
Proceedings Paper:
De Freitas, A., Septier, F., Mihaylova, L. orcid.org/0000-0001-5856-2223 et al. (1 more author) (2015) How Can Subsampling Reduce Complexity in Sequential MCMC Methods and Deal with Big Data in Target Tracking? In: Proceedings of the 18th International Conference on Information Fusion. 18th International Conference on Information Fusion, 06-09 Jul 2015, Washington DC, USA. IEEE , pp. 134-141.

How Can Subsampling Reduce Complexity in Sequential MCMC Methods and Deal with Big Data in Target Tracking?

Allan De Freitas*, François Septier§, Lyudmila Mihaylova*, Simon Godsill♮
*Department of Automatic Control and Systems Engineering, University of Sheffield, United Kingdom
§ Institute Mines Telecom/Telecom Lille, CRIStAL UMR CNRS 9189, France
♮Department of Engineering, Cambridge University, CB 1PZ, United Kingdom
Emails: *a.defreitas@sheffield.ac.uk, francois.septier@telecom-lille.fr, l.s.mihaylova@sheffield.ac.uk, sjg@eng.cam.ac.uk*

***Abstract*—Target tracking faces the challenge in coping with large volumes of data which requires efficient methods for real time applications. The complexity considered in this paper is when there is a large number of measurements which are required to be processed at each time step. Sequential Markov chain Monte Carlo (MCMC) has been shown to be a promising approach to target tracking in complex environments, especially when dealing with clutter. However, a large number of measurements usually results in large processing requirements. This paper goes beyond the current state-of-the-art and presents a novel Sequential MCMC approach that can overcome this challenge through adaptively subsampling the set of measurements. Instead of using the whole large volume of available data, the proposed algorithm performs a trade off between the number of measurements to be used and the desired accuracy of the estimates to be obtained in the presence of clutter. We show results with large improvements in processing time, more than 40 % with a negligible loss in tracking performance, compared with the solution without subsampling.**

I. Introduction

Flooded with data, richly provided by electronic sensors, the current monitoring systems face the problem of being able to process the data and monitor the phenomenon of interest at the same time. In this paper we consider the problem of target tracking in large volumes of data. There is a wealth of algorithms that can provide sequential estimation of the states of the target, e.g. for details see [1], [2]. In a Bayesian framework, the posterior distribution can be iteratively computed. However, analytically this can be achieved only when the state space model is linear and perturbed by a Gaussian noise. In this case the solution is referred to as the Kalman Filter. There are a large number of techniques which overcome the limitations of the Kalman filter based on the sequential Monte Carlo (SMC) methodology. The seminal work on SMC in target tracking was presented in [3] which was also referred to as the bootstrap particle filter (PF). The bootstrap PF and many variants thereof, broadly referred to as PFs, are commonly favoured techniques in a wide variety of applications due to the filters ability to handle non-linear state space models and/or state space models perturbed by non-Gaussian noise. However, the PF is not void of challenges. Some of the difficulties faced by PFs includes weight degeneracy and sample impoverishment. Although there are variants of the PF which have been proposed to alleviate these issues [4], [5], the PF is still susceptible to degeneracy, and these difficulties are more profound when tracking complex systems.

Markov chain Monte Carlo (MCMC) techniques are a powerful set of algorithms for sampling from a probability distribution. MCMC tecnhiques, such as the Metropolis Hastings (MH) algorithm, have been predominantly used in applications requiring static inference [6]. Recently there has been considerable interest in extending these techniques to sequentially updating the posterior distribution [7], [8]. Sequential MCMC has shown promising results for complex systems. The largest hindrance being long processing times which could limit usage in applications required to run in real time. There have also been several algorithms [9], [10], [11] which have been proposed to help reduce computational complexities when performing static inference with MCMC techniques on large datasets.

In this paper we propose a novel technique which results in an efficient sequential MCMC algorithm when applied in complex systems consisting of a large number of measurements. This is achieved through the combination of sequential inference and adaptive subsampling of the measurements at each time step. We show how the proposed adaptive subsampling sequential MCMC algorithm can be applied to target tracking and illustrate the computational savings it affords.

II. Problem Formulation

Target tracking of a complex system can be considered as sequential state estimation with multiple measurements. This can be achieved in a Bayesian framework by sequentially computing the filtering posterior distribution $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ where $\boldsymbol{x}_k \in \mathbb{R}^{n_{\boldsymbol{x}}}$ is the state vector at time t_k with $k = 1, ..., T \in \mathbb{N}$, and $\boldsymbol{z}_{1:k} = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_k\}$, represents all the measurements received up till time t_k. The measurements received at each time t_k are represented by a set $\boldsymbol{z}_k = \{\boldsymbol{z}_k^1, ..., \boldsymbol{z}_k^{M_k}\}$, where M_k is the total number of measurements and $\boldsymbol{z}_k^i \in \mathbb{R}^{n_{\boldsymbol{z}}}$. The filtering posterior distribution can be recursively updated based

on

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) \propto \int p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})d\boldsymbol{x}_{k-1}, \tag{1}$$

where $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$ is referred to as the likelihood probability density function (pdf), and $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ is referred to as the state transition pdf. An analytical solution to (1) is typically intractable when the state space model is characterised by non-linearities and/or non-Gaussian noise.

A. Sequential Markov Chain Monte Carlo

MCMC methods work by constructing a Markov chain with a desired distribution as the equilibrium distribution. A common MCMC technique used to obtain samples from the equilibrium distribution, $\pi(\boldsymbol{x})$, is the MH algorithm. This is achieved by first generating a sample from a known proposal distribution $\boldsymbol{x}^* \sim q(\cdot|\boldsymbol{x}^{m-1})$. The proposed sample is accepted as the current state of the chain, $\boldsymbol{x}^m$, if the following condition is satisfied

$$u < \frac{\pi(\boldsymbol{x}^*)q(\boldsymbol{x}^{m-1}|\boldsymbol{x}^*)}{\pi(\boldsymbol{x}^{m-1})q(\boldsymbol{x}^*|\boldsymbol{x}^{m-1})}, \tag{2}$$

where u represents a sample from a uniform random variable $u \sim U_{[0,1]}$. Using Bayes' rule and assuming that there are M conditionally independent measurements, $\boldsymbol{z}^i$, results in the further expansion of this expression

$$u < \frac{p(\boldsymbol{x}^*)q(\boldsymbol{x}^{m-1}|\boldsymbol{x}^*)}{p(\boldsymbol{x}^{m-1})q(\boldsymbol{x}^*|\boldsymbol{x}^{m-1})} \prod_{i=1}^{M} \frac{p(\boldsymbol{z}^i|\boldsymbol{x}^*)}{p(\boldsymbol{z}^i|\boldsymbol{x}^{m-1})}. \tag{3}$$

