Distributed Bayesian fault diagnosis of jump Markov systems in wireless sensor networks

Hichem Snoussi* and Cédric Richard

ICD/LM2S, University of Technology of Troyes,
12, rue Marie Curie, 10000, France
E-mail: snoussi@utt.fr
E-mail: richard@utt.fr
*Corresponding author

Abstract: A Bayesian distributed online change detection algorithm is proposed for monitoring a dynamical system by a wireless sensor network. The proposed solution relies on modelling the system dynamics by a jump Markov system with a finite set of states, including the abrupt change behaviour. For each discrete state, an observed system is assumed to evolve according to a state-space model. The collaborative strategy ensures the efficiency and the robustness of the data processing, while limiting the required communications bandwith. An efficient Rao-Blackwellised Collaborative Particle Filter (RB-CPF) is proposed to estimate the a posteriori probability of the discrete states of the observed systems. The Rao-Blackwellisation procedure combines a Sequential Monte-Carlo (SMC) filter with a bank of distributed Kalman filters. In order to prolong the sensor network lifetime, only few active (leader) nodes are selected according to a spatio-temporal selection protocol. This protocol is based on a trade-off between error propagation, communications constraints and information content complementarity of distributed data. Only sufficient statistics are communicated between leader nodes and their collaborators.

Keywords: collaborative sensor network; online change detection; Rao-Blackwellised particle filter; RB-CPF.

Reference to this paper should be made as follows: Snoussi, H. and Richard, C. (2007) 'Distributed Bayesian fault diagnosis of jump Markov systems in wireless sensor networks', *Int. J. Sensor Networks*, Vol. 2, Nos. 1/2, pp.118–127.

Biographical notes: Hichem Snoussi received the Diploma degree in Electrical Engineering from the Ecole Superieure d'Electricite (Supelec), Gif-sur-Yvette, France, in 2000. He also received the DEA degree and the PhD in Signal Processing from the University of Paris-Sud, Orsay, France, in 2000 and 2003, respectively. Between 2003 and 2004, he was postdoctoral Researcher at IRCCyN, Institut de Recherches en Communications et Cybernétiques de Nantes. He has spent short periods as visiting scientist at the Brain Science Institute, RIKEN, Japan and Olin Neuropsychiatry Research Center at the Institute of Living in USA. Since 2005, he has been Associate Professor at the University of Technology of Troyes, France. His research interests include Bayesian technics for source separation, information geometry, differential geometry, machine learning, robust statistics, with application to brain signal processing, astrophysics and advanced signal processing technics in sensor networks.

Cédric Richard received the Dipl-Ing and the MS degrees in 1994 and the PhD degree in 1998 from Compiègne University of Technology, France, all in Electrical and Computer Engineering. From 1999 to 2003, he was an Associate Professor at Troyes University of Technology, France. Since 2003, he has been a Professor at the Systems Modelling and Dependability Laboratory, Troyes University of Technology. His current research interests include statistical signal processing and machine learning. He is the author of over 70 papers. He is the General Chair of the XXIth Francophone Conference GRETSI on Signal and Image Processing to be held in Troyes, France, in 2007. In 2005, he was offered the position of chairman of a pattern recognition section of the federative CNRS research group ISIS on Information, Signal, Images and Vision. He is also in charge of the PhD students' network of this group. He is a Member of GRETSI association board, and of the EURASIP and IEEE-SP societies. He serves also as an Associate Editor of the *IEEE Transactions on Signal Processing*.

1 Introduction

A sensor network is a system made up of several tens to several hundreds of interconnected nodes, each one, of a sensor, an information processing unit and a communication block. The nodes have a zone of extremely reduced cover and are deployed in a dense way in heterogeneous environments. They are autonomous and have an energy reserve, the renewal of which could be impossible, limiting thus their lifespan. Each node must be able to treat the received data, to make a

local decision and to communicate it in an autonomous way to the closer nodes to which it is connected. This cooperation is intended to ensure best decision making in spite of the limits in terms of power consumption and processing capability. Collaborative information processing in sensor networks is thus becoming a very attractive field of research (Special Issue, 2002, 2005, 2006).

In this paper, the signal processing objective is to online detect the state change of a system observed by a sensor network. The efficient online state detection, in an automatic way, is very important for the system functioning security. In fact, according to each state, the system should adopt a specific behaviour. For example, an autonomous robot must be able to detect its state and carry out repairs if necessary, without human intervention, by processing the data received from the on-board sensors (de Freitas, 2002; Washington, 2000). Another challenging application is tracking a maneuvering target in a surveillance region (Li and Jilkov, 2003, 2005; Pasha et al., 2006). One can also mention the use of the sensor networks for the monitoring of production systems in order to face the industrial risks, the monitoring of the houses for safety or the house automation, the air and transport control in general, intelligent alarms for the prevention of natural disasters. With such systems, the automatic control of an event or an incident rests on the reliability of the network for efficient and robust decision making.

Concerning the data processing at each smart node, we adopt a probabilistic approach to model the system dynamics. The system is described by a jump Markov linear Gaussian model where the conditional Gaussians depend on the discrete state of the system and also on the sensor. An extension to non-linear models is also proposed to obtain an effective solution when dealing with real sensing nodes tracking a maneuvering target. The state change detection is resumed in the posterior marginal probability of the discrete state. To solve the inference problem, we use the particle filter as an approximate Monte-Carlo inference method able to deal with the intractable analytical aspect of the dynamical system update. Our contribution consists of proposing and implementing a collaborative distributed particle filter for estimating the marginal a posteriori probabilities of the system discrete states. Recently, distributed particle filters were proposed in literature (Ihler et al., 2005; Sheng et al., 2005). In the previously proposed distributed particle filters, the conditional distributions of the distributed collected data (likelihoods) are assumed to be independent. However, with jump Markov models, the conditional distributions are no more independent, requiring a more elaborate strategy. In fact, applying these particle filters to the jump Markov models, one needs to consider jointly the continuous and the discrete states of the system. As shown in de Freitas (2002), in a centralised processing, the particle filtering of the joint state leads to poor results. Our contribution consists thus in extending the Rao-Blackwellised approach, proposed by de Freitas (2002), in a distributed environment. The leader node collaborates with the remaining nodes at each time step. The temporal selection of the leader node is based on a trade-off between information relevance, communication cost and propagation error. The spatial selection of the leader collaborators relies on the same trade-off except that the information relevance takes an information complementarity form. The main difficulty of the spatial collaboration, within the Rao-Blackwellised distributed particle filter, is the fact that the sensors marginal likelihoods are no more independent. We show in the proposed collaborative strategy how to circumvent this difficulty while propagating only sufficient second order statistics through the sensor network.

This paper is organised as follows: in Section 2, the probabilistic change detection model and the centralised particle filter are briefly described. The Rao-Blackwellised implementation of the centralised particle filter is also recalled. Section 3 contains the main contribution of this paper which is an optimal online change detection procedure resulting from the spatio-temporal collaboration between the leader nodes and their collaborators. This strategy relies essentially on:

1 an information theoretic-based criteria for the spatio-temporal selection of the leader node and its collaborators, under communication constraints and

2 an optimal update of the sufficient statistics exchanged between leader nodes.

Section 4 is devoted to the extension of the proposed algorithm to the jump Markov non-linear models. In Section 5, numerical results corroborating the proposed algorithm effectiveness are shown.

2 Centralised online change detection

In this section, we briefly recall the particle filter method for online change detection. It is an approximate Monte-Carlo method estimating, recursively in time, the posterior probabilities of the discrete state of the system, given the observations. Moreover, the particle filter provides a point mass approximation of the distributions of the hidden continuous states. For more details and a comprehensive review of the particle filter (see Andrieu et al., 2004; Doucet et al., 2000, 2001). In the following, we consider jump Markov linear models. The extension to non-linear models is considered in Section 4.

