Gradient Based Sequential Markov Chain Monte Carlo for Multi-target Tracking with Correlated Measurements

Roland Lamberti, François Septier, Naveed Salman, Lyudmila Mihaylova

Abstract—Measurements in Wireless Sensor Networks (WSNs) are often correlated both in space and in time. This paper focuses on tracking multiple targets in WSNs by taking into consideration these measurement correlations. A Sequential Markov Chain Monte Carlo (SMCMC) approach is proposed in which a Metropolis within Gibb’s refinement step and a likelihood gradient proposal are introduced. This SMCMC filter is applied to case studies with cellular network Received Signal Strength (RSS) data in which the shadowing component correlations in space and time are estimated. The efficiency of the SMCMC approach compared to particle filtering, as well as the gradient proposal compared to a basic prior proposal, are demonstrated through numerical simulations. The accuracy improvement with the gradient-based SMCMC is above 90% when using a low number of particles. Thanks to its sequential nature, the proposed approach can be applied to various WSN applications, including traffic mobility monitoring and prediction.

Index Terms—Multiple Target Tracking, Correlated shadowing, Sequential Markov Chain Monte Carlo, Gradient-based likelihood proposal.

I. INTRODUCTION

Tracking multiple mobile targets is a challenging task which has applications in a number of fields, including that of wireless cellular communication networks and mobility prediction for intelligent transportation systems. In this area, the main structure of a system will feature target nodes whose kinematic states are unknown and need to be estimated; and sensor nodes receiving some type of noisy information about the target nodes, from which an estimation of their states can be inferred.

A variety of methods have been developed in order to solve this localization problem. The more common range-based methods (as opposed to range-free methods) depend on the distances between nodes, through measurements of received signal strengths (RSS), signal time-of-arrivals (ToA) [1] or angle-of-arrivals (AoA) [2] originating from the targets. Both ToA and AoA approaches allow for accurate distance estimations leading to good localization, however ToA requires synchronized clocks on the target nodes, while AoA requires an array of antennas and is still sensitive to errors due to multipath, making them costly solutions. The received signal strength technique [3] is a much more direct and simple approach, with low implementation costs; as such, it is a recurrent subject of performance optimization attempts. Taking into account the shadowing correlation (Gudmunson’s model [4], [5]) between different nodes (targets or sensors), which capitalizes on the fact that in a given environment, closely areas present more or less similar behaviors with regard to shadowing, and may thus be modeled as highly correlated, is one such way of improving this technique. A few examples of research include [6] which studies the combination of measurement correlation and shrinkage estimation of the covariance matrix for significant performance improvements, but is limited to the static case. In [7]–[10] the measurement correlations are taken into account and refined particle filtering (or Sequential Importance Resampling - SIR) algorithms are implemented. This results in high localization accuracy, however these algorithms inherently suffer from the limitations of the particle filtering approach. Although this approach is known to be an effective way of solving non-linear problems, it performs poorly in high-dimensional state-spaces [11].

In this paper, we present a novel Bayesian solution to tracking problems with correlated measurements based on an advanced Monte-Carlo algorithm. Firstly, we take into account the shadowing correlations both spatially and in time, that is, between either current or past positions of any targets. This allows for performance improvements both due to the correlations in time between positions of a single target, and due to the correlations between trajectories of different targets which may cross at some point in time. Finally, in order to efficiently solve the Bayesian tracking problem, we propose to use a Sequential Markov Chain Monte Carlo (SMCMC) algorithm. This technique, which is still largely under-exploited in the signal processing literature, allows for more robust and overall better performance than the more classical particle filtering, especially in high-dimensional systems [12]–[14]. The combination of these two features thus has a good potential for overall robustness in tracking performance in a wide range of scenarios. Preliminary results, including experimental analyses regarding the benefits of taking into account the spatio-temporal shadowing correlation, are already reported in our previous work [15]. We now detail and justify our choice of the SMCMC methodology and complete these results by replacing the prior proposal density of the Gibbs refinement step with a likelihood gradient proposal. This allows to better capitalize on informative measurements and guides...
the particles towards high-likelihood zones, increasing the efficiency of the algorithm. Finally, we present new simulation results demonstrating the benefits of this distribution over the prior and further justifying the superiority of SMC/MC over SIR in our model, when both have similar sampling costs and use the same proposal densities.

The paper is structured as follows. Section II details the choice of the target and observation models. Section III explains the Bayesian framework used as well as the SMC/MC solution, and Section IV details how to integrate the likelihood gradient in the proposal density of the Gibbs refinement step. Simulation results using synthetic data on the superiority of SIR in our model, when both have similar sampling costs and prior and further justifying the superiority of SMCMC over SLAM are presented in Section V, while Section VI highlights the main conclusions of this work.