The previous state of the chain is stored as the current state, $\boldsymbol{x}^m = \boldsymbol{x}^{m-1}$, when the proposed sample does not meet this criterion. We further manipulate this expression into a form with the likelihood isolated:

$$\log\left[u\frac{p(\boldsymbol{x}^{m-1})q(\boldsymbol{x}^*|\boldsymbol{x}^{m-1})}{p(\boldsymbol{x}^*)q(\boldsymbol{x}^{m-1}|\boldsymbol{x}^*)}\right] < \sum_{i=1}^{M} \log\left[\frac{p(\boldsymbol{z}^i|\boldsymbol{x}^*)}{p(\boldsymbol{z}^i|\boldsymbol{x}^{m-1})}\right]. \tag{4}$$

In [7] it was proposed to use MCMC methods, specifically the MH algorithm, to target the filtering posterior distribution in (1) as the equilibrium distribution. This allows for the iterative update of an approximation of the filtering posterior distribution by representing $p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})$ with a set of unweighted particles,

$$p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}) \approx \frac{1}{N_p}\sum_{j=1}^{N_p} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(j)}), \tag{5}$$

where N_p is the number of particles and (j) the particle index. This technique was shown to work well in state space models containing a high number of dimensions when compared to techniques relying on importance sampling, however, this direct approach may result in a high computational expense.

It was proposed in [8] to consider targeting the joint filtering posterior distribution of $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k-1}$

$$p(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k}) \propto p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}), \tag{6}$$

as the equilibrium distribution in order to help alleviate the high computational demand. In a similar fashion, an approximation for the joint filtering posterior distribution can be obtained through MCMC methods by representing $p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})$ with a set of unweighted particles. This approach has the advantage of avoiding the direct Monte Carlo computation of the predictive posterior density. Furthermore, the approximation can be marginalised to obtain the filtering posterior distribution of interest.

More specifically, at each time step, the particles are updated with a MH joint draw for $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k-1}$, followed by an individual MH draw for $\boldsymbol{x}_k$. The second step, referred to as the refinement step, is introduced to aid in the mixing of the chain. An appropriate burn in period, N_{burn}, was also introduced to minimize the effect of the initial values of the Markov chain. This results in the definition of the total number of MCMC iterations at each time step, $N = N_p + N_{burn}$. This approach is highlighted by Algorithm 1 and is referred to as standard sequential MCMC. This approach showed promising results in a multi-target environment but is still susceptible to high computational complexity when a substantially large amount of measurements are required to be processed.

B. Adaptive Subsampling

In standard sequential MCMC, it is required to perform $2NM_k$ calculations of the likelihood at each time step. This is highlighted in the computation of the log likelihood ratio, $\Lambda_1^{M_k}(\cdot)$ and $\Lambda_2^{M_k}(\cdot)$, in Algorithm 1. When M_k is very large, the log likelihood ratio becomes the most computationally expensive step of the algorithm. To reduce the computational complexity, we introduce a Monte Carlo (MC) approximation for the log likelihood ratio:

$$\Lambda_1^{S_{m,k}}(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*) = \frac{1}{S_{m,k}}\sum_{i=1}^{S_{m,k}} \log\left[\frac{p(\boldsymbol{z}_k^{i,*}|\boldsymbol{x}_k^*)}{p(\boldsymbol{z}_k^{i,*}|\boldsymbol{x}_k^{m-1})}\right] \tag{7}$$

where the set $\boldsymbol{z}_k^* = \{\boldsymbol{z}_k^{1,*}, ..., \boldsymbol{z}_k^{S_{m,k},*}\}$ is drawn uniformly without replacement from the original set of M_k measurements.

The difficulty which arises is in selecting a minimum value for $S_{m,k}$ that results in a set of subsampled measurements that contain enough information to make the correct decision in the MH step. To overcome this difficulty in standard MCMC for static inference, the authors in [10] proposed to use concentration inequalities which provide a probabilistic bound on how functions of independent random variables deviate from their expectation. In this case, the independent random variables are the log likelihood ratio terms. Thus, it is possible to obtain a bound on the deviation of the MC approximation in (7) from the complete log likelihood ratio:

$$P(|\Lambda_1^{S_{m,k}}(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*) - \Lambda_1^{M_k}(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*)| \leq c_{S_{m,k}}) \geq 1 - \delta_{S_{m,k}} \tag{8}$$

where $\delta_{S_{m,k}} > 0$, and $c_{S_{m,k}}$ is dependent on which inequality is used. There are several inequalities which could be used, in