2.1 Distributed state space model

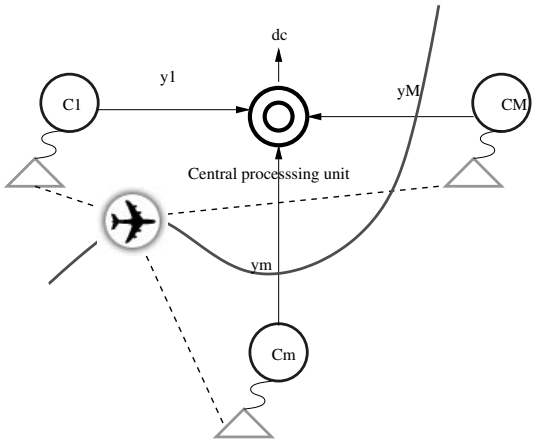
The Bayesian change detection algorithm is based on a discrete time jump Markov linear state-space model. This model involves two different hidden states: a discrete state and a continuous state. The discrete state changes in time according to a first order Markov model. For each discrete state, the system, observed by a sensor network composed of M nodes, evolves in time according to a different linear Gaussian model:

$$\begin{cases} z_t \sim P(z_t \mid z_{t-1}) \\ \boldsymbol{x}_t = \boldsymbol{A}(z_t)\boldsymbol{x}_{t-1} + \boldsymbol{B}(z_t)\boldsymbol{w}_t \\ \boldsymbol{y}_t^{(m)} = \boldsymbol{C}_m(z_t)\boldsymbol{x}_t + \boldsymbol{D}_m(z_t)\boldsymbol{v}_t^m, \quad m = 1, \ldots, M \end{cases} \tag{1}$$

where $\boldsymbol{y}_t^{(m)} \in \mathbb{R}^{n_y}$ denotes the observations transmitted from the sensor C^m at time t to the central processing unit

(see Figure 1), $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$ denotes the unknown continuous state and $z_t \in \mathcal{Z} = \{1, \ldots, K\}$ denotes the unknown discrete state. The transition probability $P(z_t \mid z_{t-1})$ represents the prior information about the dynamic variation of the system. The noises $\boldsymbol{w}_t$ and $\boldsymbol{v}_t^m$ are distributed according to i.i.d Gaussians $\mathcal{N}(0, \boldsymbol{I}_{n_x})$ and $\mathcal{N}(0, \boldsymbol{I}_{n_y})$, respectively. Note that the hidden states and their stochastic a priori models do not depend on the sensor node as they are characteristic of the observed system dynamics. The model parameters $\{\boldsymbol{A}, \boldsymbol{B}, \{\boldsymbol{C}_m\}_{m=1}^M, \{\boldsymbol{D}_m\}_{m=1}^M\}$ are assumed to be known or learned in a preprocessing step. Unsupervised learning of these parameters can be solved with an Expectation-Maximisation (EM) algorithm (Dempster et al., 1977), exploiting the latent data structure of the problem: the incomplete data are the observations and the hidden data are the continuous and the discrete states. For more details about the EM learning of linear model parameters, refer to Roweis and Ghahramani (1999).

Figure 1 Centralised processing: the sensors transmit the row data to the central unit



In this paper, we assume that, given the states $\boldsymbol{x}_t$ and z_t, the sensor noises are stochastically independent:

$$p\left(\boldsymbol{y}_t^{(1)}, \ldots, \boldsymbol{y}_t^{(M)} \mid \boldsymbol{x}_t, z_t\right) = \prod_{m=1}^{M} p_m\left(\boldsymbol{y}_t^{(m)} \mid \boldsymbol{x}_t, z_t\right)$$

Consequently, concatenating the observations gathered in the central unit, $\boldsymbol{y}_t = [\boldsymbol{y}_t^{(1)}, \ldots, \boldsymbol{y}_t^{(M)}]$ and replacing the distribution product $\prod p_m$ by an observation distribution p_y, the stochastic model (1) is rewritten as:

$$\begin{cases} z_t \sim P(z_t \mid z_{t-1}) \\ \boldsymbol{x}_t = \boldsymbol{A}(z_t)\boldsymbol{x}_{t-1} + \boldsymbol{B}(z_t)\boldsymbol{w}_t \\ \boldsymbol{y}_t \sim \mathcal{N}(\boldsymbol{C}(z_t)\boldsymbol{x}_t, \boldsymbol{R}_y(z_t)) \end{cases} \tag{2}$$

where $\boldsymbol{C} = [\boldsymbol{C}_1^T, \ldots, \boldsymbol{C}_M^T]^T$ and $\boldsymbol{R}_y$ is the block diagonal covariance matrix with block matrices equal to $\boldsymbol{D}_m\boldsymbol{D}_m^T$. Hence, the centralised processing relies on the usual jump Markov state space model.

The Bayesian online change detection is based on the estimation of the posterior marginal probability $P(z_t \mid \boldsymbol{y}_{1:t})$. However, the probabilistic system model (2) involves hidden continuous variables $\boldsymbol{x}_{0:t}$. Therefore, the computation of the marginal distribution involves two intractable integrals: integration with respect to the past of the discrete time Markov chain $z_{0:t-1}$ and integration with respect to the hidden continuous states $\boldsymbol{x}_{0:t}$:

$$P(z_t \mid \boldsymbol{y}_{1:t}) = \sum_{z_{0:t-1}} \int p(z_{0:t}, \boldsymbol{x}_{0:t} \mid \boldsymbol{y}_{1:t}) d\boldsymbol{x}_{0:t}$$

Therefore, one has to resort to Monte-Carlo approximation where the joint posterior distribution $p(z_{0:t}, \boldsymbol{x}_{0:t} | \boldsymbol{y}_{1:t})$ is approximated by the point-mass distribution of a set of weighted samples (called particles) $\{z_{0:t}^{(i)}, \boldsymbol{x}_{0:t}^{(i)}, w_t^{(i)}\}_{i=1}^N$:

$$\hat{P}_N(z_{0:t}, \boldsymbol{x}_{0:t} \mid \boldsymbol{y}_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{z_{0:t}^{(i)}, \boldsymbol{x}_{0:t}^{(i)}}(d\boldsymbol{x}_{0:t}, z_{0:t})$$

where $\delta_{z_{0:t}^{(i)}, \boldsymbol{x}_{0:t}^{(i)}}(d\boldsymbol{x}_{0:t}, z_{0:t})$ denotes the Dirac function.

Based on the same set of particles, the marginal posterior probability (of interest) $P(z_t \mid \boldsymbol{y}_{1:t})$ can also be approximated as follows:

$$P(z_t = k \mid \boldsymbol{y}_{1:t}) \simeq \sum_{i=1}^{N} w_t^{(i)} \mathcal{I}\left(z_t^{(i)} = k\right)$$

where $\mathcal{I}(.)$ denotes the indicator function.

Backward estimation of the marginal discrete state probability is also possible given the particles $\{z_{0:t+t^*}^{(i)}, \boldsymbol{x}_{0:t+t^*}^{(i)}, w_{t+t^*}^{(i)}\}_{i=1}^N$:

$$P(z_t = k \mid \boldsymbol{y}_{1:t+t^*}) \simeq \sum_{i=1}^{N} w_{t+t^*}^{(i)} \mathcal{I}(z_{t\,|\,t+t^*}^{(i)} = k)$$

In the Bayesian Importance Sampling (IS) method, the particles $\{z_{0:t}^{(i)}, \boldsymbol{x}_{0:t}^{(i)}\}_{i=1}^N$ are sampled according to a proposal distribution $\pi(z_{0:t}, \boldsymbol{x}_{0:t} \mid \boldsymbol{y}_{1:t})$ and $\{w_t^{(i)}\}$ are the corresponding normalised importance weights:

$$w_t^{(i)} \propto \frac{p\left(\boldsymbol{y}_{1:t} \mid z_{0:t}^{(i)}, \boldsymbol{x}_{0:t}^{(i)}\right) p\left(z_{0:t}^{(i)}, \boldsymbol{x}_{0:t}^{(i)}\right)}{\pi\left(z_{0:t}^{(i)}, \boldsymbol{x}_{0:t}^{(i)} \mid \boldsymbol{y}_{1:t}\right)}$$

2.2 Sequential Monte-Carlo

Sequential Monte-Carlo (SMC) consists of propagating the trajectories $\{z_{0:t}^{(i)}, \boldsymbol{x}_{0:t}^{(i)}\}_{i=1}^N$ in time without modifying the past simulated particles. This is possible for the class of proposal distributions having the following form:

$$\begin{aligned} \pi(z_{0:t}, \boldsymbol{x}_{0:t} \mid \boldsymbol{y}_{1:t}) &= \pi(z_{0:t-1}, \boldsymbol{x}_{0:t-1} \mid \boldsymbol{y}_{1:t-1}) \\ &\quad \times \pi(z_t, \boldsymbol{x}_t \mid z_{0:t-1}, \boldsymbol{x}_{0:t-1}, \boldsymbol{y}_{1:t}) \end{aligned}$$