II. TARGET AND OBSERVATION MODELS

A. Target state and motion models

In a 2-dimensional (2-D) network, the kinematic state of a single target at discrete time step \( t \) may be defined as a vector of positions and velocities \( \mathbf{x}_t = [x_{t,x}, x_{t,y}, \dot{x}_{t,x}, \dot{x}_{t,y}]^T \), although it could also contain accelerations or other variables of interest. Here, \( \mathbb{N}^* \) represents the set of all natural numbers excluding 0; the kinematic state \( \{\mathbf{x}_t \}_{t \in \mathbb{N}^*} = \{(x_{t,1})^T, (x_{t,2})^T, \ldots, (x_{t,N})^T\}_{t \in \mathbb{N}^*} \) of a set of \( N \) targets is considered to be a stochastic Markov process such that at any time step \( t \), the transition probability density function (pdf) \( p(\mathbf{x}_{t+1} | \mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{x}_t) \) is known and can either be evaluated point-wise or sampled from.

B. Correlated observation model

Consider a set of \( N \) targets evolving from time 1 to time \( T \), \( \mathbb{x}_{1:T,1:N} \), and a set of \( M \) immobile sensors \( \mathbb{s} = [s^1, \ldots, s^M] \) where \( s^i = [s^i_x, s^i_y]^T \) is the position of the \( i \)-th sensor for \( i \in \{1, \ldots, M\} \). We suppose that both \( N \) and \( M \) are fixed and known in this model. Throughout the paper, with the exception of square functions, superscripts will be used to denote a sensor \( i \), a particle \( n \) or a Monte Carlo run \( l \) (in the simulations section), and subscripts will mostly be used to denote a time step \( t \), a target \( j \) or a component \( \{x, y, \dot{x}, \dot{y}\} \). At time \( t \in \{1, \ldots, T\} \), a target \( j \in \{1, \ldots, N\} \) transmitting a signal with power \( P_{t,j} \) causes a sensor \( i \) to receive a signal with power \( P^i_{t,j} \) (the data association problem is assumed to be resolved, for example it could be assumed that the targets emit during preassigned epochs). The corresponding path-loss can be expressed as

\[
\mathcal{L}_{t,j} = 10 \log_{10} P_{t,j} - 10 \log_{10} P^i_{t,j}
\]

The observed path-loss signal \( y_{t,j}^i \) at the sensor can empirically be modeled [16]-[19] as

\[
y_{t,j}^i = \mathcal{L}_{t,j} + \mathcal{L}_0 = 10 \log_{10} d(\mathbf{x}_{t,j}, \mathbb{s}^i) + u_{t,j}^i
\]

where

\[
d(\mathbf{x}_{t,j}, \mathbb{s}^i) = \sqrt{(x_{t,j,x} - s^i_x)^2 + (x_{t,j,y} - s^i_y)^2}
\]
targets from different time steps, and can be expressed in blocks as:

\[
R_t^i = \begin{bmatrix}
\rho\left(x_{1:t-1:N}, x_{1:i:N}\right) & \cdots & \rho\left(x_{1:t-1:N}, x_{t:i:N}\right) \\
\vdots & \ddots & \vdots \\
\rho\left(x_{1:t-1:N}, x_{1:i:N}\right) & \cdots & \rho\left(x_{1:t-1:N}, x_{t:i:N}\right)
\end{bmatrix}
\]  

(6)

Finally, measurements at each sensor are supposed to be independent from measurements at all other sensors - this is justified by considering scenarios where the sensor positions are immobile and sufficiently far apart from each other thus inducing little to no correlation. Thus the joint pdf of the measurements from several sensors can be calculated as the product of the pdfs of the measurements from each one of these sensors:

\[
p(y_{1:t-1:N}^M|x_{1:t-1:N}) = \prod_{i=1}^M p(y_{1:t-1:N}^i|x_{1:t-1:N}).
\]  

(7)

### III. PROPOSED BAYESIAN SOLUTION

#### A. Recursive inference

The aim of the Bayesian inference is to recursively estimate the states of the sequence of targets by computing the expectation of its joint posterior density. At time \( t \), this posterior density can be deduced recursively as a function of its expression from the previous time step \( t-1 \):

\[
p(x_{1:t-1:N}|y_{1:t-1:N}^M) \propto 
\prod_{i=1}^M p(y_{1:t-1:N}^i|y_{1:t-1:N}^i, x_{1:t-1:N})p(x_{1:t-1:N}|x_{t-1:N}) 
\]  

(8)

However, this density is intractable mainly due to the nonlinear relationship of the hidden states in the observations and therefore needs to be approximated. In this posterior distribution of interest, the likelihood is obtained from Eq. (5) using classical conditional properties of the multivariate Gaussian distribution:

\[
p(y_{1:t-1:N}^i|y_{1:t-1:N}^i, x_{1:t-1:N}) = \mathcal{N}(\mathbf{\mu}_i, \Sigma_i),
\]  

