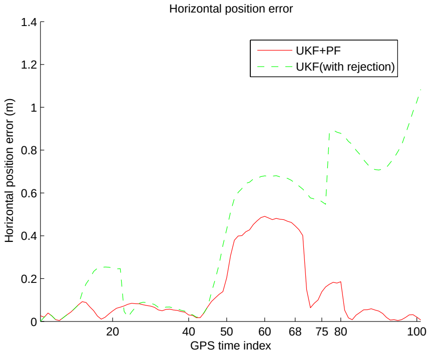

Fig. 19. Comparison of the mean square positioning error between our algorithm and UKF algorithm (with detection of sensor failures). First drift of UKF estimated position ($t = 12$) is due to the rejection of GPS data when side-slip occurs. Our algorithm better performs in this case and also in presence of GPS sensor failures during time intervals $T_1 = [50, 60]$, $T_2 = [62, 68]$ and $T_3 = [75, 80]$: UKF algorithm first rejects GPS data, but UKF estimate quickly drift and is then unable to follow the good trajectory.

Acknowledgement

We would like to sincerely thank the anonymous reviewers for the very accurate comments and recommendations that helped us to improve the overall quality of this article, and the Australian Center for Field Robotics (ACFR) for providing their data online. This work is partially supported by the Centre National de la Recherche Scientifique (CNRS) and the Région Nord-Pas de Calais.

References

[1] N. Gordon, D. Salmond, and A. Smith, “A novel approach to nonlinear/non-Gaussian Bayesian state estimation,” in *Proc. Inst. Elect. Eng. Radar Signal Process.*, vol. 140, 1993, pp. 107–113.

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

[3] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in practice*. Springer-Verlag, 2001.

[4] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[5] A. Doucet and X. Wang, “Monte Carlo methods for signal processing: a review in the statistical signal processing context,” *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 152–170, 2005.

[6] D. Crisan and A. Doucet, “A survey of convergence results on particle filtering for practitioners,” *IEEE Transactions on signal processing*, vol. 50, no. 3, 2002.

[7] Y. Bar-Shalom, X. R. Li, and T. Kirubajan, *Estimation with applications to tracking and navigation*. Editions Wiley-Interscience, 2001.

[8] S. Julier and J. Uhlmann, “A new extension of the Kalman filter to nonlinear systems,” in *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL, 1997*, 1997.

[9] C. Andrieu, M. Davy, and A. Doucet, “Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions.” *IEEE Transactions on signal processing*, vol. 51, no. 7, 2003.

[10] C. Hue, J.-P. Le Cadre, and P. Pérez, “Sequential Monte Carlo methods for multiple target tracking and data fusion,” *IEEE Transactions on signal processing*, vol. 50, no. 2, 2002.

[11] R. V. der Merwe, A. Doucet, N. de Greitas, and E. Wan, “The unscented particle filter,” in *Advances in Neural Information Processing Systems (NIPS13)*, 2000.

[12] C. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York, USA: Springer, 2000.

[13] N. de Freitas, “Rao-Blackwellised particle filtering for fault diagnosis,” in *Proc. Aerospace Conference*, 2002.

[14] T. Schön, F. Gustafsson, and P.-J. Nordlund, “Marginalized particle filters for mixed linear/nonlinear state-space models,” *IEEE Transactions on Signal Processing*, vol. 53, no. 7, 2005.

[15] R. Chen and J. Liu, “Mixture Kalman filters,” *J. R. Statist. Soc. B*, vol. 62, no. 3, pp. 493–508, 2000.

[16] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, , and P.-J. Nordlund, “Particle filters for positioning, navigation and tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, 2002.

[17] P.-J. Nordlund and F. Gustafsson, “Sequential Monte Carlo filtering techniques applied to integrated navigation systems,” in *American control conference*, Arlington, VA, 2001.

[18] Y. Wu, G. Hua, and T. Yu, “Switching observation models for contour tracking in clutter,” in *Proc. Aerospace Conference*, 2002.

[19] G. Ackerson and K. Fu, “On state estimation in switching environments,” *IEEE Transactions on Automatic Control*, vol. 15, pp. 10–17, 1970.

[20] H. Akashi and H. Kumamoto, “Random sampling approach to state estimation in switching environments,” *Automatica*, vol. 13, pp. 429–434, 1977.

[21] J. Tugnait, “Adaptive estimation and identification for discrete systems with Markov jump parameters,” *IEEE Transactions on Automatic Control*, vol. 27, no. 5, pp. 1054–1065, 1982.

[22] H. Blom and Y. Bar-Shalom, “The interacting multiple model algorithm for systems with Markovian switching coefficients,” *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.

[23] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, “Interacting multiple model methods in target tracking: a survey,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 103–123, 1998.

[24] Y. Bar-Shalom and X. R. Li, *Multitarget-multisensor tracking: principles and techniques*. YBS publishing, 1995.

[25] A. Doucet and B. Ristic, “Recursive state estimation for multiple switching models with unknown transition probabilities,” *IEEE Transactions on Automatic Control*, vol. 38, no. 3, pp. 1098–1104, 2002.