The normalised importance weights are then recursively computed in time as:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p\left(\boldsymbol{y}_t | z_t^{(i)}, \boldsymbol{x}_t^{(i)}\right) p\left(z_t^{(i)}, \boldsymbol{x}_t^{(i)} | z_{0:t-1}^{(i)}, \boldsymbol{x}_{0:t-1}^{(i)}\right)}{\pi\left(z_t^{(i)}, \boldsymbol{x}_t^{(i)} | z_{0:t-1}^{(i)}, \boldsymbol{x}_{0:t-1}^{(i)}, \boldsymbol{y}_{1:t}\right)} \tag{3}$$

For the considered jump Markov linear state-space model (2), one can adopt the transition prior as the proposal distribution:

$$\pi\left(z_t^{(i)}, \boldsymbol{x}_t^{(i)} | z_{0:t-1}^{(i)}, \boldsymbol{x}_{0:t-1}^{(i)}, \boldsymbol{y}_{1:t}\right) = p_x(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, z_t) P(z_t | z_{t-1})$$

in which case the weights are updated according to the likelihood function:

$$w_t^{(i)} \propto w_{t-1}^{(i)}\, p\left(\boldsymbol{y}_t \mid z_t^{(i)}, \boldsymbol{x}_t^{(i)}\right) \tag{4}$$

The centralised online change detection algorithm (shown in Figure 2) consists of 2 steps: the sequential importance sampling step and the selection step. The selection (resampling) step replaces the weighted particles by unweighted particles to avoid the collapse of the Monte-Carlo approximation that caused the variance increase of the weights. It consists of selecting the trajectories $\{z_{0:t}^{(i)}, \boldsymbol{x}_{0:t}^{(i)}\}$ with probabilities $w_t^{(i)}$. The trajectories with weak weights are eliminated and the trajectories with strong weights are multiplied. After the selection step, all the weights are equal to $1/N$.

Figure 2 Centralised particular filter algorithm

Step 0: Initialisation

$$(i) \quad z_0^{(i)} \sim P_0(z)$$
$$(ii) \quad \boldsymbol{x}_0^{(i)} \sim p_0(\boldsymbol{x})$$

Step 1: For $t = 1$ to T,
a- Sequential importance sampling:
- For $i = 1, ..., N$, sample from the transition priors:

$$\hat{z}_t^{(i)} \sim P(z_t \mid z_{t-1}^{(i)})$$
$$\hat{\boldsymbol{x}}_t^{(i)} \sim p_x(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}^{(i)}, u_t, \hat{z}_t^{(i)})$$

and set

$$(\hat{z}_{0:t}^{(i)}, \hat{\boldsymbol{x}}_{0:t}^{(i)}) = (\hat{z}_t^{(i)}, \hat{\boldsymbol{x}}_t^{(i)}, z_{0:t-1}^{(i)}, \boldsymbol{x}_{0:t-1}^{(i)})$$

b- Update the importance weights:
- For $i = 1, ..., N$, evaluate and normalise the weights:

$$w_t^{(i)} \propto p(\boldsymbol{y}_t \mid \hat{z}_t^{(i)}, \hat{\boldsymbol{x}}_t^{(i)})$$

c- Resampling:
- Select with replacement from $\{\hat{z}_{0:t}^{(i)}, \hat{\boldsymbol{x}}_{0:t}^{(i)}\}_{i=1}^N$ with probability $\{w_t^{(i)}\}$ to obtain N particles $z_{0:t}^{(i)}, \boldsymbol{x}_{0:t}^{(i)}\}_{i=1}^N$

2.3 Rao-Blackwellised SMC

Considering the joint state $\{\boldsymbol{x}_t, z_t\}$, the SMC algorithm yields poor online detection results. An efficient Rao-Blackwellised SMC, proposed by de Freitas (2002), considerably improves the state estimation. The principle of this procedure consists of noting that given the discrete state, the continuous state is a posteriori Gaussian. Thus, based on a bank of Kalman filters, one can sequentially update the marginal a posteriori probability $p(z_t \mid \boldsymbol{y}_{1:t})$. In fact, the probability of the trajectory $z_{0:t}$ satisfies the following recursion:

$$p(z_{0:t}|\boldsymbol{y}_{1:t}) = p(z_{0:t-1}|\boldsymbol{y}_{1:t-1}) \frac{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, z_{0:t}) P(z_t|z_{t-1})}{p(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1})}$$

In the SMC algorithm, predicting the discrete states $\{z_t^{(i)}\}$ according to the transition prior $P(z_t|z_{t-1})$ leads to the following particle weight updating:

$$w_t^{(i)} \propto w_{t-1}^{(i)}\, p\left(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1}, z_{0:t}^{(i)}\right) \tag{5}$$

The computation of the Gaussian data prediction distribution $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, z_{0:t}^{(i)})$ is based on the mean $\boldsymbol{y}_{t|t-1} = \mathrm{E}\left[\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}\right]$ and covariance $\boldsymbol{S}_t = \mathrm{cov}(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})$ online updates. These second order statistics are jointly updated with the mean and covariance of the continuous state by a Kalman filter as follows:

$$\boldsymbol{\mu}_{t|t-1}^{(i)} = \boldsymbol{A}\left(z_t^{(i)}\right) \boldsymbol{\mu}_{t-1|t-1}^{(i)}$$
$$\boldsymbol{\Sigma}_{t|t-1}^{(i)} = \boldsymbol{A}\left(z_t^{(i)}\right) \boldsymbol{\Sigma}_{t-1|t-1}^{(i)} \boldsymbol{A}\left(z_t^{(i)}\right)^T + \boldsymbol{B}\left(z_t^{(i)}\right) \boldsymbol{B}\left(z_t^{(i)}\right)^T$$
$$\boldsymbol{S}_t^{(i)} = \boldsymbol{C}\left(z_t^{(i)}\right) \boldsymbol{\Sigma}_{t|t-1}^{(i)} \boldsymbol{C}\left(z_t^{(i)}\right)^T + \boldsymbol{R}_y\left(z_t^{(i)}\right)$$
$$\boldsymbol{y}_{t|t-1}^{(i)} = \boldsymbol{C}\left(z_t^{(i)}\right) \boldsymbol{\mu}_{t|t-1}^{(i)}$$
$$\boldsymbol{\mu}_{t|t}^{(i)} = \boldsymbol{\mu}_{t|t-1}^{(i)} + \boldsymbol{\Sigma}_{t|t-1}^{(i)} \boldsymbol{C}\left(z_t^{(i)}\right)^T \boldsymbol{S}_t^{-1(i)} \left(\boldsymbol{y}_t - \boldsymbol{y}_{t|t-1}^{(i)}\right)$$
$$\boldsymbol{\Sigma}_{t|t}^{(i)} = \boldsymbol{\Sigma}_{t|t-1}^{(i)} - \boldsymbol{\Sigma}_{t|t-1}^{(i)} \boldsymbol{C}\left(z_t^{(i)}\right)^T \boldsymbol{S}_t^{-1(i)} \boldsymbol{C}\left(z_t^{(i)}\right) \boldsymbol{\Sigma}_{t|t-1}^{(i)}$$

where $\boldsymbol{\mu}_{t|t-1} = \mathrm{E}[\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t-1}]$, $\boldsymbol{\Sigma}_{t|t-1} = \mathrm{cov}(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t-1})$, $\boldsymbol{\mu}_{t|t} = \mathrm{E}[\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t}]$ and $\boldsymbol{\Sigma}_{t|t} = \mathrm{cov}(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t})$. The predictive density is then simply evaluated by:

$$p\left(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1}, z_{0:t}^{(i)}\right) = \mathcal{N}\left(\boldsymbol{y}_t\,;\, \boldsymbol{y}_{t|t-1}^{(i)}, \boldsymbol{S}_t^{(i)}\right)$$

The centralised Rao-Blackwellised SMC algorithm is summarised in Figure 3.