(9)

where

\[
\mathbf{\mu}_i = \mu_2 + \sum_{j=1}^i \Sigma_{2,1}\Sigma_{1,1}^{-1}(z - \mu_1),
\]

\[
\Sigma_i = \Sigma_{2,2} - \sum_{j=1}^i \Sigma_{2,1}\Sigma_{1,1}^{-1}\Sigma_{1,2},
\]  

(10)

with

\[
z = y_{1:t-1:N},
\]

\[
\mu_1 = [f'(x_{1:N}), \cdots, f'(x_{1:N}), \cdots, f'(x_{t-1:N}), \cdots, f'(x_{t-1:N})]^T,
\]

\[
\mu_2 = [f'(x_{1:N}), \cdots, f'(x_{t:N})]^T,
\]

\[
\Sigma_{1,1} = R_t^i,
\]

\[
\Sigma_{2,1} = [\rho'(x_{1:N}, x_{1:N}), \cdots, \rho'(x_{1:N}, x_{t-1:N})],
\]

\[
\Sigma_{1,2} = [\rho'(x_{1:N}, x_{1:N}), \cdots, \rho'(x_{t-1:N}, x_{1:N})]^T,
\]

\[
\Sigma_{2,2} = \rho'(x_{1:N}, x_{1:N}).
\]  

(11)

Given that any measurement is dependent on all of the other measurements at any time step, the sizes of the mean vector and covariance matrix of the observation defined in Eq. (5) grow with time. As a consequence, the cost of the computation of the likelihood in Eq. (9) that will be required in the filtering algorithm increases with time. In this paper, we therefore propose to use a strategy in order to have a constant computational cost by using a restriction of the size of the used history of positions, for instance through a sliding time window. One drawback of such an approximation is that it could imply the loss of interesting correlation information in cases where some targets approach past trajectories of some other targets (or themselves). Indeed, although the most significant correlations may often intuitively be the ones between positions of a same target at close time steps, simply due to their inherent proximity compared to the proximity of positions from different targets, this still depends on the chosen target motion model. It is likely to be the case if the targets move completely independently, which is clearly not always a correct assumption in real scenarios. However, the sliding time window approximation may also help in avoiding possible numerical problems in the evaluation of the likelihood (due to the inversion of a large covariance matrix). By defining the size of this sliding time window as \( t_{\text{window}} \), the computation of the likelihood in Eq. (9) will involve a modified covariance matrix of size \( (N \times (t_{\text{window}} + 1), N \times (t_{\text{window}} + 1)) \) since \( \forall (j,k) \in \{1, \ldots, N\} \), we will consider \( \text{Corr}(x_{r,j}, x_{r,k}) = 0 \) if \( |r - t| > t_{\text{window}} \).

In a single target scenario, the authors in [10] propose to use a sequential Monte-Carlo method, known as particle filter, in order to infer the single target characteristics given the observations. However, this method suffers from intrinsic limitations in high-dimensional systems ([11], [21]), as the number of samples needs to increase exponentially with the variance of the weights (which is typically a linear function of the state dimension) so as to ensure that not only a single weight will be non-null. In order to obtain a more efficient algorithm for multiple target tracking, we thus propose an alternative solution based on a more advanced methodology known as Sequential Markov Chain Monte Carlo (SMCMC) [12].

#### B. The proposed SMCMC algorithm

Traditionally, Markov chain Monte Carlo (MCMC) methods are used to draw samples from probability distributions in a non-sequential setting. The advantages of MCMC over Importance Sampling (IS), which is the main principle used in particle filters, are that it is generally more effective in high-dimensional systems, and also easier to design for complex distributions. Recently, sequential MCMC schemes were proposed in the literature - see ([13], [14]) for a review. Although there are no theoretical proofs of this yet, SMCMC has experimentally proven to be much more efficient than particle filtering (including particle filtering augmented with MCMC resample-moves [22]) at handling high-dimensional settings, because of the sequential nature of the algorithm allowing local exploration of the state-space with a single Markov chain.
at a given time step, thus reaching more relevant regions of the state-space in terms of posterior. On the contrary, a particle filter augmented with MCMC resample-moves will still suffer from the limitations of the importance sampling and resampling steps (which SMC/MCMC omits entirely), and attempt to remedy them by constructing several independent Markov chains which is much less efficient than the SMC/MCMC approach.