Algorithm 1 Sequential Markov Chain Monte Carlo

Initialize particle set: $\{\boldsymbol{x}_0^{(j)}\}_{j=1}^{N_p}$
for k = 1,...,T **do**
  for m = 1,...,N **do**
    Joint Draw
    Propose $\{\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*\} \sim q_1\left(\boldsymbol{x}_k, \boldsymbol{x}_{k-1} | \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}\right)$
    Compute $\psi_1(u, \boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1})$
    $= \frac{1}{M_k} \log\left[u \frac{p(\boldsymbol{x}_k^{m-1}|\boldsymbol{x}_{k-1}^{m-1}) p(\boldsymbol{x}_{k-1}^{m-1}|\boldsymbol{z}_{1:k-1})}{p(\boldsymbol{x}_k^*|\boldsymbol{x}_{k-1}^*) p(\boldsymbol{x}_{k-1}^*|\boldsymbol{z}_{1:k-1})} \times \frac{q_1(\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*|\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1})}{q_1(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}|\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*)}\right]$
    Compute $\Lambda_1^{M_k}(\boldsymbol{x}_k^*, \boldsymbol{x}_k^{m-1})$
    $= \frac{1}{M_k} \sum_{i=1}^{M_k} \log\left[\frac{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^*)}{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^{m-1})}\right]$
    if $\Lambda_1^{M_k}(\boldsymbol{x}_k^*, \boldsymbol{x}_k^{m-1}) > \psi_1(u, \boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1})$ **then**
      $\{\boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\} = \{\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*\}$
    else
      $\{\boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\} = \{\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}\}$
    end if
    Refinement
    Propose $\{\boldsymbol{x}_k^*\} \sim q_2\left(\boldsymbol{x}_k | \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right)$
    Compute $\psi_2(u, \boldsymbol{x}_k^*, \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m)$
    $= \frac{1}{M_k} \log\left[u \frac{p(\boldsymbol{x}_k^m|\boldsymbol{x}_{k-1}^m) q_2\left(\boldsymbol{x}_k^*|\boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right)}{p(\boldsymbol{x}_k^*|\boldsymbol{x}_{k-1}^m) q_2\left(\boldsymbol{x}_k^m|\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^m\right)}\right]$
    Compute $\Lambda_2^{M_k}(\boldsymbol{x}_k^m, \boldsymbol{x}_k^*) = \frac{1}{M_k} \sum_{i=1}^{M_k} \log\left[\frac{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^*)}{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^m)}\right]$
    if $\Lambda_2^{M_k}(\boldsymbol{x}_k^*, \boldsymbol{x}_k^m) > \psi_2(u, \boldsymbol{x}_k^*, \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m)$ **then**
      $\boldsymbol{x}_k^m = \boldsymbol{x}_k^*$
    end if
    if $m > N_{burn}$ **then**
      $\boldsymbol{x}_k^{(m-N_{burn})} = \boldsymbol{x}_k^m$
    end if
  end for
end for
$\hat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{1}{N_p} \sum_{j=1}^{N_p} \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(j)})$

this paper we make use of the empirical Bernstein inequality [12], [13], which results in:

$$c_{S_{m,k}} = \sqrt{\frac{2V_{S_{m,k}} \log(3/\delta_{S_{m,k}})}{S_{m,k}}} + \frac{3R_k \log(3/\delta_{S_{m,k}})}{S_{m,k}} \quad (9)$$

where $V_{S_{m,k}}$ represents the sample variance of the log likelihood ratio, and R_k is the range given by

$$R_k = \max_{1 \le i \le M_k} \left\{\log\left[\frac{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^*)}{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^{m-1})}\right]\right\} - \min_{1 \le i \le M_k} \left\{\log\left[\frac{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^*)}{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^{m-1})}\right]\right\} \quad (10)$$

Looking back at the standard sequential MCMC approach, we find that the joint draw is accepted based on the condition $\Lambda_1^{M_k}(\boldsymbol{x}_k^*, \boldsymbol{x}_k^{m-1}) > \psi_1(u, \boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1})$. It is required to relate this expression in terms of the MC approximation of (7). Since the MC approximation is bounded, we can state that it is not possible to make a decision when the value of $\psi_1(u, \boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1})$ falls within the region specified by the bound. Thus it is required that $|\Lambda_1^{S_{m,k}}(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*) - \psi_1(u, \boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1})| > c_{S_{m,k}}$ in order to be able to make a decision, with probability at least $1 - \delta_{S_{m,k}}$.

This forms the underlying principal for the creation of a stopping rule [10], [14]. Let $\delta \in (0, 1)$ be a user specified input parameter. The idea is to sequentially increase the size of $S_{m,k}$ while at the same time checking if the stopping criterion, $|\Lambda_1^{S_{m,k}}(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*) - \psi_1(u, \boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1})| > c_{S_{m,k}}$, is met. If the stopping criterion is never met, then this will result in $S_{m,k} = M_k$, i.e requiring the evaluation of all the measurements. Selecting $\delta_{S_{m,k}} = \frac{p-1}{pS_{m,k}^p}\delta$ results in $\sum_{S_{m,k} \ge 1} \delta_{S_{m,k}} \le \delta$. The event

$$\mathcal{E} = \bigcap_{S_{m,k} \ge 1} \left\{|\Lambda_1^{S_{m,k}}(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*) - \Lambda_1^{M_k}(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*)| \le c_{S_{m,k}}\right\} \quad (11)$$

thus holds with probability at least $1 - \delta$ by a union bound argument.

This iterative procedure allows for an adaptive size of the number of measurements required to be evaluated. However, there is cause for concern with the definition of the stopping rule. That is the fact that the range, R_k, used in the calculation of (9), is dependent on the log likelihood for all M_k measurements. Calculating this range would thus inherently require at least the same number of calculations as in the standard sequential MCMC approach. In certain applications it may be possible to obtain an expression for the range which is independent of the measurements, however, this is not the case for the current application of interest. In order to overcome the computational complexity of the calculation of the range, and to reduce the sample variance $V_{S_{m,k}}$ in the bound, a control variate has been introduced in [11], referred to as a proxy:

$$\wp_i(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*) \approx \log\left[\frac{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^*)}{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^{m-1})}\right]. \quad (12)$$

Thus the MC approximation in (7) is augmented into

$$\Lambda_1^{S_{m,k}}(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*) = \frac{1}{S_{m,k}} \sum_{i=1}^{S_{m,k}} \log\left[\frac{p(\boldsymbol{z}_k^{i,*}|\boldsymbol{x}_k^*)}{p(\boldsymbol{z}_k^{i,*}|\boldsymbol{x}_k^{m-1})}\right] - \wp_i(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*). \quad (13)$$

It is required to amend the MH acceptance accordingly to take the inclusion of the proxy into account.