Figure 3 Centralised Rao-Blackwellised particular filter algorithm

Sequential sampling step:
- For $i = 1, \ldots, N$, sample from the transition prior:

$$\hat{z}_t^{(i)} \sim P(z_t \mid z_{t-1}^{(i)})$$

Weight updating step:
-For $i = 1, \ldots, N$, update the sufficient statistics (jointly with the Kalman filter) and evaluate the importance weights:

$$w_t^{(i)} \propto p(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1}, z_{0:t}^{(i)})$$

Resampling step:
- Select with replacement from $\{\hat{z}_{0:t}^{(i)}\}_{i=1}^N$ with probabilities $\{w_t^{(i)}\}$ to obtain N particles $\{z_{0:t}^{(i)}\}_{i=1}^N$

3 Collaborative online change detection

In a sensor network, each node must be able to treat the received data, to make a local decision and to communicate it in an autonomous way with the closer nodes to which it is connected. This cooperation is intended to ensure best decision-making possible in spite of the limits in terms of power consumption and processing capability. In the following, we propose a collaborative Rao-Blackwellised particle filter where the smart nodes collaborate in sequentially updating the filtering distribution. They only exchange few statistics characterising message approximations. The observed data $\{\boldsymbol{y}_t^{(m)}\}_{m=1}^M$ are not

propagated in the sensor network. The proposed distributed algorithm is characterised by a spatial and a temporal collaborative processing of the sequentially collected observations.

3.1 Temporal leader node selection

The temporal collaboration consists of selecting, after the sequential probability update, the leader node at the next time step. The selection procedure is based on ranking the nodes according to an information-theoretic cost function $J(m)$. The first ranked node m^* ($\arg\max_m J(m)$) is the next leader candidate. At time step $t-1$, the chosen cost function is a trade-off between information gain and compression loss:

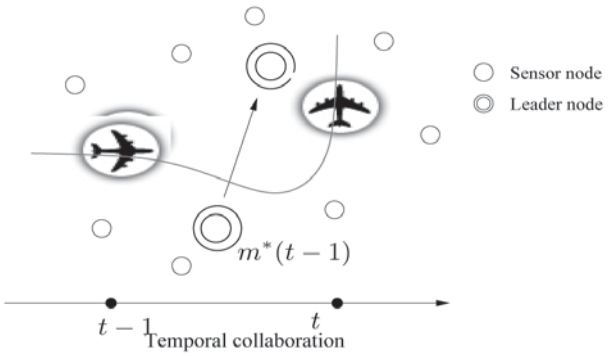
$$J_t(m) = \mathcal{I}(m) + \alpha\mathcal{E}(m) \tag{6}$$

where the first term of the above criteria represents the information content relevance of the measured data on the node m, at the time step t:

$$\mathcal{I}(m) = \mathrm{E}\Big[D_{\mathrm{KL}}\left(p\left(\boldsymbol{y}_t^m \mid \boldsymbol{x}_t, z_t\right) || p\left(\boldsymbol{y}_t^m \mid \boldsymbol{y}_{1:t-1}, z_{0:t}\right)\right)\Big] \tag{7}$$

where D_{KL} is the Kullback-Leibler divergence between the likelihood and the data predicted density, the expectation is evaluated according to the joint filtering distribution $p(\boldsymbol{x}_t, z_{0:t} \mid \boldsymbol{y}_{1:t-1})$. This can be considered as a data augmentation version of criteria proposed by Doucet et al. (2002) for sensor management. The second term $\mathcal{E}(m)$ is the message error when transferring sufficient statistics from the leader node $m^*(t)$ to node m under the communication constraint $c_m < c_{\max}$, where c_m is the communication cost of transferring information to node m. Therefore, one can easily obtain the bound $N \times c_{\max}$ on the communication cost, where N is the number of message transmissions. Note that if the budget $c_{\max}$ is very low, then the propagation error will be higher leading to a poor inference performance. The negative coefficient α represents the trade-off between the information gain and compression loss. Note that $\mathcal{E}(m^*(t-1)) = 0$, meaning that the leader node may select itself as the next leader, when the increase of the data relevance of the other nodes does not compensate the compression loss. Figure 4 illustrates the proposed selection protocol.

Figure 4 The selection of the leader node (and the hand-off) at time t is based on a trade-off between the data relevance and the communication cost (constraining the message size)



3.1.1 Computation of the information gain

In Doucet et al. (2002), a Monte-Carlo procedure is proposed to compute the first term of the cost function (6). However, in our problem setting, using the jump Markov linear state model, the term $\mathcal{I}$ can be evaluated with a Rao-Blackwellised scheme. In fact, given the discrete state trajectory $z_{0:t}^{(i)}$, the likelihood $p(\boldsymbol{y}_t^m \mid \boldsymbol{x}_t)$ and the predictive distribution $p(\boldsymbol{y}_t^m \mid \boldsymbol{y}_{1:t-1}, z_{0:t}^{(i)})$ are both Gaussians and the expectation of the Kullback-Leibler divergence[1] in expression (7) can be exactly evaluated as follows:

$$\underset{\mid z_{0:t}^{(i)}}{\mathcal{I}}(m) = \frac{1}{2}\log\left|\boldsymbol{I}_m + \left(\boldsymbol{D}_m(z_t)\boldsymbol{D}_m(z_t)^T\right)^{-1} \times \boldsymbol{C}_m(z_t)\boldsymbol{\Sigma}_{t|t-1}^{(i)}\boldsymbol{C}_m(z_t)^T\right|$$

where the subscript '$z_{0:t}(i)$' means that the expectation is evaluated conditioned on the discrete state, $\boldsymbol{I}_m$ denotes the identity matrix and $\boldsymbol{\Sigma}_{t|t-1}^{(i)}$ is the predicted covariance $\boldsymbol{A}(z_t^{(i)})\boldsymbol{\Sigma}_{t-1|t-1}^{(i)}\boldsymbol{A}(z_t^{(i)})^T + \boldsymbol{B}(z_t^{(i)})\boldsymbol{B}(z_t^{(i)})^T$. It can be easily noted that maximising the term $\mathcal{I}_{\mid z_{0:t}^{(i)}}(m)$ relies on the maximisation of the information/noise ratio, where the information content is evaluated by the matrix $\boldsymbol{C}_m(z_t)\boldsymbol{\Sigma}_{t|t-1}^{(i)}\boldsymbol{C}_m(z_t)^T$ (norm of the observation matrix in the state covariance basis). The trajectory $z_{0:t}^{(i)}$ is composed of the particle past trajectory $z_{0:t-1}^{(i)}$ having $w_{t-1}^{(i)}$ as the importance weight and the predicted $z_t^{(i)}$ according to the transition prior $P(z_t|z_{t-1})$. The information criteria $\mathcal{I}(m)$ is thus approximated by a Monte-Carlo scheme as follows:

$$\mathcal{I}(m) = \mathrm{E}\left[\mathcal{I}_{\mid z_{0:t}}\right] = \sum_{z_{0:t}} \mathcal{I}_{\mid z_{0:t}} p(z_{0:t} \mid \boldsymbol{y}_{1:t-1}) \approx \sum_{z_{0:t}} \mathcal{I}_{\mid z_{0:t}} w_{t-1}^{(i)}$$

3.1.2 Computation of the compression loss

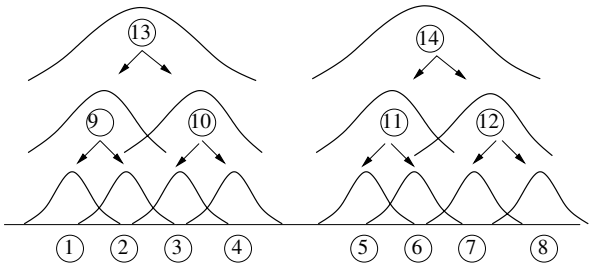
Propagating all the particles $\{\boldsymbol{\mu}_{t|t}^{(i)}, \boldsymbol{\Sigma}_{t|t}^{(i)}, w_t^{(i)}\}$ is not allowed in a wireless sensor network because of communication constraints. The KD-tree Gaussian mixture is a suitable approximation when communicating distribution messages (Ihler et al., 2004). The KD-tree is a multiscale mixture of Gaussian approximation of a given dataset. It consists of describing a large data set (particles) with a set a few subtrees, each subtree is a Gaussian whose statistics can be recursively computed. The top node of the tree is the largest scale and the leaf nodes represent the finest scales. The internal nodes represent intermediate resolutions. See Figure 5 for an illustration.