More specifically, the sequential MCMC (SMCMC) is a powerful sequential methodology for filtering that targets the joint posterior distribution defined in our case by Eq. (8). At a given time step, we use a MCMC procedure to make inference from this complex distribution (which is fixed for this time step). However, since we do not have a closed form representation of the posterior distribution $p(x_{1:t-1,1:N}|y_{1:t-1,1:N})$ at time $t-1$, it will be approximated by an empirical distribution based on the current particle set:

$$p(x_{1:1:M,1:1:N}|y_{1:1:M,1:1:N}) \approx \frac{1}{N_p} \sum_{j=1}^{N_p} \delta_{x_{1:1:M,1:1:N}}(x_{1:1:M,1:1:N})$$

where $N_p$ is the number of particles and $(j)$ the particle index. Then, by plugging this particle approximation into Eq. (8), we obtain

$$\pi(x_{1:1:N}) \propto \frac{1}{N_p} \sum_{j=1}^{N_p} \left( \prod_{i=1}^{M} p(y_{i,1:N}|y_{1:i-1,1:N},x_{1:i-1,1:N},x_{i,1:N}) \right) \times p(x_{1:1:N}|x_{1:i-1,1:N}) \delta_{x_{1:1:M,1:1:N}}(x_{1:1:M,1:1:N})$$

where $\pi(x_{1:1:N})$, an empirical approximation of the true posterior $p(x_{1:1:N}|y_{1:1:M,1:1:N})$ based on the particle set $x_{1:1:N}$, is the target distribution of the Markov chain at time step $t$. At a given iteration $n$ of the Markov chain, the variables $x_{1:i-1,1:N}$ are to be drawn according to a uniform discrete distribution (selected uniformly from the set $x_{1:i-1,1:N}$) whereas $x_{1:1:N}$ is then drawn from a continuous distribution conditional to this previous sample, hence the designation "Joint Draw" for this procedure.

Then, having made many joint draws from Eq. (13) using an appropriate MCMC scheme, the converged MCMC output for variable $x_{1:1:N}$ can be extracted to give an updated particle approximation of $p(x_{1:1:N}|y_{1:1:M,1:1:N})$ to be used at the next time iteration. More specifically, after a burn-in period of $N_{burn}$, keep every MCMC output $x_{1:1:N}^{(j)} = x_{1:1:N}$ as the new particle set for the posterior distribution (the notation $(j)$ is meant to include only the $N_p$ particles that are considered to be after the burn-in period, while $n$ may refer to any particle of the chain). In this way, sequential inference can be achieved.

In addition to this procedure, we choose to perform an additional refinement step in order to improve the quality of the samples corresponding to time $t$. Several block sampling structures could be considered [23], but in this paper, we opt to sample successively each of the individual targets using a series of Metropolis-within-Gibbs steps, which consists in drawing new samples component-wise, that is in our application, target-wise, conditionally to all other targets, and choosing whether to accept them. This allows to carefully move each component of our particles towards more interesting regions of the state-space, using densities that are focused on each component as opposed to the joint density used in the previous step. It should be emphasized that our block sampling Gibbs step is in fact target-wise and thus multivariate, rather than univariate coordinate-wise. This is much more efficient since in our setting there is a strong correlation between coordinates of a single target, which means sampling a single coordinate conditionally to other coordinates of the same target would be degenerated (close to deterministic).

To further take advantage of this approach, we also use a Langevin-type gradient proposal density for sampling in the refinement (this aspect will be detailed in Section IV). It is interesting to note that performing both a MH Joint Draw and next a component-wise Gibbs refinement is complementary. Indeed, the Gibbs step improves upon the previous joint sampling. However, if we were to omit the initial Joint Draw and only perform this refinement step, the sampling might become degenerated if there is high correlation between targets’ measurements since we use a density conditional on all targets other than the current component [12]. Additionally, while in our chosen algorithm we only perform a single MH Joint Draw step, several iterations could potentially help improve the mixing for the Markov chain [24] (once again especially when there is strong correlation between blocks, which in our case would correspond to targets in close proximity).

In short, at time $t$ and at the $n$-th MCMC iteration, the following procedure is thus performed to obtain samples from $p(x_{1:1:N}|y_{1:1:M,1:1:N})$:

- Make a joint draw for $x_{1:1:N}$ using a Metropolis-Hastings step.
- Refine the hidden state at current time $t$, $x_{t,1:N}$, using a series of Metropolis-Hastings-within-Gibbs steps.

The complete proposed algorithm is summarized in Alg. 1 (which also includes the Langevin-type gradient proposal explained in Section IV).