We propose using a first order Taylor series as an approximation for the log likelihood, $\ell_i(\boldsymbol{x}) = \log p(\boldsymbol{z}^i|\boldsymbol{x})$, given as

$$\hat{\ell}_i(\boldsymbol{x}) = \ell_i(\boldsymbol{x}^+) + (\nabla \ell_i)_{\boldsymbol{x}^+}^T \cdot (\boldsymbol{x} - \boldsymbol{x}^+), \quad (14)$$

where $(\nabla \ell_i)_{\boldsymbol{x}^+}$ represents the gradient of $\ell_i(\boldsymbol{x})$ evaluated at $\boldsymbol{x}^+$. This results in the following form of the proxy

$$\wp_i(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*) = \hat{\ell}_i(\boldsymbol{x}_k^*) - \hat{\ell}_i(\boldsymbol{x}_k^{m-1}), = (\nabla \ell_i)_{\boldsymbol{x}^+}^T \cdot (\boldsymbol{x}_k^* - \boldsymbol{x}_k^{m-1}). \quad (15)$$

With the inclusion of the proxy, the range, R_k, is now computed as,

$$R_k = \max_{1\leq i\leq M_k}\left\{\frac{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^*)}{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^{m-1})} - \wp_i(\boldsymbol{x}_k^{m-1},\boldsymbol{x}_k^*)\right\} - \min_{1\leq i\leq M_k}\left\{\frac{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^*)}{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^{m-1})} - \wp_i(\boldsymbol{x}_k^{m-1},\boldsymbol{x}_k^*)\right\}. \quad (16)$$

We can derive an upper bound for the range, R_k^B, i.e where $R_k^B \geq R_k$, which can be computed efficiently

$$\begin{aligned} R_k^B &= 2\max_{1\leq i\leq M_k}\left\{\left|\log\left[\frac{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^*)}{p(\boldsymbol{z}_k^i|\boldsymbol{x}_k^{m-1})}\right] - \wp_i(\boldsymbol{x}_k^{m-1},\boldsymbol{x}_k^*)\right|\right\} \\ &= 2\max_{1\leq i\leq M_k}\left\{\left|\ell_i(\boldsymbol{x}_k^*) - \ell_i(\boldsymbol{x}_k^{m-1}) - \hat{\ell}_i(\boldsymbol{x}_k^*) + \hat{\ell}_i(\boldsymbol{x}_k^{m-1})\right|\right\} \\ &= 2\max_{1\leq i\leq M_k}\left\{\left|B_k(\boldsymbol{x}_k^{m-1}) - B_k(\boldsymbol{x}_k^*)\right|\right\} \end{aligned} \quad (17)$$

where $B_k(\boldsymbol{x}) = \ell_i(\boldsymbol{x}) - \hat{\ell}_i(\boldsymbol{x})$ is the remainder of the Taylor approximation. The Taylor-Lagrange inequality gives us an upper bound on the remainder term. More specifically, if $|f^{(n+1)}(\boldsymbol{x})| \leq Y$, then $|B_k(\boldsymbol{x})| \leq \frac{Y|\boldsymbol{x}-\boldsymbol{x}^+|^{n+1}}{(n+1)!}$, where in our case $n+1=2$. Upper bounding the Taylor remainder finally results in the following upper bound on the range

$$R_k^B = 2\left|\left|B_k(\boldsymbol{x}_k^{m-1})\right| + \left|B_k(\boldsymbol{x}_k^*)\right|\right|, \quad (18)$$

which is dependent on the maximum of the Hessian of the log likelihood, Y. The complete adaptive subsampling sequential MCMC approach is illustrated by Algorithms 2 and 3.

III. Application to Target Tracking in Complex Systems

A. Target and Sensor Modelling

In this application the state vector consists of the position and velocity of the target in a two dimensional space, $\boldsymbol{x}_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^T$.The target motion prediction is performed according to the near constant velocity model. This results in the state transition density having the form

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{A}_k\boldsymbol{x}_{k-1}, \boldsymbol{Q}_k), \quad (19)$$

where $\mathcal{N}(\cdot)$ represents the normal distribution, and matrices $\boldsymbol{A}_k$ and $\boldsymbol{Q}_k$ are defined as $\boldsymbol{A}_k = \begin{bmatrix} \boldsymbol{I}_2 & T_s\boldsymbol{I}_2 \\ \boldsymbol{0}_2 & \boldsymbol{I}_2 \end{bmatrix}$ and $\boldsymbol{Q}_k = \sigma_x^2\begin{bmatrix} (T_s^3/3)\boldsymbol{I}_2 & (T_s^2/2)\boldsymbol{I}_2 \\ (T_s^2/2)\boldsymbol{I}_2 & T_s\boldsymbol{I}_2 \end{bmatrix}$, where $T_s = t_k - t_{k-1}$.

In this application, the total number of measurements received is given by $M_k = M_k^x + M_k^c$, where M_k^x represents the number of target measurements, and M_k^c represents the number of clutter measurements. The number of target and clutter measurements are Poisson distributed with mean λ_X and λ_C respectively. The likelihood density thus takes the form [15]:

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k) \propto \prod_{i=1}^{M_k}\lambda_X p_X(\boldsymbol{z}_k^i|\boldsymbol{x}_k) + \lambda_C p_C(\boldsymbol{z}_k^i), \quad (20)$$