For the detection purpose, the KD-tree is separately applied on K particle groups, each group corresponding to a discrete state. This is an interesting feature of the KD-tree approximation as it maintains the multimodality aspect of the Kalman mixture structure. The set of weighted sufficient statistics $\{\boldsymbol{\mu}_{t|t}^{(i)}, \boldsymbol{\Sigma}_{t|t}^{(i)}, w_t^{(i)}\}_{i\in\mathcal{T}_k}$, where $\mathcal{T}_k = \{i \mid z_t^{(i)} = k\}$, is approximated by a set of nodes $\mathcal{S}$ containing one and only one ancestor of each leaf node:

$$\hat{p}(x_t \mid y_{1:t}, z_t = k) = \sum_{i \in \mathcal{T}_k} w_t^{(i)} \mathcal{N}\left(\boldsymbol{\mu}_{t|t}^{(i)}, \boldsymbol{\Sigma}_{t|t}^{(i)}\right) \approx \sum_{s \in \mathcal{S}} \alpha_s \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)$$

where $\mathcal{N}(.)$ denotes the Gaussian density and $|\mathcal{S}| \ll |\mathcal{T}_k|$. Another interesting feature of the KD-tree approximation is the fact that the second order moments $(\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)$ can be considered as Kalman mean and covariance updates $(\tilde{\boldsymbol{\mu}}_{t|t}^{(i)}, \tilde{\boldsymbol{\Sigma}}_{t|t}^{(i)})$. The weight α_s is the sum of the leaf nodes weights $w_t^{(i)}$ corresponding to the subtree s. Figure 5 illustrates the KD-tree approximation adapted to the mixture Kalman filter.

Figure 5 KD-tree approximation of the Kalman mixture updates: components 1–4 are the leaf nodes for the state $z_t = 1$ and components 5–8 are the leaf nodes for the state $z_t = 2$



Increasing the resolution of the KD-tree representation is simply done by replacing the nodes $s \in \mathcal{S}$ by their left and right children nodes. In order to control error propagation, one needs a divergence measure between probability densities. Following the arguments in Ihler et al. (2004), the maximum log-error:

$$\mathrm{ML}(p, q) = \max_x \left| \frac{\log p(x)}{q(x)} \right| \tag{8}$$

is very suitable for bounding the belief propagation error and also it is adapted to the KD-tree representation. In fact, the error $\mathrm{ML}(\hat{p}, q_{\mathcal{S}})$ between the particle representation $\hat{p}$ and the KD-tree approximation $q_{\mathcal{S}}$ is bounded as follows:

$$\mathrm{ML}(\hat{p}, q_{\mathcal{S}}) \leq \max_{s \in \mathcal{S}} \mathrm{ML}(\hat{p}_s, q_s) \tag{9}$$

where $\mathrm{ML}(\hat{p}_s, q_s)$ is the error measure between the Gaussian stored at the node s and its corresponding leaf nodes.

Therefore, controlling the temporal propagation error while respecting the communication constraints consists of a trade-off between the resolution of the KD-tree representation and its encoding cost. As the resolution increases (going from top to bottom in the tree), the approximation error decreases while the communication cost increases. This can be easily implemented by recursively dividing the node $s \in \mathcal{S}$ having the maximum error measure in (9) while respecting the allowed communication cost.

Deciding the hand-over consists of comparing the information gain/compression loss ratio, computed for the selected leader candidate m_t^*, with a threshold β. In words, the hand-over to the node m_t^* is allowed if:

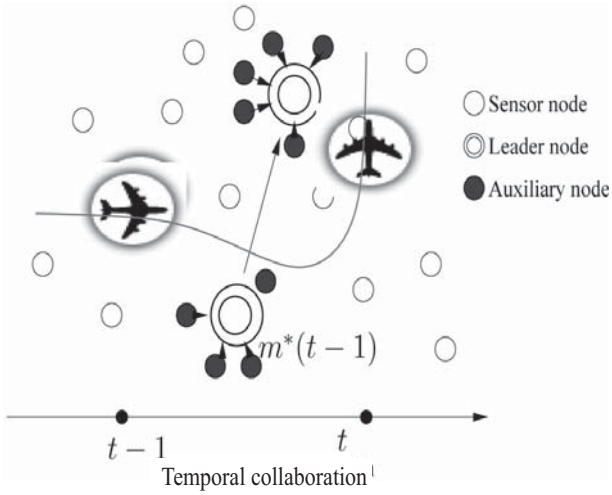
$$\frac{\mathcal{I}\left(m_t^*\right)}{\mathcal{I}\left(m_t^*\right) + \alpha \mathcal{E}\left(m_t^*\right)} > \beta$$

The threshold β is an increasing function of the energy reserve communicated by the active node's battery. If the energy reserve is very low ($\beta \approx 0$), the hand-over is almost surely done. However, if the energy reserve is at a correct level, the active node will take into consideration the information gain before performing the hand-over.

3.2 Spatial collaborative detection

A simple approach for spatial collaboration consists of local data transmission: the selected leader m_t^* at time t receives data sent by its neighbouring nodes. The neighbourhood is defined on a proximity basis in order to minimise the communication energy consumption. For instance, the nodes located at a distance less than a predetermined threshold (fixed according to consumption requirements and sensor technology) can be considered as neighbours. The leader node updates thus the system state based on its measured data and the data sent by its neighbours. Figure 6 illustrates this approach.

Figure 6 Spatial collaboration based on a clustering approach



In this paper, we propose a new spatial collaboration protocol based on the selection by the leader node of its collaborators at each time step. The spatial collaboration is based on two alternating steps:

1 the selection of the collaborator nodes path with a recursive procedure ensuring the distributed data information complementarity and

2 the spatial update of the particle weights, the particles being predicted in the leader node.

In the following, we outline the above two steps. For the clarity of presentation and notation convenience, $(\boldsymbol{\mu}_{t|t}^{(i,0)}, \boldsymbol{\Sigma}_{t|t}^{(i,0)})$ will denote the predicted Kalman mean and covariance $(\boldsymbol{\mu}_{t|t-1}^{(i)}, \boldsymbol{\Sigma}_{t|t-1}^{(i)})$, $\{w_t^{(i,0)}\}$ denotes their

corresponding importance weight computed in the leader node C_0. The prediction is performed in the leader node C_0.

3.2.1 Particle weight updating

In this paragraph, we show how the weight of a predicted state is updated taking into account the data of the leader node and also the data collected by the collaborator nodes. The communication constraints do not allow the propagation of raw data. Therefore, only sufficient statistics are exchanged between the leader node and its collaborators. The data measured at the leader node C_0 and its L collaborators $C_1, \ldots, C_L$ are denoted $\{\boldsymbol{y}_t^0, \boldsymbol{y}_t^1, \ldots, \boldsymbol{y}_t^L\}$, respectively. Contrary to the previously proposed distributed particle filters in literature, in the jump Markov model, the likelihood of the discrete state $p(\boldsymbol{y}_t^0, \boldsymbol{y}_t^1, \ldots, \boldsymbol{y}_t^L \mid \boldsymbol{y}_{1:t-1}, z_{0:t})$ cannot be factorised into $\prod_{l=0}^{L} p(\boldsymbol{y}_t^l \mid \boldsymbol{y}_{1:t-1}, z_{0:t})$. In fact, the predictive densities are dependent through the hidden continuous state. Consequently, the weight $w_t^{(i)} \propto p(\boldsymbol{y}_t^0, \boldsymbol{y}_t^1, \ldots, \boldsymbol{y}_t^L \mid \boldsymbol{y}_{1:t-1}, z_{0:t})$ of the predicted state $z_t^{(i)}$ cannot be updated by a simple cumulative product. However, the computation of the complete likelihood can be performed with a Kalman filter procedure. In fact, the complete likelihood can be decomposed with the sequential Bayes' rule as follows:

$$p\left(\boldsymbol{y}_t^0, \boldsymbol{y}_t^1, \ldots, \boldsymbol{y}_t^L \mid \boldsymbol{y}_{1:t-1}, z_{0:t}\right) = p\left(\boldsymbol{y}_t^0 \mid \boldsymbol{y}_{1:t-1}, z_{0:t}\right) \times \prod_{l=1}^{L} p\left(\boldsymbol{y}_t^l \mid \boldsymbol{y}_t^{l-1}, \ldots, \boldsymbol{y}_t^0, \boldsymbol{y}_{1:t-1}, z_{0:t}\right) \tag{10}$$