Following the acquisition of this set of particles (selected after a burn-in period) asymptotically drawn according to the density $p(x_{1:1:N}|y_{1:1:M,1:1:N})$, the target state estimation at time $t$ can be performed using the minimum mean square error criterion as the mean of the particles, which corresponds to the empirical approximation of the expectation of the marginalized posterior density $p(x_{1:1:N}|y_{1:1:M,1:1:N})$:

$$\hat{x}_{t,1:N} = \frac{1}{N_p} \sum_{j=1}^{N_p} x_{1:1:N}^{(j)}$$

IV. GRADIENT-BASED PROPOSAL DENSITY

The choice of a relevant proposal density to propagate the kinematic states in the Metropolis-within-Gibbs refinement steps is crucial for the algorithm to be able to "lock-on" to the targets. In our case, using the prior probability density may
At time $t$, to compute the $n$-th SMCMC particle trajectory $x_{t+1}^{n,1} \sim N$:

Data: $y_{t+1}^{M,N}$ (all of the measurements available at time $t$), $x_{t+1}^{n,N}$ (the particle set constituting the empirical approximation of the posterior density from time $t-1$), $x_{t+1}^{n,N}$ (the result of the previous step $n-1$ of the algorithm)

Joint Draw using Metropolis-Hastings
- Randomly select a joint particle trajectory $x_{t+1}^{n,1} \sim N$ by sampling it from the empirical measure of $p(x_{t+1}^{n,1} \mid y_{t+1}^{N})$ obtained at the previous time iteration:
  \[ \tilde{x}_{t+1}^{n,1,N} \sim \frac{1}{N_{p}} \sum_{j=1}^{N_{p}} \delta_{\tilde{y}_{t+1}^{j,1,N}}(x_{t+1}^{n,1,N}) \]  
  (15)
- Draw a random sample for the current $t$-th time step:
  \[ \tilde{x}_{t+1}^{n,1,N} \sim p(\cdot \mid \tilde{x}_{t+1}^{n-1,1,N}) \]  
  (16)
- Calculate the acceptance ratio which compares the likelihood given $\tilde{x}_{t+1}^{n,1,N}$ with the likelihood given $x_{t+1}^{n-1,1,N}$ (which is the one from the previous iteration $n-1$):
  \[ \alpha = \min \left( 1, \frac{\prod_{i=1}^{M} p(y_{t+1}^{i} \mid x_{t}^{n-1,1,N}, x_{t+1}^{n-1,1,N})}{\prod_{i=1}^{M} p(y_{t+1}^{i} \mid x_{t+1}^{n,1,N}, x_{t+1}^{n-1,1,N})} \right) \]  
  (17)
- Accept this proposed particle or reject it:
  \[
  \begin{array}{ll}
    \text{if } (a < \alpha) \\
    \text{accept the particle, thus } x_{t+1}^{n,1,N} := \tilde{x}_{t+1}^{n,1,N} \\
    \text{else} \ 
    \text{reject the particle, thus } x_{t+1}^{n-1,1,N} := x_{t+1}^{n-1,1,N}.
  \end{array}
  \]

Refinement using Metropolis-within-Gibbs
- Successively sample each target:
  for $b = 1$ to $N$ do
    - Define $\bar{x}_{t+1}^{n,1,N} := x_{t+1}^{n,1,N}$.
    - Draw a new sample from the gradient-based proposal density $q$ in (19), for the $b$-th target at current time $t$:
      \[ \tilde{x}_{t,b} \sim q(\cdot \mid \bar{x}_{t,b}^{n,1,N}) \]  
      (18)
    - Calculate the acceptance ratio as $\alpha = \min(1, \beta)$ where $\beta$ is from equation (28), with the modified particle $\bar{x}_{t+1}^{n,1,N}$.
    - Accept this proposed particle or reject it:
      \[
      \begin{array}{ll}
        \text{if } (a < \alpha) & \text{accept the particle, } x_{t,b}^{n,1,N} := \tilde{x}_{t,b} \\
        \text{else} & \text{reject the particle, do not update the } b \text{-th block in } x_{t+1}^{n,1,N}.
      \end{array}
      \]
  end

Output: Sample $x_{t+1}^{n,1,N}$

Algorithm 1: Proposed SMCMC for multi-target tracking.

We choose a Langevin-type ([25], [26]) proposal density $q(\cdot)$ which is based on the gradient of the target density (which includes the likelihood). For a target $k$, at the $n$-th step of the Metropolis-Hastings algorithm, the Gibbs refinement sample will be drawn as follows:

\[ \tilde{x}_{t,k} \sim q(\cdot \mid \tilde{x}_{t,k}^{n}) = N(\mu_{k}^i, \Sigma^i) \]  
(19)

where $\tilde{x}_{t,k}^{n}$ results from the Metropolis-Hastings Joint Draw, and

\[ m = \frac{h}{2} \nabla \left( \log \Pi(x_{t,k}) \right) \mid _{x_{t,k} = \tilde{x}_{t,k}^{n}} \]  
(20)