Algorithm 2 Adaptive Subsampling Sequential Markov Chain Monte Carlo

Initialize particle set: $\{\boldsymbol{x}_0^{(j)}\}_{j=1}^{N_p}$
Determine initial proxy parameters. (See Section III-B for more details.)
for k = 1,...,T **do**
  for m = 1,...,N **do**
    Update proxy parameters. (See Section III-B for more details.)
    Joint Draw
    Propose $\{\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*\} \sim q_1\left(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}\right)$
    Compute $\psi_1(u, \boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1})$
    $= \frac{1}{M_k}\log\left[u\frac{p(\boldsymbol{x}_k^{m-1}|\boldsymbol{x}_{k-1}^{m-1})p(\boldsymbol{x}_{k-1}^{m-1}|\boldsymbol{z}_{1:k-1})}{p(\boldsymbol{x}_k^*|\boldsymbol{x}_{k-1}^*)p(\boldsymbol{x}_{k-1}^*|\boldsymbol{z}_{1:k-1})}\times \frac{q_1\left(\boldsymbol{x}_k^*,\boldsymbol{x}_{k-1}^*|\boldsymbol{x}_k^{m-1},\boldsymbol{x}_{k-1}^{m-1}\right)}{q_1\left(\boldsymbol{x}_k^{m-1},\boldsymbol{x}_{k-1}^{m-1}|\boldsymbol{x}_k^*,\boldsymbol{x}_{k-1}^*\right)}\right]$
    Compute $\Lambda_1^{S_{m,k}}(\boldsymbol{x}_k^*, \boldsymbol{x}_k^{m-1})$ and $\{\wp_i(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*)\}_{i=1}^{M_k}$ with the routine described by Algorithm 3.
    if $\Lambda_1^{S_{m,k}}(\boldsymbol{x}_k^*, \boldsymbol{x}_k^{m-1}) > \psi_1(u, \boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}) - \frac{1}{M_k}\sum_{i=1}^{M_k}\wp_i(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*)$ **then**
      $\{\boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\} = \{\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*\}$
    else
      $\{\boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\} = \{\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}\}$
    end if
    Refinement
    Propose $\{\boldsymbol{x}_k^*\} \sim q_2\left(\boldsymbol{x}_k|\boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right)$
    Compute $\psi_2(u, \boldsymbol{x}_k^*, \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m)$
    $= \frac{1}{M_k}\log\left[u\frac{p(\boldsymbol{x}_k^m|\boldsymbol{x}_{k-1}^m)q_2\left(\boldsymbol{x}_k^*|\boldsymbol{x}_k^m,\boldsymbol{x}_{k-1}^m\right)}{p(\boldsymbol{x}_k^*|\boldsymbol{x}_{k-1}^m)q_2\left(\boldsymbol{x}_k^m|\boldsymbol{x}_k^*,\boldsymbol{x}_{k-1}^m\right)}\right]$
    Compute $\Lambda_2^{S_{m,k}}(\boldsymbol{x}_k^m, \boldsymbol{x}_k^*)$ and $\{\wp_i(\boldsymbol{x}_k^m, \boldsymbol{x}_k^*)\}_{i=1}^{M_k}$ with the routine described by Algorithm 3.
    if $\Lambda_2^{S_{m,k}}(\boldsymbol{x}_k^*, \boldsymbol{x}_k^m) > \psi_2(u, \boldsymbol{x}_k^*, \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m) - \frac{1}{M_k}\sum_{i=1}^{M_k}\wp_i(\boldsymbol{x}_k^m, \boldsymbol{x}_k^*)$ **then**
      $\{\boldsymbol{x}_k^m\} = \{\boldsymbol{x}_k^*\}$
    end if
    if $m > N_{burn}$ **then**
      $\boldsymbol{x}_k^{(m-N_{burn})} = \boldsymbol{x}_k^m$
    end if
  end for
end for
$\hat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{1}{N_p}\sum_{j=1}^{N_p}\delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(j)})$

where $p_X(\cdot)$ and $p_C(\cdot)$ represent the likelihood of the target and clutter measurements respectively. Each individual measurement represents a point in the two dimensional observation space, $\boldsymbol{z}_k^i = \left[z_{x,k}^i, z_{y,k}^i\right]^T$. In the case of a measurement from the target, the likelihood is modelled as $p_X(\boldsymbol{z}_k^i|\boldsymbol{x}_k) = \mathcal{N}(\boldsymbol{z}_k^i; \boldsymbol{x}_k, \Sigma)$. The clutter measurements are independent of the state of the target and are uniformly distributed in the visible region of the sensor, resulting in the clutter likelihood taking the form of $p_C(\boldsymbol{z}_k^i) = U_{R_x}(z_{x,k}^i)U_{R_y}(z_{y,k}^i)$.

Algorithm 3 Adaptive Subsampling Routine

1: Given: The current and proposed states of the Markov chain, $\{\boldsymbol{x}_k,\ \boldsymbol{x}_k^*\}$, the complete measurement set, $\boldsymbol{z}_k = \{\boldsymbol{z}_k^1, ..., \boldsymbol{z}_k^{M_k}\}$, δ, and $\psi(\cdot)$.
2: Initialise: Number of sub-sampled measurements, $S_{m,k} = 0$, Approximate log likelihood ratio subtracted by proxy, $\Lambda = 0$, set of sub-sampled measurements, $\boldsymbol{z}_k^* = \emptyset$, initial batchsize, $b = 1$, while loop counter, $w = 0$.
3: Compute an upper bound for the range, R_k^B, according to (18).
4: Compute the proxy, $\{\wp_i(\boldsymbol{x}_k, \boldsymbol{x}_k^*)\}_{i=1}^{M_k}$, according to (15).
5: DONE = FALSE
6: **while** DONE == FALSE **do**
7: $w = w + 1$
8: $\{\boldsymbol{z}_k^{S_{m,k}+1,*}, ..., \boldsymbol{z}_k^{b,*}\} \sim_{w/repl.} \boldsymbol{z}_k \setminus \boldsymbol{z}_k^*$
9: $\boldsymbol{z}_k^* = \boldsymbol{z}_k^* \cup \{\boldsymbol{z}_k^{S_{m,k}+1,*}, ..., \boldsymbol{z}_k^{b,*}\}$
10: $\Lambda = \frac{1}{b}\Big(S_{m,k}\Lambda + \sum_{i=S_{m,k}+1}^{b}\Big[\log\frac{p(\boldsymbol{z}_k^{i,*}|\boldsymbol{x}_k^*)}{p(\boldsymbol{z}_k^{i,*}|\boldsymbol{x}_k)} - \wp_i(\boldsymbol{x}_k, \boldsymbol{x}_k^*)\Big]\Big)$
11: $S_{m,k} = b$
12: $\delta_w = \frac{p-1}{pw^p}\delta$
13: Compute c according to (9) utilising δ_w.
14: $b = \gamma S_{m,k} \wedge M_k$
15: **if** $|\Lambda + \frac{1}{M_k}\sum_{i=1}^{M_k}\wp_i(\boldsymbol{x}_k, \boldsymbol{x}_k^*) - \psi(\cdot)| \geq c$ **or** $S_{m,k} == M_k$ **then**
16: DONE = TRUE
17: **end if**
18: **end while**
19: **return** Λ and $\{\wp_i(\boldsymbol{x}_k, \boldsymbol{x}_k^*)\}_{i=1}^{M_k}$