The predictive density $p(\boldsymbol{y}_t^0 \mid \boldsymbol{y}_{1:t-1}, z_{0:t})$ in the product (10) is updated according to the usual Kalman filter based on the data $\boldsymbol{y}_t^0$:

$$\begin{aligned} w_t^{(i,0)} &\propto p\left(\boldsymbol{y}_t^0 \mid \boldsymbol{y}_{1:t-1}, z_{0:t}^{(i)}\right) \\ &\propto \mathcal{N}\left(\boldsymbol{y}_t^0;\ \boldsymbol{y}_{t|t-1}^{i,0}, \boldsymbol{S}_t^{i,0}\right) \end{aligned}$$

where the sufficient statistics $(\boldsymbol{y}_{t|t-1}^{i,0}, \boldsymbol{S}_t^{i,0})$ are evaluated as follows:

$$\begin{aligned} \boldsymbol{y}_{t|t-1}^{(i,0)} &= \boldsymbol{C}_0\left(z_t^{(i)}\right) \boldsymbol{\mu}_{t|t-1}^{(i)} \\ \boldsymbol{S}_t^{(i,0)} &= \boldsymbol{C}_0\left(z_t^{(i)}\right) \boldsymbol{\Sigma}_{t|t-1}^{(i)} \boldsymbol{C}_0\left(z_t^{(i)}\right)^T + \boldsymbol{D}_0\left(z_t^{(i)}\right) \boldsymbol{D}_0\left(z_t^{(i)}\right)^T \end{aligned}$$

The mean and covariance updates are then,

$$\begin{aligned} \boldsymbol{\mu}_{t|t}^{(i,1)} &= \boldsymbol{\mu}_{t|t-1}^{(i)} + \boldsymbol{\Sigma}_{t|t-1}^{(i)} \boldsymbol{C}_0\left(z_t^{(i)}\right)^T \boldsymbol{S}_t^{-1(i,0)} \left(\boldsymbol{y}_t^0 - \boldsymbol{y}_{t|t-1}^{(i,0)}\right) \\ \boldsymbol{\Sigma}_{t|t}^{(i,1)} &= \boldsymbol{\Sigma}_{t|t-1}^{(i)} - \boldsymbol{\Sigma}_{t|t-1}^{(i)} \boldsymbol{C}_0\left(z_t^{(i)}\right)^T \boldsymbol{S}_t^{-1(i,0)} \boldsymbol{C}_0\left(z_t^{(i)}\right) \boldsymbol{\Sigma}_{t|t-1}^{(i)} \end{aligned}$$

Similarly, the subsequent predictive data densities $p(\boldsymbol{y}_t^l | \boldsymbol{y}_t^{l-1}, \ldots, \boldsymbol{y}_t^0, \boldsymbol{y}_{1:t-1}, z_{0:t})$ are evaluated by a Kalman filter, where the predicted mean and covariance are the updated mean and covariance computed and sent by the node C_{l-1}. Thus, the main difference with the usual Kalman filter is the fact that there is not a temporal prediction, the predicted statistics are the updated statistics by the previous collaborator node. The predictive density $p(\boldsymbol{y}_t^l | \boldsymbol{y}_t^{l-1}, \ldots, \boldsymbol{y}_t^0, \boldsymbol{y}_{1:t-1}, z_{0:t})$ is calculated as follows:

$$\begin{aligned} w_t^{(i,l)} &\propto p\left(\boldsymbol{y}_t^l \mid \boldsymbol{y}_t^{l-1}, \ldots, \boldsymbol{y}_t^0, \boldsymbol{y}_{1:t-1}, z_{0:t}\right) \\ &\propto \mathcal{N}\left(\boldsymbol{y}_t^l;\ \boldsymbol{y}_{t|t-1}^{i,l}, \boldsymbol{S}_t^{i,l}\right) \end{aligned} \tag{11}$$

where the sufficient statistics $(\boldsymbol{y}_{t|t-1}^{i,l}, \boldsymbol{S}_t^{i,l})$ are evaluated as follows:

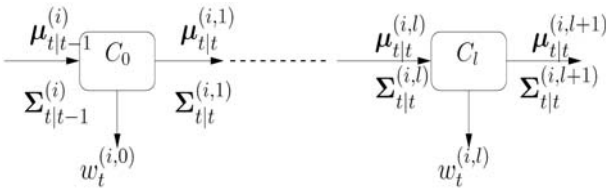
$$\begin{aligned} \boldsymbol{y}_{t|t}^{(i,l)} &= \boldsymbol{C}_l\left(z_t^{(i)}\right) \boldsymbol{\mu}_{t|t}^{(i,l)} \\ \boldsymbol{S}_t^{(i,l)} &= \boldsymbol{C}_l\left(z_t^{(i)}\right) \boldsymbol{\Sigma}_{t|t}^{(i,l)} \boldsymbol{C}_l\left(z_t^{(i)}\right)^T + \boldsymbol{D}_l\left(z_t^{(i)}\right) \boldsymbol{D}_l\left(z_t^{(i)}\right)^T \end{aligned}$$

The mean and covariance updated by the collaborator node C_l are then,

$$\begin{aligned} \boldsymbol{\mu}_{t|t}^{(i,l+1)} &= \boldsymbol{\mu}_{t|t}^{(i,l)} + \boldsymbol{\Sigma}_{t|t}^{(i,l)} \boldsymbol{C}_l\left(z_t^{(i)}\right)^T \boldsymbol{S}_t^{-1(i,l)} \left(\boldsymbol{y}_t^l - \boldsymbol{y}_{t|t}^{(i,l)}\right) \\ \boldsymbol{\Sigma}_{t|t}^{(i,l+1)} &= \boldsymbol{\Sigma}_{t|t}^{(i,l)} - \boldsymbol{\Sigma}_{t|t}^{(i,l)} \boldsymbol{C}_l\left(z_t^{(i)}\right)^T \boldsymbol{S}_t^{-1(i,l)} \boldsymbol{C}_l\left(z_t^{(i)}\right) \boldsymbol{\Sigma}_{t|t}^{(i,l)} \end{aligned} \tag{12}$$

Figure 7 illustrates the collaborative updating of the particle weights at each time step.

Figure 7 Spatial Kalman update of the mean, covariance and particle weight



Until now, we have considered the spatial update of one particle weight $w_t^{(i)}$. As we have mentioned in the previous section, updating all the particles is not possible under communication constraints. Fortunately, the KD-tree approximation preserves the same structure of the Kalman mixture scheme. The computed means, covariances and weights of the KD-tree Gaussian mixture can be put in correspondence with the updated Kalman means $\boldsymbol{\mu}_{t|t}^{(i)}$, the updated Kalman covariances $\boldsymbol{\Sigma}_{t|t}^{(i)}$ and the particle weights $w_t^{(i)}$. Then, the same spatial Kalman updating is applied on the KD-tree Gaussians.

3.2.2 Recursive path selection

The selection of collaborator nodes can be performed in a recursive manner: each selected collaborator, after updating the particle weights, selects one and only one next collaborator (see Figure 8). This recursion is necessary to ensure the information complementarity and thus avoid unnecessary redundant information. The selection is based on the same cost function (6) as in the temporal case, leading to similar expressions. The information gain $\mathcal{I}(m)$ computed by the node C_l to select the next collaborator C_{l+1} takes the form of information content complementarity. In fact, given

the discrete state trajectory $z_{0:t}^{(i)}$, the likelihood $p(\boldsymbol{y}_t^m \mid \boldsymbol{x}_t, z_{0:t})$ and the predictive distribution $p(\boldsymbol{y}_t^m \mid \boldsymbol{y}_t^l, ..., \boldsymbol{y}_t^0, \boldsymbol{y}_{1:t-1}, z_{0:t})$ of the candidate C_m data are both Gaussians and the expectation of the Kullback-Leibler divergence in expression (7) can be exactly evaluated as follows:

$$\underset{|\, z_{0:t}^{(i)}}{\mathcal{I}}(m) = \frac{1}{2}\log\left|\boldsymbol{I}_m + \left(\boldsymbol{D}_m(z_t)\boldsymbol{D}_m(z_t)^T\right)^{-1} \times \boldsymbol{C}_m(z_t)\boldsymbol{\Sigma}_{t|t}^{(i,l+1)}\boldsymbol{C}_m(z_t)^T\right|$$

where the subscript '$z_{0:t}(i)$' means that the expectation is evaluated conditioned on the discrete state, $\boldsymbol{I}_m$ denotes the identity matrix and $\boldsymbol{\Sigma}_{t|t}^{(i,l+1)}$ is the updated covariance by the node C_l. Note that the basic difference with the temporal selection procedure is that the predicted covariance $\boldsymbol{\Sigma}_{t|t-1}^{(i)}$ is replaced by the updated covariance $\boldsymbol{\Sigma}_{t|t}^{(i,l+1)}$. The information gain is then simply computed as:

$$\mathcal{I}(m) \approx \sum_{z_{0:t}} \mathcal{I}_{|\, z_{0:t}} w_t^{(i,l)}$$

Figure 9 illustrates the global spatio-temporal path of selected leader and auxiliary collaborator nodes.