$h$ being a step which needs to be chosen empirically so that the performance of the algorithm is optimal, and $\Pi(x_{t,k})$, proportional to the conditional posterior density $p(x_{t,k} \mid y_{t+1}^{1,M}, x_{t-1}^{n-1,1,N}, x_{t+1}^{n,1,N})$ for the current target of interest, being the product of the corresponding likelihood and prior terms, derived from (8); thus

\[ \Pi(x_{t,k}) = \prod_{i=1}^{M} p(y_{t+1}^{i} \mid x_{t}^{n-1,1,N}, x_{t+1}^{n,1,N}, x_{t+1}^{n,1,N} \setminus k), x_{t+1}^{n,1,N}) \]  
(21)

Using (9) and (10),

\[ \log \Pi(x_{t,k}) = \sum_{i=1}^{M} \log p(y_{t+1}^{i} \mid \mu_{k}^{i}, \Sigma^{i}) + \log p(x_{t,k} \mid x_{t}^{n-1,1,N}, x_{t+1}^{n,1,N} \setminus k). \]  
(23)

We need to calculate the gradient of

\[ \log N(y_{t+1}^{i} \mid \mu_{k}^{i}, \Sigma^{i}) = -\frac{1}{2} \log 2\pi - \log |\Sigma^{i}| - \frac{1}{2} (y_{t+1}^{i} - \mu_{k}^{i})^{T} \Sigma_{k}^{-1} (y_{t+1}^{i} - \mu_{k}^{i}) \]  
(24)

with respect to $x_{t,k}$ and evaluate it at $x_{t,k}^{n}$. In this expression, both $\mu_{k}^{i}$ and $\Sigma^{i}$ are dependent on $x_{t,k}$ (see (10) and (11)). In order to simplify this problem, we assume all terms related to covariances (thus $\Sigma^{i}$ as well as the covariance terms in $\mu_{k}^{i}$) to be constant for the derivative. Under this assumption, denoting $A = (y_{t+1}^{i} - \mu_{k}^{i})$ and $S = \Sigma^{i}$, we calculate for each component $c$ of $x_{t,k}^{n}$ in $(x_{t,k,x}, x_{t,k,y}, x_{t,k,z}, x_{t,k,y})^{T}$ (thus $c \in \{x, y, z, \hat{y}\}$):

\[ \frac{\partial A^{T} S A}{\partial x_{t,k,c}} = \left( \frac{\partial A^{T} S A}{\partial \mu_{k}^{i}} \right)^{T} \frac{\partial \mu_{k}^{i}}{\partial x_{t,k,c}} \]  
(25)

with

\[ \frac{\partial A^{T} S A}{\partial \mu_{k}^{i}} = -2S \frac{\partial f_{t,k}(x_{t,k})}{\partial x_{t,k,c}} \]  
(26)

because all terms dependent on targets other than the $k$-th target have derivatives equal to 0. Moreover, the only term derived in $\mu_{k}^{i}$ is $\mu_{2}$ (from (11)), according to our assumption, since the only other term in $\mu_{k}^{i}$ that depends on $x_{t,k}$ is a covariance (namely $\Sigma_{2,1}^{i}$). On the other hand,
the derivatives with respect to velocities being zero since the observations are only dependent on the distances between targets, thus only on target positions. Thus the propagation of velocity values is handled by the prior component alone. The derivative of this prior component is calculated in a similar way, except no assumptions are necessary.

Lastly, the expression of the acceptance ratio for the refinement step needs to be updated as it is dependent on the proposal density \( q \) (with the same notations as Algorithm 1):

\[
\beta = \prod_{i=1}^{M} p(y_{t+1|1:N}^{i}|y_{t-1|1:N}, x_{t+1,1:N}^{i}) \prod_{i=1}^{M} p(y_{t|1:N}^{i}|y_{t-1|1:N}, x_{t,1:N}^{i}) \times \frac{p(x_{t,k}|x_{t-2,1:N}^{i}, x_{t-1,1:N}^{i}, x_{t,1:N}^{i}) q(x_{t-1,k}^{i}|x_{t,k}^{i})}{p(x_{t,k}^{i}|x_{t-2,1:N}^{i}, x_{t-1,1:N}^{i}, x_{t,1:N}^{i}) q(x_{t-1,k}^{i}|x_{t,k}^{i})},
\]

(28)

V. SIMULATION RESULTS

In order to illustrate the performance improvements induced by

- using a SMCMC approach compared to a classical particle filtering approach (Section V-A with a figure showing the superiority of SMCMC the higher the dimension of the state-space is),
- replacing the prior proposal density with the gradient proposal density (Sections V-B and V-C with figures showing the varying amounts of performance gains depending on shadowing noise variance and number of particles),

we assume that each target evolves independently from the others in a field of 16 sensors as illustrated in Fig. 1, according to a nearly constant velocity model [27] [28] which is defined as follows for the \( j \)-th target:

\[
x_{t,j} = F_{t}x_{t-1,j} + u_{t,j}
\]