The Taylor approximations used by the proxy in (15) are dependent on the gradient and Hessian of the log likelihood for individual measurements. Substituting the terms for the target and clutter likelihood in (20) and taking the logarithm results in the log likelihood for each measurement having the form

$$\ell_i(\boldsymbol{x}_k) = \log\left(\lambda_X \mathcal{N}(\boldsymbol{z}_{\boldsymbol{k}}^{\boldsymbol{i}}; \boldsymbol{x}_k, \boldsymbol{\Sigma}) + \frac{\lambda_C}{A_C}\right), \tag{21}$$

where $A_C = R_x \times R_y$ represents the clutter area. The gradient can then be computed as

$$\nabla\ell_i = \frac{\lambda_X \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_k^i - \boldsymbol{x}_k)\mathcal{N}(\boldsymbol{z}_k^i; \boldsymbol{x}_k, \boldsymbol{\Sigma})}{\lambda_X \mathcal{N}(\boldsymbol{z}_k^i; \boldsymbol{x}_k, \boldsymbol{\Sigma}) + \frac{\lambda_C}{A_C}}, \tag{22}$$

and the Hessian is given by

$$\boldsymbol{H} = \frac{-\lambda_X \boldsymbol{\Sigma}^{-1}\mathcal{N}(\boldsymbol{z}_k^i; \boldsymbol{x}_k, \boldsymbol{\Sigma})\left(\lambda_X \mathcal{N}(\boldsymbol{z}_k^i; \boldsymbol{x}_k, \boldsymbol{\Sigma}) + \frac{\lambda_C}{A_C}\right)}{\left(\lambda_X \mathcal{N}(\boldsymbol{z}_k^i; \boldsymbol{x}_k, \boldsymbol{\Sigma}) + \frac{\lambda_C}{A_C}\right)^2} + \frac{\frac{\lambda_C \lambda_X}{A_C}\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_k^i - \boldsymbol{x}_k)\left(\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_k^i - \boldsymbol{x}_k)\right)^T \mathcal{N}(\boldsymbol{z}_k^i; \boldsymbol{x}_k, \boldsymbol{\Sigma})}{\left(\lambda_X \mathcal{N}(\boldsymbol{z}_k^i; \boldsymbol{x}_k, \boldsymbol{\Sigma}) + \frac{\lambda_C}{A_C}\right)^2} \tag{23}$$

B. Implementation Considerations

The primary difference between the standard and adaptive subsampling sequential MCMC is that the latter requires less evaluations of the log likelihood. However, there are also additional computations which are introduced to achieve this. These calculations are minimal and typically performed for a fraction of the time spent on the calculation of the likelihood, when M_k is sufficiently large, and are thus considered negligible. In this section we discuss these computations in more detail.

The proxy, given in (15) is extremely efficient to compute in comparison to the log likelihood. This is conditioned on the availability of the gradient of the log likelihood (i.e. (22)) evaluated at a specific point. Currently, we only update this twice per time step (represented by line 5 in Algorithm 2). Once, at the beginning of a time step, where the specific point used is the predicted mean of the Markov chain at the previous time step. Secondly, the current state of the Markov chain after the burn in period. As the number of MCMC particles, N, is typically several magnitudes larger than 2, these calculations are considered negligible.

The calculation of an upper bound on the range in (18) is also extremely efficient to compute in comparison to the log likelihood. This is conditioned on the availability of the maximum of the Hessian in (23). In our application we found that the maximum of the Hessian is independent of the measurements and can hence be determined prior to the running of the algorithm (represented by line 2 in Algorithm 2).

The proposal distribution used for the joint draw step in the tracking scenario is defined as:

$$q_1\left(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})\frac{1}{N_p}\sum_{j=1}^{N_p}\delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(j)}). \tag{24}$$

The proposal distribution used for the refinement step in the tracking scenario is defined as:

$$q_2\left(\boldsymbol{x}_k|\boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right) = \mathcal{N}(\boldsymbol{x}_k^m, \boldsymbol{\Sigma}_q). \tag{25}$$

The refinement step represents a local move.

IV. Results

Consider the scenario of a target moving through a highly cluttered environment. A sensor monitoring the target returns multiple target and clutter measurements at each time step. We applied the standard and adaptive subsampling sequential MCMC algorithms for the inference of the latent states of the target over several experiments with different parameters.

Two different metrics are used to compare the performance of the algorithms. Firstly, the root mean square error (RMSE) of the position. The RMSE for each time step is calculated over a number of independent simulation runs according to

$$RMSE = \sqrt{\frac{1}{N_I}\sum_{i=1}^{N_I}(\hat{X}_i - X_i)^2}, \tag{26}$$

where X_i represents the ground truth, $\hat{X}_i$ represents the algorithm estimate, which corresponds to the mean of the N MCMC samples in this application, and N_I represents the number of independent runs. The RMSE of the states corresponding to the position are averaged to obtain a single result. The RMSE of the position illustrates the tracking accuracy of the two algorithms.

The second metric is the normalized number of sub-sampled measurements required for likelihood calculations.

$$D = \frac{1}{T}\sum_{k=1}^{T}\frac{\sum_{m=1}^{N}(S_{m,k})_{JD} + (S_{m,k})_{R}}{2NM_k} \tag{27}$$

where $(S_{m,k})_{JD}$ and $(S_{m,k})_{R}$ refer to the number of sub-sampled measurements from the joint draw step and refinement step respectively. The standard sequential MCMC algorithm requires to evaluate the likelihood $2NM_k$ times at each time step, this corresponds to $D = 1$. Thus the D value is only shown for the adaptive subsampling sequential MCMC algorithm. It illustrates the fraction of likelihood evaluations which are required at each time step versus the standard sequential MCMC algorithm.