Figure 8 Recursive spatial collaborator path selection

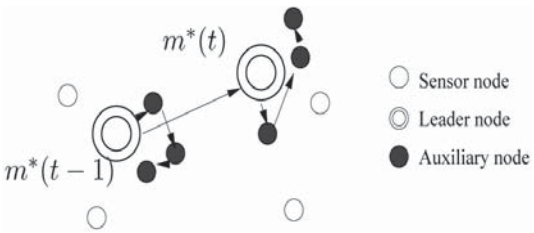
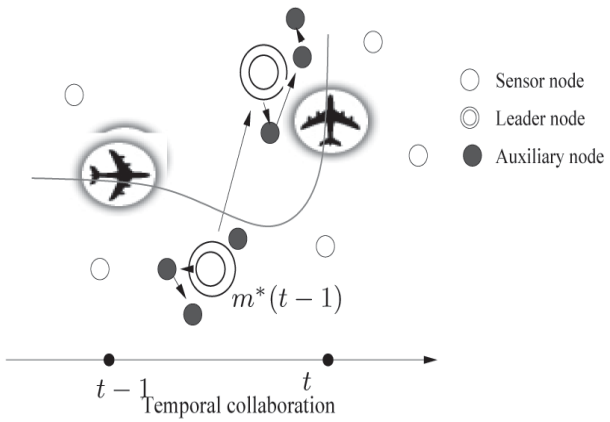


Figure 9 Temporal leader selection + Recursive spatial collaborator path selection



4 Non-linear jump Markov systems

In many real situations such as tracking manoeuvering targets with range sensors, the system dynamics as well as the measurement models are non-linear. The state-space model describing the transition prior and state/measurement relation takes the following form:

$$\begin{cases} z_t \sim P(z_t \mid z_{t-1}) \\ \boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, z_t) + \boldsymbol{B}(z_t)\boldsymbol{w}_t \\ \boldsymbol{y}_t^{(m)} = h(\boldsymbol{x}_t, m, z_t) + \boldsymbol{D}_m(z_t)\boldsymbol{v}_t^m, \quad m = 1, \ldots, M \end{cases} \tag{13}$$

where $f(.)$ and $h(.)$ are non-linear functions depending on the discrete state z_t. In addition, the function $h(.)$ depends on the sensor C_m. The noises $\boldsymbol{w}_t$ and $\boldsymbol{v}_t^m$ can still be assumed to be distributed according to i.i.d Gaussians $\mathcal{N}(0, \boldsymbol{I}_{n_x})$ and $\mathcal{N}(0, \boldsymbol{I}_{n_y})$, respectively. In a manner similar to the Extended Kalman Filter (EKF), the distributed Rao-Blackwellised Particle Filter, proposed in the previous section, can be extended to the non-linear jump Markov model (13). In other words, given a particle $z_{0:t}^{(i)}$, the predicted Kalman means $\boldsymbol{\mu}_{t|t-1}^{(i)}$ and the predicted data $\boldsymbol{y}_{t|t-1}^{(i)}$ are exactly computed according to the non-linear functions $f(.)$ and $h(.)$, respectively. However, to compute the predicted covariance and the updated covariances, the non-linear functions are linearised around the current state and the Jacobian (matrices of partial derivatives) are then used in a similar way as the matrices $\boldsymbol{A}(z_t^{(i)})$ and $\boldsymbol{C}_m(z_t^{(i)})$. This results in the updating scheme by the leader node C_m:

$$\begin{aligned}
&\boldsymbol{\mu}_{t|t-1}^{(i)} = f\left(\boldsymbol{\mu}_{t-1|t-1}^{(i)}, z_t^{(i)}\right) \\
&\boldsymbol{\Sigma}_{t|t-1}^{(i)} = \boldsymbol{F}_t\left(z_t^{(i)}\right)\boldsymbol{\Sigma}_{t-1|t-1}^{(i)}\boldsymbol{F}_t\left(z_t^{(i)}\right)^T + \boldsymbol{B}\left(z_t^{(i)}\right)\boldsymbol{B}\left(z_t^{(i)}\right)^T \\
&\boldsymbol{S}_t^{(i)} = \boldsymbol{H}_t\left(z_t^{(i)}, m\right)\boldsymbol{\Sigma}_{t|t-1}^{(i)}\boldsymbol{H}_t\left(z_t^{(i)}, m\right)^T \\
&\quad + \boldsymbol{D}_m\left(z_t^{(i)}\right)\boldsymbol{D}_m\left(z_t^{(i)}\right)^T \\
&\boldsymbol{y}_{t|t-1}^{(i)} = h\left(\boldsymbol{\mu}_{t|t-1}^{(i)}, m, z_t^{(i)}\right) \\
&\boldsymbol{\mu}_{t|t}^{(i)} = \boldsymbol{\mu}_{t|t-1}^{(i)} + \boldsymbol{\Sigma}_{t|t-1}^{(i)}\boldsymbol{H}_t\left(z_t^{(i)}, m\right)^T \boldsymbol{S}_t^{-1(i)}\left(\boldsymbol{y}_t - \boldsymbol{y}_{t|t-1}^{(i)}\right) \\
&\boldsymbol{\Sigma}_{t|t}^{(i)} = \boldsymbol{\Sigma}_{t|t-1}^{(i)} - \boldsymbol{\Sigma}_{t|t-1}^{(i)}\boldsymbol{H}_t\left(z_t^{(i)}, m\right)^T \\
&\quad \times \boldsymbol{S}_t^{-1(i)}\boldsymbol{H}_t\left(z_t^{(i)}, m\right)\boldsymbol{\Sigma}_{t|t-1}^{(i)}
\end{aligned} \tag{14}$$

where the state transition and observation matrices are defined by the partial derivatives:

$$\begin{cases} \boldsymbol{F}_t\left(z_t^{(i)}\right) = \left.\frac{\partial f}{\partial x}\right|_{\boldsymbol{\mu}_{t-1|t-1}^{(i)}} \\ \boldsymbol{H}_t\left(z_t^{(i)}, m\right) = \left.\frac{\partial h}{\partial x}\right|_{\boldsymbol{\mu}_{t|t-1}^{(i)}} \end{cases}$$

The predictive density is then simply approximated by:

$$p\left(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1}, z_{0:t}^{(i)}\right) = \mathcal{N}\left(\boldsymbol{y}_t\,;\, \boldsymbol{y}_{t|t-1}^{(i)}, \boldsymbol{S}_t^{(i)}\right)$$

Concerning the spatial updates (see Section 3.2 for details) by the collaborative nodes, a similar approximating scheme as (14) can be adopted (without a temporal prediction step) where $\boldsymbol{\mu}_{t|t-1}^{(i)}$ and $\boldsymbol{\Sigma}_{t|t-1}^{(i)}$ are replaced by $\boldsymbol{\mu}_{t|t}^{(i,l)}$ and $\boldsymbol{\Sigma}_{t|t}^{(i,l)}$ sent by the node C_{l-1}. Only the partial derivatives of the observation function $h(m, z_t)$ is needed. The approximating observation matrix is then computed as the Jacobian applied to $\boldsymbol{\mu}_{t|t}^{(i,l)}$:

$$\boldsymbol{H}_t\left(z_t^{(i)}, l\right) = \left.\frac{\partial h}{\partial x}\right|_{\boldsymbol{\mu}_{t|t}^{(i,l)}}$$

5 Numerical results

The proposed algorithm is applied on synthetic data generated according to the distributed jump Markov linear state space model (1). The system has three hidden discrete states ($K = 3$). The transition stochastic matrix is set as follows:

$$P(z_t \mid z_{t-1}) = \begin{pmatrix} 0.1 & 0.5 & 0.4 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.3 & 0.6 \end{pmatrix}$$

where the occurrence of the first state is lower than the second and third states. The matrices $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}_m, \boldsymbol{D}_m)$ are set at random according to Gaussian distributions. The dimension of the hidden continuous state is set to $n_x = 2$ and the dimension of the observation is set to $n_y = 6$. The number of particles sequentially sampled at the leader nodes is $N = 100$. We have fixed severe communication constraints such that the maximum allowed collaborating nodes is 3 (leader node + 2 spatially collaborating nodes). Under these communication constraints, the resolution of the KD-tree approximation is only one Gaussian for each discrete state. In other words, the leader node communicates only 3 vector means and 3 covariances representing the Kalman mixture to its spatially collaborating nodes.