(29)

where \( F_{t} \) would be a \( 4 \times 4 \) transition matrix and \( u_{t,j} \), a vector of independent realizations of \( N(0, Q_{t}) \) with \( Q_{t} \) a \( 4 \times 4 \) state noise covariance matrix, both \( F_{t} \) and \( Q_{t} \) depending only on the time interval between \( t \) and \( t-1 \). Here \( F_{t} \) and \( Q_{t} \) are defined as:

\[
F_{t} = \begin{bmatrix} I_{2} & \tau_{t} I_{2} \\ 0_{2} & I_{2} \end{bmatrix}, \quad Q_{t} = \sigma_{\text{target}}^{2} \begin{bmatrix} (\tau_{t}^{2}/3) I_{2} & (\tau_{t}^{2}/2) I_{2} \\ (\tau_{t}^{2}/2) I_{2} & \tau_{t} I_{2} \end{bmatrix}
\]

(30)

with \( \tau_{t} \) the time interval between two time steps, which is chosen constant and equal to 1 second, and \( \sigma_{\text{target}}^{2} = 10 \text{ m}^{2} \).

Fig. 1 shows an example of two trajectories created with these parameters and chosen to be confined within a grid of sensors. Due to \( \sigma_{\text{target}}^{2} \) having a relatively large value, the trajectories are chaotic, representing a difficult tracking scenario.

In order to assess the accuracy of the different algorithms when the measurements are randomly generated with standard deviation \( \sigma \) (for the shadowing noise) equal for all target-sensor links, we compute the root mean square error (RMSE) between the estimations and the real positions of the target (the estimations of other variables such as velocities or accelerations are not taken into account), in time, averaged on a number of Monte Carlo (MC) runs:

\[
\text{RMSE}_{t} = \sqrt{\frac{1}{N_{MC}N} \sum_{j=1}^{N} \sum_{l=1}^{N_{MC}} \| \hat{x}_{t,j}^{l} - x_{t,j} \|^2}
\]

(31)

where \( \hat{x}_{t,j}^{l} \) is the estimated state of the \( j \)-th target from the \( l \)-th MC run. Throughout this section, we choose \( N_{MC} = 10 \), \( N_{\text{burn}} = \frac{M}{16} \) and unless specified otherwise we keep a number of sensors \( M = 16 \), a number of targets \( N = 2 \) and a number of time steps \( T = 100 \). For reference, the experiments used Matlab software and a laptop featuring a 2.80 GHz Intel(R) Core(TM) i7-4810MQ CPU.

A. Performance compared to particle filtering

First, we compare the proposed SMCMC algorithm with the particle filtering approach which was proposed in [10] in a similar context for single target tracking. More specifically, the particle filter used in this section is the Sequential Importance Resampling (SIR) [29] in which a resample-move strategy is employed after the resampling stage in order to diversify the set of particles [22]. This strategy uses exactly the same step described as the refinement step in our proposed SMCMC, including the gradient-based proposal, thus allowing for a fair comparison between the two algorithms. For further comparison we also implement a SIR without resample-move but instead using a custom EKF-based proposal [30]. Fig. 2 shows the RMSE obtained with all three algorithms in which \( N_{p} = 200 \) particles are used to do the inference, and the actual target trajectories are generated with \( \sigma_{\text{target}}^{2} = 0.01 \text{ m}^{2} \) thus
much smoother than those from Fig. 1 (Section V-B below studies this more challenging case). In this simulation, the shadowing variance is $\sigma^2 = (0.1)^2$ dB which is a low noise in the context of our experiment, and different numbers of targets are used ($N = 3, 6, 9$). Indeed, the SIR algorithm’s main weakness comes from the degeneration of the importance weights in situations where either the likelihood becomes too informative (with a too small variance) and no longer covers regions where the proposal distribution is high, or more interestingly in difficult situations where the state-space is high-dimensional. In such a multi-dimensional scenario, the results show the significant superiority of the proposed SMCMC against the SIR, with computational times of the same order; additionally, the average RMSE per target remains about the same for SMCMC as the total number of targets increases, while it deteriorates in the case of SIR. For reference, the computational time of this experiment with 3 targets, averaged over the MC runs and the time steps, is approximately $3.4144$ seconds for SIR with resample-move, $1.6141$ seconds for SIR with EKF-based proposal and $3.8601$ seconds for SMCMC. The difference between SIR with resample-move and SMCMC corresponds almost exactly to an increase of 10% in the computational time, while the SMCMC method uses an additional burn-in period of precisely 10% of the total number of particles (in this case, 222 particles including the burn-in, compared to 200 for SIR with resample-move and for SMCMC without burn-in). Thus, SIR with resample-move and SMCMC have a similar computational time for a single particle (and the same remarks still apply with higher numbers of targets).