A. Parameters

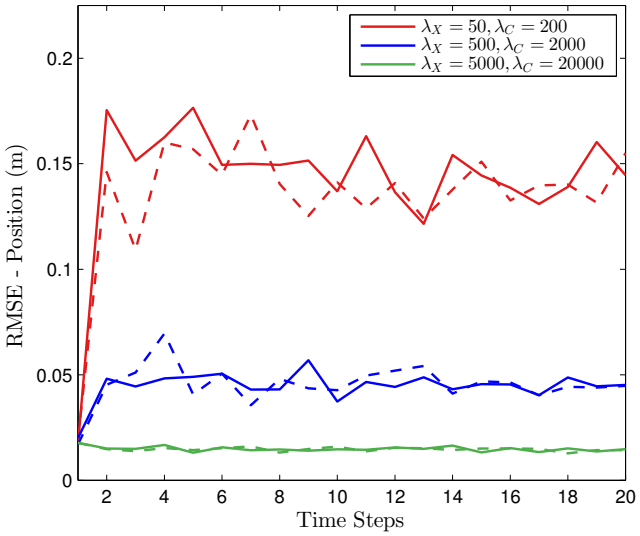
The following parameters, unless otherwise specified, were used for all experiments. Simulation parameters: $N = 500$, $N_{burn} = 125$, $T = 20$, $N_I = 50$, $\mathbf{\Sigma}_q = 0.01\boldsymbol{I}$. Motion model parameters: $T_s = 1$, $\sigma_x = 0.5$. Target observation model parameters: $\lambda_X = 500$, $\mathbf{\Sigma} = \boldsymbol{I}$. Clutter parameters: $\lambda_C = 2000$, $A_c = 4 \times 10^4$. Subsampling parameters: $\gamma = 1.2$, $\delta = 0.1$, $p = 2$.

B. Performance Evaluation

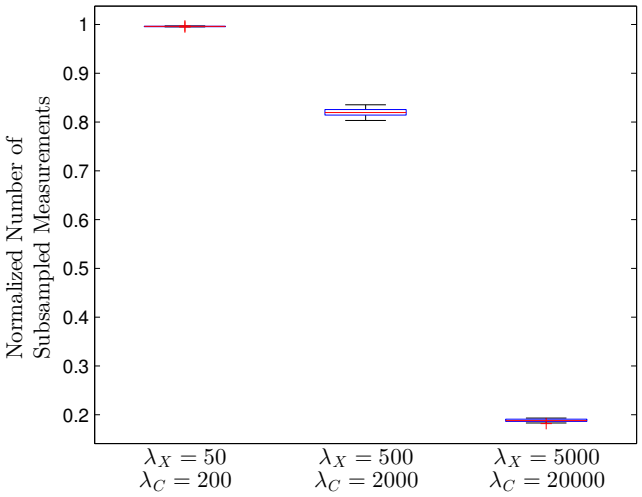
The first experiment illustrates the performance of the algorithms for different values of the mean total number of measurements in Fig. 1. The ratio between the mean number of clutter measurements and mean number of target measurements is fixed at 4:1. The RMSEs of the algorithms are in agreement, however, it is noted that an increase in the total mean number of measurements results in substantial computational savings. The amount of computational saving is as high as 80% with no significant loss in tracking performance.

In Fig. 2 the ratio between the mean number of clutter measurements and mean number of target measurements is varied. This allows for the observation of the performance when there is a varied amount of information about the target present in the measurements. The RMSE results show agreement between the two algorithms with an increase in computational savings when the mean number of target measurements is higher.

Fig. 3 illustrates the influence of varying the covariance matrix of the target observation model. The RMSEs of the two algorithms are in agreement. It is noted that a smaller computational saving is incurred as the measurement model becomes more precise. This result seems counter-intuitive. The reason for this is due to the Taylor approximation for the proxy. The upper bound for the range, R_k^B, becomes a weaker bound as the observation model becomes more peaked.



(a) RMSE comparison for different values of mean number of total measurements. The dotted lines represent the results from the standard sequential MCMC, and the full lines represent the results from the adaptive subsampling sequential MCMC.



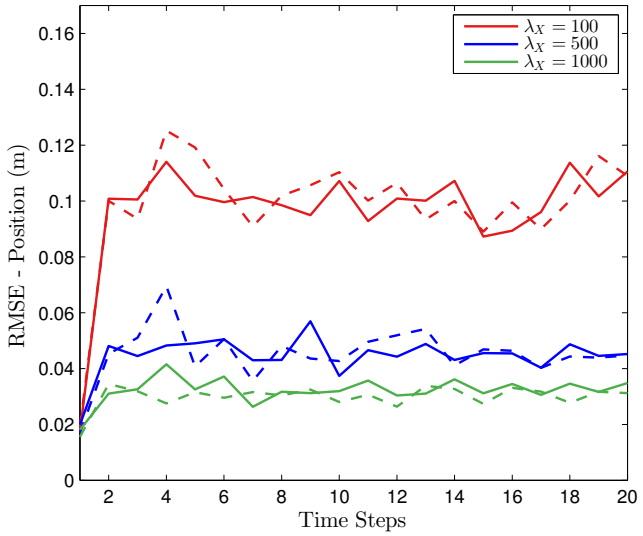
(b) Comparison of the normalized number of subsampled measurements evaluated in the adaptive subsampling sequential MCMC for different values of mean number of total measurements.

Fig. 1: Performance comparison for a different mean number of total measurements with a constant clutter to target measurement ratio of 4:1.

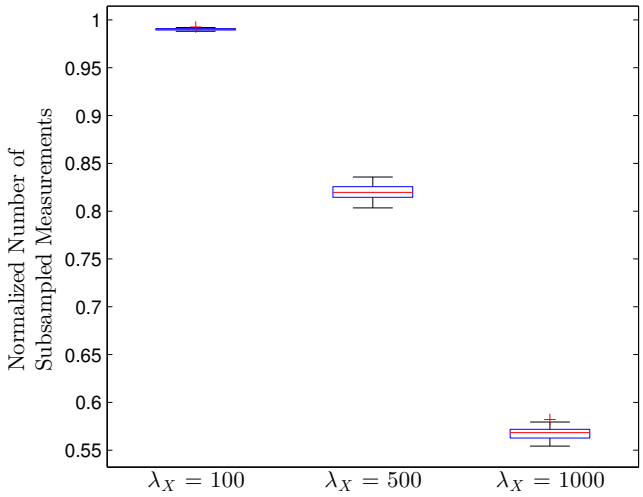
V. Conclusion

In this paper, we presented an adaptive subsampling sequential MCMC algorithm for target tracking. We have shown that this approach results in substantial computational savings when there is a large number of measurements, and most importantly, without sacrificing tracking performance.

There is a wide scope for future work. From an application perspective, considering a multi-target scenario with different levels of clutter, and also from an algorithmic perspective, further research on the influence and implementation of a more efficient proxy.