Figure 10 shows the estimated a posteriori marginal discrete state probabilities $p(z_t \mid \boldsymbol{y}_{1:t})$. Note that, at each time step, the discrete states are not a posteriori equally distributed, avoiding ambiguity when estimating the states. In Figure 11, the MAP estimate of the discrete states is plotted with the true discrete states. Note the accuracy of the proposed collaborative online detection, which is about 88%. The centralised Rao-blackwellised particle filter is also applied on the same set of data. Figure 12 shows the MAP discrete state estimates with the centralised processing whose classification precision is the same as the collaborative distributed algorithm (88%). This corroborates the efficiency of the proposed strategy under severe communication constraints. In order to further illustrate the effectiveness of the spatial collaboration strategy, Figure 13 shows the detection performance of a distributed Rao-Blackwellised particle filter with only one leader node (no collaborator nodes). Note that the performance has degraded to (68%).

Figure 10 A posteriori probabilities of the system discrete state

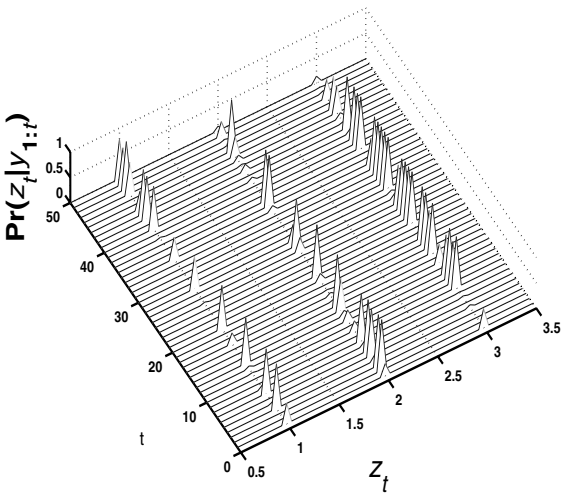


Figure 11 Maximum a posteriori estimate of the system discrete state

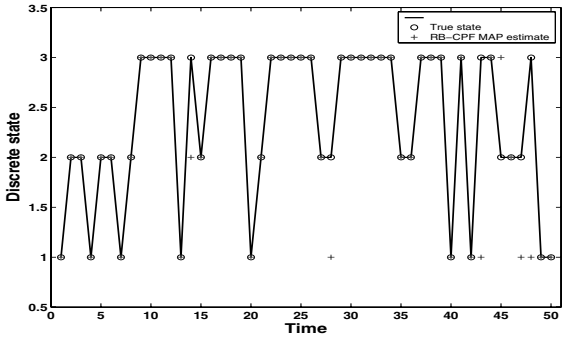


Figure 12 Maximum a posteriori estimate of the system discrete state with a centralised processing

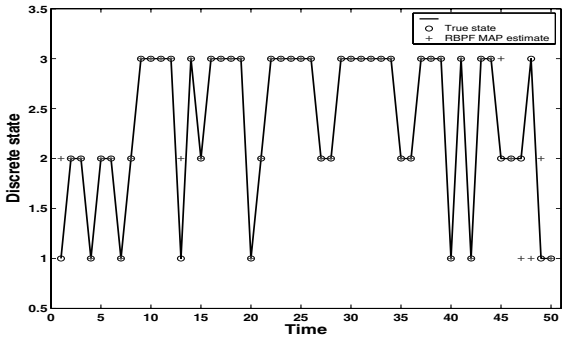
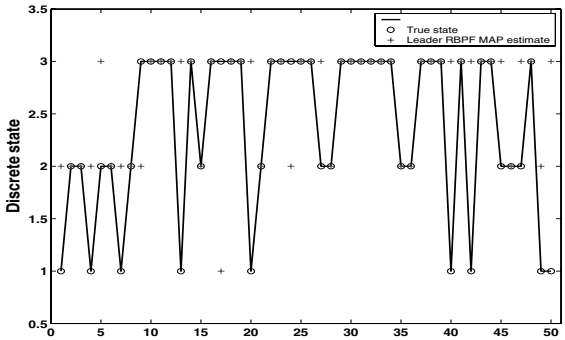


Figure 13 Maximum a posteriori estimate of the system discrete state with only one leader node



6 Conclusion

We have proposed a distributed and collaborative version of the Rao-Blackwellised particle filter for online change detection. At each time step t, the selected leader node updates the posterior probability of the system discrete state. This update is based on a spatial collaboration with other nodes called collaborator nodes. The nodes exchange only sufficient statistics (second order moments). The temporal selection of the leader node is based on a trade-off between information data *relevance* and compression loss under communication constraints. Similarly, the spatial selection of collaborator nodes path is recursively designed and relies on a trade-off between information *complementarity* and compression loss under the communication constraints.

In this paper, we have assumed a jump Markov linear state space model for the observed system. The matrices involved in this model are assumed to be known (estimated in a training

step). We are currently working on the extension to non-linear models and the possibility to incorporate an unsupervised estimation of the model parameters in distributed fashion adapted to the wireless sensor network.

References

Andrieu, C., Doucet, A., Singh, S. and Tadic, V. (2004) 'Particle methods for change detection, system identification, and control', *Proceedings of the IEEE*, March, Vol. 92, No. 3, pp.423–438.

de Freitas, N. (2002) 'Rao-Blackwellised particle filtering for fault diagnosis', *IEEE Aerospace*.

Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977) 'Maximum likelihood from incomplete data via the em algorithm', *Journal of the Royal Statistical Society, Series B*, Vol. 39, pp.1–38.

Doucet, A., Godsill, S. and Andrieu, C. (2000) 'On sequential Monte Carlo sampling methods for Bayesian filtering', *Statistics and Computing*, Vol. 10, No. 3, pp.197–208.

Doucet, A., de Freitas, N. and Gordon, N. (2001) *Sequential Monte Carlo Methods in Practice*, Springer-Verlag.

Doucet, A., Vo, B., Andrieu, C. and Davy, M. (2002) 'Particle filtering for multitarget tracking and sensor management', *Proceedings of the International Conference on Information Fusion*, Vol. 1, pp.474–481.

Ihler, A., Fisher III, J. and Willsky, A. (2004) 'Using sample-based representations under communications constraints', *Technical Report 2601*, MIT, Laboratory for Information and Decision Systems.

Ihler, A., Fisher III, J. and Willsky, A. (2005) 'Particle filtering under communications constraints', *Proceedings of Statistical Signal Processing (SSP)*.

Li, X. and Jilkov, V. (2003) 'Survey of maneuvering target tracking. Part i: dynamic models', *IEEE Transactions on AES*, Vol. 39, No. 4, pp.1333–1364.

Li, X. and Jilkov, V. (2005) 'Survey of maneuvering target tracking. Part v: multiple-model methods', *IEEE Transactions AES*, Vol. 41, No. 4, pp.1255–1321.

Pasha, A., Vo, B., Tuan, H. and Ma, W. (2006) 'Closed form filtering for linear jump Markov models', *Proceedings of FUSION*, Florence.

Roweis, S. and Ghahramani, Z. (1999) 'A unifying review of linear Gaussian models', *Neural Computation*, Vol. 11, No. 2, p.305.

Special Issue (2002) 'Collaborative information processing', *IEEE Signal Processing Magazine*, Vol. 19, No. 2.

Special Issue (2005) 'Special issue on self-organizing distributed collaborative sensor networks', *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 4.

Special Issue (2006) 'Distributed signal processing in sensor networks', *IEEE Signal Processing Magazine*, Vol. 16.

Sheng, X., Hu, Y. and Ramanathan, P. (2005) 'Distributed particle filter with GMM approximation for multiple targets localization and tracking ion wireless sensor network', *Information Processing in Sensor Networks. IPSN'05*, pp.181–188.

Washington, R. (2000) 'On-board real-time state and fault identification for rovers', *IEEE International Conference on Robotics and Automation*.

Note

[1] The Kullback-Leibler divergence between two Gaussians $(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ is $1/2(\mathrm{tr}\left[\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_2^{-1}\right] - \log \boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_2^{-1} - m + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2))$.