1 (thus $N = 2$), and either $\sigma^2 = 1^2$ dB (average-to-high measurement noise) or $\sigma^2 = 4^2$ dB (high measurement noise), we run our algorithm with either the prior proposal density or the gradient proposal density from section IV. Fig. 3 shows the resulting RMSE performance with respect to the time step, for $N_0 = 500$. As expected, the estimator using the gradient proposal performs better. For reference, the computational time of this experiment, averaged over the MC runs and the time steps, is approximately $1.5205$ seconds with the prior proposal and $3.2878$ seconds with the gradient proposal, when $\sigma^2 = 1^2$ dB. When $\sigma^2 = 4^2$ dB, the computational times are $1.8128$ second and $3.8967$ seconds, respectively.

Still using this scenario, monitoring the relevance of the proposed particles through values of the acceptance ratio from the Gibbs refinement step in Table I shows that the gradient proposal allows to significantly reduce the rejection of samples due to this step, which is proof that the set of particles propagated is able to reach regions of the state-space featuring much higher likelihood than with the prior proposal. Moreover, from a theoretical point of view, the gain of performance due to using the gradient proposal should decrease when the measurement noise increases, since this density aims to guide the sampled particles towards these regions of high likelihood, because then such regions become very wide and inaccurate. Table I confirms that the gain in acceptance ratio indeed decreases when the measurement noise increases. Fig. 4 from section V-C also confirms this by showing much larger performance gain with lower noise (and also less chaotic trajectories) than in Fig. 3’s difficult scenario.

**B. Performance in difficult, noisy scenarios**

We now demonstrate the benefits of using the gradient proposal distribution presented in Section IV instead of the basic prior proposal in our Gibbs refinement moves, in difficult scenarios. Given the chaotic hidden state trajectories from Fig. 1 (difficult scenario).

**C. Performance in easier scenarios with varying number of particles**

Another benefit of using this gradient-based density can be demonstrated when reducing the number of particles used for the filter. Indeed, using the Gaussian prior density implies that in order to draw only a few particles which will be located

---

Fig. 2: Log-scale performance of our SMCMC algorithm versus a SIR algorithm with the same Gibbs refinement step as resample-move. Low measurement noise ($\sigma^2 = (0.1)^2$ dB) and process noise ($\sigma^2_{\text{target}} = 0.01$ m²)

Fig. 3: Log-scale performance of the SMCMC algorithm using the gradient-based proposal density versus the prior density, for high measurement noise values on the chaotic trajectory set from Fig. 1 (difficult scenario).
TABLE I: Acceptance ratios in the Gibbs refinement step when using either the prior proposal density or the gradient-based one, with varying measurement noise.

<table>
<thead>
<tr>
<th></th>
<th>Prior</th>
<th>Gradient</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma^2 = (0.1)^2$</td>
<td>0.0066</td>
<td>0.0392</td>
</tr>
<tr>
<td>$\sigma^2 = (0.3)^2$</td>
<td>0.0075</td>
<td>0.2669</td>
</tr>
<tr>
<td>$\sigma^2 = (1)^2$</td>
<td>0.0377</td>
<td>0.8198</td>
</tr>
<tr>
<td>$\sigma^2 = (2)^2$</td>
<td>0.1940</td>
<td>0.9117</td>
</tr>
<tr>
<td>$\sigma^2 = (4)^2$</td>
<td>0.4860</td>
<td>0.9500</td>
</tr>
</tbody>
</table>

in regions of interest where the likelihood function is high, it is required to draw a very large number of particles in total, whereas the gradient-based density has no such drawback (as the acceptance ratios from Table I also demonstrate). Fig. 4 shows RMSE values in time for different numbers of particles, displaying how the linear gap of performance between the two proposal densities increases when the number of particles used decreases. For reference, the computational time of this experiment, averaged over the MC runs and the time steps, is approximately 0.24 second when using 25 particles and 1.66 second when using 200 particles.

Fig. 4: Log-scale performance of the algorithm using the gradient-based proposal density versus the prior density, with low measurement noise ($\sigma^2 = (0.1)^2$ dB) and average process noise ($\sigma_{\text{target}}^2 = 1 \text{ m}^2$)

VI. CONCLUSIONS

This paper proposes a sequential MCMC solution to multi-target tracking with RSS measurements, taking into account spatio-temporal correlations between targets and using a gradient-based proposal density for drawing particles in the Gibbs refinement moves. The simulation results show a performance improvement (about 50% increase in accuracy) in any scenario compared to using the prior density as a proposal, and the gain is especially large (above 90%) when using low number of particles or when the model considered features informative measurements. The SMCMC approach is also shown to be superior to particle filtering in this setting when both use the same Gibbs refinement moves.

REFERENCES