[26] V. Jilkov and X. R. Li, “Online Bayesian estimation of transition probabilities for Markovian jump systems,” *IEEE Transactions on Signal Processing*, vol. 52, no. 6, pp. 1620–1630, 2004.

[27] A. Doucet, N. Gordon, and V. Krishnamurthy, “Particle filters for state estimation of jump Markov linear systems,” *IEEE Transactions on Signal Processing*, vol. 49, no. 3, pp. 613–624, 2001.

[28] R. Chen, X. Wang, and J. Liu, “Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering,” *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 2079–2094, 2000.

[29] X. Wang, R. Chen, and D. Guo, “Delayed-pilot sampling for mixture Kalman filter with application in fading channels,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 241–254, 2002.

[30] S. Särkkä, A. Vehtari, and J. Lampinen, “Rao-Blackwellized Monte Carlo data association for multiple target tracking,” in *Proceedings of the seventh conference on Information Fusion*, Stockholm, Sweden, 2004.

[31] A. Doucet, B. Vo, C. Andrieu, and M. Davy, “Particle filtering for multi-target tracking and sensor management,” in *Proceedings of the fifth conference on Information Fusion*, vol. 1, Annapolis, Maryland, USA, 2002, pp. 474–481.

[32] J. Vermaak, S. Godsill, and P. Pérez, “Monte Carlo filtering for multi-target tracking and data association,” *IEEE Transactions on Aerospace and Electronic systems*, vol. 41, no. 1, pp. 309–332, 2005.

[33] V. Verma, G. Gordon, R. Simmons, and S. Thrun, “Particle filters for rover fault diagnosis,” *IEEE Robotics & Automation Magazine special issue on Human Centered Robotics and Dependability*, 2004.

[34] J. Flores-Quintanilla, R. Morales-Menendez, R. Ramirez-Mendoza, L. Garza-Castanon, and F. Cantu-Ortiz, “Toward a new fault diagnosis system for electric machines based on dynamic probabilistic models,” in *American Control Conference*, Portland, USA, 2005.

[35] E. Baziw, “Real-time seismic signal enhancement utilizing a hybrid Rao-Blackwellized particle filter and hidden Markov model filter,” *IEEE Geoscience and Remote Sensing Letters*, vol. 2, no. 4, pp. 418–422, 2005.

[36] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, 2002.

[37] B. Carlin, N. Polson, and D. Stoffer, “A Monte Carlo approach to nonnormal and nonlinear state-space modeling,” *Journal of the American Statistical Association*, vol. 87, pp. 493–500, 1992.

[38] N. Yang, W. F. Tian, Z. H. Jin, and C. B. Zhang, “Particle filter for sensor fusion in a land vehicle navigation system,” *Measurement science and technology*, vol. 16, pp. 677–681, 2004.

[39] A. Giremus, A. Doucet, V. Calmettes, and J.-Y. Tourneret, “A Rao-Blackwellized particle filter for INS/GPS integration,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’04)*, Montreal, Canada, 2001.

[40] A. Giremus and J.-Y. Tourneret, “Joint detection/estimation of multipath effects for the global positioning system,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’05)*, Philadelphia, USA, 2005.

[41] J. Guivant, F. Masson, and E. Nebot, “Simultaneous localization and map building using natural features and absolute information,” *International journal of Robotics and Autonomous Systems*, vol. 40, pp. 79–90, 2002.

[42] J. Guivant, E. Nebot, and S. Baiker, “Autonomous navigation and map building using laser range sensors in outdoor applications,” *Journal of Robotic Systems*, vol. 17, no. 10, pp. 565–583, 2000.

[43] M. Grewal, L. Weill, and A. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley-Son, 2001.

[44] M. Briers, S. Maskell, and R. Wright, “A Rao-Blackwellised unscented Kalman filter,” in *International conference on Information Fusion*, Cairns, Queensland, Australia, 2003.

[45] E. Wan and R. van der Merwe, “The Unscented Kalman Filter for Nonlinear Estimation,” in *Proc. of IEEE Symposium 2000 (AS-SPCC), Lake Louise, Alberta, Canada, Oct. 2000*, 2000.

[46] S. Julier, J. Uhlmann, and H. Durrant-Whyte, “A new method for the nonlinear transformation of means and covariances in filters and estimators,” *IEEE Transactions on Automatic Control*, vol. 45, no. 3, 2000.

Appendix

A. UKF based sensor failure detection

The innovation ν_t, which measures the difference between the observation and the predicted observations follows, if Kalman filters assumptions are verified, a zero mean Gaussian pdf of covariance matrix S_t.

$$\nu_t \sim \mathcal{N}(0, S_t)$$

The quadratic normalized innovation r_t, defined by

$$r_t = \nu_t^T S_t^{-1} \nu_t$$

then follows a χ^2 pdf with $l = \dim \nu_t$ degrees of freedom [7]. A study of the value of r_t permits, if this value is very high and improbable compared to χ^2 tables, to detect a conflict between the new observation and the predicted observation (given by the last estimate and the evolution model). If the last estimate and the evolution model are supposed correct, then the sensor is supposed failing and the measure is rejected. This study permits to avoid to integrate failed data in the Kalman filter, causing long term damages.