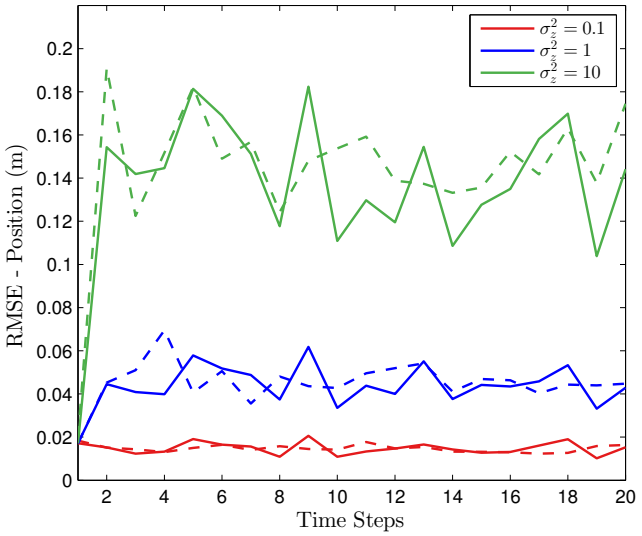


(a) RMSE comparison of different values of mean target measurements. The dotted lines represent the results from the standard sequential MCMC, and the full lines represent the results from the adaptive subsampling sequential MCMC.

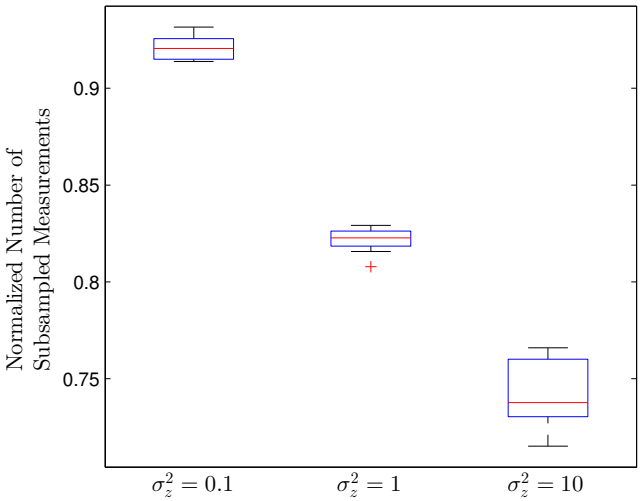


(b) Comparison of the normalized number of subsampled measurements evaluated in the adaptive subsampling sequential MCMC for different values of mean target measurements.

Fig. 2: Performance comparison for a different number of mean clutter to target measurements ratios.



(a) RMSE comparison for different measurement covariance matrices. The dotted lines represent the results from the standard sequential MCMC, and the full lines represent the results from the adaptive subsampling sequential MCMC.



(b) Comparison of the normalized number of subsampled measurements evaluated in the adaptive subsampling sequential MCMC for different covariance matrices.

Fig. 3: Performance comparison for different covariance matrices where $\boldsymbol{\Sigma} = \sigma_z^2 \boldsymbol{I}$.

ACKNOWLEDGMENTS

We would like to thank Rémi Bardenet for the constructive discussions on this work. We also acknowledge the support from the UK Engineering and Physical Sciences Research Council (EPSRC) via the Bayesian Tracking and Reasoning over Time (BTaRoT) grant EP/K021516/1 and EC Seventh Framework Programme [FP7 2013-2017] TRAcking in compleX sensor systems (TRAX) Grant agreement no.: 607400.

REFERENCES

[1] L. Mihaylova, A. Carmi, F. Septier, A. Gning, S. Pang, and S. Godsill, "Overview of Bayesian sequential Monte Carlo methods for group and extended object tracking," *Digital Signal Processing: A Review Journal*, vol. 25, no. 1, pp. 1–16, 2014.

[2] S. Balakrishnan and D. Madigan, "A one-pass sequential Monte Carlo method for Bayesian analysis of massive datasets," *Bayesian Anal.*, vol. 1, no. 2, pp. 345–361, June 2006.

[3] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F Radar and Signal Proc.*, vol. 140, no. 2, pp. 107–113, April 1993.

[4] W. R. Gilks and C. Berzuini, "Following a moving target-Monte Carlo inference for dynamic Bayesian models," *J. Royal Statist. Soc.: Series B*, vol. 63, no. 1, pp. 127–146, 2001.

[5] P. Djuric and M. Bugallo, "Particle filtering for high-dimensional systems," in *Proc. of the 5th IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Dec. 2013, pp. 352–355.

[6] A. Jasra, D. A. Stephens, and C. C. Holmes, "On population-based simulation for static inference," *Stat. Comput.*, vol. 17, no. 3, pp. 263–

279, 2007.

[7] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1805 –1819, Nov. 2005.

[8] F. Septier, S. K. Pang, A. Carmi, and S. Godsill, "On MCMC-Based particle methods for Bayesian filtering: Application to multitarget tracking," in *Proc. of the IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Dec. 2009, pp. 360–363.

[9] A. Korattikara, Y. Chen, and M. Welling, "Austerity in MCMC land: Cutting the Metropolis-Hastings Budget," in *Proc. of the Int. Conf. on Machine Learning*, 2014.

[10] R. Bardenet, A. Doucet, and C. Holmes, "Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach," in *Proc. of the Int. Conf. on Machine Learning*, 2014, pp. 405–413.

[11] ——, "Markov chain Monte Carlo and tall data," *preprint, http://arxiv.org/abs/1505.02827*, May 2015.

[12] J.-Y. Audibert, R. Munos, and C. Szepesvári, "Exploration-exploitation tradeoff using variance estimates in multi-armed bandits," *Theoretical Computer Science*, vol. 410, no. 19, pp. 1876 – 1902, 2009.

[13] R. Bardenet and O.-A. Maillard, "Concentration inequalities for sampling without replacement," *To appear in Bernoulli*, 2015. [Online]. Available: arxiv.org/abs/1309.4029

[14] V. Mnih, C. Szepesvári, and J.-Y. Audibert, "Empirical bernstein stopping," in *Proc. of the Int. Conf. on Machine Learning*, 2008, pp. 672–679.

[15] K. Gilholm and D. Salmond, "Spatial distribution model for tracking extended objects," *IEE Proc. Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 364–371, Oct. 2005.